KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA
2024

PENGEMBANGAN GIM

Rini Melati
Muhammad Ardhan Fadlurrahman

SMK/MAK Kelas XI-XII

Hak Cipta pada Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia
Dilindungi Undang-Undang

Penafian: Buku ini disiapkan oleh Pemerintah dalam rangka pemenuhan kebutuhan buku pendidikan yang bermutu, murah, dan merata sesuai dengan amanat dalam UU No. 3 Tahun 2017. Buku ini disusun dan ditelaah oleh berbagai pihak di bawah koordinasi Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi. Buku ini merupakan dokumen hidup yang senantiasa diperbaiki, diperbarui, dan dimutakhirkan sesuai dengan dinamika kebutuhan dan perubahan zaman. Masukan dari berbagai kalangan yang dialamatkan kepada penulis atau melalui alamat surel buku@kemdikbud.go.id diharapkan dapat meningkatkan kualitas buku ini.

Pengembangan Gim
untuk SMK/MAK Kelas XI-XII

Penulis
Rini Melati
Muhammad Ardhan Fadlurrahman

Penelaah
Asep Wahyudin
Ratna Wardani

Penyelia/Penyelaras
Supriyatno
Wijanarko Adi Nugroho
Erlina Indarti
Arifin Fajar Setia utama

Kontributor
Ruly Dwi Setyaningrum
Agus Indra Cahaya
Fadhli Hibatul Haqqi

Ilustrator
Yol Yulianto

Editor
Khofifa Najma Iftitah
Erlina Indarti

Editor Visual
Ilhamsyah

Desainer
Kiata Alma Setra

Penerbit
Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi

Dikeluarkan oleh
Pusat Perbukuan
Kompleks Kemdikbudristek Jalan RS. Fatmawati, Cipete, Jakarta Selatan
https://buku.kemdikbud.go.id

Cetakan pertama, 2024
ISBN 978-634-00-0360-4

Isi buku ini menggunakan huruf Noto Sans 10/18 pt, SIL Open Font License.
xx, 332 hlm.: 21 x 27 cm.

Kata Pengantar . . .

Pusat Perbukuan, Badan Standar, Kurikulum, dan Asesmen Pendidikan, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi memiliki tugas mengembangkan buku pendidikan di tingkat Pendidikan Anak Usia Dini, Pendidikan Dasar, dan Pendidikan Menengah. Penyusunan Buku Teks Utama ini mengacu pada Kurikulum Merdeka. Kurikulum tersebut memberikan keleluasaan kepada satuan pendidikan dalam melaksanakan pembelajaran sesuai dengan prinsip diversifikasi, memperhatikan kondisi masing-masing satuan pendidikan, potensi daerah, dan kebutuhan peserta didik.

Dalam mendukung implementasi Kurikulum Merdeka, pemerintah, melalui Pusat Perbukuan, mengembangkan buku siswa dan buku panduan guru sebagai sumber bahan pembelajaran. Buku-buku ini dapat dijadikan referensi atau inspirasi yang dapat dimodifikasi atau digunakan sebagai contoh, maupun rujukan dalam merancang dan mengembangkan pembelajaran yang sesuai dengan karakteristik, potensi, dan kebutuhan peserta didik. Buku ini disusun untuk mendukung siswa SMK agar memiliki pengetahuan, keterampilan, dan karakter yang relevan dan siap menghadapi tantangan dunia kerja. Buku ini berisi muatan/materi yang sesuai dengan kurikulum dan kebutuhan industri, sehingga peserta didik tidak hanya mendapatkan pengetahuan teori, tetapi juga mampu mengaplikasikan keterampilan secara langsung dalam kehidupan.

Sebagai dokumen yang terus berkembang, buku ini dapat diperbaiki dan disesuaikan dengan perkembangan keilmuan dan teknologi. Oleh karena itu, saran dan masukan dari guru, peserta didik, orang tua, dan masyarakat sangat diharapkan untuk pengembangan buku ini di masa mendatang. Buku ini diharapkan dapat memberikan inspirasi dan motivasi bagi seluruh pembaca untuk bersama-sama membangun pendidikan kejuruan yang berkualitas dan relevan dengan kebutuhan zaman. Pusat Perbukuan mengucapkan terima kasih kepada semua pihak yang telah berkontribusi dalam penyusunan buku ini, dan semoga buku ini bermanfaat, khususnya bagi peserta didik dan guru, dalam meningkatkan mutu pembelajaran.

Jakarta, Agustus 2024
Kepala Pusat,

Supriyatno
NIP 196804051988121001

Prakata . . .

Puji syukur penulis haturkan kepada Tuhan Yang Maha Esa atas segala rahmat dan anugerah-Nya, yang telah memberikan kesempatan bagi penulis untuk menyelesaikan buku pengembangan gim ini. Buku ini ditujukan sebagai sumber belajar bagi peserta didik SMK kelas XI dan XII fase F dalam mempelajari ilmu terkait pengembangan gim.

Industri gim di seluruh dunia, menurut informasi dari *Newzoo*, diperkirakan memiliki nilai estimasi mencapai 187,7 miliar USD. Di Indonesia sendiri, sekitar 43,7 juta orang tergolong sebagai pemain gim. Data ini menunjukkan betapa pesatnya pertumbuhan pasar gim di tanah air.

Saat ini, Indonesia menduduki peringkat ke-16 sebagai negara dengan pasar gim terbesar di dunia. Data dari Asosiasi Game Indonesia menunjukkan bahwa dalam dua tahun terakhir (2023-2024), industri gim mengalami perkembangan yang sangat signifikan, baik dari jumlah gim yang dihasilkan maupun munculnya perusahaan-perusahaan gim baru.

Perusahaan-perusahaan tersebut diharapkan dapat berperan sebagai agen budaya dan mampu bersaing di tingkat global. Sumber daya manusia (SDM) merupakan elemen penting dalam perkembangan industri ini. Oleh karena itu, buku ini diharapkan dapat memberikan kontribusi dalam meningkatkan kualitas SDM di bidang gim.

Buku ini disusun berdasarkan Capaian Pembelajaran untuk membekali peserta didik dengan pengetahuan, keterampilan, dan sikap (*hard skills* dan *soft skills)* yang diperlukan. Materi yang disajikan mencakup pemodelan gim, perancangan artistik, *prototyping*, implementasi fitur melalui pemrograman, penerapan aset audiovisual, serta pengujian kualitas produk gim.

Dengan penggunaan bahasa yang mudah dipahami dan dilengkapi pengayaan di setiap bab, buku ini diharapkan dapat membantu peserta didik memahami materi dengan lebih baik. Penulis berharap informasi yang disampaikan dalam buku ini akan memberikan wawasan mendalam bagi peserta didik kelas XI dan XII fase F.

Terakhir, penulis mengucapkan terima kasih kepada semua pihak yang telah berkontribusi dalam penyusunan buku ini. Kritik dan saran sangat diharapkan untuk perbaikan di masa depan. Semoga buku ini bermanfaat bagi peserta didik dan pembaca secara umum, serta dapat mendorong pengembangan industri gim di Indonesia melalui talenta-talenta muda yang unggul.

Jakarta, Agustus 2024

Penulis.

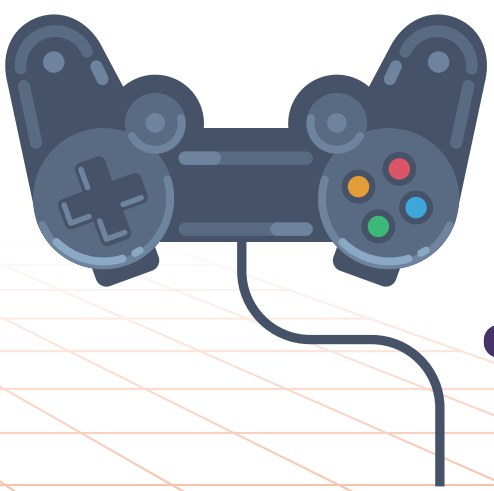
Daftar Isi . . .

Kata Pengantar iii
Prakata iv
Daftar Isi v
Daftar Gambar ix
Petunjuk Penggunaan Buku xvii



BAB 1

Pemodelan Gim 1
Tujuan Pembelajaran 2
Kata Kunci 2
Soft Skill 2
Peta Materi 3
A. Konsep Gim 4
B. Desain Gim 29
C. Game Design Document (GDD) 60
Uji Kompetensi 68
Pengayaan 71
Refleksi 72





BAB 2

Perancangan Artistik Gim **73**

Tujuan Pembelajaran 74

Kata Kunci 74

Soft Skill 74

Peta Materi 75

A. Desain Artistik (*Key Concept Art*) 76

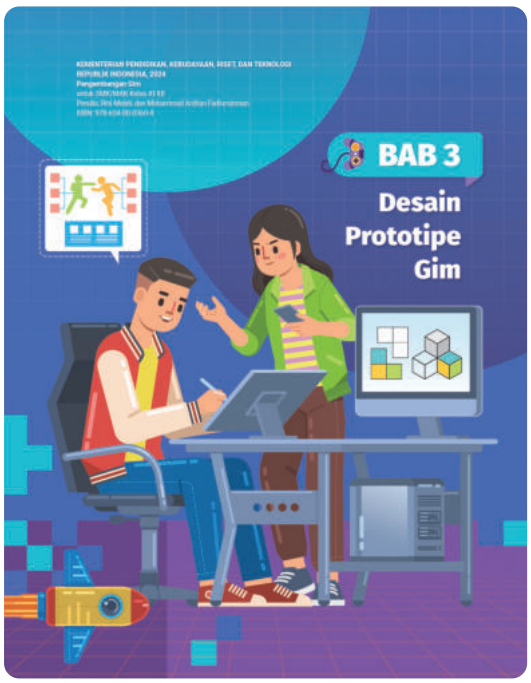
B. Aset Audio 92

C. Dokumen Desain Artistik (*Art Design Document*) 97

Uji Kompetensi 104

Pengayaan 105

Refleksi 106



BAB 3

Desain Prototipe Gim **107**

Tujuan Pembelajaran 108

Kata Kunci 108

Soft Skill 108

Peta Materi 108

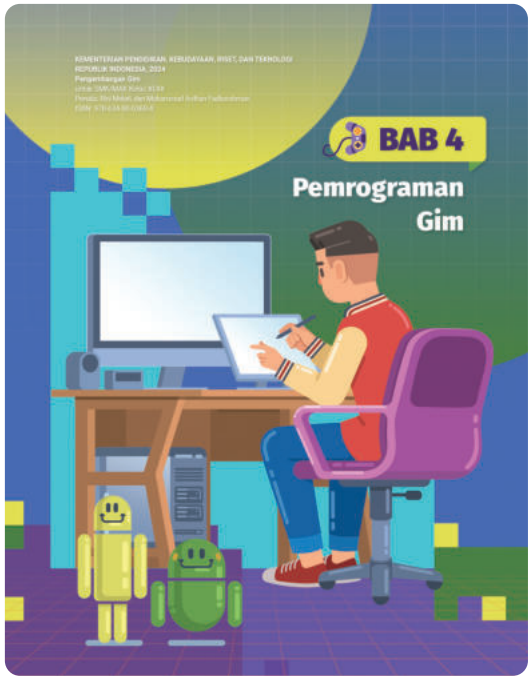
A. Desain Prototipe Gim (*Game Design Prototype*) 110

B. Keseimbangan Gim 130

Uji Kompetensi 149

Pengayaan 153

Refleksi 154



BAB 4

Pemrograman Gim **155**

Tujuan Pembelajaran 156

Kata Kunci 156

Soft Skill 156

Peta Materi 157

A. Struktur Data 159

B. Integrasi Objek 173

C. Fungsionalitas Tambahan 191

D. Pemrograman Berbasis Teks dan Grafis 194

Uji Kompetensi 216

Pengayaan 218

Refleksi 218

Bab 5

Aset Audio Visual **219**

Tujuan Pembelajaran 220

Kata Kunci 220

Soft Skill 220

Peta Materi 221

A. Aset Visual 2D 222

B. Aset Visual 3D 226

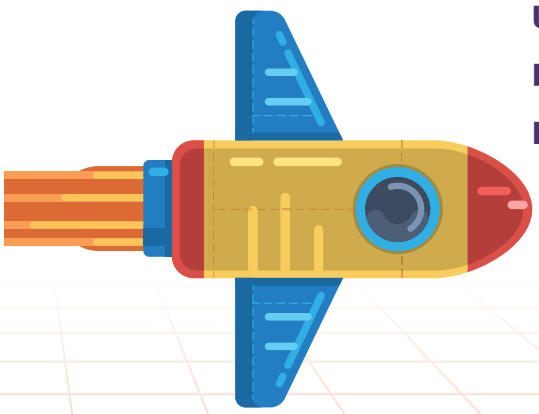
C. Aset Audio 258

D. *Dubbing* 270

Uji Kompetensi **277**

Pengayaan **277**

Refleksi **278**





BAB 6

Pengujian Gim **281**

Tujuan Pembelajaran 282

Kata Kunci 282

Soft Skill 282

Peta Materi 283

A. *Debugging* Gim 284

B. Optimasi Gim dan Pembaruan 295

Uji Kompetensi 305

Pengayaan 305

Refleksi 306

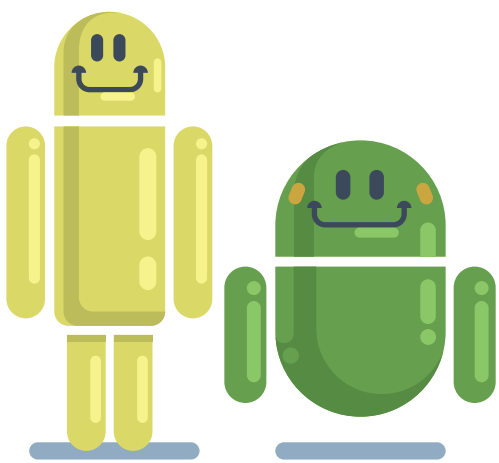
Glosarium **307**

Daftar Pustaka **310**

Daftar Sumber Gambar **315**

Index **322**

Profil Pelaku Perbukuan **324**



Daftar Gambar . . .

Gambar 1.1 Super Mario Odyssey *Cover Art* 6

Gambar 1.2 Contoh Tampilan Objektif Utama di dalam Gim 6

Gambar 1.3 Contoh Tampilan Objektif Tambahan Pada Gim Eksplorasi 7

Gambar 1.4 Contoh Barang Koleksi dalam Sebuah Gim *Platformer* 7

Gambar 1.5 Contoh Objektif Dengan Waktu di Sebuah Gim Balap 8

Gambar 1.6 Contoh Sistem Objektif Eksplorasi di dalam Gim 8

Gambar 1.7 Contoh Sistem Objektif Kooperatif di Gim MOBA 9

Gambar 1.8 Contoh Objektif Kemajuan di dalam Sebuah Gim 9

Gambar 1.9 Contoh Objektif Cerita di dalam Sebuah Gim 10

Gambar 1.10 Contoh Imbalan Poin/Skor di dalam Gim 11

Gambar 1.11 Contoh Imbalan Perlangkapan di dalam Gim 12

Gambar 1.12 Contoh Imbalan Poin Pengalaman di dalam Gim 12

Gambar 1.13 Imbalan *Unlockable* di dalam Gim 13

Gambar 1.14 Contoh Imbalan *Achievement* di dalam Gim 13

Gambar 1.15 Contoh Imbalan Kemajuan Cerita di dalam Gim 14

Gambar 1.16 Contoh Imbalan Mata Uang di dalam Gim 14

Gambar 1.17 Contoh Imbalan Kustomisasi di dalam Gim 15

Gambar 1.18 Contoh Gim Video (*Video Game*) 20

Gambar 1.19 Contoh Gim Aksi dari Pengembang Gim Indonesia bernama Kidbash. 21

Gambar 1.20 Contoh Gim Balap dari Pengembang Gim Indonesia bernama Baro Kart 22

Gambar 1.21 Contoh Gim Pertarungan Dari Pengembang Gim Indonesia bernama Battle of Guardian. 22

Gambar 1.22 Contoh Gim Teka Teki Bernama TETRIS. 23

Gambar 1.23 Contoh Gim Strategi *Real-Time* yang Memperlihatkan Dua Tipe Tentara yang Sedang Menyerang dan Bertahan 24

Gambar 1.24 Contoh Gim Strategi *Turn-Based* dari Pengembang Gim Indonesia Bernama Kriegsfront Tactics 24

Gambar 1.25 Contoh dari Sebuah Gim Papan 25
Gambar 1.26 Contoh *Combat Mechanics* dalam Sebuah Gim 32
Gambar 1.27 Contoh *Resource Management Mechanics* dalam Sebuah Gim 33
Gambar 1.28 Contoh *Progression Mechanics* dalam Sebuah Gim 34
Gambar 1.29 Contoh Cuplikan Narasi dari Gim *Last of Us* 40
Gambar 1.30 Contoh Cuplikan Narasi dari Gim *Bioshock* 40
Gambar 1.31 Contoh *Progression System* di dalam Gim Jetpack Joyride 2 43
Gambar 1.32 Contoh Lanjutan *Progression System* di dalam Gim Jetpack Joyride 2 43
Gambar 1.33 Sebuah representasi *Linear Level* di dalam Gim 51
Gambar 1.34 Sebuah representasi *Mission-Based Level* di dalam Gim 51
Gambar 1.35 Sebuah representasi *Puzzle Level* di dalam Gim 52
Gambar 1.36 Sebuah representasi *Stealth Level* di dalam Gim 52
Gambar 1.37 Sebuah representasi *Narrative Level* di dalam Gim 53
Gambar 2.1 Super Mario Odyssey 75
Gambar 2.2 Gim Balap Kart Rider 75
Gambar 2.3 Desain Karakter Seorang Bapak dalam Gim Mr. Alien Butuh Uang Jajan! 78
Gambar 2.4 Desain Lingkungan dalam Mr. Alien Butuh Uang Jajan! 79
Gambar 2.5 Desain Objek dan Item dalam Gim Mr. Alien Butuh Uang Jajan! 79
Gambar 2.6 Desain Efek Visual dalam Mr. Alien Butuh Uang Jajan! 80
Gambar 2.7 Contoh Desain Artistik yang Menggambarkan Dunia Berwarna Warni 81
Gambar 2.8 Representasi Desain Artistik yang Mencekam dan suasana yang Gelap 81
Gambar 2.9 Desain karakter Mr. Alien Butuh Uang Jajan! 82
Gambar 2.10 Desain Latar Belakang dalam Gim Mr. Alien Butuh Uang Jajan! 86
Gambar 2.11 Desain Properti dalam Gim Mr. Alien Butuh Uang Jajan 90
Gambar 2.12 Contoh Gim *Space for the Unbound* *103*
Gambar 3.1 Prototipe Gim 109
Gambar 3.2 Peralatan Prototipe Fisik 117
Gambar 3.3 Empat Lembar Kertas dan Gambar Kotak Berukuran 10 × 10 Pada Setiap Kotak 118

Gambar 3.4 Contoh Permainan Kapal Pada Lembar Kertas Kotak Kotak 119
Gambar 3.5 Manik-Manik dengan Warna Berbeda untuk Menandai Perbedaan Antara Hit dan Miss. 119
Gambar 3.6 Prototipe Digital Permainan Bola Biliard 126
Gambar 4.1 Representasi Gambar Sel Array 160
Gambar 4.2 Bentuk *Tree* Yang Direpresentasikan dalam Komputer 161
Gambar 4.3 Contoh State yang Merepresentasikan *Decision Tree* dalam Gim Mr. Alien 165
Gambar 4.4 *Sparse* Data Tersimpan dalam *Array* 169
Gambar 4.5 Tabel *Hash* yang Menunjukkan *Array* Dasar 170
Gambar 4.6 Tabel *Hash* yang Menunjukkan *Array* Dasar 171
Gambar 4.7 Dinding Sebagai Batas Ruang dan Strategi 174
Gambar 4.8 Pohon dan Vegetasi Sebagai Rute dan Estetika 174
Gambar 4.9 Bangunan Sebagai Konteks dan Interaksi Minimal 174
Gambar 4.10 Batu dan Rintangan Sebagai Tantangan dan Estetika 175
Gambar 4.11 Patung dan Monumen Sebagai Konteks Budaya dan Tujuan Visual 175
Gambar 4.12 Jalan dan Permukaan Lain Sebagai Navigasi dan Estetika 175
Gambar 4.13 Furnitur dan Objek Interior Sebagai Detail Lingkungan dan Pengaturan Cerita 175
Gambar 4.14 Purwarupa Gim Mr. Alien dengan Objek Statis 176
Gambar 4.15 Karakter Pemain Sebagai Interaksi Langsung dan Pengembangan Cerita 177
Gambar 4.16 Musuh Sebagai Tantangan Konflik dan AI Strategi 177
Gambar 4.17 Item Interaktif Berupa Pemotong Baut Sebagai Peningkatan Kemampuan dan Penyelesaian Teka-Teki 178
Gambar 4.18 Proyektil yang Berbentuk Lingkaran yang Mengenai Pemain Sebagai Interaksi Fisik dan *Gameplay* yang Dinamis 178
Gambar 4.19 Objek Dinamis Pada Gim Mr. Alien 178
Gambar 4.20 Contoh Latar Belakang dari Gim Mr. Alien 180
Gambar 4.21 Contoh Tampilan *Game Engine* untuk Proyek Gim Mr. Alien 180
Gambar 4.22 Memasukan Gambar ke dalam *Game Engine*. 181
Gambar 4.23 Cara Memasukan Aset dari Menu Impor Asset dalam Jendela Proyek 181

Gambar 4.24 Cara Memasukan Aset dari Menu Impor Asset dalam Jendela Proyek. 181
Gambar 4.25 Contoh Gambar Latar Telah Masuk ke dalam Proyek 182
Gambar 4.26 Contoh Gambar Latar Telah Masuk ke dalam *Hierarchy* Proyek 182
Gambar 4.27 Gambar Latar yang Telah Diberikan Input Tambahan dalam *Inspector*. 182
Gambar 4.28 Contoh Gambar Latar yang Sudah Berfungsi di dalam Proyek dan Berjalan dalam *Runtime*. 183
Gambar 4.29 Contoh Gambar yang Dipakai di dalam Mr. Alien 183
Gambar 4.30 Contoh Salah Satu Level di dalam Mr. Alien 184
Gambar 4.31 Objek Item Sudah Masuk ke Dalam *Game Engine* 184
Gambar 4.32 Objek Item Sudah Masuk ke dalam *Hierarchy* di *Game Engine*. 185
Gambar 4.33 Penjelasan Mengenai Komponen *Box Collider 2D* 185
Gambar 4.34 Contoh Box Collider Setiap Item di Gim *Mr. Alien*. 186
Gambar 4.35 Cara membuat Skrip Baru di dalam *Game Engine* 187
Gambar 4.36 Skrip Baru Telah Dibuat dan Dinamakan Sesuai Kebutuhan 187
Gambar 4.37 Cuplikan Kode pada skrip Mekanik_Koin 188
Gambar 4.38 Fungsi *OnCollisionEnter2D* untuk Menggunakan Fitur *Collider* 188
Gambar 4.39 Cara Menarik File *Script* ke dalam *Inspector* 189
Gambar 4.40 Tampilan Inspector setelah *Script* dimasukan. 189
Gambar 4.41 Dompet Telah Menjadi Objek Dinamis 190
Gambar 4.42 Tampilan Inspector untuk Mengganti *Variable Particle System* 191
Gambar 4.43 Contoh Kode Implementasi UI 192
Gambar 4.44 Contoh Memasukan *Script* UI di dalam *Button* 193
Gambar 4.45 Contoh Output setiap Klik Tombol akan muncul tulisan *Button Clicked* 193
Gambar 4.46 Contoh Pembuatan Proyek Baru di dalam *Game Engine Unity* 195
Gambar 4.47 Contoh Tampilan Jendela *Asset* di dalam *Game Engine* 196
Gambar 4.48 Contoh *Game Object* di dalam G*ame Engine*. 196
Gambar 4.49 Contoh Sebuah *Scene* di dalam *Game Engine* 197
Gambar 4.50 Jendela *Build Profile* di dalam *Game Engine*. 199

Gambar 4.51 Contoh Memasukan Aset Pembantu 200
Gambar 4.52 Contoh Melakukan *Setup Input Mapping* agar Tombol Input dapat di Deteksi Sistem 201
Gambar 4.53 Contoh Melakukan *Setup Input Mapping* yang Sesuai Dengan Pemodelan Gim 201
Gambar 4.54 Contoh Cara Membuat Object 2D *Circle Sprite* 201
Gambar 4.55 Contoh Mengganti *Sprite* dengan Animasi ataupun Gambar 2D 202
Gambar 4.56 Contoh Saat Memberikan *Collider* Agar Karakter Dapat Terdeteksi sistem Fisika 202
Gambar 4.57 Contoh Saat Menambahkan *Rigidbody* 202
Gambar 4.58 Contoh Karakter Mr. Alien Terdeteksi Sistem Fisika dan Gravitasi Pada *Game Engine* 203
Gambar 4.59 Membuat Skrip Baru Bernama *Player_Input* 204
Gambar 4.60 Contoh Akan Muncul Fungsi *Start*() dan *Update*() Ketika *Script* Dibuat 204
Gambar 4.61 Contoh Implementasi Skrip Baru Bernama *Player_Input* 205
Gambar 4.62 Implementasi *Awake*() yang Akan Dijalankan Sebelum Fungsi *Start*()/*Update*() 205
Gambar 4.63 Implementasi *OnEnable*() *and OnDisable*() 206
Gambar 4.64 Implementasi *Update*() untuk Melakukan Update Posisi Setiap Detik 206
Gambar 4.65 Contoh Penambahan Variabel *jumpHeight* dengan *Tipe Float* 206
Gambar 4.66 Contoh Penambahan Suatu Kode dalam Fungsi *Start*() 207
Gambar 4.67 Contoh Mr. Alien Sudah Dapat Melompat ke Atas 207
Gambar 4.68 Contoh Mr. Alien Sudah Dapat Melompati Lubang di Antara Dua Papan 207
Gambar 4.69 Contoh Visualisasi Sederhana dari Koin (Kuning) dan Musuh (Biru). 208
Gambar 4.70 Contoh Membuat Visualisasi *UI Text Score* dalam *Game Engine* 209
Gambar 4.71 Contoh Susunan Komponen *Canvas* sebagai *Parent Class* dari *UI Text*. 209
Gambar 4.72 Contoh Implementasi dari *UI Text* dengan Beberapa Opsi Pilihan 209
Gambar 4.73 Contoh Implementasi Sederhana *UI Text* yang Sudah Berfungsi Pada Layar 210
Gambar 4.74 Contoh Variable Text yang akan Melakukan Visualisasi *Scoretext* Pada Gim 210
Gambar 4.75 Contoh Implementasi *Awake*() Tambahan untuk *UI Text* 210
Gambar 4.76 Contoh Implementasi *Start*() Tambahan untuk *UI Text* 211
Gambar 4.77 Contoh Implementasi Variable Score 211

Gambar 4.78 Contoh Implementasi Fitur Untuk Melakukan Inisiasi Awal *Score* dengan *Variable Score*. 211
Gambar 4.79 Fungsi OnTriggerEnter2D(Collider2D collision) 212
Gambar 4.80 Contoh Awal Skornya 0 karena Baru Memulai Permainan 212
Gambar 4.81 Contoh Awal Skor Bertambah 213
Gambar 4.82 Contoh Variabel UI Teks Tambahan untuk *State "Gameover"*. 213
Gambar 4.83 Contoh Fungsi Tambahan untuk *State "GameOver"*. 213
Gambar 4.84 Contoh Skor Bertambah Ketika Mr Alien Menyentuh "Koin" Pada Gim 214
Gambar 4.85 Contoh Ketika Mr. Alien Menyentuh "Musuh" dan Permainan Berakhir 214
Gambar 5.1 Potongan Gambar *Puppeteer* 222
Gambar 5.2 Contoh 2D *Puppet* yang digabungkan 223
Gambar 5.3 Contoh sketsa karakter 223
Gambar 5.4 Contoh *Storyboard* Karakter Memesan Makan 224
Gambar 5.5 Contoh Ilustrasi Karakter 224
Gambar 5.6 Contoh Ekspresi Karakter 225
Gambar 5.7 *Diffuse Map* di *Unity* 227
Gambar 5.8 *Specular Map* di *Unity* 227
Gambar 5.9 *Diffuse Map* di *Unity* 227
Gambar 5.10 Contoh Material Batu Bata di *Game Engine* 228
Gambar 5.11 Contoh Material Batu Bata yang sudah dimasukan ke dalam *Game* Objek di dalam *Game Engine* 228
Gambar 5.12 Contoh U dan V untuk Melakukan *Mapping* Koordinat Material ke dalam Objek ..230
Gambar 5.13 UV *Mapping* di dalam Unity 231
Gambar 5.14 UV *Scalling* di Unity 231
Gambar 5.15 Contoh *Automate Rigging* dengan *Control Rig* di Unreal Engine 5 232
Gambar 5.16 Contoh Animasi Anjing di dalam *Game Engine* Unity 234
Gambar 5.17 Gambaran *Bone* dan Juga Dependensi 235
Gambar 5.18 Proses Pembuatan *cutscene* Menggunakan *Optical Motion Capture* 236
Gambar 5.19 Contoh Penggunaan *Inertial Motion Capture* 237

Gambar 5.20 Contoh Penggunaan *Facial Motion Capture* 237
Gambar 5.21 Objek Karakter yang Memerlukan Data *Motion Capture* 239
Gambar 5.22 Aktor Mempersiapkan Baju untuk Setup *Environment Motion Capture* 239
Gambar 5.23 Aktor Merekam Gerakan *Motion Capture* untuk Keperluan Scene Dialog 239
Gambar 5.24 *Motion Capture* yang Sudah Diintegrasikan ke Model Karakter 3D 239
Gambar 5.25 Komponen simulasi fisika bernama *Rigidbody 2D* 241
Gambar 5.26 Jenis Implementasi *Collider* di dalam *Game Engine Unity* 241
Gambar 5.27 Mengambil Botol di Gim *Half-Life*: Alyx dengan Simulasi Cairan. 242
Gambar 5.28 Gim yang Memakai Sistem Fisika 242
Gambar 5.29 Implementasi Komponen *Cloth* 244
Gambar 5.30 Contoh karakter dengan komponen *Cloth* 244
Gambar 5.31 Sebuah Objek 3D Menggunakan *Flat Shading* 247
Gambar 5.32 Sebuah Objek 3D Menggunakan *Gouraud Shading* 247
Gambar 5.33 Sebuah Objek Menggunakan *Phong Shading* 248
Gambar 5.34 *Physically Based Rendering* (PBR) Pada Konferensi *Unity* 249
Gambar 5.35 *Pipeline Rendering* Sebuah Gambar Tiga Dimensi 251
Gambar 5.36 Contoh Penggunaan *Vertex Shader* 252
Gambar 5.37 Contoh Objek 3D yang Belum Masuk ke dalam *Pixel Processor* Untuk *Fragment Shader* 252
Gambar 5.38 Contoh Objek 3D Yang Sudah Di Proses Pixel *Processor* Untuk *Fragment Shader* Tanpa Cahaya 252
Gambar 5.39 Contoh Objek 3D yang Sudah Diproses *Pixel Processor* untuk *Fragment Shader* Dengan Cahaya. 253
Gambar 5.40 Penggunaan *Shader* Pada Gim *Journey*. 253
Gambar 5.41 Penerapan Teknik *Shading* PBR Pada Gim Gran Turismo 7 254
Gambar 5.42 Contoh Gim yang Menggunakan *Shader* 254
Gambar 5.43 Contoh *Output* Pemrograman *Shader* di dalam *Game Engine*. 256
Gambar 5.44 Musik Kenny untuk Audio Gim 260
Gambar 5.45 Tampilan Awal dari Program Audacity 261

Gambar 5.46 Tampilan Awal dari Program wavacity.....261
Gambar 5.47 Tampilan Saat Menyatukan Beberapa Sampel Audio di Dalam Program Audacity.262
Gambar 5.48 Tampilan Saat Menyatukan Beberapa Sampel Audio di Dalam Program Wavacity 262
Gambar 5.49 Tampilan Saat Ingin Melakukan *Export* 263
Gambar 5.50 Tampilan Saat Ingin Melakukan *Export di Wavacity* 263
Gambar 5.51 Contoh Hasil Audio Export dari Program Audio yang Digunakan.....264
Gambar 5.52 Tampilan Ketika Membuka Hasil Audio pada Tahap Sebelumnya 265
Gambar 5.53 Tampilan Proyek Gim Mr. Alien di Dalam *Unity*.....265
Gambar 5.54 File Musik Latar Gim Sudah Dimasukan Ke dalam Proyek Gim Mr.Alien.....266
Gambar 5.55 Komponen *Audiosource* untuk Memainkan Audio di dalam Gim. 266
Gambar 5.56 Tampilan *Game Engine* saat Melakukan *Testing*.....267
Gambar 5.57 Gim *A Space for The Unbound*.....268
Gambar 5.58 Gim *Battle Royale*, ...269
Gambar 5.59 Dokumen Digital Skrip *AGNI Village of Calamity* 274
Gambar 5.60 Pengisi Suara Sedang Melakukan *Take*.....274
Gambar 5.61 Proses Editing dan Penyuntingan Oleh Pengembang Gim 274
Gambar 6.1 Contoh perbedaan Level of Detail sebuah objek ketika dilihat dari dekat (LOD 0) dan dari jauh (LOD 1).....294
Gambar 6.2 Contoh Program Profiler 300

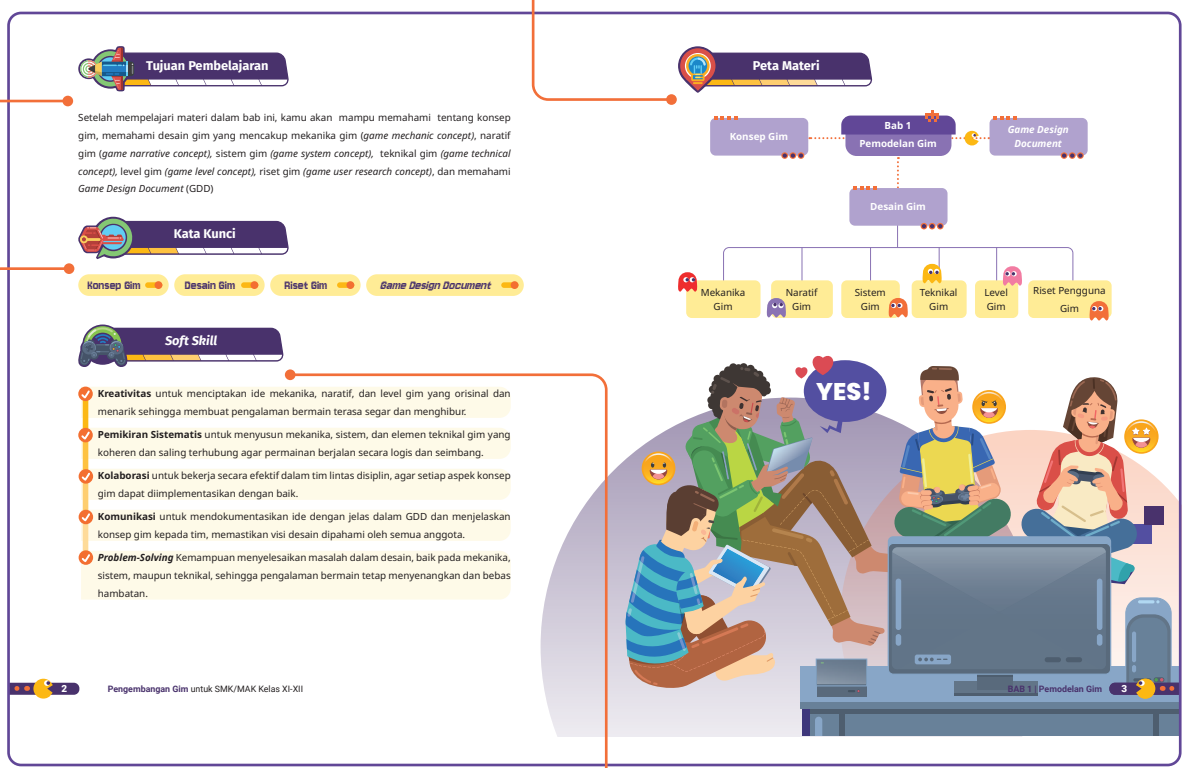
Petunjuk Penggunaan Buku . . .

Peta Materi

Peta materi menggambarkan materi apa saja yang akan dibahas dalam bab. Peserta didik harus mencermati terlebih dahulu peta materi ini agar mendapatkan gambaran tentang materi yang akan dibahas secara luas.

Tujuan Pembelajaran

Menggambarkan apa saja yang harus dicapai peserta didik dalam proses pembelajarannya.



Kata Kunci

Merupakan kata yang berisi informasi penting dalam materi buku, yang berfungsi untuk memudahkan peserta didik dalam menemukan informasi penting apa yang disampaikan dalam setiap bab.

Soft Skill

Soft skill akan menggambarkan kemampuan yang diharapkan dimiliki dan bisa diterapkan peserta didik saat melakukan pembelajaran.



Tentunya kamu pernah bermain gim, apakah itu menyenangkan? Pasti ada keseruan saat kamu bermain gim. Mengapa gim tersebut sangat seru dan menyenangkan saat dimainkan? Lalu, bagaimana caranya agar dapat membuat gim yang seru dan menyenangkan? Untuk membuat gim yang seru dan menyenangkan, kamu perlu mempelajari terlebih dahulu tentang pengembangan gim secara utuh, mencakup konsep gim, desain gim dan dokumen desain gim.

A. Konsep Gim

Saat kamu bermain gim, kamu akan merasakan hiburan yang sifatnya interaktif, melibatkan satu atau lebih pemain lainnya, dan semua aktivitas yang kamu lakukan dalam permainan akan mengikuti aturan-aturan yang ada untuk mencapai target atau tujuan dari gim tersebut. Kegiatan yang kamu lakukan ini adalah pengertian dari gim. Keunikan gim terletak pada interaksi antara pemain dan mekanisme permainan, yang memungkinkan pemain berpartisipasi aktif dan menentukan arah permainan. Interaktivitas ini merupakan elemen inti yang membedakan gim dengan bentuk hiburan lainnya. Kamu tidak hanya menjadi penonton, tapi kamu juga memiliki peran untuk mengendalikan gim, membuat keputusan yang bisa memberikan dampak langsung pada alur permainan yang kamu mainkan. Kondisi ini akan membuat kamu bisa merasakan adanya keterlibatan emosional yang mendalam dan kamu akan merasakan kepuasan dalam memainkan gim, karena kamu bisa mengatur dan menentukan pilihan saat bermain untuk menciptakan pengalaman yang berbeda dalam permainannya. (Katie Salen dan Eric Zimmerman, 2003)

Selain aktivitas, dalam gim juga terdapat aturan-aturan yang harus kamu ikuti. Aturan ini memiliki fungsi sebagai batasan yang menciptakan dinamika menarik, membantu pemain memahami bagaimana cara berinteraksi dengan elemen-elemen dalam gim.

Sebuah gim juga memiliki tujuan yang akan kamu capai dalam permainannya. Tujuan ini menjadi bagian penting dalam sebuah gim, dengan tujuan ini aktivitas yang kamu lakukan dalam gim akan lebih terarah. Tujuan yang jelas, baik itu besar ataupun kecil, sangat penting

Apersepsi

Bagian awal bab berisi penjelasan yang menghubungkan materi baru dengan pengetahuan dan pengalaman yang sudah dimiliki peserta didik, sebagai informasi pendahuluan sebelum guru menyampaikan materi berikutnya.

Aktivitas Belajar

Aktivitas belajar terdapat pada setiap subbab yang melatih kemampuan berpikir dan keterampilan peserta didik sesuai dengan materi. aktivitas belajar ini terdiri dari aktivitas literasi mandiri, aktivitas praktik mandiri dan tim kerja.

Aktivitas 1.6 **Literasi Mandiri**

Untuk menambah pemahaman kamu mengenai desain gim, lakukanlah aktivitas berikut ini secara mandiri:

1. Cari dan pilihlah salah satu gim yang sering kamu mainkan.
2. Kemudian kamu identifikasi dan tentukan mekanika gim, teknikal gim, naratif gim dan level gim serta sistem gim dari gim yang kamu pilih tadi.
3. Lengkapi lah tabel berikut ini:

Desain Gim	Deskripsi
Nama Gim	
Jenis/Genre Gim	
Mekanika Gim	
Teknikal Gim	
Naratif Gim	
Level Gim	
Sistem Gim	

4. Setelah itu kamu presentasikan hasil identifikasi kamu di depan kelas untuk saling berbagi informasi dengan teman lainnya.



Uji Kompetensi

Coba kamu jawab pertanyaan di bawah ini untuk mengetahui apakah kamu sudah menguasai materi tentang Pemodelan Gim.

1. Berikan tanda ceklis (✓) pada kolom benar atau salah dari pernyataan-pernyataan berikut ini :

No	Pernyataan	Benar	Salah
1	Aturan tidak hanya memberikan batasan tetapi juga menciptakan peluang bagi pemain untuk berstrategi.		
2	Gim adalah bentuk hiburan interaktif yang melibatkan satu atau lebih pemain dalam aktivitas dengan aturan tertentu dan tujuan yang jelas.		
3	Dengan menetapkan tujuan yang jelas, gim memberikan arah dan fokus bagi pemain, mendorong pemain untuk berinteraksi dengan dunia permainan. Tujuan ini tidak bisa bervariasi		

2. Jika kamu membuat sebuah gim, bagaimana hierarki tantangan dalam gim tersebut? Berapa banyak level yang kamu harapkan? Tantangan apa saja yang umum di setiap level?

3. Pasangkan pernyataan berikut dengan menarik garis dari titik ke arah elemen yang tepat!

Elemen ini memainkan peran penting dalam menjaga keterlibatan, motivasi, dan peluang bagi pemain untuk mengembangkan keterampilan dan strategi mereka.	Imbalan
Penghargaan yang diberikan kepada pemain setelah mereka berhasil mencapai tujuan atau mengatasi tantangan. Hal ini menciptakan rasa pencapaian dan memotivasi pemain untuk terus berinteraksi dengan permainan	Tujuan
Misi atau target yang harus dicapai oleh pemain dalam permainan. Dengan menetapkan tujuan yang jelas, gim memberikan arah dan fokus bagi pemain, mendorong pemain untuk berinteraksi dengan dunia permainan.	Tantangan

4. Empat elemen utama dalam gim bertujuan untuk menciptakan pengalaman bermain yang menarik dan menyenangkan. Elemen-elemen ini membentuk dasar dari desain gim dan membantu memastikan bahwa pemain ikut terlibat ataupun merasakan secara langsung. Berikut ini yang bukan elemen utama gim adalah ...
   a. Biaya
   b. Tujuan
   c. Aturan
   d. Tantangan
   e. Interaksi

5. Apakah kamu setuju atau tidak setuju dengan definisi gim? Jika kamu tidak setuju, apa yang akan kamu tambahkan, hapus, atau ubah?

Uji Kompetensi

Uji kompetensi ini berada pada bagian akhir bab, yang bertujuan agar peserta didik bisa mengukur kemampuannya dalam menguasai materi yang telah dibahas. Peserta didik akan diberikan soal dengan tingkat kesulitan yang berbeda mulai dari yang sederhana hingga kompleks.

Pengayaan

Pada bagian ini peserta didik akan diberikan informasi atau materi tambahan yang terkait dengan materi bab, serta dapat mengeksplorasinya untuk menambah wawasan. Informasi tersebut bisa berupa link atau pun literatur.

Pengayaan

Untuk memperdalam pemahaman kamu tentang pemodelan gim ini, kamu bisa mencari contoh atau bentuk lain dari sebuah GDD melalui internet. Kemudian kamu lakukan analisis mendalam tentang GDD tersebut, mulai dari tema, genre, mekanika, naratif dan lainnya. Kegiatan yang kamu lakukan ini akan membantu kamu memperkaya pemahaman tentang konsep gim.

Video berikut ini bisa kamu gunakan untuk menambah wawasan dan keterampilan kamu: *Game Development and Production*, Erik Bethke, Wordware Publishing, Inc., 2003

Referensi Video:

Pindai Aku!

1. **[Design] Skill Tree Online Episode 04: You'll Never Finish Your Game Design Document**
   https://www.youtube.com/watch?v=1xtw5y3NQIw
2. **Apa Itu Core Loop Pada Game?**
   https://www.youtube.com/watch?v=ApLYaWjLU3k
3. **Skill Tree Online #81 - Deconstructing Games as A Game Designer**
   https://www.youtube.com/watch?v=EWpJJzImPNA

Refleksi

Pada akhir bab, guru akan mengajak peserta didik untuk meninjau kembali apa yang sudah dipelajari serta melihat sejauh mana peserta didik mampu memahami dan menguasai materi.

Refleksi

Kamu bisa memberikan tanda centang (✓) pada kotak yang kamu anggap sesuai!

Setelah kamu mempelajari bab ini, bagaimanakah penguasaan kamu terhadap materi-materi berikut?

No	Materi	Tidak Menguasai	Menguasai	Sangat Menguasai
1	Konsep Gim			
2	Desain Gim • Mekanika Gim • Naratif Gim • Sistem Gim • Teknikal Gim • Level Gim • Riset Gim			
3	*Game Design Document* (GDD)			

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2024
Pengembangan Gim
untuk SMK/MAK Kelas XI-XII
Penulis: Rini Melati dan Muhammad Ardhan Fadlurrahman
ISBN: 978-634-00-0360-4

BAB 1

Pemodelan Gim



Tujuan Pembelajaran

Setelah mempelajari materi dalam bab ini, kamu akan mampu memahami tentang konsep gim, memahami desain gim yang mencakup mekanika gim (*game mechanic concept)*, naratif gim (*game narrative concept),* sistem gim *(game system concept),* teknikal gim *(game technical concept),* level gim *(game level concept),* riset gim *(game user research concept)*, dan memahami *Game Design Document* (GDD)

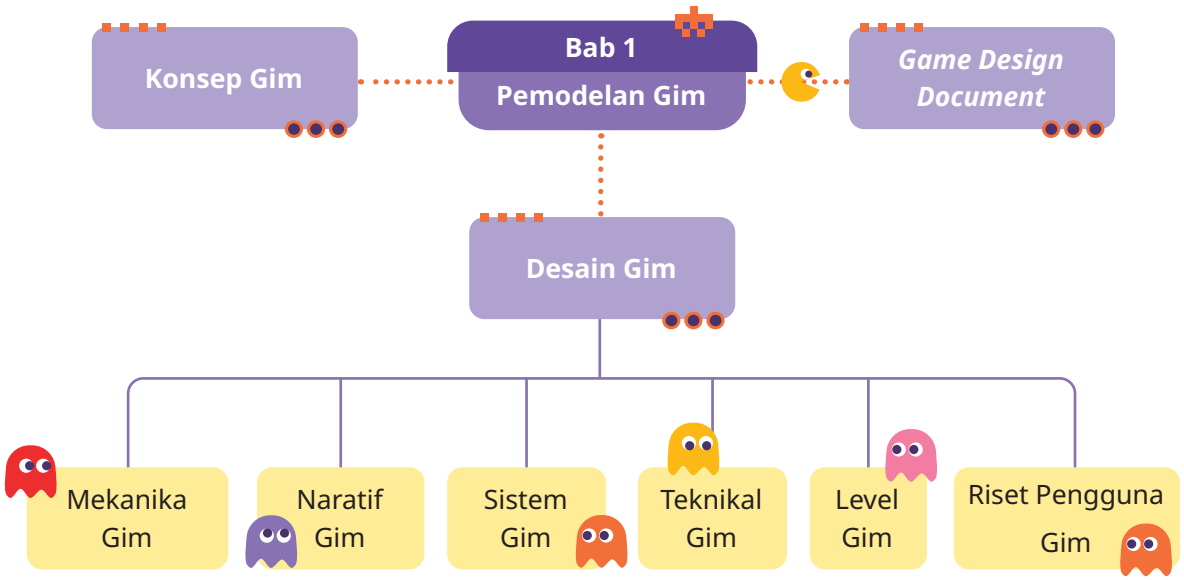
Kata Kunci

Konsep Gim **Desain Gim** **Riset Gim** ***Game Design Document***

Soft Skill

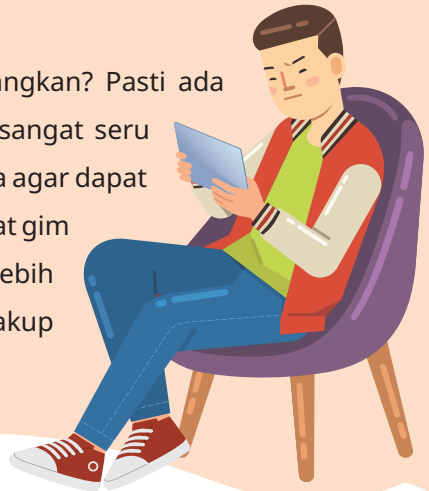
- **Kreativitas** untuk menciptakan ide mekanika, naratif, dan level gim yang orisinal dan menarik sehingga membuat pengalaman bermain terasa segar dan menghibur.
- **Pemikiran Sistematis** untuk menyusun mekanika, sistem, dan elemen teknikal gim yang koheren dan saling terhubung agar permainan berjalan secara logis dan seimbang.
- **Kolaborasi** untuk bekerja secara efektif dalam tim lintas disiplin, agar setiap aspek konsep gim dapat diimplementasikan dengan baik.
- **Komunikasi** untuk mendokumentasikan ide dengan jelas dalam GDD dan menjelaskan konsep gim kepada tim, memastikan visi desain dipahami oleh semua anggota.
- ***Problem-Solving*** Kemampuan menyelesaikan masalah dalam desain, baik pada mekanika, sistem, maupun teknikal, sehingga pengalaman bermain tetap menyenangkan dan bebas hambatan.

Peta Materi


Konsep Gim
Bab 1
Pemodelan Gim
Game Design Document
Desain Gim
Mekanika Gim
Naratif Gim
Sistem Gim
Teknikal Gim
Level Gim
Riset Pengguna Gim


YES!

Tentunya kamu pernah bermain gim, apakah itu menyenangkan? Pasti ada keseruan saat kamu bermain gim. Mengapa gim tersebut sangat seru dan menyenangkan saat dimainkan? Lalu, bagaimana caranya agar dapat membuat gim yang seru dan menyenangkan? Untuk membuat gim yang seru dan menyenangkan, kamu perlu mempelajari terlebih dahulu tentang pengembangan gim secara utuh, mencakup konsep gim, desain gim dan dokumen desain gim.



A. Konsep Gim

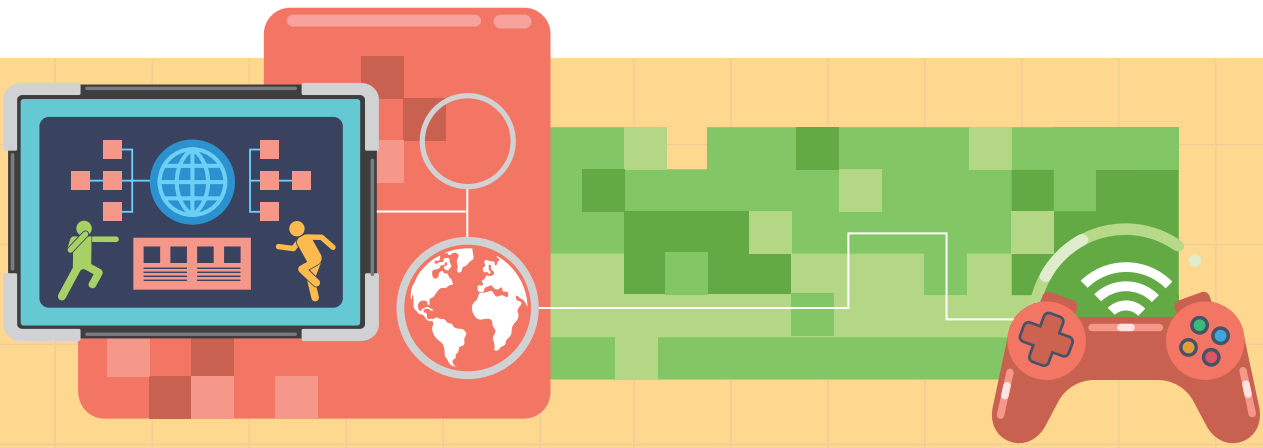
Saat kamu bermain gim, kamu akan merasakan hiburan yang sifatnya interaktif, melibatkan satu atau lebih pemain lainnya, dan semua aktivitas yang kamu lakukan dalam permainan akan mengikuti aturan-aturan yang ada untuk mencapai target atau tujuan dari gim tersebut. Kegiatan yang kamu lakukan ini adalah pengertian dari gim. Keunikan gim terletak pada interaksi antara pemain dan mekanisme permainan, yang memungkinkan pemain berpartisipasi aktif dan menentukan arah permainan. Interaktivitas ini merupakan elemen inti yang membedakan gim dengan bentuk hiburan lainnya. Kamu tidak hanya menjadi penonton, tapi kamu juga memiliki peran untuk mengendalikan gim, membuat keputusan yang bisa memberikan dampak langsung pada alur permainan yang kamu mainkan. Kondisi ini akan membuat kamu bisa merasakan adanya keterlibatan emosional yang mendalam dan kamu akan merasakan kepuasan dalam memainkan gim, karena kamu bisa mengatur dan menentukan pilihan saat bermain untuk menciptakan pengalaman yang berbeda dalam permainannya. (Katie Salen dan Eric Zimmerman, 2003)

Selain aktivitas, dalam gim juga terdapat aturan-aturan yang harus kamu ikuti. Aturan ini memiliki fungsi sebagai batasan yang menciptakan dinamika menarik, membantu pemain memahami bagaimana cara berinteraksi dengan elemen-elemen dalam gim.

Sebuah gim juga memiliki tujuan yang akan kamu capai dalam permainannya. Tujuan ini menjadi bagian penting dalam sebuah gim, dengan tujuan ini aktivitas yang kamu lakukan dalam gim akan lebih terarah. Tujuan yang jelas, baik itu besar ataupun kecil, sangat penting

untuk menjaga keterlibatan para pemain. Tujuan ini akan memberikan umpan balik yang positif dan juga bisa memberikan dorongan kepada pemain untuk terus memainkan gim tersebut. Oleh karena itu, gim yang dirancang dengan baik mampu menciptakan pengalaman yang mendalam, di mana pemain merasakan pertumbuhan dan kemajuan melalui interaksi yang mereka lakukan (Brenda Brathwaite dan Ian Schreiber, 2009)

1. Elemen Gim



Setelah kamu mengetahui pengertian gim dan penjelasannya di atas, menurut kamu elemen-elemen penting apa saja yang ada dalam sebuah gim? Berikut ini penjelasannya:

a. Interaktivitas

Dalam interaktivitas ini, menunjukkan adanya tindakan yang diambil dan dilakukan oleh pemain untuk bisa mempengaruhi jalannya permainan dan tujuan yang dicapai. Interaktivitas menciptakan pengalaman yang lebih mendalam karena pemain tidak hanya menjadi penonton tetapi juga pengendali cerita dan hasilnya. Dengan memberikan kendali kepada pemain, gim dapat menjadi medium yang sangat dinamis dan adaptif. **(Tracy Fullerton,2004)**

b. Aturan

Aturan merupakan elemen dasar dalam pembuatan sebuah gim. Aturan ini akan memberikan struktur atau batasan dalam permainannya. Aturan ini juga akan memberikan pemahaman yang sama bagi semua pemain tentang bagaimana interaksi dalam permainan tersebut. Jika tidak ada aturan yang jelas, maka sebuah gim akan sulit untuk dipahami dan dimainkan.

c. Tujuan



Gambar 1.1 Super Mario Odyssey *Cover Art*
Sumber: Nintendo (2017)

Coba kamu perhatikan gambar di atas. Gambar di atas adalah salah satu jenis gim yang memiliki tujuan dan misi yang akan dicapai di akhir permainannya. Tujuan ini menggambarkan misi atau target yang akan kamu capai dalam sebuah permainan. Dengan adanya tujuan yang jelas, permainan yang kamu mainkan akan lebih terarah dan fokus, kamu juga akan terdorong untuk terus memainkan gim dan terlibat dalam *gameplay*-nya. Banyak sekali bentuk dari tujuan yang ingin dicapai dalam sebuah gim seperti menyelesaikan level, mengumpulkan item atau mengalahkan musuh tertentu.

Berikut adalah beberapa jenis tujuan yang umum diterapkan dalam sebuah gim:

1. Objektif Utama (Main Objective)

Tujuan utama yang harus dicapai untuk menyelesaikan permainan, seperti menyelesaikan level atau mengalahkan bos akhir.



Gambar 1.2 Contoh Tampilan Objektif Utama di dalam Gim
Sumber: Auroch Digital/Ripstone (2018)

2. Objektif Tambahan (*Side Objective*)

Misi tambahan yang tidak wajib untuk menyelesaikan permainan, tetapi memberikan manfaat tambahan, seperti item atau cerita tambahan.



Gambar 1.3 Contoh Tampilan Objektif Tambahan Pada Gim Eksplorasi
Sumber: Nintendo (2023)



3. Barang Koleksi (*Collectibles*)

Mengumpulkan item tertentu dalam permainan, seperti koin, bintang, atau artefak, yang dapat memberikan bonus atau membuka konten baru.

Gambar 1.4 Contoh Barang Koleksi dalam Sebuah Gim *Platformer*
Sumber: Arzest/Nintendo (2014)

4. Objektif dengan Waktu (*Time-Limited Objectives*)

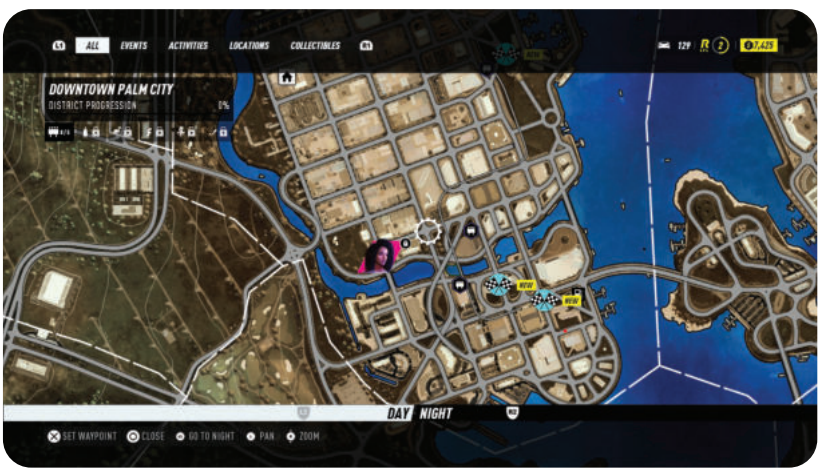
Tujuan yang harus dicapai dalam batas waktu tertentu, menciptakan tekanan dan tantangan tambahan bagi pemain.



Gambar 1.5 Contoh Objektif Dengan Waktu di Sebuah Gim Balap
Sumber: Nexon (2020)

5. Objektif Eksplorasi (*Exploration Objective*)

Mendorong pemain untuk menjelajahi dunia permainan dan menemukan lokasi atau rahasia tertentu.



Gambar 1.6 Contoh Sistem Objektif Eksplorasi di dalam Gim
Sumber: EA Gothenburg/Electronic Art (2019)

6. Objektif Kooperatif (*Cooperative Objective*)

Misi yang dapat dicapai dengan kerja sama antar pemain, sering ditemukan dalam gim *multiplayer*.



Gambar 1.7 Contoh Sistem Objektif Kooperatif di Gim MOBA
Sumber: Moonton (2016)

7. Objektif Kemajuan (*Progression Objective*)

Tujuan yang berfokus pada peningkatan karakter atau kemampuan, seperti *leveling up* atau meningkatkan senjata.



Gambar 1.8 Contoh Objektif Kemajuan di dalam Sebuah Gim
Sumber: Volition/Deep Silver (2017)

8. Objektif Cerita (*Story Objective*):

Misi yang terkait dengan pengembangan cerita, mengharuskan pemain untuk menyelesaikan tugas tertentu untuk melanjutkan narasi.



Gambar 1.9 Contoh Objektif Cerita di dalam Sebuah Gim
Sumber: Ubisoft Montreal/Ubisoft (2017)

d. Tantangan

Tantangan akan kamu temukan saat kamu bermain gim. Tantangan berfungsi sebagai rintangan atau kesulitan yang akan kamu hadapi dalam mencapai tujuan permainan. Tantangan yang baik tidak hanya membuatmu meningkatkan pengalaman bermain, tetapi juga memberikan kamu sebuah peluang untuk mengembangkan keterampilan dan strategi permainan. Jika tantangan ini tidak ada dalam sebuah permainan, maka kamu akan merasakan permainan yang monoton dan kurang memuaskan. (Tracy Fullerton, 2004)

e. Imbalan (*Reward*)

Saat kamu berhasil mencapai tujuan dari gim tersebut atau kamu berhasil menyelesaikan tantangan yang ada dalam gim, maka kamu akan mendapatkan penghargaan atau imbalan dari hasil yang kamu capai. Tentu saja hal ini akan membuat kamu merasa termotivasi untuk terus memainkan gim tersebut.

Ada beberapa bentuk imbalan yang bisa kamu dapatkan dalam sebuah gim, di antaranya poin, item baru, peningkatan karakter dan juga kemajuan dalam cerita gim tersebut. (Scott Rogers, 2008)

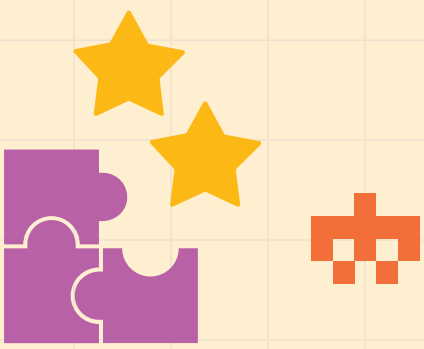
1. Poin atau Skor (*Points or Score*)



Poin ini bisa kamu peroleh saat kamu berhasil menyelesaikan tugas atau tantangan dalam gim yang kamu mainkan. Biasanya gim ini bersifat kompetitif untuk menunjukkan sebuah prestasi.



Gambar 1.10 Contoh Imbalan Poin/Skor di dalam Gim
Sumber: Pixel Perfect Dude S.A./Nintendo (2021)



2. Item atau Perlengkapan (*Items or Equipment*)

Item atau perlengkapan merupakan hadiah yang bisa kamu dapatkan dalam sebuah permainan, seperti item baru, senjata, armor, atau alat lainnya. Hadiah ini berfungsi untuk meningkatkan kemampuan karakter atau memberikan keuntungan dalam gim yang kamu mainkan.





Gambar 1.11 Contoh Imbalan Perlangkapan di dalam Gim
Sumber: Halfbrick Studio (2022)

3. Poin Pengalaman (Experience Points - XP)

Imbalan ini memungkinkan karakter yang kamu mainkan untuk naik level dalam gim, membuka kemampuan baru, atau meningkatkan atribut karakter yang dimainkan.





Gambar 1.12 Contoh Imbalan Poin Pengalaman di dalam Gim
Sumber: Volition/Deep Silver (2017)



Gambar 1.13 Imbalan *Unlockable* di dalam Gim
Sumber: Next Level Games/Nintendo (2013)

4. *Unlockables*

Konten tambahan yang dibuka setelah mencapai tujuan tertentu, seperti karakter baru, level, atau mode permainan.



5. Pencapaian atau Piala (*Achievements or Trophies*)

Imbalan lainnya yang bisa kamu dapatkan dalam sebuah permainan yaitu pengakuan atas keberhasilan dalam mencapai tujuan permainan. Hal ini sering kali dapat meningkatkan *replayability*.

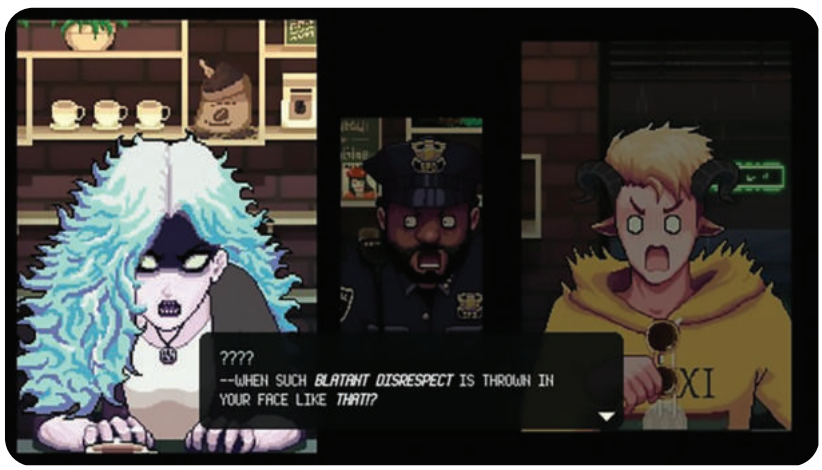




Gambar 1.14 Contoh Imbalan *Achievement* di dalam Gim
Sumber: Gearbox Software/2K (2014)

6. Kemajuan Cerita (*Story Progression*)

Ketika kamu berhasil menyelesaikan sebuah gim, kamu dapat memperoleh imbalan tambahan berupa pengungkapan bagian cerita atau kemajuan dalam alur cerita, yang membantu menyelesaikan misi atau tantangan dalam permainan.





Gambar 1.15 Contoh Imbalan Kemajuan Cerita di dalam Gim
Sumber: Toge Production (2023)

7. Mata Uang (*Currency*)

Imbalan berupa Mata uang ini juga bisa kamu dapatkan dalam permainan yang dapat digunakan untuk membeli item, *upgrade*, atau konten lainnya.

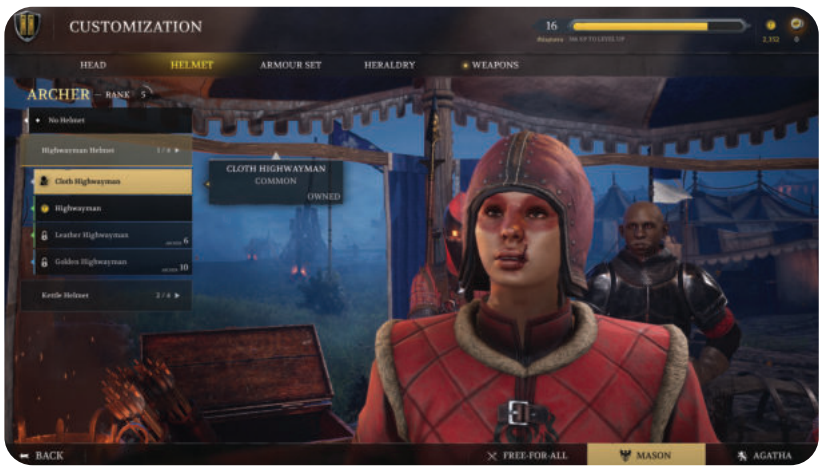


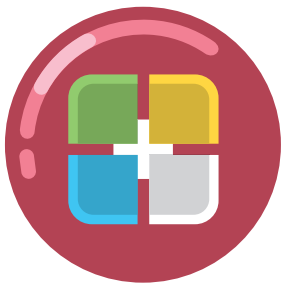


Gambar 1.16 Contoh Imbalan Mata Uang di dalam Gim
Sumber: Joyseed Gametribe (2020)

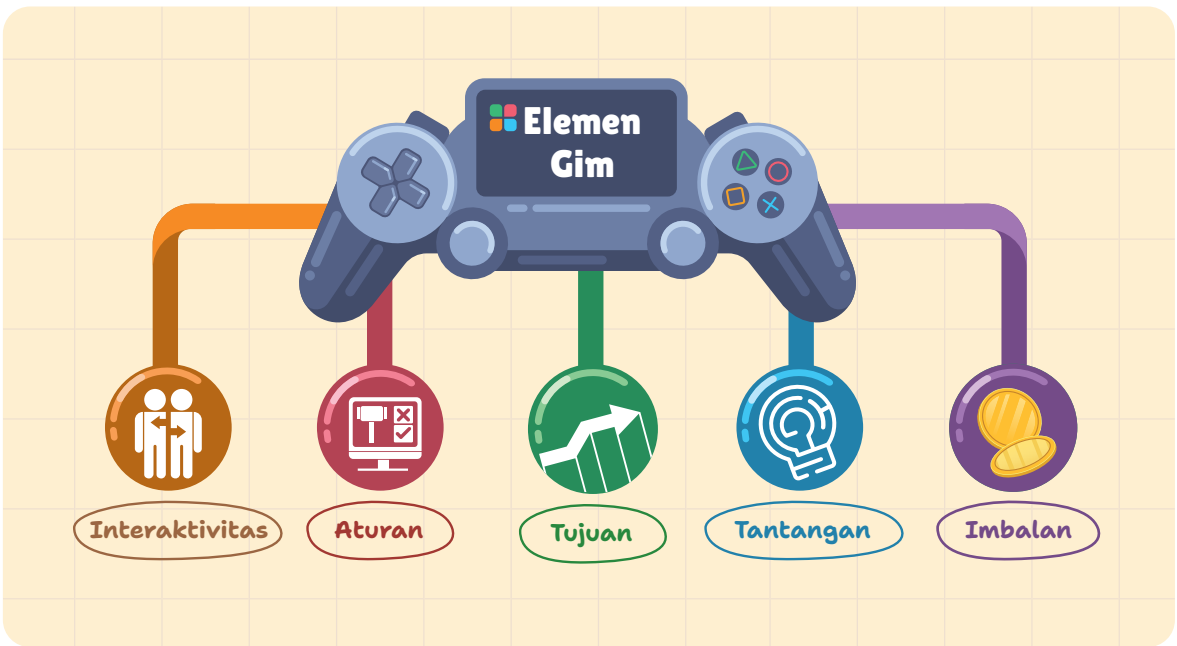
8. Opsi Kustomisasi (*Customization Options*)

kamu juga bisa mendapatkan imbalan atas kemajuan yang kamu capai dalam memainkan sebuah gim berupa pilihan untuk mengubah tampilan karakter atau elemen dalam permainan.





Gambar 1.17 Contoh Imbalan Kustomisasi di dalam Gim
Sumber: Tripwire Interactive/Deep Silver (2021)

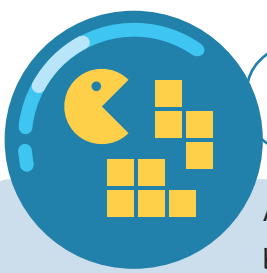


2. Gim yang Menyenangkan



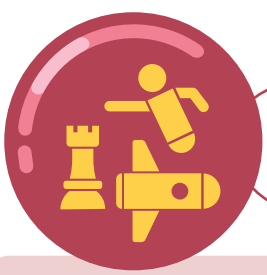
Lalu bagaimana sebuah gim itu bisa asyik dan menyenangkan untuk dimainkan? Menurut Patrick McCabe (2018) ada beberapa pertanyaan yang bisa kamu gunakan untuk memulai membuat sebuah gim yang menyenangkan untuk dimainkan, di antaranya:

Pertanyaan-Pertanyaan Sebelum Membuat Gim



Gim seperti apa yang kamu buat?

Apakah gim tersebut bertema perang, olahraga, atau permainan memasak untuk anak-anak, atau mungkin tema gim yang lainnya. Menentukan tema gim yang akan dibuat akan membantumu merancang alur cerita serta mengkontekstualisasikan berbagai keputusan penting dalam proses desain gim tersebut.



Jenis gim apa yang akan kamu buat?

Banyak jenis gim yang bisa kamu buat, di antaranya gim aksi, gim petualangan, dan gim strategi. Setiap jenis gim memiliki perbedaan, dan kamu harus bisa membuat gim sesuai dengan jenisnya.

Perasaan apa yang ingin kamu munculkan



Apakah kamu ingin pemain gim merasakan perasaan pemain yang kuat atau lemah? Sebuah gim akan mengajak pemain merasakan dan mengeksplorasi alur cerita dari gim, dan hal ini akan membuatmu menentukan alur cerita serta mekanika gim.



Siapa yang akan memainkan gim ini

Saat kamu membuat sebuah gim, maka kamu akan memikirkan bagaimana kamu akan membingkai cerita atau karakter gim seperti apa yang akan kamu buat. Apakah pemain gim akan berperan sebagai seorang pahlawan, seorang petani atau yang lainnya.

Apa yang menjadi tujuan pemain

Setiap gim pasti mempunyai tujuan yang akan dicapai. Apakah tujuannya untuk mengalahkan penjahat, menyelamatkan putri, memenangkan pertandingan olahraga dan lain-lain. Maka kamu harus membuat alur cerita dengan karakter yang sesuai dengan tujuan yang akan dicapai dalam sebuah gim.



3. Indonesia Game Rating System



Apakah kamu tahu tentang simbol yang ada dalam gambar tersebut? Ya, itu adalah IGRS atau *Indonesia Game Rating System*. Mereka yang memainkan gim atau pengembang gim, harus mengetahui tentang IGRS. IGRS ini merupakan penerapan dari Peraturan Menteri Kominfo No. 2 Tahun 2024 tentang Klasifikasi Permainan Interaktif Elektronik.

Dari gambar simbol di atas, kamu bisa mengetahui klasifikasi gim dari kategori usia. Simbol berwarna biru dengan angka 3+ ini menunjukkan permainan yang bisa dimainkan oleh pemain berusia 3 tahun atau lebih. Simbol berwarna kuning dengan angka 7+ menunjukkan permainan yang bisa dimainkan oleh pemain berusia 7 tahun ke atas. Begitu juga dengan simbol berwarna oranye dan hijau yang menunjukkan permainan yang bisa dimainkan oleh pemain berusia 13 tahun ke atas atau 15 tahun ke atas. Sedangkan simbol berwarna merah dengan tulisan 18+ menunjukkan bahwa permainan tersebut bisa dimainkan oleh pemain yang berusia lebih dari 18 tahun.

Aktivitas 1.1 **Kerja Tim**

Dalam aktivitas ini, kamu diminta untuk membentuk kelompok 2-3 orang. Tugas kamu adalah mencari informasi gim apa saja yang termasuk dalam kategori IGRS berikut ini:

Nama Gim	Usia				
	3+	7+	13+	15+	18+

4. Jenis Gim

Nah, sekarang kamu sudah mengetahui gim apa saja yang bisa dimainkan sesuai dengan usia pemainnya. Selanjutnya, coba kamu perhatikan gambar di bawah ini.



Gambar 1.18 Contoh Gim Video (*Video Game*)
Sumber: Stairway Games/Humble Games (2022)

Dalam dunia permainan, terdapat berbagai jenis gim yang menawarkan pengalaman yang beragam kepada pemain. Gambar di atas salah satu contoh gim yang berjenis gim video. Selain gim video, jenis gim lainnya adalah gim papan. Masing-masing jenis gim ini memiliki karakteristik unik yang memengaruhi cara pemain berinteraksi dan menikmati pengalaman bermain.

a Gim Video

Saat kamu memainkan sebuah gim video, maka kamu akan terhubung dengan perangkat elektronik berupa komputer, konsol ataupun ponsel. Gim video yang kamu mainkan tentu saja menggunakan teknologi yang bisa menciptakan sebuah pengalaman yang bersifat interaktif dan mendalam saat gim tersebut dimainkan. (Game Design Workshop, 2004).

Karakteristik Gim Video

Salah satu karakteristik utama gim video adalah kemampuan untuk menyajikan grafik dan audio dalam gim tersebut. Kedua hal ini tentu saja akan dapat menciptakan suasana yang lebih menarik dan menambah kedalaman cerita dalam sebuah video gim. Saat kamu memainkan gim ini, kamu tidak hanya melakukan interaksi dengan mekanika gimnya saja, tapi kamu juga akan merasakan adanya emosi yang hadir melalui desain visual yang disertai dengan suara musiknya. (The Art of Game Design, 2008)

Interaktivitas Gim Video

Fitur kunci yang membedakan antara gim video dan bentuk permainan lainnya adalah adanya interaktivitas. Saat kamu memainkan sebuah gim video, kamu mempunyai kendali atas karakter dan tindakan yang akan kamu ambil dalam gim yang kamu mainkan. Selain itu kamu juga bisa mengubah alur cerita sesuai dengan keputusan yang kamu ambil sehingga kamu merasakan ada pengalaman yang unik dan dalam memainkan gim tersebut. Kamu akan merasakan adanya pencapaian yang besar dalam gim video saat kamu berhasil menyelesaikan semua tantangan dalam permainan. (Scott Rogers, 2010)

Menurut Jeannie Novak (2022) berikut adalah contoh gim yang termasuk pada gim video:

1. Gim Aksi

Gim aksi biasanya dimainkan dengan banyaknya pergerakan cepat untuk menghancurkan musuh dan memastikan karakter kamu sebagai pemain tidak hancur melawan musuh. Gim jenis ini juga akan lebih banyak fokus pada waktu dan reaksi dari pemain.



Gambar 1.19 Contoh Gim Aksi dari Pengembang Gim Indonesia bernama Kidbash.
Sumber: Fat Raccoon (2024)

2. Gim Balap

Dalam gim balap ini, fokus kamu akan tertuju pada sekitar kendaraan yang sedang melakukan balapan dengan kendaraan lainnya. Pemain akan membuat pergerakan pada kendaraan secepat mungkin dengan tidak kehilangan kendali.



Gambar 1.20 Contoh Gim Balap dari Pengembang Gim Indonesia bernama Baro Kart
Sumber: Satriver Studio (2023)

3. Gim Pertarungan

Dalam gim pertarungan ini, kamu dan pemain lainnya akan banyak melakukan gerakan seperti memukul, menahan serangan atau aksi lainnya. Aksi tersebut bertujuan untuk menyerang lawan atau bertahan dari serangan yang diberikan oleh lawan.



Gambar 1.21 Contoh Gim Pertarungan Dari Pengembang Gim Indonesia bernama Battle of Guardian.
Sumber: Miracle Gates Entertainment (2023)

4. Gim Teka-Teki

Sebagai pemain gim teka teki, kamu akan fokus untuk menyelesaikan permainan berdasarkan masalah yang disajikan dalam gim tersebut. Contoh dari gim teka teki ini adalah gim tetris yang selalu memberikan tantangan pada pemainnya untuk bisa menyusun dan membentuk balok panjang menjadi garis lurus agar blok garis tersebut bisa hilang.



Gambar 1.22 Contoh Gim Teka Teki Bernama TETRIS.
Sumber: Digital Eclipse (2024)

5. Gim Strategi

Tujuan utama dari gim strategi adalah untuk menentukan sebuah strategi yang akan digunakan dalam menyelesaikan masalah yang disajikan ole sistem permainannya. Terdapat dua jenis gim strategi yang bisa dimainkan yaitu Gim Strategi *Real-Time (RTS)* dan Gim Strategi *Turn-Based (TBS). RTS* memiliki sebuah tantangan waktu, karena waktu akan terus berjalan setiap detik dan semua pemain dapat berjalan secara bersamaan. Berbanding terbalik dengan *TBS,* dimana setiap pemain memiliki giliran dalam melakukan aksi, sehingga waktu bukan menjadi tantangan dan pemain yang lain dapat berpikir sampai gilirannya tiba.

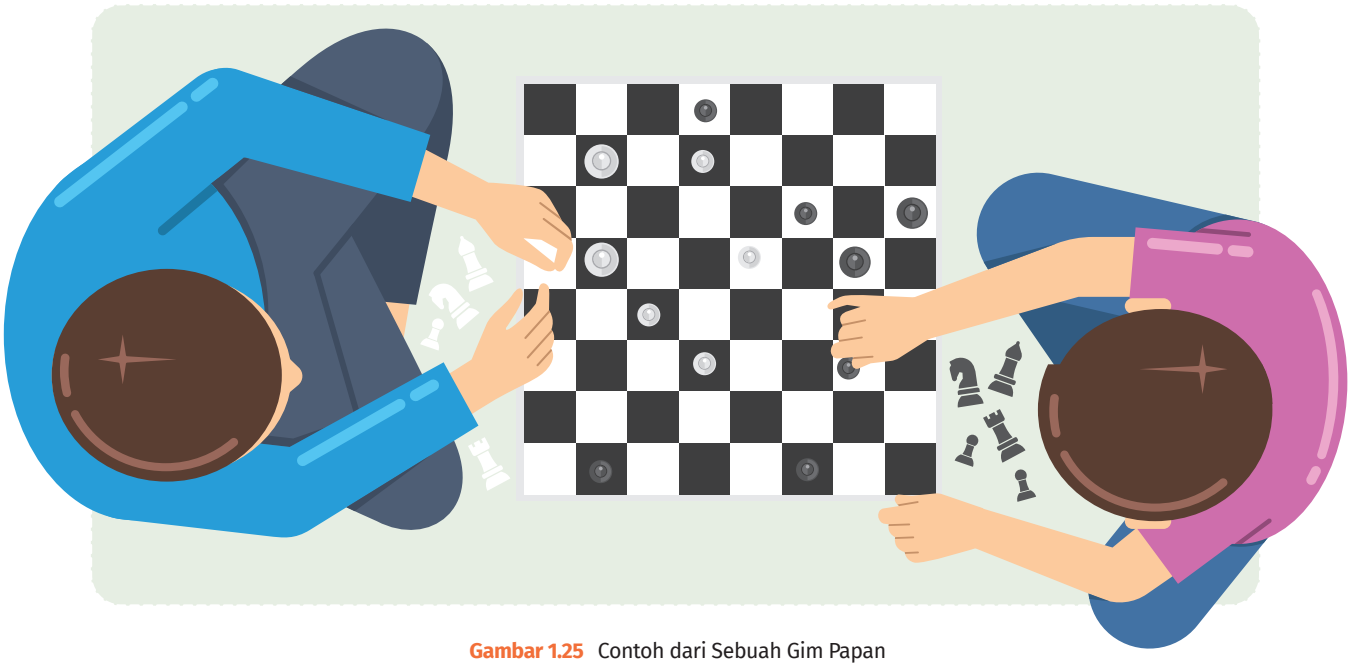


Gambar 1.23 Contoh Gim Strategi *Real-Time* yang Memperlihatkan Dua Tipe Tentara yang Sedang Menyerang dan Bertahan
Sumber: Forgotten Empires/Xbox Game Studios (2019)



Gambar 1.24 Contoh Gim Strategi *Turn-Based* dari Pengembang Gim Indonesia Bernama Kriegsfront Tactics
Sumber: Toge Productions (2024)

b Gim Papan



Gambar 1.25 Contoh dari Sebuah Gim Papan

Seperti namanya, kamu tentu bisa mengetahui bahwa gim ini bisa kamu mainkan di atas papan dengan beberapa komponen berupa pion. Dalam gim papan ini kamu akan mendapatkan sebuah pengalaman sosial yang berbeda, karena kamu memainkan gim ini secara bersama-sama dengan berkumpul bersama pemain lainnya dan melakukan interaksi secara langsung (Brenda Brathwaite dan Ian Schreiber, 2008). Gim papan yang sangat populer dimainkan dan mungkin kamu juga pernah memainkannya adalah gim catur. Dalam gim ini kamu akan merasakan adanya strategi dan interaksi dengan pemain lainnya.

Aspek Sosial Gim Papan

Saat kamu memainkan gim papan ini, kamu akan merasakan adanya aspek sosial yang sangat signifikan dalam permainannya. Gim ini bisa menciptakan suasana kerja sama dan kompetisi antar pemain. Karena adanya interaksi secara langsung, maka dalam gim ini akan ada diskusi dan juga negosiasi. Hal ini lah yang akan membuat kamu merasakan adanya kenangan dan pengalaman sosial yang sangat kuat dibanding saat kamu bermain gim video (Lee Sheldon, 2011).

Mekanika Permainan Gim Papan

Mekanika permainan dalam gim papan sering kali lebih sederhana dan mudah dipahami. Dalam *Game Feel* oleh Steve Swink (2009), penulis menguraikan bahwa gim papan biasanya memiliki aturan yang jelas dan terbatas, sehingga pemain dapat dengan cepat memahami cara bermain. Hal ini membuat gim papan lebih mudah diakses bagi pemain baru dan menciptakan suasana yang lebih santai.

Keterlibatan Emosional dalam Gim Papan

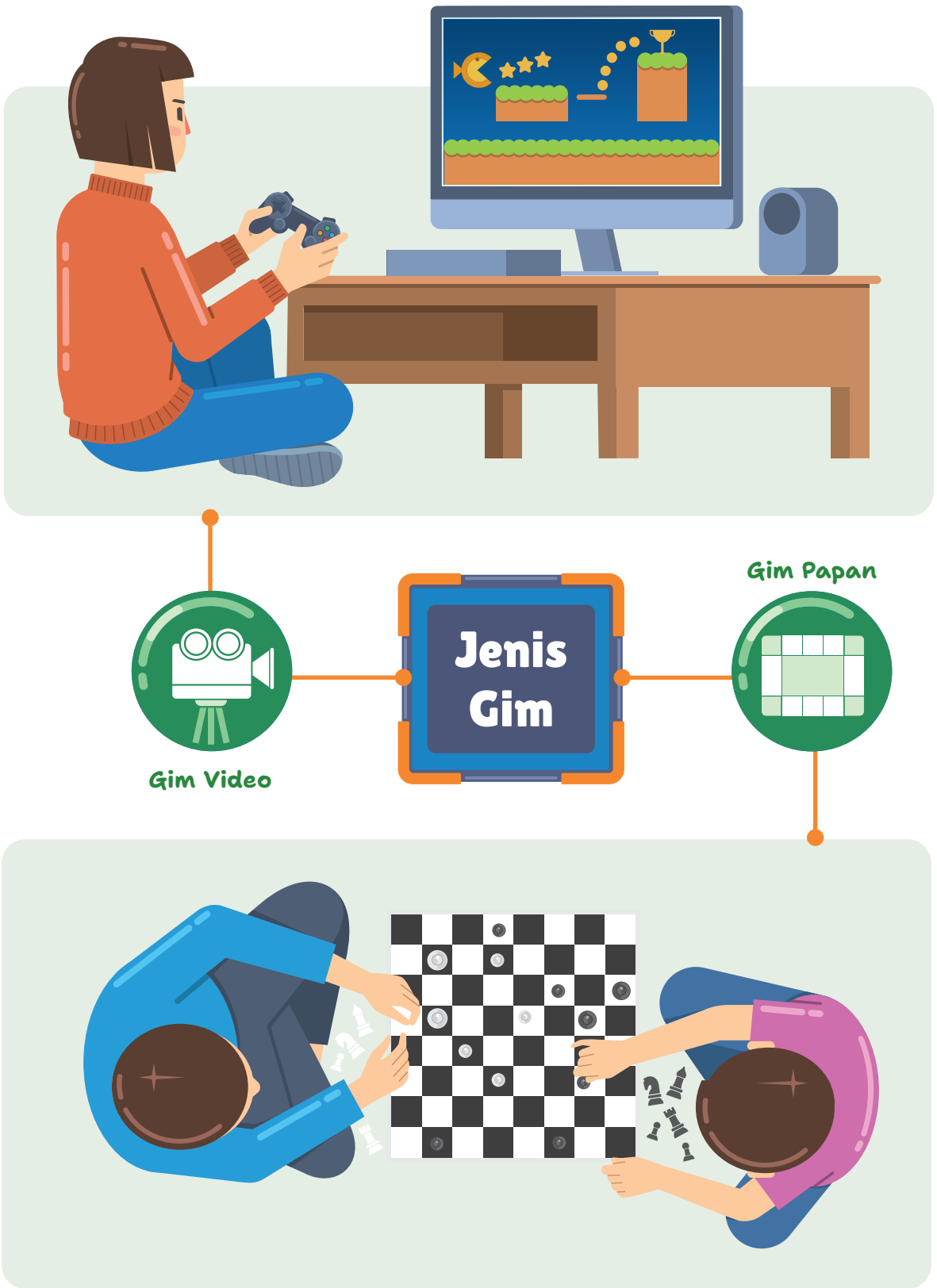
Gim papan juga dapat memberikan keterlibatan emosional yang mendalam. Dalam *A Theory of Fun for Game Design* oleh Raph Koster (2004), disebutkan bahwa gim papan sering kali membangkitkan rasa nostalgia dan kenangan bagi pemain. Pengalaman bermain bersama teman atau keluarga dapat menciptakan momen berharga yang diingat selama bertahun-tahun.

Proses pengembangan gim video dan gim papan juga sangat berbeda. saat kamu akan melakukan pengembangan gim video, maka kamu akan melibatkan berbagai disiplin ilmu, termasuk pemrograman, desain grafis, dan pengembangan cerita. sedangkan dalam gim papan, kamu akan lebih fokus pada desain mekanika dan elemen fisik, yang sering kali memerlukan *prototyping* dan pengujian di dunia nyata.

Aktivitas 1.2 Literasi Mandiri

Dalam aktivitas ini, kamu diminta untuk membentuk kelompok 2-3 orang. Tugas kamu adalah mencari informasi gim apa saja yang termasuk dalam kategori IGRS berikut ini:

Jenis Gim	Nama/ Judul Gim	Tujuan Gim	Tantangan Gim
Gim Video			
Gim Papan			


Jenis Gim
Gim Video
Gim Papan

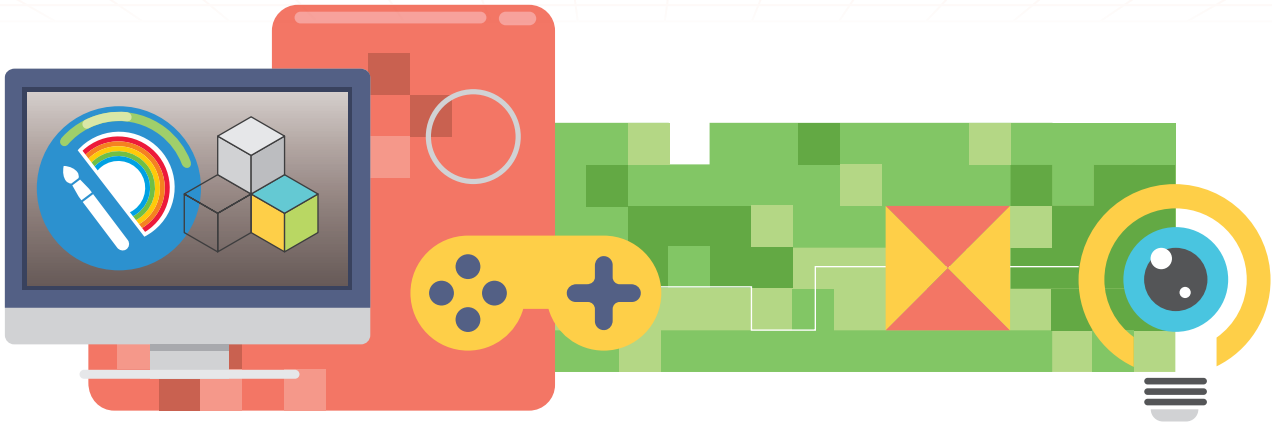
Aktivitas 1.3 Literasi Mandiri

Nah, sekarang coba kamu cari tahu tentang gim apa saja yang paling banyak dimainkan oleh teman-teman kamu di sekolah dan apa alasannya. Kemudian, kamu lengkapi tabel berikut ini:

Nama Gim	Jenis Gim	Alasan Menyukai Gim

Setelah itu, silahkan kamu presentasikan hasil yang kamu dapatkan di depan kelas untuk bisa saling berbagi informasi dengan teman yang lainnya.

B. Desain Gim



Desain gim yang baik memberikan kesempatan bagi pemain untuk bisa merasakan kepuasan bermain gim melalui eksplorasi. Sebagai pemain, kamu akan dihadapkan pada situasi yang mengharuskan berpikir kritis dan mencari solusi dan menciptakan pengalaman yang tidak hanya menghibur, tetapi juga mendidik. Dengan demikian, permainan dalam gim tersebut akan lebih menarik, menimbulkan rasa ingin tahu, serta meningkatkan kreativitas pemain (Jesse Schell, 2008). Desain yang mempertimbangkan aturan dengan baik tidak hanya mengarahkan pemain, tetapi juga memungkinkan mereka untuk merencanakan strategi dan membuat keputusan yang lebih bijaksana, menjadikan setiap langkah mereka memiliki makna (Tracy Fullerton, 2004)

Desain gim merupakan suatu proses untuk menciptakan konten dan peraturan di dalam gim. Ketika merancang desain gim yang baik, kamu perlu menciptakan tujuan yang mampu memotivasi pemain untuk terus bermain hingga mencapai tujuan tersebut, sambil tetap mengikuti aturan yang telah ditetapkan dalam gim. Perancang gim akan membuat pemain mengikuti permainan sesuai arah yang diinginkan oleh perancangnya, seperti mengikuti aturan untuk mengelilingi papan atau terus maju melangkah ke level berikutnya. Jika pemain tidak mempunyai alasan untuk mengikuti aturan ini maka permainan ini akan tidak menyenangkan. Dalam membuat sebuah gim, kamu harus bisa berpikir atau melihat dari di sudut pandang seorang pemain, dan menjawab beberapa pertanyaan berikut ini:

1. Gim ini tentang apa?
2. Bagaimana cara saya memainkannya?
3. Bagaimana saya akan memenangkannya?
4. Mengapa saya harus memainkannya?
5. Hal-hal apa saja yang saya butuhkan untuk melakukannya?

Dengan menjawab pertanyaan-pertanyaan di atas, kamu akan bisa membuat sebuah desain gim yang menarik dan menyenangkan.

1. Jenis Desain Gim

Seperti halnya jenis gim, dalam desain gim juga ada beberapa jenis desain gim. Berikut beberapa Jenis Desain Gim menurut Brenda Brathwaite dan Ian Schreiber (2009).

a World Design

Penciptaan latar belakang, pengaturan, dan tema dari gim yang akan dibuat secara keseluruhan akan dilakukan dalam *world design* ini. Umumnya World Design akan menjadi jangkar dalam membuat *System Design, Content Design, dan Level Design.*

b System Design

Menciptakan aturan dan pola matematika yang menjadi pola dasar dalam sebuah permainan akan dilakukan dalam *system design*. *System design* merupakan satu-satunya tugas dalam desain gim yang umumnya dilakukan dalam semua gim karena semua gim memiliki aturan.

c Content Design

Menciptakan karakter, item, teka-teki dan misi dilakukan dalam *content design*. Walaupun ini lebih banyak dilakukan dalam video gim, namun aturan permainan dan pengumpulan atau koleksi kartu juga lebih banyak ditampilkan dalam konten sebuah gim.

d Level Design

Pembuatan level gim yang termasuk di dalamnya membuat tata letak sebuah peta, penempatan objek, serta tantangan yang ada dalam peta tersebut dilakukan dalam *level design* ini.

Aktivitas 1.4 **Kerja Tim**

Buatlah tim yang beranggotakan tiga orang, kemudian carilah sebuah contoh desain gim yang ada di internet. Setelah itu kalian diskusikan dengan teman kalian contoh desain gim tersebut. Setelah berdiskusi, buatlah sebuah ringkasan dalam bentuk infografis berupa bagan dan gambar sederhana, disertai penjelasan singkat yang bisa kamu kerjakan dengan ditulis tangan.

2. Mekanika Gim

Mekanika gim merupakan elemen dasar dan aturan yang harus kamu ketahui dalam menentukan bagaimana sebuah permainan akan berfungsi dan bisa melakukan interaksi dengan pemainnya. Di dalamnya terdapat aturan, batasan, dan bagaimana pemain itu bisa mempengaruhi permainannya. Kamu bisa menganggap mekanika gim ini sebagai komponen yang mengatur interaksi antara pemain dan dunia permainan, menciptakan pengalaman bermain yang mendalam dan interaktif.

Mampu memahami dengan jelas tentang mekanika gim merupakan kunci bagi kamu untuk merancang sebuah gim yang menarik dan menyenangkan. Jika mekanika yang dirancang tidak baik maka pemain akan merasakan permainan yang monoton dan membosankan **(Tracy Fullerton, 2004).** Mekanika gim memiliki beberapa kategori, di antaranya:

a Movement Mechanics

Movement mechanics mencakup aturan dan cara pemain dapat bergerak dalam dunia permainan. Contohnya, berjalan, melompat, berlari, atau terbang. contoh mekanika seperti ini bisa kamu lihat dalam gim *Super Mario Bros*.

b Combat Mechanics

Combat mechanics adalah sistem yang mengatur pertempuran, termasuk cara menyerang, bertahan, dan menggunakan senjata atau kemampuan. Contoh mekanika ini dapat kamu lihat dalam gim *Dark Souls*, di mana setiap serangan dan pertahanan pemain memiliki konsekuensi yang nyata, menambah kedalaman pada setiap pertempuran.

c Content Design

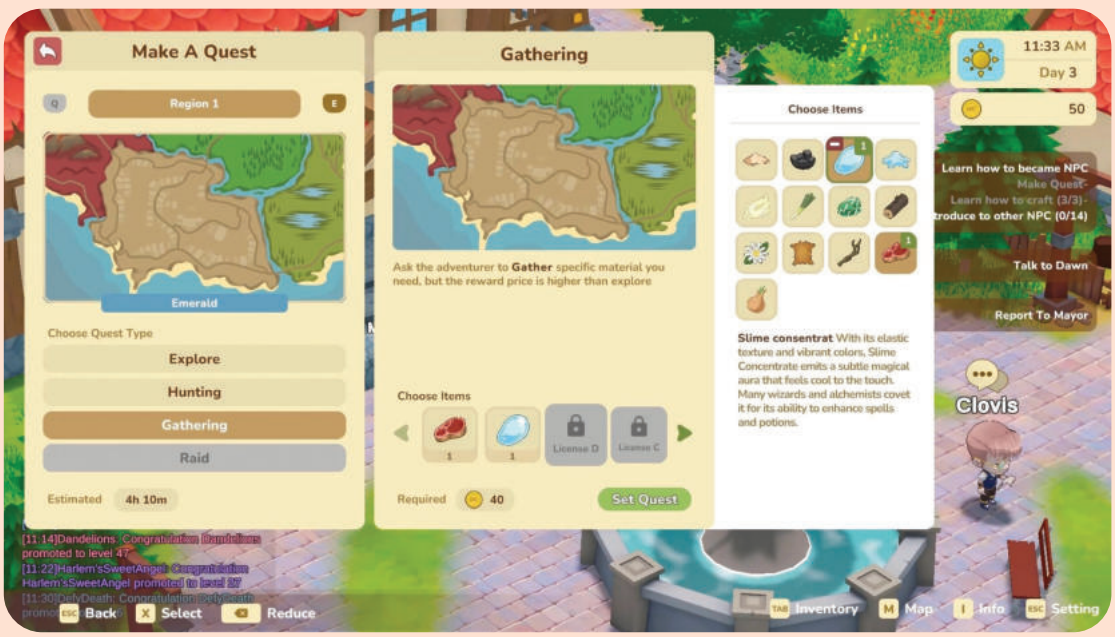
Menciptakan karakter, item, teka-teki dan misi dilakukan dalam *content design*. Walaupun ini lebih banyak dilakukan dalam video gim, namun aturan permainan dan pengumpulan atau koleksi kartu juga lebih banyak ditampilkan dalam konten sebuah gim.



Gambar 1.26 Contoh *Combat Mechanics* dalam Sebuah Gim
Sumber: FromSoftware/Bandai Namco Entertainment (2020)

d Resource Management

Resource Management adalah mekanika yang mengatur pengumpulan dan penggunaan sumber daya, seperti uang, bahan, atau energi. Contoh mekanika ini dapat kamu lihat dalam gim *NPCVille: A Story of Blacksmith*. Dalam gim tersebut pemain harus mengelola sumber daya dan waktu untuk mencapai kemajuan, menciptakan pengalaman *gameplay* yang mendalam dan berulang.



Gambar 1.27 Contoh *Resource Management Mechanics* dalam Sebuah Gim
Sumber: GINVO Studio (2023)

e Progression Mechanics

Progression Mechanics berkaitan dengan kemajuan pemain, seperti *leveling up*, *mengupgrade* karakter, atau membuka konten baru. Contoh mekanika ini dapat kamu lihat dalam gim *The Elder Scrolls V: Skyrim*, di mana pemain dapat mengembangkan keterampilan dan membuka konten baru, menjaga rasa pencapaian dan keterlibatan.



Gambar 1.28 Contoh *Progression Mechanics* dalam Sebuah Gim
Sumber: Bethesda Game Studios/Bethesda Softworks (2016)

f Puzzle Mechanics

Puzzle Mechanics mengatur cara pemain menyelesaikan teka-teki, sering kali melibatkan logika atau strategi. Dalam *Challenges for Game Designers* oleh Brenda Brathwaite dan Ian Schreiber (2009), dijelaskan bahwa teka-teki yang dirancang dengan baik dapat memberikan kepuasan besar saat dipecahkan. Contohnya, dalam gim berjudul *Portal*, mekanika *puzzle* yang inovatif menantang pemain untuk berpikir kreatif dan menemukan solusi untuk bergerak maju dalam permainan.

g Sosial Mechanics

Social Mechanics mengatur interaksi antar pemain, seperti *trading*, berkolaborasi, atau berkompetisi. Dalam *The Multiplayer Classroom* oleh Lee Sheldon (2011), menekankan bahwa interaksi sosial yang baik dapat memperkaya pengalaman bermain. Misalnya, dalam gim *World of Warcraft*, pemain dapat berkolaborasi dalam *raid* atau melakukan *trading*, menciptakan komunitas yang aktif dan terlibat.


a Movement Mechanics
b Combat Mechanics
c Resource Management
d Progression Mechanics
e Puzzle Mechanics
f Social Mechanics
Mekanika Gim

Setelah kamu mengetahui kategori dari mekanika gim, berikutnya kamu perlu mengetahui beberapa hal penting dalam mekanika gim

a. Integrasi Mekanika Gim dalam *Game Loop*

Mekanika-mekanika ini saling terintegrasi dalam *Game Loop*, menciptakan siklus interaksi yang berkelanjutan. Setiap kali pemain melakukan input, mekanika yang relevan akan diproses dan memberikan umpan balik yang sesuai, yang pada gilirannya mempengaruhi pengalaman pemain. Dengan demikian, *Game Loop* berfungsi sebagai kerangka kerja yang mendukung interaksi mekanika ini (Jeannie Novak, 2022).

b. Pengujian Mekanika Gim

Pengujian mekanika gim merupakan langkah penting yang kamu lakukan dalam proses pengembangan gim. Penting melakukan pengujian untuk memastikan mekanika berfungsi sebagaimana mestinya dan menciptakan pengalaman yang menyenangkan. Melalui pengujian, kamu dapat menemukan dan memperbaiki masalah mekanika yang dapat merugikan *gameplay*.

c. Menciptakan Dinamika yang Menarik

Saat kamu dapat menggunakan berbagai mekanika gim, maka kamu dapat menciptakan dinamika yang menarik dalam sebuah permainan. menggabungkan berbagai mekanika ini perlu dilakukan untuk menciptakan pengalaman yang unik dan beragam. Kombinasi yang tepat dapat meningkatkan *replayability* dan memberikan pengalaman bermain yang lebih kaya.

d. Menggunakan Data untuk Meningkatkan Mekanika Gim

Data analitik dapat digunakan untuk meningkatkan desain mekanika gim. penting bagi kamu untuk melakukan analisis perilaku pemain untuk memahami bagaimana mereka berinteraksi dengan mekanika. Informasi ini dapat kamu gunakan untuk menyesuaikan mekanika agar lebih sesuai dengan preferensi pemain.

3. Naratif Gim

Konsep naratif gim akan mengacu pada struktur, tema, dan elemen naratif yang nantinya akan membentuk alur cerita dalam sebuah gim. Beberapa elemen naratif yang dimaksud adalah karakter, latar, konflik, dan juga resolusi serta cara bagaimana mengintegrasikannya dengan *gameplay*. Hal Ini akan menjadi dasar bagi pemain untuk bisa merasakan pengalaman yang lebih mendalam yang di penuhi oleh cerita dan emosi dan tidak hanya sekedar bermain gim (Katie Salen dan Eric Zimmerman, 2004).

a Elemen Utama

1. Karakter

Elemen ini merupakan elemen yang penting untuk bisa kamu hadirkan dalam sebuah naratif gim dimana pemain akan bisa merasakan adanya keterikatan emosional terhadap karakter-karakter dalam gim yang akan memperkuat pengalaman bermain. ketika pemain bisa mengembangkan karakternya dengan baik maka hal ini akan menciptakan peluang untuk berkembanganya alur cerita menjadi lebih mendalam dan menarik (Jesse Schell, 2004).

2. Latar

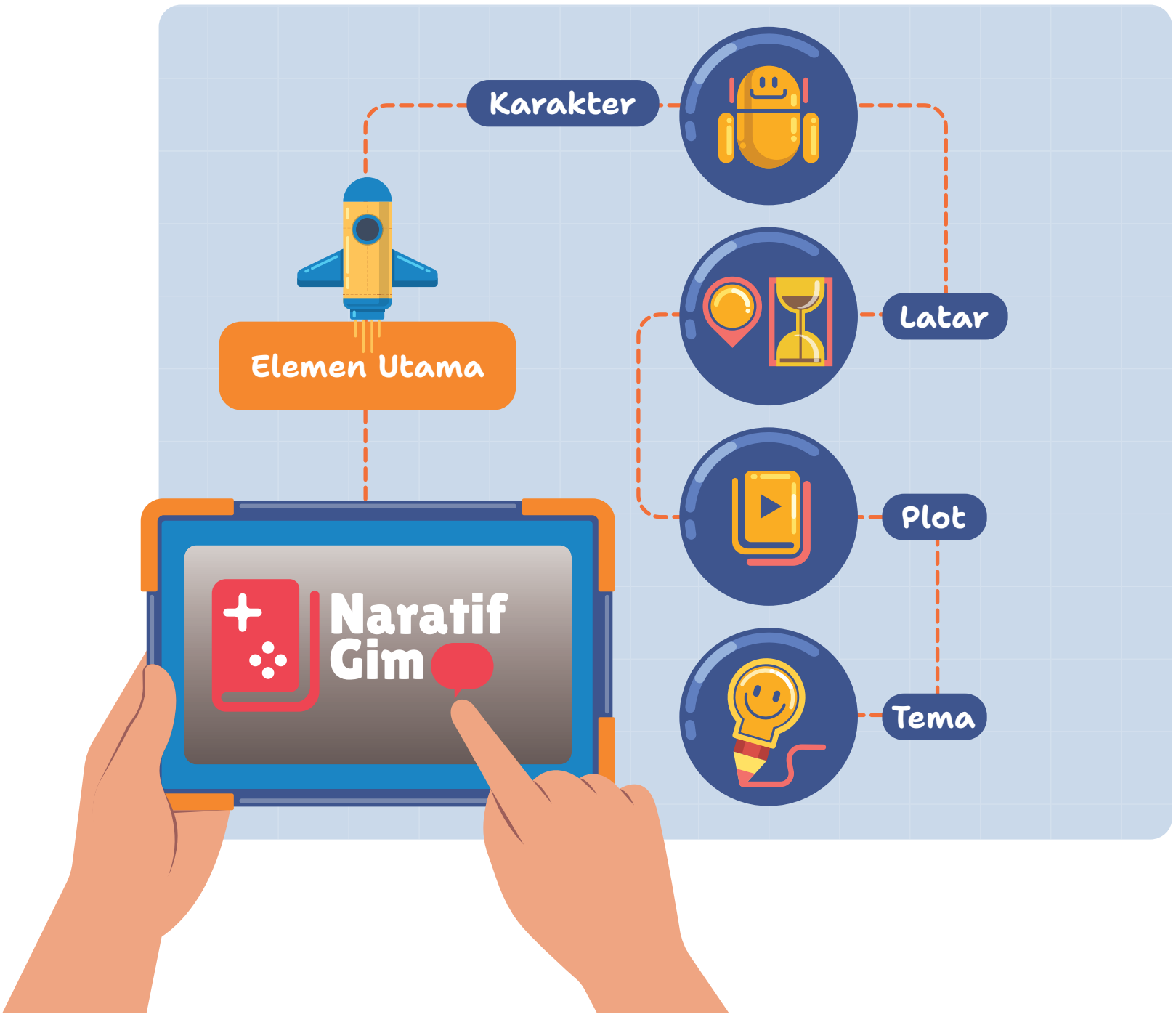
Mengatur tempat dan waktu pada cerita yang sedang berlangsung bisa kamu lakukan dalam elemen latar ini. Sebuah gim yang dirancang dengan latar yang baik akan mampu menciptakan suasana yang terasa hidup dan memungkinkan pemain untuk terlibat lebih dalam dengan cerita yang diberikan (Steve Swink, 2008).

3. Plot

Saat kamu membuat elemen ini maka kamu akan menentukan alur cerita yang di dalamnya terdiri dari rangkaian peristiwa, adanya pengenalan, konflik, klimaks dan juga resolusi dari cerita. Elemen plot yang kamu buat harus bisa menarik perhatian pemain dan juga memberikan kejutan sehingga bisa meningkatkan daya tarik naratif gim. (Jeannie Novak, 2022)

4. Tema

Pesan atau ide yang ingin disampaikan melalui cerita. tema ini bisa berupa moral, sosial atau juga emosional yang mendasari pengalaman pemain (Simon Egenfeldt-Nielsen, 2013).



b Interaktivitas

1. Pilihan Pemain

Banyak gim memberikan pemain kebebasan untuk membuat keputusan yang mempengaruhi alur cerita. Pilihan yang diberikan kepada pemain meningkatkan keterlibatan dan memungkinkan mereka merasa lebih berinvestasi dalam hasil cerita. Hal Ini menciptakan rasa kepemilikan atas pengalaman yang dihadapi (Tobias Heussner, 2018).

2. Pengalaman Naratif Dinamis

Beberapa gim menggunakan sistem yang memungkinkan perubahan cerita berdasarkan tindakan pemain, menciptakan pengalaman unik bagi setiap individu. Sistem naratif dinamis ini membuat setiap *playthrough* terasa baru dan menarik, memberikan *replayability* yang tinggi (Tracy Fullerton, 2004).

c Teknik Penyampaian

1. *Cutscenes*

Potongan video yang menyajikan bagian penting dari cerita dengan cara sinematik. *cutscenes* sering digunakan untuk memperlihatkan momen-momen krusial dalam alur cerita. Mereka dapat menciptakan ketegangan dan membantu menjelaskan konteks yang mungkin sulit diungkapkan melalui *gameplay* (Tracy Fullerton, 2004).

2. Dialog Interaktif

Merupakan percakapan antara karakter yang dapat dipilih oleh pemain, sering kali mempengaruhi arah cerita dan hubungan antar karakter. Dialog yang dinamis meningkatkan kedalaman narasi dan memberikan pemain lebih banyak kebebasan untuk membentuk cerita (Lee Sheldon, 2008).

3. *Environmental Storytelling*

Environmental storytelling menggunakan lingkungan dan objek di dalam gim untuk menceritakan cerita tanpa kata-kata. Teknik ini memberikan konteks dan latar belakang kepada pemain dengan cara yang menarik dan tidak langsung.

d Contoh Gim Bernarasi Kuat

1. The Last of Us

Contoh yang sangat baik dari gim dengan narasi yang kuat. Gim ini menggambarkan hubungan emosional antara karakter utama, disertai dengan tema pengorbanan dan kemanusiaan. Kekuatan naratif gim ini terletak pada pengembangan karakter dan situasi yang realistis.



Gambar 1.29 Contoh Cuplikan Narasi dari Gim *Last of Us*
Sumber: Naughty Dog LLC/ PlayStation Publishing LLC (2023)

2. Bioshock

Bioshock menawarkan pengalaman naratif yang mendalam dengan elemen filosofi dan moral yang kompleks. Gim ini menciptakan *twist* yang mengejutkan dan membuat pemain berpikir tentang konsekuensi dari tindakan mereka.



Gambar 1.30 Contoh Cuplikan Narasi dari Gim *Bioshock*
Sumber: Irrational Games/2K (2013)

e Tantangan Naratif gim

1. Keseimbangan antara Gameplay dan Cerita

Pengembang harus memastikan bahwa narasi tidak mengganggu pengalaman bermain, melainkan meningkatkan kedalaman *gameplay*. Dalam *Game Design Workshop* oleh Tracy Fullerton (2004), ditekankan bahwa keseimbangan ini adalah kunci untuk menciptakan pengalaman yang memuaskan bagi pemain.

2. Keterbatasan Teknologi

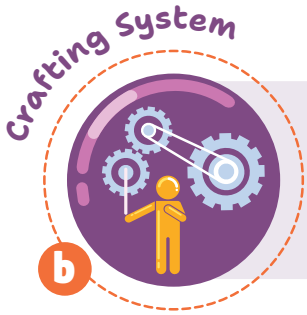
Gim dengan narasi kompleks mungkin menghadapi kendala teknis, seperti waktu pemuatan atau kapasitas penyimpanan, yang dapat mempengaruhi penyampaian cerita. Dalam *Challenges for Game Designers* oleh Brenda Brathwaite dan Ian Schreiber (2008), penulis mencatat bahwa tantangan teknis ini sering kali mengharuskan desainer untuk menemukan cara-cara inovatif untuk menyampaikan cerita.

4. Sistem Gim

Kumpulan mekanika gim terkadang disebut juga sebagai *System Mechanics*. *System Mechanics* mengacu pada kumpulan mekanika gim yang bekerja bersama untuk menciptakan interaksi dan pengalaman yang lebih kompleks dalam sebuah permainan. Setiap *system mechanic* terdiri dari beberapa mekanika yang saling terhubung dan mempengaruhi satu sama lain. Dengan memahami bagaimana berbagai mekanik ini berinteraksi, kamu dapat menciptakan *gameplay* yang lebih mendalam dan menarik (Tracy Fullerton, 2004).



Salah satu dari *system mechanics* adalah sistem pertempuran (*Combat System*). Dalam konteks ini, berbagai mekanik seperti *movement*, *attack*, dan *defense* bekerja sama untuk menciptakan pengalaman pertempuran yang dinamis dan strategis. Penting bagi kamu untuk melakukan desain sistem yang seimbang agar setiap elemen dapat berkontribusi pada *gameplay* yang baik.



Sistem *crafting* adalah contoh lain dari *system mechanics* yang kompleks. Dalam sistem ini, berbagai mekanik saling terkait untuk menciptakan pengalaman yang lebih menyeluruh bagi pemain.



Progression System adalah mekanik lain yang berfungsi untuk memberikan rasa kemajuan kepada pemain. Kemajuan yang terasa memuaskan dapat meningkatkan motivasi pemain untuk terus bermain. Salah satu contoh *progression system* dalam gim *Jetpack Joyride 2*



Gambar 1.31 Contoh *Progression System* di dalam Gim Jetpack Joyride 2
Sumber: Muhammad Ardhan Fadhlurrahman/Halfbrick Studio (2024)

Di dalam gim ini, terdapat *progression system* yang berbentuk *upgrade health* dan *attack*. Di mana pemain dapat mengumpulkan koin yang didapat dari permainan dan melakukan pembelian atau *upgrade* untuk menambah *health* pemain ataupun menambahkan *attack* agar serangan lebih kuat.



Gambar 1.32 Contoh Lanjutan *Progression System* di dalam Gim Jetpack Joyride 2
Sumber: Muhammad Ardhan Fadhlurrahman/Halfbrick Studio (2024)


Combat System
Crafting System
Progression System
Sistem Gim

5. Teknikal Gim

Dalam melakukan implementasinya, sebuah desain gim yang kamu buat harus juga dilengkapi dengan konsep teknikal gim. Kerangka kerja atau *blueprint* inilah yang nantinya akan menggambarkan bagaimana aspek teknis dari gim yang akan kamu implementasikan. Dalam konsep ini juga ada keputusan tentang teknologi, perangkat keras dan perangkat lunak serta metode yang akan digunakan dalam proses pengembangannya. memahami teknikal gim merupakan salah satu aspek teknis yang akan menentukan keberhasilan dari sebuah proyek gim. Apakah kamu tahu kenapa konsep teknikal gim ini sangat penting dalam pengembangan gim? berikut ini penjelasannya:

Panduan Teknis Pengembangan

Konsep teknikal gim ini akan dapat memberikan arah bagi kamu saat mengembangkan sebuah gim. Kamu bisa memastikan semua anggota dalam tim memiliki pemahaman yang sama tentang bagaimana gim ini akan dibangun. Kerangka kerja yang kamu buat dengan jelas, akan membantu mencegah adanya kebingungan antara anggota dalam tim, juga bisa mengurangi resiko terjadinya kesalahan dalam implementasinya.

Pengelolaan Sumber Daya Pengembangan

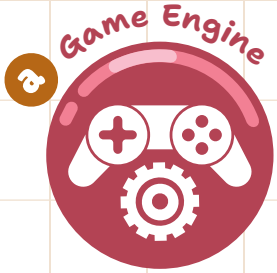
Dengan memetakan aspek teknis, kamu dapat lebih efektif dalam mengalokasikan sumber daya dan waktu yang dibutuhkan dalam proses pengembangan gim ini dilakukan. Hal ini penting kamu lakukan untuk meningkatkan efisiensi tim dan juga mampu mengoptimalkan proses pengembangan gimnya.

Identifikasi Tantangan dan Batasan

Konsep teknikal gim ini juga memungkinkan kamu untuk mengidentifikasi tantangan yang ada dan batasan teknisnya sejak awal sehingga dapat direncanakan solusi yang tepat. Perlu ditekankan bahwa pentingnya pengembangan dengan mempertimbangkan tantangan teknis untuk mencegah masalah di masa depan.

Lalu apakah kamu tau komponen apa saja yang ada dalam konsep teknikal gim ini? Berikut penjelasan beberapa komponen tersebut:





Kamu bisa memilih *game engine* yang akan digunakan diantaranya Unity, Unreal Engine atau Godot untuk mengembangkan sebuah gim. Berapa hal yang harus kamu perhatikan dalam memilih *game engine,* seperti memastikan *game engine* ini bisa mengelola kompleksitas grafis dan *gameplay*-nya. Kemudian fitur yang ada dalam *game engine* harus bisa memenuhi kebutuhan pengembangan gimnya. Kemudahan yang diberikan oleh *game engine* terkait dengan pengembangan gim kedepannya.



Kamu perlu menentukan platform yang akan digunakan untuk merilis gim yang kamu buat, apakah itu PC, konsol, atau perangkat seluler. Dalam memilih platform, ada beberapa hal yang perlu diperhatikan, seperti memastikan spesifikasi perangkat keras yang mampu menjalankan gim dengan baik, merancang antarmuka yang sesuai dengan platform untuk memberikan interaksi optimal kepada pengguna, serta memahami kebutuhan dan pengalaman pengguna untuk meningkatkan daya tarik gim.



Kamu juga akan merancang sistem input yang akan digunakan pemain termasuk *kontroler* keyboard, *mouse*, atau layar sentuh. Sistem input yang akan kamu buat harus bisa mempertimbangkan dan memperhatikan beberapa hal, seperti responsivitas input yang mengukur seberapa cepat input pemain diterima dan diproses, kemudian intuitivitas kontrol yang mendesain kontrol yang mudah dipahami oleh pemain.



Kamu membuat Konsep untuk elemen visual permainan, termasuk gaya seni, pemodelan 3D atau 2D, dan efek visual. Untuk menentukan dan membuat grafis dan visual ini kamu harus memperhatikan beberapa hal, seperti kualitas grafis yang dapat memastikan grafis berkualitas tinggi yang menarik pemain karena hal ini akan sangat berpengaruh pada pengalaman bermain. Kemudian kamu juga harus memperhatikan optimasi dari grafis agar sesuai dengan spesifikasi perangkat kerasnya. Lalu kamu juga harus memperhatikan, kesesuaian tema gim yang akan kamu buat karena hal ini akan sangat berpengaruh pada visual yang akan dibuat dalam gim.



Dalam elemen audio ini kamu merancang beberapa komponen termasuk didalamnya berupa musik, efek suara dan dialog. Hal yang harus kamu perhatikan dalam audio ini, seperti kualitas audio yang kamu rancang haruslah berkualitas tinggi untuk meningkatkan pengalaman pemain. Kemudian integrasi dengan *gameplay* yang harus mendukung dalam memberikan umpan balik yang jelas pada pemain. Lalu penggunaan teknologi audio yang tepat menggunakan teknologi audio terkini.

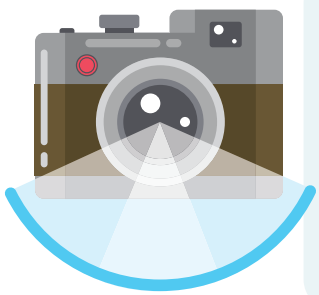


Jika gim yang kamu buat memiliki elemen *multiplayer* maka kamu perlu merancang sistem jaringan untuk mendukung interaksi antar pemain. Beberapa hal yang perlu kamu perhatikan diantaranya terkait dengan sinkronisasi, manajemen server dan juga latensi agar interaksi akan terasa lebih real-time.

6. Level Gim



Setelah kamu menentukan mekanika sebuah gim, maka berikutnya yang akan kamu lakukan adalah membuat tantangan sesuai dengan level dari gim dengan memahami konsep level gim. Konsep level gim ini merupakan pendekatan dalam merancang level permainan. Konsep level gim mencakup tata letak, mekanik, tujuan, dan pengalaman pengguna. Hal ini penting kamu lakukan untuk membantu dalam memahami bagaimana setiap level berkontribusi terhadap keseluruhan pengalaman bermain. Kamu bisa membuat level gim ini dengan memperhatikan komponen berikut ini (Tracy Fullerton, 2004):



a Kamera

Kamu bisa mengibaratkan kamera ini sebagai jendela bagi pemain untuk melihat dan berinteraksi dengan dunia permainannya. sudut pandang dari kamera ini akan sangat mempengaruhi pengalaman pemain apakah perspektif orang pertama, orang ketiga, atau isometrik, dan pemilihan jenis kamera juga harus bisa disesuaikan dengan tujuan *gameplay*.



b Kontrol

Komponen ini merupakan aspek yang sangat krusial dalam sebuah gim karena hal ini akan menentukan bagaimana pemain mengendalikan karakter. membuat kontrol yang baik akan memberikan dampak yang sangat signifikan pada *gameplay*. Kamu perlu melakukan pengujian level dengan berbagai perangkat untuk memastikan bahwa kontrol terasa intuitif dan responsif.



c Karakter

Karakter merupakan elemen kunci yang harus kamu perhatikan, karena hal ini berkaitan dengan bagaimana kamu akan menciptakan keterikatan emosional dengan pemain. Ada beberapa hal yang harus kamu perhatikan dalam pengembangan karakter ini di antaranya, kamu harus memahami kemampuan karakter dan mengajarkannya pada pemain secara bertahap. Kemudian menilai kekuatan karakter di setiap fase permainan dan kamu harus bisa memastikan tantangan yang ada juga sudah sesuai. Selain itu kamu juga harus bisa menyediakan opsi karakter yang bisa di kustomisasi sendiri oleh pemain sesuai dengan gaya bermainnya (Jesse Schell, 2008).

Dengan memanfaatkan ketiga komponen di atas dengan baik maka desain level yang kamu buat akan lebih efektif dan optimal. Level yang kamu buat harus bisa memperkenalkan mekanik yang baru secara bertahap yang dapat mengembangkan pengalaman bermain para pemainnya. Ada beberapa jenis konsep level dalam pengembangan gim, di antaranya:

a *Linear Levels*

Dalam *linear* level ini pemain akan mengikuti alur yang telah ditentukan di mana pemain akan bergerak dari satu titik ke titik berikutnya dengan tantangan yang terstruktur. Pengalaman bermain yang diberikan dalam gim ini akan lebih jelas dan fokus. Contoh *linear* level ini ada pada gim mario bros.





Gambar 1.33 Sebuah representasi *Linear Level* di dalam Gim
Sumber: Nintendo (2024)

b Mission–Based Levels

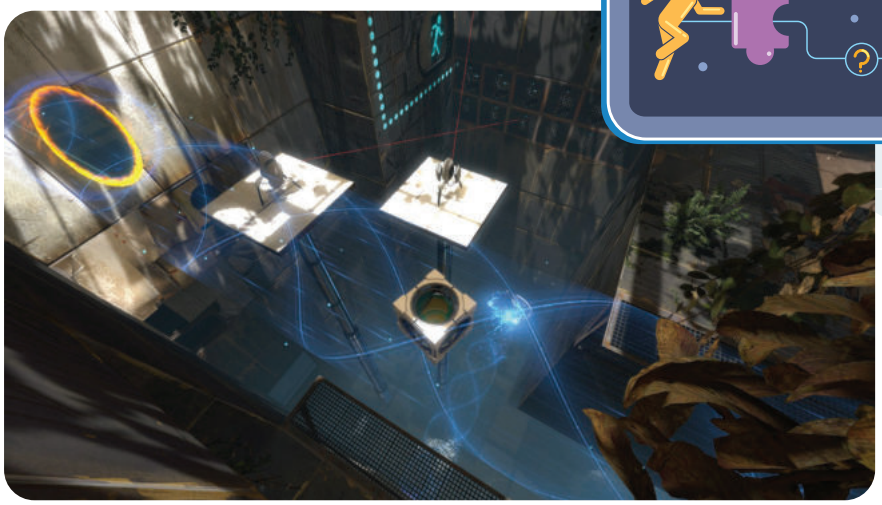
Dalam level jenis ini pemain akan menyelesaikan misi tertentu di setiap levelnya. Dalam setiap misi akan ada tantangan yang berbeda-beda. Contoh gim *Hitman* dan *Assassin's Creed*,



Gambar 1.34 Sebuah representasi *Mission-Based Level* di dalam Gim
Sumber: IO Interactive/Square Enix (2016)

c Puzzle Levels

Level jenis ini lebih difokuskan pada penyelesaian teka-teki untuk maju, contohnya *Portal* dan *The Witness.*





Gambar 1.35 Sebuah representasi *Puzzle Level* di dalam Gim
Sumber: Valve (2011)

d Stealth Levels

Level ini Dirancang untuk *gameplay* yang mengutamakan penyembunyian. Contohnya *Metal Gear Solid* dan *Dishonored*, di mana keberhasilan bergantung pada kemampuan untuk tetap tersembunyi, dalam gim ini akan ada pengalaman yang menegangkan.



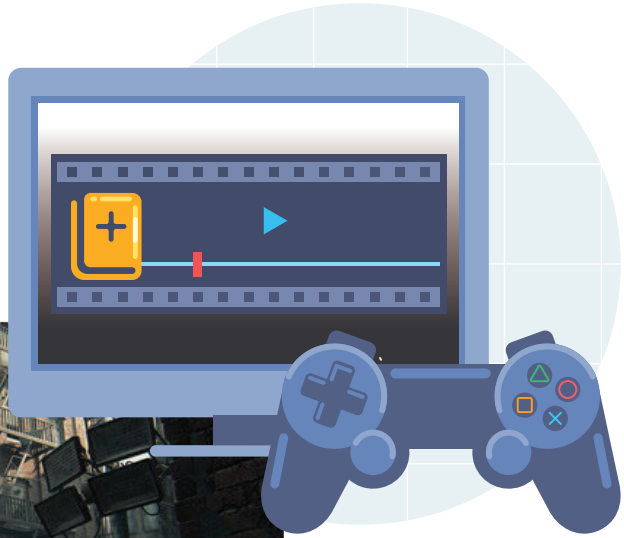


Gambar 1.36 Sebuah representasi *Stealth Level* di dalam Gim
Sumber: KONAMI (2015)

e Narrative Levels

Level yang berfokus pada penyampaian cerita. Contohnya *Life is Strange* dan *The Last of Us*, di mana *gameplay* dan narasi saling terkait.





Gambar 1.37 Sebuah representasi *Narrative Level* di dalam Gim
Sumber: Naughty Dog LLC/ PlayStation Publishing LLC (2023)



7. Riset Pengguna Gim

Riset pengguna gim ini biasanya dilakukan untuk mencari tahu seperti apa perilaku, interaksi dan pengalaman seorang pemain dalam memainkan sebuah gim. Beberapa cara yang digunakan untuk melakukan riset ini di antaranya dengan melakukan observasi, wawancara atau dengan melakukan survei untuk mengumpulkan data.

Industri gim melakukan riset agar mereka bisa mengetahui gim yang mereka ciptakan bisa menarik perhatian pemain dan tetap memainkan gim mereka daripada memainkan gim pesaing mereka. Berikut ini adalah beberapa gambaran mengapa riset pengguna gim ini penting untuk dilakukan:

Aktivitas 1.5 Literasi Mandiri

Coba kamu perhatikan dampak-dampak berikut jika kamu menggunakan riset pengguna dan tidak menggunakan riset pengguna ini:

a. gim ini akan sangat rumit untuk dimainkan

b. hanya kamu saja yang mengetahui dan mengidentifikasi kesalahan yang ada dalam gim tersebut dan memperbaikinya.

c. Pengguna gim ini akan melakukan identifikasi dan menyoroti banyak masalah yang bisa mereka temukan dalam gim yang kamu buat, dan hasil identifikasi ini bisa kamu gunakan untuk memperbaiki permasalahan yang muncul

d. tidak laku dipasaran dan tidak ada yang minat untuk memainkannya

e. setelah gim ini dimainkan oleh penggunanya, maka akan ditemukan banyaknya *bug* dalam gim yang kamu buat.

f. Lalu pengguna gim tersebut juga merasakan kesulitan dan tidak mengerti dalam memainkan gim ini.

g. kamu akan memperoleh banyak data dari pengguna gim terkait dengan gim yang mereka mainkan.

h. kamu akan merasa yakin bahwa gim yang kamu buat akan berhasil di terima di pasaran dan banyak peminatnya.

Kemudian kamu lengkapi tabel berikut ini dengan memasukkan dampak yang terjadi jika menggunakan atau tidak menggunakan riset pengguna!

Menggunakan Riset Pengguna	Tidak Menggunakan Riset Pengguna

Metodologi yang dapat digunakan untuk melakukan riset gim terdiri dari dua jenis, yaitu penelitian kualitatif dan penelitian kuantitatif.

a Penelitian Kualitatif

Penelitian kualitatif berfokus pada pemahaman mendalam terhadap pengguna dengan melibatkan kelompok kecil. Pendekatan ini bertujuan untuk menggali bagaimana pemain merasakan, bertindak, dan berpikir saat memainkan gim, sehingga dapat membantu merancang desain yang lebih baik. Berikut beberapa metode yang dapat digunakan:

1. Wawancara Langsung

Selama proses pengembangan gim, wawancara langsung dengan pengguna memungkinkan kamu memahami perilaku dan preferensi mereka. Pada tahap awal, wawancara ini dapat digunakan untuk menggali ide, mengetahui pengalaman mereka, serta memahami apa yang mereka rasakan saat bermain gim.

2. Observasi Pengguna

Melalui observasi langsung, kamu dapat memperhatikan bagaimana pengguna memainkan gim. Catat pilihan mereka, tingkat keberhasilan, serta emosi yang mereka tunjukkan selama bermain. Metode ini memberikan wawasan tentang pengalaman pengguna secara nyata dan mendalam.

Penelitian kualitatif ini sangat membantu dalam pengujian dan pengembangan gim yang lebih efektif dan sesuai dengan kebutuhan pengguna.

b Penelitian Kuantitatif

Selain penelitian kualitatif, penelitian kuantitatif juga dapat dilakukan untuk memahami perilaku pengguna dalam jumlah besar atau secara massal. Penelitian ini menghasilkan data statistik yang berguna untuk menganalisis pola permainan dan perilaku pemain. Berikut beberapa metode yang dapat digunakan:

1. Analitik Data Pemain

Dengan menggunakan analitik, kamu dapat memantau tindakan pemain selama bermain gim. Data yang dikumpulkan bisa mencakup waktu yang dihabiskan pemain, jumlah pengulangan level, serta banyaknya komponen yang berhasil dikumpulkan. Informasi ini memberikan gambaran objektif tentang cara pemain berinteraksi dengan gim.

2. Survei dan Kuesioner

Untuk melengkapi data statistik, kamu dapat menyebarkan survei atau kuesioner kepada pemain. Pertanyaan-pertanyaan yang disusun secara terstruktur dapat menggali perasaan, pengalaman, dan masukan mereka selama bermain gim.

Penelitian kuantitatif ini memberikan pandangan berbasis data yang dapat mendukung pengembangan gim dengan lebih terukur dan efektif.

Dalam riset pengguna gim ini sering kali berasal dari berbagai latar belakang ilmu pengetahuan, seperti ahli ilmu komputer, psikolog, sosiologi, studi media dan desain, dan sebagainya. Para ahli akan terlibat dalam menganalisis permainan dari berbagai sudut pandang keilmuan. Metode yang digunakan bisa berbentuk interviu, data analisis, survey.





Pengujian pengguna adalah proses mengamati pemain saat mereka bermain gim untuk mengidentifikasi masalah dalam desain, kontrol, dan mekanik. Pengujian ini dapat dilakukan pada berbagai tahap pengembangan, mulai dari prototipe hingga versi akhir gim. Pengujian yang efektif memungkinkan tim pengembang menemukan masalah yang mungkin tidak terdeteksi selama proses pengembangan internal.

Dalam penelitian *User Testing in Game Development* oleh Alena Denisova dkk. (2024), *playtesting* atau pengujian pengguna dimulai dengan mendefinisikan pertanyaan riset dan tujuan spesifik dari sesi *playtest*. Setelah itu, metode yang sesuai dirancang untuk mencapai tujuan tersebut. Metode ini dapat berupa:

1. **Pengumpulan Data Perilaku (Behavioral Data)**

   Contohnya, "Apakah pemain dapat menyelesaikan tugas tertentu dalam gim?" Metode ini fokus pada bagaimana pemain berinteraksi dengan gim dan apakah mereka dapat menyelesaikan elemen tertentu seperti yang diharapkan.

2. **Pengumpulan Data Sikap (Attitudinal Data)**

   Contohnya, "Apakah pemain menyukai fitur tertentu dalam gim?" Metode ini bertujuan untuk memahami persepsi dan kepuasan pemain terhadap berbagai aspek gim.

Data yang diperoleh dari sesi *playtesting* kemudian dianalisis lebih lanjut dengan *Gameplay Data Analysis* untuk menghasilkan wawasan yang berguna dalam memperbaiki desain dan mekanik gim. Proses ini memastikan gim yang dikembangkan lebih menarik dan ramah pengguna.

b Analisis Data Gameplay



Analisis data *gameplay* melibatkan penggunaan data analitik untuk memantau interaksi pemain dengan gim. kamu dapat mengumpulkan beberapa informasi seperti waktu yang dihabiskan di level tertentu, tingkat penyelesaian, dan pola perilaku lainnya. Aktivitas ini disebut juga *data mining* dimana menurut Günter Wallner (2019), *data mining* pada pengembangan gim memiliki empat kegunaan:

1. Melakukan improvisasi dari pemodelan gim dan pengalaman pemain
2. Sebagai pertimbangan dalam keputusan bisnis
3. Sebagai alat bantu untuk melakukan inovasi dan optimalisasi dari sebuah teknologi gim
4. Pemahaman lebih mendalam tentang pemain

Menarik data permainan dan melakukan *update* dari permainan sesuai dengan analisis data permainan tersebut agar lebih diterima oleh target market dari gim yang dihasilkan (Günter Wallner, 2019).

c Focus Groups Discussion



Focus groups discussion adalah diskusi kelompok yang diadakan dengan pemain untuk mendapatkan umpan balik mendalam tentang pengalaman dalam gim. Sangat penting untuk mendapatkan perspektif yang lebih luas melalui diskusi ini, karena bisa mengungkapkan ide-ide untuk perbaikan yang mungkin tidak terdeteksi dalam metode lain (Tracy Fullerton, 2004).

Aktivitas 1.6 Literasi Mandiri

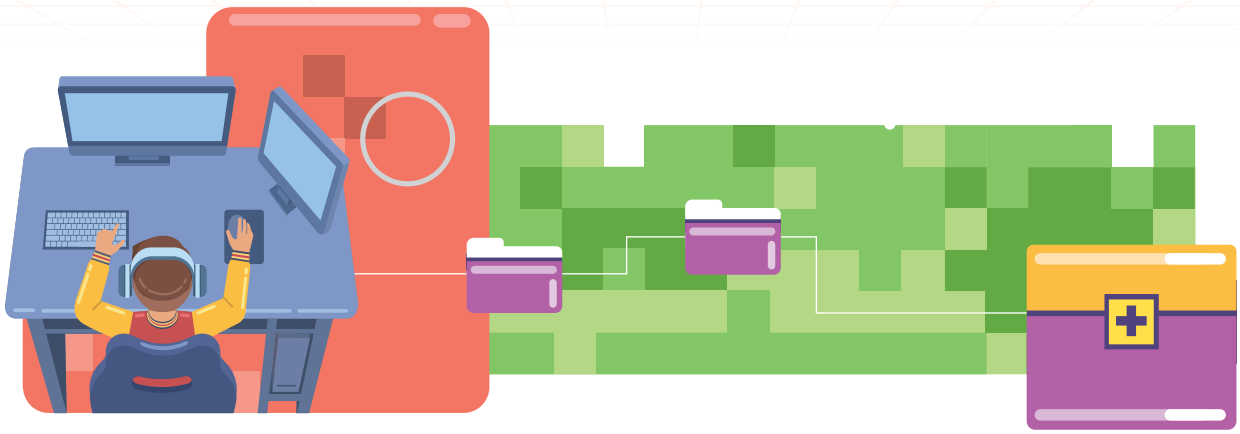
Untuk menambah pemahaman kamu mengenai desain gim, lakukanlah aktivitas berikut ini secara mandiri:

1. Cari dan pilihlah salah satu gim yang sering kamu mainkan.
2. Kemudian kamu identifikasi dan tentukan mekanika gim, teknikal gim, naratif gim dan level gim serta sistem gim dari gim yang kamu pilih tadi.
3. Lengkapi lah tabel berikut ini:

Desain Gim	Deskripsi
Nama Gim	
Jenis/Genre Gim	
Mekanika Gim	
Teknikal Gim	
Naratif Gim	
Level Gim	
Sistem Gim	

4. Setelah itu kamu presentasikan hasil identifikasi kamu di depan kelas untuk saling berbagi informasi dengan teman lainnya.

C. Game Design Document (GDD)



Aktivitas 1.7 Literasi Mandiri

Coba kamu pilih salah satu gim yang paling kamu sukai untuk dimainkan. Identifikasilah beberapa hal yang menarik dari gim tersebut, lalu diskusikan Bersama teman kamu dan tuliskan hasil identifikasi kamu dalam buku catatan ya.

Sekarang coba kamu membayangkan jika kamu diberi tugas untuk membuat gim seperti gim yang kamu sukai tersebut, apa yang akan kamu lakukan agar gim yang kamu buat itu bisa menarik dan menyenangkan untuk dimainkan? Apa saja yang harus kamu perhatikan dan siapkan agar gim kamu dimainkan oleh banyak orang?

Gim yang menarik dan menyenangkan bukan hanya karena ide dan kreativitas dan pembuatnya saja, tapi gim itu dibuat dan dikembangkan dengan mengikuti desain dokumen gimnya atau *Game Design Document* (GDD). GDD ini merupakan kerangka dari pembuatan suatu gim secara utuh. GDD adalah dokumen terperinci yang berfungsi sebagai *blueprint* atau panduan untuk pengembangan sebuah permainan bagi seluruh tim pengembang.

1. Komponen Utama GDD

Ada beberapa komponen utama dalam *Game Design Document* yang harus kamu pahami di antaranya:

a. Konsep Permainan

Dalam komponen ini kamu akan membuat sebuah deskripsi singkat tentang gim yang akan kamu buat mencakup tema, genre, dan tujuan utama dari gim tersebut. Selain itu kamu juga bisa menetapkan visi dari gim yang dibuat. Konsep yang jelas akan bisa memberikan pemahaman yang sama pada tim yang lain dan fokus selama proses pengembangan dilakukan (Tracy Fullerton, 2004).

b. *Gameplay* dan Mekanik

Gameplay dan mekanik adalah aspek kunci yang harus bisa kamu jelaskan secara mendetail dalam GDD. Hal Ini mencakup mekanik inti, cara pemain berinteraksi dengan gim, serta sistem yang digunakan, seperti pertempuran, crafting, atau eksplorasi. Mekanik ini merupakan jantung dari *gameplay* yang menarik (Ernest Adams dan Andrew Rollings, 2004).

c. Cerita dan Karakter

Membuat detail dari alur cerita, karakter utama, dan perkembangan naratif ini sangat penting kamu buat dalam GDD. Cerita yang kuat akan dapat meningkatkan keterlibatan pemain dalam gim tersebut. Deskripsi karakter yang jelas juga akan memberikan arahan yang jelas pada pemain untuk berinteraksi dengan gim dan dengan pemain lainnya (Jesse Schell, 2008).

d. Desain Level

Dalam desain level ini kamu akan memberikan gambaran tentang struktur level, termasuk tata letak, tantangan, dan tujuan di setiap level. desain level yang baik dapat menciptakan pengalaman bermain yang seimbang dan menarik. Ini adalah area di mana kreativitas dan teknik bertemu untuk menciptakan tantangan yang memuaskan bagi pemain (Scott Rogers, 2008).

e. Estetika dan Gaya Seni

Kamu bisa menggambarkan visual gim dalam komponen estetika dan gaya seni ini. Visual gim mencakup palet warna dan elemen artistik lainnya. Aspek visual ini menjadi sangat penting karena akan mampu menarik perhatian pemain dan menciptakan suasana gim yang tepat. GDD yang kamu buat harus bisa mendefinisikan gaya seni yang ingin dicapai agar semua anggota tim memiliki panduan yang sama.

f. Audio

Dalam komponen ini kamu bisa menggambarkan rencana untuk elemen audio, seperti musik, efek suara, dan dialog, ke dalam GDD. Audio yang baik dapat memperkuat pengalaman pemain dan memberikan konteks emosional pada *gameplay*. Rencana audio yang jelas akan memastikan elemen suara selaras dengan estetika visual.

g. Teknik dan Teknologi

GDD harus mencakup informasi tentang teknologi yang akan digunakan, termasuk *game engine*, perangkat keras, dan perangkat lunak yang diperlukan. Teknologi yang tepat akan dapat membantu kamu memahami batasan dan kemampuan yang dimiliki. Hal ini juga akan memudahkan komunikasi antara programer dan desainer.

h. Target Audiens

Penjelasan tentang siapa pemain yang menjadi sasaran, termasuk demografi dan preferensi, merupakan bagian penting yang harus kamu buat dalam GDD. pemahaman yang baik tentang target audien ini akan dapat mempengaruhi keputusan desain dan pemasaran yang dibuat. Mengetahui siapa yang akan memainkan gim juga akan memungkinkan kamu untuk menyesuaikan mekanik dan pengalaman bermain sesuai kebutuhan gim.

i. Rencana Pemasaran dan Distribusi

Setelah semua komponen selesai kamu buat, berikutnya kamu harus menyiapkan strategi pemasaran dan distribusi untuk gim yang sudah selesai kamu buat. Rencana yang jelas dalam pemasaran ini perlu kamu siapkan agar kamu bisa menjangkau audiens target dan memperkenalkan gimnya. rencana ini meliputi pemilihan platform distribusi, strategi promosi, dan pengembangan komunitas pemain.

Menurut kamu, kenapa GDD ini sangat penting dalam pembuatan gim? Mengapa kamu harus membuat GDD sebelum kamu memulai membuat atau mengembangkan sebuah gim? Jawaban dari pertanyaan ini tampak pada gambar berikut:



Panduan Yang Jelas

Salah satu fungsi utama dari GDD adalah memberikan panduan yang jelas untuk semua anggota tim. Tanpa GDD, komunikasi antar tim dapat menjadi kabur, yang dapat menyebabkan kebingungan dan miskomunikasi.



Penyelarasan Visi

GDD membantumu menyelaraskan visi antara desainer, programer, dan anggota tim lainnya. Memiliki dokumen yang dapat dirujuk bersama akan memudahkan kolaborasi dan pengambilan keputusan. Ini juga akan memungkinkan tim untuk tetap fokus pada tujuan akhir.



Referensi untuk Iterasi

Selama proses pengembangan, GDD berfungsi sebagai referensi untuk iterasi. kamu dapat merujuk kembali ke GDD untuk memastikan bahwa perubahan yang dibuat masih sesuai dengan visi awal. Ini juga akan membantu menjaga konsistensi dalam desain.



Pengelolaan Risiko

GDD memungkinkan kamu untuk mengidentifikasi dan mengelola risiko lebih awal. dengan mendokumentasikan semua elemen desain, kamu dapat dengan mudah menemukan masalah potensial sebelum berkembang menjadi hal yang lebih besar.



Dokumentasi untuk Pengujian

GDD yang kamu buat juga akan berfungsi sebagai dokumentasi untuk pengujian. Pengujian yang efektif akan membutuhkan pemahaman yang jelas tentang mekanik dan tujuan desainnya. GDD yang kamu buat ini juga harus bisa menyediakan informasi yang diperlukan untuk merancang pengujian yang relevan.



Pemasaran yang Efektif

GDD yang kamu buat juga harus bisa membantu dalam merencanakan strategi pemasaran yang efektif. pemahaman audien dan elemen kunci dari permainan dapat membantu dalam merancang kampanye pemasaran yang lebih baik. Hal ini mengarah pada potensi yang lebih besar untuk sukses di pasar.

Contoh GDD Pada Permainan "Mr Alien Butuh Uang untuk Jajan!"

Belum ada bentuk atau format baku untuk membuat sebuah rancangan GDD. Berikut ini adalah salah satu contoh bentuk atau format pembuatan sebuah GDD untuk gim "Mr Alien Butuh Uang untuk Jajan!"

1. Konsep Permainan

"Mr Alien Butuh Uang untuk Jajan!" adalah permainan platformer 2D dimana pemain mengendalikan seorang alien bernama Mr. Alien. Mr Alien membutuhkan uang untuk jajan tetapi tidak diberikan oleh ayahnya, sehingga Mr.Alien harus mengumpulkan koin di berbagai level sambil menghindari rintangan dan tantangan yang sudah dibuat oleh ayahnya. Pemain hanya dapat melakukan aksi lompat dan bergerak, sehingga strategi dalam melompat menjadi kunci untuk menyelesaikan setiap level.

2. Gameplay dan Mekanik

"Mr Alien Butuh Uang untuk Jajan!" adalah permainan platformer 2D dimana pemain mengendalikan seorang alien bernama Mr. Alien. Mr Alien membutuhkan uang untuk jajan tetapi tidak diberikan oleh ayahnya, sehingga Mr.Alien harus mengumpulkan koin di berbagai level sambil menghindari rintangan dan tantangan yang sudah dibuat oleh ayahnya. Pemain hanya dapat melakukan aksi lompat dan bergerak, sehingga strategi dalam melompat menjadi kunci untuk menyelesaikan setiap level.

a. Komponen Utama:

- **Kontrol Sederhana:**

  Pemain hanya menggunakan tombol untuk melompat, membuatnya mudah diakses.

- **Level Beragam:**

  Berbagai level dengan tema dan tantangan yang berbeda, seperti hutan, gua, dan kota alien.

- **Koin:**

  "Koin yang dapat dikumpulkan untuk meningkatkan skor; ada koin spesial yang memberikan bonus yang lebih banyak"

- **Rintangan:**

  "Berbagai rintangan seperti duri, platform bergerak, dan musuh yang harus dihindari."

- **Power-ups:**

  "Item yang memberikan kemampuan khusus untuk jangka waktu tertentu, seperti lompatan tinggi atau kecepatan ekstra."

b. Gameplay:

- **Tujuan Pemain**

  Mengumpulkan sebanyak mungkin koin sambil mencapai akhir level tanpa terkena rintangan.

- **Mekanika**

  1. Melompat:

     Pemain menggunakan satu tombol untuk melompat. Tinggi lompatan dapat bervariasi tergantung waktu penekanan tombol

  2. Penghindaran:

  Pemain harus menghindari rintangan yang muncul, baik secara statis maupun dinamis.

- **Skor**

  Setiap koin yang dikumpulkan menambah skor. Terdapat leaderboard untuk menampilkan pemain dengan skor tertinggi.

3. Cerita Karakter

Mr Alien: sebuah alien yang sedang sangat ingin untuk jajan dan maka dari itu dirinya ingin mendapatkan banyak uang agar dia bisa jajan sesuka hatinya.

4. Desain Level

a. **Hutan Alien**

- **Rintangan:** Platform Kayu, Duri
- **Jenis Koin:** 20 koin biasa, 2 koin spesial.

b. **Gua Gelap**

- **Rintangan:** Batu Jatuh, stalaktit.
- **Jenis Koin:** 25 koin biasa, 3 koin spesial.

c. **Kota Alien**

- **Rintangan:** Robot penjaga, platform bergerak.
- **Jenis Koin:** 30 koin biasa, 5 koin spesial.

5. Estetika dan Gaya Seni

Gaya kartun yang cerah dan penuh warna, desain karakter alien yang imut dan menarik.

6. Audio

Musik latar yang ceria dan energik untuk menambah suasana permainan.

7. Teknik dan Teknologi

Gim akan dibuat dengan teknologi Unity dan bahasa C#.

8. Target Audiens

Game Casual yang bisa dimainkan oleh semua umur yang suka dengan tema ringan dan ceria.

8. Rencana Pengembangan, Bisnis dan Distribusi

a. Rencana Pengembangan

- **Pra-produksi:** 2 bulan
- **Produksi:** 6 bulan
- **Uji Coba dan Peluncuran:** 2 bulan.

b. Komposisi Tim Pengembang

- **Game Desainer:** 1 orang
- **Programmer:** 1 orang
- **Grafik Desainer:** 1 orang
- **Musik dan Suara:** 1 orang.

c. Rencana Bisnis

- **Freemium dengan Ads:** gim Gratis dengan iklan yang bisa dihilangkan dengan pembelian dalam aplikasi
- **Item Pembelian:** Skins untuk karakter, power-ups permanen.

d. Rencana Distribusi

- **Uji Beta:** Melakukan uji coba dengan sekelompok pemain untuk mendapatkan umpan balik tentang kontrol dan kesulitan level.
- **Perbaikan Berbasis Umpan Balik:** Mengadaptasi tingkat kesulitan dan menambahkan fitur berdasarkan umpan balik pemain.
- **Rilis:** Melakukan rilis setelah melakukan perbaikan terakhir dan memastikan semua konten telah dimasukan.
- ***Update Post Rilis:*** Melakukan update level tambahan setelah rilis untuk memperpanjang waktu permainan.



Uji Kompetensi

Coba kamu jawab pertanyaan di bawah ini untuk mengetahui apakah kamu sudah menguasai materi tentang Pemodelan Gim.

1. Berikan tanda ceklis (✓) pada kolom benar atau salah dari pernyataan-pernyataan berikut ini :

No	Pernyataan	Benar	Salah
1	Aturan tidak hanya memberikan batasan tetapi juga menciptakan peluang bagi pemain untuk berstrategi.		
2	Gim adalah bentuk hiburan interaktif yang melibatkan satu atau lebih pemain dalam aktivitas dengan aturan tertentu dan tujuan yang jelas.		
3	Dengan menetapkan tujuan yang jelas, gim memberikan arah dan fokus bagi pemain, mendorong pemain untuk berinteraksi dengan dunia permainan. Tujuan ini tidak bisa bervariasi		

2. Jika kamu membuat sebuah gim, bagaimana hierarki tantangan dalam gim tersebut? Berapa banyak level yang kamu harapkan? Tantangan apa saja yang umum di setiap level?

3. Pasangkan pernyataan berikut dengan menarik garis dari titik ke arah elemen yang tepat!

Elemen ini memainkan peran penting dalam menjaga keterlibatan, motivasi, dan peluang bagi pemain untuk mengembangkan keterampilan dan strategi mereka.	Imbalan
Penghargaan yang diberikan kepada pemain setelah mereka berhasil mencapai tujuan atau mengatasi tantangan. Hal ini menciptakan rasa pencapaian dan memotivasi pemain untuk terus berinteraksi dengan permainan	Tujuan
Misi atau target yang harus dicapai oleh pemain dalam permainan. Dengan menetapkan tujuan yang jelas, gim memberikan arah dan fokus bagi pemain, mendorong pemain untuk berinteraksi dengan dunia permainan.	Tantangan

4. Empat elemen utama dalam gim bertujuan untuk menciptakan pengalaman bermain yang menarik dan menyenangkan. Elemen-elemen ini membentuk dasar dari desain gim dan membantu memastikan bahwa pemain ikut terlibat ataupun merasakan secara langsung. Berikut ini yang bukan elemen utama gim adalah ...
   a. Biaya
   b. Tujuan
   c. Aturan
   d. Tantangan
   e. Interaksi

5. Apakah kamu setuju atau tidak setuju dengan definisi gim? Jika kamu tidak setuju, apa yang akan kamu tambahkan, hapus, atau ubah?

6. Tentukanlah pernyataan yang benar atau salah terkait pentingnya GDD, dengan memberikan tanda centang (✓) pada kolom yang sesuai

No	Pernyataan	Benar	Salah
1	Membantu menambah pekerjaan dan lembur setiap anggota tim dalam memahami peran dan tanggung jawab mereka, dan meningkatkan efisiensi kerja.		
2	Berfungsi sebagai sebuah referensi yang tidak bisa diubah selama proses pengembangan dan membantu saat ada perubahan atau penambahan elemen.		
3	Membantu dalam pengaturan jadwal, pembagian tugas, dan pemantauan kemajuan proyek.		
4	Memberikan panduan terperinci tentang visi dan tujuan gim dan memastikan semua anggota tim memiliki pemahaman yang sama.		

7. Buatlah permainan kompetitif sederhana untuk dua pemain dan sebuah bola yang tidak melibatkan lemparan atau tendangan. Buktikan bahwa itu adalah sebuah permainan dengan menunjukkan bagaimana bola tersebut mengandung semua elemen penting.

8. Dengan menggunakan papan catur dan jenis-jenis bidak serta gerakan yang tersedia dalam catur, rancanglah semacam permainan kooperatif untuk dua orang, di mana mereka harus bekerja sama untuk mencapai kondisi kemenangan. Kamu tidak perlu menggunakan kondisi awal catur, atau semua bidak. Gambarkan aturan dan kondisi kemenangan yang kamu buat!.

9. Jelaskan elemen-elemen permainan dalam setiap permainan berikut: *mario*, *bowling*, dan *Botticelli*.

10. Identifikasilah sebuah gim yang kamu pilih terkait konsep gim dan desain dari gim tersebut. Kemudian kalian buatkan contoh GDD untuk gim tersebut!

Pengayaan

Untuk memperdalam pemahaman kamu tentang pemodelan gim ini, kamu bisa mencari contoh atau bentuk lain dari sebuah GDD melalui internet. Kemudian kamu lakukan analisis mendalam tentang GDD tersebut, mulai dari tema, genre, mekanika, naratif dan lainnya. Kegiatan yang kamu lakukan ini akan membantu kamu memperkaya pemahaman tentang konsep gim.

Video berikut ini bisa kamu gunakan untuk menambah wawasan dan keterampilan kamu: *Game Development and Production*, Erik Bethke, Wordware Publishing, Inc., 2003

Referensi Video:



1. **[Design] Skill Tree Online Episode 04: You'll Never Finish Your Game Design Document**

   https://www.youtube.com/watch?v=1xtw5y3NQIw

2. **Apa Itu Core Loop Pada Game?**

   https://www.youtube.com/watch?v=ApLYaWjLU3k

3. **Skill Tree Online #81 - Deconstructing Games as A Game Designer**

   https://www.youtube.com/watch?v=EWpJJzImPNA



Refleksi

Kamu bisa memberikan tanda centang (✓) pada kotak yang kamu anggap sesuai!

Setelah kamu mempelajari bab ini, bagaimanakah penguasaan kamu terhadap materi-materi berikut?

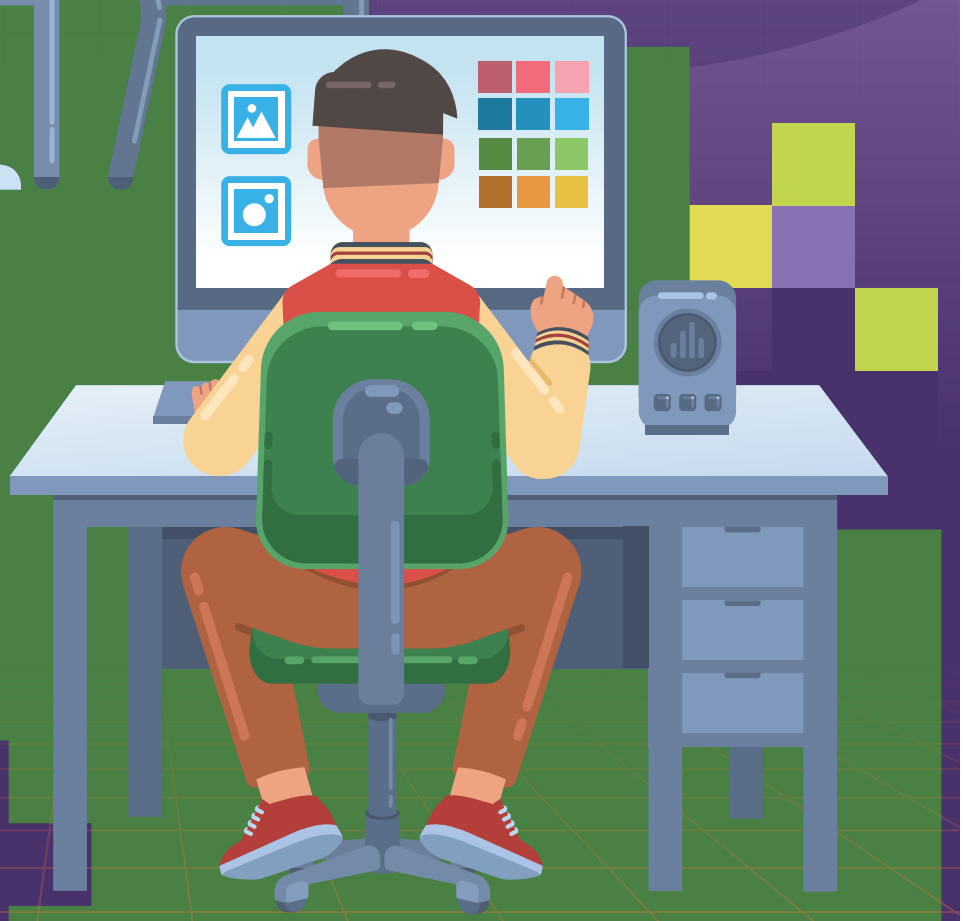
No	Materi	Tidak Menguasai	Menguasai	Sangat Menguasai
1	Konsep Gim			
2	Desain Gim • Mekanika Gim • Naratif Gim • Sistem Gim • Teknikal Gim • Level Gim • Riset Gim			
3	*Game Design Document* (GDD)			

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2024
Pengembangan Gim
untuk SMK/MAK Kelas XI-XII
Penulis: Rini Melati dan Muhammad Ardhan Fadlurrahman
ISBN: 978-634-00-0360-4

BAB 2

Perancangan Artistik Gim







Tujuan Pembelajaran

Setelah mempelajari materi dalam bab ini, kamu akan mampu menerapkan desain artistik (*key concept art)* menerapkan desain karakter (*character design)*, menerapkan desain latar belakang (*environment design)*, menerapkan desain properti (*property design)*, menerapkan perencanaan kebutuhan aset audio pada pemodelan gim, menerapkan *art design document*

Kata Kunci

Key Concept Art *Environment Design* *Property Design*

Character Design *Aset Audio* *Art Design Document*

Soft Skill

- **Cermat** dalam melakukan pengkategorian aset visual
- **Kreatif** dalam membuat beberapa ide untuk aset visual
- **Mampu mengkomunikasikan** dengan baik konsep aset visual yang dibuat
- **Disiplin dan tanggung jawab** dalam penyelesaian tugas pembuatan aset visual sesuai dengan waktu yang sudah ditentukan
- **Berwawasan luas** sehingga dapat merancang artistic gim dengan baik

Peta Materi





Gambar 2.1 Super Mario Odyssey
Sumber: Nintendo.com (2017)



Gambar 2.2 Gim Balap Kart Rider
Sumber: Gameuidatabase.com/Nexon (2020)

Apakah kalian pernah memainkan gim di atas? Super Mario dan Kart Rider merupakan salah satu gim yang memiliki visual menarik. Pernahkah kalian memainkan sebuah gim yang memiliki tampilan menarik dan membuat kalian nyaman dengan tampilan tersebut. Gim-gim tersebut melewati proses perancangan artistik gim sehingga menghasilkan visual yang baik, nyaman, dan membuat kamu menikmati visual pada gim tersebut. Dalam membuat tampilan menarik dan baik, sebuah permainan mario , kart rider, atau permainan yang kamu sukai dibuat dengan visual dan properti yang tepat sehingga membuat pemain akan terus memainkan gim tersebut. Melalui perancangan artistik, gim akan divisualisasikan seperti memvisualisasikan karakter, lingkungan, dan objek lainnya termasuk membuat aset audio dan dokumen desain artistiknya yang disesuaikan dengan tema permainan yang telah dipilih.

A. Desain Artistik Gim (*Key Concept Art*)

Desain artistik gim (*key concept art*) merupakan elemen dasar yang akan menjadi titik awal untuk melakukan pengembangan sebuah gim. Dalam desain artistik ini kamu akan menciptakan dan mengembangkan visual serta estetika dari gim. Semua ide-ide dan visi kreatif yang akan kamu bangun bisa kamu visualkan dengan desain artistik ini. kenapa kamu harus membuat desain artistik ini dan bagaimana kamu akan membuatnya? kamu bisa mengikuti pembahasannya dalam sub bab ini.



Ide-ide awal yang akan kamu buat terkait dengan karakter, lingkungan, objek, dan elemen lain dalam sebuah gim bisa kamu ilustrasikan secara visual. Hasil ilustrasi akan kamu gunakan sebagai jembatan antara ide kreatif dan implementasi teknis. Hal itu disebut dengan desain artistik atau *key concept art* (Tracy Fullerton, 2004).



Manfaat Membuat Desain Artistik Gim

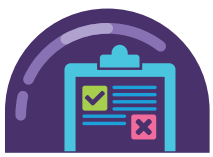


1

Membantu dalam Visualisasi

Membantu kamu dalam **memvisualisasikan secara jelas** tentang bagaimana elemen dalam gim akan terlihat, sehingga kamu dan tim lainnya dapat memiliki pemahaman yang sama tentang visi kreatif yang diinginkan.

2



Sebagai Panduan Desain

Menjadi **panduan desain** yang menyediakan referensi visual. Panduan desain dapat diikuti oleh tim selama proses pengembangan, sehingga akan menciptakan konsistensi dalam desain gim yang dikembangkan.

3

Memudahkan dalam Berkomunikasi

Memudahkan kamu dalam **berkomunikasi** dengan anggota tim lainnya, termasuk desainer, seniman, dan programmer, untuk mendiskusikan dan mengembangkan ide dengan lebih efektif.



Elemen Desain Artistik Gim



Setelah kamu mengetahui definisi dan manfaat dibuatnya desain artistik ini, kamu juga harus tahu elemen apa saja yang ada dalam desain artistik.



Gambar 2.3 Desain Karakter Seorang Bapak dalam Gim Mr. Alien Butuh Uang Jajan!

Sumber : Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Desain Karakter

Elemen desain karakter terkait dengan penampilan fisik, kostum, warna, ekspresi wajah, dan pose. Kamu akan membuat sebuah karakter yang kuat dan menarik sehingga pemain dalam gim akan terhubung secara emosional. Kamu harus meyakini bahwa pembuatan karakter yang baik hendaklah memiliki cerita dan latar belakang yang bisa mendukung desain visualnya. (Jesse Schell. 2008)

Gambar di samping adalah salah satu bentuk desain karakter alien yang ada dalam gim Mr. Alien. Karakter di atas yang merupakan bapak dari Mr.Alien yang tidak memberikan Mr. Alien jajan. Karakter ini nampak dengan warna kuning dengan baju coklat kusam seperti memiliki beban yang berat. Ekspresi dari karakter wmelambangkan seorang bapak yang bekerja dengan ekspresi datar tetapi tajam karena sudah melalui tantangan dan rintangan pada saat muda.



Lingkungan

Dalam elemen ini kamu akan membuat sebuah konsep tentang dunia di mana gim berlangsung. Termasuk di dalamnya detail arsitektur, flora, fauna, dan atmosfer untuk menciptakan pengalaman yang lebih mendalam bagi pemain gim.



Gambar 2.4 Desain Lingkungan dalam Mr. Alien Butuh Uang Jajan!
Sumber : Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Gambar di atas adalah salah satu contoh lingkungan Hutan Alien di mana Bapak Mr. Alien menaruh beberapa koin untuk ditemukan oleh Mr. Alien. Hutan ini terlihat seperti hutan pada umumnya tapi diberikan warna yang berbeda agar terlihat seperti hutan yang tidak umum karena ini adalah hutan alien, sesuai dengan karakter atau tema yang diusung. Mewakili kesesuaian dengan karakter hutan tersebut terdapat dua warna pohon yang memungkinkan terlihat seperti hutan yang berada di planet lain dengan tanah yang berpola.





Gambar 2.5 Desain Objek dan Item dalam Gim Mr. Alien Butuh Uang Jajan!
Sumber : Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

Objek dan Item

Dalam elemen ini kamu akan membuat desain untuk objek yang digunakan dalam gim, seperti senjata, alat, atau item yang dapat dikoleksi.



Gambar di atas adalah contoh objek koin dan item *power up*. Koin merupakan suatu objek yang harus ditemukan oleh Mr. Alien agar dapat memenuhi objektifnya yaitu mengumpulkan koin untuk dia jajan. *Power up* adalah item yang bisa ditemukan dan membuat Mr. Alien memiliki kekuatan lompat dua kali lipat dari biasanya.

Efek Visual

Dalam elemen ini kamu akan membuat efek visual, seperti cahaya, asap, elemen magis atau hal lainnya. Elemen ini perlu kamu perhatikan karena dalam dengan menggunakan elemen ini kamu harus bisa menciptakan suasana yang sesuai dengan tema gimnya. Gambar di bawah ini merupakan contoh *spritesheet* efek visual koin ketika koin ditemukan oleh Mr. Alien. *Spritesheet* adalah rangkaian gambar yang akan dipakai untuk melakukan 2D Animasi dengan membagi *spritesheet* menjadi satu gambar yang dapat berubah-ubah.



Gambar 2.6 Desain Efek Visual dalam Mr. Alien Butuh Uang Jajan!
Sumber : Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Gambar di atas merupakan contoh *spritesheet* efek visual koin ketika koin ditemukan oleh Mr. Alien. *Spritesheet* adalah rangkaian gambar yang akan dipakai untuk melakukan 2D Animasi dengan membagi *spritesheet* menjadi satu gambar yang dapat berubah-ubah.

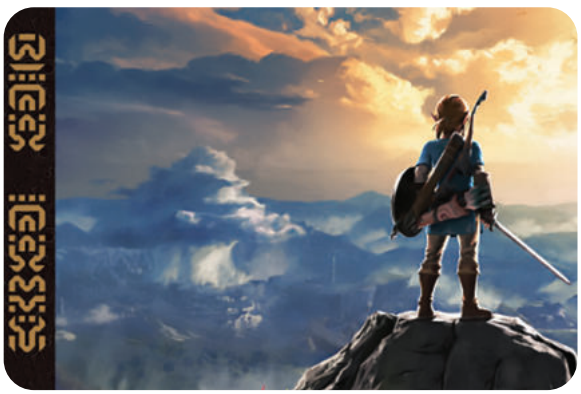
Dalam membuat desain artistik, ada beberapa langkah yang harus kamu perhatikan diantaranya:

1. **Melakukan riset** untuk memperoleh inspirasi agar kamu bisa membuat dan mengembangkan ide-ide yang lebih kaya dan beragam.
2. **Membuat sketsa awal** yang berguna dalam mengeksplorasi kemungkinan-kemungkinan desain yang akan digunakan sebelum kamu menyempurnakannya. sketsa awal ini adalah langkah penting yang perlu kamu lakukan agar kamu bisa menemukan arah yang tepat dalam desain.
3. **Revisi dan umpan balik** untuk melakukan penyesuaian dengan kebutuhan gim yang akan kamu kembangkan.

4. **Presentasi** kepada pemangku kepentingan seperti penerbit atau tim pemasaran, agar kamu bisa memastikan apakah desain yang kamu buat sudah memenuhi harapan dan visi yang diinginkan sehingga keputusan yang diambil tentang desain ini akan tepat dan sesuai.

Berikut ini adalah beberapa contoh gim yang memanfaatkan desain artistik yang efektif, diantaranya:

- **The Legend of Zelda**: desain yang digunakan sangat kaya dan beragam, sehingga dapat menggambarkan dunia yang sangat luas dan berwarna warni serta karakter yang sangat ikonik.
- **Dark Souls:** desain artistik dalam gim ini menggunakan karakter yang menakutkan, menciptakan suasana yang gelap, dan mendalam sesuai dengan tema dari gimnya.



Gambar 2.7 Contoh Desain Artistik yang Menggambarkan Dunia Berwarna Warni
Sumber: *The Legend of Zelda: Art & Artifacts/Oregon/Dark Horse Books* (2017)



Gambar 2.8 Representasi Desain Artistik yang Mencekam dan suasana yang Gelap
Sumber: FromSoftware/Dark Souls Design Works/Tokyo (2014)

1. Desain Karakter



Aktivitas 2.1 **Literasi Mandiri**

Gambar berikut ini adalah salah satu contoh desain karakter yang ada dalam gim Mr. Alien Butuh Uang untuk Jajan!.

Ada beberapa unsur penting yang bisa kamu gunakan untuk meningkatkan daya tarik sebuah desain karakter di antaranya kepribadian, penampilan fisik, ekspresi wajah, pakaian dan aksesori, latar belakang dan cerita serta kemampuan atau kekuatan khusus.

Coba kamu cari tahu penjelasan dari unsur penting yang ada dalam tabel berikut ini dengan melakukan pencarian informasi dari buku atau di internet melalui berita, artikel, jurnal atau video.



Gambar 2.9 Desain karakter Mr. Alien Butuh Uang Jajan!
Sumber : Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Unsur Penting	Penjelasan
Kepribadian	
Penampilan fisik	
Ekspresi wajah	
Pakaian dan aksesori	
Latar belakang dan cerita	
Kemampuan atau kekuatan khusus	

Desain karakter (*character design*) dalam pengembangan gim merupakan proses yang kompleks dan mendalam dengan melibatkan berbagai aspek kreatif dan teknis. Karakter berfungsi sebagai elemen sentral dalam banyak gim, yang akan mempengaruhi narasi, *gameplay*, dan pengalaman pemain. Proses yang kamu lakukan dalam menciptakan dan mengembangkan sebuah karakter secara visual dan naratif mencakup penentuan penampilan fisik, kepribadian, latar belakang, serta peran karakter dalam cerita merupakan pengertian dari desain karakter.

Lalu, apa tujuan kamu membuat desain karakter dalam pengembangan sebuah gim?

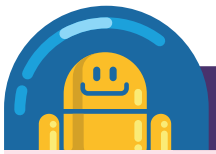
a

Menciptakan **keterikatan emosional**, ketika kamu bisa membuat sebuah desain karakter yang menarik dan berkembang maka hal ini akan dapat membantu pemain bisa merasa terhubung secara emosional dengan cerita yang ada dalam gim.

b

Mendukung narasi dari gim di mana desain karakter harus bisa selaras dengan cerita dan tema dari gimnya.

Dalam pembuatan desain karakter, ada beberapa elemen yang harus kamu perhatikan, di antaranya penampilan fisik, kepribadian dan karakter juga kemampuan dan atributnya.



Penampilan Fisik

Dalam elemen ini kamu harus bisa memperhatikan **bentuk dan proporsi** dari desain yang kamu buat, karena hal ini akan bisa menciptakan sebuah karakter yang unik. Contohnya, karakter tinggi dan ramping dapat menunjukkan keanggunan, sementara karakter pendek dan kekar menunjukkan kekuatan. Selain itu **pemilihan palet warna** juga harus kamu perhatikan dengan baik karena warna yang dipilih akan mencerminkan kepribadian dan tema dari karakter dalam gim. Contohnya, warna cerah dapat menunjukkan sifat ceria, sementara warna gelap dapat menggambarkan ketegangan atau misteri. Penampilan fisik lainnya yang harus jadi perhatian kamu yaitu **pakaian dan aksesori**, karena desain kostum dapat mencerminkan budaya, latar belakang, dan peran karakter dalam cerita. Aksesori juga dapat memberikan informasi tambahan tentang karakter.



Kepribadian dan Karakter

Kamu harus memahami apa yang menjadi motivasi karakter dan tujuannya dalam cerita sebuah gim, karena hal ini akan membantu dalam menciptakan dialog dan interaksi yang konsisten dalam permainannya. selain itu kamu juga harus bisa melakukan pengembangan terhadap karakter dalam alur ceritanya.



Kemampuan dan Atribut

Dalam elemen ini kamu harus bisa membuat desain karakter yang memiliki kekuatan dan kelemahan yang dapat mempengaruhi *gameplay*. Contohnya, karakter yang cepat tetapi lemah dalam pertahanan dapat memengaruhi cara pemain mendekati tantangan dalam gim. Selain kekuatan dan kelemahan, kamu harus bisa membuat karakter yang mencerminkan keunikan karakter tersebut. seperti, karakter dengan kemampuan magis akan memungkinkan memiliki elemen visual yang menunjukkan hubungan mereka dengan kekuatan tersebut.

Aktivitas 2.2 **Aktivitas Mandiri**

Setelah kamu memahami tentang desain karakter, cobalah kamu membuat sebuah desain karakter kreasimu sendiri, dengan memperhatikan tujuan dan elemen-elemen dalam pembuatan desain karakter! kamu boleh menggunakan aplikasi yang pernah kamu gunakan dalam membuat sebuah desain gambar. kemudian tunjukkan dan ceritakanlah pada temanmu tentang desain karakter tersebut! Semangat berkreasi ya.

2. Desain Latar Belakang



Selain desain karakter, ada desain latar belakang atau *environment design* yang harus kamu ketahui dalam pembuatan sebuah gim. Desain latar belakang mencakup penciptaan dan perancangan dunia gim di mana *gameplay* berlangsung. Penciptaan lingkungan visual tidak hanya mendukung cerita dan karakter, tetapi juga memengaruhi mekanika *gameplay* dan pengalaman pemain secara keseluruhan.



Gambar 2.10 Desain Latar Belakang dalam Gim Mr. Alien Butuh Uang Jajan!
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Gambar di atas merupakan salah satu desain latar belakang yang ada dalam gim Mr. Alien minta jajan. dengan melihat gambar di atas kamu akan tahu ada proses yang harus kamu lakukan untuk menciptakan dan mengatur elemen-elemen fisik dari dunia gim termasuk di dalamnya lokasi, struktur, iklim, dan atmosfer. Semua ini akan menjadi elemen yang membentuk latar belakang permainan, dari pemandangan luar hingga detail kecil dalam ruangan.

Seperti halnya desain karakter, dalam pembuatan desain latar belakang juga memiliki beberapa **tujuan** di antaranya:

1. **Menciptakan atmosfer** yang bisa menimbulkan suasana tertentu yang mendukung tema dan cerita dari gim.
2. **Mendukung narasi** yang dapat membantu dalam menceritakan kisah. Contohnya, lokasi yang ditinggalkan dapat menciptakan rasa kesepian atau misteri.
3. **Memengaruhi *gameplay*** dalam gim sehingga juga akan memengaruhi cara pemain berinteraksi dengan dunia, termasuk navigasi, strategi, dan tantangan yang dihadapi.

Setelah kamu mengetahui tujuan dibuatnya desain latar belakang ini, selanjutnya kamu perlu mempertimbangkan elemen-elemen berikut dalam melakukan desain latar belakang sebuah gim, di antaranya latar belakang dan tema, struktur dan arsitektur, elemen alam dan juga detail dan aksesori.



Latar Belakang dan Tema

Dalam membuat desain latar belakang kamu harus bisa membuat desain lingkungan yang selaras dengan narasi. Misalnya, dunia pasca-apokaliptik akan berbeda jauh dari dunia fantasi yang cerah. Kemudian gaya visual yang kamu gunakan dalam memilih gaya artistik, seperti realisme, kartun, atau *stylized*, juga harus kamu perhatikan karena hal ini akan dapat memengaruhi bagaimana lingkungan dirancang.



Struktur dan Arsitektur

Bentuk dari struktur dan arsitektur yang bisa kamu buat di antaranya merancang gedung, jembatan, dan elemen arsitektur lain yang menciptakan suasana dan mendukung *gameplay*. kemudian kamu juga harus menentukan skala untuk ukuran dan proporsi elemen-elemen dalam lingkungan guna menciptakan rasa kedalaman dan ruang.



Elemen Alam

Kamu bisa menggunakan flora, seperti pohon, semak, dan tanaman, untuk menambah kehidupan pada lingkungan. Jenis tanaman yang dipilih harus sesuai dengan tema dan lokasi. kemudian kamu juga harus bisa menentukan bagaimana kondisi cuaca (hujan, salju, terik matahari), hal ini juga bisa mempengaruhi suasana dan *gameplay*. Elemen alam dapat berpengaruh pada visibilitas dan navigasi.



Detail dan Aksesori

Kamu bisa menambahkan elemen berupa objek interaktif yang dapat berinteraksi dengan pemain, seperti pintu, tombol, atau item *collectible*. kemudian kamu juga perlu menentukan detail dekorasi dan lingkungan kecil seperti poster, furnitur, atau alat agar dapat memberikan konteks lebih dalam dan memperkaya pengalaman visual.

Aktivitas 2.3 **Aktivitas Mandiri**

Kamu sudah memahami tentang desain latar belakang, tujuan dan elemen pembuatannya. Nah, sekarang cobalah kamu buat sebuah desain latar belakang sesuai kreasimu. (bisa disamakan tema yang kamu pilih di aktivitas 2.2). Kamu bisa menggunakan aplikasi yang pernah kamu gunakan untuk membuat desain gambar. Setelah itu, coba kamu berbagi cerita dengan teman-temanmu tentang konsep dari desain latar belakang yang kamu buat Semangat berkreasi!

3. Desain Properti



Desain properti atau *property design* adalah aspek krusial dalam pengembangan gim, mencakup desain elemen-elemen statis yang ada di dalam dunia gim. Elemen-elemen ini, sering disebut sebagai "*props*", meliputi objek, item, atau elemen lingkungan yang mendukung *gameplay*, narasi, dan estetika visual.

Proses yang kamu lakukan untuk menciptakan dan mengembangkan berbagai objek dalam gim yang bukan merupakan karakter atau lingkungan utama disebut dengan desain properti. Desain properti mencakup furnitur, alat, kendaraan, dan barang *collectible*. perlu kamu ketahui bahwa meskipun *props* sering dianggap sebagai elemen kecil, tetapi mereka mempunyai peran besar dalam menciptakan suasana dan meningkatkan interaksi pemain dengan dunia gim.

Ada beberapa **tujuan** mengapa kamu harus membuat desain properti, di antaranya:

1. **Mendukung narasi** sebuah gim, properti dapat membantu menceritakan kisah. Seperti, sebuah surat yang ditemukan di meja dapat memberikan latar belakang karakter atau konflik.
2. **Menciptakan suasana** tertentu yang memberikan rasa keaslian dan kedalaman pada lingkungan.
3. **Menciptakan interaksi dengan pemain** sehingga menambah dinamika *gameplay* pada gim.
4. **Mengisi ruang** dalam dunia gim, menjadikannya lebih hidup dan realistis.

Berikut ini beberapa **elemen** yang penting untuk kamu pertimbangkan dalam membuat desain properti, di antaranya:



Fungsi dan Tujuan

Interaktif vs Statis

Penentukan sifat dari properti merupakan langkah awal yang penting. Kamu perlu menentukan apakah properti akan bersifat interaktif seperti pintu yang bisa dibuka atau statis seperti meja yang hanya berfungsi sebagai dekorasi.

Penggunaan dalam *Gameplay*

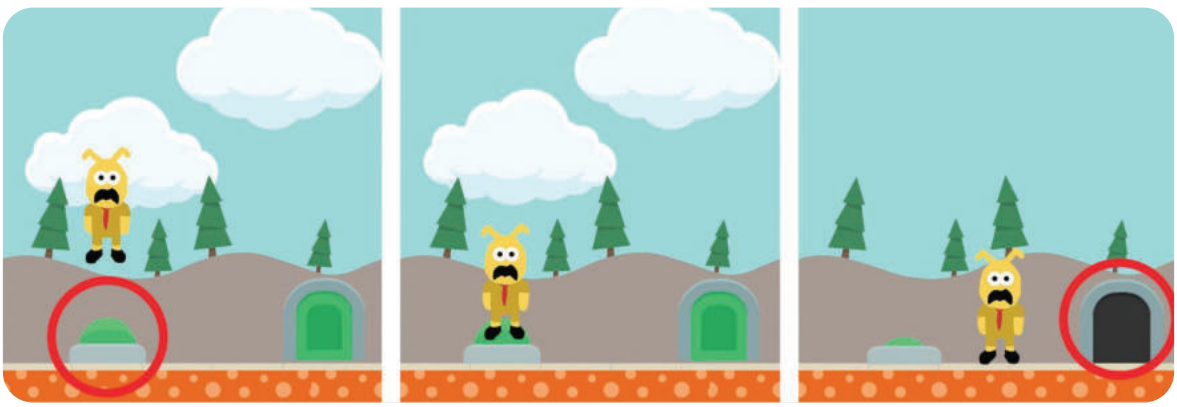
Kamu juga perlu memikirkan bagaimana properti akan digunakan dalam *gameplay*, termasuk interaksi pemain.

Desain Visual

Bentuk dan gaya visual properti harus selaras dengan tema dan gaya keseluruhan gim. Seperti pemilihan warna, material, dan tekstur. Kamu juga dapat menambahkan detail seperti goresan, kerusakan, atau dekorasi. Detail-detail tersebut memberikan karakter dan ciri khas pada objek.

Konteks dan Cerita

Properti harus sesuai dengan latar tempat di mana mereka berada. Misalnya, barang-barang di dunia futuristik akan berbeda dari barang-barang di setting sejarah. Melalui properti juga dapat menceritakan unsur tertentu dari sebuah cerita, hal ini memberikan konteks tambahan kepada pemain.



Gambar 2.11 Desain Properti dalam Gim Mr. Alien Butuh Uang Jajan
Sumber : Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Pada gambar di atas, terdapat property dalam tombol yang jika diinjak, maka pintu akan terbuka. Suatu objek di dalam gim yang dapat berinteraksi dengan pemain dan memberikan output dinamakan *props* atau properti.

Aktivitas 2.4 Aktivitas Mandiri

Untuk mengetahui pemahamanmu tentang desain properti ini, cobalah kamu membuat sebuah desain properti versimu sendiri ya! Desain properti yang kamu buat bisa berkaitan dengan karakter dan latar belakang pada hasil aktivitas sebelumnya. Kemudian, ceritakanlah di depan kelas tentang desainmu serta catatlah saran dan masukan untuk desain properti kamu.

Aktivitas 2.5 Kerja Tim

Buatlah sebuah tim yang beranggotakan 4 orang. Kemudian kalian ikuti petunjuk kerja berikut ini:

1. Tentukanlah sebuah tema desain artistik yang kalian buat.
2. Buatlah desain karakter, desain latar belakang dan desain properti dengan menyesuaikan tema yang telah kalian tentukan tadi dengan memperhatikan unsur-unsur dan urutan pembuatan desainnya.
3. Setelah itu, kalian presentasikan hasil karya kalian di depan kelas untuk saling berbagi kreativitas dengan teman yang lainnya.

B. Aset Audio

Langkah penting lainnya yang perlu kamu persiapkan dalam membuat sebuah gim adalah perencanaan kebutuhan aset audio. Hal ini untuk memastikan semua elemen audio mendukung *gameplay* dan narasi secara efektif. Beberapa proses yang dilakukan di antaranya identifikasi, perencanaan, dan pengorganisasian semua aset audio yang diperlukan untuk menciptakan pengalaman bermain yang lebih mendalam. Berikut ini adalah beberapa langkah-langkah yang perlu kamu lakukan dalam merencanakan kebutuhan aset audio.



Langkah–Langkah Perencanaan Kebutuhan Aset Audio



Melakukan Analisis Kebutuhan Audio

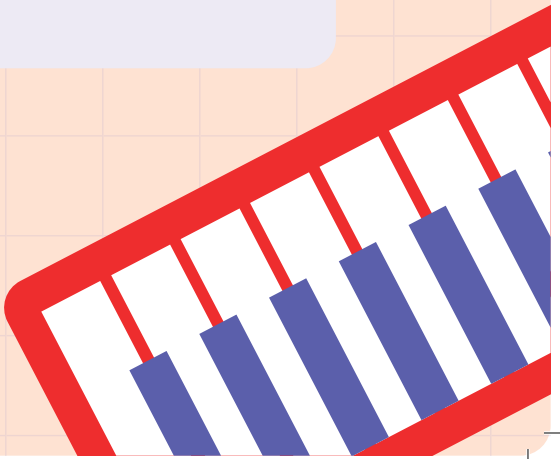
Dalam proses ini kamu perlu mengidentifikasi genre dan tema dari gim yang akan kamu buat. apakah bergenre aksi, petualangan, horor dan lainnya. Langkah ini perlu kamu lakukan karena ini akan mempengaruhi jenis audio yang dibutuhkan.

Setiap genre memiliki konvensi audio tersendiri yang dapat meningkatkan pengalaman bermain. Selain itu kamu juga harus bisa menentukan tema dan atmosfer yang akan diciptakan dalam gim. seperti, gim horor membutuhkan efek suara yang menegangkan dan gelap, sedangkan gim petualangan mungkin memerlukan musik yang ceria dan cerah. Dalam gim Mr. Alien jenis audio yang digunakan cenderung ceria karena cerita yang ditawarkan dalam Mr.Alien cukup ringan dan tidak terlalu kompleks serta target utama dari gim Mr.Alien tersebut adalah semua umur, sehingga dengan musik ceria diharapkan pemain akan lebih memiliki mood yang ceria juga ketika memainkan gim tersebut.



Mengidentifikasi Aksi pada Gameplay dan Menambah Imersi dalam Interaksi

Kamu harus bisa memahami aksi utama yang dilakukan pemain dalam gim seperti berlari, melompat, atau bertarung, dan jenis suara yang sesuai untuk setiap aksi. Kemudian tentukan bagaimana pemain berinteraksi dengan objek dan lingkungan. Hal ini mencakup suara yang muncul saat berinteraksi, yang dapat menambah kedalaman *gameplay* dan pengalaman pemain.



Melakukan Klasifikasi Aset Audio

Audio memiliki beberapa klasifikasi, berikut beberapa jenis audio dalam gim.

a Musik Latar

Kamu perlu menentukan jenis musik latar yang dibutuhkan, seperti musik latar untuk eksplorasi atau musik latar untuk pertarungan. Setiap segmen permainan harus memiliki musik yang sesuai untuk membangun suasana hati pemain. Kamu juga perlu merencanakan durasi setiap trek musik dan variasi yang diperlukan untuk menghindari kebosanan. Musik yang monoton dapat mengganggu pengalaman bermain.

b Efek Suara

Efek suara dalam subuah gim seperti langkah, suara senjata, suara interaksi, dan *ambient*. Efek suara yang tepat dapat meningkatkan realisme dan responsivitas gim. Buatlah daftar semua efek suara yang kamu perlukan sehingga produksi audio dapat berjalan dengan lancar. Jangan lupa untuk menentukan detail yang kamu inginkan untuk setiap efek suara, apakah akan direkam secara langsung atau menggunakan *sound library*. Kualitas audio yang tinggi sangat penting untuk pengalaman bermain yang lebih mendalam

c Dialog

Lakukan identifikasi semua dialog yang diperlukan, termasuk percakapan, monolog, dan reaksi karakter. Kamu harus menulis dialog dengan baik, karena dapat memperkuat narasi. Pertimbangkan bahasa yang digunakan dalam gim dan apakah akan ada dubing untuk karakter, serta kebutuhan untuk perekaman suara. Ketersediaan dan dukungan pada bahasa tertentu Ini penting untuk aksesibilitas lebih jauh untuk menjangkau pasar global.

d Suasana (Ambience)

Tentukan suara latar yang akan menambah atmosfer, seperti suara alam, keramaian, atau suara cuaca. Suara *ambience* dapat menciptakan nuansa yang lebih dalam. Kamu perlu merencanakan bagaimana suara *ambience* akan berulang dan bagaimana transisi akan dilakukan antara berbagai suasana. Transisi yang mulus antara suara dapat mencegah gangguan pengalaman.

Implementasi Aset Audio

Audio memiliki beberapa klasifikasi, berikut beberapa jenis audio dalam gim.

a Integrasi ke dalam Game Engine

Sebelum mengintegrasikan audio ke dalam *game engine*, Pastikan semua aset audio dalam **format yang sesuai** misalnya, WAV atau OGG. Format yang tepat penting untuk kinerja dan kualitas. Di dalam gim Mr. Alien, terdapat dua format file yang digunakan. Format WAV dengan kualitas suara yang lebih bagus tetapi lebih besar ukuran filenya, akan dipakai sebagai format file musik di dalam gim. Sedangkan OGG yang merupakan format file dengan kualitas yang lebih rendah tetapi dengan ukuran file yang lebih rendah juga, akan dipakai untuk berbagai efek suara yang cukup banyak tapi tidak membuat file menjadi sangat besar.

Dalam mengembangkan aset audio, kamu perlu mengatur aset audio dalam folder yang terstruktur untuk memudahkan akses. Pengorganisasian aset yang baik dapat mengurangi waktu pencarian dan meningkatkan produktivitas .

b Pengaturan dan Penyesuaian

Pengaturan dan penyesuaian audio itu penting untuk membuat pengalaman bermain jadi lebih seru. Pertama, **atur level volume** dengan cermat. Pastikan musik, efek suara, dan dialog terdengar seimbang. Keseimbangan yang baik dapat meningkatkan pengalaman bermain. Jika kamu ingin gim lebih interaktif, coba **gunakan scripting.** Scripting digunakan untuk mengatur pemicu audio. Misalnya, saat karakter kamu menyentuh atau menggunakan item, maka akan muncul audio tertentu. Dengan interaktivitas dan responsivitas yang diperoleh, pemain akan merasa terhubung dengan gim.

c Dokumentasi Aset Audio

Apakah kamu pernah bingung tentang bagaimana mengelolal aset audio di gim yang sedang kamu kembangkan? Jika kamu bingung mungkin kamu perlu **membuat dokumentasi**. Pertama, kamu perlu mencatat semua detail tentang aset audio. Mulai dari sumber, format, dan penggunaan dalam gim. Di dalam Gim Mr.Alien, terdapat dua jenis format yang digunakan untuk tempat yang berbeda. Jika ingin menambahkan audio musik maka format yang diperlukan adalah WAV, tetapi jika ingin menambah format efek suara maka format yang diperlukan adalah OGG. Jadi dengan dokumentasi ini kamu bisa menambahkan audio dengan format yang sesuai. Dengan dokumentasi yang rapi, kamu akan mudah untuk menemukan atau merujuk data kapan saja.

Selain dokumentasi, kamu juga perlu **membuat sebuah panduan**. Panduan ini tentang bagaimana menggunakan dan mengintegrasikan aset audio. Panduan ini tidak hanya untuk kamu tapi juga anggota tim lainnya, sehingga kamu perlu membuat panduan yang sangat jelas. Pada gim Mr.Alien sudah terdapat panduan, sehingga pengembang dapat tau format apa digunakan dan untuk apa.

d Pemeliharaan Aset Audio

Pernahkah kamu mendengar istilah pemeliharaan aset audio? Setelah gim dirilis, pekerjaan tidak langsung selesai. Saatnya kamu untuk merencanakan pemeliharaan dan pembaharuan pada aset audio. Setelah pengembangan gim Mr.Alien selesai, maka akan dilanjutkan dengan riset apakah ukuran size file musik dapat di compress tanpa mengubah format. Hal ini penting karena akan membantu kamu dalam mengelolal ukuran file gim agar tetap efisien. Dengan pembaruan yang teratur akan dapat meningkatkan kepuasan pemain.

Hal lainnya yang tidak kalah penting, mencatat *bug* atau masalah yang muncul terkait audio dan merencanakan perbaikan dalam pembaruan mendatang. Dalam pengembangan gim Mr.Alien terdapat kasus suatu efek suara lompat tidak berjalan ketika player menekan input lompat, hal ini yang dinamakan sebuah *Bug. Bug* adalah suatu peristiwa di mana terjadi hal yang tidak diharapkan dan tidak berfungsi sebagaimana mestinya. Pada kasus Mr. Alien adalah efek suara lompat tidak berjalan, hal ini perlu dicatat agar dapat dilakukan perbaikan dalam pembaruan mendatang. Pemeliharaan berkelanjutan adalah bagian penting dari siklus hidup gim.

Aktivitas 2.6 Kerja Tim

Buatlah sebuah tim yang beranggotakan 3 orang, kemudian cobalah kalian membuat sebuah aset audio yang temanya kalian sesuaikan dengan aktivitas 2.5. Selamat berkreasi ya!

C. Dokumen Desain Artistik (*Art Design Document*)



Mungkin kamu bertanya-tanya, apa itu dokumen desain artistik atau *Art Design Document*? Kenapa desain artistik harus didokumentasikan? Dan bagaimana membuat sebuah dokumen desain artistik? *Art design document* adalah dokumentasi semua aspek visual dan artistik dari proyek sebuah gim. Hal ini merupakan proses penting yang perlu dilakukan. Nantinya, Dokumen ini berfungsi sebagai panduan untuk kamu memastikan konsistensi sebuah desain dan menyelaraskan visi artistik di seluruh elemen gim yang kamu buat.

Ada beberapa **tujuan** mengapa sebuah dokumen desain artistik dibuat, di antaranya:

1. Sebagai **panduan visual** yang akan menjadi pedoman dari sebuah desain visual, gaya, warna, dan elemen grafis. Kamu dan tim lainnya akan berpedoman pada panduan ini, agar memiliki pemahaman yang sama terkait gaya dan estetika gim, dan harapan desain yang sama serta memperkuat konsistensi visual dalam gim.
2. Terciptanya **konsistensi visual** dalam gim, hal inisangat penting dalam menciptakan pengalaman bermain yang terintegrasi dan menarik.
3. Memfasilitasi kamu dalam **berkomunikasi** dengan anggota tim lainnya, termasuk seniman, desainer, dan pengembang. Dengan dokumen ini, semua anggota tim dapat merujuk ke sumber yang sama dan hal ini akan dapat mengurangi kebingungan.
4. Menjadi **referensi** selama seluruh proses pengembangan, dari perencanaan hingga pengujian. Dengan ini, kamu dan tim akan terbantu untuk tetap fokus dan di arah yang jelas sepanjang proses kreatif dilakukan.

Bagian–Bagian Dalam Dokumen Desain Artistik

1. Ringkasan Proyek



Pada bagian ini, kamu akan menentukan judul dan deksripsi gim yang akan kamu buat. Nama gim dan deskripsi yang sedang dikembangkan harus jelas dan mencolok. Nama dan deskripsi gim akan menciptakan identitas awal atau memberikan gambaran awal tentang apa saja yang diharapkan dari gim yang kamu buat (Stevens, 2020).

2. Visi Artistik



Disini kamu akan mendeskripsikan tentang gaya visual yang diinginkan dalam gim, seperti realistis, kartun, atau retro. Deskripsi ini untuk membantu kamu menetapkan nada artistik dari awal.

3. Desain Karakter



Pada bagian ini akan ada dijelaskan mengenai setiap karakter, termasuk latar belakang, motivasi, dan peran dalam cerita. Selain itu juga kamu bisa membuat sebuah sketsa dan ilustrasi yang menunjukkan desain fisik karakter, ekspresi wajah, dan variasi kostum. Ini memberikan gambaran visual yang lebih konkret. Karakter yang kuat memerlukan pemahaman mendalam tentang sifat dan tujuan mereka (Davis, 2022).

4. Desain Lingkungan



Pada bagian ini, kamu akan menyajikan latar belakang dan lokasi dari gim yang sudah kamu tentukan. Deskripsi lokasi-lokasi dalam gim, termasuk suasana dan tema masing-masing akan membantu menciptakan dunia yang lebih imersif dalam sebuah gim (Holt, 2019). Selain itu kamu juga bisa menggambarkan konsep lingkungan dari gim yang dibuat. Gambaran dapat berupa ilustrasi dan sketsa dari berbagai area dalam gim, seperti kota, hutan, atau ruang bawah tanah, yang dapat memberikan panduan visual untuk lingkungan gimmu (Meyer, 2018).

5. Desain Properti



Disini akan dicantumkan daftar properti dan desain visual gim. Buatlah dengan lengkap semua objek yang diperlukan dalam gim, seperti alat, kendaraan, furniture, dan lainnya. Jangan lupa untuk membuat sketsa dan deskripsi dari setiap properti, termasuk fungsinya dalam gameplay. Daftar ini akan membantu kamu dan tim untuk bisa memastikan bahwa semua elemen visual terencana dengan baik.

6. UI/UX Design



Pada bagian ini merupakan deskripsi dari antarmuka pengguna dan pengalaman pengguna. Antarmuka pengguna (UI) seperti menu, tombol, dan indikator. Sedangkan untuk pengalaman pengguna (UX) berisi penjelasan tentang bagaimana pengguna akan berinteraksi dengan antarmuka dan elemen visual. UX yang baik penting untuk kepuasan pemain (Rogers, 2014).

7. Skema Warna dan Palet



Pada bagian ini saatnya kamu menentukan palet warna dan psikologi warna yang akan digunakan. Palet warna yang dimaksud, termasuk penggunaan warna untuk karakter, lingkungan, dan elemen UI. Pemilihan warna yang tepat dapat mempengaruhi pengalaman pemain (Kessler, 2016). Kemudian jelaskan bagaimana warna yang kamu pilih akan mempengaruhi suasana dan pengalaman pemain. Warna memiliki dampak emosional yang kuat dalam desain gim (Phelps, 2019).

8. Animasi dan Efek Visual



Bagian ini akan memberikan deskripsi tentang jenis animasi yang akan digunakan, apakah 2D, 3D, frame-by-frame dan lainnya. Gaya animasi harus konsisten dengan visi artistik (Benshop, 2021). Selain itu, tambahkan rincian tentang efek visual yang dibuat seperti efek khusus, seperti ledakan, sihir, atau efek cuaca. Efek visual yang baik dapat meningkatkan pengalaman bermain yang mendalam bagi pemain (D'Angelo, 2020).

9. Referensi dan Riset



Disini dicantumkan hasil riset pasar dan referensi gaya. Di dalam termasuk informasi tentang kompetisi dan tren dalam industri yang relevan dengan gim. Memahami pasar membantu dalam menciptakan produk yang sukses (Greene, 2015). Kemudian dalam referensi gaya akan dapat membantu dalam menciptakan elemen yang unik yang akan mempengaruhi desain visual.

Berikut ini beberapa contoh Pengunaan *Art Design Document* (ADD) dalam menentukan arah artistik sebuah gim

a. The Legend of Zelda:

Gim ini menggunakan ADD untuk merencanakan gaya visual yang unik dan menciptakan lingkungan yang beragam. Desain yang detail dan terarah membantu menciptakan pengalaman yang imersif (Williams, 2022).

b. Cuphead:

Cuphead mendokumentasikan gaya seni 1930-an dengan ilustrasi tangan dan animasi yang khas. ADD sangat penting dalam menjaga konsistensi gaya visual yang unik ini (Kowalski, 2017).

c. Journey:

Journey menggunakan palet warna yang konsisten dan desain karakter minimalis untuk menciptakan pengalaman emosional yang mendalam. ADD membantu tim menjaga fokus pada aspek visual yang paling penting (Miller, 2020).

Untuk membuat kamu lebih memahami bagaimana dokumen desain artistik itu disajikan, berikut adalah contoh atau referensi ADD untuk gim Mr.Alien Butuh Uang Jajan:

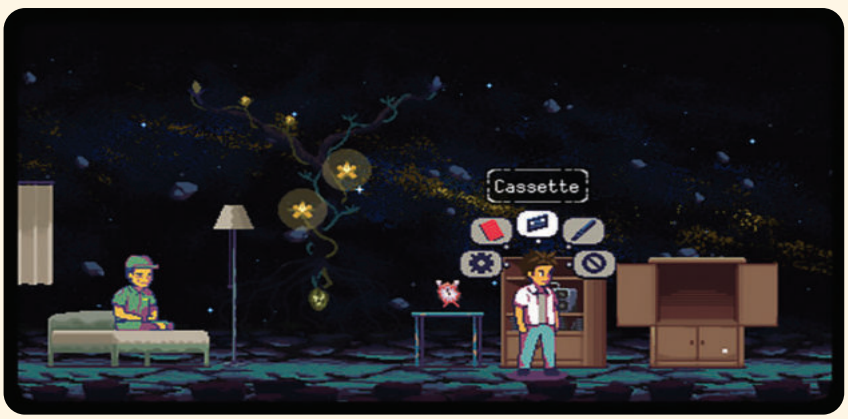
Tabel 2.1 Contoh pembuatan ADD terkait gim "Mr Alien Butuh Uang untuk Jajan!"

Komponen	Deskripsi	Detail Tambahan
Ringkasan Project	Sebuah gim aksi platformer 2D yang berjudul "Mr Alien Butuh Uang jajan", Pemain akan menjelajah beberapa tempat untuk mengumpulkan koin agar Mr. Alien bisa jajan semau hatinya. Koin tersebut disebarkan oleh Bapak dari Mr. Alien agar anaknya tetap aktif keluar rumah dan berolahraga.	Mengandung elemen cerita yang mendalam, dengan berbagai tantangan untuk Mr. Alien.
Visi Artistik	Gaya visual dari gim ini adalah minimalis, ringan dan kontras untuk menciptakan suasana ceria dan penuh tantangan	Menggunakan tekstur berlapis untuk memberikan kedalaman dan juga teknis cahaya yang menambah imersif.
Desain Karakter	Karakter Utama: Seorang Alien berwarna hijau bernama Mr. Alien Karakter Pendukung: Alien berwarna kuning yang merupakan Bapak dari Mr. Alien.	karakter memiliki latar belakang yang menarik dan kemampuan khusus yang dapat mendukung gameplay dari gim.

Komponen	Deskripsi	Detail Tambahan
Desain Latar Belakang	Terdapat beberapa tempat bioma Hutan Alien, Gua Gelap, maupun Kota Alien di mana masing-masing akan memberikan sebuah pemandangan berbeda. Hutan Alien akan digambarkan sebagai Hutan lebat, Gua Gelap akan digambarkan sebagai sebuah gua dengan banyak interaksi dan lubang rahasia, sedangkan Kota Alien digambarkan sebagai kota gemerlap dan hidup sebagai tempat terakhir.	Masing-masing bioma memiliki bangunan, flora, fauna, dan arsitektur spesifik untuk menciptakan pengalaman eksplorasi yang kaya.
Desain Properti	Objek yang dapat diambil seperti Koin, Power Up ataupun yang bisa berinteraksi dengan pemain seperti tuas, tombol, maupun kunci untuk membuka pintu dan maju ke tantangan selanjutnya.	Setiap properti memiliki fungsi dan interaksi yang berbeda dalam gameplay, seperti meningkatkan kekuatan lompat ataupun membuka pintu.
Desain UI/UX	User Interface atau Antarmuka yang intuitif dengan simbol yang mudah dimengerti. Menu yang mudah diakses dan animasi transisi yang halus. User Experience atau pengalaman pengguna merupakan pengalaman yang dirasakan pemain saat berinteraksi dengan gim tersebut. Mencakup aspek kenyamanan, kemudahan pengguna, daya tarik, hingga aspek emosionaldan diperoleh melalui wawancara kusioner angker dan sebagainya.	Desain UI akan minimalis, sehingga tidak mengganggu permainan dan hanya informasi yang diperlukan saja yang akan ditampilkan di satu waktu.
Skema Warna dan Palet	Palet warna cerah seperti: hijau daun, cokelat tanah, kuning cerah, dan biru langit. Warna yang berbeda untuk menandai area dan objek yang bisa berinteraksi dengan pemain.	Palet warna digunakan untuk memandu pemain melalui level di setiap bioma, dengan warna yang lebih gelap di area berbahaya untuk memberikan informasi kesulitan tanpa mengganggu permainan.
Animasi dan Efek Visual	Animasi berjalan yang halus untuk Mr. Alien sebagai karakter utama. Efek visual seperti 2D Partikel yang muncul ketika mengambil koin	Efek partikel dapat digunakan untuk memberikan umpan balik visual pada interaksi, seperti saat koin diambil atau saat karakter beraksi.
Referensi dan Riset	Menggunakan referensi dan seni lukis minimalis dengan target market semua umur dan spesifik ke anak-remaja di mana objek tidak terlalu kompleks dan juga studi tentang ekosistem hutan untuk Hutan Alien, ekosistem gua untuk Gua Gelap, dan tata kota untuk Kota Alien.	Melakukan wawancara kepada target market sehingga bioma dapat diatur sesuai dengan insight yang didapat setelah melakukan wawancara target market terkait. Wawancara dilakukan untuk mendapatkan insight yang menjadi acuan.

Aktivitas 2.7 Literasi Bersama

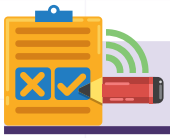
Buatlah sebuah tim beranggotakan maksimal 5 orang. Kemudian kalian perhatikan gambar berikut ini :



Gambar 2.12 Contoh Gim *Space for the Unbound*
Sumber : Mojiken Studio/Toge Production (2023)

Buatlah sebuah dokumen desain artistic atau ADD untuk sebuah gim di atas bersama tim kalian dengan memperhatikan bagian-bagian pembuatan ADD. Kalian bisa mencari referensi pembuatan ADD lainnya di internet. Kemudian, isilah data berikut ini :

1. Ringkasan Project: Penjelasan singkat tentang proyek, tujuan dokumen, dan ringkasan visi artistik.
2. Gaya Visual: Deskripsi tentang gaya visual yang diinginkan, termasuk inspirasi, referensi artistik, dan palet warna.
3. Desain Karakter: Rincian desain karakter.
4. Desain: Terdiri dari Desain latar belakang , property dan UI/UX
5. Referensi dan Inspirasi: Koleksi gambar, foto, dan karya seni lain yang menjadi inspirasi visual untuk proyek
6. Animasi dan Efek Visual: Efek visual seperti 2D Partikel yang muncul ketika mengambil koin.



Uji Kompetensi

Coba kamu jawab pertanyaan di bawah ini untuk mengetahui apakah kamu sudah menguasai materi tentang Perancangan Artistik Gim.

1. Desain Artistik (*Key Concept Art)* adalah elemen fundamental dalam pengembangan gim yang menjadi titik awal untuk menciptakan dan mengembangkan visual serta estetika game. Melalui Key Concept Art, pengembang dapat memvisualisasikan ide-ide mereka dan menyampaikan visi kreatif kepada seluruh tim. Berikan alasan kenapa *key concept art* dapat membantu visualisasi, menjadi panduan desain, dan dapat memfasilitasi komunikasi!
2. Apa saja yang harus diperhatikan dalam membuat sebuah desain properti?
3. Berikan tanda centang (✓) di kolom yang sesuai dengan pernyataan-pernyataan berikut!

Pertanyaan	Benar	Salah
Desain properti merujuk pada proses menciptakan dan mengembangkan berbagai objek dalam gim yang bukan merupakan karakter atau lingkungan utama. Ini termasuk furniture, alat, kendaraan, dan barang *collectible*		
Perencanaan kebutuhan aset audio adalah langkah penting dalam pengembangan gim yang memastikan bahwa semua elemen audio mendukung *gameplay* dan narasi secara efektif. Proses ini hanya melibatkan identifikasi, perencanaan		
Karakter berfungsi sebagai elemen sentral dalam banyak gim, mempengaruhi narasi, *gameplay*, dan pengalaman pemain		

4. Menurut kamu, kapan desain latar belakang itu bisa digunakan?
   a. Pada saat akan melakukan sebuah proses menciptakan lingkungan visual atau setting tempat cerita atau adegan yang akan berlangsung
   b. Saat semua *art design* dokumennya selesai
   c. Saat konsep dan ide cerita telah selesai dibuat
   d. Saat akan melakukan sebuah desain karakter
   e. Ketika akan melakukan presentasi konsep gim yang akan dibuat

5. Carilah sebuah contoh gim yang sering dimainkan oleh teman-temanmu. Coba kamu buatkan dokumen desain artistik untuk gim tersebut dengan melengkapi komponen berikut ini:

Komponen ADD	Penjelasan
Nama Gim	
Tema Gim	
Desain Karakter	
Desain Properti	
Animasi dan Efek Visual	

Projek Akhir

1. Buatlah sebuah desain karakter gim dengan tema petualangan!
2. Buatlah sebuah desain latar belakang gim dengan tema teka teki!



Pengayaan

Untuk memperdalam pemahaman kamu tentang perancangan artistik gim ini, kamu bisa mencari contoh atau bentuk lain dari sebuah desain karakter, desain properti, desain latar belakang lainnya melalui internet, kemudian coba kalian membuatnya dengan konsep yang baru. Kegiatan yang kamu lakukan ini akan membantu kamu memperkaya pemahaman tentang perancangan artistik gim.

Buku dan video berikut ini bisa kamu gunakan untuk menambah wawasan dan mengembangkan keterampilan kamu.

Referensi Buku

Artificial Intelligence for games, Ian Millington And John Funge, Morgan Kaufmann Publishers is an imprint of Elsevier., 2009

Referensi Video:

1 Pindai Aku!

2 Pindai Aku!

1. **5 Steps To Gorgeous Game Art**
   https://www.youtube.com/watch?v=gTZT8GE5928&t=439s
2. **The Sandbox Game Maker: Creating an Art Design Document**
   https://www.youtube.com/watch?v=BERw4NolWh0

Refleksi

Kamu bisa memberikan tanda centang (✓) pada kotak yang kamu anggap sesuai!

Setelah kamu mempelajari bab ini, bagaimanakah penguasaan kamu terhadap materi-materi berikut ?

No	Materi	Tidak Menguasai	Menguasai	Sangat Menguasai
1	Desain Artistik • Desain karakter • Desain latar belakang • Desain properti			
2	Aset Audio			
3	Dokumen desain artistik			

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2024
Pengembangan Gim
untuk SMK/MAK Kelas XI-XII
Penulis: Rini Melati dan Muhammad Ardhan Fadlurrahman
ISBN: 978-634-00-0360-4

BAB 3

Desain Prototipe Gim



Tujuan Pembelajaran

Setelah mempelajari materi dalam bab ini, kamu dapat memahami desain prototipe gim (*game design prototype*) dan keseimbangan gim (*game design balancing)*

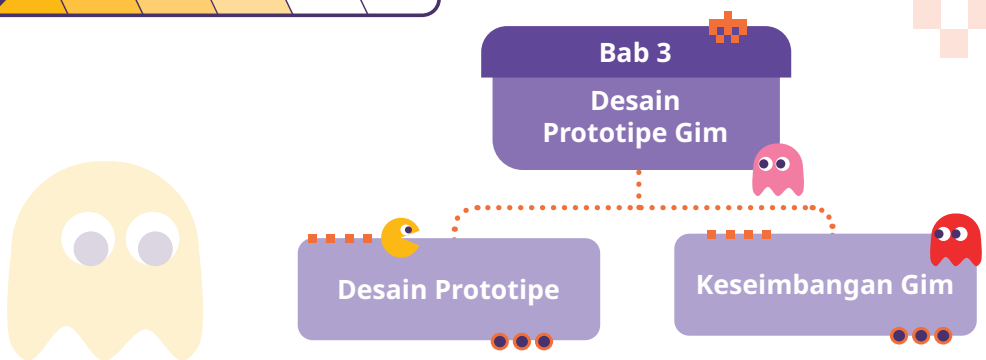
Kata Kunci

Game Design Prototype *Game Design Balancing*

Soft Skill

- **Komunikasi yang baik** dalam menjelaskan ide, memberikan umpan balik, dan bekerja dengan tim
- **Kreatif** dalam mengembangkan prototipe gim untuk menciptakan konsep-konsep baru yang inovatif
- **Pemecahan masalah** untuk menemukan solusi saat menghadapi tantangan dalam pengembangan prototipe gim
- **Kerja sama** dalam mengembangkan prototipe gim sehingga mampu bekerja dengan baik dalam tim dan berkontribusi secara positif

Peta Materi



Prototipe adalah istilah yang mungkin sudah pernah kamu dengar sebelumnya, tapi apakah kamu sudah mengetahui apa dan mengapa kita membutuhkannya dalam pengembangan gim? Coba kamu perhatikan gambar berikut ini:



Gambar 3.1 Prototipe Gim
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Gambar di atas anggaplah sebuah ide gim yang akan kamu buat, yaitu sebuah gim pertarungan atau gim *platformer*. Apakah kamu akan langsung membuat dan mengembangkan gim tersebut menjadi sebuah produk yang nyata? Saat kamu mempunyai ide atau konsep yang cemerlang untuk membuat sebuah gim, lalu kamu ingin membuat dan mengembangkan gim tersebut, tapi kamu terkendala dengan biaya pembuatan ataupun peralatan untuk membuat gim, maka kamu bisa menggunakan sebuah prototipe. Saat kamu ragu dan tidak yakin apakah idemu ini bisa diwujudkan atau diimplementasikan menjadi sebuah produk yang nyata, maka kamu juga bisa menggunakan sebuah prototipe. Dalam bab ini, kamu akan diberikan pengetahuan, wawasan, dan pemahaman tentang prototipe yang mencakup desain prototipe gim dan keseimbangan gim.

A. Desain Prototipe Gim (*Game Design Prototype*)

Pembuatan prototipe gim adalah langkah awal yang krusial dalam pengembangan gim. Dalam tahap ini, kamu akan menguji konsep dan ide *gameplay* dalam bentuk model sederhana atau versi awal permainan. Proses ini akan membantu kamu untuk mendapatkan wawasan tentang apa yang berhasil dan apa yang perlu diperbaiki sebelum melanjutkan ke produksi keseluruhan. Berikut adalah hal-hal yang perlu kamu ketahui tentang prototipe di antaranya tentang tujuan, proses, jenis, dan manfaat dari *game prototyping*.

1. Tujuan Pembuatan Prototipe Gim



a **Memastikan pemodelan gim dapat diimplementasikan menjadi Produk.**



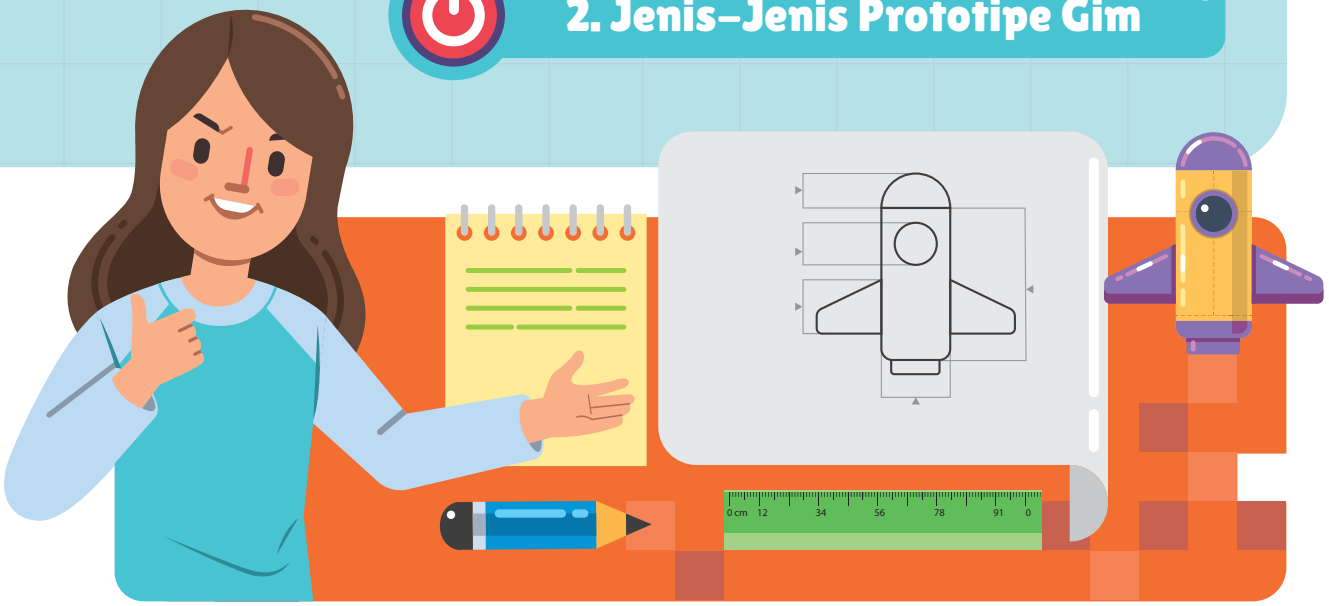
Pemodelan gim yang kamu buat tidak dapat kamu pastikan akan berhasil dan sukses. mereka yang sudah pernah menjadi desainer gim cukup lama pernah mengalami kegagalan dalam melakukan pemodelan gimnya. Kamu akan sangat kesulitan untuk mengetahui apakah pemodelan yang kamu buat ini akan berhasil atau tidak hanya dengan membaca rancangan yang hanya di dalam kertas saja. Membaca rancangan saja akan terdapat berbagai macam asumsi yang muncul dalam mengimplementasikan model gim yang akan dibuat. Maka perlu dibuatkan sebuah prototipe yang akan dapat memastikan apakah ide dalam pemodelan gim memungkinkan untuk dilakukan atau diimplementasikan? apakah ide tersebut seru untuk dimainkan? atau terdapat kelebihan atau kekurangan dari ide tersebut yang terlihat setelah sudah dilakukan implementasi singkat di dalam prototipe (Tynan Sylvester, 2017).

b **Sebagai bentuk proses perbaikan dalam pembaharuan agar disesuaikan dengan target pengguna gim.**



Dalam membuat atau merancang sebuah gim tentu kamu ingin gim yang kamu buat itu bisa disukai dan dinikmati sesuai target pengguna. Agar produk gim sesuai dengan target pengguna, maka kamu harus bisa mencatat semua *feedback* target pengguna yang memainkan gim kita. Semakin banyak orang yang memainkan gim kamu, dan semakin banyak kamu mendapatkan feedback maka gim kamu akan menjadi semakin baik. Setiap komen yang kamu dapat dari orang yang memainkan gim kamu baik positif atau negatif, akan membantu kamu melakukan pembaharuan gim agar sesuai dengan yang diharapkan calon pengguna (Tynan Sylvester, 2017).

2. Jenis–Jenis Prototipe Gim



a Prototipe Fisik

Prototipe jenis ini termasuk prototipe yang paling mudah untuk kamu gunakan karena hanya memerlukan beberapa bahan saja seperti kertas, karton, ataupun objek rumahan yang bisa dipakai untuk merepresentasikan simbolis dari komponen gim yang dibuat. tidak perlu skill pemrograman ataupun skill khusus untuk membuat prototipe jenis ini, kamu bisa lebih fokus untuk melakukan implementasi ide mekanik gim dan melakukan *testing* untuk memperbaiki ide mekanik gim tersebut.

Prototipe fisik berguna untuk membuat struktur dari gim yang akan dibuat, bagaimana setiap komponen berinteraksi dengan yang lain, dan memperbaiki formula permainan secara cepat dan melihat bagaimana keseluruhan sistem akan berjalan dan dinikmati oleh pemain tanpa harus melakukan implementasi teknis yang kompleks (Tracy Fullerton, 2004).

Prototipe ini akan membantu kamu menjelaskan konsep ide gim yang kamu buat pada anggota tim lainnya. Selain itu, ini akan membantu kamu untuk berkomunikasi dengan tim lainnya sehingga mereka dapat ikut berkontribusi dalam pengembangan ide gim yang sedang kamu buat tanpa harus memikirkan seberapa komplek implementasi sistem yang dibuat.

Fungsi Prototipe Fisik

1. Kecepatan dalam melakukan Pengembangan Prototipe

Kelebihan prototipe fisik adalah kemudahan dalam pengembangan prototipe, hanya dengan mengombinasikan kertas karton, *sticky notes*, dan elemen simple lain untuk membuat gim dengan waktu yang cepat. Walaupun sebagai desainer gim profesional yang mempunyai banyak pengalaman, memulai pengembangan gim prototipe digital juga membutuhkan waktu yang cukup lama dan *resource* yang banyak, maka prototipe fisik juga dapat membantu hal ini.

2. Kemudahan untuk melakukan kolaborasi

Karena tidak memerlukan skill teknis yang tinggi untuk melakukan pengembangan prototipe. Prototipe Fisik hanya memerlukan bahan-bahan sederhana, semua pihak dalam tim pengembangan bisa ikut berkontribusi di dalamnya. Hal ini membuka peluang untuk mendapatkan ide yang bagus dari seluruh tim yang kemungkinan tidak bisa berkontribusi secara efektif ketika membangun prototipe digital.

3. Kecepatan dalam melakukan Iterasi Pembaharuan

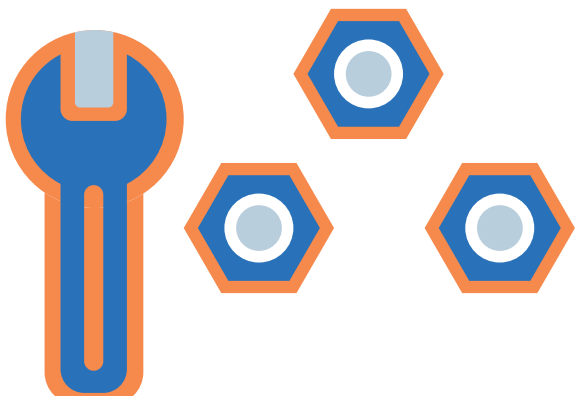
Melakukan perubahan dalam prototipe fisik sangat cepat dibandingkan dengan prototipe digital. Pada prototipe digital ketika kamu ingin melakukan perubahan dari sisi desain, maka kamu harus memastikan apakah komponen pemrogramannya atau komponen teknis lainnya tetap bisa menampung perubahan ide tersebut. Prototipe fisik bisa diubah bahkan ketika dimainkan, karena kemudahan dalam perubahan ini prototipe fisik sangat bermanfaat untuk melakukan *game brainstorming* awal di dalam pra-produksi suatu proyek gim.



Alat untuk Prototipe Fisik

Tabel 3.1 Alat yang digunakan untuk Propotipe Fisik

Alat	Fungsi Pada Prototipe Fisik
Kertas Ukuran Besar	Kertas bisa kamu bentuk seperti apapun yang kamu mau. Misalnya, kamu potong-potong atau lipat. Kamu bisa Menyusun potongan-potongan kertas untuk melakukan simulasi level desain dari gim yang kamu buat.
Sticky Notes	*Sticky Notes* sangat membantu kamu ketika membutuhkan alat untuk mengurutkan ide ataupun menyusun kumpulan informasi secara fisik.
Whiteboards	*Whiteboards* sangat berguna sebagai medium untuk *brainstorming* dalam proses pengembangan prototipe fisik. Kamu dapat menggunakannya untuk menyusun ide, menempelkan s*ticky notes*, maupun berbagai coretan ide hasil perubahan mekanik.
LEGO blocks	Jika prototipe fisik kamu membutuhkan suatu bangunan, simbolisme karakter, ataupun suatu objek yang merepresentasikan komponen di dalam gim, kamu bisa menggunakan *LEGO blocks* untuk membuat objek tersebut dengan cepat.
Buku Catatan	Buku catatan berguna untuk mencatat perubahan dan juga *feedback* pemain dari prototipe fisik. Catatan tersebut dapat kamu pakai ketika melanjutkan prototipe fisik ke dalam prototipe digital.



Waktu Terbaik Menggunakan Prototipe Fisik

Sebelum kamu memutuskan untuk membuat prototipe fisik, kamu harus mengetahui dahulu kapan waktu terbaik untuk menggunakan prototipe fisik:

1. Saat ingin memahami pergerakan pemain selama permainan

Prototipe fisik memungkinkan kamu untuk melihat langsung bagaimana pemain bergerak dalam gim. Dengan menggunakan prototipe ini, kamu bisa lebih cepat memahami cara pemain mengambil keputusan atau langkah saat bermain (Tylan Sylvester, 2017).

2. Saat ingin mengetahui keseimbangan dari sistem sederhana pada gim

Prototipe fisik memungkinkan kamu untuk merancangan dan menyeimbangkan sistem sederhana. Semua variable pada sistem yang sederhana dapat diseimbangkan dengan lebih cepat. Contohnya, sistem memiliki dua jenis senjata yaitu Shotgun dan Machine Gun. Shotgun memiliki serangan yang kuat tetapi tidak akurat, sedangkan Machine Gun yang memiliki serangan yang tidak terlalu kuat tetapi akurat. Dengan prototipe fisik, kamu bisa lebih cepat menguji keseimbangan antar variabel ini dan memastikan gim kamu terasa adil untuk dimainkan.

3. Saat ingin melihat grafis dari *User Interface*

Kamu bisa membuat mockup untuk *User Interface* menggunakan prototipe fisik. Dengan melakukan print UI yang diinginkan lalu ajak pemain untuk mencoba interaksi fisik dengan desain tersebut. Contohnya, mereka "menekan" gambar tombol secara langsung. Hal ini dapat membantu kamu memoderasi interaksi UI dan memperbaiki desain sebelum masuk ke versi digital.

4. Saat ingin mencoba ide yang tidak biasa

Mudahnya melakukan perubahan dalam prototipe fisik memungkinan kamu untuk mencoba aturan yang sangat berbeda. Jika ternyata ide tersebut tidak berjalan dengan baik, kamu bisa lebih mudah untuk mengatur ulang atau membuat perubahan tanpa dibandingkan dengan prototipe digital.

Waktu Kurang Tepat Menggunakan Prototipe Fisik

Ternyata tidak semua ide pemodelan gim cocok menggunakan prototipe fisik. Berikut beberapa hal yang tidak cocok dilakukan dengan prototipe fisik:

1. Saat mencatat dan memproses informasi yang kompleks

Ketika suatu sistem mulai cukup kompleks dan membutuhkan untuk menyimpan banyak jenis informasi prototipe fisik tidak lagi disarankan. Contohnya, bagaimana banyak variabel berpengaruh dengan yang lain sehingga setiap langkah mengharuskan kita menghitung banyak angka secara manual. Sistem yang kompelks akan lebih mudah ketika dilakukan di dalam prototipe digital.

2. Saat melakukan *testing* teknis input fisik ataupun jenis Input lain

Walaupun dalam prototipe fisik kita dapat melakukan *mockup* UI dan melakukan simulasi, hal ini tidak bisa dilakukan jika kita harus mengecek apakah input *gamepad* dari pemain memiliki respon yang cepat dari aksi yang dilakukan. Testing teknis input hanya bisa dilakukan di prototipe digital.

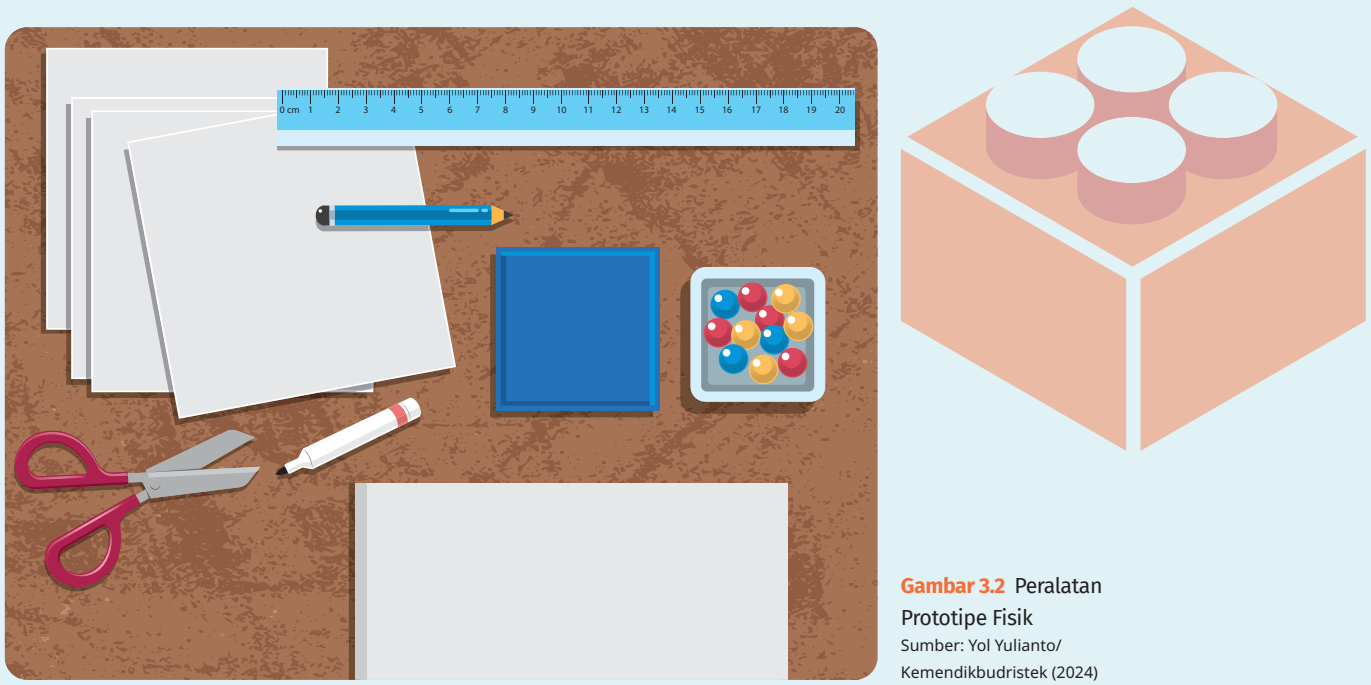
Contoh Prototipe Fisik

Di sini kamu akan diberikan contoh membuat gim dengan menggunakan prototipe fisik. Contoh ini akan membantu kamu mendapatkan gambaran tentang bagaimana cara pembuatan atau penggunaan prototipe fisik.

Permainan ini adalah Kapal perang, mungkin ada di antara kamu yang belum mengenal permainan ini. Permainan ini merupakan permainan papan untuk dua pemain, tujuan permainan ini adalah menjadi pemain pertama yang menenggelamkan armada lawan.

Yuk, kita mulai langkah-langkah! **Pertama** yang kamu lakukan adalah melakukan identifikasi elemen utama yang ada dalam permainan. Dalam gim Kapal Perang, elemen-elemen utama meliputi kapal perang sebagai objek representasi pemain, kotak koordinasi untuk menaruh kapal perang tersebut sebagai ruang permainan, serta perlatan untuk "menembak" misil ke ruang permainan musuh dan mengenai kapal perang musuh.

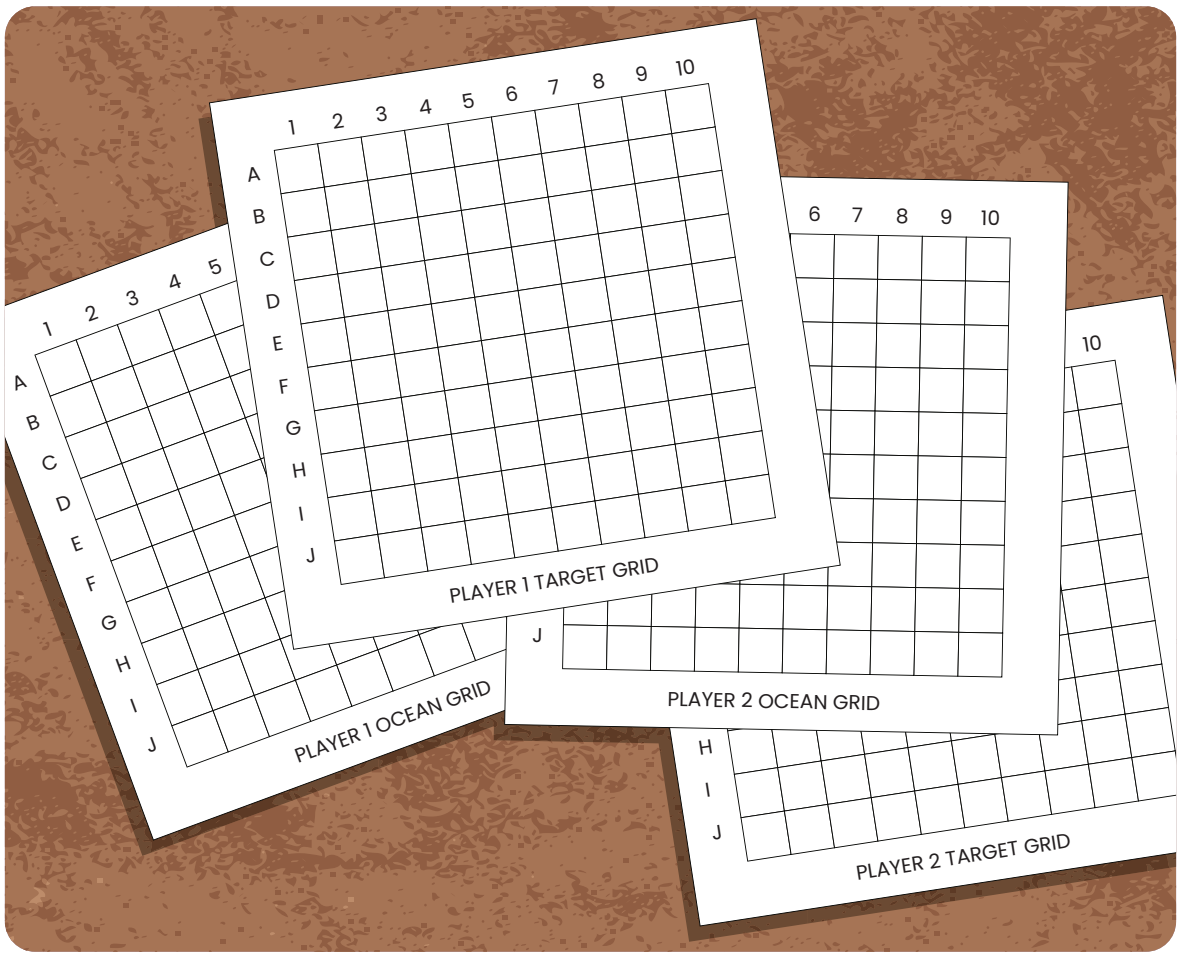
Kedua, setelah elemen-elemen utama teridentifikasi, siapkan peralatan. Kamu bisa menggunakan kertas, karton, penghapus, koin, pena, pin, penggaris, dan pensil sesuai dengan kebutuhanmu.



Gambar 3.2 Peralatan Prototipe Fisik
Sumber: Yol Yulianto/ Kemendikbudristek (2024)

Ketiga, buatlah bagian-bagian permainan. Dalam contoh permainan ini ikuti langkah-langkah berikut:

1. Ambil empat lembar kertas dan gambar kotak berukuran 10 × 10 pada setiap kotak.
2. Beri label baris pada setiap kotak dengan huruf A hingga J.
3. Beri label kolom pada setiap kotak 1 hingga 10.
4. Letakkan judul pada empat kotak: Kotak Laut Pemain 1, Kotak Sasaran Pemain 1, Kotak Laut Pemain 2, dan Kotak Sasaran Pemain 2.



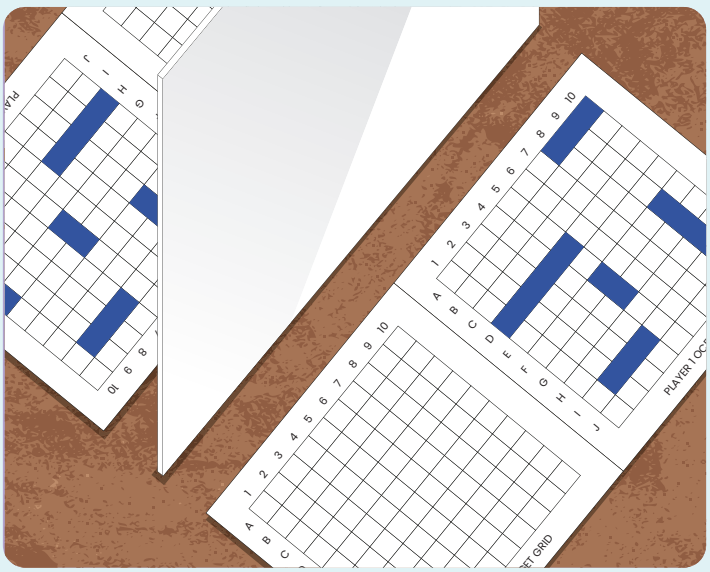
Gambar 3.3 Empat Lembar Kertas dan Gambar Kotak Berukuran 10 × 10 Pada Setiap Kotak
Sumber: Yol Yulianto/Kemendikbudristek (2024)

Keempat, cari dua pemain dan berikan masing-masing kotak laut, kotak sasaran, dan beberapa spidol berwarna, seperti manik-manik kecil. Pemain harus melindungi kotak mereka dari pandangan lawan. Setiap pemain mendistribusikan lima kapal berikut dengan menggambar di Kotak Lautnya.

Angka dalam tanda kurung adalah ukuran kapal pada petak:

- Kapal induk (1 × 5 sel)
- Kapal perang (1 × 4 sel)
- Kapal perusak (1 × 3 sel)
- Kapal selam (1 × 3 sel)
- Kapal patroli (1 × 2 sel)

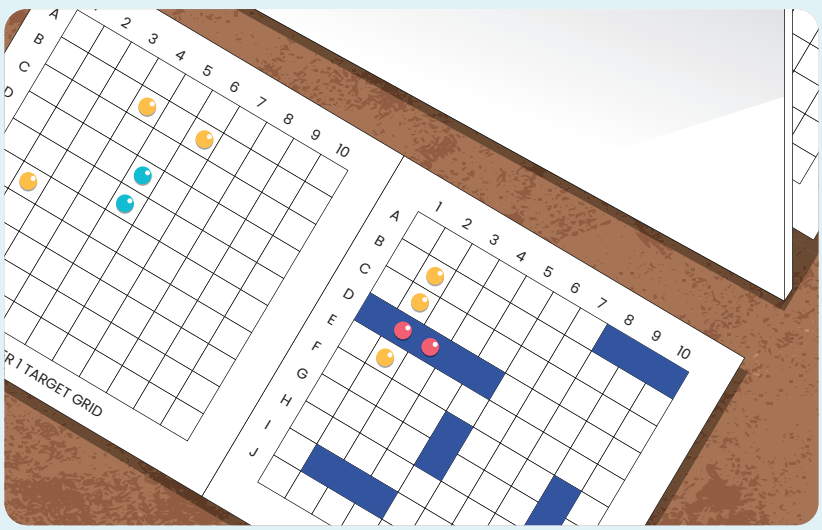
Semua segmen kapal harus digambar pada petak permainan. Kapal tidak boleh ditempatkan secara diagonal. Gambar berikut menunjukkan contoh kapal yang ditempatkan pada petak.



Gambar 3.4 Contoh Permainan Kapal Pada Lembar Kertas Kotak Kotak
Sumber: Yol Yulianto/Kemendikbudristek (2024)

Sekarang setelah prototipe dirakit, saatnya bermain. Pada giliran pemain, pemain tersebut menyebutkan koordinat petak, seperti "B5". Jika lawan memiliki kapal di sel tersebut, maka ia menjawab "kena sasaran". Jika tidak, ia menjawab "kena sasaran". Ketika semua segmen kapal telah terkena sasaran, lawan berkata, "Kau menenggelamkan kapal perangku!". Cukup sederhana kan? Pemain melacak hit dan miss pada grid target mereka. Jika B5 adalah hit, pemain menandai hit pada grid targetnya.

Dalam prototipe ini, kamu bisa menggunakan manik-manik dengan warna berbeda untuk menandai perbedaan antara hit dan miss. Pemain bergiliran menyebutkan koordinat seperti ini hingga satu pemain menenggelamkan kelima kapal lawan. Gambar berikut ini menunjukkan contoh tampilan grid selama permainan.



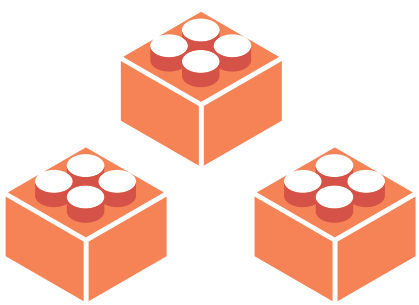
Gambar 3.5 Manik-Manik dengan Warna Berbeda untuk Menandai Perbedaan Antara Hit dan Miss.
Sumber: Yol Yulianto/Kemendikbudristek (2024)

Cobalah kamu mainkan permainan di atas. Kemudian pikirkan, Bagaimana fungsi prototipe. Apakah prototipe fisik secara akurat menggambarkan mekanisme permainan? Meskipun gambarnya kasar dan aturannya tidak jelas, apakah memberikan pengalaman yang cukup bagi pengembang gim untuk memahami permainan dan memberikan umpan balik? Jika demikian, maka prototipe yang kamu buat tersebut berhasil.

Seperti yang kamu lihat, membuat prototipe permainan yang dapat dimainkan tidak memerlukan keterampilan pemrograman atau keterampilan seni. Keuntungan dari pembuatan prototipe fisik adalah saat kamu menyusun permainan, kamu akan memperoleh sensasi yang bisa kamu rasakan tentang bagaimana mekanisme permainan saling terkait. Aturan abstrak akan terasa menjadi konkret. kamu dapat melihat pada kotak-kotak dan bertanya pada diri sendiri, "Bagaimana jika saya membuatnya lebih besar? Bagaimana hal tersebut akan memengaruhi permainan?". Kamu dapat memperbesar kotak-dengan menggambar kotak yang lebih besar. Kemudian, kamu dapat memainkan ulang permainan dan melihat apakah pengalamannya lebih baik atau lebih buruk.

Aktivitas 3.1 Praktik Mandiri

Dalam aktivitas ini kamu diminta untuk melakukan modifikasi prototipe yang kamu buat. Gunakan prototipe Kapal Perang kemudian modifikasi tiga bagian dari permainan tersebut. Kamu bisa mengubah kotak-kotak, kapal, atau objek lain yang ada dalam permainan. Kamu juga bisa mengubah prosedur permainan yang ingin kamu modifikasi. Buatlah perubahan tersebut menjadi lebih kreatif. Setelah perubahan dilakukan mainkanlah permainan tersebut dengan temanmu dan jelaskanlah bagaimana perubahan tersebut dapat mempengaruhi permainannya.





b Prototipe Digital

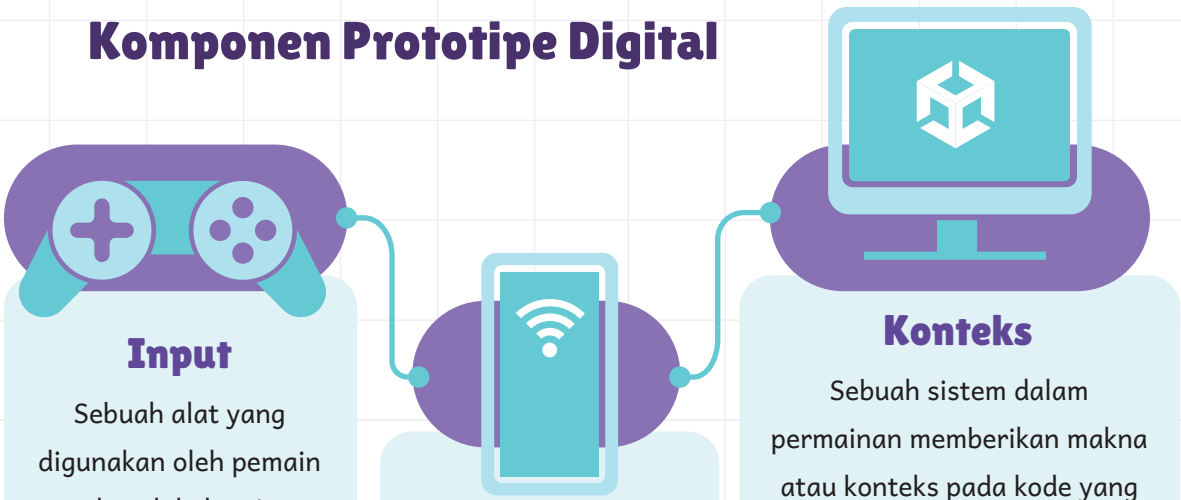


Sekarang setelah kamu memiliki beberapa pengalaman dalam membuat prototipe fisik, kamu akan merasakan prototipe fisik memiliki keterbatasan. Jika gim yang kamu buat telah selesai dan akan dirilis ke dalam platform digital, maka kamu perlu membuat prototipe digital dari konsep yang telah kamu buat sebelumnya. Bukan berarti kamu memulainya dari awal ya.

Prototipe fisik membantu kamu memformalkan dan menguji dasar dari mekanisme permainan yang kamu buat. Prototipe digital akan memperluas pekerjaan desain tersebut ke dalam bentuk digital dan memungkinkan kamu untuk menguji esensi permainan dalam format yang diinginkan. Pemahaman tentang sistem formal yang kamu diperoleh dari pengalaman pembuatan prototipe fisik akan menghidupkan desain permainan digital.

Prototipe digital merupakan prototipe yang telah diimplementasikan di dalam suatu sistem digital dan dimainkan dengan alat digital seperti komputer, perangkat genggam, maupun konsol. Prototipe digital adalah bentuk terkecil dari gim yang sedang kamu kembangkan. Berikut ini ada beberapa komponen yang diperlukan dalam prototipe digital (Steve Swink, 2008).

Komponen Prototipe Digital



Input

Sebuah alat yang digunakan oleh pemain untuk melakukan input dalam berinteraksi di dalam gim. alat ini bisa dalam bentuk *Gamepad*, *Keyboard*, ataupun *Controller* spesifik untuk memainkan suatu gim.

Respon

Bagaimana sebuah sistem memproses input dan melakukan respon input dari pemain secara *real time*.

Konteks

Sebuah sistem dalam permainan memberikan makna atau konteks pada kode yang digunakan saat implementasi. Sistem ini membantu pemain memahami bagaimana mekanisme permainan bekerja dan dapat mengubah cara mereka merasakan atau memandang permainan tersebut.

1 Jenis Prototipe Digital

Hal penting yang perlu kamu ingat saat membuat prototipe digital adalah mempertimbangkan alasan kamu membuat prototipe tersebut. Apakah kamu mencoba menjawab pertanyaan desain gim atau teknis? Apakah kamu mencoba membangun alur produksi yang efektif? Atau apakah kamu mencoba mengkomunikasikan visi kamu kepada tim atau penerbit? Bagian berikut akan membantu kamu membuat prototipe digital untuk situasi yang sangat berbeda. Terdapat beberapa jenis prototipe digital berdasarkan tujuan dari prototipe digital tersebut, seperti:

Prototipe Digital: Mekanik Gim

Sepertinya yang sudah dijelaskan, prototipe fisik membantumu melakukan definisi dan iterasi dari mekanik gim, tetapi terkadang tidak semua mekanik gim dapat dibuat menjadi prototipe fisik dimana terkadang kamu harus membuat prototipe digital untuk melakukan definisi dan iterasi dari Mekanik Gim (Tracy Fullerton, 2004).

Contohnya, gim *First Person Shooter* dimana kamu harus memiliki input dan respons yang saling berkaitan dan sangat presisi yang sulit dimodelkan dengan Prototipe Fisik, ataupun mekanik gim yang cukup kompleks sehingga sulit untuk dimodelkan di dalam Prototipe Fisik, seperti yang sudah dijelaskan oleh Tylan Sylvester dalam bukunya yaitu Designing Games (Tylan Sylvester, 2017).

Prototipe Digital: Desain Estetika

Desain estetika merupakan salah satu implementasi dari sebuah *context* di dalam gim. Desain estetika dibagi dibagi menjadi:

Storyboard

Storyboard adalah sebuah gabungan dari sketsa gambaran narasi yang menunjukan konsep kasar dari sebuah urutan cerita. Pada umumnya digunakan dalam pengembangan film tetapi juga bisa digunakan di dalam pengembangan gim.

Concept Art

Concept art adalah sebuah kumpulan lukisan atau sketsa yang menjelaskan tentang informasi visual sebuah karakter maupun lingkungan di dalam gim. Informasi yang diperoleh dari lukisan maupun sketsa adalah gambaran kasar untuk palet warna, serta *style* dari komponen visual gim tersebut.

Animasi *Mock-Up*

Animasi *Mock-Up* merupakan sebuah jenis animasi yang belum final tetapi cukup merepresentasikan secara kasar animasi yang akan digunakan sepanjang permainan. Animasi *Mock-Up* bisa dalam bentuk animasi karakter, animasi lingkungan, maupun animasi visual efek yang akan membantu meningkatkan komponen imersif di dalam sebuah gim.

Prototipe dari *Interface*

Prototipe dari komponen *interface* atau UI dapat membantu menceritakan konsep kasar sebuah gim. Seperti bagaimana alur informasi pada sebuah gim, informasi terkait jumlah koin, urutan tombol atau layout UI. Komponen *Interface* ini belum final dan pada umumnya hanya berbentuk bidang sederhana seperti lingkaran maupun persegi.

Prototipe dari Audio

Prototipe dari audio dapat membantu merepresentasikan secara kasar komponen audio yang akan dipakai di dalam sebuah gim. Bisa dalam bentuk *background* musik maupun *sound effect* yang mereprentasikan secara sederhana bagaimana komponen audio dapat menambah imersi dari gim yang dibuat. Audio yang dipakai belum dalam tahap final dan terkadang memakai audio dari produk lain karena tidak akan dijual secara komersil, hanya cukup untuk merepresentasikan bayangan produk akhir.

Dalam pemodelan gim, penggunaan prototipe digital adalah hal yang umum dilakukan untuk menggambarkan estetika gim. Prototipe ini digunakan untuk menunjukkan gaya seni (*art style*), visi gameplay, serta rancangan antarmuka pengguna (*user interface*) dari gim yang sedang dikembangkan. Terdapat beberapa pertanyaan yang harus dilemparkan oleh tim estetika saat melakukan prototipe digital untuk desa estetika (Jarek Majewski, 2022):

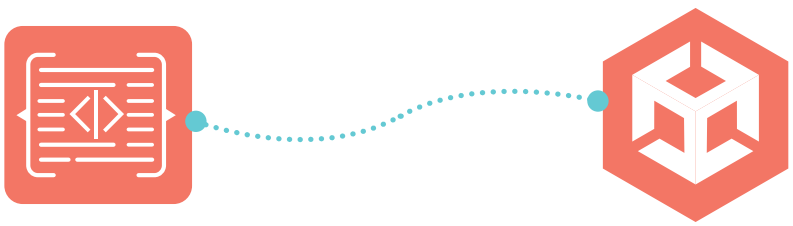
1. Bagaimana perspektif kamera secara umum di dalam gim yang sedang dibuat?
2. Bagaimana ukuran karakter pemain ketika dibandingkan dengan lingkungan dia berada, dan dibandingkan dengan ukuran layar dari target platform gim yang sedang dibuat?
3. Bagaimana art style cocok dengan target platform, tema, serta audiens gim yang sedang dibuat?
4. Bagaimana ukuran, posisi dan visibility dari elemen User Interface?

Prototipe Digital: Teknologi

Prototipe teknologi adalah jenis prototipe yang dirancang untuk mengembangkan teknologi utama yang mendukung pembuatan gim. Teknologi ini mencakup, misalnya, sistem fisika, kecerdasan buatan (AI), atau teknologi lain yang diperlukan untuk mengimplementasikan ide gim. Fokus dari prototipe ini adalah menguji alat bantu (*tools*) atau alur kerja (*workflow*) yang didukung oleh alat tersebut, sehingga konten gim dapat dimasukkan dan diintegrasikan dengan lebih mudah.

2 Alat yang digunakan untuk Prototipe Digital

Jika kamu ingin membuat prototipe digital, ada beberapa alat penting yang kamu butuhkan:



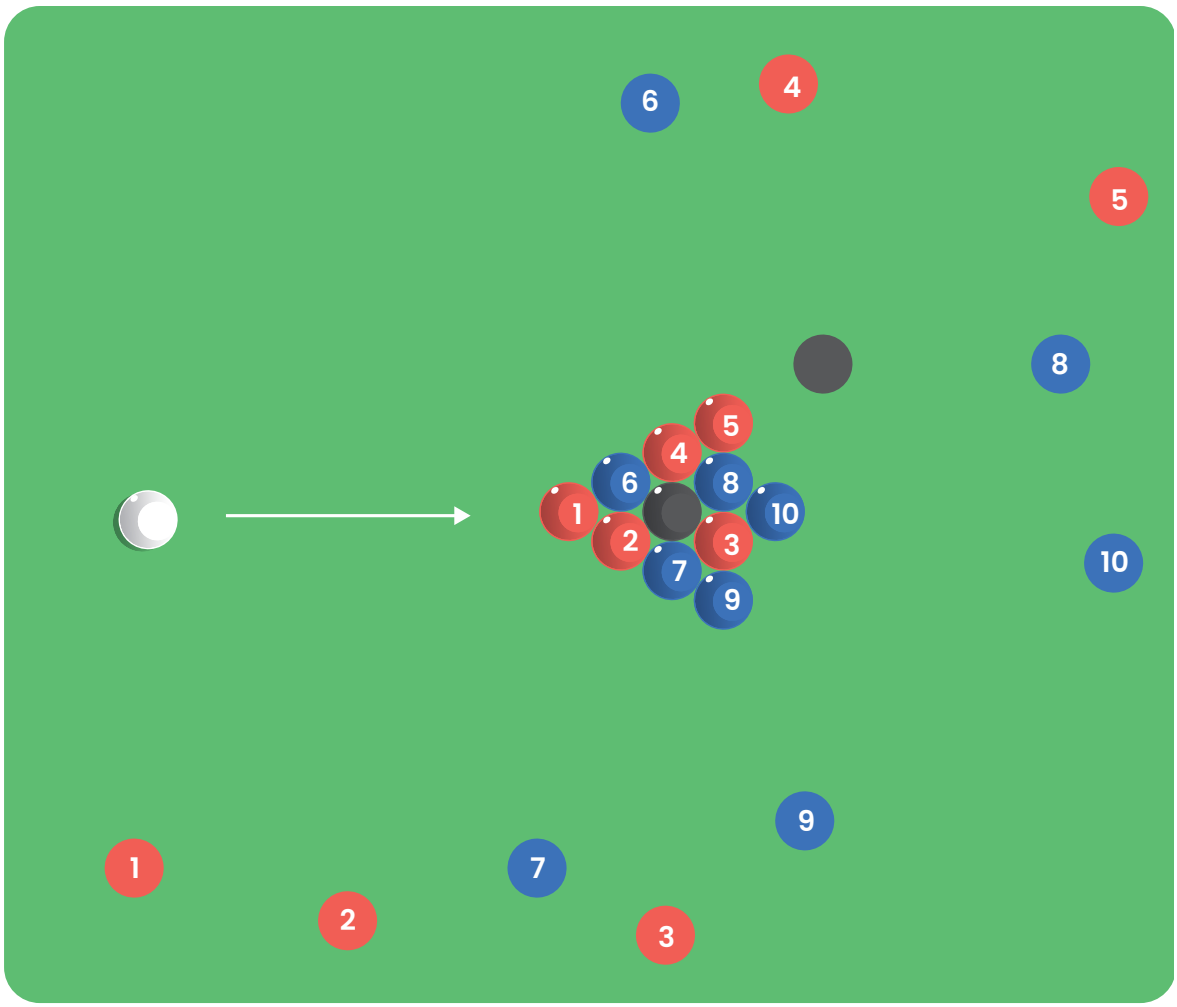
Bahasa Pemrograman (*Programming Languages*)

Bahasa pemrograman merupakan dasar dalam mengembangkan suatu sistem digital. Seorang desainer gim perlu memahami konsep pemrograman agar mampu membuat pemodelan gim yang dapat diimplementasikan secara teknis. Selain itu, pemahaman ini juga mempermudah desainer dalam mengomunikasikan kebutuhan teknis kepada tim pengembang.

Game Engines

Untuk mempercepat proses pembuatan prototipe, penggunaan *Game Engine* sangat disarankan. Dengan *Game Engine*, kamu tidak perlu membangun semua sistem dari awal, karena fitur-fitur dasar seperti pengaturan suara dan sistem input sudah tersedia. Bahkan, beberapa desainer gim sering mengambil keputusan desain mekanik berdasarkan fitur yang dimiliki oleh *Game Engine* tersebut. Contoh *Game Engine* yang umum digunakan adalah Unity dan Unreal Engine.

Sebagai contoh penerapan, bayangkan sebuah permainan bola biliar. Dalam permainan ini, kamu dapat menambahkan fitur yang memungkinkan pemain untuk memprediksi posisi akhir bola setelah dipukul. Dengan demikian, pemain seolah dapat melihat hasil dari setiap tindakan mereka sebelum benar-benar melakukannya.



Gambar 3.6 Prototipe Digital Permainan Bola Biliard
Sumber: Yol Yulianto/Kemendikbudristek (2024)

Melalui penggunaan komponen-komponen ini, Prototipe Digital yang kamu buat akan menjadi lebih efisien dan memberikan gambaran yang jelas mengenai konsep permainan yang dirancang.

3 Waktu Terbaik Menggunakan Prototipe Digital

Penjelasan di bawah ini bisa kamu jadikan pertimbangan kapan waktu yang tepat untuk membuat prototipe digital:

a. Saat memiliki ide mekanik desain, sudah ditesting, dan diiterasikan di prototipe fisik.

Tracy Fullerton (2004) menyebutkan bahwa di dalam proses pembuatan prototipe fisik, tidak perlu terlalu memikirkan kualitas teknis. Hal ini karena prototipe digital berfokus untuk melakukan pengembangan sistem secara cepat namun kasar, sehingga kamu bisa melihat bagaimana gim bekerja dari sisi pemodelan gim. Prototipe digital ini nantinya akan membantu memperbaiki kualitas teknis setelah kamu yakin prototipe fisik sudah disempurnakan. Setelah itu, kamu bisa fokus untuk melakukan perbaikan teknis tanpa perlu memikirkan lagi bagaimana game bekerja.

b. Saat memiliki skill yang memadai ataupun rekan tim yang memadai dalam mengembangkan prototipe digital.

Chris Plummer, salah satu *Co-Head of Mobile* di Activision dalam buku *Game Design Workshop* mengatakan bahwa salah satu kekurangan dari prototipe digital adalah ketika melakukan perubahan dari sisi game design, prototipe digital lebih sulit karena harus memastikan tidak terdapat konflik dari sisi teknis pada versi sebelumnya dan juga harus melakukan estimasi berapa banyak dan juga tingkat perubahan dari komponen yang berubah dari sisi teknis ketika perubahan dilakukan. Dibandingkan dengan Prototipe fisik yang tidak terlalu memikirkan konflik tersebut, sehingga prototipe digital harus memiliki skill yang lebih tinggi atau memadai dibandingkan prototipe fisik (Tracy Fullerton, 2004).

c. Saat ingin melakukan *testing* komponen pembantu gim.

Prototipe digital biasanya difokuskan satu atau dua aspek berbeda di dalam sistem sehingga setiap komponen dapat dilakukan validasi dalam iterasi selanjutnya. (Tracy Fullerton, 2004)

4 Waktu Kurang Tepat Menggunakan Prototipe Digital

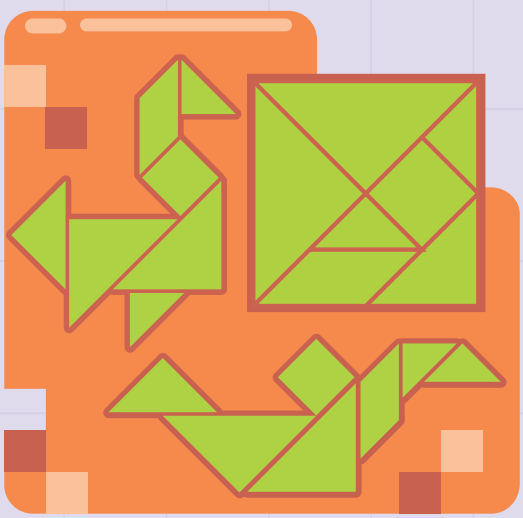
Sama halnya dengan prototipe fisik, tidak semua ide pemodelan gim cocok menggunakan prototipe digital. Berikut beberapa hal yang tidak cocok dilakukan dengan prototipe digital:

a. **Saat tidak memiliki skill teknis, *resource* pengembangan atau rekan tim yang memadai untuk melakukan implementasi.**

b. **Saat belum melakukan testing ide mekanik dari sebuah gim.**

   Jika implementasi tidak tepat maka ide mekanik tidak tersampaikan dengan baik. Dengan hal tersebut akan terjadi kesulitan ketika memproses suatu insight dan melakukan iterasi pembaharuan dari ide mekanik gim. Semakin banyak perubahan, maka *resource* yang diperlukan dalam melakukan prototipe digital menjadi sangat jauh dibandingkan dengan prototipe fisik (Tracy Fullerton, 2004). Jadi sebaiknya tidak menggunakan prototipe digital ketika ide gim kita belum pasti dan masih sangat berubah-ubah setiap iterasi.

3. Perbedaan Prototipe Fisik dan Digital

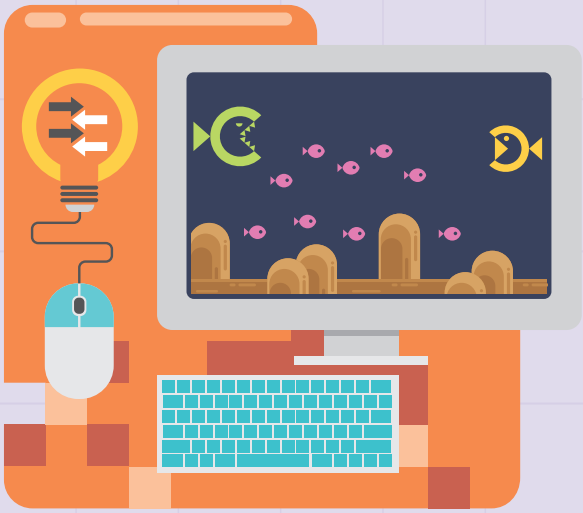


Prototipe Fisik

Fokus untuk melakukan **iterasi pemodelan gim dan melakukan testing** terkait pemodelan gim tanpa harus memikirkan teknologi dan implementasi aspek lain di dalam sebuah gim. Prototipe fisik membuat kamu lebih fokus untuk mendefinisikan dan melakukan iterasi dari *gameplay* dibandingkan teknologi (Tracy Fullerton, 2004).

Prototipe Digital

Prototipe digital merupakan pengembangan lanjutan dari prototipe fisik. Dalam prototipe ini, terdapat dua hal utama yang dikembangkan yaitu sistem input dan output sistem digital. Sistem *input* yang dikembangkan untuk prototipe digital seperti *keyboard*, *mice,* layar sentuh, ataupun jenis input lain yang mendukung target perangkat digital dari gim yang sedang dikembangkan (Tracy Fullerton, 2004). Selanjutnya, output dari sistem digital seperti respon dan konteks seperti yang sudah dijelaskan di atas (Steve Winks, 2008).



Aktivitas 3.2 Kerja Tim

Dalam aktivitas ini, kalian akan melakukan kerja tim yang terdiri dari 3 orang anggota. Tugas kalian adalah mencari tahu contoh prototipe digital dalam sebuah gim yang sudah dijelaskan di atas! Kemudian kalian lengkapi tabel berikut dan mempresentasikannya di depan kelas untuk berbagi informasi!

Jenis Prototipe Digital	Contoh Prototipe Gim	Nama Gim
Mekanik Gim		
Desain Estetika		
Teknologi		

*sertakan tangkapan layar pada contoh prototipe gim.

B. Keseimbangan Gim

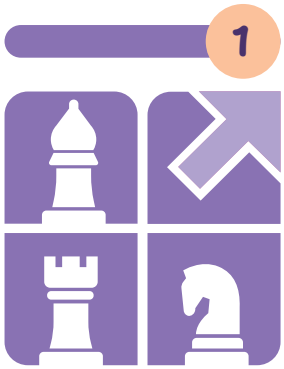
Jika permainanmu terasa mustahil untuk dimainkan, dipelajari, dan dikuasai, pemain akan berhenti bermain! Di sisi lain, jika permainanmu terlalu mudah, maka pemain akan bosan dan akan berhenti bermain! Lalu apa yang harus kamu lakukan? Kamu harus melakukan keseimbangan gim. Keseimbangan gim atau *game balance* adalah proses mendesain gim yang berkelanjutan tentang menyesuaikan komposisi mekanika, hadiah, dan tantangan gim. Tujuannya adalah menciptakan permainan yang adil sekaligus memberikan pengalaman bermain yang berarti dengan tingkat kesulitan yang tepat. Dengan menyeimbangkan gim secara konsisten, kamu bisa menjaga pemain agar tetap tertarik, dan juga dapat memperpanjang waktu mereka bermain gim.

Konsep keseimbangan dalam gim akan sangat bergantung pada jenis permainan yang kamu buat. Apakah gim itu berbicara tentang pemain melawan pemain lain (*Player versus Player* atau PvP), atau gim yang berfokus pada pemain menghadapi tantangan dari dunia gim (*Player versus Environment* atau PvE)?

1. Karakteristik Gim Seimbang

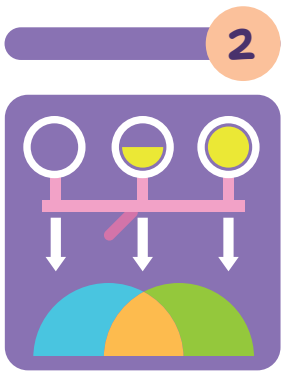


Kamu harus mampu memperhatikan keseimbangan antara PvP dan PvE ini dengan baik. Berikut adalah beberapa karakteristik gim yang seimbang, baik untuk PvP maupun PvE, seperti yang dijelaskan oleh Ernest Adams (2014):



Gim menyediakan pilihan yang bermakna (*meaningful choice*).

Artinya, jika gim memungkinkan pemain memilih berbagai strategi untuk menghadapi tantangan, tidak boleh ada strategi yang secara konsisten lebih efektif dibandingkan strategi lain. Jika pemain merasa tidak ada gunanya mencoba strategi berbeda karena satu strategi selalu dominan, itu tanda bahwa gim tersebut tidak seimbang.



Peran peluang tidaklah begitu besar sehingga keterampilan pemain menjadi tidak relevan.

Bukan berarti pemain tidak boleh mengalami hal buruk, tetapi dalam jangan panjang baik di satu permainan yang panjang maupun dalam beberapa permainan pendek, pemain yang lebih terampil seharusnya tetap lebih sukses daripada pemain yang kurang terampil.

a Karakteristik Gim PvP yang Seimbang

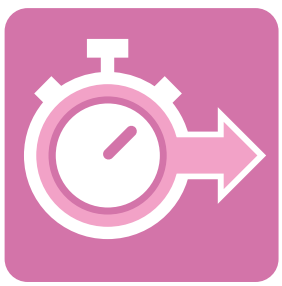
Untuk menciptakan gim PvP yang seimbang, ada beberapa karakteristik penting yang perlu kamu perhatikan:



1 Pemain Merasa Permainan Adil

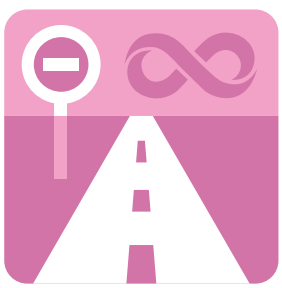
Setiap pemain mempunyai cara bermainnya yang berbeda beda karena pola pikir yang mungkin tidak sama dengan perancang gim. Tugas kita adalah memastikan Sebagian besar dari pemain tidak merasa dirugikan pada komponen tertentu ataupun tidak terdapat satu strategi yang akan selalu menang ketika dipakai dalam permainan. Permainan yang adil memastikan permainan yang menyenangkan, membuat pemain ingin tetap bermain, serta menambah rasa ingin tahu dari pemain untuk melakukan eksplorasi tipe pemain yang lain yang bisa ditemukan di permainan dan melakukan adaptasi strategi dari tipe pemain tersebut.

Keseimbangan dalam permainan sangat penting untuk memastikan pengalaman yang adil dan menyenangkan bagi pemain. Permainan yang adil tidak hanya membuat pemain ingin terus bermain, tetapi juga mendorong eksplorasi strategi serta adaptasi terhadap berbagai tipe pemain lain. Oleh karena itu, tugas utama dalam perancangan gim adalah memastikan bahwa tidak ada satu strategi yang selalu menang dan tidak ada komponen tertentu yang merugikan sebagian besar pemain. Perbedaan cara bermain setiap pemain, yang dipengaruhi oleh pola pikir mereka, sering kali tidak sejalan dengan pemikiran perancang gim, sehingga keseimbangan menjadi tantangan utama dalam desain permainan.



2 Kesempatan untuk Mengerjar Ketertinggalan

Jika seorang pemain tertinggal di awal permainan, mereka harus tetap diberi kesempatan yang adil untuk mengejar ketertinggalan sebelum permainan berakhir. Misalnya, Jika seorang pemain tertinggal dalam 10 menit pertama dari permainan selama 2 jam dan aturan permainan tidak memberinya kesempatan untuk mengejar ketertinggalan, sebagian besar pemain akan menganggap permainan itu tidak adil. Demikian pula, permainan yang dimaksudkan dirancang untuk berlangsung selama 2 jam tetapi seseorang selalu menang dalam 15 menit juga tidak memberi pemain lain waktu untuk mengejar ketertinggalan atau bahkan untuk menguji keterampilan mereka. Ketidakseimbangan ini sering kali menunjukkan masalah dengan umpan balik positif.



3 Permainan Tidak Berakhir dengan Jalan Buntu

Permainan yang seimbang jarang berakhir dengan jalan buntu, terutama antara pemain yang memiliki kemampuan yang setara. Jalan buntu bisa sangat mengecewakan pemain karena usaha mereka tidak menghasilkan pemenang. Jika jalan buntu sering terjadi di antara pemain yang kemampuannya tidak setara, ini melanggar prinsip bahwa keterampilan pemain harus mempengaruhi hasil lebih dari faktor lainnya. Contohnya, dalam permainan catur, meskipun catur adalah permainan yang seimbang, jalan buntu jarang terjadi di antara pemain dengan kemampuan yang tidak setara.

b Karakteristik Gim PvE yang Seimbang

Untuk menciptakan gim PvE yang seimbang, ada beberapa karakteristik penting yang perlu kamu perhatikan:

1 Pemain Merasa Permainan Adil

Dalam permainan PvE, persepsi pemain tentang keadilan melibatkan sejumlah faktor dan menjadi rumit karena tidak adanya lawan manusia atau simulasi. Contohnya misal di dalam satu game terdapat karakter yang kuat dalam menyerang dan satu karakter lain kuat dalam bertahan, kita sebagai perancang gim harus memastikan bagaimana karakter yang kuat bertahan tetap dapat melakukan serangan untuk mengalahkan musuh , dan juga karakter yang kuat menyerang dapat tetap bertahan agar tidak mudah dikalahkan musuh, misalkan dengan atribut upgrade tertentu sehingga permainan tetap adil walaupun pemain memilih kedua pemain tersebut.

2 Tingkat Kesulitan Permainan Harus Konsisten

Tingkat kesulitan yang dirasakan dari tantangan permainan, tetap dalam kisaran yang wajar agar tidak mengejutkan pemain dengan lompatan atau penurunan yang tiba-tiba. Tingkat kesulitan yang dirasakan mungkin rendah atau tinggi tetapi tidak boleh berubah secara tiba-tiba, terutama dalam satu level permainan.

2. Teknik Menyeimbangkan Gim

Untuk melakukukan kesimbangan gim, ada beberapa teknik yang bisa kamu lakukan, diantaranya:

a Menghindari Strategi Dominan



Strategi dalam permainan adalah rencana atau pendekatan yang digunakan pemain untuk mencapai kemenangan. Setiap pemain biasanya memiliki strategi yang diyakini dapat membawa kemenangan, berdasarkan gaya atau preferensi mereka. Misalnya, satu pemain mungkin lebih suka pendekatan agresif, sementara pemain lain lebih mengandalkan pertahanan yang kuat. Masing-masing percaya bahwa strategi mereka akan memberi mereka peluang terbaik untuk menang.

Namun, ada istilah yang disebut strategi dominan, yang berasal dari teori permainan formal. Strategi dominan secara konsisten, menghasilkan hasil terbaik untuk pemain terlepas dari tindakan lawannya. Meskipun terdengar ideal, strategi dominan justru tidak diinginkan dalam desain permainan. Kenapa? Karena begitu pemain menemukan strategi dominan, mereka tidak akan pernah merasa perlu menggunakan strategi lain. Pilihan lain menjadi tidak berguna, yang tentunya mengurangi keseruan dan kedalaman permainan.

Yang lebih buruk lagi adalah jika strategi dominan hanya dipakai oleh satu pemain, sementara pemain lain tidak menggunakannya, biasanya dalam permainan asimetris. Dalam kasus ini, strategi dominan tidak hanya meniadakan strategi lain, tetapi juga menciptakan ketidakseimbangan dan ketidakadilan dalam permainan.

Oleh karena itu, ketika merancang mekanisme permainan, sangat penting untuk menghindari strategi dominan. Hal ini adalah bagian penting dari keseimbangan permainan. Terkadang, satu pilihan tunggal bisa menjadi strategi dominan jika pilihan tersebut memberi pemain terlalu banyak keuntungan dibandingkan opsi lainnya. Maka dari itu penting untuk merancang opsi dan pilihan pemain dengan hati-hati, karena salah satu dari opsi ini dapat menyebabkan satu strategi mendominasi yang lainnya, mengurangi variasi dan keseruan dalam permainan.

b Menggabungkan Elemen Peluang



Peran peluang dalam sebuah permainan sangat bervariasi. Beberapa permainan, seperti catur, sama sekali tidak menggunakan elemen peluang. Dalam permainan lain, peluang adalah yang terpenting. Namun, seperti yang kita bahas sebelumnya bahwa keterampilan harus menjadi faktor utama dalam menentukan keberhasilan pemain (kecuali dalam permainan untuk anak-anak).

Jika peluang tetap berperan dalam permainan yang kamu desain, maka perlu kamu pastikan bahwa pemain yang lebih terampil adalah pemenangnya. Salah satu caranya dengan menggunakan peluang dengan bijak. Kamu bisa merancang permainan sehingga peluang hanya mempengaruhi sebagian kecil tindakan pemain yang mengarah pada kemenangan, dan sebagian besar tindakan pemain bergantung pada keterampilan. Ini adalah solusi paling sederhana, tetapi tidak cocok untuk semua jenis permainan.

Jika peluang memainkan peran yang lebih besar dalam permainan, maka keseimbangan harus dilakukan dengan cara-cara berikut:

1. Gunakan peluang dalam tantangan yang sering terjadi dengan risiko dan imbalan yang kecil daripada dalam tantangan yang jarang terjadi dengan risiko dan imbalan yang besar.

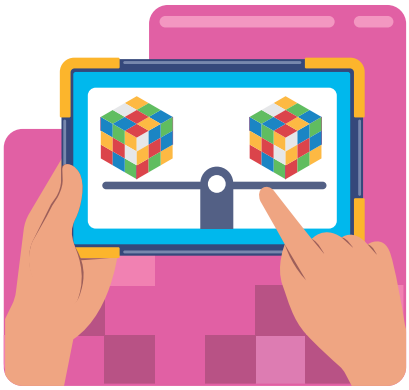
   Contoh: untuk sebuah gim aksi, terdapat sebuah aksi serangan umum kepada musuh terdapat peluang untuk meningkatkan +3 serangan tetapi juga dapat mengenai diri sendiri sehingga dapat mengurangi -3 dari nyawa pemain, tetapi untuk serangan khusus terdapat peluang untuk meningkatkan +10 serangan dan juga dapat mengenai diri sendiri sehingga dapat mengurangi -10 dari nyawa pemain. Serangan khusus hanya dapat dilakukan setelah 10 kali melakukan serangan umum.

2. Izinkan pemain memilih tindakan untuk memanfaatkan peluang demi keuntungannya
3. Izinkan pemain untuk memutuskan seberapa besar risikonya

   Untuk kedua poin ini, kita contohkan seperti contoh sebelumnya tetapi di dalam kasus ini terdapat tombol khusus untuk melakukan serangan khusus sehingga keputusan penggunaan tidak di dalam sistem.

C Membuat Gim yang Adil

1. Gim PvP



Pemain umumnya akan menganggap permainan PvP adil jika mereka merasa bahwa aturan permainan memberikan setiap pemain peluang yang sama untuk menang sejak awal permainan. Artinya, tidak ada pemain yang diuntungkan atau dirugikan secara tidak adil hanya karena mekanisme permainan itu sendiri. Selain itu, aturan permainan juga harus memastikan bahwa selama permainan berlangsung, tidak ada faktor yang memberi keuntungan atau kerugian kepada pemain secara tidak merata, kecuali jika hal tersebut berasal dari elemen peluang yang memang menjadi bagian dari desain permainan.

Dengan kata lain, pemain ingin merasa bahwa keberhasilan mereka tergantung pada keterampilan dan keputusan mereka sendiri, bukan karena adanya ketidakseimbangan yang tak dapat mereka pengaruhi. Jika aturan permainan tidak mendukung ini, maka pemain bisa merasa bahwa permainan itu tidak adil dan kurang menyenangkan untuk dimainkan. Sebaliknya, dengan menjaga kesetaraan dalam peluang yang diberikan kepada setiap pemain, permainan PvP bisa tetap menarik dan memberikan pengalaman yang adil bagi semua pihak.

2. Gim PvE

Karena tantangan dalam permainan PvE berasal dari lingkungan dan bukan dari pemain lain, maka membuat permainan yang adil melibatkan lebih dari sekadar memberi semua pemain kesempatan yang sama untuk berhasil. Secara umum, pemain akan menganggap permainan PvE adil jika permainan tersebut menunjukkan kualitas-kualitas berikut:

1. Permainan harus menawarkan tantangan kepada pemain pada tingkat kesulitan maksimum yang konsisten, tanpa lonjakan tiba-tiba
2. Pemain tidak boleh tiba-tiba kalah dalam permainan tanpa peringatan dan bukan karena kesalahannya sendiri
3. Tidak adanya kebuntuan
4. Permainan ini tidak meminta pemain untuk membuat keputusan penting tanpa informasi yang memadai.

5. Semua pengetahuan faktual yang dibutuhkan untuk memenangkan permainan harus ada di dalam permainan
6. Permainan ini tidak mengharuskan pemain untuk menghadapi tantangan yang biasanya tidak ada dalam genre permainan tersebut

d Mengelola Kesulitan

Keseimbangan gim mencakup pengelolaan tingkat kesulitan tantangan. Jika kamu membuat tantangan dalam gim terlalu sulit, maka pemain akan menjadi cemas, dan jika tantangan yang terlalu mudah, maka pemain akan menjadi bosan. Hal itu untuk menjaga pemain dalam kondisi *flow*, titik di mana kemampuan mereka sesuai dengan masalah yang mereka hadapi. Tujuanmu bukan hanya untuk menetapkan tingkat kesulitan, tetapi untuk memikirkan cara menyesuaikan tingkat kesulitan untuk memaksimalkan kesenangan pemain.

e Desain untuk Mempermudah Pengaturan

Pada tahap akhir pengembangan gim, kamu akan menghabiskan banyak waktu untuk menyempurnakan gim yang kamu buat. Biasanya hal itu untuk meningkatkan keseimbangan, menghilangkan strategi dominan, atau lonjakan kesulitan yang mungkin muncul. Berikut beberapa saran untuk mempermudah melakukan proses ini:

1. **Ubah hanya satu parameter pada satu waktu**

   Sesuaikan satu parameter, lalu periksa hasilnya, lalu sesuaikan parameter lain, dan seterusnya. Mungkin akan tampak membosankan tetapi ini sangat penting. Jika kamu mengubah lebih dari satu parameter pada satu waktu, kamu tidak akan tahu perubahan mana yang berhasil dan perubahan yang tidak berhasil.

2. **Saat mengubah parameter, lakukan penyesuaian besar, bukan penyesuaian kecil**

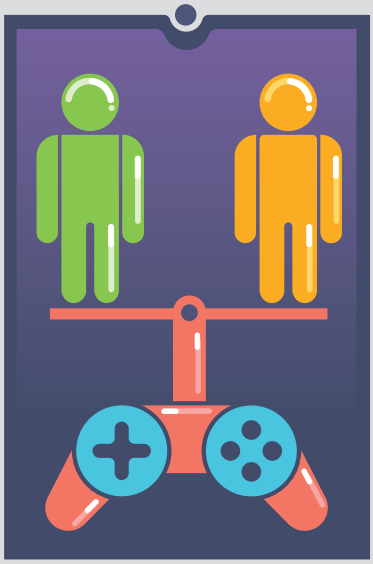
   Penyesuaian kecil dapat menghasilkan perubahan yang sangat halus sehingga kamu tidak dapat mendeteksinya. Buat perubahan besar, lalu kurangi dan uji secara berulang, bergerak menuju nilai ideal. Mengubah dengan faktor yang besar akan memudahkan kamu untuk mencapai pengaturan optimal

3. Simpan catatan

Penguji yang baik selalu mencatat apa yang mereka lakukan sehingga mereka tidak membuang-buang tenaga dengan mencoba hal yang sama dua kali. Hasilnya, mereka dapat melihat dampak perubahan yang telah mereka buat dan belajar dari pengalaman.

3. Jenis Keseimbangan Gim

a Keadilan (Fairness)



Salah satu kualitas yang secara umum dicari oleh pemain dari sebuah permainan adalah keadilan atau fairness.

1. Gim Simetris

Saat kamu memainkan sebuah gim, tentu kamu ingin merasakan bahwa tantangan atau kekuatan yang ada dalam permainan tidak memiliki keuntungan yang membuat mereka mustahil untuk dikalahkan, bukan? Nah, salah satu cara paling sederhana untuk memastikan hal ini adalah dengan memberikan sumber daya dan kekuasaan yang setara bagi semua pemain dalam permainan. Pendekatan ini dikenal sebagai permainan simetris.

Contoh permainan yang menggunakan metode simetris adalah catur dan monopoli. Dalam kedua permainan ini, setiap pemain memulai dengan keunggulan yang sama dan aturan yang adil, sehingga tidak ada pihak yang memiliki keuntungan lebih besar dibandingkan pemain lainnya.

2. Gim Asimetris

Jika kamu ingin memberikan sumber daya dan kemampuan yang berbeda kepada para pemain dalam sebuah gim, maka kamu harus mampu menciptakan serta menyeimbangkan gim tersebut dengan signifikan. Kenapa begitu? Karena semua setiap orang memiliki cara

bermain yang berbeda, *approach* yang berbeda kepada sebuah masalah, sehingga Ketika satu problem dapat diselesaikan dengan cara berbeda, maka keputusan pemain akan lebih impactful dan merasakan pengalaman yang *personalized* sesuai dengan caranya menyelesaikan suatu masalah.

Ada beberapa alasan yang mungkin membuat kamu memilih untuk menciptakan gim asimetris. Salah satunya adalah keinginan untuk mensimulasikan situasi nyata. Misalnya, kamu ingin menciptakan gim yang mencerminkan peran berbeda dalam suatu skenario tertentu, sehingga pemain bisa merasakan pengalaman yang lebih autentik.

Selain itu, gim asimetris juga memberikan kesempatan kepada pemain untuk menjelajahi ruang permainan dengan cara yang unik dan personal. Ketika kamu mendesain permainan seperti ini, pemain bisa memilih kekuatan yang sesuai dengan kemampuan atau gaya bermain mereka. Dengan begitu, mereka dapat fokus pada keahlian terbaik mereka, membuat pengalaman bermain menjadi lebih bermakna dan menarik.

Alasan lainnya adalah untuk menyamakan kedudukan antar pemain dengan cara yang berbeda dan menciptakan situasi permainan yang lebih dinamis. Dengan memberikan kemampuan yang berbeda kepada setiap pemain, kamu bisa menghadirkan tantangan baru yang membuat permainan lebih menarik dan tidak monoton.

3. Gim Gunting, Kertas, Batu

Cara sederhana lainnya yang bisa kamu gunakan untuk menyeimbangkan elemen demi keadilan dalam permainan adalah dengan dengan memastikan bahwa setiap elemen yang memiliki keunggulan di satu sisi juga memiliki kelemahan di sisi lain. Pendekatan ini membuat setiap pilihan menjadi seimbang dan tidak ada yang terlalu dominan.

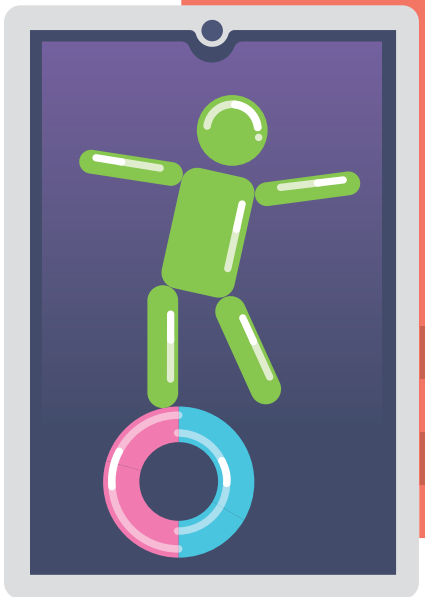
Contoh ikonik dari konsep ini adalah permainan batu, kertas, gunting dengan ketentuan gunting pemecah batu, kertas membungkus batu dan gunting memotong kertas. Dengan aturan ini, maka setiap komponen memiliki kelebihan dan kekurangan dari komponen lain yang menyebabkan keterikatan antar komponen yang membuat sistem game tidak terdapat celah atau gameloop bisa dikatakan tertutup.

Aktivitas 3.3 **Kerja Tim**

Buatlah tim yang beranggotakan 3 orang, selanjutnya kalian jawab beberapa pertanyaan di bawah ini:

1. Apakah gim yang kalian buat asimetris atau simetris? Berikan alasannya!
2. Bagaimana kalian memastikan gim tersebut sesuai dengan jawaban pada pertanyaan pertama, jelaskan menggunakan komponen gim yang akan kalian buat!
3. Menurut kalian mana yang lebih penting, membuat gim yang bisa mengetahui siapa yang memiliki keterampilan paling banyak atau gim yang bisa memberikan tantangan yang menarik bagi pemain?

b Tantangan dan Kesuksesan



Bisa membuat pemain tetap berada dan bertahan memainkan gim yang kamu buat adalah sesuatu yang akan selalu kamu inginkan. Jika pemain memainkan gim dengan sangat mudah maka ini bisa membuat pemain jenuh. Begitu juga jika pemain memainkan gim yang terlalu menantang, bisa membuat mereka frustasi. Untuk itu kamu harus bisa menjaga keseimbangan dikondisi tengah dengan tetap menjaga keseimbangan pengalaman, tantangan, dan kesuksesan dengan tepat.

Menjaga kesimbangan ini akan jadi terasa sulit jika para pemain memiliki keterampilan yang berbeda-beda. Bagi sebagian pemain akan merasakan bosan dengan permainannya, tapi bisa jadi pemain lain justru merasa ada tantangan dalam permainan yang sama. Berikut ini beberapa teknik yang bisa digunakan untuk menjaga keseimbangan yang tepat dalam gim:

1. Meningkatkan tingkat kesulitan berdasarkan keberhasilan-keberhasilan yang dicapai oleh pemain.
2. Membiarkan pemain melewati bagian-bagian yang mudah dari gim dengan cepat.
3. Membuat beberapa lapisan atau tingkatan untuk setiap tantangan.
4. Membiarkan pemain memilih tingkatan levelnya sendiri
5. Melakukan *playtest* dengan berbagai macam pemain gim.

C Pilihan Bermakna (*Meaningful Choices*)

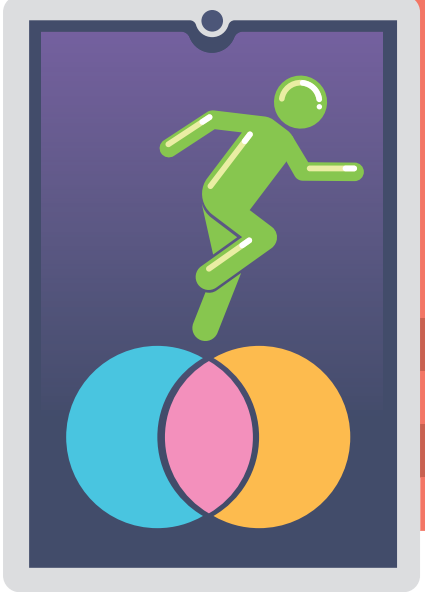


Dalam sebuah gim, jika kamu bisa memberikan pilihan bermakna kepada pemain, maka hal ini akan sangat memengaruhi apa yang terjadi selanjutnya dalam permainan dan bahkan bagaimana permainan tersebut akan berakhir. Pilihan yang bermakna membuat pemain merasa lebih aktif dalam permainan yang menyebabkan permainan akan lebih *imersif* dan pemain akan lebih menikmati permainan dibandingkan pilihan yang tidak bermakna membuat pemain merasa keputusan yang dia buat tidak memiliki efek apapun di game tersebut sehingga membuatnya berhenti bermain.

Pilihan bermakna bisa dirasakan oleh pemain dengan cara menjawab beberapa pertanyaan berikut:

1. Bagaimana saya harus membentuk karakter gim saya?
2. Apakah saya akan menyelesaikan permainan ini dengan cepat atau berhati-hati?
3. Apakah dalam permainan ini saya akan lebih menekankan fokus untuk menyerang atau bertahan?
4. Strategi apa yang akan saya gunakan?
5. Apakah saya harus bermain aman atau mengambil resiko?

d Keterampilan vs Peluang (*Skill vs Chance*)



Pada kenyataannya, dua hal ini tidak dapat dipisahkan dan akan menjadi dua kekuatan yang berlawanan dalam gim apapun. Jika terlalu banyak peluang atau *chance,* maka akan berdampak pada hilangnya keterampilan atau *skill*, begitu juga sebaliknya. Memang akan sulit untuk menentukan kedua kondisi ini, karena ada beberapa pemain yang lebih menyukai banyak elemen peluang dan ada juga beberapa pemain yang menginginkan banyaknya elemen keterampilan. Metode yang tepat untuk menyeimbangkan kedua hal ini adalah dengan menggunakan keduanya secara bergantian.

e *Head vs Hands*



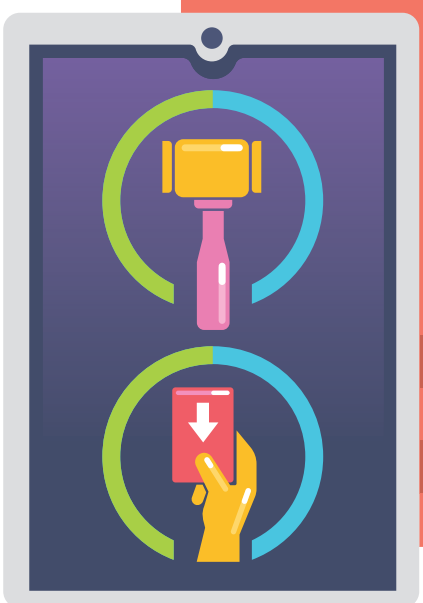
Ini adalah bentuk keseimbangan yang cukup mudah. Kalian cukup menyeimbangkan berapa banyak permainan yang memerlukan banyak aktivitas fisik yang menantang seperti melempar, menekan tombol atau menyetir, dengan berapa banyak aktivitas yang kalian perlukan yang melibatkan pemikiran. Banyak gim yang menggabungkan elemen strategi dan ketangkasan secara bersamaan dan berkelanjutan. Kedua elemen ini saling melengkapi dan sulit untuk dipisahkan. Dalam gim semacam ini, kamu akan diminta untuk memikirkan strategi yang matang sekaligus mengandalkan ketangkasan dan ketepatan untuk menjalankan strategi tersebut.

f Kompetitif vs Kooperatif (*Competition vs. Cooperation*)



Dalam sebuah gim, permainan kompetitif itu lebih banyak dimainkan dari pada permainan kooperatif, walaupun banyak sekali permainan kooperatif yang telah diciptakan. Ada permainan yang dimenangkan dengan cara melakukan koordinasi bersama dengan beberapa pemain, dan ada juga permainan yang dimainkan oleh beberapa pemain secara bersamaan dan berdampingan untuk bisa naik ke level gim berikutnya.

g Hukuman (*Punishment*)



Ketika ada ide untuk memberikan hukuman pada pemain dalam sebuah permainan adalah hal yang dianggap tidak umum. "Bukankah permainan itu seharusnya menyenangkan untuk dimainkan?" tetapi bisa saja terjadi bahwa hukuman yang diberikan dalam permainan ini justru dapat menambah keseruan dan keasyikan dalam memainkan gim tersebut. Lalu kenapa sebuah permainan memberikan hukuman? Karena permainan akan lebih mengasyikan jika pemain dapat mengambil resiko, hukuman dalam permainan memungkinkan untuk meningkatkan sebuah tantangan.

Beberapa contoh hukuman yang umum digunakan dalam sebuah permainan di antaranya:

1. Kehilangan poin.
2. Waktu permainan akan lebih singkat karena adanya pengurangan waktu.
3. Hilangnya kekuatan dalam permainan.
4. Berkurangnya sumber daya dalam permainan

Penggunaan hukuman dalam sebuah permainan hendaknya dilakukan dengan hati-hati, karena para pemain memainkan gim dengan keinginan dan kemauan sendiri. Melakukan keseimbangan yang tepat dalam pemberian hukuman akan mampu menciptakan permainan yang lebih bermakna, karena pemain akan merasakan kebanggaan saat berhasil memainkan permainan yang kamu buat.

h Rewards



Beberapa bentuk *rewards* atau hadiah yang diberikan gim pada para pemainnya:

1. Pujian

Sebagai hadiah yang paling sederhana, permainan ini hanya memberi tahu pemain bahwa mereka telah melakukan pekerjaan dengan baik, baik melalui pernyataan eksplisit, efek suara khusus, atau bahkan karakter dalam permainan yang berbicara kepada pemain. Semuanya mengarah pada hal yang sama yaitu permainan telah menilai pemain, dan menyetujuinya. Permainan Nintendo terkenal karena memberikan banyak pujian sekunder kepada pemain melalui suara dan animasi untuk setiap hadiah yang mereka dapatkan.

2. Poin

Dalam banyak permainan, poin tidak memiliki tujuan selain sebagai ukuran keberhasilan pemain, baik melalui keterampilan maupun keberuntungan. Terkadang poin ini merupakan gerbang menuju hadiah lain, tetapi sering kali, ukuran keberhasilan ini sudah cukup, terutama jika orang lain dapat melihatnya pada daftar skor tertinggi.

3. Kekuatan

Menjadi lebih kuat adalah sesuatu yang diinginkan semua orang dalam kehidupan nyata, dan dalam permainan. Menjadi lebih kuat kemungkinan akan meningkatkan penilaian permainan terhadap keberhasilan pemain. Kekuatan ini dapat muncul dalam berbagai bentuk, seperti menjadi "raja" dalam Catur, menjadi tinggi dalam Super Mario World, mempercepat langkah dalam Sonic The Hedgehog. Hal yang sama dari semua kekuatan adalah bahwa mereka memberi pemain cara untuk mencapai tujuan lebih cepat daripada sebelumnya.

4. Sumber Daya

Permainan video lebih sering memberi hadiah berupa sumber daya yang hanya dapat digunakan dalam permainan, seperti makanan, energi, amunisi, poin hit. Beberapa permainan memberikan sumber daya secara langsung, seperti uang virtual yang dapat dipilih pemain untuk digunakan.

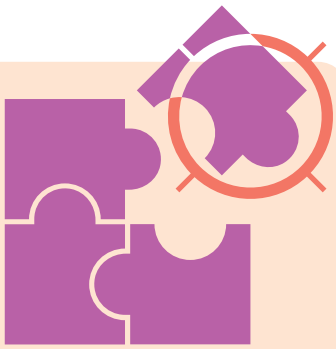
5. Penyelesaian

Menyelesaikan semua tujuan dalam permainan memberikan perasaan akhir yang istimewa bagi pemain, yang jarang mereka dapatkan dari menyelesaikan masalah dalam kehidupan nyata. Dalam banyak permainan, ini adalah hadiah utama. ketika pemain telah mencapai titik ini, sering kali tidak ada gunanya untuk memainkan permainan lebih jauh.

4. Metode Keseimbangan Gim

Setelah kamu mengetahui tentang bentuk dari keseimbangan gim, sekarang kamu akan mempelajari bagaimana metode umum yang digunakan dalam melakukan keseimbangan gim. Kamu perlu melakukan beberapa percobaan untuk menemukan metode yang tepat untuk kamu gunakan. Berikut beberapa metode yang bisa kamu gunakan (Jesse Schell 2008):

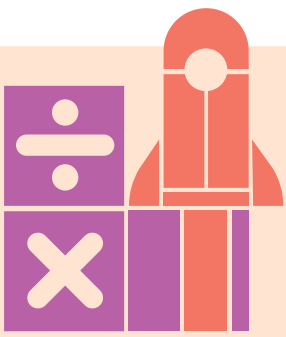
a. Gunakan sudut pandang pernyataan masalah



Gim yang tidak seimbang sering kali menjadi tantangan besar bagi desainer, tetapi masalah ini dapat diatasi dengan memulai dari pernyataan masalah yang jelas. Sebuah pernyataan masalah yang terdefinisi dengan baik membantu kamu memahami akar penyebab ketidakseimbangan dan menemukan solusi yang tepat.

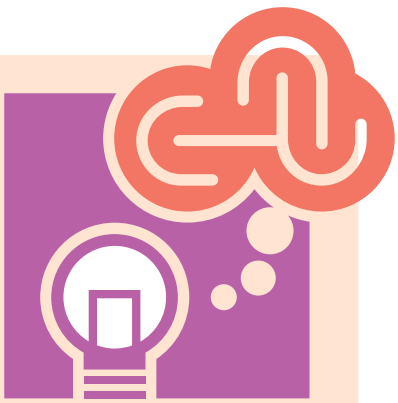
Banyak desainer yang terburu-buru mencoba memperbaiki gim mereka dengan memberikan solusi penyeimbangan tanpa benar-benar menganalisis masalahnya terlebih dahulu. Akibatnya, mereka malah membuat perubahan yang tidak efektif atau bahkan menciptakan masalah baru dalam gim tersebut.

b. *Doubling and halving*



Saat mengubah nilai untuk menyeimbangkan permainan, kamu akan membuang-buang waktu dengan mengubahnya dalam jumlah kecil. Sebaliknya, mulailah dengan menggandakan atau membagi dua nilai ke arah yang seharusnya. Misalnya, jika roket menghasilkan 100 poin kerusakan, dan kamu pikir itu mungkin terlalu banyak, jangan kurangi 10 atau 20, tetapi tetapkan nilai kerusakan menjadi 50, dan lihat bagaimana hasilnya. Jika terlalu rendah, cobalah angka di antara 50 dan 100. Dengan mendorong nilai lebih jauh dari intuisi kamu, maka batas keseimbangan yang baik akan mulai menjadi lebih jelas dengan cepat

c. Latih intuisi kamu untuk menebak dengan tepat



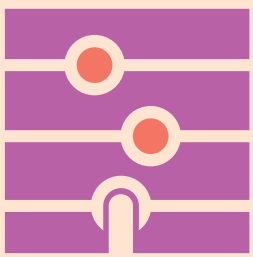
Semakin banyak desain permainan yang kamu buat, intuisi kamu akan semakin baik. Kamu akan dapat melatih intuisimu untuk menyeimbangkan permainan baik dengan berlatih menebak dengan tepat. Setelah kamu sampai pada tebakan intuitif ini, masukkan dan lihat. Kamu mungkin merasa tebakannya terlalu rendah, atau terlalu tinggi, atau mungkin bahkan tepat sekali. jika kamu sudah terbiasa melatih intuisimu maka kamu akan dapat membuat tebakan yang akurat.

d. Dokumentasikan model



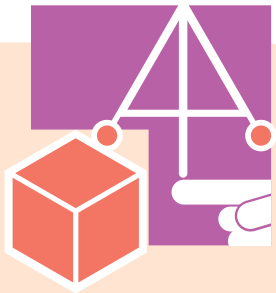
Kamu harus bisa menuliskan apa saja yang menurutmu berkaitan dengan keseimbangan gim yang kamu buat. Ini akan membantu memperjelas pemikiran kamu dan memberi kamu kerangka kerja untuk mencatat hasil eksperimen penyeimbangan permainan yang kamu buat.

e. Sesuaikan model saat kamu mengatur permainan yang kamu buat



Dengan setiap percobaan penyeimbangan yang sudah kamu coba, kamu tidak hanya mencatat apakah itu meningkatkan permainan, tetapi juga apakah percobaan tersebut sesuai dengan model yang kamu buat. Kemudian, kamu harus mengubah modelnya jika tidak sesuai dengan yang kamu harapkan. Menuliskan pengamatan dan model ini akan sangat membantu kamu.

f. Rencanakan untuk melakukan keseimbangan



Saat kamu melakukan perancangan sebuah gim, kamu mungkin memiliki ide yang cukup bagus tentang aspek apa saja yang perlu diseimbangkan. Manfaatkan itu, dan terapkan sistem yang memudahkan kamu untuk mengubah nilai yang diharapkan harus diseimbangkan.

Aktivitas 3.4 Kerja Tim

Jika kalian sudah memahami materi keseimbangan gim, buatlah sebuah tim yang beranggotakan 3 orang. Kemudian kalian ikuti petunjuk kerja berikut ini:

1. Ubahlah aturan dari gim catur untuk membuat permainan menjadi asimetris.
2. Uji coba hasilnya dengan seorang teman dalam tim kalian untuk melihat apakah permainan masih adil.
3. Buatlah sebuah makalah singkat yang menjelaskan perubahan yang kamu lakukan, termasuk jenis dan jumlah unit, jenis gerakan yang diizinkan, dan perubahan pada kondisi kemenangan untuk satu atau kedua belah pihak.

Setelah itu, presentasikan hasil kerja kalian di depan kelas untuk saling berbagi informasi dengan teman yang lainnya.



Uji Kompetensi

Coba kamu jawab pertanyaan di bawah ini untuk mengetahui apakah kamu sudah memahami materi tentang desain prototipe gim dalam bab ini!

1. Sebutkan komponen yang penting ketika membuat prototipe gim?
2. Apakah prioritas tertinggi ketika melakukan prototipe fisik dari gim yang sedang kamu buat?
3. Apakah perbedaan dari Prototipe Fisik dan Prototipe Digital?
4. Kapan kita disarankan lebih memakai Prototipe Fisik dibandingkan Prototipe Digital?
5. Kapan kita disarankan lebih memakai Prototipe Digital dibandingkan Prototipe Fisik?
6. Keseimbangan merupakan hal penting yang harus kalian perhatikan dalam membuat atau merancang sebuah permainan. Jika kamu memainkan sebuah permainan yang terlalu mudah, maka kamu akan merasa jenuh atau saat kamu memainkan sebuah permainan yang sangat sulit untuk diselesaikan atau dimenangkan, maka kamu akan merasa frustasi. Untuk Maka dari itu diperlukan keseimbangan dalam sebuah permainan. Tentukanlah pernyataan yang benar atau salah terkait keseimbangan gim, dengan memberikan tanda centang (✓) pada kolom yang sesuai.

No	Pernyataan	Benar	Salah
1	Keseimbangan gim atau *game balance* adalah proses mendesain gim yang berkelanjutan tentang menyesuaikan komposisi mekanika, hadiah, tantangan gim, dll.		
2	Saat kamu melakukan perancangan sebuah gim, kamu mungkin memiliki ide yang cukup bagus tentang aspek apa saja yang perlu diseimbangkan. Manfaatkan itu, dan terapkan sistem yang memudahkan kamu untuk mengubah nilai yang diharapkan harus diseimbangkan.		
3	Gim yang seimbang adalah masalah yang akan sangat diuntungkan dengan pernyataan masalah yang jelas. Banyak desainer yang akhirnya mengacaukan gim mereka dengan langsung memberikan solusi penyeimbangan sebelum mereka berpikir jernih tentang apa sebenarnya masalahnya.		
4	Menjadi lebih kuat adalah sesuatu yang diinginkan semua orang dalam kehidupan nyata. Dalam permainan, menjadi lebih kuat akan meningkatkan penilaian permainan terhadap keberhasilan pemain.		

7. Saat kamu mempunyai ide besar dalam membuat sebuah gim, maka kamu harus bisa mencari tahu atau memeriksa apakah gim kamu akan berhasil terwujud atau tidak dan apakah layak untuk di kembangkan atau tidak. Maka saat seperti inilah kamu membutuhkan sebuah prototipe. Berikut ini adalah hal-hal yang tepat terkait dengan prototipe, kecuali ...

   a. Versi awal dari sebuah gim yang digunakan untuk menguji konsep, mekanika, dan fitur-fitur utama dari gim tersebut sebelum dikembangkan lebih lanjut menjadi gim yang lebih lengkap
   b. Prototipe memungkinkan untuk menguji dan mengevaluasi konsep atau ide secara praktis sebelum komitmen besar terhadap produksi atau implementasi
   c. Dengan menggunakan prototipe, masalah atau kekurangan dalam desain atau fungsi sulit untuk dapat diidentifikasi lebih awal, sehingga memungkinkan perbaikan sebelum tahap produksi
   d. Prototipe merupakan langkah awal yang bisa dilakukan untuk menggambarkan konsep atau ide yang bisa dipastikan dapat terwujud atau tidak
   e. Prototipe yang sudah pernah dibuat bisa dijadikan sebagai bahan evaluasi untuk mengembangkan gim baru berikutnya, dengan mencari solusi terbaik dari kekurang yang ditemukan pada prototipe sebelumnya.

8. Pertanyaan berikut ini adalah contoh pertanyaan yang bisa kamu gunakan untuk melakukan keseimbangan antara keterampilan dan peluang (*skill and chance*), kecuali ...

   a. Apakah pemain yang memainkan gim saya ini akan dinilai atau untuk mengambil resiko?
   b. Apakah gim yang akan saya buat ini serius atau santai?
   c. Apakah bagian dari gim yang saya buat ini membosankan?
   d. Apakah gim yang saya buat ini tidak jelas dan acak?
   e. Apakah para pemain butuh pilihan bermakna?

9. Pasangkan alasan dan penjelasan yang sesuai

Mensimulasikan suatu situasi yang ada dalam dunia nyata	Permainan yang kamu buat apakah akan dijadikan sebagai ukuran standar dari keterampilan pemain ataukah hanya ingin memberikan tantangan pada para pemain
Memberikan cara lain menjelajahi ruang permainan pada para pemain	Pemain yang diberi kekuatan yang berbeda sesuai kemampuan mereka, maka mereka akan mampu membuat permainan dengan menekankan pada keahlian terbaik mereka.
Personalisasi	Jika kamu membuat gim bertujuan untuk menggambarkan atau mensimulasikan perang Dunia II, maka gim simetris ini menjadi tidak masuk akal
Menyamakan kedudukan	Dari banyak permainan yang dibuat, lebih banyak permainan asimeteris yang dimainkan. Seringkali Para pemain lebih tertarik permainan yang mengadu kekuatan asimetris satu dan yang lainnya, karena strategi yang digunakan tidak terlalu jelas.
Menciptakan situasi yang menarik	Pemain akan sangat senang melakukan eksplorasi terhadap suatu permainan dengan memainkan permainan yang sama tapi dengan sumber daya dan kekuatan yang berbeda.

10. Perhatikan pertanyaan-pertanyaan berikut :

1) Apakah dengan menambahkan lebih banyak tempat untuk gim puzzle akan membuat gim ini menjadi lebih menarik?
2) Apakah ada gim yang membuat otak pemain itu santai dari permainan tanpa berpikir?
3) Apakah saya dapat memberikan pilihan pada pemain untuk sukses dengan cara meningkatnya ketangkasan atau dengan kecerdasan sebuah strategi?

Pertanyaan-pertanyaan di atas adalah cara untuk membuat keseimbangan jenis ...

a. *Skill vs chance*
b. *Hands vs head*
c. *Meaningfull choice*
d. *Challenge vs success*
e. *Fairness*

11. Dalam sebuah gim, jika kamu bisa memberikan pilihan bermakna pada para pemain gim tersebut, maka ini akan sangat besar pengaruhnya pada apa yang akan terjadi selanjutnya dalam gim dan bagaimana gim itu akan berakhir. Pilihan bermakna ini bisa dirasakan oleh pemain dengan cara menjawab beberapa pertanyaan berikut ini pada diri mereka sendiri. Berikan tanda centang pada pertanyaan yang tepat untuk bisa merasakan pilihan bermakna bagi pemain dalam sebuah permainan.

Pertanyaan	Pilih yang tepat
Bagaimana saya harus membandingkan karakter gim saya?	
Apakah permainan ini akan saya selesaikan dengan cepat atau berhati-hati?	
Apakah saya harus bermain aman atau mengambil resiko?	
Apakah saya harus menyediakan waktu khusus untuk menyelesaikan gim?	
Strategi apa yang akan saya gunakan?	

Projek Akhir

Buatlah sebuah prototipe permainan teka teki dengan kebutuhan teknis:

1. Permainan ini merupakan permainan pemain tunggal
2. Buatlah mekanisme mu sendiri untuk mendapatkan poin
3. Anda harus membuat setidaknya 10 jenis tantangan yang berbeda.
4. Tunjukan Tindakan apa yang harus digunakan pemain untuk menjawab setiap tantangan dan hadiah apa yang didapatkan oleh pemain jika menyelesaikan tantangan tersebut.
5. Buatlah desain antarmuka sederhana secara terperinci tanpa harus membuatnya dengan versi final, pastikan semua komponen terdeskripsi dan tata letak kontrol permainan terdefinisi.



Pengayaan

Bagi kamu yang sudah memahami tentang materi dalam bab ini, kamu bisa melakukan eksplorasi lebih mendalam tentang bagaimana membuat sebuah prototipe dan keseimbangan gim ini dengan literasi dan praktik mandiri melalui buku berikut ini:

Referensi Buku

1. *Game design workshop a playcentric approach to creating innovative games*, fourth Edition, Tracy Fullerton, 2019
2. *The Art of Game Design A Book of Lenses,* Jesse Schell, 2008



Refleksi

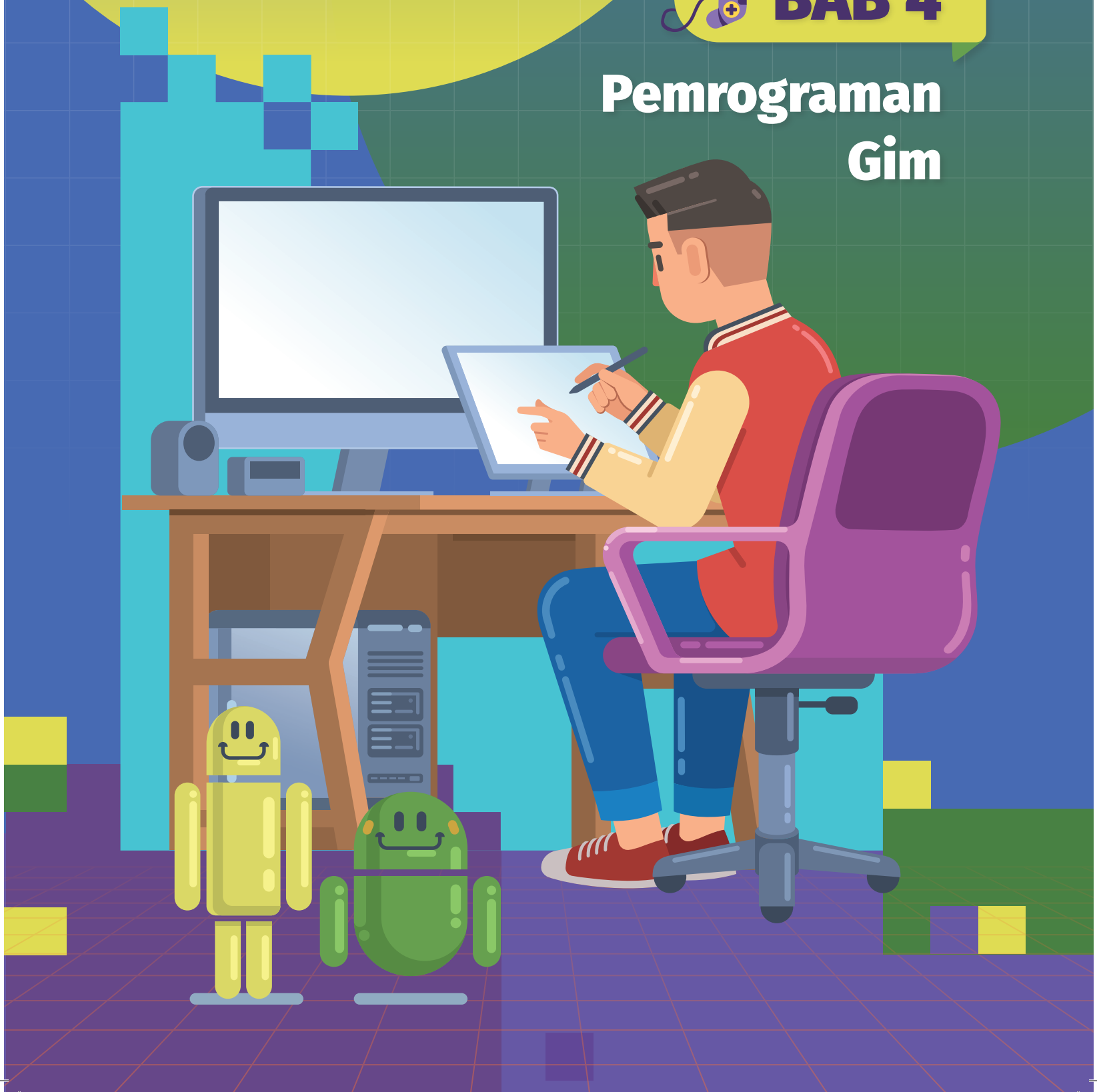
Setelah kalian mempelajari bab ini, sekarang coba kalian melakukan refleksi diri dengan menjawab pertanyaan berikut ini :

1. Apa yang kamu rasakan setelah memahami materi prototipe gim?
2. Apakah kamu semakin tertarik untuk mengembangkan gim terbaikmu?
3. Apakah kamu semakin yakin untuk membuat gim versi kamu sendiri? Jika yakin, bagian mana dalam materi bab ini yang benar-benar kamu kuasai dan bagian mana yang perlu kamu pelajari lebih lanjut lagi?
4. Menurut kamu *soft skill* apa yang harus kamu miliki dan kembangkan untuk bisa bekerja di dunia pengembangan gim ini?
5. Bagaimana caranya agar kamu bisa meningkatkan *soft skill* tersebut?

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2024
Pengembangan Gim
untuk SMK/MAK Kelas XI-XII
Penulis: Rini Melati dan Muhammad Ardhan Fadlurrahman
ISBN: 978-634-00-0360-4

BAB 4

Pemrograman Gim



Tujuan Pembelajaran

Setelah mempelajari materi dalam bab ini, kamu akan dapat menerapkan struktur data, menerapkan integrasi objek statis dan dinamis (*static and dynamic assets integration*), menerapkan fungsionalitas tambahan pada *game engine* (*tools and plugin implementation*), dan menerapkan pemrograman berbasis teks dan grafis yang diintegrasikan pada pemrograman gim (*game engine*), mencakup *gameplay* dan UI/UX (*graphical user interface*).

Kata Kunci

UI/UX

Objek Statis

Objek Dinamis

Game Engine

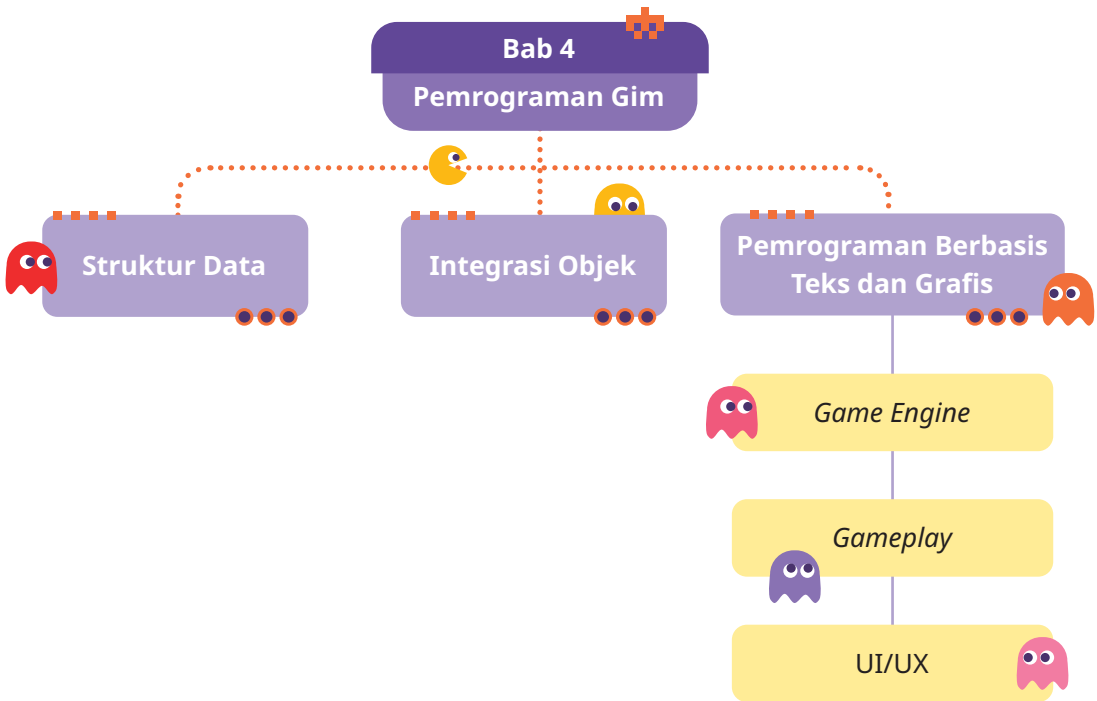
Struktur Data

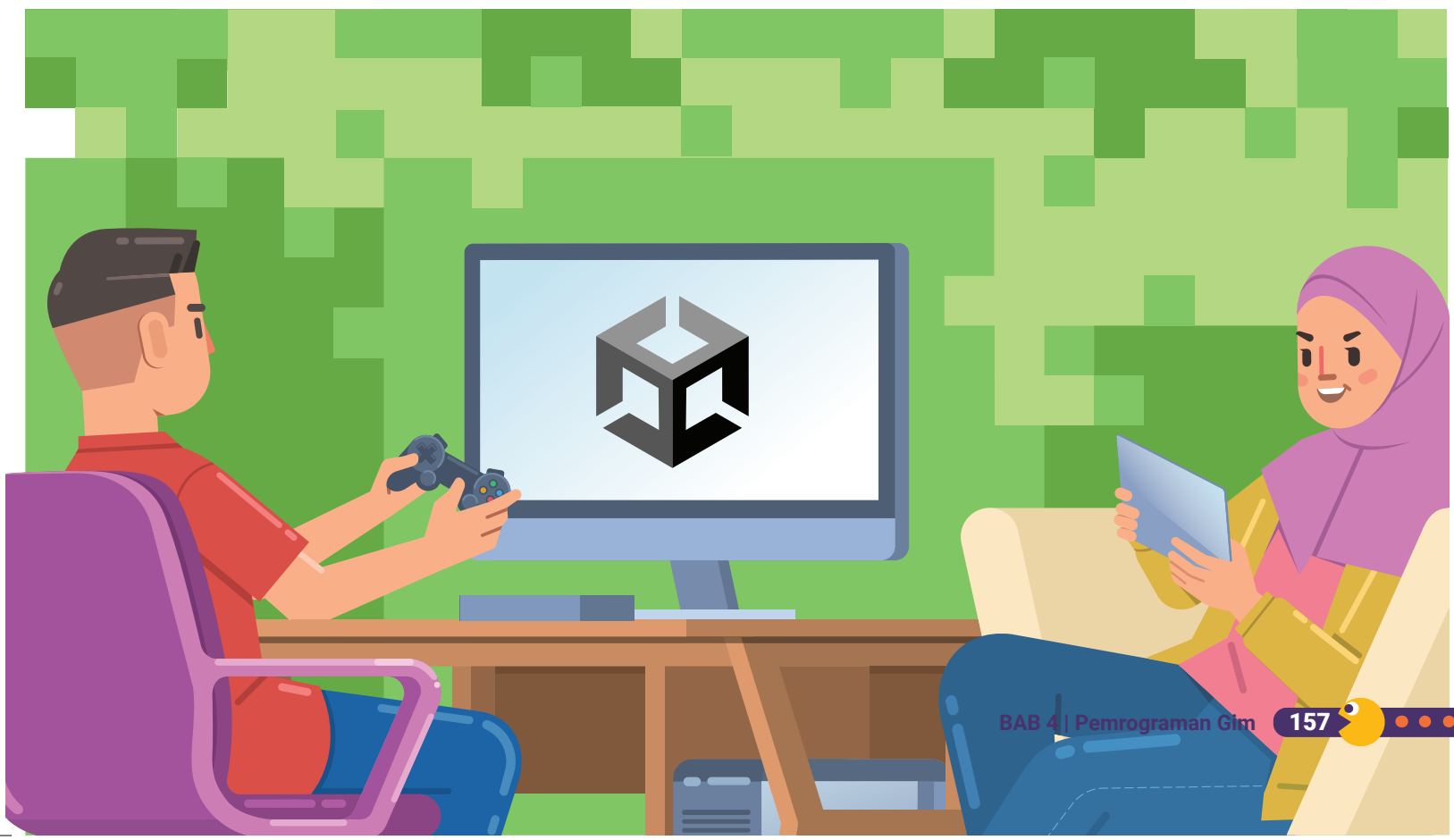
Plug in

Soft Skill

- **Pemecahan masalah** untuk menemukan solusi kreatif dan efektif dalam masalah *bug* dan masalah teknis yang kompleks dalam pengembangan gim.
- **Teliti** dalam mencegah *bug*, menemukan masalah performa, serta memastikan setiap bagian gim berjalan lancar.
- **Bertanggung jawab** atas kualitas kode dan menyelesaikan tugas tepat waktu
- **Kerja sama** dengan anggota tim lain untuk mengintegrasikan berbagai elemen dalam gim secara efektif.
- **Beradaptasi** dengan cepat terhadap perubahan teknologi dan desain yang terjadi selama proses pengembangan gim.

Peta Materi


Bab 4
Pemrograman Gim
Struktur Data
Integrasi Objek
Pemrograman Berbasis Teks dan Grafis
Game Engine
Gameplay
UI/UX



Saat kamu menyukai sebuah gim dan mahir dalam memainkan gim tersebut. Apakah kamu penasaran bagaimana gim itu dibuat? Apakah kamu ingin membuat gim yang banyak dimainkan dan disukai oleh banyak orang? Jika iya, kamu harus tahu dulu tentang pemrograman gim itu sendiri mulai dari bahasa pemrograman yang digunakan, struktur penulisan program, serta UI/UX.

Selain itu, kamu bisa memilih *software* apa yang akan kamu gunakan, dalam membuat pemrograman gim. Salah satu *software* yang umum digunakan dalam pembuatan sebuah gim adalah Unity. Unity merupakan salah satu *platform powerfull* dan memiliki banyak fitur yang dapat kamu manfaatkan untuk membuat sebuah gim. Unity ini juga sangat mendukung pada *multiplatform* seperti PC, *mobile* dan *console*. Selain Unity, kamu juga bisa menggunakan *Unreal Engine* yang juga memiliki fitur-fitur dengan kemampuan membuat gim tanpa koding serta memiliki *blueprint* dan *support* untuk *Augmented Reality* dan *Virtual Reality*.

Pemahaman terhadap dasar-dasar pemrograman diperlukan untuk dapat mengikuti materi dalam bab ini. Materi tersebut telah disampaikan pada dasar-dasar pengembangan perangkat lunak dan gim kelas X fase E. Dalam bab ini akan dibahas penerapan pemrograman gim yang berkaitan dengan struktur data, integrasi objek, fungsionalitas tambahan dan pemrograman berbasis teks dan grafis.

A. Struktur Data



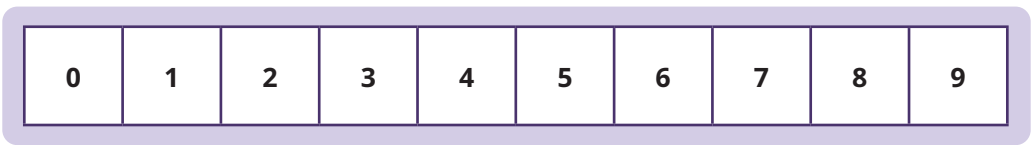
Bayangkan kamu memiliki lemari pakaian, apakah kamu melempar pakaian ke dalam lemari pakaianmu? Atau menyusunnya satu per satu, mengelompokkan pakaian sesuai jenisnya, sehingga kamu mudah mengambil pakaian tersebut. Nah, sama halnya dengan struktur data, yaitu kumpulan elemen data yang kamu kumpulkan dalam satu tempat untuk dirapikan atau dikelola secara lebih spesifik, sehingga kamu dapat menggunakan data secara efisien (Thareja Reema, 2014). Dalam pemrograman gim, pemahaman mengenai struktur data adalah kunci untuk mengelola informasi secara efisien. Struktur data yang tepat tidak hanya meningkatkan kinerja gim tetapi juga mempermudah pengelolaan logika dan interaksi antar elemen.

Jika kamu seorang programmer, maka kamu tidak cukup hanya menulis kode program untuk menyelesaikan masalah tetapi kamu juga harus mampu menganalisis masalah dengan baik. Salah satu aspek penting dalam analisis ini adalah pemilihan bentuk struktur data untuk pekerjaan yang sedang dilakukan. Jika struktur data yang kamu gunakan tidak sesuai, maka dapat mengakibatkan kinerja yang buruk seperti kecepatan operasi yang lambat. Oleh karena itu, kamu harus memiliki pemahaman yang baik tentang konsep dari struktur data, agar dapat menyelesaikan masalah dengan tepat.

Berikut beberapa beberapa bentuk struktur data menurut Ron Penton (2003), di antaranya ***array, tree, stacks and queues, hash tables*** **dan** ***graphs***.

1. Arrays

Dalam komputer, *array* merupakan istilah yang sudah lama ada dan struktur yang paling banyak digunakan. Kamu bisa menganggap *array* ini sebagai sebuah blok penyimpanan. Array merupakan sebuah struktur data satu dimensi yang panjang dan berisi banyak sel. Dalam setiap selnya kamu akan menemukan tepat satu item di dalamnya dengan nomor indeks yang akan digunakan untuk mengakses item tersebut (Ron Penton, 2003). Ketika kamu akan merepresentasikan *array* dalam bentuk gambar, kamu akan menggunakan kotak untuk merepresentasikan selnya, seperti tampak pada gambar berikut ini:



Gambar 4.1 Representasi Gambar Sel Array
Sumber: Rini Melati/Kemendikbudristek (2024)

Saat kamu bekerja dengan *array* maka setiap sel yang ada akan memiliki nomor indeks yang dimulai dari indeks 0 (nol). *Array* ini juga bisa dikatakan struktur akses acak karena kamu dapat langsung mengakses item apapun dalam *array* jika kamu mengetahui indeksnya. Berapapun banyaknya item dalam *array*, kamu tetap membutuhkan waktu yang sama untuk mengakses indeks apapun. berikut ini adalah contoh penerapan *array* dalam gim Mr. Alien:

```
Pada gim Mr. Alien kita akan menggunakan array untuk menyimpan informasi berapa
banyak koin yang akan ditampilkan dalam masing-masing level:

misal kita akan mendefinisikan variabel int[] koinLevel dimana memiliki pengertian
berikut:
int menggambarkan integer yaitu jenis variabel berupa angka bulat
int[] merupakan array integer, sebuah array yang menyimpan jenis data integer.

int[] koinLevel = new koinLevel[] {1,3,5,7,9};

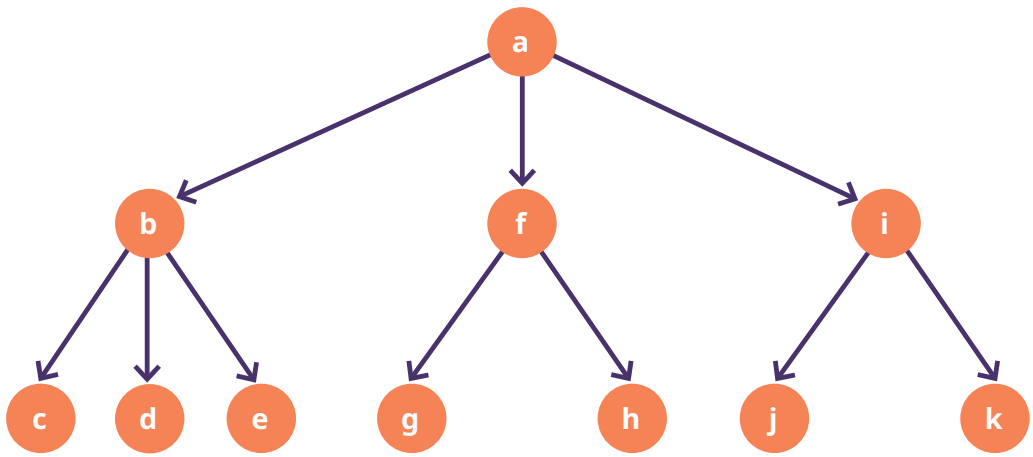
Berikut adalah penjelasan dari satu baris kode di atas:
new koinLevel[] itu adalah sebuah perintah untuk sistem
melakukan inisiasi variabel array integer bernama koinLevel.

{1,3,5,7,9} adalah sebuah inisiasi isi dari data struktur sebuah array yang
terdiri dari 5 buah sehingga ukuran dari array integer sebanyak 5 dengan isi
dari array tersebut ialah angka 1,3,5,7,9. Angka-angka ini merepresentasikan
informasi terkait berapa banyak koin yang ada untuk setiap level.

Contoh mencetak sebuah rekaman debug informasi integer di dalam Unity dalam sebuah
array yang sudah didefinisikan:
Debug.Log(koinLevel[2]);
```

2. Trees

Trees diambil dari kosakata bahasa inggris yang artinya pohon. Kamu pasti sudah pernah melihat pohon, amatilah pohon di sekeliling kalian. Lihatlah struktur pohon tersebut, pohon memiliki beberapa komponen utama, yaitu batang dan akar di bagian bawah. Cabang-cabang yang tumbuh pada bagian batang akan menyebar menjadi ranting yang memiliki daun. Coba kamu perhatikan gambar berikut ini, apakah ini mirip dengan struktur pohon yang kamu lihat?



Gambar 4.2 Bentuk *Tree* Yang Direpresentasikan dalam Komputer
Sumber: Rini Melati/Kemendikbudristek (2024)

Berdasarkan gambar di atas, ada beberapa terminologi dari struktur *tree* yang perlu kamu ketahui, di antaranya:

Istilah	Penjelasan	Contoh
Root	Node teratas dalam struktur *Tree*	node a adalah akar (*root*)
Child	node di bawah yang lain dalam cabang	b, f, i adalah anak (*child*) dari a
Parent	node di atas yang lain dalam cabang	b adalah orang tua (*parent*) dari c, d, e
Sibling	node yang memiliki level yang sama dengan node lainnya	c adalah saudara (*sibling*) dari d dan e
Leaf	node yang tidak memiliki anak	c, d, e, g, h, j dan k adalah node *leaf*
Level	menggambarkan tinggi dari node	a berada di level 0, b, f dan i berada di level 1, c, d, e,d,h,j, dan kberada di level 2
Subtree	struktur *tree* berada dalam struktur *tree* yang lain	b, f dan i adalah akar dari *subtree* a

Dalam struktur *tree*, node akan menunjuk pada anak-anaknya (*childs*), dan setiap node dapat memiliki beberapa anak.

Berikut ini adalah contoh penerapan struktur *tree* dalam gim Mr. Alien:

Pada gim Mr.Alien, dibutuhkan untuk membuat musuh dengan kecerdasan buatan (AI) di mana musuh tersebut dapat mendeteksi apakah mr.Alien berada di dekatnya, dan bagaimana dia bisa memilih pergerakan dirinya selama permainan.

Dengan kebutuhan seperti itu, kita membutuhkan struktur data *Decision Tree*, yang memiliki komponen seperti:

- Memiliki kondisi untuk di cek
- Memiliki sebuah *nodes* yang menjelaskan secara spesifik aksi apa yang dilakukan.

Berikut contoh kodenya:

```
using UnityEngine;
public class EnemyAI : MonoBehaviour
{

  /*
    berikut adalah penjelasan variable
    jangkauanDeteksi = seberapa besar Enemy bisa mulai mendeteksi pemain?
    jangkauanSerang = seberapa besar Enemy bisa mulai menyerang pemain?
    kecepatanGerak = seberapa cepat Enemy dapat bergerak
    Transform pemain = informasi terkait komponen posisi pemain di Unity
  */

    public float jangkauanDeteksi = 5f;
    public float jangkauanSerang = 1.5f;
    public float kecepatanGerak = 2f;
    private Transform pemain;

  /*
    berikut adalah penjelasan dari Decision Tree dari sebuah  EnemyState
    Idle = musuh Diam;
    Kejar = musuh Mengejar;
    Serang = musuh Menyerang;
  */

private enum EnemyState
{
    Idle,
    Kejar,,
    Serang
}

    //stateSaatIni = merupakan variable yang menyimpan informasi state pemain saat ini
    private EnemyState stateSaatIni;

  /*
    fungsi Start() pada unity dijalankan sekali di awal permainan, biasanya digunakan
    untuk melakukan inisiasi dari variable misalkan disini inisiasi variable pemain yang
    disambungkan dengan informasi objek “Player” di dalam permainan dan juga melakukan
```

```
    inisiasi variable stateSaatIni untuk memastikan di awal variabelnya statenya adalah
    Idle
  */

private void Start()
{
    pemain = GameObject.FindingWithTag(“Player”).transform;
    stateSaatIni = EnemyState.Idle;
}

  /*
    fungsi Update() pada unity dijalankan berulang kali sepanjang permainan, disini
    merupakan
    fungsi sebuah gameplay dijalankan sampai akhir permainan, pada fungsi Update() ini
    kita melakukan cek state dari sebuah Enemy sepanjang permainan, dan fungsi dijalankan
    sesuai dari perubahan state tersebut. Misalnya pada suatu waktu terdapat perubahan dari
    Idle ke Kejar, maka fungsi KejarState() akan dijalankan.
  */

private void Update()
{
    switch (stateSaatIni)
{
    case EnemyState.Idle;
    IdleState();
    break;

    case EnemyState.Kejar;
    KejarState();
    break;

    case EnemyState.Serang;
    break;
}
}

  /*
    fungsi idleState() ini digunakan hanya untuk mengecek apakah player berada di
    jangkauan Deteksi , jika iya maka akan terdapat perubahan state dari Idle menjadi Chase
    yang akhirnya akan menjalankan fungsi ChaseState()
  /*


private void IdleState()
{
    if (IsPlayerInjangkauanDeteksi())
{
    stateSaatIni = EnemyState.Chase;
}
}

  /*
    fungsi KejarState() akan dijalankan ketika state dari Enemynya merupakan idle,
    di fungsi ini kita memastikan jika memang dalam jangkauan serang maka
    berubah dari Idle ke Serang dan fungsi lain yang dijalankan, tetapi jika tidak
    maka Enemy akan berjalaln ke arah pemain untuk memastikan musuh tetap
    berada di dekat pemain. disini juga di cek jika memang Pemain tidak berada di
    jangkauan Deteksi, maka kembali ke state Idle.
  */
```

```
private void KejarState()
{
    if (IsPlayerInjangkauanSerang())
{
    stateSaatIni = EnemyState.Serang;
}
    else if (!IsPlayerInJangkauanDeteksi())
{
    stateSaatIni = EnemyState.Idle;
}
    else
{
    Vector3 arahEnemy = (pemain.position - transform.position).normalized;
    transform.position += arahEnemy * kecepatanGerak * Time.deltaTime;
}
}

  /*
    fungsi SerangState() akan dijalankan ketika enemy state berada pada state Serang
    dan mengecek apakah pemain berada pada jangkauan serang dari enemy tersebut
    jika iya maka
  */

private void SerangState()
{
    if (!IsPlayerInjangkauanSerang())
{
    stateSaatIni = EnemyState.Kejar;
}

    //masukan tambahan fungsi menyerang tambahan disini

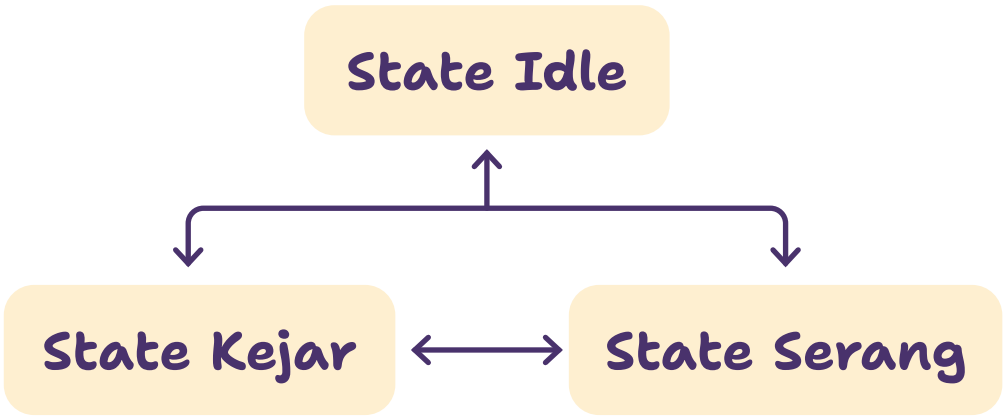
    Debug.Log("Serang Pemainnya!");
}

    //IsPlayerInjangkauanDeteksi adalah fungsi untuk mendeteksi apakah pemain
    masuk ke dalam jangkauanDeteksi dengan mengukur jarak antara posisi pemain
    dan posisi enemy dan dibandingkan dengan jangkauanDeteksi

private bool IsPlayerInjangkauanDeteksi()
{
    return Vector3.Distance(transform.position , player.position) <= jangkauanDeteksi;
}
    //IsPlayerInjangkauanSerang adalah fungsi untuk mendeteksi apakah pemain
    masuk ke dalam jangkauanSerang dengan mengukur jarak antara posisi pemain
    dan posisi enemy dan dibandingkan dengan jangkauanSerang

private bool IsPlayerInjangkauanSerang()
{
    return Vector3.Distance(transform.position, player.position) <= jangkauanSerang;
}
}
```

Output dari kode program di samping:



Gambar 4.3 Contoh State yang Merepresentasikan *Decision Tree* dalam Gim Mr. Alien
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Jadi terdapat 3 *state* untuk musuh di dalam gim Mr.Alien. Pada awalnya berada di *State Idle*, tetapi terdapat suatu kondisi di mana posisi musuh dan pemain cukup dekat maka musuh akan mengganti *state* menjadi state kejar dan mulai mengejar pemain. Jika terdapat suatu kondisi yang memenuhi, maka musuh berganti mengganti *state*nya menjadi *state* serang.

3. Stack dan Queues

Apakah kamu baru mendengar istilah *stack* dan *queues*? kamu yang baru mendengar, mungkin saja kamu pernah menggunakannya dalam membuat program, karena hampir semua program menggunakan *stack* dan *queue*.

a Stack

Gambaran tentang *stack* ini bisa kamu lihat saat kamu menumpuk piring. Hanya ada dua hal yang bisa kamu lakukan pada tumpukan piring. Kamu mengambil piring yang ada di atas atau kamu meletakkan piring di atasnya. Dalam komputer, kamu **meletakkan** sesuatu pada *stack* atau tumpukan disebut **mendorong**, sedangkan saat kamu **mengambil** sesuatu dari atas disebut dengan **mengeluarkan**. Tumpukan seperti ini dikenal dengan istilah LIFO atau *Last In First Out*. **Item yang terakhir masuk akan menjadi item yang pertama kali keluar.**

Aktivitas 4.1 **Literasi Mandiri**

Ada beberapa cara yang dapat kamu lakukan untuk menerapkan *stack* ini, di antaranya dengan *linked list* dan menggunakan *array*. coba kamu cari tau bagaimana kode programnya dan output yang dihasilkan jika menggunakan array !

Berikut ini contoh penerapan *stack* dengan *linked list* yang di beri nama **LStack**.

Kode program struktur stack

Misalkan dalam *game turn based* strategi, terdapat sebuah fitur yang memungkinkan kita untuk mengulang aksi kita sebelumnya, fitur tersebut dapat terbantu dari struktur data *Stack*, berikut contoh kodenya dalam engine Unity:

```
using System.Collection.Generic;
using UnityEngine;

  /*
    UndoSystem merupakan contoh penggunaan stack di dalam sebuah game strategi
    dimana dalam kasus ini sebuah game strategi memiliki fitur untuk mengulang kembali
    posisi pemain jika terdapat kesalahan.

    beberapa variable yang dipakai:
    positionHistory =  adalah sebuah Stack dengan tipe Vector3 (x,y,z) untuk menyimpan
                       informasi posisi pemain dalam 3 sumbu
    objectToMove    =  adalah sebuah variable yang akan menjadi referensi object pemain
                       untuk dipindahkan ke posisi sebelumnya.
  /*

public class UndoSystem : MonoBehaviour
{
    private Stack<Vector3> positionHistory = new Stack<Vector3>();
    private Transform objectToMove;

    // fungsi Start() pada unity akan dijalankan sekali di awal permainan, pada fungsi ini
    objectToMove akan disambungkan dengan object yang akan dipindahkan posisinya

    void Start()
    {
        objectToMove = GameObject.Find("MovableObject").transform;
    }

    //fungsi Update() ini akan dijalankan terus menerus sepanjang permainan, pada fungsi
    ini input pemain ketika menekan huruf "W" pada keyboard, maka posisi akan dijalankan
    dan disimpan di dalam sebuah stack bernama positionHistory, ditambah juga object akan
    berjalan keatas.
```

```
tetapi jika pemain ingin mengulang posisi, maka pemain dapat menekan huruf “U” dan
stack akan dipakai dengan fungsi Pop() , dan posisi objectToMove akan kembali ke posisi
terakhir yang ada di dalam stack.

void Update()
{
    if (Input.GetKey(KeyCode.W))
    {
        positonHistory.Push(objectToMove.position);
        objectToMove.position += Vector3.up * Time.deltaTime;
    }

    if (Input.GetKey(KeyCode.U) && positionHistory.Count > 0)
    {
        objectToMove.position = positionHistory.Pop();
    }
  }
}
```

Keuntungan menggunakan *linked list* pada *stack* adalah kamu bisa memasukkan item sebanyak yang kamu inginkan ke dalam *stack* tanpa perlu mengkhawatirkan habisnya ruang penyimpanannya.

Output dari kode program di atas:

Jika posisi pemain adalah (0, 1, 2) dan pemain menekan “W” maka pemain akan berjalan menuju (0, 1, 2) ke (0, 2, 2) karena memakai Vector2.Up.

Posisi object sekarang menjadi (0, 2, 2), tetapi ketika pemain menekan “U” maka objek akan kembali ke (0, 1, 2).

b Queue

Queue artinya antrian. Kamu pasti sering mendengar atau bahkan kamu sendiri pernah mengalami yang namanya antrian, seperti antrian membeli tiket bioskop ataupun tiket kereta api. Pada dasarnya antrian ini menggunakan struktur FIFO atau *First In First Out*, mereka yang masuk antrian pertama maka mereka yang akan keluar pertama. Sama seperti tumpukan, hanya saja antrian memiliki dua fungsi yaitu *Enqueue* dan *Dequeue*. Kamu bisa memasukkan atau menambahkan item terakhir ke dalam antrian dan kamu dapat menghapus item dari depan antrian (Ron Penton, 2003).

Penerapan *queue* ini bisa dilakukan dengan array. Berikut contoh penerapan *queue* dalam gim Mr. Alien

Kode program struktur queue dengan array:

Pada gim Mr. Alien perlu adanya sebuah pengontrol enemy yang muncul pada setiap level, maka dari itu kita memakai *queue* untuk melakukan kontrol dari enemy yang muncul. Pengontrol tersebut kita namakan EnemySpawner, berikut contoh kodenya:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class EnemySpawner : MonoBehaviour
{
  /*
    Berikut penjelasan dari variable yang digunakan:
    enemyPrefab = objek musuh yang akan ditampilkan (spawn)
    spawnPoint = posisi musuh akan dimunculkan/ditampilkan (posisi spawn)
    enemyQueue = sebuah Queue yang dipakai untuk memunculkan musuh
    spawnInterval = jarak antara dimunculkannya satu musuh dengan musuh lain


  */
    public GameObject enemyPrefab;
    public Transform spawnPoint;
    private Queue<GameObject> enemyQueue = new Queue<GameObject>();
    private float spawnInterval = 2f;

    //fungsi Start() di dalam unity akan dipanggil satu kali di awal permainan, di dalam
    fungsi ini musuh akan dibuat dengan fungsi “Instantiate” dan dimasukan kedalam queue
    untuk nantinya akan ditampilkan pada saatnya, enemy tidak diaktifkan dahulu maka dari
    itu SetActive(false) digunakan.

    Start Coroutine digunakan untuk fungsi yang membutuhkan elemen “waktu” seperti timer.

    void Start()
    {
        for(int i = 0; i < 5; i++)
        {
            GameObject enemy = Instantiate(enemyPrefab, spawnPoint.position, Quaternion.
            identity);
            enemyQueue.Enqueue(enemy);
            enemy.SetActive(false);
        }

        StartCoroutine(SpawnEnemies());
    }


    IEnumerator adalah bentuk konstruksi khusus sebuah fungsi yang membutuhkan variable
    waktu karena mempunyai beberapa fitur khusus seperti WaitForSeconds untuk timer,
    pada IEnumerator ini, jika EnemyQueuenya lebih dari 0 (ada isinya) maka enemy akan
    dimunculkan dan dikeluarkan dari queue yang disebut Dequeue, lalu diaktifkan dengan
    SetActive(true) agar dapat berinteraksi dengan objek lain di dalam proyek tersebut.
```

```
IEnumerator SpawnEnemies()
{
    while (enemyQueue.Count > 0)
    {
        GameObject enemyToSpawn = enemyQueue.Dequeue();
        enemyToSpawn.SetActive(true);
        yield return new WaitForSeconds(spawnInterval);
    }
   }

}
```

output dari kode program diatas:

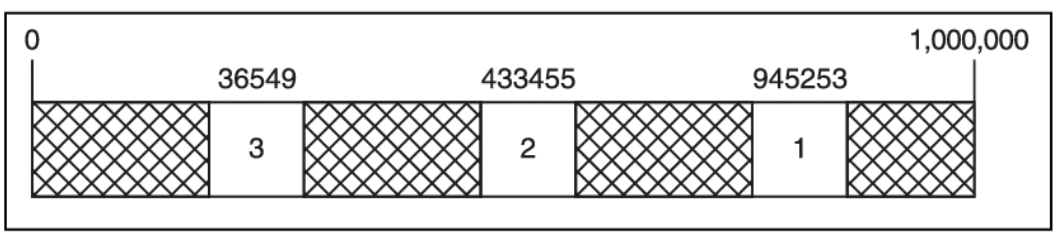
Terdapat 5 musuh yang muncul setelah gim di mainkan

4. *Hash Tables*

Ada beberapa hal yang melatarbelakangi penggunaan *hash table* yang perlu kamu ketahui. Hal pertama yang harus kamu ketahui adalah *hash table* menjadi solusi efisien untuk menangani data yang jarang (*sparse* data). Kamu bisa membayangkan sedang membuat sebuah dunia gim yang kompleks, di mana setiap pemain diakses oleh sebuah nomor identifikasi. Nomor ini sering disebut dengan kunci atau *key* dalam pemrograman. Setiap kunci unik, tidak ada pemain yang akan memiliki kunci yang sama. Namun, kunci-kunci yang dihasilkan oleh algoritma menghasilkan angka-angka acak dari 0 hingga 1.000.000. Misalnya, dalam sistem dengan tiga pemain, para pemain mungkin memiliki kunci-kunci seperti di samping.

Pemain 1: 945,253
Pemain 2: 433,455
Pemain 3: 36,549

Angka-angka ini menunjukkan data yang jarang dan saling berjauhan. Sekarang apa yang terjadi ketika kamu akan menyimpan pemain dalam struktur data agar kamu dapat dengan mudah mengakses pemain dengan kunci mereka? yang pertama kamu pikirkan mungkin menggunakan *array* agar kamu dapat mengaksesnya dengan cepat, tetapi kamu akan mendapatkan *array* seperti berikut:



Gambar 4.4 *Sparse* Data Tersimpan dalam *Array*
Sumber: Rini Melati/Kemendikbudristek (2024)

Jika kamu menyimpan dengan cara di atas maka banyak ruang yang terbuang atau kosong, hal ini menyebabkan pemborosan memori. Pilihan lainnya yang bisa kamu lakukan adalah menyimpan data dalam *linked list*. Meskipun *linked list* berfungsi dengan baik hanya untuk data dalam jumlah kecil, tapi apa yang terjadi jika permainan membutuhkan 10.000? Mencari melalui *linked list* yang berisi 10.000 pemain akan membuatmu buang-buang waktu hanya untuk menemukan satu pemain.

Melihat beberapa gambaran di atas, maka struktur data tabel *hash* bisa menjadi solusi yang tepat untuk masalah tersebut. Beberapa kemungkinan berikut ini bisa kamu lakukan, di antaranya :

1. Kamu bisa menyimpan data berbasis kunci yang jarang atau memiliki ruang kosong yang banyak dengan cepat dalam jumlah ruang yang wajar
2. Kamu juga bisa menentukan dengan cepat apakah kunci tertentu ada di dalam tabel

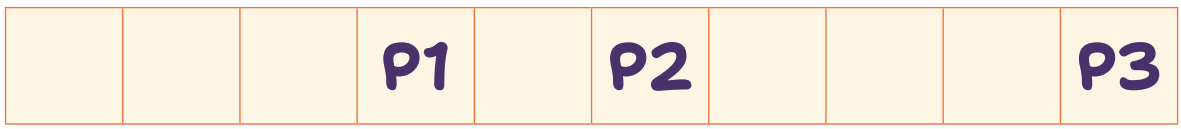
Kunci dari apa yang kamu lakukan adalah kecepatan. Untuk saat ini, kamu dapat menganggap tabel *hash* hanya sebagai *array*. Jika kamu ingin menyimpan 10 pemain, kamu harus membuat *array* dengan 10 sel. Gambar berikut ini, menunjukkan tabel *hash* yang sangat mendasar.

0	1	2	3	4	5	6	7	8	9

Gambar 4.5 Tabel *Hash* yang Menunjukkan *Array* Dasar
Sumber: Rini Melati/Kemendikbudristek (2024)

Jika kamu ingin memasukkan Pemain 1 ke dalam tabel, lalu di mana kamu akan menaruhnya? Cara termudah dan paling umum untuk menempatkan kunci ke dalam tabel *hash* adalah dengan memodulasi kunci berdasarkan ukuran tabel.

Misalnya, Pemain 1 akan ditempatkan di sel 3 karena 945253 % 10 = 3. Demikian pula, Pemain 2 akan ditempatkan di sel 5, dan Pemain 3 akan ditempatkan di sel 9. Penempatan di atas menghasilkan *array* yang terlihat seperti gambar berikut:



Gambar 4.6 Tabel *Hash* yang Menunjukkan *Array* Dasar
Sumber: Rini Melati/Kemendikbudristek (2024)

Setiap kali kamu ingin mengakses pemain dengan kuncinya, kamu cukup menggunakan algoritma sederhana seperti ini:

```
player = table[key % 10];
```

Di atas adalah contoh tabel *hash* yang sangat sederhana yang bisa kamu pahami. Kamu dapat melihat bagaimana tabel *hash* dapat menjadi alat yang tepat untuk kamu gunakan. Pada dasarnya algoritma untuk menentukan apakah suatu item ada di dalam tabel bersifat instan. Tabel *hash* yang ideal dapat mencari item dalam waktu yang cepat, yang merupakan manfaat luar biasa untuk program yang cepat.

Salah satu masalah utama dengan tabel *hash* adalah tabrakan (*collision*) sering terjadi pada tabel sederhana. Misalnya, coba masukkan kedua pemain dengan kunci ini ke dalam tabel sepuluh sel: 143.674 dan 645.394. Kamu tidak akan bisa melakukannya, karena kedua angka modulo turun hingga 4. Keduanya harus ditempatkan di sel yang sama, tetapi sel hanya dapat menampung satu item, ini disebut tabrakan. Satu-satunya cara untuk mengatasi tabrakan adalah dengan menggunakan fungsi *hash* atau memodifikasi tabel dengan cara yang membuat tabrakan dapat diterima. Penggunaan hashing tertutup bisa menjadi salah satu caranya, karena dalam hashing tertutup semua elemen akan disimpan secara langsung dalam tabel hash nya, tanpa menggunakan struktur data tambahan seperti *linked list*.

Jika fungsi *hash* yang kamu buat itu efisien dan tidak menyebabkan banyak tabrakan, maka secara teoritis kamu sudah mencapai waktu pencarian yang hampir instan. Untuk mencari data dalam tabel semacam ini, pertama-tama yang harus kamu lakukan adalah membuat *hash* pada kunci untuk menemukan indeks yang diinginkan. Setelah menemukan indeks yang benar, kamu perlu mencari di *linked list* dalam sel tersebut dan tidak di tempat lain. Jika kamu menemukan data, berarti kamu hanya mencari satu atau dua item (tergantung pada seberapa baik fungsi *hash* kamu bekerja, bisa lebih). Jika kamu tidak menemukan data, berarti kamu hanya mencari satu atau dua item. Ini lebih baik karena kamu tidak membuang waktu mencari data yang tidak ada di tabel.

contoh penulisan hash tables dalam kode program:

```
HashTable<int,int> table( size, Hash );
HashEntry<int,int>* entry;
```

Dalam program di atas akan ditanyakan ukuran tabel yang akan disimpan dalam variabel *entry*. Tabel dibuat dengan ukuran yang telah kamu masukkan dalam fungsi *Hash*. Perhatikan bahwa penunjuk ke HashEntry juga dibuat. Ini nantinya akan digunakan untuk mencari tabel.

Aktivitas 4.2 Literasi dan Praktik Mandiri

Cobalah menerapkan struktur data pada gim yang kamu buat sendiri. Kamu bisa menggunakan salah satu bahasa pemrograman yang pernah kamu pelajari. Setelah itu, lengkapi tabel berikut sesuai dengan hasil penerapan yang telah kamu lakukan.

Struktur Data	Contoh Penerapan	Tangkapan Layar	
		Kode Program	Output Program
Array			
Trees			
Stack and Queue			
Hash Table			

B. Integrasi Objek

Dalam konteks pengembangan gim, terdapat dua jenis objek yaitu objek statis dan objek dinamis. Objek statis adalah elemen yang tidak berubah posisi atau bentuk selama permainan berlangsung. Berbeda dengan objek dinamis, yang dapat bergerak atau berinteraksi secara aktif. Kamu bisa menerapkan objek statis untuk membangun lingkungan atau latar belakang gim (Meyer, 2017).



1. Jenis Objek

a Objek Statis

Objek statis memiliki beberapa ciri, di antaranya:

1. Tidak bergerak atau tidak mengubah posisi objek
2. Lebih sering digunakan untuk membuat elemen lingkungan seperti dinding, pohon, bangunan.
3. Proses pembuatan dan pengolah lebih sederhana.

Contoh Objek Statis dalam beberapa gim:

Dinding



Gambar 4.7 Dinding Sebagai Batas Ruang dan Strategi
Sumber: Separuh Interactive (2024)

Pohon dan Vegetasi



Gambar 4.8 Pohon dan Vegetasi Sebagai Rute dan Estetika
Sumber: Separuh Interactive (2024)

Bangunan



Gambar 4.9 Bangunan Sebagai Konteks dan Interaksi Minimal
Sumber: Separuh Interactive (2024)



Batu dan Rintangan

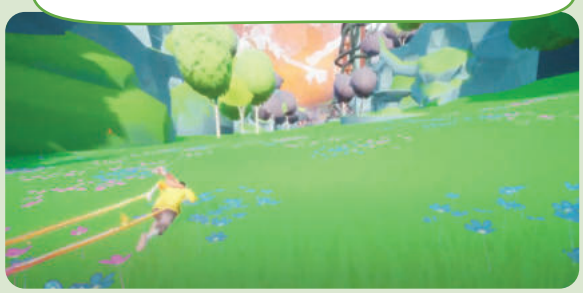
Gambar 4.10 Batu dan Rintangan Sebagai Tantangan dan Estetika
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Patung dan Monumen



Gambar 4.11 Patung dan Monumen Sebagai Konteks Budaya dan Tujuan Visual
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Jalan dan Permukaan Lain



Gambar 4.12 Jalan dan Permukaan Lain Sebagai Navigasi dan Estetika
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Furnitur dan Objek Interior



Gambar 4.13 Furnitur dan Objek Interior Sebagai Detail Lingkungan dan Pengaturan Cerita
Sumber: Separuh Interactive (2024)

Pada gim Mr. Alien di bawah ini, tampak layar biru yang merupakan latar dari gim. Latar ini merupakan salah satu objek statis dalam gim Mr. Alien. Objek statis tersebut dibuat dalam aplikasi Unity Engine. Kamu bisa membuat desain objek-objek statis seperti Mr. Alien atau contoh-contoh di atas menggunakan aplikasi desain objek yang sudah pernah kamu pelajari. Kamu juga bisa membuat kreasi objek statis mu sendiri menyesuaikan dengan tema gim yang sedang kamu kembangkan.



Gambar 4.14 Purwarupa Gim Mr. Alien dengan Objek Statis
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Aktivitas 4.3 Praktik Mandiri

Buatlah beberapa desain objek statis yang disesuaikan dengan tema gim yang akan kamu kembangkan, kemudian kamu jelaskan tentang desain tersebut!

Tangkapan Layar Desain Objek Statis	Penjelasan Tentang Desain Objek Statis

b Objek Dinamis

Selain objek statis, ada jenis objek yang sangat penting dalam pengembangan gim, yaitu objek dinamis. Objek dinamis adalah elemen yang dapat bergerak dan berinteraksi dengan pemain dan lingkungan. Contohnya, karakter, musuh, dan item yang dapat diambil. Objek dinamis ini memainkan peran kunci dalam menciptakan gameplay yang interaktif dan responsif (Davis, 2019).

Beberapa ciri dari objek dinamis, di antaranya:

1. Memiliki kemampuan untuk bergerak seperti berpindah tempat, berputar, atau berubah bentuk.
2. Dapat melakukan interaksi dengan pemain atau objek lainnya.
3. Dapat mengalami perubahan status seperti kesehatan dan posisi yang akan mempengaruhi pada *gameplay*.

Perubahan dalam objek dinamis akan membuat gim menjadi lebih menarik dan dapat memberikan tantangan bagi pemain. Berikut ini adalah beberapa contoh objek dinamis yang ada dalam beberapa gim.

Karakter Pemain



Gambar 4.15 Karakter Pemain Sebagai Interaksi Langsung dan Pengembangan Cerita
Sumber: Gamecom Team (2024)

Musuh



Gambar 4.16 Musuh Sebagai Tantangan Konflik dan AI Strategi.
Sumber: Mojiken Studio/ Toge Production (2018)

Item Interaktif



Gambar 4.17 Item Interaktif Berupa Pemotong Baut Sebagai Peningkatan Kemampuan dan Penyelesaian Teka-Teki
Sumber: Separuh Interactive (2024)

Proyektil



Gambar 4.18 Proyektil yang Berbentuk Lingkaran yang Mengenai Pemain Sebagai Interaksi Fisik dan *Gameplay* yang Dinamis
Sumber: Mojiken Studio/Toge Production (2018)

Pada gim Mr. Alien, kamu bisa menemukan beberapa objek dinamis. Lihatlah gambar di bawah, beberapa objek dinamis yang ada yaitu karakter pemain, lingkaran berwarna kuning sebagai item interaktif berupa koin dan lingkaran berwarna biru sebagai karakter musuh. Objek ini dikatakan dinamis karena ada pergerakan dan interaksi yang dilakukan yaitu Mr. Alien akan bergerak ke atas dan ke bawah atau ke depan dan ke belakang untuk mengambil koin dan menghindar dari musuh. Saat Mr. Alien menabrak koin (objek warna kuning) maka skor permainan akan bertambah 1 dan saat Mr. Alien bertabrakan dengan musuh (lingkaran biru) maka permainan akan berakhir. Objek-objek ini dibuat dalam aplikasi pembuat objek dinamis.



Gambar 4.19 Objek Dinamis Pada Gim Mr. Alien
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Masih banyak objek dinamis lainnya yang bisa kamu jadikan contoh atau referensi dalam membuat desain objek dinamis mu sendiri. Kamu bisa menyesuaikan aplikasi atau engine yang akan kamu gunakan untuk membuat objek dinamis ini. Tentu saja objek yang kamu buat bisa kamu sesuaikan dengan tema gim yang akan kamu rancang. Kamus bisa membuatnya melalui aktivitas berikut:

Aktivitas 4.4 **Praktik Mandiri**

Buatlah beberapa desain objek dinamis yang disesuaikan dengan tema gim yang kamu kembangkan pada aktivitas 4.2, kemudian kamu jelaskan tentang desain tersebut!

Tangkapan Layar Desain Objek Dinamis	Penjelasan Tentang Desain Objek Dinamis

2. Integrasi Objek Statis Gim Mr. Alien

Setelah mengetahui seperti apa objek statis dan dinamis dalam pengembangan gim, berikutnya kamu akan mempelajari bagaimana melakukan integrasi kedua objek tersebut. Integrasi ini bukan hanya tentang bagaimana mereka berinteraksi satu sama lain, tetapi juga tentang bagaimana keduanya mempengaruhi pengalaman pemain secara keseluruhan. Kamu akan mempelajari integrasi objek statis dan dinamis yang ada dalam gim Mr. Alien.

Berikut adalah contoh integrasi objek statis di dalam *game engine* unity, di mana pada contoh kali ini adalah *background* dari salah satu level di dalam gim Mr. Alien:

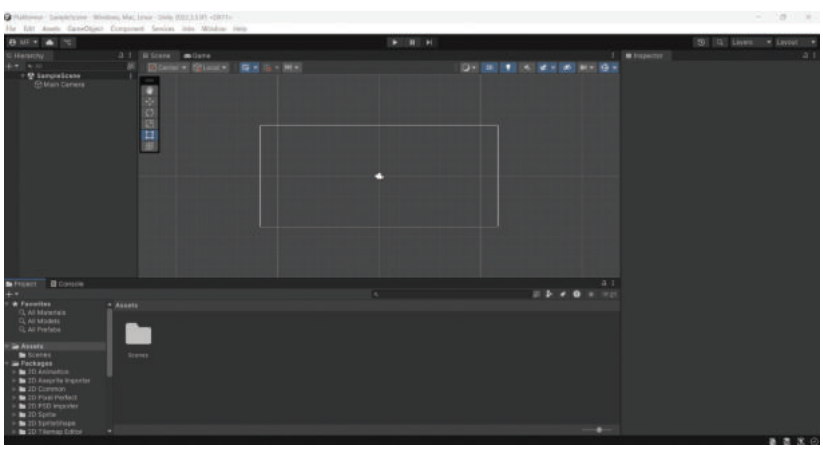
a Siapkan Gambar Latar untuk Gim



Gambar 4.20 Contoh Latar Belakang dari Gim Mr. Alien
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Berikut adalah contoh gambar latar belakang yang akan di pakai di dalam gim Mr.Alien yaitu Hutan Alien yang berwarna orange.

1. buka *game engine* dan membuka proyek gim Mr.Alien

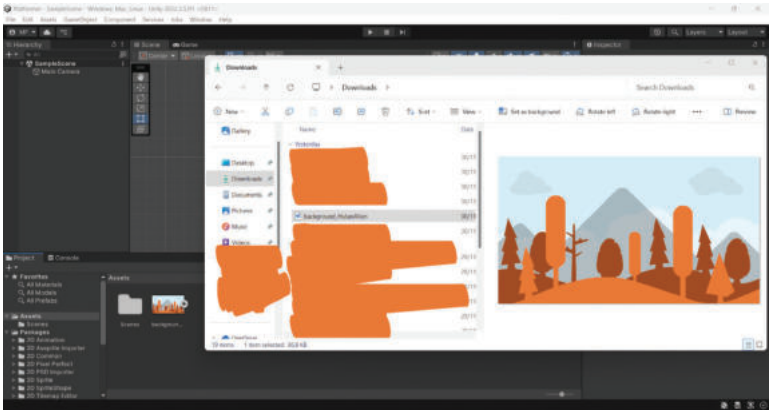


Gambar 4.21 Contoh Tampilan *Game Engine* untuk Proyek Gim Mr. Alien
Sumber Muhammad Ardhan Fadhlurrahman / 2024

berikut adalah contoh proyek unity yang dibuka, dengan scene (level) yang bernama SampleScene.

2. Terdapat dua cara untuk memasukan asset ke unity

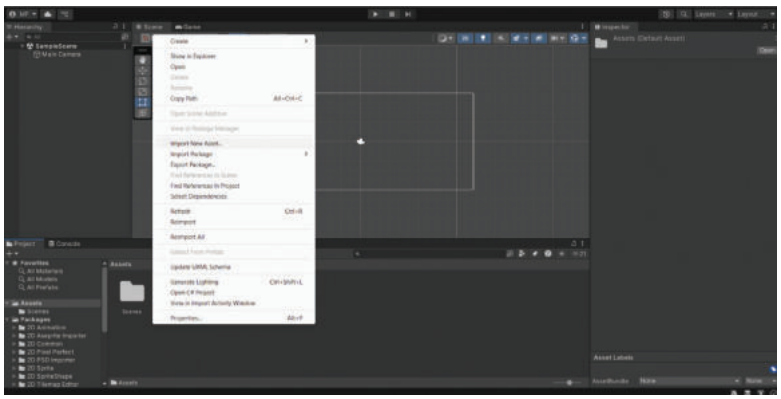
a. melakukan *drag and drop image* ke dalam jendela project



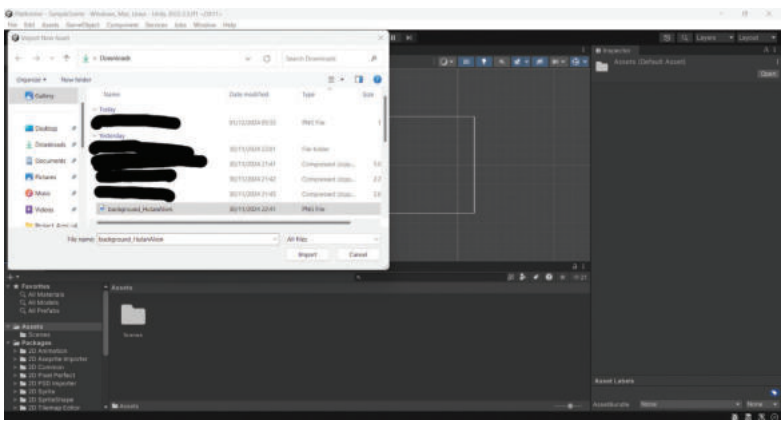
Gambar 4.22 Memasukan Gambar ke dalam *Game Engine.*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

pada gambar di atas, sebenarnya kita bisa langsung melakukan *drag and drop* kepada gambar latar belakang ke dalam jendela "proyek" dan sistem akan otomatis memasukan gambar ke dalamnya.

b. memasukan asset dari menu *impor asset* dalam jendela proyek

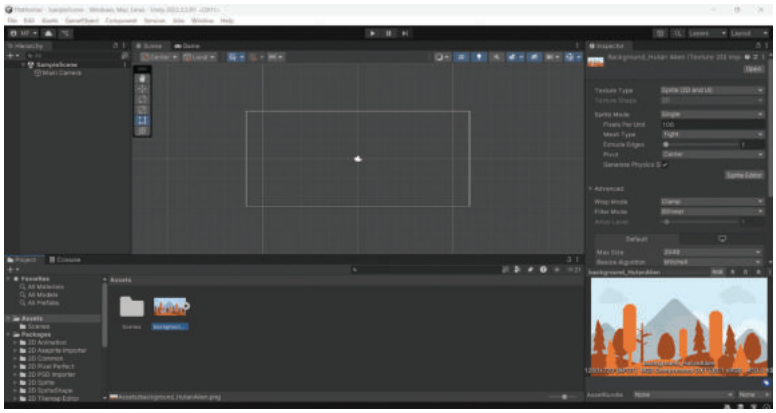


Gambar 4.23 Cara Memasukan Aset dari Menu Impor Asset dalam Jendela Proyek.
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)



Gambar 4.24 Cara Memasukan Aset dari Menu Impor Asset dalam Jendela Proyek.
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

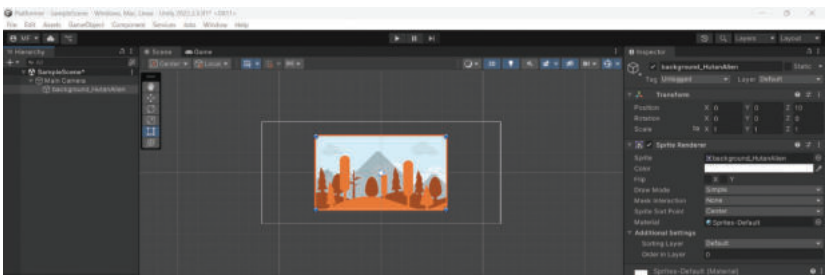
c. cara kedua adalah dengan menu "*impor New Asset*", memilih gambar mana yang akan dimasukan dan memasukan gambar ke dalam proyek seperti gambar di bawah ini



Gambar 4.25 Contoh Gambar Latar Telah Masuk ke dalam Proyek
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

3. Masukan gambar background ke dalam scene hierarchy.

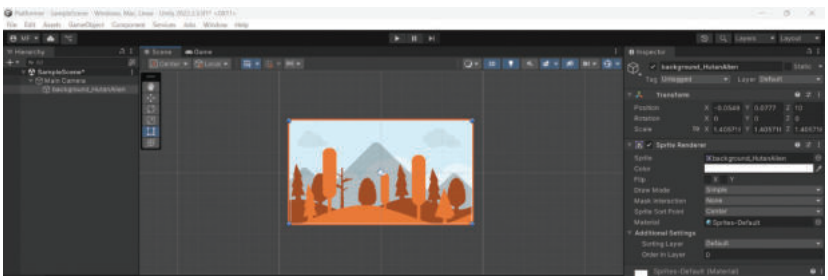
Setelah gambar berada di dalam proyek, maka kita harus memasukan gambar tersebut ke dalam level.



Gambar 4.26 Contoh Gambar Latar Telah Masuk ke dalam *Hierarchy* Proyek
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

4. Setting inspector untuk melakukan pengaturan tambahan di dalam scene.

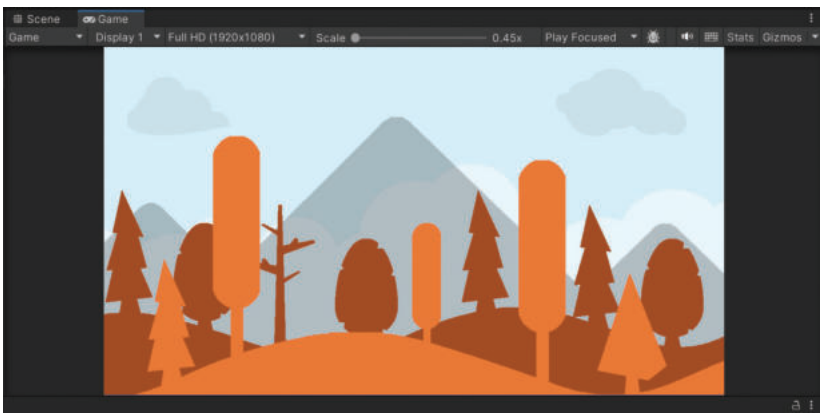
Setelah gambar masuk ke dalam scene, maka kita bisa mengedit skala, posisi, rotasi maupun setting lainnya di dalam inspector, dalam hal ini kita merubah skala dari gambar latar belakang tersebut.



Gambar 4.27 Gambar Latar yang Telah Diberikan Input Tambahan dalam *Inspector*.
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

5. Test hasil akhir

berikut adalah hasil akhir gambar pada game dimana latar belakang sudah dipasangkan



Gambar 4.28 Contoh Gambar Latar yang Sudah Berfungsi di dalam Proyek dan Berjalan dalam *Runtime*.
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

3. Integrasi Objek Dinamis Gim Mr. Alien

Kamu dapat mencoba mengintegrasikan objek dinamis menggunakan aplikasi yang kamu ketahui. Contoh kali ini, kita akan membahas bagaimana mengintegrasikan objek dinamis di dalam *game engine* Unity. Kita akan menggunakan objek uang, dompet, dan koin di dalam gim Mr. Alien:

a Siapkan item yang akan dikumpulkan oleh mr.alien

Mr. Alien dapat mengumpulkan 3 jenis item dengan cara menyentuh objek tersebut. Item-item yang dikumpulkan yaitu Dompet, Uang, dan Koin. Item-item tersebut memiliki besaran masing-masing. Dompet bernilai 10 koin, Koin bernilai 1 koin dan Uang bernilai 5 koin.


Sprite_Dompet

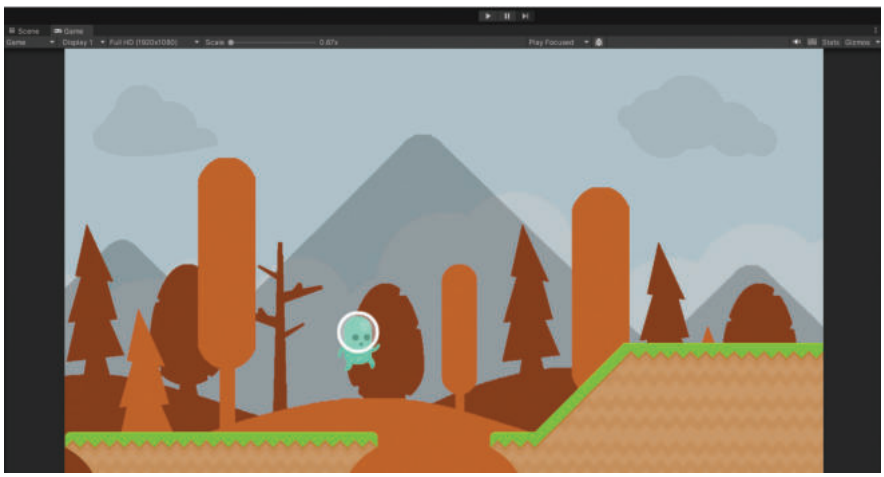

Sprite_Koin


Sprite_Uang

Gambar 4.29 Contoh Gambar yang Dipakai di dalam Mr. Alien
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

b Membuka level gim Mr.Alien yang menjadi target level objek yang ingin dimasukan.

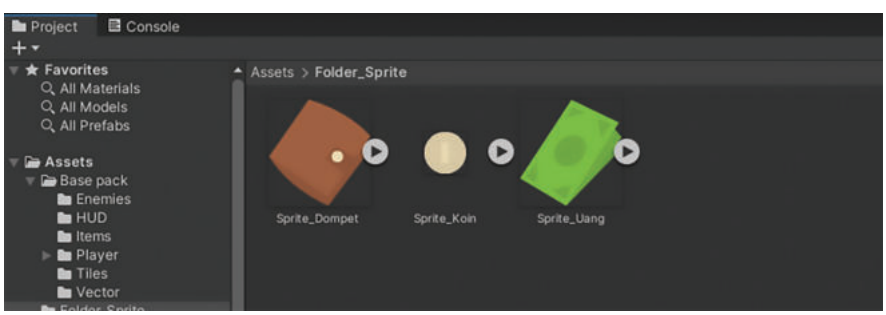
Berikut adalah contoh salah satu level dari gim Mr.Alien tanpa objek item dimana tidak terdapat objektif yang menyebabkan gim menjadi tidak memiliki tujuan apapun sehingga gim menjadi tidak menyenangkan. Karena hal tersebut maka perlu ditambahkan objektif inti dari gim mr.Alien, yaitu mengumpulkan uang yang ditaruh secara acak di dalam gim.



Gambar 4.30 Contoh Salah Satu Level di dalam Mr. Alien
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

C Memasukan objek yang diperlukan *game engine*

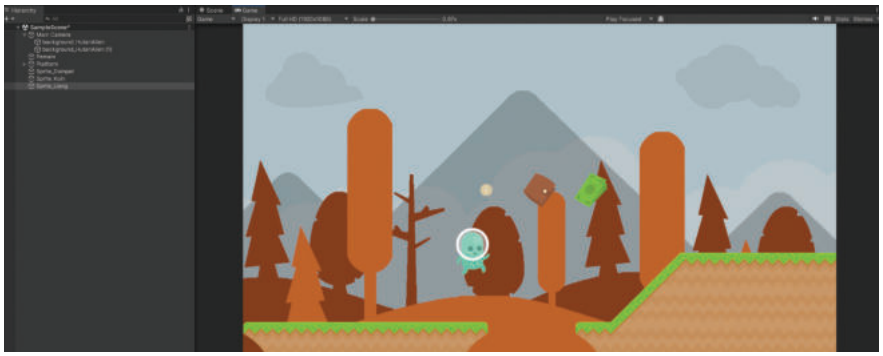
Seperti yang telah dijelaskan di bagian objek *static* sebelumnya, kita melakukan hal yang sama sehingga objek tersebut sudah terdapat dalam *game engine* seperti gambar berikut:



Gambar 4.31 Objek Item Sudah Masuk ke Dalam *Game Engine*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

d Memasukan objek item ke dalam *scene hierarchy*

Memasukkan objek ke dalam Scene *Hierarchy* agar objek dapat masuk ke dalam *scene* (level)



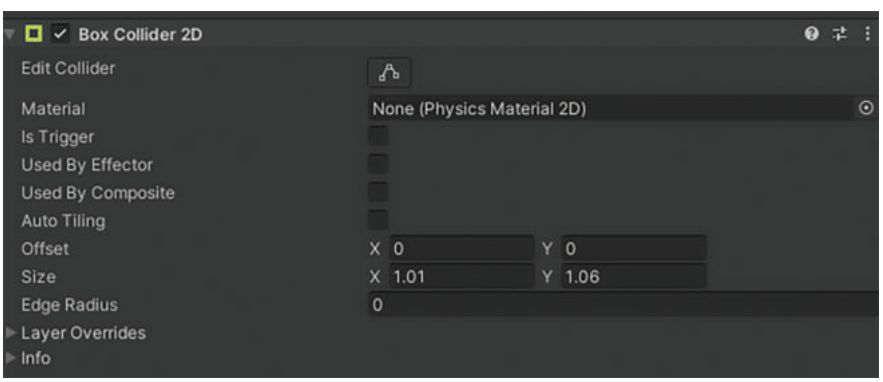
Gambar 4.32 Objek Item Sudah Masuk ke dalam *Hierarchy* di *Game Engine.*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

Pada langkah ini objek masih sebagai objek statis karena tidak memiliki komponen apapun yang dapat berinteraksi dengan pemain. Maka, ketika pemain menyentuh objek tersebut tidak akan terjadi apa-apa karena belum terdapat interaktivitas pada item tersebut.

e Memasukan komponen yang diperlukan untuk membuat objek statis menjadi objek dinamis.

Collider

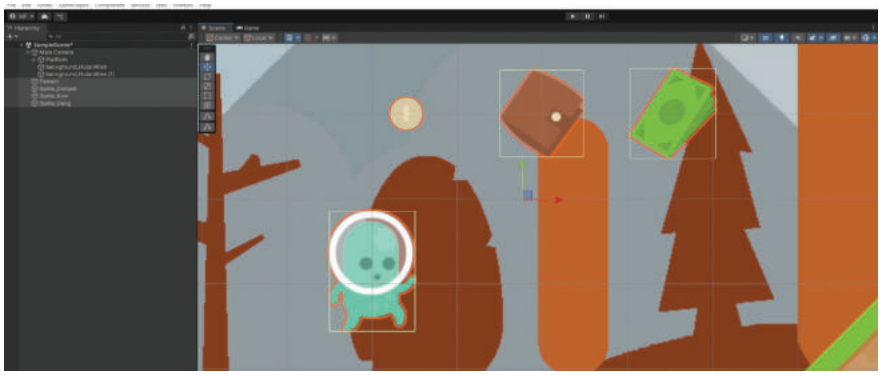
Collider adalah sebuah komponen di Unity yang berguna untuk mendefinisikan bentuk agar dapat mendeteksi *overlap* atau bersentuhan dari satu objek dengan objek lainnya. Berikut adalah contoh komponen *Collider* bernama *Box Collider 2D* dalam *GameObject* di engine Unity (Unity Documentation, 2024):



Gambar 4.33 Penjelasan Mengenai Komponen *Box Collider 2D*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

f Menambahkan script agar interaksi sesuai dengan yang diinginkan

Ketika object ditambahkan sebuah *collider*, maka di dalam *scene* akan terdeteksi menjadi seperti di bawah ini:



Gambar 4.34 Contoh Box Collider Setiap Item di Gim *Mr. Alien*.
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

Terdapat kotak warna hijau di sekitar objek pemain, serta beberapa objek item lain (kecuali koin yang berbentuk bulat). Kotak atau lingkaran tersebut merupakan aspek *collider*. Dalam sistem *game engine* tidak mengenai visual seperti gambar pemain atau gambar uang, sistem *game engine* hanya mengenali itu sebagai suatu objek kotak maupun bulat saja. Lalu apa yang terjadi setelah diberikan *collider*? Objek pemain akan menabrak item tanpa terjadi apa-apa. Lalu apa yang harus kita lakukan? kita harus menambahkan *script* untuk memasukan logika permainan atau melakukan implementasi mekanika dari gim Mr.Alien.

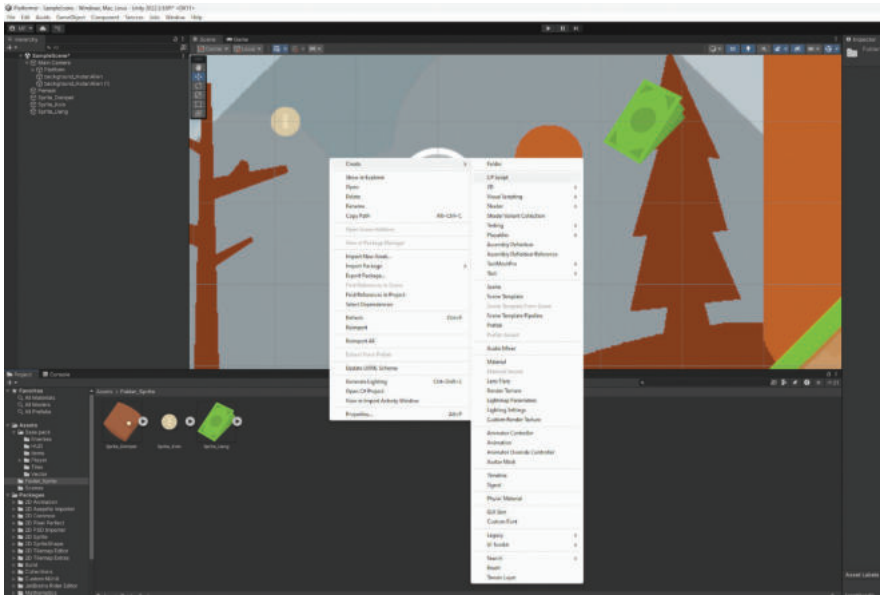
Pada Game Design Document (GDD) gim Mr. Alien, terdapat penjelasan mengenai kondisi saat karakter Mr. Alien menyentuh berbagai item seperti koin, dompet, maupun uang dengan aturan sebagai berikut:

1. Jika pemain menyentuh Koin, maka uang pemain akan bertambah 1.
2. Jika pemain menyentuh Tumpukan Uang, maka uang pemain akan bertambah 5.
3. Jika pemain menyentuh Dompet, maka uang pemain akan bertambah 10.

Untuk mengimplementasikan aturan tersebut, kita harus membuat *script* yang mengakomodasi semua aturan di atas. Berikut langkah-langkahnya:

1. Membuat Script baru bernama Mekanik_Koin di dalam Game Engine.

Agar dapat sebuah fungsionalitas berjalan di dalam sebuah game, kita akan membuat sebuah fungsi baru yang akan merepresentasikan aturan yang telah dijelaskan di atas. Pertama, kita harus membuat *script* baru bernama "*Mekanik_Koin*" dengan cara seperti di bawah.



Gambar 4.35 Cara membuat Skrip Baru di dalam *Game Engine*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

setelah dibuat kita akan menamakannya sebagai skrip Mekanik_Koin:



Gambar 4.36 Skrip Baru Telah Dibuat dan Dinamakan Sesuai Kebutuhan
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

2. Membuat fitur dengan bantuan OnCollisionEnter2D didalam Script Mekanik_Koin.

Setelah file script "Mekanik_Koin" dibuat, maka perlu dilakukan implementasi fungsi dengan menggunakan komponen Collider. Ketika kita meng-click script mekanik koin, maka akan muncul 2 fungsi yang akan dibuat secara otomatis, yaitu Start() dan Update(), dengan penjelasan:

Nama Fungsi	Penjelasan
Start()	Fungsi akan dijalankan di awal permainan dan hanya sekali
Update()	Fungsi akan dijalankan berulang sepanjang permainan

```
Mekanik_Koin.cs
Miscellaneous Files                    Mekanik_Koin
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class Mekanik_Koin : MonoBehaviour
6   {
7       // Start is called before the first frame update
8       void Start()
9       {
10
11      }
12
13      // Update is called once per frame
14      void Update()
15      {
16
17      }
```

Gambar 4.37 Cuplikan Kode pada skrip Mekanik_Koin
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Tetapi untuk menggunakan komponen Collider terdapat 3 fungsi tambahan yang bisa digunakan dalam game 2D, seperti:

Nama Fungsi	Penjelasan
OnCollisionEnter2D()	Fungsi akan dijalankan ketika pertama kali kedua objek tersebut mulai bertabrakan
OnCollisionExit2D()	Fungsi akan dijalankan setelah pertama kali kedua object yang bertabrakan sudah tidak bertabrakan.
OnCollisionStay2D()	Fungsi akan terus dijalankan secara berulang ketika object sedang bertabrakan

```
Unity Message | 0 references | Gpt
private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.name == "Dompet")
    {
        skorPemain += 10;
        UI_SkorPemain.text = " " + skorPemain;
    }
    else if (collision.gameObject.name == "Tumpukan Uang")
    {
        skorPemain += 5;
        UI_SkorPemain.text = " " + skorPemain;
    }
    else if (collision.gameObject.name == "Koin")
    {
        skorPemain += 1;
        UI_SkorPemain.text = " " + skorPemain;
    }
}
```

Dengan spesifikasi GDD yang telah dijelaskan sebelumnya, maka kita akan memakai fungsi OnCollisionEnter2D() dalam kasus ini dengan screenshot seperti di samping.

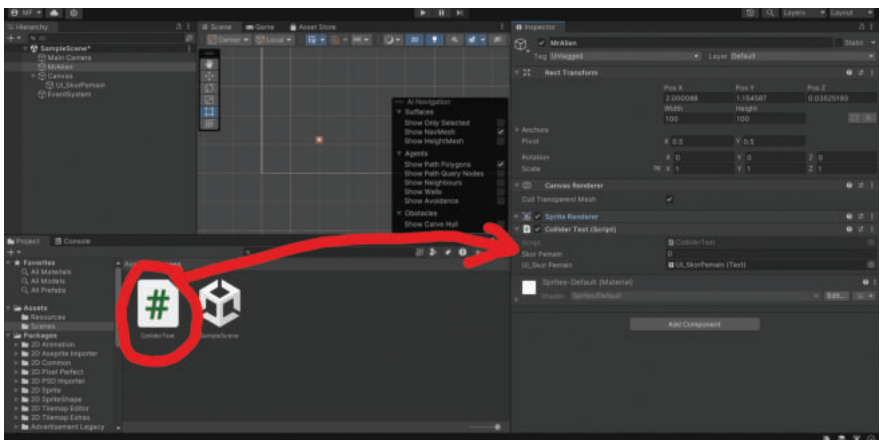
Gambar 4.38 Fungsi *OnCollisionEnter2D* untuk Menggunakan Fitur *Collider*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

Berikut penjelasan singkat pada potongan kode tersebut adalah:

Di dalam fungsi OnCollisionEnter2D terdapat sebuah variable Collision2D bernama collision yang memiliki tujuan sebagai entitas object yang bertabrakan dengan objek yang memiliki skrip ini, pada kasus ini player yang memiliki skrip ini sehingga, variable collision menyimpan informasi objek terakhir yang bertabrakan dengan pemain. Lalu akan di check apakah collision tersebut adalah Dompet, Tumpukan Uang, atau Koin dengan kondisinya masing2.

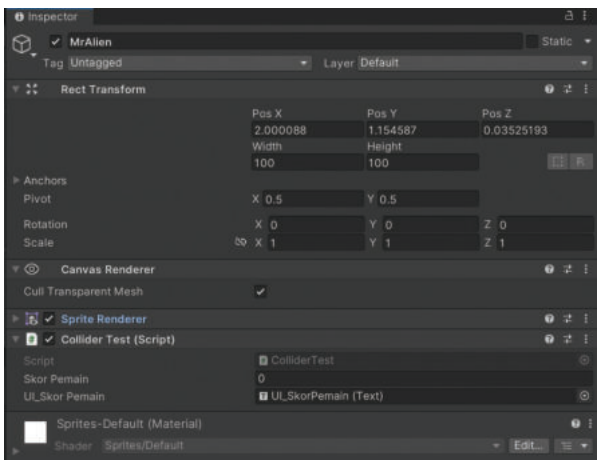
3. Memasukan Komponen Script Mekanik_Koin ke dalam objek "Koin".

Setelah fitur di implementasikan, maka kalian perlu memasang script tersebut ke dalam game objek, caranya adalah dengan menarik (drag) file script ke dalam jendela inspector seperti di bawah ini:



Gambar 4.39 Cara Menarik File *Script* ke dalam *Inspector*.
Sumber: Muhammad Ardhan Fadhlurrahman / Kemendikbud (2025)

Gambar di samping adalah contoh hasil object Mr.Alien di dalam jendela *Inspector* setelah file script ditarik.

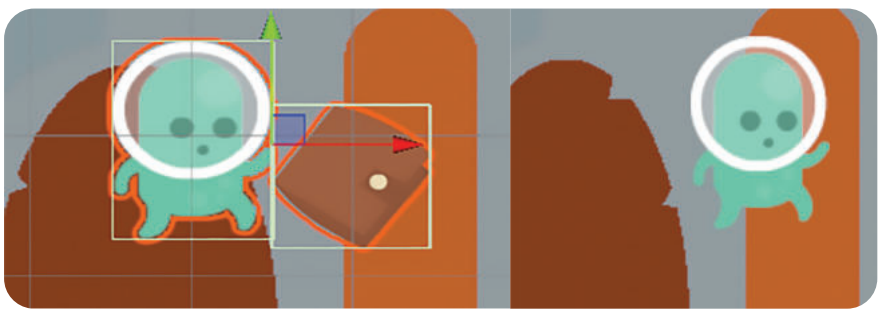


Gambar 4.40 Tampilan Inspector setelah *Script* dimasukan.
Sumber: Muhammad Ardhan Fadhlurrahman / Kemendikbud (2025)

4. Melakukan testing komponen objek "Koin".

Tahap terakhir, lakukan tes hasil akhir pada gim Mr. Alien. Jika *script* berhasil, maka Mr. Alien akan menyentuh dompet dan koin pemain akan menambah menjadi 10 lalu dompet akan hilang dari level. Objek dompet ini dinamakan objek dinamis karena dapat berinteraksi dengan pemain. Jika tes hasil akhir tidak terjadi interaksi, adapun hal yang bisa di cek dan diperbaiki seperti:

1. Komponen Collider tidak terpasang ke dalam objek.
2. Komponen Skrip tidak terpasang ke dalam objek.
3. Komponen pengecekan nama pada OnCollisionEnter2D memiliki kesalahan pengetikan pada target objek.



Gambar 4.41 Dompet Telah Menjadi Objek Dinamis
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

Aktivitas 4.5 Kerja Tim

Bentuklah sebuah tim yang terdiri dari 3 orang. Kemudian buatlah sebuah integrasi desain objek dinamis dan desain objek statis sesuai dengan tema gim yang akan kalian buat. Setelah itu carilah dan catat kesalahan atau kendala yang kalian alami selama praktik tersebut. Lengkapilah tabel berikut dengan menyebutkan temuan kalian terkait kesalahan atau kendala yang terjadi.

Tangkapan Layar	Kendala/Kesalahan	Solusi

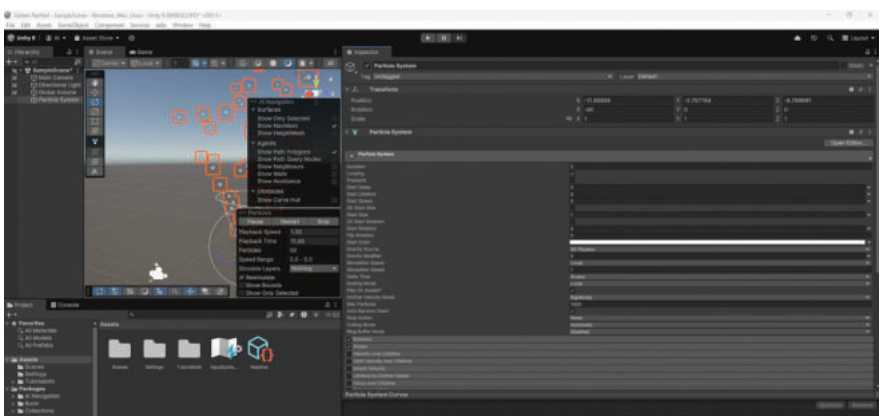
Lakukan presentasi di depan kelas, untuk bisa saling berbagi informasi dengan teman lainnya.

C. Fungsionalitas Tambahan



Selain melakukan integrasi objek statis dan dinamis, terkadang terdapat sebuah fungsionalitas yang perlu ditambahkan di dalam *game design* baik *tools* maupun *plugins*. Fungsionalitas tambahan pada *game engine* mencakup berbagai fitur dan alat yang dirancang untuk memperluas kemampuan dasar *engine*, membantu pengembang dalam menciptakan, mengelola, dan mengoptimalkan gim yang kamu kembangkan. Berikut ini kamu bisa mempelajari lebih jelas tentang bagaimana melakukan implementasi penambahan fungsionalitas pada *game engine* Unity, lengkap dengan contoh kodenya.

1. Kustomisasi Parameter



Pada panel **Inspector**, atur parameter seperti **Start Lifetime**, **Start Size**, dan **Emission Rate**.

Gambar 4.42 Tampilan Inspector untuk Mengganti *Variable Particle System*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

2. Membuat Skrip Interaksi

Buat file C# baru bernama UIController.cs.

Contoh Kode untuk UI

UIController.cs

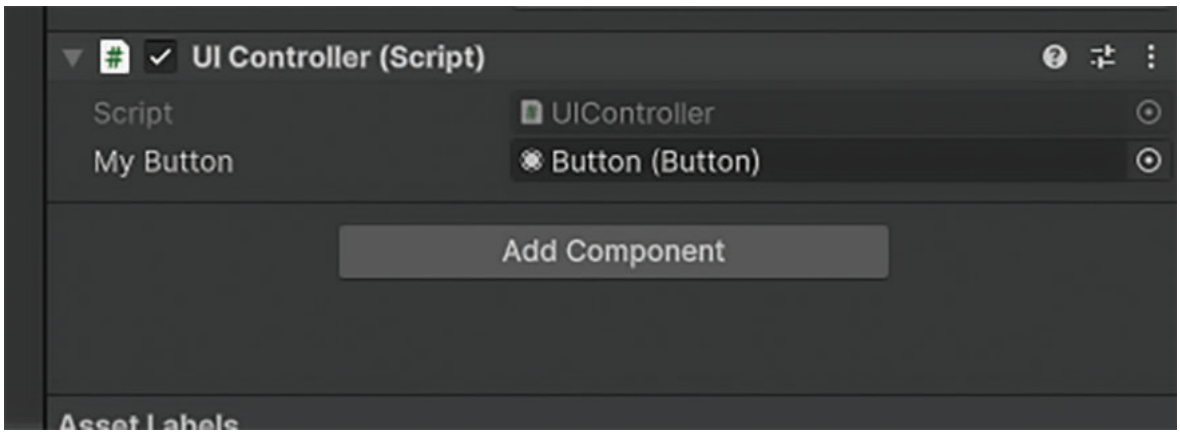
Miscellaneous Files — UIController

```
4   public class UIController : MonoBehaviour
5   {
6       public Button myButton;
7       // Start is called once before the first execution of Update after the MonoBehaviour is created
8       void Start()
9       {
10          myButton.onClick.AddListener(OnButtonClick);
11      }
12
13      void OnButtonClick()
14      {
15          Debug.Log("Button Clicked!");
16      }
17
18      // Update is called once per frame
19      void Update()
20      {
21
22      }
23  }
24
```

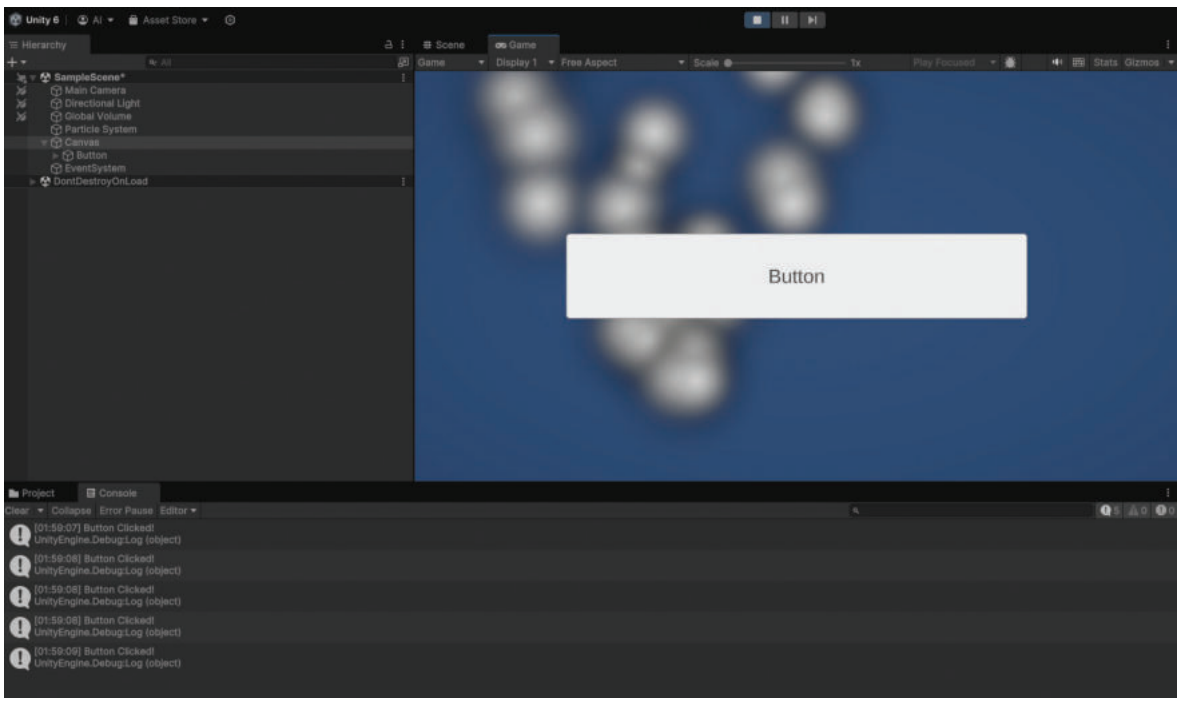
Gambar 4.43 Contoh Kode Implementasi UI
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Cara Menggunakan:

1. Tambahkan *script* ini ke objek kosong di dalam Canvas.
2. Seret tombol yang telah dibuat ke dalam variabel myButton di inspector.

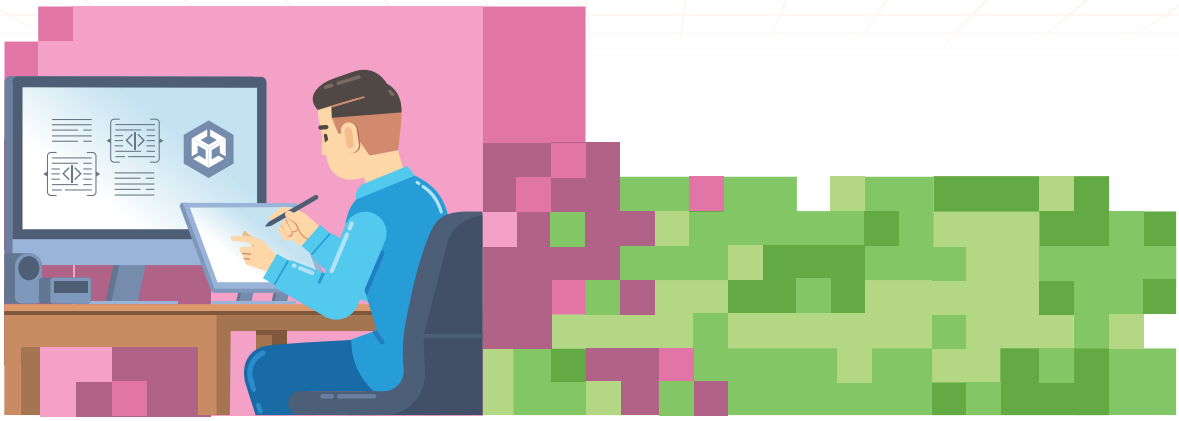


Gambar 4.44 Contoh Memasukan *Script* UI di dalam *Button*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)



Gambar 4.45 Contoh Output setiap Klik Tombol akan muncul tulisan *Button Clicked*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

D. Pemrograman Berbasis Teks dan Grafis



Dalam membuat gim, kamu akan berkenalan dengan pemrograman berbasis teks dan grafis. Pemrograman teks membantumu menulis kode untuk mengatur bagaimana gim berjalan, sedangkan pemrograman grafis menghidupkan tampilan visual yang menarik. Di bagian ini, kamu akan belajar menggunakan *game engine* sebagai alat utama untuk membangun gim, merancang gameplay agar permainan terasa seru, dan membuat *User Interface* (UI) yang memudahkan pemain berinteraksi dengan dunia gim yang kamu buat.

1. *Game Engine*



Setelah memahami beberapa struktur data dalam pemrograman gim, kamu bisa mencoba melakukan implementasi sederhana menggunakan *game engine*. Gunakan *game engine* yang kamu pahami dan sesuaikan dengan materi yang kamu pelajari saat ini. Pada materi berikut, kamu akan diajak mengenal pengaturan awal (*setup)* dan fungsi-fungsi dasar dari *game engine* Unity. Selain itu, kamu juga akan mempelajari komponen-komponen yang bisa kamu gunakan untuk bisa merancang sebuah gim yang ada dalam Unity. Tujuan akhirnya agar kamu dapat memahami bagaimana input yang kamu buat bisa dijalankan di dalam sistem sehingga menghasilkan sebuah pengalaman gim yang menarik.

Ada beberapa istilah penting yang perlu kamu ketahui dalam *game engine*. *Game engine* yang digunakan disini adalah **Unity**. Beberapa istilah mungkin ada yang sama dengan *game engine* lain yang pernah kamu pelajari atau gunakan.

Istilah Dalam Game Engine

a Proyek

Saat kamu ingin memulai pengembangan gim dan memulai proyek baru di dalam *game engine*, kamu akan diminta beberapa informasi sebelum proyek dimulai, seperti:

1. **Nama Proyek**

   Nama proyek gim ini mencakup nama judul gim ataupun hal lain untuk melakukan identifikasi proyek dengan mudah.

2. **Target Platform**

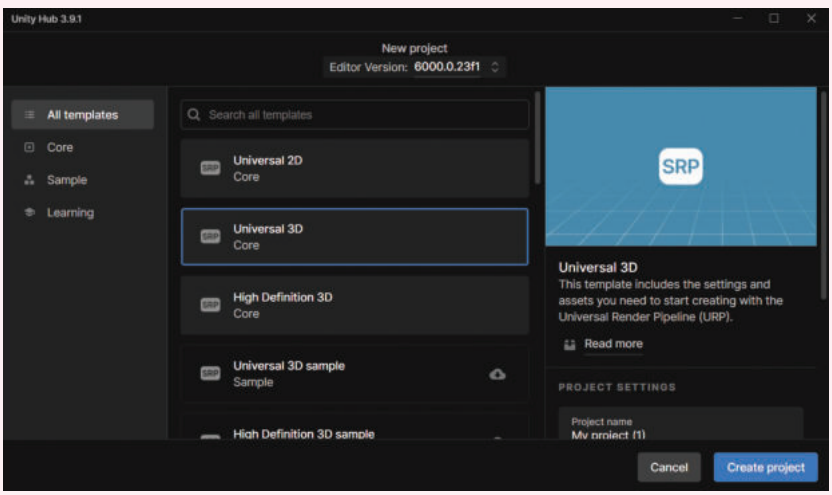
   Target platform biasa digunakan untuk menentukan *layout*, *framework*, maupun hal lain yang diperlukan untuk target platform tersebut.

3. **Target Lokasi Penyimpanan Proyek**

   Target Lokasi Penyimpanan Proyek adalah lokasi yang digunakan di dalam perangkat komputer.

4. **[*Optional*] Template yang digunakan**

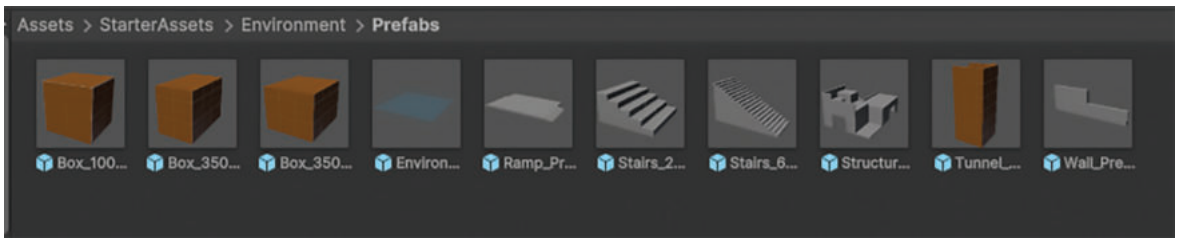
   Pada beberapa *game engine*, terdapat template yang dapat dipilih untuk dapat digunakan agar pengembangan menjadi lebih cepat.



Gambar 4.46 Contoh Pembuatan Proyek Baru di dalam *Game Engine Unity*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

b Aset (*Asset*)

Aset adalah sebuah representasi dari item yang digunakan di dalam gim atau proyek yang kamu buat (Unity Documentation, 2024) . Aset dapat berbentuk 3D model, berkas audio, *visual effects*, animasi ataupun gambar. Berikut ini adalah contoh UI siap pakai di Unity.



Gambar 4.47 Contoh Tampilan Jendela *Asset* di dalam *Game Engine*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

c *Game Object* atau *Actor*

GameObject merepresentasikan setiap komponen yang ada di dalam *scene*. *GameObject* merupakan bagian yang membangun sebuah *scene* di dalam Unity dan menjadi tempat bagi berbagai komponen (*component*). Komponen tersebut akan menentukan bagaimana sebuah *GameObject* berfungsi dan berperilaku di dalam *scene*.

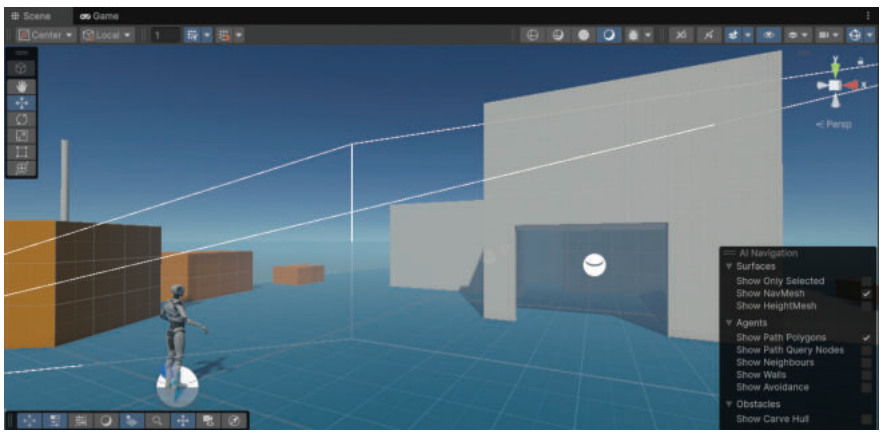


Gambar 4.48 Contoh *Game Object* di dalam G*ame Engine.*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Pada bagian kiri gambar, terdapat **Hierarchy** UI yang menampilkan daftar dari *GameObject* dalam *scene*. Dalam contoh ini, *GameObject* bernama *PlayerArmature* dipilih. Di bagian kanan, tampak *Inspector*, yaitu panel yang menunjukkan daftar komponen yang melekat pada *GameObject* tersebut. Pada contoh ini, *PlayerArmature* terdapat beberapa komponen utama yaitu *Transform, Animator,* dan *Character Controller.*

d *Scene* atau *Level*

Scene merupakan salah satu nama yang dipakai untuk merepresentasikan level di Unity. Jika kamu memakai Unreal Engine nama yang digunakan adalah *Maps*. Sebuah gim dapat terdiri dari beberapa *scene* yang digabungkan menjadi satu.



Gambar 4.49 Contoh Sebuah *Scene* di dalam *Game Engine*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

e Komponen *(Component)*

Component merupakan sebuah elemen yang mendefinisikan perilaku dari sebuah *GameObject*. Jenis-jenis komponen yang terdapat pada *game engine* Unity, seperti:

1 Audio

Jenis komponen yang mengatur ruang lingkup audio, seperti bagaimana suatu *GameObject* dapat mengeluarkan suara, dan kontrol terhadap file suara yang ada di dalam suatu permainan.

2 Effect

Jenis komponen yang mengatur ruang lingkup *effect* seperti visual *effect* maupun bagaimana visual *effect* tersebut terbentuk secara spesifik.

3 Event

Jenis komponen yang mengatur ruang lingkup *event* di dalam *game engine*. *Event* adalah sistem yang dapat membuat antar *script* melakukan komunikasi satu sama lain ataupun bagaimana script dapat di-*trigger* berdasarkan suatu deteksi visual dari suatu sistem. Misalkan, object UI diklik oleh *user* maka sistem akan mendeteksi klik tersebut dengan komponen bernama *Event Trigger* yang akhirnya menjalankan suatu fungsi UI tersebut.

4 Input

Jenis komponen yang mengatur ruang lingkup input dari pemain. Misalkan, mengatur fungsi tombol, ataupun melakukan manajemen kompleks untuk masing-masing pemain.

5 Layout

Jenis komponen yang mengatur beberapa jenis *layout* UI dari sebuah gim, umumnya diletakan pada *GameObject* untuk mengatur *layout* dari sebuah UI pada sebuah gim.

6 Mesh

Jenis komponen yang mengatur *mesh* dari sebuah *GameObject*. *Mesh* adalah gabungan dari *texture* yang merepresentasikan visual dari sebuah *GameObject*.

7 Miscellaneous

Jenis komponen spesifik yang tidak dapat dilakukan *grouping*, seperti komponen animasi pada gim di engine Unity. Komponen lain dalam *miscellaneous*, seperti *grid, look at constraint, parent constraint,* dan lain-lain.

8 Navigation

Jenis komponen spesifik untuk ruang lingkup kecerdasan buatan (*Artificial Intelligence*) dalam melakukan navigasi di dalam gim yang sedang dibuat dan pengembang gim dapat memakai berbagai jenis navigasi yang telah disediakan.

9 Physics

Jenis komponen untuk ruang lingkup sistem fisika di dalam gim. Sistem fisika ini dibagi menjadi beberapa komponen seperti *Collider* dan *Rigidbody*.

- *Collider* merupakan sebuah komponen untuk mendeteksi bentuk dari sebuah object, di mana akan dipakai untuk mendeteksi apakah object tersebut bertabrakan atau beririsan dengan object lain dan sistem akan menggunakan untuk melakukan *trigger* dalam menjalankan fungsi.
- *Rigidbody* merupakan sebuah *component* yang membuat sebuah GameObject dapat memiliki komponen fisika seperti gravitasi, gesekan, maupun komponen fisika lainnya yang dapat disimulasikan di dalam gim.

10 Rendering

Jenis *Component* untuk ruang lingkup *Rendering*, atau proses bagaimana sistem memproses visual dari sebuah gim kepada pemainnya. Seperti *component Occlusion* yang dapat membuat sistem hanya melakukan *Rendering* Object yang terlihat oleh pemain dan mematikan *Rendering* Object yang tidak terlihat oleh pemain.

11 Scripts

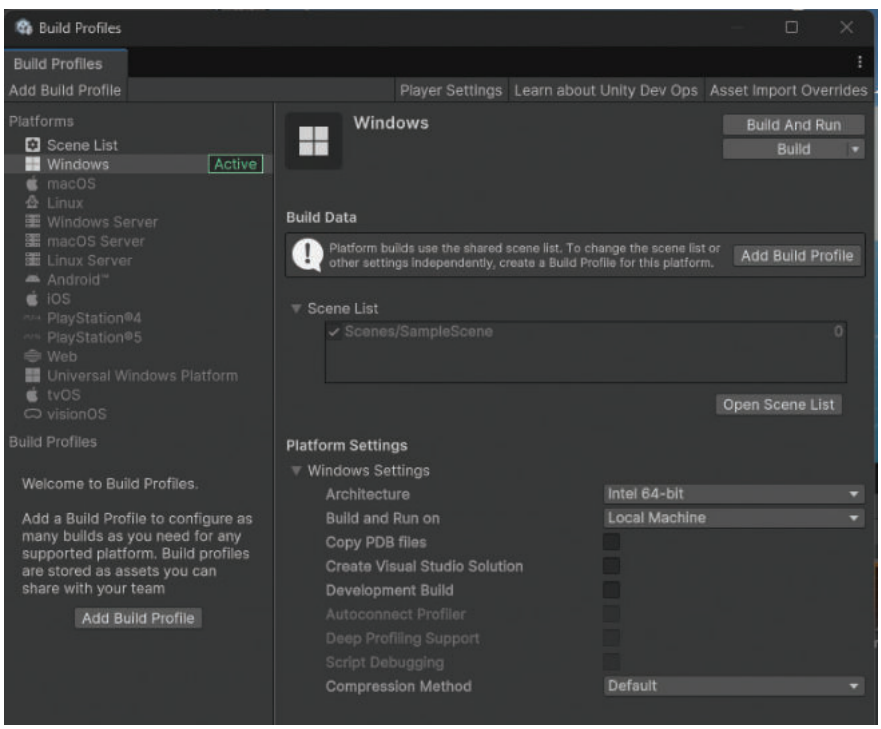
Jenis *Component* untuk ruang lingkup *Scripting*, yaitu komponen pemrograman dari sebuah GameObject.

12 UI

Jenis *Component* untuk ruang lingkup UI yang merupakan komponen dari sebuah Canvas / Layout UI. Disini kamu bisa menemukan Button, Text, Gambar yang dapat kamu susun untuk gim yang kamu buat.

Build

Build adalah sebuah istilah yang merepresentasikan hasil akhir dari pengerjaan yang sudah dapat dimainkan. Pada *game engine* Unity terdapat sebuah UI spesifik untuk mengatur *Build* jika memang pengembangan sudah selesai dilakukan, berikut contohnya:



Gambar 4.50 Jendela *Build Profile* di dalam *Game Engine*.
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

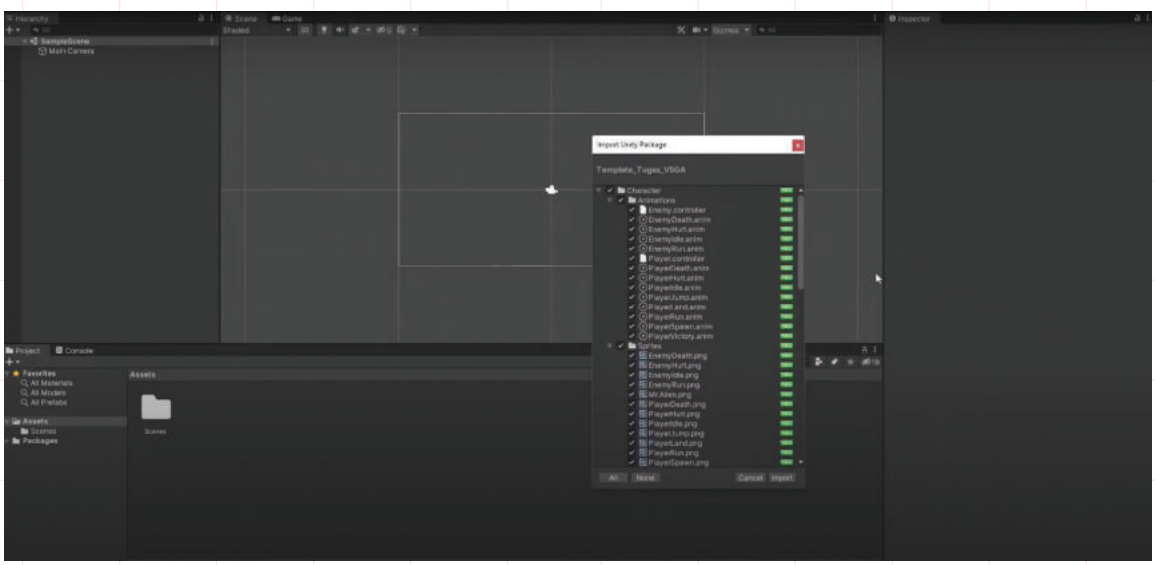
Pada UI *Build* Profile ini, kamu dapat memilih target platform jika ingin mengganti target platform dari yang sebelumnya telah dipilih. Setelah itu, kamu bisa memastikan scene apa saja yang akan dimasukkan ke dalam build. Langkah terakhir adalah melakukan pengaturan di Platform Settings sesuai dengan platform yang dituju atau ditargetkan.

Pembuatan Purwarupa Mr. Alien

Setelah mengetahui beberapa komponen dalam *game engine*, materi berikutnya akan mempelajari proses pembuatan purwarupa Mr. Alien di dalam *game engine* Unity seperti yang dijelaskan pada bagian berikut ini:

1. Gim baru dibuat dengan nama yang sesuai atau mewakili ide gim. Perlu kamu ingat, saat membuat nama gim, tidak hanya mengenai organisasi file yang baik tapi juga identitas dari gim itu sendiri.

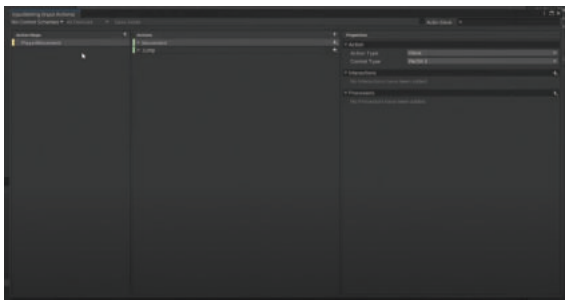
2. Di dalam Unity, kamu bisa melakukan impor aset pembantu yang dibutuhkan. Aset pembantu dalam gim Mr. Alien seperti animasi, *sprite* dan lain sebagainya. Proses ini bukan sekedar mengimpor file, tetapi kamu perlu melakukan pengecekan isi aset.



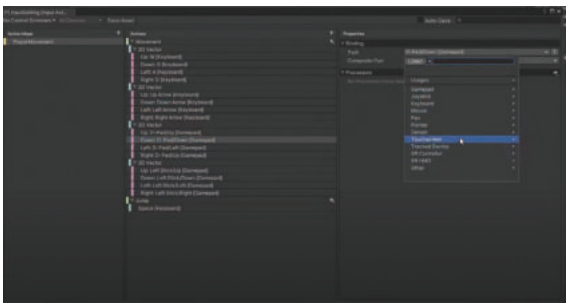
Gambar 4.51 Contoh Memasukan Aset Pembantu
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

3

Setelah itu kamu dapat melakukan *Setup Input Setting*, hal ini untuk mempermudah kamu ketika memilih fungsi tombol.



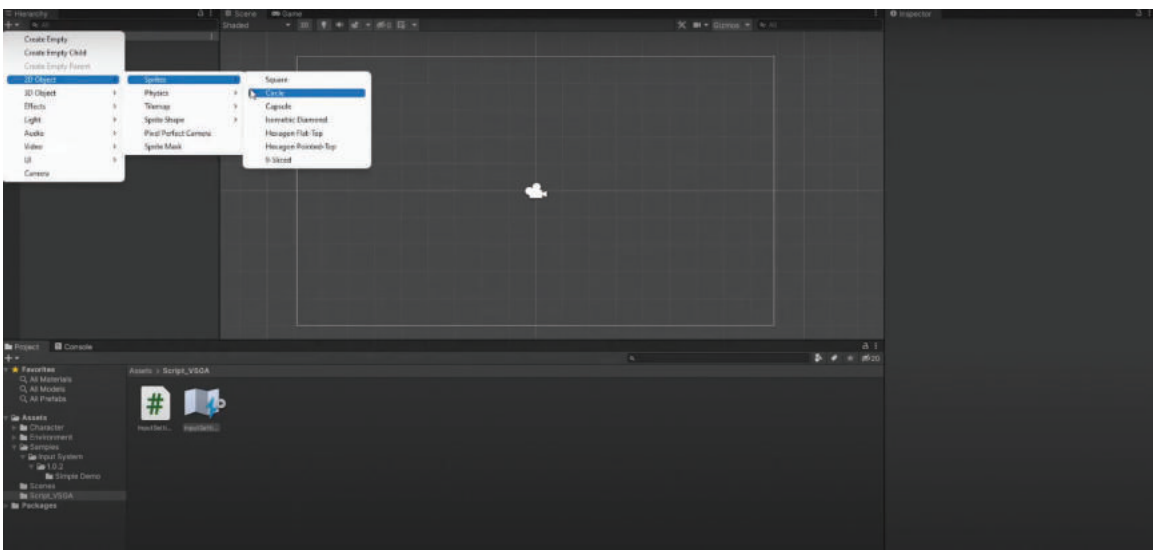
Gambar 4.52 Contoh Melakukan *Setup Input Mapping* agar Tombol Input dapat di Deteksi Sistem
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek 2024



Gambar 4.53 Contoh Melakukan *Setup Input Mapping* yang Sesuai Dengan Pemodelan Gim
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

4

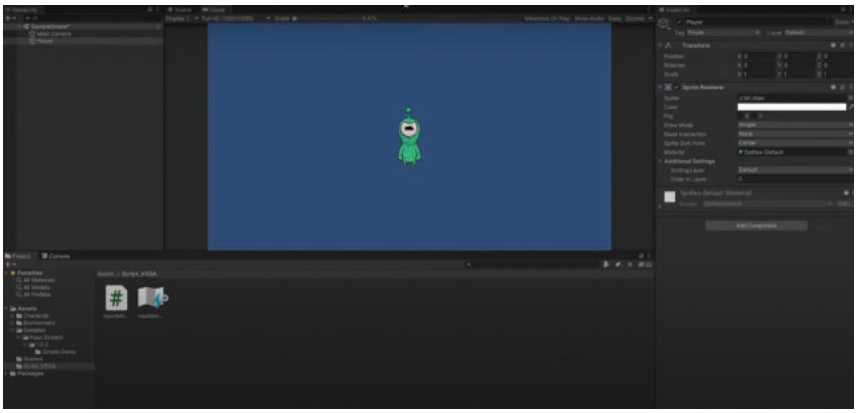
Berikutnya kamu bisa membuat objek 2D dengan menentukan pilihan *sprites* baik *Square* ataupun *Circle*. Pada proyek gim Mr. Alien memakai 2D *Circle Sprite*



Gambar 4.54 Contoh Cara Membuat Object 2D *Circle Sprite*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

5

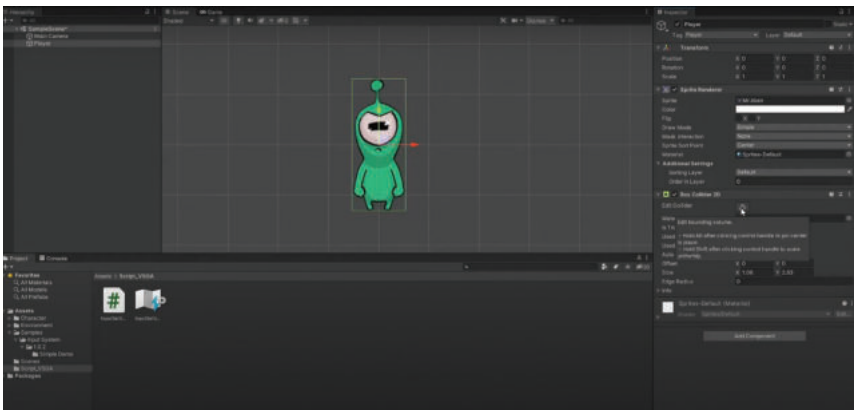
Kemudian kamu bisa mengganti *sprite* dengan animasi ataupun gambar 2D karakter yang kamu inginkan, seperti dalam proyek ini nama gambar 2D bernama Mr. Alien



Gambar 4.55 Contoh Mengganti *Sprite* dengan Animasi ataupun Gambar 2D
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

6

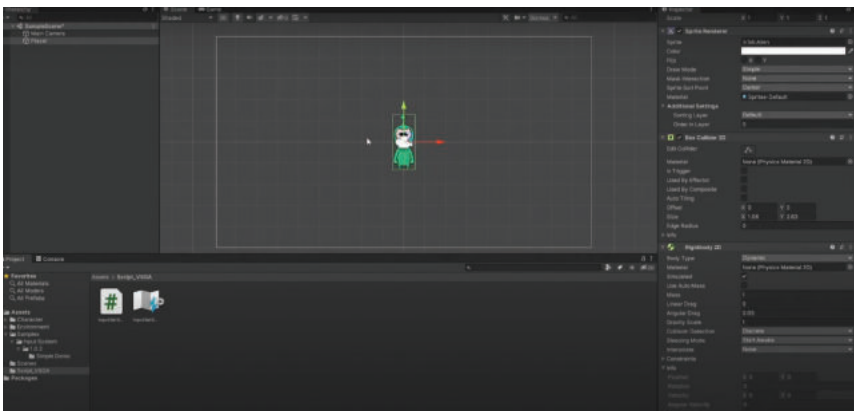
Berikan *collider* agar karakter tersebut dapat terdeteksi ketika bertabrakan atau *overlapped* dengan objek lain di dalam gim, dalam proyek ini akan memakai Box Collider 2D



Gambar 4.56 Contoh Saat Memberikan *Collider* Agar Karakter Dapat Terdeteksi sistem Fisika
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

7

Berikan *collider* agar karakter tersebut dapat terdeteksi ketika bertabrakan atau *overlapped* dengan objek lain di dalam gim, dalam proyek ini akan memakai Box Collider 2D



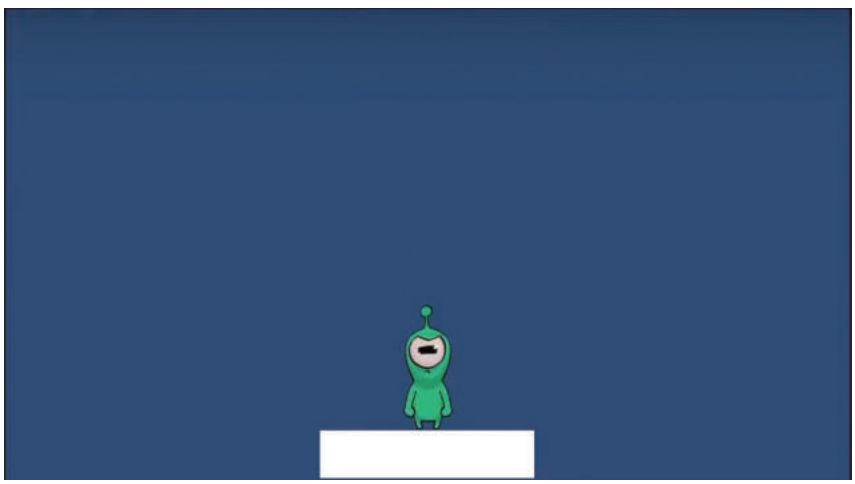
Gambar 4.57 Contoh Saat Menambahkan *Rigidbody*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

8

Tambahkan komponen bernama RigidBody agar karakter dapat kamu manipulasi dengan Gravitasi dan Sistem Fisika lainnya. Dalam Unity, kamu bisa menggunakan RigidBody2D

9

Hasil akhir dari proyek ini adalah karakter yang dapat berdiri di platform, karena sudah memakai komponen Rigidbody dan Collider



Gambar 4.58 Contoh Karakter Mr. Alien Terdeteksi Sistem Fisika dan Gravitasi Pada *Game Engine*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

Aktivitas 4.6 Praktik Mandiri

Coba Terapkan semua materi *game engine* melalui proyek gim kreasi mu sendiri. Praktekkan setiap tahapannya dan lengkapilah tabel berikut dengan tangkapan layar dari setiap tahap yang telah kamu lakukan dan hasil akhirnya:

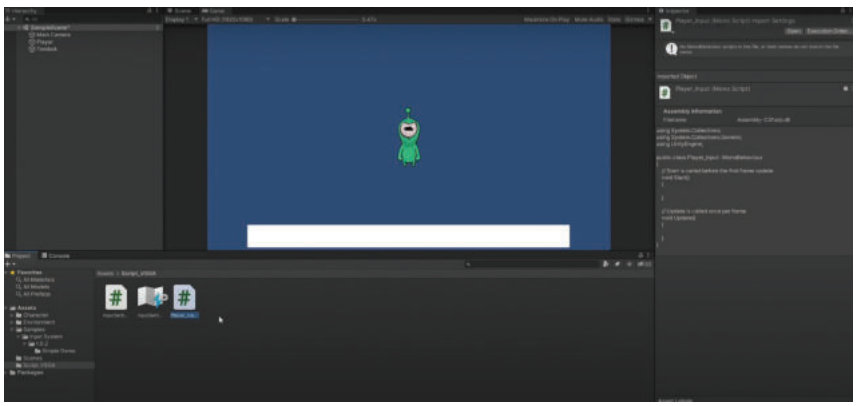
Tahapan Implementasi	Tangkapan Layar Tahapan	Tangkapan Layar Output

2. Gameplay



Setelah berhasil menciptakan karakter dengan menambahkan *Collider* dan *RigidBody*, selanjutnya kamu akan membuat *gameplay* sederhana. *Gameplay* dibuat dengan menyambungkan *input* pemain dan beberapa fungsi yang bisa membuat karakter bergerak atau melompat serta memberikan tantangan dan skor. Berikut ini tahapan membuat *gameplay* sederhana:

1. Untuk membuat karakter dapat dikendalikan oleh pemain, kamu harus membuat sebuah skrip yang diberi nama **Player_Input.**



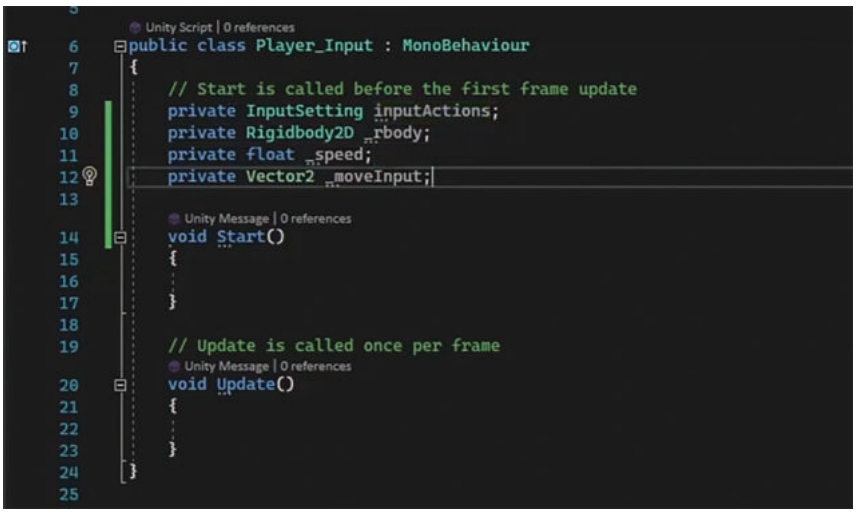
Gambar 4.59 Membuat Skrip Baru Bernama *Player_Input*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

2. Perlu kamu ketahui, di dalam Unity, ketika membuat *script* baru, maka akan terdapat dua fungsi otomatis yang dibuat. Pertama, Start() yang akan dijalankan satu kali pada awal permainan. Kedua, Update() yang akan dijalankan secara *looping* sampai permainan berakhir, seperti tampak pada gambar berikut:

```
6   public class Player_Input : MonoBehaviour
7   {
8       // Start is called before the first frame update
            Unity Message | 0 references
9       void Start()
10      {
11
12      }
13
14      // Update is called once per frame
            Unity Message | 0 references
15      void Update()
16      {
17
18      }
19  }
```

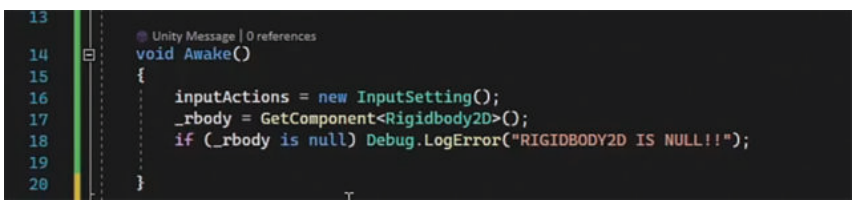
Gambar 4.60 Contoh Akan Muncul Fungsi *Start*() dan *Update*() Ketika *Script* Dibuat
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

3 Kamu harus menyiapkan beberapa variabel seperti *InputSetting* yang telah kamu pelajari sebelumnya. Variabel ini digunakan untuk menyimpan *Rigidbody* yang melekat pada karakter, variabel *float* untuk mencatat kecepatan, dan juga variabel *Vector2* untuk mengidentifikasi arah gerak karakter, seperti tampak pada gambar berikut



Gambar 4.61 Contoh Implementasi Skrip Baru Bernama *Player_Input*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

4 Sebelum permainan dimulai, kamu harus meng-*assign* variabel sesuai dengan kegunaannya. Terdapat fungsi Awake() yang akan dijalankan terlebih dahulu sebelum Start() dijalankan. Tujuannya adalah untuk memastikan tidak ada *error* pada variabel sehingga gim dapat dijalankan.



Gambar 4.62 Implementasi *Awake*() yang Akan Dijalankan Sebelum Fungsi *Start*()/*Update*()
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

5 Perhatikan gambar di bawah ini. Gambar tersebut menunjukkan ada dua fungsi yang perlu kamu buat agar bisa menghentikan pemain dan mengontrol karakter saat *game over*, *success* dalam menjalankan misi. Dua fungsi tersebut adalah OnEnable() dan OnDisable()

```
Unity Message | 0 references
private void OnEnable()
{
    inputActions.PlayerMovement.Enable();
}

Unity Message | 0 references
private void OnDisable()
{
    inputActions.PlayerMovement.Disable();
}
```

Gambar 4.63 Implementasi *OnEnable*() *and OnDisable*()
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

6 Selanjutnya, kamu hanya perlu memasukan kode ke dalam fungsi Update() agar fungsi dapat dijalankan terus menerus selama permainan. Sistem akan mengecek arah *input* pemain dan menjalankan kode program agar karakter Mr. Alien bergerak

```
// Update is called once per frame
Unity Message | 0 references
void Update()
{
    _moveInput = inputActions.PlayerMovement.Movement.ReadValue<Vector2>();
    _rbody.velocity = new Vector2(_moveInput.x * _speed, _rbody.velocity.y);
}
```

Gambar 4.64 Implementasi *Update*() untuk Melakukan Update Posisi Setiap Detik
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

7 Setelah itu, lakukan tes apakah Mr. Alien yang kamu buat tadi bisa bergerak sesuai dengan arah *input* pemain.

8 Berikutnya, kamu bisa menambahkan fitur lompat pada Mr. Alien. Untuk itu, kamu memerlukan tambahan variabel bernama jumpHeight dengan tipe data *float*, seperti tampak pada gambar berikut:

```
12      private float _speed;
13      private Vector2 _moveInput;
14      [SerializeField]
15      private float jumpHeight;
16
```

Gambar 4.65 Contoh Penambahan Variabel *jumpHeight* dengan *Tipe Float*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

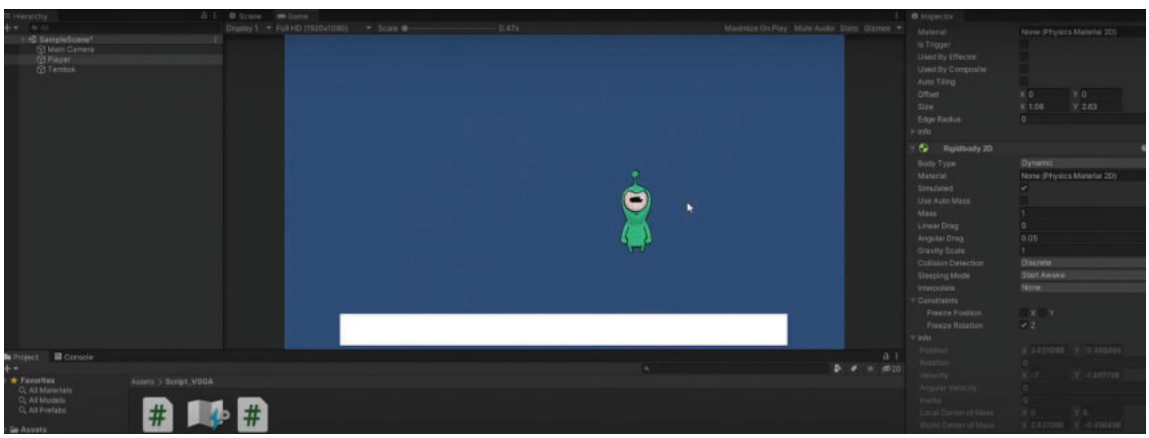
9 Kemudian, buatlah sebuah fungsi dengan ketentuan sistem akan melakukan cek terhadap input pemain apakah dia menekan tombol lompat. Jika iya, maka fungsi lompat akan dijalankan.

```
1 reference
private void Jump(UnityEngine.InputSystem.InputAction.CallbackContext context)
{
    //Caranya untuk melakukan Jump? caranya adalah dengan Add Force
    _rbody.AddForce(Vector2.up * jumpHeight, ForceMode2D.Impulse);
}

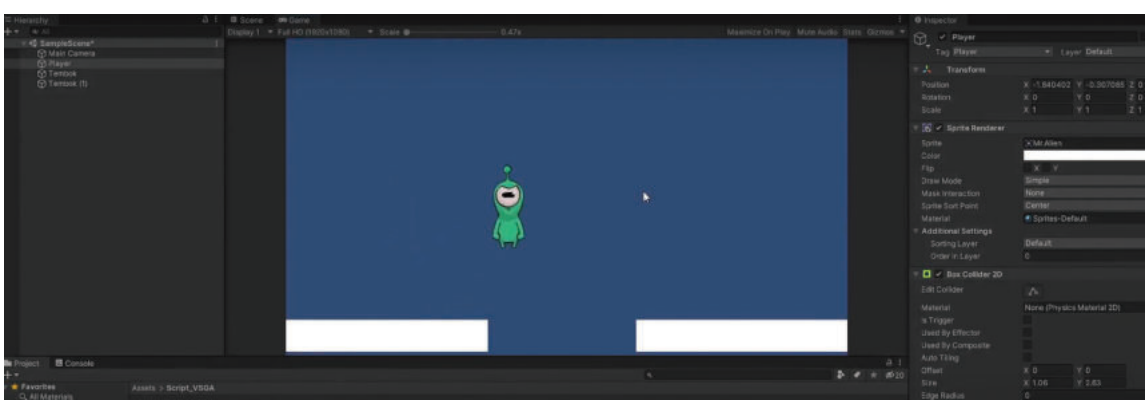
Unity Message | 0 references
void Start()
{
    inputActions.PlayerMovement.Jump.performed += Jump;
}
```

Gambar 4.66 Contoh Penambahan Suatu Kode dalam Fungsi *Start*()
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

10 *Output* dari fungsi jump di atas adalah Mr. Alien dapat melompat seperti tampak pada gambar berikut ini:.



Gambar 4.67 Contoh Mr. Alien Sudah Dapat Melompat ke Atas
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)



Gambar 4.68 Contoh Mr. Alien Sudah Dapat Melompati Lubang di Antara Dua Papan
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

Aktivitas 4.7 **Praktik Mandiri**

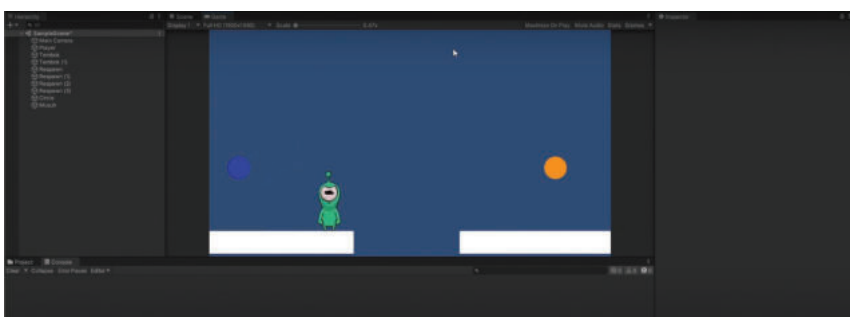
Dalam aktivitas ini, kamu dapat menerapkan materi *gameplay* dengan menggunakan gim kreasimu sendiri. Praktikkan setiap tahapannya, lalu isilah tabel berikut ini:

Tahapan Implementasi	Tangkapan Layar Tahapan	Tangkapan Layar Output

3. *User Interface* (UI)

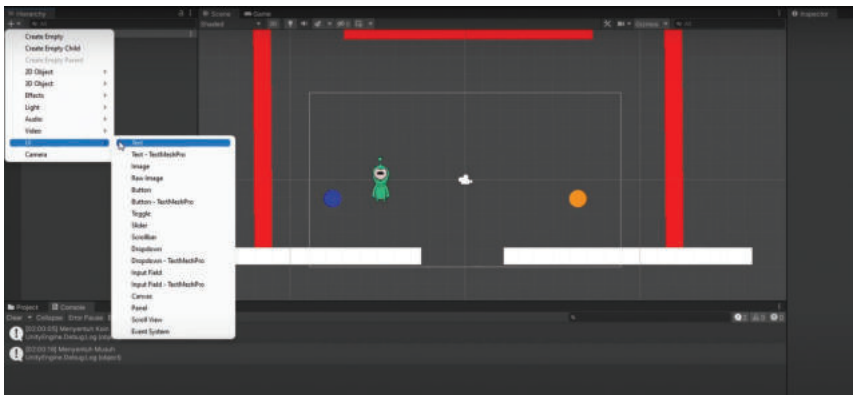
Setelah kamu berhasil membuat Mr. Alien dapat berjalan ke kiri dan ke kanan serta melompat, sekarang kamu akan membuat fungsi agar Mr. Alien dapat berinteraksi dengan objek lain. Objek lain yang akan kamu buat adalah koin dan musuh. Sebuah *User Interface* yang akan ditambahkan untuk memberikan informasi terkait skor dalam gim ini. Cara kerjanya, skor akan bertambah saat Mr. Alien bertabrakan dengan koin, dan gim akan berakhir (*game over*) saat Mr. Alien bertabrakan dengan musuh. Objek lingkaran biru menunjukkan koin, dan objek lingkaran kuning menunjukkan musuh.

1. Tambahkan sebuah UI untuk skor dan musuh Mr. Alien, sehingga ketika terjadi overlap dengan objek tersebut, maka skor akan bertambah atau berkurang.



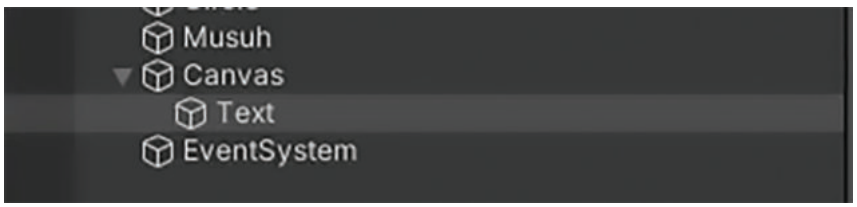
Gambar 4.69 Contoh Visualisasi Sederhana dari Koin (Kuning) dan Musuh (Biru).
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

2 Setelah itu, kamu juga akan membuat UI untuk *Text Score*, yang akan menunjukkan penambahan dan pengurangan skor. Kamu bisa klik Tab *Hierarchy* dan memilih *UI Text*



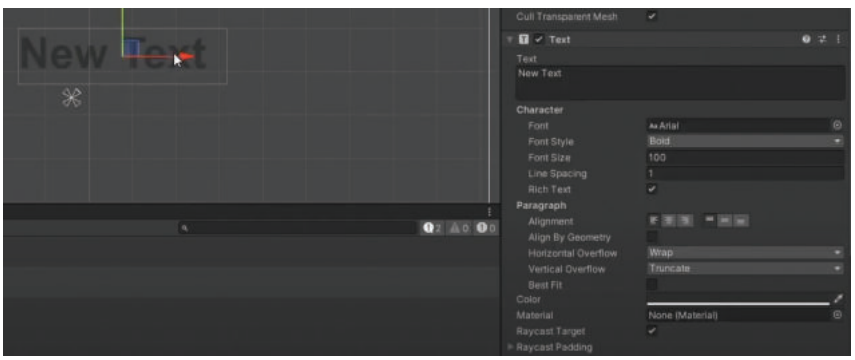
Gambar 4.70 Contoh Membuat Visualisasi *UI Text Score* dalam *Game Engine*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

3 Kemudian akan dibuat secara otomatis objek *Canvas* bernama *Text.*



Gambar 4.71 Contoh Susunan Komponen *Canvas* sebagai *Parent Class* dari *UI Text.*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

4 Selanjutnya, kamu bisa memilih font yang sesuai maupun bahasa yang kamu inginkan sesuai dengan GDD di tab *Inspector.*



Gambar 4.72 Contoh Implementasi dari *UI Text* dengan Beberapa Opsi Pilihan
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

5 Kemudian, berilah tulisan "Score:" dengan setting *UI Text* seperti ini gambar berikut:



Gambar 4.73 Contoh Implementasi Sederhana *UI Text* yang Sudah Berfungsi Pada Layar
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

6 Tambahkan sebuah variabel untuk memastikan *UI Text* dapat di *update* dalam *script*. Nama variabel yang digunakan adalah ScoreText, seperti tampak pada gambar dibawah ini:

```
private Vector2 _playerStartPosition;
private Text ScoreText;

Unity Message | 0 references
```

Gambar 4.74 Contoh Variable Text yang akan Melakukan Visualisasi *Scoretext* Pada Gim
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

7 Kamu juga perlu memasukkan kode tambahan pada fungsi Awake() untuk memastikan tidak ada variabel yang bermasalah sebelum bermain.

```
45  Unity Message | 0 references
    void Awake()
46  {
47      inputActions = new InputSetting();
48      _rbody = GetComponent<Rigidbody2D>();
49      if (_rbody is null) Debug.LogError("RIGIDBODY2D IS NULL!!");
50      ScoreText = FindObjectOfType<Text>();
51
52  }
```

Gambar 4.75 Contoh Implementasi *Awake*() Tambahan untuk *UI Text*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

8 Selanjutnya, kamu akan melakukan setting untuk variabel ScoreText di awal permainan. ScoreText akan diatur menjadi "Score: " saja, dengan kode sebagai berikut:

```
private Text ScoreText;
[SerializeField]
private int score;
```

Gambar 4.76 Contoh Implementasi *Start*() Tambahan untuk *UI Text*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

9 Untuk mencatat skor, kamu memerlukan variabel int dengan nama score agar sistem dapat mencatat skor dari pemain, seperti pada kode berikut:

```
Unity Message | 0 references
void Start()
{
    inputActions.PlayerMovement.Jump.performed += Jump;
    ScoreText.text = "Score: "
}
```

Gambar 4.77 Contoh Implementasi Variable Score
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

10 Lalu, lakukan penggabungan *Text* dengan skor aktual menggunakan kode seperti berikut:

```
Unity Message | 0 references
void Start()
{
    inputActions.PlayerMovement.Jump.performed += Jump;
    ScoreText.text = "Score: " + score;
}
```

Gambar 4.78 Contoh Implementasi Fitur Untuk Melakukan Inisiasi Awal *Score* dengan *Variable Score*.
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

11 Setelah itu, buatlah sebuah fungsi yang bernama OnTriggerEnter2D(Collider2D collision). Fungsi tersebut akan membuat sistem dapat mendeteksi apakah karakter pemain sedang bertabrakan dengan objek lain. Kita ingin sistem bisa mengecek dua kondisi yang berbeda dengan menggunakan konsep *Branch If Statement* untuk dua jenis objek, yaitu "Coin" dan "Musuh". Berikut kode programnya:

```
Unity Message | 0 references
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.tag == "Coin")
    {
        Debug.Log("Menyentuh Koin");
        Destroy(collision.gameObject);
        score++;
    }

    if (collision.tag == "Musuh")
    {
        Debug.Log("Menyentuh Musuh");
        Destroy(collision.gameObject);
        Destroy(this.gameObject);
    }
}
```

Gambar 4.79 Fungsi OnTriggerEnter2D(Collider2D collision)
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

Penjelasan fungsi:

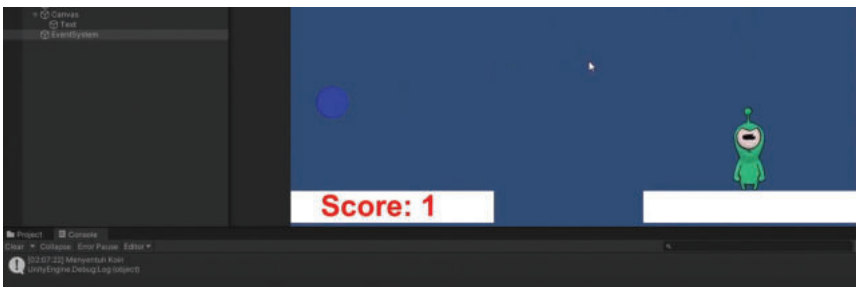
a. Ketika bertabrakan dengan koin, maka koin akan dihancurkan dan terjadi penambahan pada skor
b. Ketika bertabrakan dengan musuh, maka musuh dan pemain akan dihancurkan.

12 Sekarang coba kamu jalankan programnya dan lihat hasilnya apakah sesuai dengan instruksi program atau tidak. Untuk pemberian nilai awal score adalah 0 (nol).



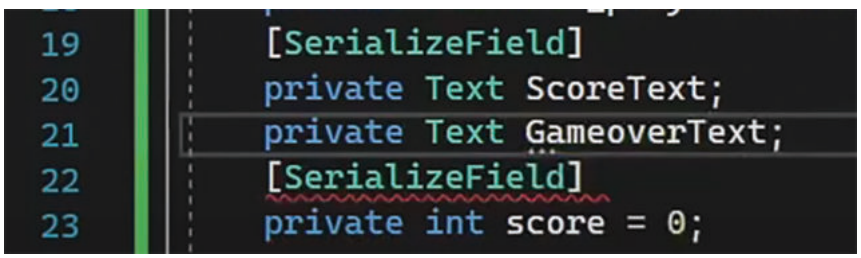
Gambar 4.80 Contoh Awal Skornya 0 karena Baru Memulai Permainan
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

13 Kemudian saat Mr. Alien menabrak koin (objek berwarna kuning) maka score akan bertambah 1 dengan visual text "Score: 1" seperti tampak pada gambar berikut ini:



Gambar 4.81 Contoh Awal Skor Bertambah
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

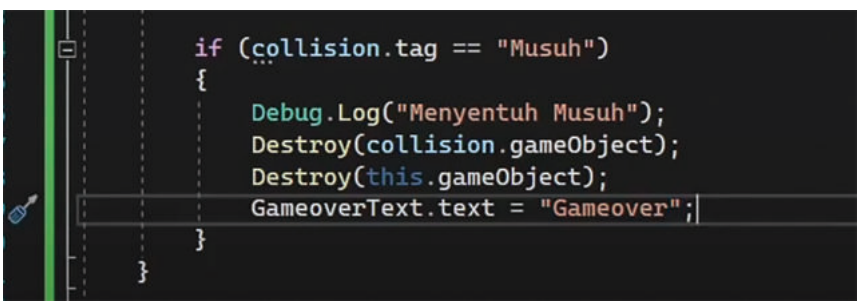
14 Selanjutnya jika Mr. Alien menyentuh musuh (objek berwarna biru) maka akan muncul tulisan "Game Over" . untuk itu kamu perlu menambahkan satu *UI text* lagi dan memasukkan kode programnya seperti tampak pada gambar berikut iniL:



```
19      [SerializeField]
20      private Text ScoreText;
21      private Text GameoverText;
22      [SerializeField]
23      private int score = 0;
```

Gambar 4.82 Contoh Variabel UI Teks Tambahan untuk *State "Gameover".*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

15 Kemudian tambahkan kode pada *branch* "Musuh" dengan mengganti tulisan GameoverText menjadi "Gameover" seperti berikut:



```
        if (collision.tag == "Musuh")
        {
            Debug.Log("Menyentuh Musuh");
            Destroy(collision.gameObject);
            Destroy(this.gameObject);
            GameoverText.text = "Gameover";
        }
    }
```

Gambar 4.83 Contoh Fungsi Tambahan untuk *State "GameOver".*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

16 Sekarang coba kamu jalankan programnya dan lihat apakah *output*nya sesuai dengan kode program yang kamu buat tadi.



Gambar 4.84 Contoh Skor Bertambah Ketika Mr Alien Menyentuh "Koin" Pada Gim
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)



Gambar 4.85 Contoh Ketika Mr. Alien Menyentuh "Musuh" dan Permainan Berakhir
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

Aktivitas 4.8 Praktik Mandiri

Dalam praktik kali ini kamu diminta untuk menambahkan sebuah objek dalam gim Mr. Alien, dimana objek ini akan menambahkan nyawa Mr. Alien sebanyak 2 kali. Nyawa ini akan berkurang jika Mr. Alien mengenai musuh. Presentasikan kode program dan output programmu di depan kelas.

Kode Program
Tangkapan Layar Objek dan Output Kode Program

Setelah kamu memahami bagaimana menggunakan *game engine* dan menerapkan *gameplay* serta *User Interface* nya, kamu bisa menerapkan materi ini untuk gim kreasimu sendiri dengan mengerjakan aktivitas di bawah ini:.

Aktivitas 4.9 Kerja Tim

Dalam aktivitas ini, kamu akan melakukan kerja tim yang terdiri dari 3 orang anggota. Tugas kalian adalah membuat sebuah *gameplay* dan UI/UX untuk gim dengan karakteristik kalian sendiri. Selain Unity, kalian bisa menggunakan *engine* gim lain yang pernah kalian pelajari atau gunakan. Kemudian kalian lengkapi tabel berikut dan mempresentasikannya di depan kelas untuk berbagi informasi!mudian kamu jelaskan tentang desain tersebut!

	Kode Program	Tangkapan Layar Hasil Output
Gameplay		
UI/UX		



Uji Kompetensi

Untuk menguji dan mengasah keterampilan kamu lebih lanjut, sekarang saatnya menyusun sebuah portofolio. Berikut adalah langkah-langkah yang perlu kamu lakukan:

1. Buat dan integrasikan objek statis serta objek dinamis dalam gim yang kamu kembangkan.
2. Instal dan tambahkan fungsi *plugin* di Unity dan Unreal Engine atau *game engine* lainnya yang pernah kamu gunakan.
3. Buatlah sebuah objek dinamis dengan menggunakan *script* atau kode program.
4. Buatlah antarmuka pengguna (UI) dan pengalaman pengguna (UX) dengan menggunakan *script* atau kode program.

5. Setelah itu lengkapi tabel berikut ini :

Tangkapan Layar Kerjamu, Disini!
1
2
3
4

Pengayaan

Bagi kamu yang sudah memahami tentang materi dalam bab ini, kamu bisa melakukan eksplorasi lebih mendalam tentang bagaimana pemrograman gim yang akan kamu buat berdasarkan gim yang ingin kamu ciptakan sendiri menggunakan *engine* lain dan Bahasa pemrograman lainnya!

Kamu bisa mempelajari dan mempraktekkan materi yang ada dalam buku berikut ini:

1. Game Coding Complete, Fourth Edition, Mike "MrMike" McShaffry and David "Rez" Graham, 2013 Course Technology
2. Holistic Game Development with Unity, Penny de Byl , CRC Press, 2017

Refleksi

Kamu bisa memberikan tanda centang (✓) pada kotak yang kamu anggap sesuai!

Setelah kamu mempelajari bab ini, bagaimanakah penguasaan kamu terhadap materi-materi berikut?

No	Materi	Tidak Menguasai	Menguasai	Sangat Menguasai
1	Struktur Data			
2	Integrasi Objek			
3	Fungsional Tambahan			
4	Pemograman berbasis teks dan grafis • Game engine • Gameplay • UI/UX			

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2024
Pengembangan Gim
untuk SMK/MAK Kelas XI-XII
Penulis: Rini Melati dan Muhammad Ardhan Fadlurrahman
ISBN: 978-634-00-0360-4

BAB 5

Aset Audio Visual



Tujuan Pembelajaran

Setelah mempelajari materi dalam bab ini, diharapkan kalian mampu menerapkan pengembangan aset visual gim sesuai minatnya, yang meliputi pembuatan aset visual 2D menggunakan teknik *puppeteer* (*cut out animation*), pengembangan model 3D dengan teknik digital *sculpting*. Selain itu, kalian juga diharapkan mampu mengembangkan aset audio berupa efek suara dan musik latar, serta menerapkan teknik perekaman suara (*dubbing*).

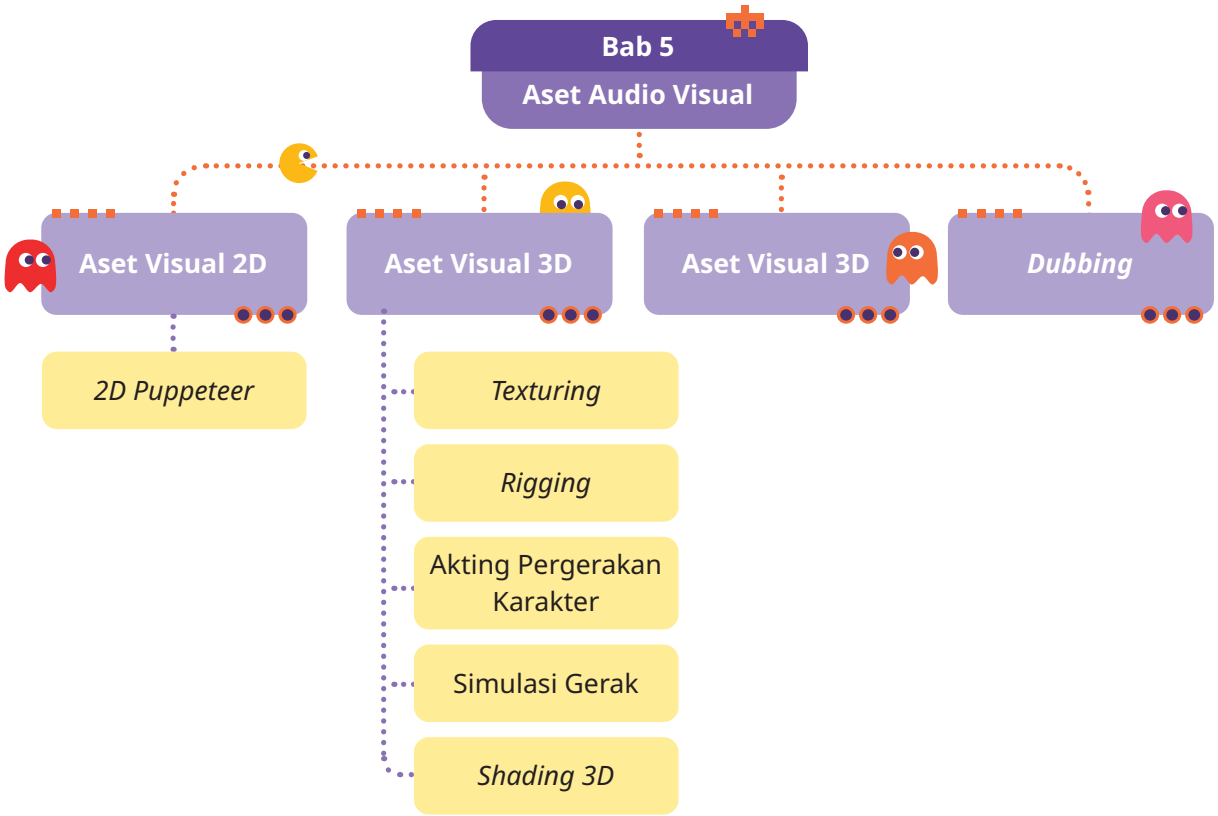
Kata Kunci

Aset Audio · Aset Visual · *Texturing* · *Puppeteer* · *Dubbing* · *Rigging*

Soft Skill

- **Kreatif** dalam menghasilkan ide-ide baru dan orisinal yang dapat membuat aset visual dan audio menjadi menarik serta sesuai dengan tema gim.
- **Berkolaborasi** dengan tim lintas disiplin (desainer, animator, dan sound designer) untuk memastikan bahwa semua elemen terintegrasi dengan baik.
- **Berkomunikasi secara efektif** dalam menyampaikan ide dan mendapatkan umpan balik serta memastikan visi desain dipahami oleh seluruh tim.
- **Teliti** saat memperhatikan detail kecil dalam desain untuk memastikan kualitas tinggi dan konsistensi dalam semua aset yang dikembangkan.
- **Adaptabilitas** dengan perubahan dalam arahan desain atau teknologi baru selama proses pengembangan.

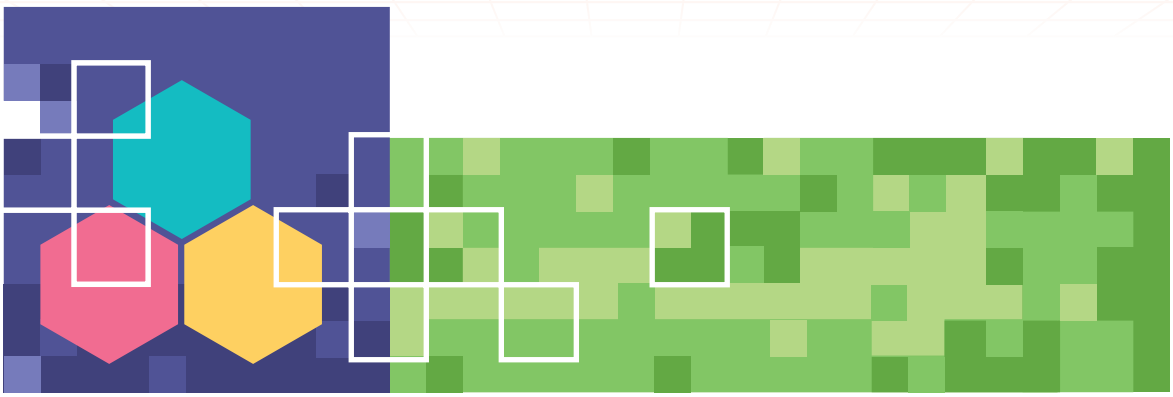
Peta Materi



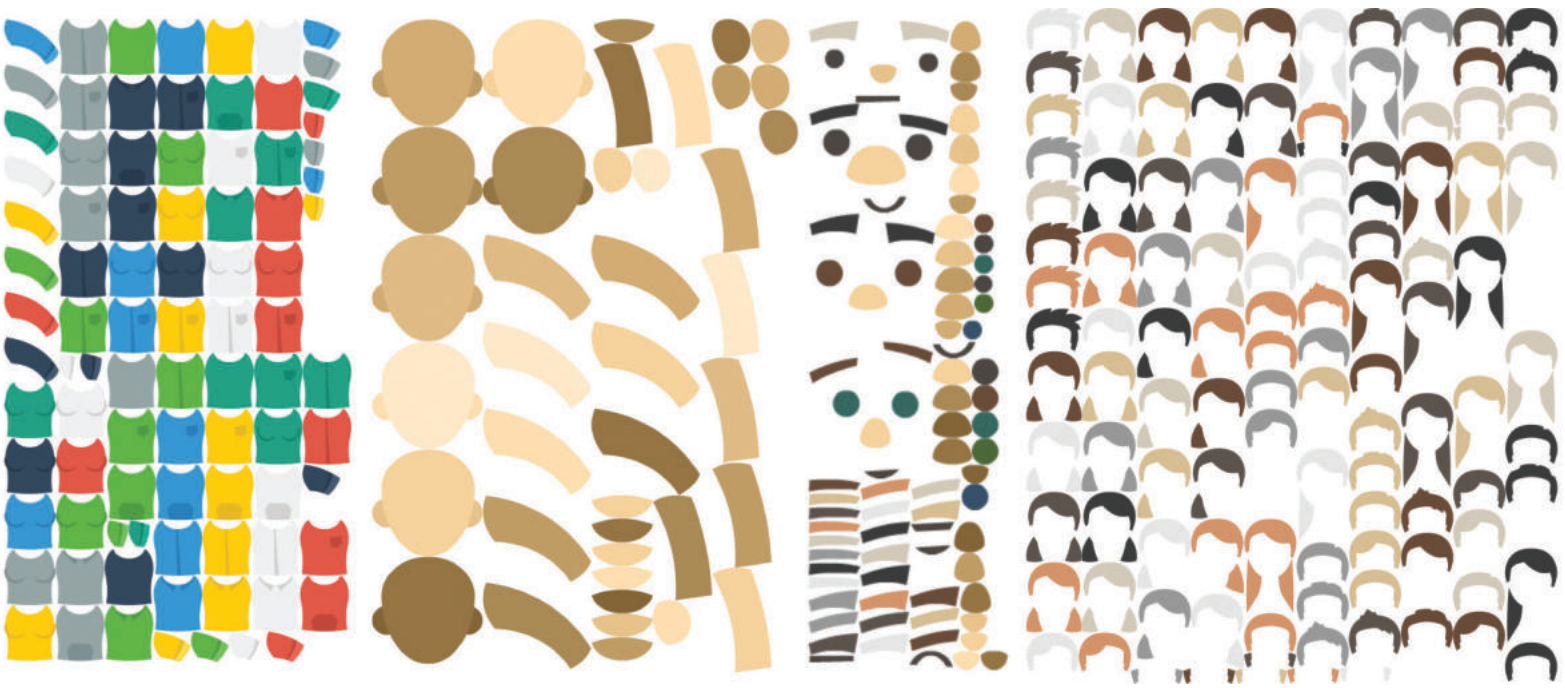
Pernahkah kamu memainkan gim dengan grafik yang sangat keren hingga membuatmu ingin terus bermain? Misalnya, saat melihat karakter yang bergerak dengan lincah atau pemandangan dalam gim yang terlihat nyata. Lalu, bayangkan jika gim itu juga memiliki musik seru dan suara yang pas dengan setiap aksi, seperti suara pedang saat menyerang atau suara langkah kaki ketika berjalan di hutan. Apakah itu membuatmu semakin suka bermain? Itulah kekuatan aset audio-visual dalam sebuah gim. Grafik yang bagus membuat mata kita senang, dan suara yang tepat membuat kita merasa seperti benar-benar berada di dalam gim. Coba ingat kembali gim favoritmu, apa yang paling kamu sukai? Apakah grafiknya yang keren atau musiknya yang membuatmu semangat?

Dalam bab ini, kita akan belajar tentang aset visual, baik yang 2D maupun 3D, serta bagaimana audio dan *dubbing* menambah keseruan saat bermain gim. Setelah mempelajarinya, kamu bisa mulai berpikir, "Jika aku membuat gim, seperti apa ya grafik dan suara yang akan membuat pemain lain betah bermain seperti aku?"

A. Aset Visual 2D



2D Puppeteer Cut Out Animation adalah teknik yang memungkinkan kamu untuk menciptakan gerakan karakter yang halus dan dinamis. Dengan menggunakan potongan gambar (*cut outs*) dan sistem rigging mirip boneka (*puppet*), kamu dapat menghasilkan animasi yang efisien dan menarik. Berikut merupakan contoh potongan gambar yang akan dipakai sistem rigging mirip boneka (*puppet*).



Gambar 5.1 Potongan Gambar *Puppeteer*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)



Gambar 5.2 Contoh 2D *Puppet* yang digabungkan
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

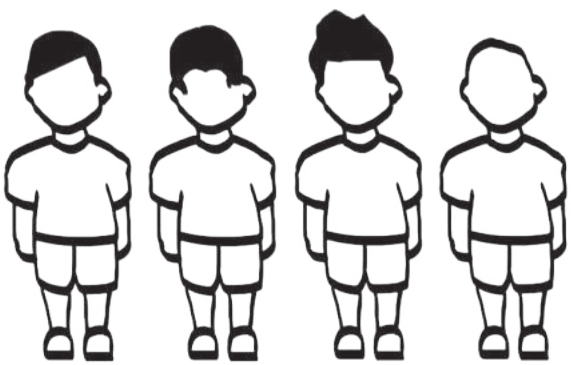
Dengan potongan gambar tersebut kita dapat membuat beberapa karakter. Kamu bisa melihat bahwa gambar 2D *puppeteer* di atas dibuat sebagai karakter pembeli makanan dalam suatu gim simulasi rumah makan. Dengan membuat karakter dari *cut-out*, banyak tipe karakter yang dapat dibuat dari aset gambar yang lebih sedikit dan lebih efisien.

Berikut ini adalah beberapa proses yang perlu kamu lakukan untuk membuat sebuah desain 2D *puppeteer* ini:

1. Konsep Desain

Dalam proses ini, kamu akan memulai tahap menciptakan sebuah konsep dari karakter dan elemen lain yang dibutuhkan. Kamu bisa memulainya dengan membuat sketsa awal yang menggambarkan bentuk dan gaya visual dari karakter tersebut. Karakter yang kamu buat harus memiliki kepribadian dan daya tarik visual yang disesuaikan dengan tema gim yang kamu kembangkan.

Di samping ini adalah contoh sketsa karakter pembeli makanan dalam sebuah gim yang bertema simulasi rumah makan.

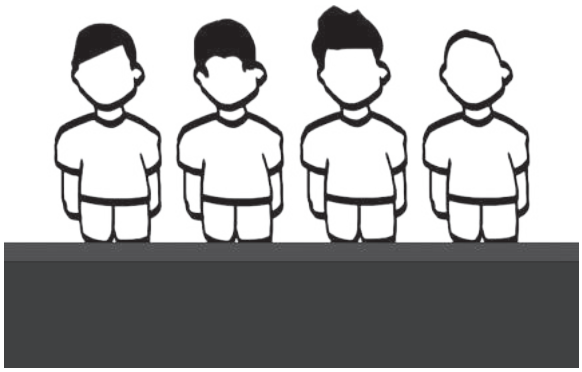


Pelanggan/pembeli makanan dapat mempunyai rambut, baju, sepatu, bahkan jenis kelamin berbeda.

Gambar 5.3 Contoh sketsa karakter
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

2. *Storyboarding*

Setelah konsep desain selesai, langkah berikutnya yang kamu lakukan adalah membuat *storyboard*. *Storyboard* berfungsi sebagai panduan visual yang merencanakan alur cerita dan interaksi antar karakter. Pada permainan rumah makan, storyboard digambarkan secara kasar posisi dari pembeli yang berada di belakang meja kasir saat memesan makanan.



Gambar 5.4 Contoh *Storyboard* Karakter Memesan Makan
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

3. Ilustrasi Karakter

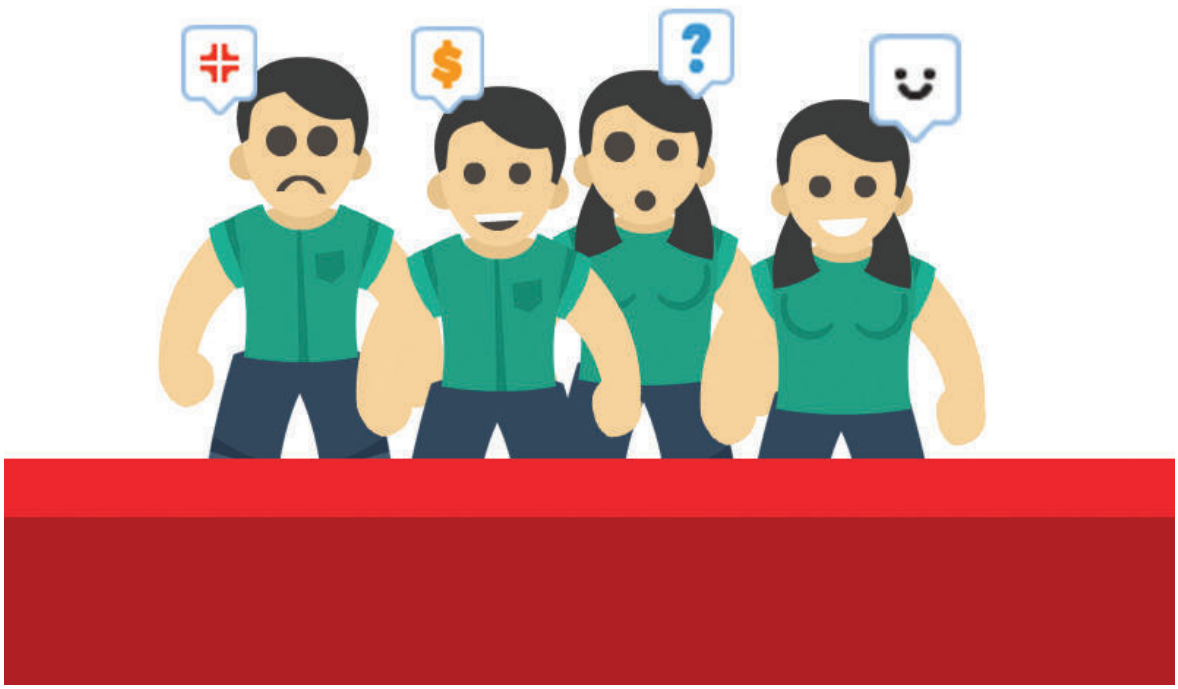
Setelah semua persiapan dilakukan, kamu bisa mulai membuat aset visual. Dengan Menggunakan perangkat lunak seperti *Adobe Illustrator* atau *Photoshop*, kamu dapat membuat potongan gambar dari karakter. Setiap bagian, seperti kepala, lengan, dan kaki, harus dibuat sebagai *layer* terpisah untuk memungkinkan fleksibilitas saat animasi (Hirsch, 2017). Ilustrasi karakter ini merupakan pengembangan dari *storyboard* sehingga mulai terlihat lebih jelas bagaimana pelanggan menunggu di belakang meja kasir.



Gambar 5.5 Contoh Ilustrasi Karakter
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

4. Ekspresi dan Pose

Kamu juga perlu membuat variasi ekspresi wajah dan pose untuk karakter. Hal ini penting untuk menambahkan kedalaman emosional pada karakter dan memungkinkan mereka beradaptasi dengan berbagai situasi dalam gim (Freeman, 2012). Gambar berikut ini menunjukkan berbagai ekspresi dari pembeli, mulau dari senang sampai dengan sedih yang membuat gambar menjadi lebih hidup.



Gambar 5.6 Contoh Ekspresi Karakter
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Aktivitas 5.1 **Praktik Mandiri**

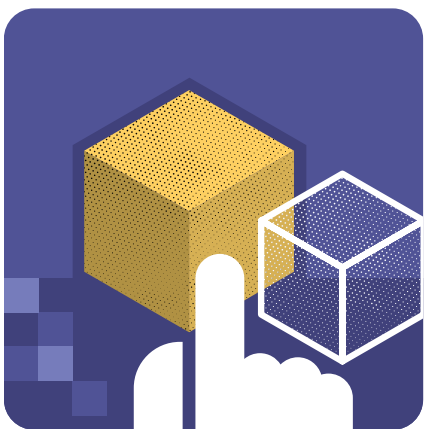
Dengan memperhatikan langkah-langkah pembuatan 2D *Puppeteer*, coba kamu buat sebuah desain gambar karaktermu sendiri untuk dijadikan *puppeteer*. Kamu bisa menggunakan gambar manual dengan aplikasi yang sudah pernah kamu gunakan.

B. Aset Visual 3D



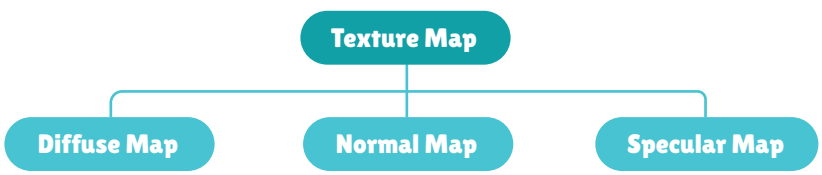
Pengembangan gim 3D memerlukan berbagai langkah yang kompleks, termasuk *texturing* dan *rigging*. *Texturing* berfokus pada penerapan gambar ke permukaan model 3D untuk menciptakan detail visual, sedangkan rigging melibatkan penciptaan sistem tulang untuk memungkinkan animasi. Di bawah ini, kamu akan diberikan penjelasan tentang bagaimana melakukan dan membuat aset visual 3D dengan proses *texturing* dan *rigging*.

1. Texturing



a Konsep dan Jenis Tekstur

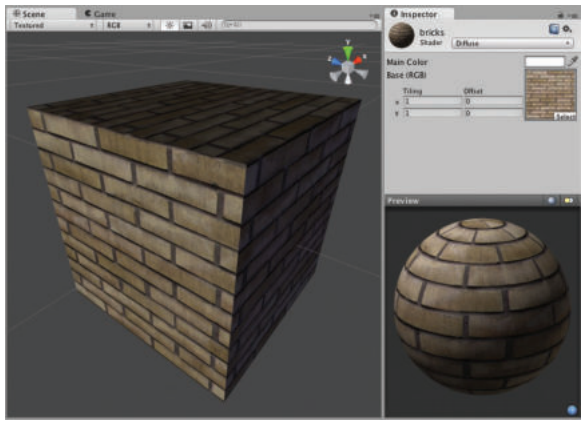
Texturing adalah proses penting dalam pengembangan gim 3D. Tekstur memberikan karakter dan nuansa pada objek, membuat dunia gim lebih realistis dan menarik. Berikut adalah jenis tekstur yang perlu kamu ketahui:



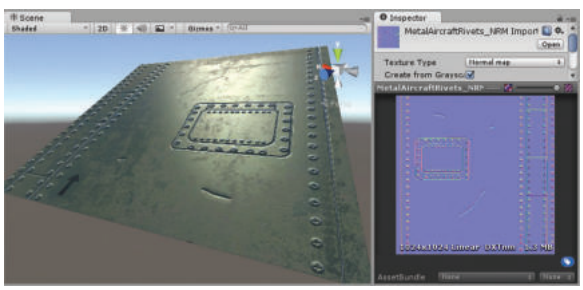
TEXTURE MAP

1. Diffuse Map

Menentukan warna dasar objek dan merupakan gambar yang paling umum digunakan (Meyer, 2008).



Gambar 5.7 *Diffuse Map* di *Unity*
Sumber: Shader - Shader-NormalDiffuse/ Unity Technologies (2024)



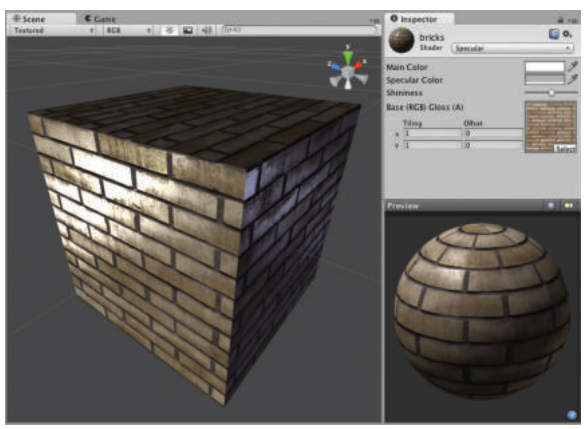
Gambar 5.9 *Diffuse Map* di *Unity*
Sumber: Shader - Shader-NormalDiffuse/ Unity Technologies (2024)

2. Normal Map

Memberikan ilusi detail permukaan dengan memanipulasi cara cahaya berinteraksi dengan objek, menciptakan kedalaman tanpa menambah geometri (Baker, 2016).

3. Specular Map

Mengontrol bagaimana permukaan memantulkan cahaya, menambahkan kilau dan refleksi (Blinn, 1978).

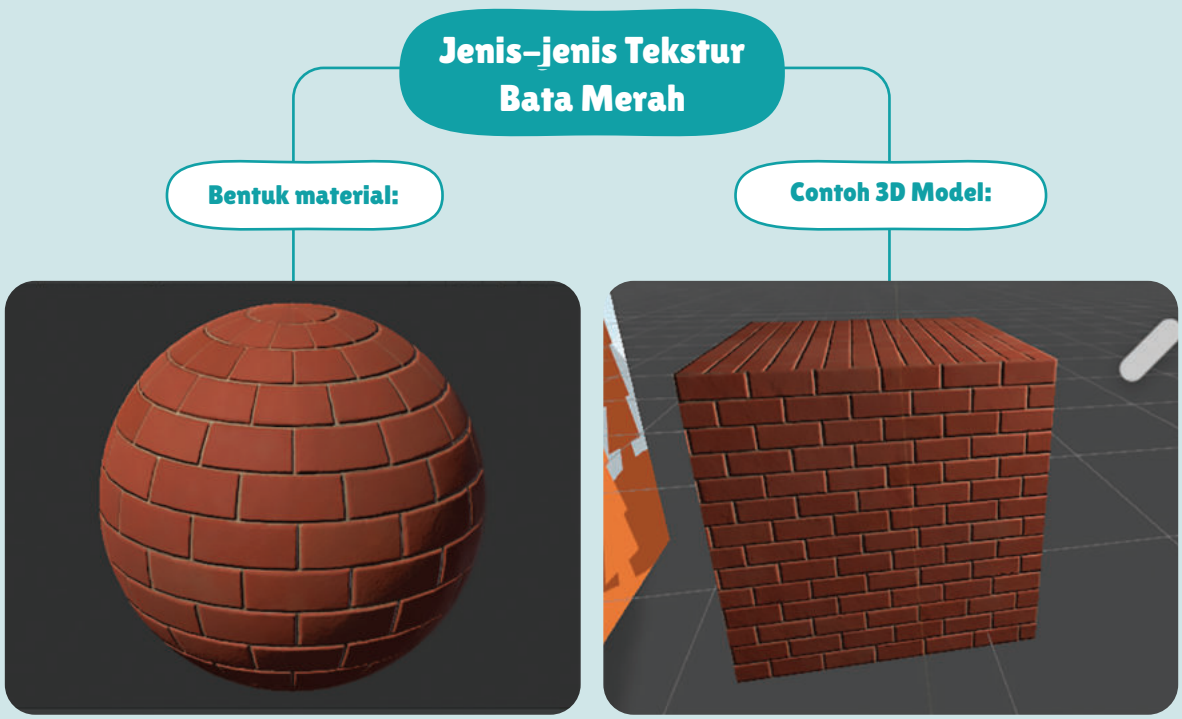


Gambar 5.8 *Specular Map* di *Unity*
Sumber: Shader - Shader-NormalDiffuse/ Unity Technologies (2024)

b Pembuatan Tekstur

Dalam proses pembuatan tekstur, kamu bisa mengikuti beberapa langkah penting agar hasilnya sesuai dengan kebutuhan desain. Langkah pertama yang perlu kamu lakukan adalah **mengumpulkan referensi visual**. Referensi ini berguna untuk memahami pola, material, dan warna yang diinginkan pada objek. Gambar-gambar referensi akan membantumu menciptakan tekstur yang lebih realistis dan konsisten dengan konsep desain (Snyder, 2019).

Setelah referensi terkumpul, kamu bisa mulai **membuat gambar tekstur menggunakan perangkat lunak** seperti Photoshop, Substance Painter, atau GIMP. Pada tahap ini, kamu dapat menggunakan berbagai teknik seperti *digital painting*, penggabungan foto, atau metode lainnya untuk menciptakan tekstur yang sesuai dengan karakteristik objek yang akan digunakan dalam gim. Berikutnya, kamu akan mempelajari langkah-langkah pembuatan tekstur aset 3D menggunakan Photoshop. Namun, kamu juga bebas menggunakan aplikasi lain yang sudah kamu pelajari sebelumnya.



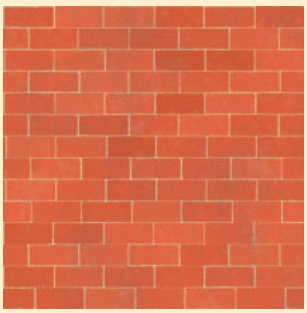
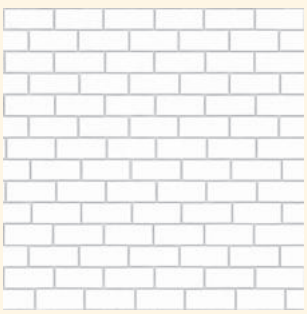
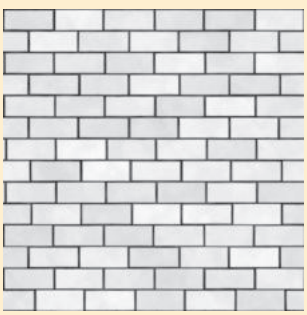
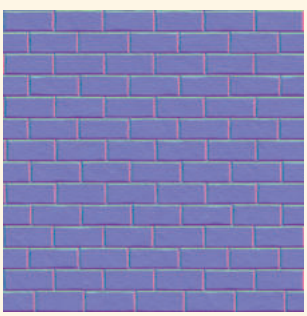
Gambar 5.10 Contoh Material Batu Bata di *Game Engine*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Gambar 5.11 Contoh Material Batu Bata yang sudah dimasukan ke dalam *Game* Objek di dalam *Game Engine*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Material tersebut terdiri dari beberapa gambar yang disatukan sebagai informasi suatu objek tersebut, contohnya:

Tabel 6.1 Berbagai jenis *Texture Map* yang digunakan

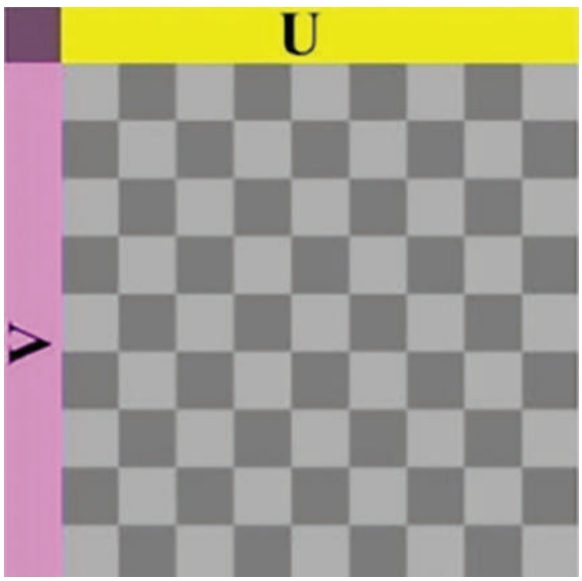
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Nama Bagian	Contoh Material	Kegunaan
Albedo		Memberikan warna pada objek 3D.
AO (*Ambience Occlusion*)		Teknik untuk membantu melakukan simulasi *soft shadow* untuk membuat objek menjadi lebih realistis
Height Map		Gambar yang memodifikasi geometri objek berdasarkan warna hitam dan putih. Semakin hitam maka semakin rendah dan semakin putih maka semakin tinggi.
Normal Map		Turunan dari *height map* yang berfungsi mirip tetapi lebih kompleks. *Normal map* akan membantu membuat objek menjadi lebih detail ketika tersorot oleh cahaya. Biasanya digunakan bersama dengan *height map*.

Nama Bagian	Contoh Material	Kegunaan
Roughness Map		Turunan dari *normal map* yang menentukan seberapa kasar (*roughness*) ketika disinari oleh cahaya. Gambar ini memakai putih dan hitam seperti *height map*. Biasanya digabungkan dengan *smoothness map* untuk penyesuaian lebih kompleks.
Smoothness Map		Turunan dari *normal map* yang memastikan seberapa halus (*smoothness*) ketika disinari oleh cahaya. Gambar ini memakai putih dan hitam seperti *height map*. Dipakai secara optional untuk melakukan adjustment lebih kompleks dan digabungkan dengan *smoothness map*.

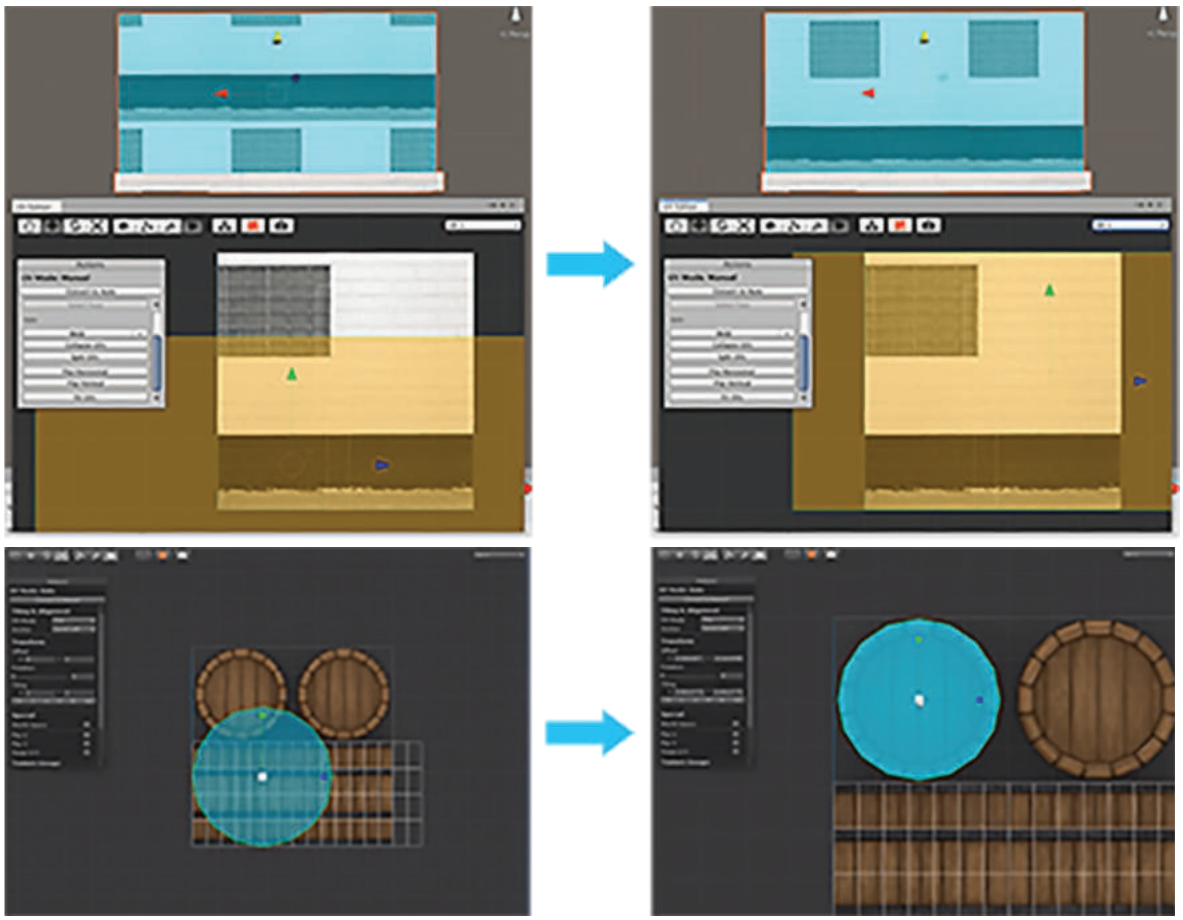
C UV Mapping

Sue Blackman dalam bukunya *Beginning 3D Game Development with Unity* 4 menjelaskan bahwa dalam *UV Mapping* menunjukan koordinat proyeksi. Pada koordinat vektor biasanya mereka akan memakai X,Y, dan Z. Namun, untuk menghindari kebingungan dalam merender gambar dalam objek, digunakan simbol lain yaitu U,V, dan W dimana U,V dipakai untuk map 2D dan U,V,W untuk map 3D, di mana W adalah axis yang digunakan sebagai variabel koordinat ketika memanipulasi map dalam 3D (Sue Blackman, 2013).



Gambar 5.12 Contoh U dan V untuk Melakukan *Mapping* Koordinat Material ke dalam Objek
Sumber: Sue Blackman, 2013

UV mapping adalah proses ketika kamu meratakan model 3D ke dalam bentuk dua dimensi menggunakan sumbu **U** dan **V**. Kedua sumbu ini menentukan bagaimana tekstur akan ditempelkan pada permukaan objek. Dalam proses ini, kamu akan membagi model menjadi beberapa bagian yang kemudian bisa "dijahit" kembali untuk menciptakan peta UV yang efisien dan mudah diterapkan pada tekstur (Blinn, 1978).



Gambar 5.13 UV *Mapping* di dalam Unity
Sumber: Probuilder - UVExample_Translate/ Unity Technologies (2024)

Setelah peta UV selesai kamu buat, tahap berikutnya adalah **pengeditan UV**. Pengeditan ini bertujuan untuk memastikan bahwa tekstur diterapkan secara tepat dan tidak mengalami distorsi pada permukaan objek. Dengan pengaturan skala dan posisi yang akurat, kamu bisa menghindari kesalahan visual yang sering muncul saat objek dirender (Baker, 2016).

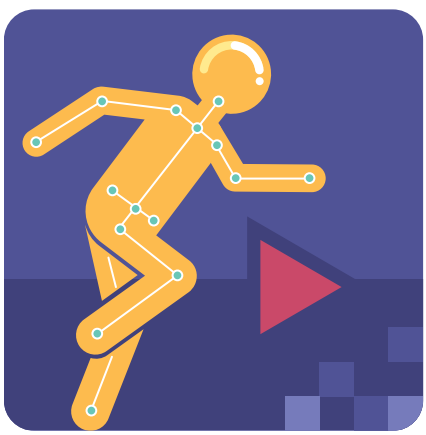


Gambar 5.14 UV *Scalling* di Unity
Sumber: Probuilder - UVExamples_Translate/Unity Technologies (2024)

Aktivitas 5.2 **Praktik Mandiri**

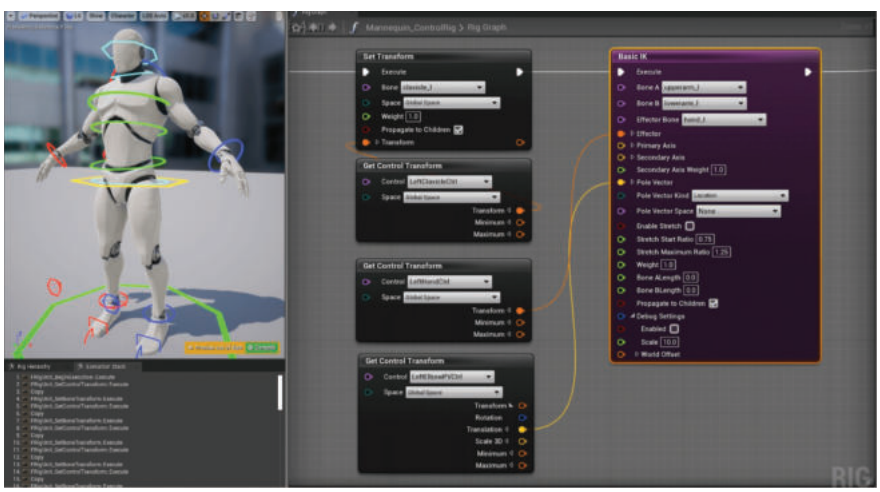
Melanjutkan aktivitas sebelumnya, kamu dapat menggunakan desain ini atau kamu membuat desain yang berbeda. Setelah itu, lakukanlah *texturing* pada desainmu dengan mengikuti alur proses *texturing* yang telah dibahas di atas. Kamu boleh menggunakan *engine* yang pernah kamu gunakan sebelumnya. Terakhir, presentasikan hasil kerjamu di depan teman yang lain untuk saling berbagi informasi.

2. Rigging



Untuk membuat objek bergerak dalam tiga dimensi, kamu perlu menambahkan rigging pada setiap objek agar dapat dianimasikan oleh animator. Dalam pengembangan gim 3D, *rigging* adalah proses membuat sistem tulang (*bones*) dan pengontrol (*controllers*) untuk model 3D, memungkinkan model tersebut bergerak dan beranimasi dengan cara yang realistis.

a Struktur Rig



Gambar 5.15 Contoh *Automate Rigging* dengan *Control Rig* di Unreal Engine 5
Sumber: Documentation - Control Rig / Epic Games (2024)

Dalam proses rigging, kamu perlu membangun **struktur rig** sebagai dasar dari animasi karakter 3D. Langkah pertama adalah **membuat tulang (bones)** dengan menentukan posisi dan rotasi setiap tulang sesuai dengan bagian tubuh atau elemen yang ingin kamu animasikan. Struktur ini menjadi dasar dari semua gerakan yang akan dilakukan oleh karakter (Hernandez, 2020).

Setelah membuat tulang, kamu perlu menyusun tulang-tulang tersebut dalam sebuah **hierarki**. Dalam hierarki ini, tulang yang satu akan terhubung ke tulang lainnya sesuai struktur tubuh. Misalnya, tulang lengan akan kamu hubungkan ke tulang tubuh, sehingga saat tubuh bergerak, lengan juga ikut bergerak. Pengaturan hierarki ini penting agar animasi karakter terlihat lebih realistis dan gerakannya mengikuti logika anatomi (Jones, 2015).

b Penghubungan Geometri dengan Tulang (*Skinning*)

Setelah struktur tulang selesai dibuat, langkah berikutnya adalah menghubungkan geometri model 3D dengan sistem tulang melalui proses yang disebut **skinning**. Pada tahap ini, kamu akan menentukan seberapa besar pengaruh setiap tulang terhadap bagian-bagian tertentu dari *mesh*. (Freeman, 2012).

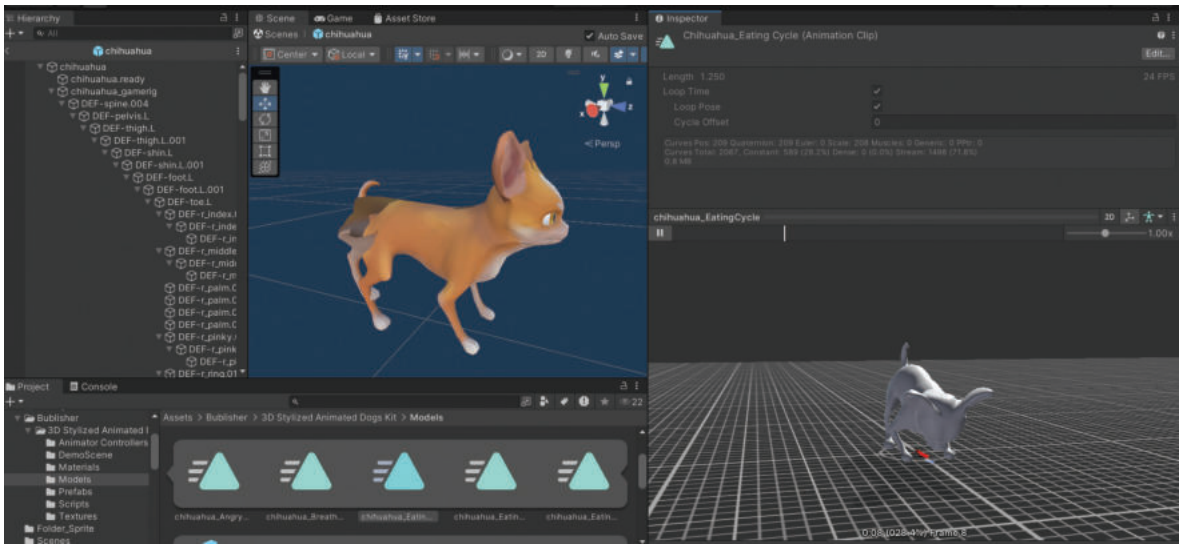
Selanjutnya, kamu akan menggunakan teknik **weight painting** untuk mengatur bobot (*weight*) pada setiap *vertex* di dalam *mesh*. Teknik ini membantu kamu mengontrol seberapa besar pengaruh tulang terhadap posisi *vertex* tertentu. Dengan *weight painting* yang tepat, kamu bisa menciptakan deformasi gerakan yang lebih halus dan natural saat karakter bergerak (Baker, 2016).

c Pengontrol (*Controllers*)

Dalam proses animasi karakter, kamu bisa menggunakan **pengontrol** untuk mempermudah pengaturan gerakan. Pengontrol adalah objek tambahan yang biasanya berbentuk geometris sederhana, seperti lingkaran atau kotak, dan lebih mudah kamu atur dibandingkan menggerakkan tulang secara langsung (Snyder, 2019).

Pengontrol ini juga bisa kamu atur agar dapat menggerakkan beberapa tulang sekaligus. **Pengaturan fungsionalitas** seperti ini sangat membantu saat membuat animasi yang kompleks dan memungkinkan kamu memiliki kontrol yang lebih besar terhadap keseluruhan gerakan karakter (Hirsch, 2017).

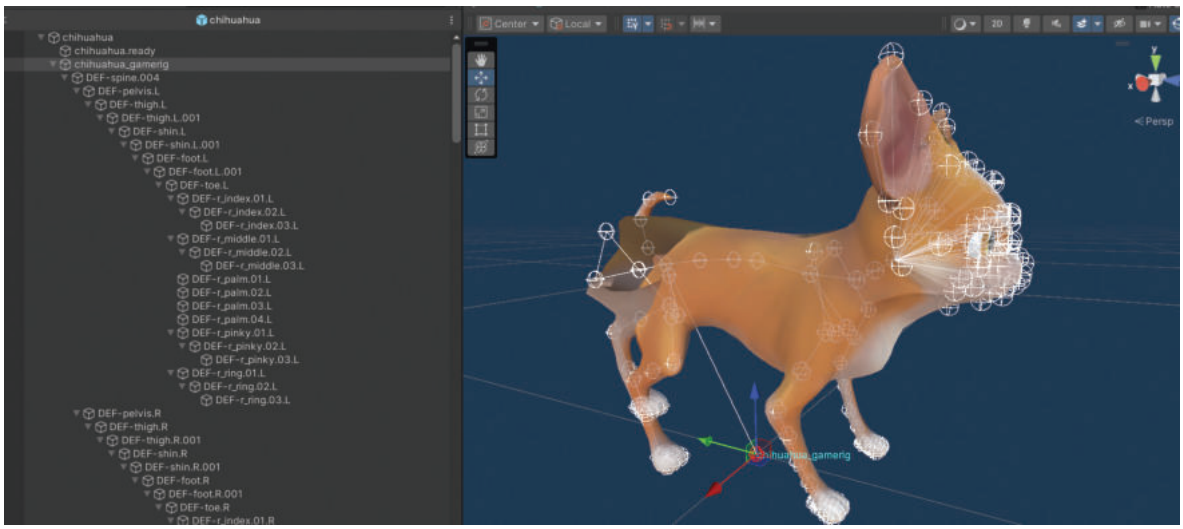
Berikut adalah contoh animasi yang digunakan di dalam objek “chihuahua”



Gambar 5.16 Contoh Animasi Anjing di dalam *Game Engine* Unity
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

Objek “*Chihuahua*” pada gambar di bawah ini mempunyai urutan bones (tulang (*bones)* seperti berikut. Di dalam *game engine*, setiap tulang mencatat siapa anak dan orang tua dari bones tersebut. Yang dimaksud anak dan orang tua dari tulang adalah sebagai berikut:

1. **Anak dari tulang** disebut juga *child bones (tulang anak),* yaitu *bones* yang memiliki dependensi terhadap tulang yang sedang dipilih. Misalnya *bone foot* memiliki anak seperti *bone toe, index, middle* dan sebagainya. Ketika *bone foot* diangkat, maka anaknya seperti *bone toe, bone index, bone middle,* dan semua anak dari *bone foot* akan juga bergerak mengikuti *bone foot.*
2. **Orang tua dari tulang** disebut juga *parent bones* (tulang orang tua), yaitu memiliki definisi *bones* (tulang) yang menjadi dependensi dari tulang yang sedang dipilih. Misalnya *bone foot* diangkat, maka orang tuanya seperti *bone shin*, atau *thigh* tidak akan berubah atau bergerak mengikuti *bone foot.* Namun, jika kita menggerakan *bone shin*, maka *bone foot* akan berubah karena *bone* tersebut adalah anak dari *bone shin.*



Gambar 5.17 Gambaran *Bone* dan Juga Dependensi
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Aktivitas 5.3 **Praktik Mandiri**

Melanjutkan aktivitas sebelumnya, kamu dapat menggunakan desain ini atau kamu membuat desain yang berbeda. Setelah itu, lakukanlah *rigging* pada desainmu sendiri dengan mengikuti alur proses *rigging* yang telah dibahas sebelumnya. Kamu dapat menggunakan *engine* yang pernah kamu pelajari atau gunakan sebelumnya. Lalu presentasikanlah hasil kerjamu dengan teman-teman yang lain untuk saling berbagi informasi.

3. Akting Pergerakan Karakter



Motion Capture (Mocap) adalah salah satu teknologi canggih yang banyak digunakan dalam pengembangan gim dan film untuk menciptakan animasi karakter yang realistis. Dengan merekam gerakan fisik aktor, *mocap* mampu menangkap nuansa gerakan manusia secara halus dan kompleks.

Motion capture adalah proses saat kamu merekam gerakan fisik aktor menggunakan sensor atau kamera. Data gerakan ini kemudian diterjemahkan menjadi animasi untuk karakter digital dalam lingkungan 3D. Teknologi ini memungkinkan animasi karakter yang sangat realistis dan responsif terhadap input pemain (Cohen, 2013).

Tujuan utama dari *motion capture* adalah membantu kamu menangkap nuansa gerakan manusia yang halus, termasuk ekspresi wajah, gerakan tangan, dan postur tubuh. Dengan demikian, karakter digital yang kamu buat dapat bergerak secara alami, memberikan pengalaman yang lebih imersif kepada pemain (Meyer, 2008). Berikut adalah penjelasan mendetail mengenai teknik, proses, integrasi, dalam penggunaan teknologi *motion capture*.

a Teknik Motion Capture

1. Optical Motion Capture

Optical motion capture adalah metode yang paling umum kamu temui. Dalam teknik ini, aktor mengenakan pakaian khusus dengan *marker* yang memantulkan cahaya. Kamera akan menangkap posisi *marker* tersebut, dan perangkat lunak mengubahnya menjadi data gerakan. Metode ini sering dipakai dalam film dan permainan video untuk menghasilkan animasi yang sangat akurat (Baker, 2016). Berikut contoh penggunaan *Optical Motion Capture* bisa kamu lihat dalam proses pembuatan *cutscene The Last of Us,* seperti yang dijelaskan oleh Sony Playstation pada event Playstation Experience 2014



Gambar 5.18 Proses Pembuatan *cutscene* Menggunakan *Optical Motion Capture*
Sumber: (Sony Playstation, 2014)

2. *Inertial Motion Capture*

Berbeda dengan *optical mocap*, *inertial motion capture* menggunakan sensor inersia seperti giroskop dan akselerometer untuk merekam gerakan. Teknik Ini memberikan kamu kebebasan gerak yang lebih besar dan memungkinkan pengambilan data di luar studio, sehingga cocok untuk pengambilan gambar di lokasi (Freeman, 2012). Salah satu contoh penggunaan *inertial motion capture* bisa kamu lihat dalam proses pembuatan *cutscene Agni: Village of Calamity,* di mana aktornya memakai pakaian khusus yang dilengkapi sensor inersia seperti giroskop dan akselerometer untuk merekam gerakan.



Gambar 5.19 Contoh Penggunaan *Inertial Motion Capture*
Sumber: Separuh Interactive (2024)

3. *Facial Motion Capture*

Facial motion capture fokus pada menangkap ekspresi wajah. Teknik ini sering dilakukan dengan menggunakan kamera khusus yang merekam detail wajah aktor saat berakting. Data yang dikumpulkan ini digunakan untuk memberikan ekspresi realistis pada karakter digital (Snyder, 2019). Berikut contoh penggunaan *facial motion capture* yang bisa kamu lihat dalam proses pembuatan *cutscene Agni: Village of Calamity.*



Gambar 5.20 Contoh Penggunaan *Facial Motion Capture*
Sumber: Separuh Interactive (2024)

b Proses Motion Capture

1. Rekaman

Sebelum proses *motion capture* dimulai, jika kamu sebagai aktor maka kamu harus mempersiapkan diri dengan mengenakan pakaian *mocap* dan *marker*. Studio tempat kamu melakukan rekaman juga harus disiapkan dengan memiliki lingkungan yang sesuai dengan pencahayaan yang tepat untuk mendapatkan hasil yang optimal (Hirsch, 2017).

2. Pengolahan Data

Setelah rekaman selesai, kamu perlu memproses data mentah untuk menghilangkan *noise* dan kesalahan. Perangkat lunak akan mengkonversi data menjadi informasi yang dapat digunakan oleh model 3D. Kamu perlu perhatian ekstra pada proses ini untuk memastikan kualitas data yang dihasilkan tetap tinggi (Baker, 2016).

c Integrasi dengan Model 3D

1. Penerapan Data

Data gerakan yang sudah kamu proses akan diterapkan ke model 3D karakter. Pada tahap ini, animator melakukan penyesuaian agar gerakan karakter sesuai dengan desain dan proporsi karakter (Meyer, 2008).

2. Rigging

Model 3D yang kamu gunakan untuk *motion capture* harus memiliki rig yang sudah disiapkan. Rig ini sangat penting karena memungkinkan data gerakan diterapkan dengan benar dan menghasilkan animasi yang halus. Proses *rigging* yang tepat sangat krusial untuk membantu kamu mengoptimalkan hasil dari *mocap* (Freeman, 2012).

d Penyempurnaan dan Pengeditan

1. Cleaning Up Data

Meskipun *mocap* bisa menghasilkan data gerakan yang sangat akurat, biasanya kamu tetap memerlukan penyempurnaan. Kamu atau animator dapat mengedit data untuk menghilangkan *jitter* atau artefak lainnya yang mungkin muncul selama pengambilan data (Cohen, 2013).

2. Tweaking

Selain itu, kamu juga dapat menambahkan detail atau memperbaiki gerakan tertentu yang mungkin tidak ditangkap dengan baik oleh sistem *mocap*. Proses tweaking ini penting supaya gerakan karakter yang kamu buat terlihat alami dan sesuai dengan konteks cerita atau adegannya (Hirsch, 2017).

Dalam Proses Pembuatan *AGNI: Village of Calamity*, Separuh Interactive menggunakan *motion capture* di mana memiliki proses pembuatan *mocap* seperti berikut:

1 Mempersiapkan objek karakter yang memerlukan motion capture



Gambar 5.21 Objek Karakter yang Memerlukan Data *Motion Capture*
Sumber: Separuh Interactive (2024)

2 Mempersiapkan aktor ke dalam setup environment motion capture



Gambar 5.22 Aktor Mempersiapkan Baju untuk Setup *Environment Motion Capture*
Sumber: Separuh Interactive (2024)

3 Merekam motion capture



Gambar 5.23 Aktor Merekam Gerakan *Motion Capture* untuk Keperluan Scene Dialog
Sumber: Separuh Interactive (2024)

4 Mengintegrasikan dengan model 3D



Gambar 5.24 *Motion Capture* yang Sudah Diintegrasikan ke Model Karakter 3D
Sumber: Separuh Interactive (2024)

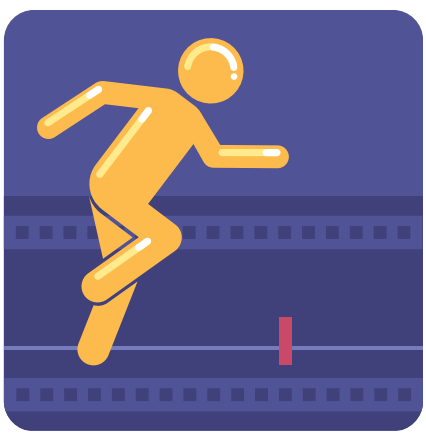
5 Cleanup dan pengeditan

Setelah kamu melakukan integrasi, langkah berikutnya adalah *clean-up* footage, yaitu proses pembersihan data *motion capture* untuk menghilangkan gerakan yang stuttering atau tersendat-sendat. Selain itu, kamu juga akan mengurangi jumlah *keyframe* supaya animasi tidak terlalu berat saat digunakan di dalam permainan.

Aktivitas 5.4 **Praktik Mandiri**

Melanjutkan aktivitas sebelumnya, kamu bisa menggunakan desain ini atau kamu membuat desain yang berbeda, kemudian lakukanlah *mocap* pada desainmu sendiri dengan mengikuti alur proses *mocap* yang telah dibahas sebelumnya. Kamu bisa menggunakan *engine* yang pernah kamu gunakan pelajari. Lalu presentasikanlah hasil kerjamu dengan teman yang lain untuk saling berbagi informasi

4. Simulasi Gerak



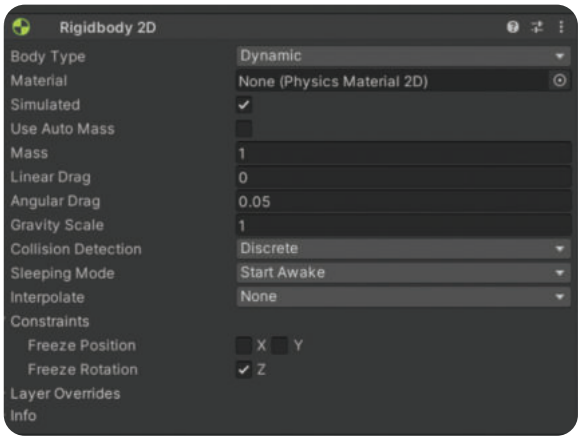
Simulasi gerak digital adalah teknik penting yang bisa kamu gunakan dalam pengembangan gim untuk membuat objek berinteraksi secara realistis. Ada dua kategori utama dalam simulasi ini, yaitu *Rigid Body* (benda kaku) dan *Soft Body* (benda lunak). Kedua jenis simulasi ini membantu kamu menciptakan dinamika yang lebih hidup di dunia virtual, sehingga pengalaman bermain terasa lebih imersif dan menantang.

a Simulasi *Rigid Body*

Rigid body simulation adalah proses saat kamu mensimulasikan objek yang tidak mengalami perubahan bentuk ketika berinteraksi dengan objek lain. Objek-objek kaku seperti mobil, bola, atau bangunan diprogram untuk bereaksi terhadap gaya fisik di sekitarnya tanpa mengubah bentuk aslinya (Reynolds, 2015).

1. Fisik Dasar

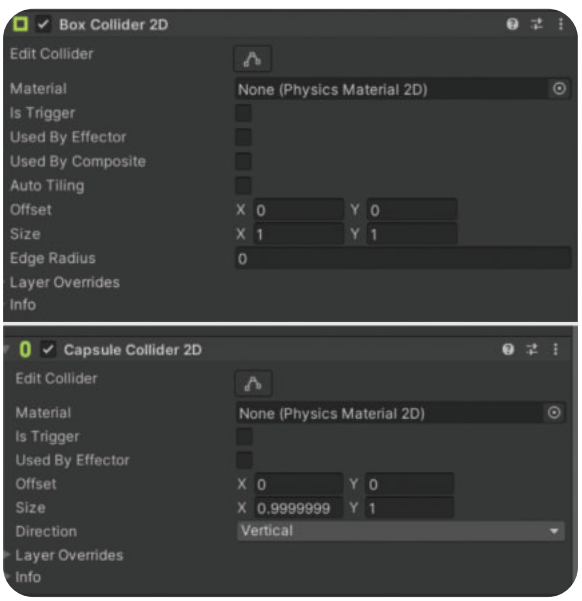
Dalam simulasi *rigid body*, kamu akan menggunakan hukum-hukum fisika, termasuk hukum Newton tentang gerakan, untuk menghitung bagaimana objek bereaksi terhadap gaya yang diberikan. Gaya-gaya ini mencakup gravitasi, tumbukan, dan gesekan, yang semuanya berperan penting dalam menciptakan interaksi yang realistis antar objek (Müller et al., 2018). Berikut contoh komponen simulasi fisika bernama Rigidbody 2D yang spesifik untuk mensimulasikan fisika di ruang dua dimensi. Disini bisa dilihat terdapat komponen *Mass* (berat), *Drag* (gesekan), dan simulasi fisika lain untuk satu objek di dalam gim.



Gambar 5.25 Komponen simulasi fisika bernama *Rigidbody 2D*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

2. *Collision Detection*

Proses *collision detection* adalah langkah saat kamu mendeteksi apakah dua objek *rigid* berinteraksi. Proses ini melibatkan perhitungan matematis untuk menentukan apakah dua objek bertabrakan, dan bagaimana mereka akan bereaksi satu sama lain (Baraff & Witkin, 1998). Berikut adalah dua jenis implementasi *collider* di dalam *game engine* Unity yaitu Box Collider dan Capsule Collider. Dua jenis tersebut memiliki fungsi yang cukup sama tetapi dengan bentuk yang berbeda.



Gambar 5.26 Jenis Implementasi *Collider* di dalam *Game Engine Unity*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

b Penggunaan dalam Gim

1. Interaksi Realistis

Simulasi *rigid* body memungkinkan objek dalam gim yang kamu buat berperilaku seperti di dunia nyata. Contohnya, mobil yang bergerak di jalur dan benda-benda yang jatuh menciptakan pengalaman *gameplay* yang lebih mendalam (Erleben et al., 2009).



Gambar 5.27 Mengambil Botol di Gim *Half-Life*: Alyx dengan Simulasi Cairan.
Sumber: Valve Game Studio (2024)

Di dalam satu gim *virtual reality* bernama *Half-Life: Alyx,* kamu bisa melihat bagaimana sistem fisikanya dibuat sangat realistis. Misalnya, saat kamu mengambil sebuah botol, kamu bisa melihat cairan di dalam botol itu bergerak persis seperti di dunia nyata. Detail seperti ini membuat interaksi terasa jauh lebih nyata dan secara langsung menambah tingkat imersif dalam permainan.

2. *Gameplay* Dinamis

Simulasi fisika juga menyediakan mekanika gameplay yang menarik, seperti teka-teki berbasis fisika, di mana kamu harus memanipulasi objek untuk mencapai tujuan tertentu. Mekanika ini tidak hanya menambah tantangan, tapi juga mendorong kreativitas kamu sebagai pemain dalam menyelesaikan masalah (Kelley, 2017). Berikut adalah contoh gim yang memakai sistem fisika untuk membuat *gameplay* menjadi dinamis, yaitu Where's my Water dan Cut the Rope.



Gambar 5.28 Gim yang Memakai Sistem Fisika
Sumber: Creature Feep, Disney Mobile (2024)

C Simulasi *Soft Body*

Soft body simulation berfokus pada objek yang dapat mengalami deformasi ketika berinteraksi dengan objek lain atau ketika dipengaruhi oleh gaya. Contoh objek *soft body* termasuk karakter, kain, dan benda lunak lainnya (Baraff, 1994). Berikut beberapa komponen yang digunakan untuk mengatur jalannya simulasi *soft body,* seperti:

1. *Mesh Deformation*

Simulasi *soft body,* kamu akan menggunakan *mesh* untuk merepresentasikan objek. *Mesh* ini bisa kamu modifikasi supaya bisa menciptakan efek deformasi, sehingga animasi terasa lebih hidup dan dinamis (Witkin & Heck, 1994).

2. *Physics Constraints*

Kamu juga akan menggunakan *physics constraints* untuk menentukan batasan gerakan pada *mesh*, supaya deformasi tetap terlihat realistis. Batasan ini membantu menjaga kekuatan material dan membuat perilaku objek tetap alami (O'Brien & Hodgins, 1999).

1. *Particle Systems*

Untuk mengelola perilaku deformasi, kamu bisa menggunakan sistem partikel. Dalam sistem ini, setiap partikel bisa berinteraksi dengan partikel lain dan dipengaruhi oleh gaya eksternal, sehingga menghasilkan gerakan yang lebih alami dan kompleks (Carlson et al., 2004).

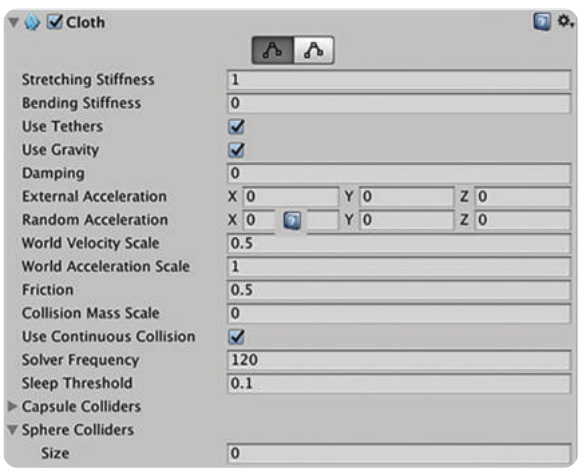
d Penggunaan Simulasi *Soft Body* dalam Gim

1. Karakter dan Efek Realistis

Dengan menggunakan simulasi *soft body*, kamu bisa membuat karakter bergerak dengan cara yang lebih alami. Ekspresi wajah dan gerakan tubuh yang halus jadi lebih mudah kamu capai, sehingga karakter terasa lebih hidup dan menarik perhatian pemain (Baker, 2016).

1. Kain dan Benda Lunak

Simulasi *soft body* juga penting banget untuk kamu terapkan pada objek seperti pakaian dan benda-benda lunak lainnya. Dengan teknik ini, kain bisa bergerak dengan cara yang dinamis dan natural, yang pastinya akan menambah kesan realisme dalam *gameplay* (James & Twigg, 2005).

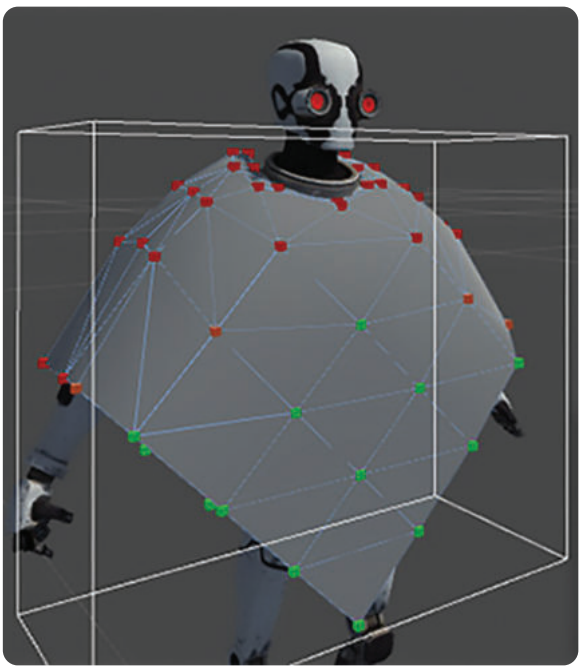


Gambar 5.29 Implementasi Komponen *Cloth*
Sumber: Muhammad Ardhan Fadhlurrrahman/ Kemendikbudristek (2024)

Berikut ini adalah contoh implementasi komponen *Cloth*, di mana kamu sudah bisa menemukan beberapa *setting* lanjutan seperti *stretching stiffness*, *bending stiffness*, dan pengaturan lainnya. Dengan menggunakan pengaturan ini, kamu bisa melakukan simulasi kain secara lebih efektif dan efisien.

e Perbandingan *Rigid* dan *Soft Body*

Aspek	*Rigid Body*	*Soft Body*
Deformasi	Tidak mengalami deformasi	Mengalami deformasi
Penghitungan	Lebih sederhana, berbasis titik dan rotasi	Lebih kompleks, memerlukan mesh dan partikel
Kinerja	Lebih ringan, lebih cepat	Lebih berat, membutuhkan lebih banyak sumber daya
Contoh	Mobil, bangunan	Kain, karakter, objek lunak lainnya



Gambar 5.30 Contoh karakter dengan komponen *Cloth*
Sumber: Unity/Unity (2024)

f Teknologi dan Alat

Dalam pengembangan animasi karakter dan simulasi fisika di dalam gim, kamu akan menggunakan berbagai teknologi dan alat untuk mendukung proses kerja. Salah satu yang paling penting adalah ***physics engine.*** Banyak gim modern menggunakan mesin fisika seperti Unity (dengan *Rigidbody* dan Softbody) atau *Unreal Engine* (dengan PhysX) untuk menangani simulasi fisika. Mesin ini menyediakan framework yang kamu perlukan untuk mengimplementasikan baik *rigid* dan *soft body simulations* (Kelley, 2017).

Selain engine, kamu juga akan bekerja dengan berbagai **algoritma** untuk menghitung interaksi dan gerakan antar objek. Contohnya, algoritma *Verlet* atau *damped springs* sering digunakan untuk simulasi *soft body*. Algoritma ini membantu kamu melakukan perhitungan yang lebih efisien dan akurat dalam simulasi (Baraff, 1994).

Tantangan dan Simulasi

Dalam menerapkan simulasi fisika, kamu akan menghadapi beberapa tantangan yang perlu diperhatikan agar hasil akhirnya tetap optimal. Salah satu tantangan utama adalah **kinerja**. Simulasi *soft body* seringkali lebih intensif secara komputasi dibandingkan *rigid body*. Hal ini dapat memengaruhi kinerja gim, terutama pada perangkat dengan sumber daya terbatas. Oleh karena itu, kamu harus melakukan optimisasi untuk menjaga kinerja gim (Reynolds, 2015).

Selain itu, kamu juga harus menjaga **stabilitas** simulasi. Menjaga simulasi tetap stabil dan realistis adalah tantangan tersendiri, terutama saat kamu bekerja dengan gim kompleks dengan banyak interaksi objek. Kamu sebagai pengembang perlu memastikan bahwa simulasi tidak menghasilkan artefak visual yang mengganggu (Hahn, 2016).

Tantangan lainnya adalah **penyesuaian** antara realisme fisika dan desain *gameplay*. Keseimbangan antara realisme fisika dan keseruan permainan sangat penting supaya gim yang kamu buat tetap menyenangkan dan menarik untuk dimainkan (Müller et al., 2018).

Aplikasi dalam *Gameplay*

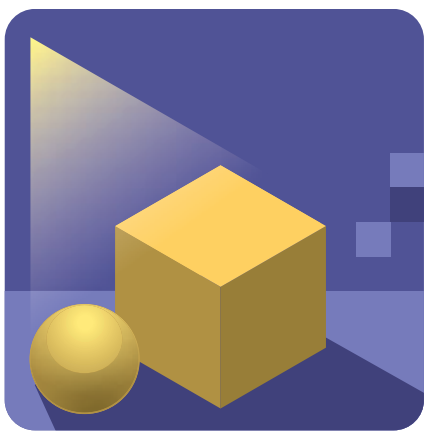
Simulasi fisika tidak hanya berfungsi sebagai elemen teknis, tetapi juga bisa kamu manfaatkan untuk memperkaya pengalaman bermain. Salah satu penerapan yang sering digunakan adalah dalam **puzzle dan teka-teki**. Dalam jenis permainan ini, interaksi antar objek berbasis fisika dapat menciptakan tantangan yang menarik.Misalnya, permainan yang melibatkan gerakan objek dengan menggunakan fisika dapat meningkatkan tantangan dan membuat pemain lebih terlibat (James & Twigg, 2005).

Selain itu, simulasi fisika juga berkontribusi besar terhadap **kenyamanan visual** dalam gim. Simulasi gerak digital memberikan efek visual yang menarik dan menambah realisme dalam *gameplay*. Dengan objek yang bergerak dan berinteraksi secara natural, pemain dapat merasakan pengalaman yang lebih imersif (O'Brien & Hodgins, 1999).

Aktivitas 5.5 **Praktik Mandiri**

Melanjutkan aktivitas sebelumnya, kamu bisa menggunakan desain yang sudah ada atau membuat desain yang berbeda. Setelah itu, buatlah simulasi gerak pada desainmu dengan mengikuti alur proses simulasi gerak yang telah dibahas sebelumnya. Kamu bisa menggunakan *engine* yang pernah kamu gunakan atau pelajari sebelumnya. Lalu presentasikanlah hasil kerjamu dengan teman yang lain untuk saling berbagi informasi.

5. Shading 3D



Setelah membahas tentang simulasi, sekarang saatnya kita mempelajari konsep *shading* dalam 3D. *Shading* adalah proses penting dalam pengembangan gim yang menentukan bagaimana permukaan objek terlihat saat terkena pencahayaan. Teknik ini digunakan untuk menghitung warna dan cahaya pada permukaan objek 3D berdasarkan beberapa faktor, seperti posisi sumber cahaya, warna objek, dan sudut pandang kamera. Tujuan utamanya adalah menciptakan ilusi kedalaman dan detail visual, sehingga objek terlihat lebih menarik dan realistis. Menurut Meyer et al. (2017), *shading* merupakan bagian penting dari proses *rendering* yang membantu membangun atmosfer visual dalam sebuah gim.

a Jenis Shading

1. *Flat Shading*

Flat shading adalah teknik paling dasar dalam *shading* 3D. Dalam teknik ini, setiap polygon dari objek diberi warna yang sama tanpa gradasi. Hasilnya adalah tampilan yang sangat sederhana dan tidak realistis. Meskipun begitu, kamu dapat menggunakan *flat shading* untuk gaya visual tertentu, seperti dalam gim retro atau kartun. López et al. (2019) mencatat bahwa *flat shading* memberikan kejelasan visual, meskipun tidak menonjolkan detail.



Gambar 5.31 Sebuah Objek 3D Menggunakan *Flat Shading*
Sumber: Mike Bailey dan Steve Cunningham/ Graphics Shaders: Theory and Practice (2012)

2. *Smooth Shading*

Smooth shading adalah teknik lanjutan yang lebih kompleks dibandingkan dengan *flat shading*. pada *smooth shading* ini dibagi menjadi beberapa jenis (Mike Bailey & Steve Cunningham, 2012), seperti:

a. Gouraud Shading



Gambar 5.32 Sebuah Objek 3D Menggunakan *Gouraud Shading*
Sumber: Mike Bailey dan Steve Cunningham/ Graphics Shaders: Theory and Practice (2012)

Gouraud shading meningkatkan *flat shading* dengan menghitung warna di setiap *vertex* dari objek. Warna ini kemudian diinterpolasi antara *vertex* untuk memberikan tampilan yang lebih halus. Meskipun *gouraud shading* lebih realistis dibandingkan *flat shading*, teknik ini dapat kehilangan detail pada area dengan pencahayaan yang kompleks. Menurut *Foley et al. (1996)*, *gouraud shading* sering digunakan ketika performa menjadi prioritas, karena walaupun *phong shading* memiliki ketajaman yang lebih tinggi tetapi cukup berat untuk dilakukan sehingga performa bisa menjadi menurun.

b. *Phong Shading*

Phong shading adalah teknik yang kamu gunakan untuk menghitung warna di setiap *pixel* (atau *fragment*) dengan cara menginterpolasi *normal* permukaan, bukan hanya warna di *vertex* seperti pada *Gouraud shading*. Teknik ini memberikan hasil visual yang lebih halus dan realistis, terutama untuk permukaan yang reflektif atau mengkilap. Watt dan Policarpo (2007) menjelaskan bahwa *Phong shading* sangat efektif untuk menciptakan objek yang terlihat lebih hidup dan menarik. Dibandingkan dengan *Gouraud shading*, *Phong shading* menawarkan detail pencahayaan yang lebih tajam dan presisi, tetapi membutuhkan proses komputasi yang lebih berat, sehingga bisa mempengaruhi performa jika digunakan dalam jumlah besar.



Gambar 5.33 Sebuah Objek Menggunakan *Phong Shading*
Sumber: Mike Bailey dan Steve Cunningham/ Graphics Shaders: Theory and Practice (2012)

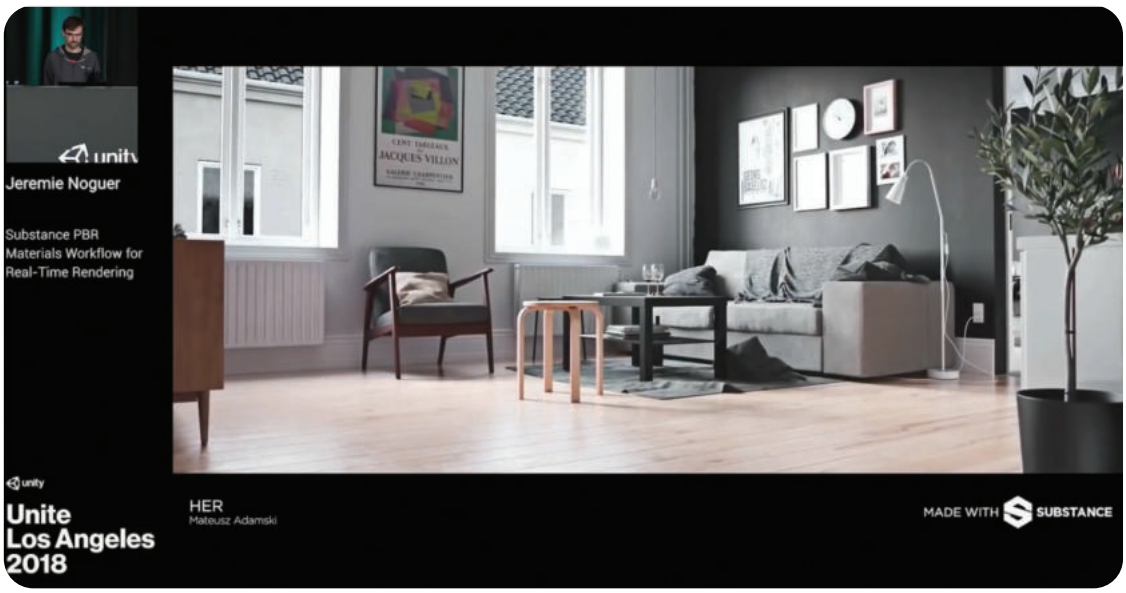
c. *Blinn-Phong Shading*

Blinn-Phong shading adalah variasi dari *Phong shading* yang bisa kamu gunakan untuk menghasilkan pencahayaan yang lebih efisien, terutama pada permukaan yang reflektif. Teknik ini menggunakan pendekatan pencahayaan yang sedikit berbeda agar highlight terlihat lebih tajam, sehingga meningkatkan kualitas visual tanpa membebani performa terlalu banyak. Bennett (2015) menjelaskan bahwa *Blinn-Phong shading* sering dipilih dalam pengembangan gim karena memberikan keseimbangan yang baik antara kualitas visual dan performa. Meskipun hasilnya tidak sedetail *Phong shading*, performanya lebih baik dan lebih ringan untuk dijalankan. Di dalam *game engine* Unity, *simple shading merupakan* salah satu implementasi dari *Blinn-Phong shading* yang bisa kamu gunakan dalam proyekmu.

d. *Physically Based Rendering* (PBR)

Physically Based Rendering (PBR) adalah pendekatan modern dalam *shading* yang bisa kamu gunakan untuk menciptakan tampilan visual yang sangat realistis. Teknik ini menggunakan model pencahayaan berbasis fisika nyata, dan mempertimbangkan berbagai properti material seperti *albedo*, *metalness*, dan *roughness*. Dengan mempertimbangkan bagaimana

cahaya berinteraksi dengan permukaan dalam kondisi yang berbeda, PBR memungkinkan kamu menghasilkan visual yang mendekati kenyataan atau sering disebut *photorealism*. Zhang et al. (2018) mencatat bahwa PBR telah menjadi standar baru di industri gim karena kemampuannya menghadirkan tampilan yang kaya dan mendetail. Meskipun perhitungannya lebih kompleks dan cukup berat secara performa, hasil visual yang kamu dapatkan akan jauh lebih imersif dan realistis.



Gambar 5.34 *Physically Based Rendering* (PBR) Pada Konferensi *Unity*
Sumber: Unity (2018)

b Teknik Pencahayaan

Dalam pengembangan gim, pencahayaan memiliki peran penting dalam menciptakan suasana dan memperkuat pengalaman visual. Beberapa teknik pencahayaan dasar bisa kamu gunakan untuk mencapai efek yang diinginkan, tergantung pada kebutuhan desain dan konteks adegan.

1. *Ambient Light*

Ambient light adalah jenis pencahayaan dasar yang bisa kamu gunakan untuk menerangi seluruh *scene* secara merata, tanpa arah cahaya yang spesifik. Teknik ini membantu menciptakan suasana yang seimbang dan mencegah adanya area yang terlalu gelap dalam permainan. Karena tidak bergantung pada posisi sumber cahaya, *ambient light* sering digunakan sebagai

pencahayaan pendukung dalam gim. Müller et al. (2018) mencatat bahwa *ambient light* sangat penting untuk menjaga pencahayaan tetap merata dan nyaman dilihat oleh pemain.

2. *Directional Light*

Directional light adalah jenis pencahayaan yang bisa kamu gunakan untuk mensimulasikan cahaya dari sumber yang sangat jauh, seperti matahari. Cahaya ini datang dari satu arah yang konsisten, sehingga menghasilkan bayangan yang tajam dan jelas. Teknik ini sangat efektif untuk menciptakan pencahayaan luar ruangan yang realistis. Menurut Kelley (2017), *directional light* juga bisa kamu manfaatkan untuk menghadirkan efek dramatis dalam *gameplay*, terutama dalam adegan-adegan yang membutuhkan nuansa kuat dari pencahayaan.

3. *Point Light*

Point light adalah sumber cahaya yang memancarkan cahaya ke segala arah dari satu titik tertentu. Jenis pencahayaan ini menciptakan efek cahaya yang lebih lembut dan merata, sehingga cocok kamu gunakan untuk pencahayaan dalam ruangan. Menurut Baraff & Witkin (1998), *point light* sering dimanfaatkan dalam gim untuk menciptakan atmosfer yang lebih intim dan hangat, seperti suasana di kamar atau ruangan tertutup.

4. *Spotlight*

Spotlight adalah jenis cahaya yang memancarkan cahaya secara terfokus ke area tertentu, mirip seperti senter. Kamu bisa menggunakan teknik ini untuk menciptakan efek dramatis atau mengarahkan perhatian pemain ke elemen-elemen penting dalam gim. Menurut Reynolds (2015), *spotlight* sangat efektif untuk meningkatkan imersi pemain karena dapat menciptakan fokus visual yang kuat dalam *scene*.

5. *Area Light*

Area light adalah sumber cahaya yang memiliki dimensi fisik dan memancarkan cahaya dari area tertentu, bukan hanya dari satu titik. Cahaya ini menghasilkan bayangan yang lebih lembut dan realistis, serta memberikan detail tambahan pada lingkungan gim. Kamu bisa menggunakan *area light* untuk menciptakan suasana pencahayaan yang lebih kompleks dan natural. Fowler & Turing (2019) mencatat bahwa *area light* sangat berguna untuk menghadirkan efek visual yang kaya dan realistis dalam berbagai skenario pencahayaan.

c Material dan *Textures*

Dalam dunia pengembangan gim, material dan tekstur berperan penting dalam menentukan bagaimana sebuah objek tampil di layar. Kedua elemen ini bekerja sama untuk menciptakan visual yang realistis, menarik, dan sesuai dengan suasana permainan yang kamu bangun.

1. *Material Properties*

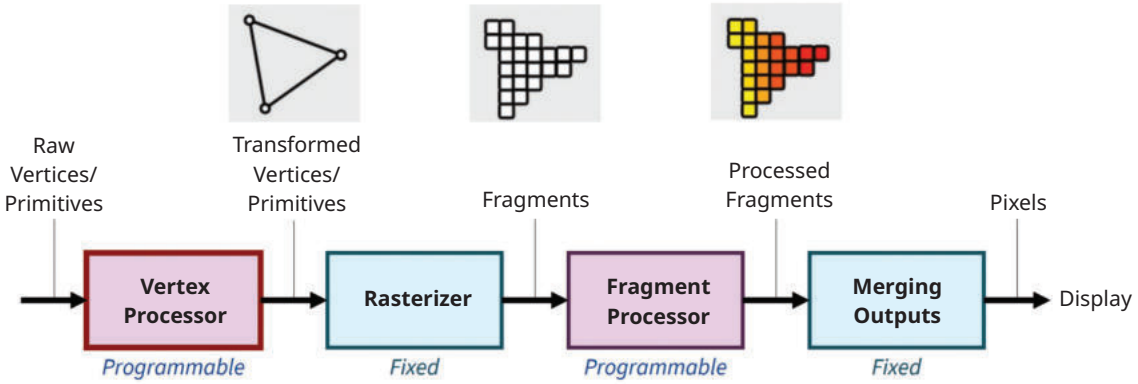
Material properties adalah pengaturan yang menentukan bagaimana permukaan objek berinteraksi dengan cahaya. Ini mencakup aspek seperti warna dasar (*albedo*), tingkat reflektivitas, dan kekasaran (*roughness*). Dengan memilih dan mengatur material yang tepat, kamu bisa memengaruhi tampilan visual keseluruhan dari objek dalam gim. Menurut Erleben et al. (2009), memahami cara kerja material sangat penting kalau kamu ingin menciptakan efek visual yang meyakinkan dan realistis.

2. *Texture Maps*

Texture maps menggunakan gambar (tekstur) untuk memberikan detail tambahan pada permukaan objek. Ini termasuk diffuse maps, normal maps, specular maps, dan lainnya. Texture maps membantu menciptakan tampilan yang kaya dan bervariasi, sehingga meningkatkan realisme visual objek. *Hahn (2016)* menjelaskan bahwa tekstur yang baik dapat membuat perbedaan besar dalam visualisasi objek.

d *Shader Programming*

Untuk memahami secara garis besar sebuah *pipeline rendering* maka kita harus mengerti dulu terkait *shader*. Apakah itu *shader*? *shader* adalah sebuah program yang berjalan pada kartu grafis atau disebut GPU untuk memanipulasi sebuah scene 3D sebelum gambar ditunjukan pada layar. *Shader* ini juga dapat membentuk sebuah efek *rendering* yang sangat cepat dan biasanya bisa dipakai untuk sebuah produk interaktif seperti gim. Youngdo Lee, dalam websitenya menjelaskan tentang proses untuk merender suatu objek ke dalam layar, seperti:



Gambar 5.35 *Pipeline Rendering* Sebuah Gambar Tiga Dimensi
sumber: Youngdo Lee (2024)

1. *Vertex Shaders*



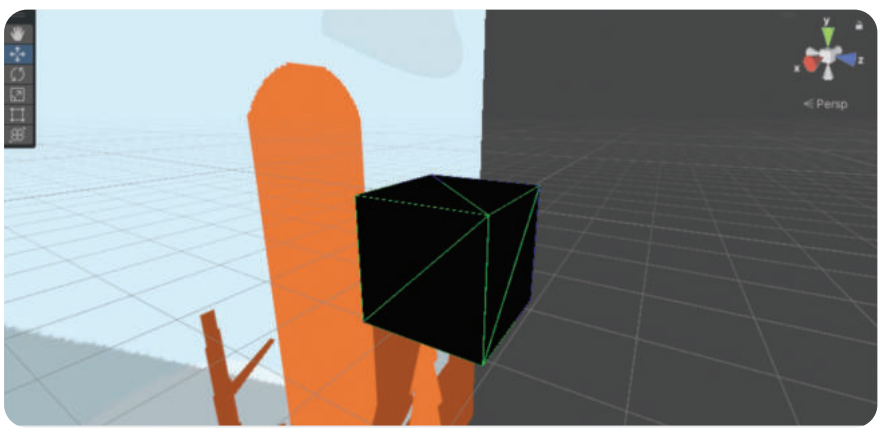
Gambar 5.36 Contoh Penggunaan *Vertex Shader*
Sumber: Game dev Academy (2015)

Vertex shader adalah bagian penting dalam *rendering pipeline* yang kamu gunakan untuk menghitung posisi *vertex* dalam ruang 3D serta memanipulasi data *vertex* sebelum masuk ke tahap *rasterisasi*. Proses ini memungkinkan kamu menciptakan efek visual yang dinamis pada objek, seperti deformasi, animasi, atau perubahan bentuk berdasarkan parameter tertentu. McGuire et al. (2012) mencatat bahwa *vertex shaders* memainkan peran krusial dalam visualisasi real-time. Dalam Unity Documentation (2024), dijelaskan bahwa *vertex shader* berfokus untuk memodifikasi bentuk geometri di level atau *scene*. Shader ini akan dieksekusi pada setiap *vertex* dan menghasilkan output berupa koordinat proyeksi, warna, tekstur, dan data lain

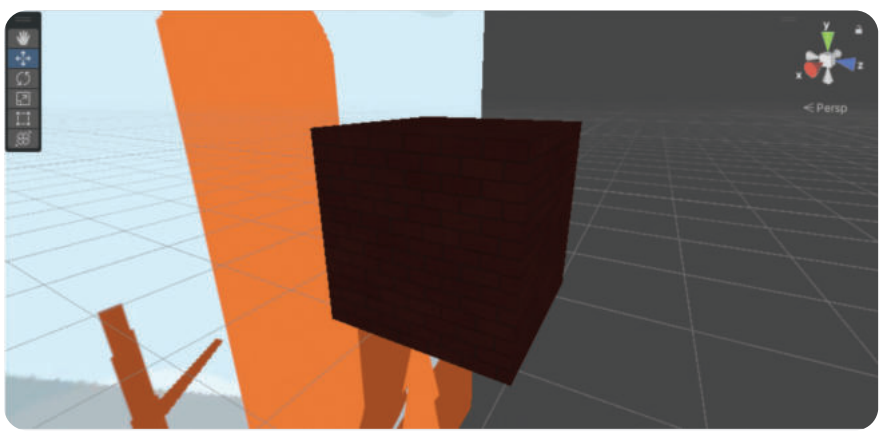
yang nantinya akan diproses lebih lanjut oleh *fragment shader*. Berikut adalah contoh dari perubahan menggunakan vertex shader, objek tetap sama tetapi dapat dimanipulasi sehingga menimbulkan perubahan bentuk pada geometri 3D

2. *Fragment Shaders (Pixel Shaders)*

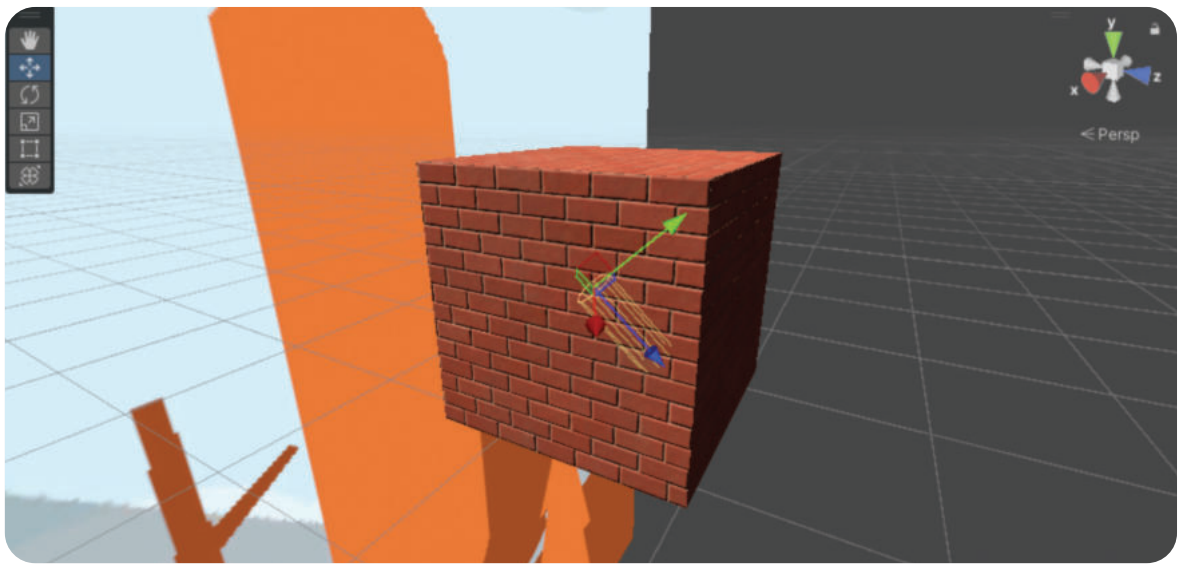
Fragment shader adalah bagian dari proses rendering yang kamu gunakan untuk menghitung warna akhir dari setiap pixel, berdasarkan pencahayaan, material, dan informasi tekstur. Shader ini dieksekusi untuk setiap pixel, sehingga memungkinkan kamu menerapkan shading yang lebih kompleks dan detail yang halus pada permukaan objek. Menurut Bennett (2015), fragment shaders memberikan fleksibilitas besar dalam menciptakan berbagai efek visual, mulai dari bayangan lembut hingga pencahayaan real-time yang realistis. Secara umum, fragment shader berfungsi memodifikasi properti visual sebuah gambar dalam program, menjadikannya tempat utama untuk menyempurnakan tampilan akhir dari sebuah objek di layar.



Gambar 5.37 Contoh Objek 3D yang Belum Masuk ke dalam *Pixel Processor* Untuk *Fragment Shader*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)



Gambar 5.38 Contoh Objek 3D Yang Sudah Di Proses Pixel *Processor* Untuk *Fragment Shader* Tanpa Cahaya
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)



Gambar 5.39 Contoh Objek 3D yang Sudah Diproses *Pixel Processor* untuk *Fragment Shader* Dengan Cahaya.
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

e Penggunaan dalam Gim

Teknik *shading* memainkan peran besar dalam memperkuat tampilan visual dan pengalaman bermain dalam gim. Dengan kombinasi pencahayaan, material, dan efek visual yang tepat, kamu bisa menciptakan dunia gim yang tidak hanya menarik secara estetika, tetapi juga imersif secara emosional. Berikut beberapa penggunaan shading dalam gim:

1. Penciptaan Atmosfer



Gambar 5.40 Penggunaan *Shader* Pada Gim *Journey*.
Sumber: ThatGameCompany (2012)

Shading yang baik dapat menciptakan suasana dan meningkatkan pengalaman imersif pemain. Dengan teknik pencahayaan yang tepat dan material yang sesuai, kamu dapat menciptakan dunia yang terasa hidup dan menarik. Meyer et al. (2017) menekankan bahwa atmosfir yang diciptakan oleh *shading* dapat sangat mempengaruhi pengalaman pemain. *Journey* adalah contoh gim yang menggunakan *shader* untuk menciptakan atmosfir seperti dunia mimpi sehingga pemain merasa nyaman dan santai saat memaikan gim tersebut.

2. Realisme



Gambar 5.41 Penerapan Teknik *Shading* PBR Pada Gim Gran Turismo 7
Sumber: Polyphony Digital Inc. / Sony Interactive Entertainment (2022)

Penerapan teknik *shading* yang canggih, seperti PBR, membantu kamu menciptakan objek dan lingkungan yang lebih realistis. Dengan detail yang lebih dalam dan pencahayaan yang menyerupai dunia nyata, pemain bisa merasakan pengalaman visual yang jauh lebih imersif saat menjelajahi dunia gim. Foley et al. (1996) menjelaskan bahwa realisme visual memegang peran penting dalam meningkatkan pengalaman pengguna secara keseluruhan. *Gran Turismo 7* merupakan contoh penerapan teknik *shading* seperti PBR, yang memberikan pengalaman simulasi balap yang sangat mirip dengan dunia nyata berkat kualitas visual yang sangat tinggi.

3. Efek Khusus



Gambar 5.42 Contoh Gim yang Menggunakan *Shader*
Sumber: Anatarupa Studio (2020)

Shading juga digunakan untuk menciptakan efek khusus, seperti bayangan dinamis, refleksi, dan transparansi. Efek ini meningkatkan visualisasi dalam gim dan memberikan pengalaman bermain yang lebih menyeluruh. *Erleben et al. (2009)* mencatat bahwa efek visual yang menarik dapat meningkatkan keterlibatan pemain. *Lokapala* merupakan salah satu contoh game yang menggunakan shader sebagai efek khusus. Ketika karakter pemain mengeluarkan skill, maka akan terdapat sebuah efek visual. Hal ini sesuai dengan yang disebutkan Erleben et al. (2009*)* bahwa efek visual yang menarik meningkatkan keterlibatan pemain dan diharapkan dapat membuat pengalaman bermain menjadi lebih menyenangkan.

f Optimasi Shading

Dalam pengembangan gim, *shading* memainkan peran besar dalam menciptakan visual yang menarik, namun proses ini juga bisa berdampak pada kinerja permainan. Oleh karena itu, kamu perlu melakukan optimasi *shading* agar gim tetap berjalan lancar tanpa harus mengorbankan kualitas tampilan.

Salah satu hal yang perlu kamu perhatikan adalah **performance considerations**. Proses *shading* dapat mempengaruhi kinerja gim. Kamu mungkin akan sering mencari keseimbangan antara kualitas visual dan kinerja, terutama di platform dengan sumber daya terbatas. Optimasi *shading* sangat penting untuk memastikan bahwa gim tetap berjalan lancar tanpa mengorbankan kualitas visual. McGuire et al. (2012) menekankan pentingnya optimasi dalam pengembangan gim modern.

Selain itu, kamu juga bisa menggunakan teknik **Level of Detail (LOD)** sebagai strategi optimasi. *Level of Detail* (LOD) adalah teknik yang bisa kamu gunakan untuk meningkatkan performa gim tanpa harus mengorbankan kualitas visual secara keseluruhan. Dengan menggunakan model dan teknik *shading* yang berbeda tergantung pada jarak objek dari kamera, kamu bisa mengurangi kompleksitas objek yang jauh dari pandangan. Hal ini membantu proses *rendering* menjadi lebih ringan dan menjaga performa tetap stabil. Hahn (2016) menjelaskan bahwa LOD adalah salah satu strategi yang sangat efektif dalam pengembangan gim, terutama ketika kamu ingin menjaga keseimbangan antara visual yang memukau dan performa yang optimal.

Pada contoh berkut ini kita akan membuat efek vignette, dimana efeknya tersebut membuat gambar yang direnderberbentuk bulat dan memiliki efek hitam di pinggir gambarnya. Bagaimana cara membuatnya? Berikut contoh cuplikan kode untuk efek *vignette*:

```
Shader “Custom/WindowCoordinates/Vignetting” {
    SubShader {
        Pass {
            CGPROGRAM

            #pragma vertex vert
            #pragma fragment frag
            #pragma target 3.0

            #include “UnityCG.cginc”

            float4 vert(appdata_base v) : POSITION {
                return mul (UNITY_MATRIX_MVP, v.vertex);
            }

            fixed4 frag(float4 sp:VPOS) : SV_Target {
                float2 wcoord = sp.xy/_ScreenParams.xy;
                float vig = clamp(3.0*length(wcoord-0.5),0.0,1.0);
                return lerp (fixed4(wcoord,0.0,1.0),fixed4(0.3,0.3,0.3,1.0),vig);
            }
            ENDCG
        }
    }
}
```

hasil shader vignette berdasarkan cuplikan kode:



Gambar 5.43 Contoh *Output* Pemrograman *Shader* di dalam *Game Engine.*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Dengan kode *shader* ini seluruh gambar akan berada pada bagian warna warni sehingga *scene* 3D akan memiliki batas hitam di pinggir gambar. Hasil shader *vignette* berdasarkan cuplikan kode:

Aktivitas 5.6 Praktik Mandiri

Melanjutkan aktivitas sebelumnya kamu bisa menggunakan desain ini atau membuat desain yang berbeda. Kemudian buatlah *shading* 3D pada desainmu sendiri dengan mengikuti alur proses *shading* 3D yang telah dibahas sebelumnya. Kamu bisa menggunakan *engine* yang pernah kamu gunakan. Lalu presentasikanlah hasil kerjamu dengan teman yang lain untuk saling berbagi informasi

Aktivitas 5.7 Praktik Portofolio Tim

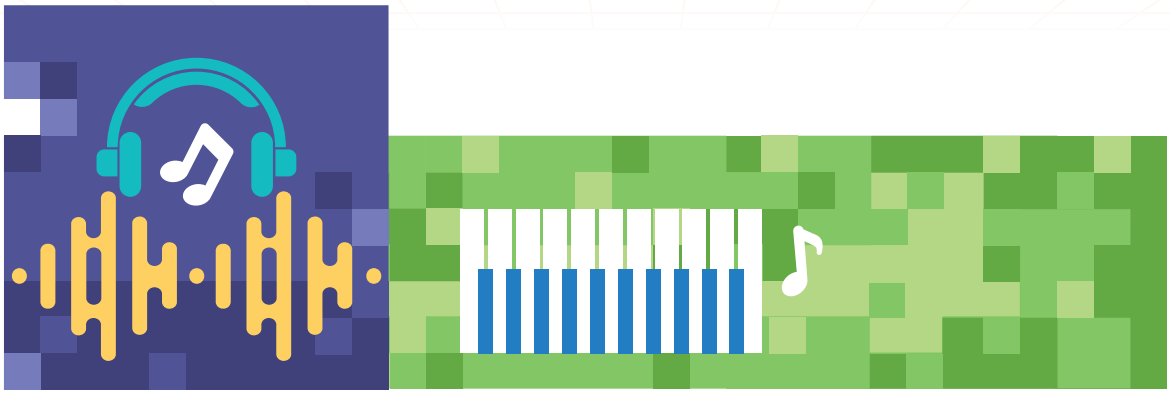
Dalam aktivitas ini, kalian akan bekerja dalam tim yang terdiri dari empat orang untuk membuat portofolio aset visual 3D. Ikuti setiap langkah yang telah dibahas dalam materi Aset Visual 3D dan hasilkan karya terbaik yang akan dimasukkan dalam portofolio tim. Portofolio ini akan mencakup berbagai tahapan pembuatan aset visual 3D yaitu *texturing*, *rigging*, akting pergerakan karakter, simulasi gerak, dan shading 3D.

Portofolio yang disusun harus mencakup bagian berikut ini:

1. Judul Proyek: Nama proyek portofolio.
2. Deskripsi Singkat: Penjelasan singkat mengenai proyek, tujuan, dan peran setiap anggota tim.
3. Dokumentasi dan Penjelasan setiap Tahapan Pembuatan

Setelah selesai, kumpulkan portofolio tim dalam bentuk digital dan presentasikan hasilnya di depan kelas. Pastikan portofolio kalian mencerminkan kerja tim yang baik dan menunjukkan keterampilan kalian dalam mengembangkan aset visual 3D.

C. Aset Audio



Setelah membahas aspek visual dalam pengembangan gim, penting untuk beralih ke aset audio, yang juga memainkan peran vital dalam menciptakan pengalaman imersif dan emosional bagi pemain. Aset audio mencakup efek suara dan musik latar yang dapat meningkatkan atmosfer, mendukung gameplay, dan memperkaya narasi.

Desain audio adalah langkah pertama yang krusial dalam pengembangan aset audio. Pada tahap ini, kamu harus harus bisa menentukan tujuan audio dalam gim. Apa suasana yang ingin kamu ciptakan? Apakah audio berfungsi untuk mendukung *gameplay*, ataukah juga untuk memperkaya narasi? kemudian kamu bisa melakukan identifikasi jenis audio yang diperlukan. Jenis audio yang umum digunakan dalam gim termasuk efek suara untuk tindakan karakter, suara lingkungan, dan musik latar untuk berbagai situasi dalam gim.

1. Efek Suara



Efek suara atau *sound effects* (SFX) adalah elemen penting dalam gim yang membantu memperkuat suasana dan memberikan umpan balik audio kepada pemain. Untuk menghasilkan efek suara yang berkualitas, kamu perlu melakukan proses yang terstruktur, mulai dari perekaman hingga implementasi ke dalam gim.

a Perekaman dan Pengumpulan

Penggunaan teknik perekaman suara yang baik sangat penting untuk menghasilkan efek suara berkualitas. kamu juga dapat mengumpulkan efek suara dari perpustakaan suara atau menggunakan perangkat lunak sintetis.

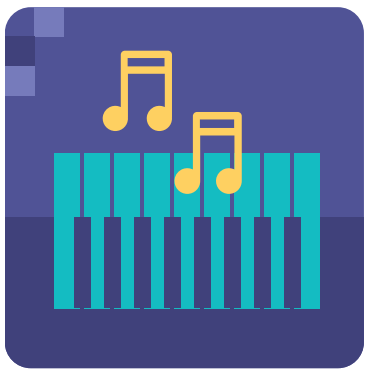
b Pengeditan dan Pemrosesan

Setelah perekaman, langkah selanjutnya adalah pengeditan dan pemrosesan efek suara. kamu bisa menggunakan perangkat lunak audio seperti *Adobe Audition, Audacity*, atau *Pro Tools* untuk mengedit dan memproses suara. Hal ini termasuk pemotongan, pengaturan level, dan penambahan efek seperti *reverb* atau *echo*.

c Pengeditan dan Pemrosesan

Pertimbangkan untuk menggunakan sistem audio yang mendukung pemicu dinamis, sehingga efek suara dapat berubah berdasarkan konteks permainan. Misalnya, suara langkah yang berbeda dapat digunakan di permukaan yang berbeda, seperti tanah, batu, atau air.

2. Musik Latar



Musik latar memainkan peran besar dalam membentuk suasana dan emosi selama permainan. Dengan memilih atau menciptakan musik yang tepat, kamu bisa meningkatkan keterlibatan pemain dan memperkuat narasi yang dibangun dalam gim.

Jika kamu memilih untuk membuat musik asli, mulailah dari tahap **komposisi musik**. Di sini, kamu perlu mempertimbangkan tema, emosi, dan ritme yang sesuai dengan adegan dalam gim. Musik saat pertempuran tentu akan berbeda dengan musik saat eksplorasi atau adegan emosional. Hahn (2016) menjelaskan bahwa musik yang baik bisa mengubah suasana permainan dan memperdalam pengalaman emosional pemain.

Selain komposisi, kamu juga harus memperhatikan aspek **looping dan transisi**. Musik latar harus bisa di-*loop* dengan mulus agar tidak terasa terputus saat dimainkan berulang.

Transisi antar bagian musik juga harus berjalan lancar, terutama ketika *gameplay* berpindah dari satu situasi ke situasi lain. Menurut Müller et al. (2018), transisi musik yang mulus membantu menjaga fokus dan imersi pemain.

Jika kamu memilih untuk menggunakan musik yang sudah ada, pastikan kamu memahami pentingnya **penggunaan musik lisensi**. Menggunakan musik tanpa izin bisa menimbulkan masalah hukum yang serius. Oleh karena itu, kamu perlu mendapatkan lisensi yang sah sebelum memasukkan musik tersebut ke dalam gim. Foley et al. (1996) menekankan bahwa pengembang harus berhati-hati dalam penggunaan musik berhak cipta demi menjaga integritas dan kelancaran proyek.

3. Implementasi Audio dalam Gim

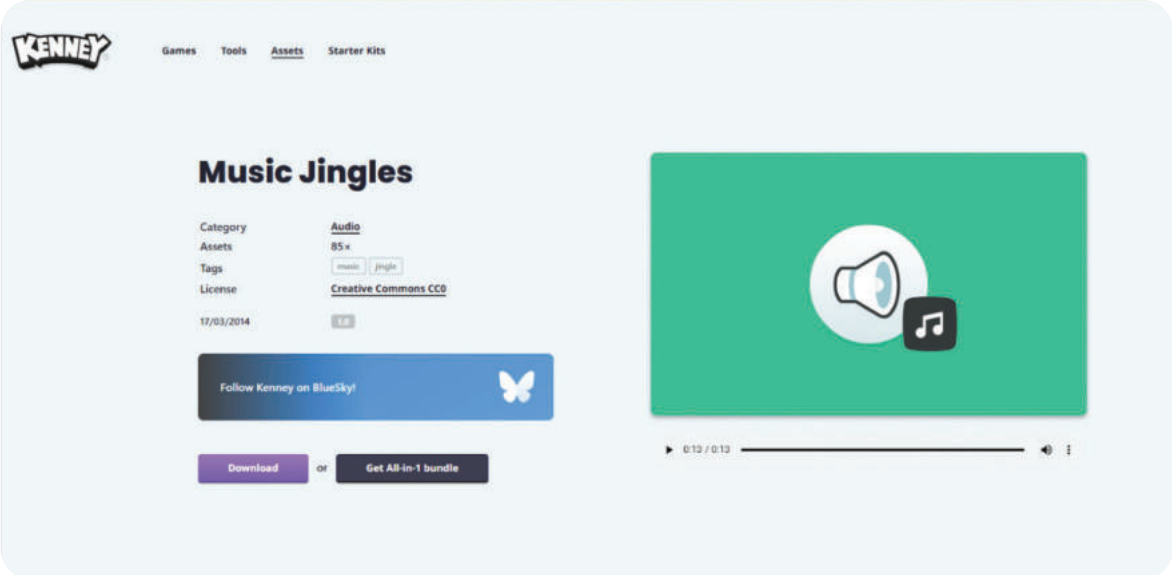


Setelah efek suara dan musik latar siap, langkah berikutnya adalah mengimplementasikan semua aset audio ke dalam gim. Proses ini memerlukan perencanaan yang baik agar suara tidak hanya terdengar jelas, tetapi juga terintegrasi secara harmonis dengan alur permainan.

Langkah pertama adalah memilih **game engine** yang mendukung pengelolaan audio secara optimal. Unity dan Unreal Engine adalah dua contoh *engine* yang memiliki alat untuk mengintegrasikan dan mengelola aset audio secara efektif. Erleben et al. (2009) menunjukkan bahwa pemilihan *engine* yang tepat dapat mempengaruhi efisiensi dan kualitas audio dalam gim.

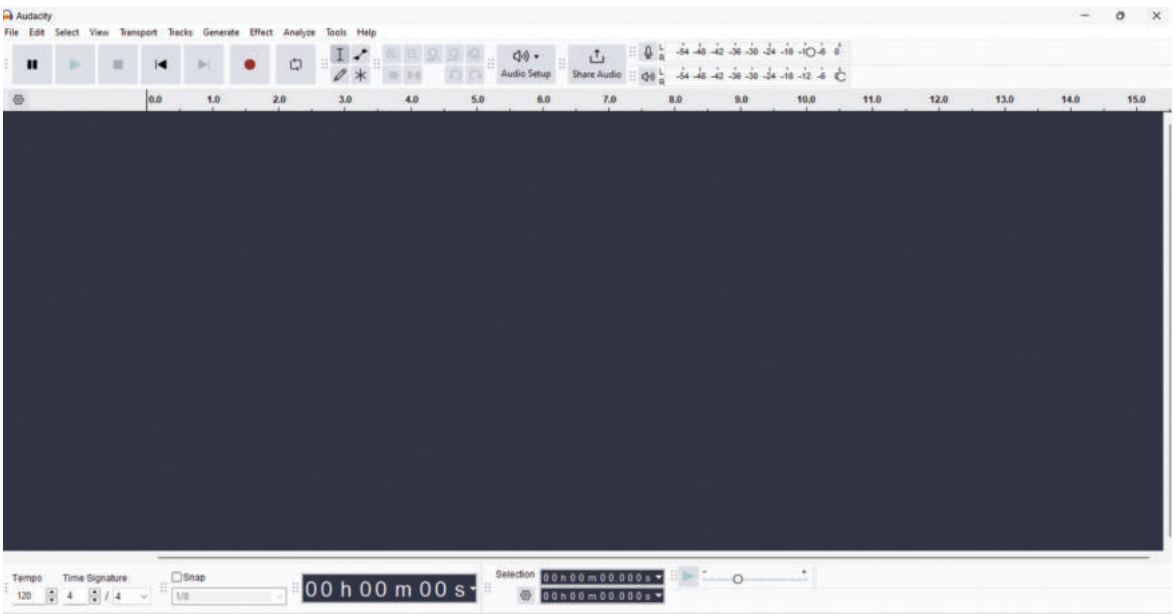
Selain mengandalkan fitur bawaan dari engine, kamu juga bisa menggunakan **audio middleware** seperti *FMOD* atau *Wwise*. Alat ini memungkinkan kamu untuk mengelola audio dengan lebih fleksibel dan memberikan fitur seperti *mixing*, *spatialization*, dan pemicu dinamis. McGuire et al. (2012) mencatat bahwa penggunaan *middleware* dapat menghemat waktu dan meningkatkan kualitas audio secara keseluruhan.

Pada. pembuatan aset audio ini untuk gim Mr. Alien. Aplikasi yang digunakan adalah Audacity, tetapi kamu bisa menggunakan aplikasi lain yang sudah pernah kamu pelajari. Contoh kali ini kita akan mencoba membuat suatu sampel yang merupakan gabungan dari sampel gratis yang bisa kamu dapatkan di internet sebagai contoh di dalam sini terdapat kumpulan audio gratis di dalam website bernama Kenny seperti di gambar:



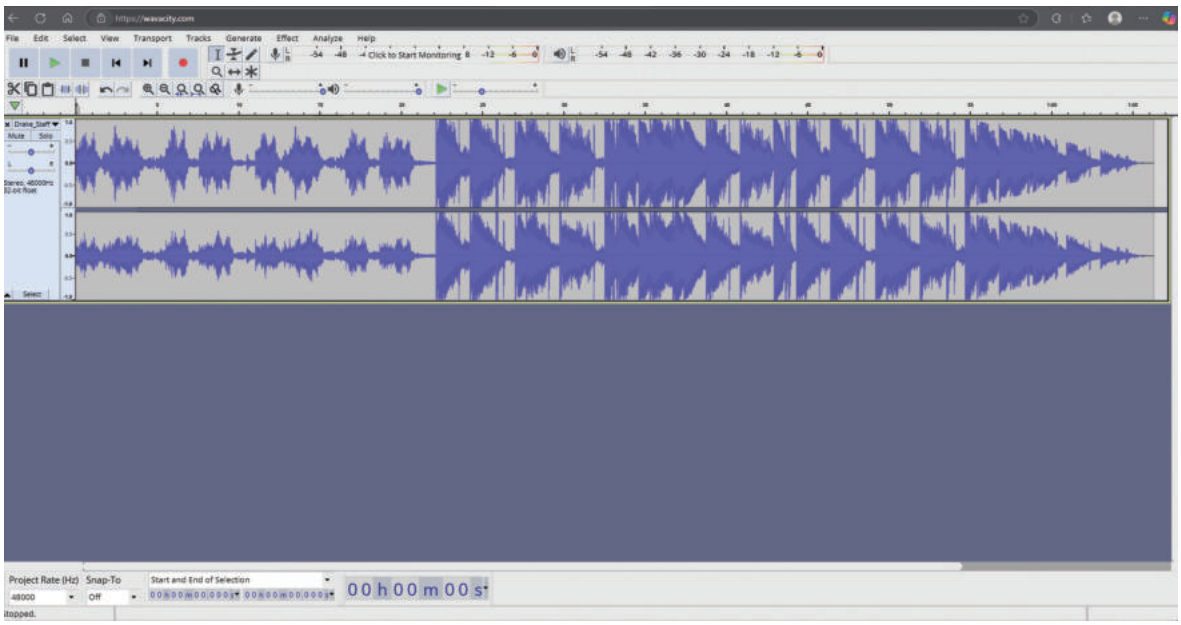
Gambar 5.44 Musik Kenny untuk Audio Gim
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Selanjutnya, kita akan menggunakan program bernama *Audacity* untuk melakukan manipulasi dari audio tersebut. Kita akan memasukan beberapa sampel ke dalam program *Audacity.*



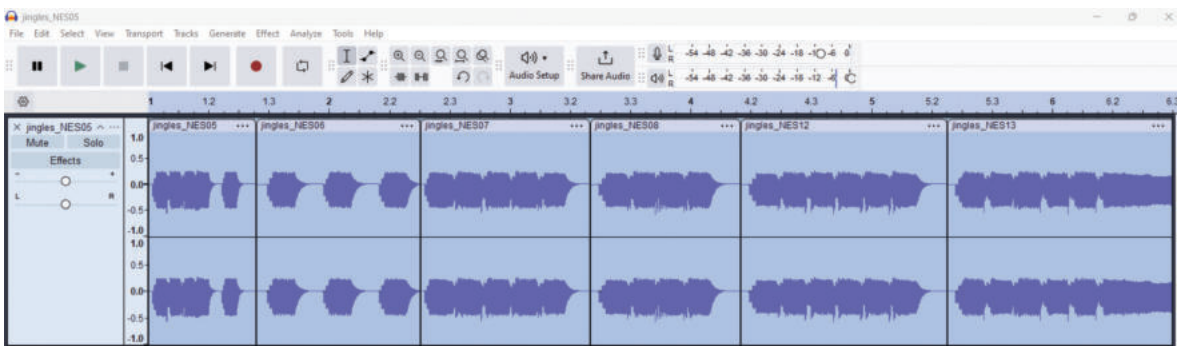
Gambar 5.45 Tampilan Awal dari Program Audacity
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Berikut contoh lain menggunakan program bernama *Wavacity* untuk melakukan manipulasi dari audio tersebut. Kita akan memasukan beberapa sampel ke dalam program *Wavacity.*



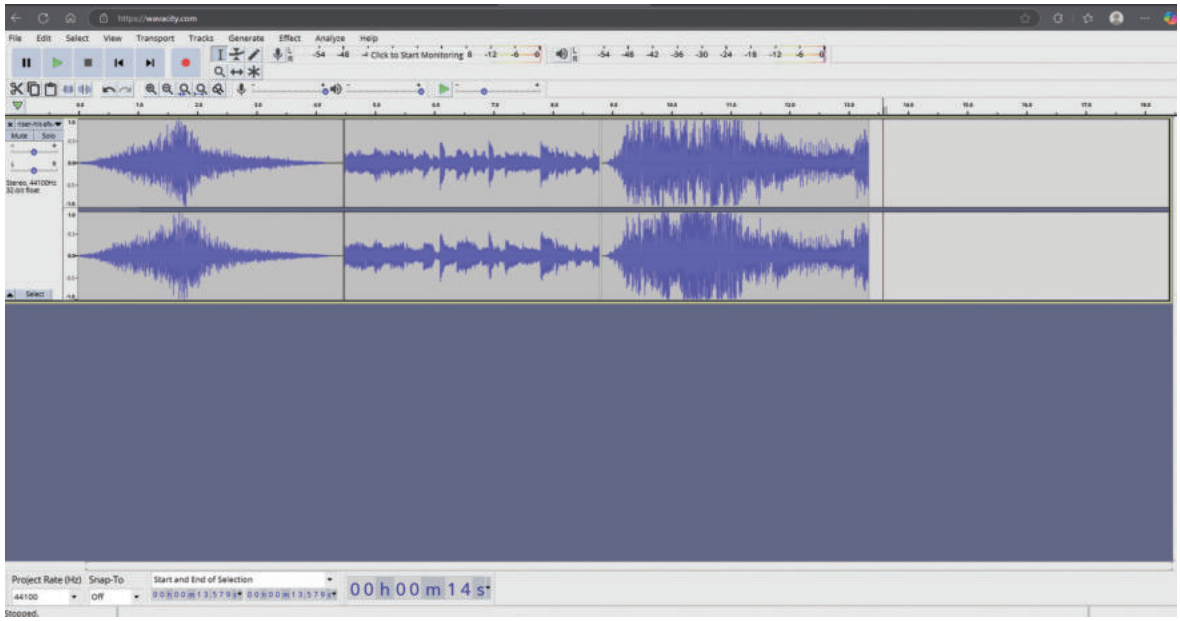
Gambar 5.46 Tampilan Awal dari Program wavacity
Sumber: Muhammad Ardhan Fadhlurrahman / Kemendikbudristek (2025)

Kamu bisa menyatukan beberapa sampel audio tersebut menjadi satu file untuk menjadi musik audio untuk gim mr.alien.



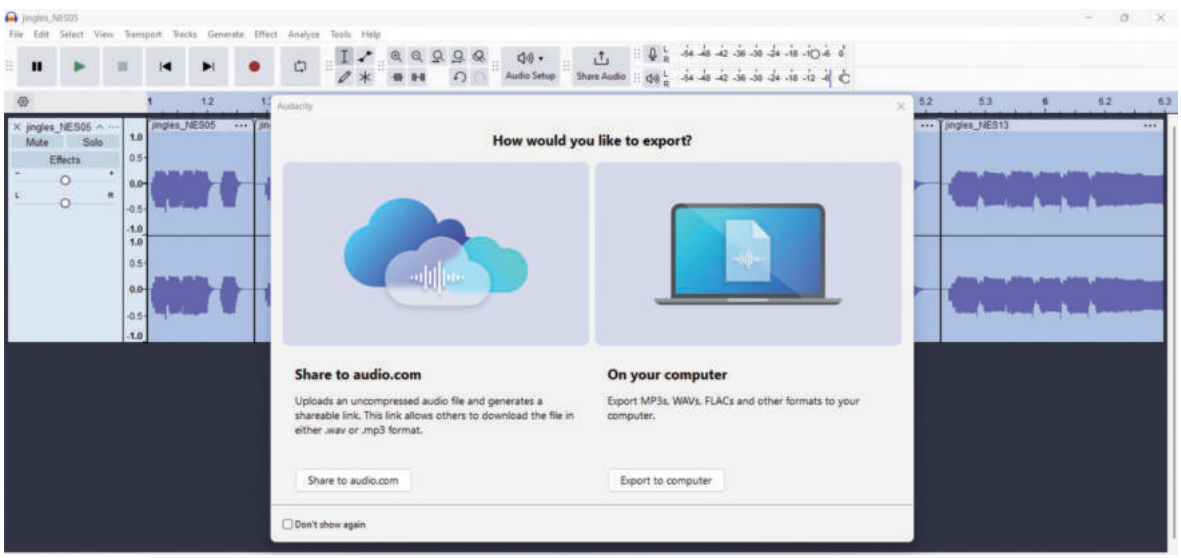
Gambar 5.47 Tampilan Saat Menyatukan Beberapa Sampel Audio di Dalam Program Audacity
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Berikut tampilan jika menggunakan Wavacity dan menyatukan beberapa sampel audio tersebut menjadi satu file untuk menjadi musik audio untuk gim mr. alien.



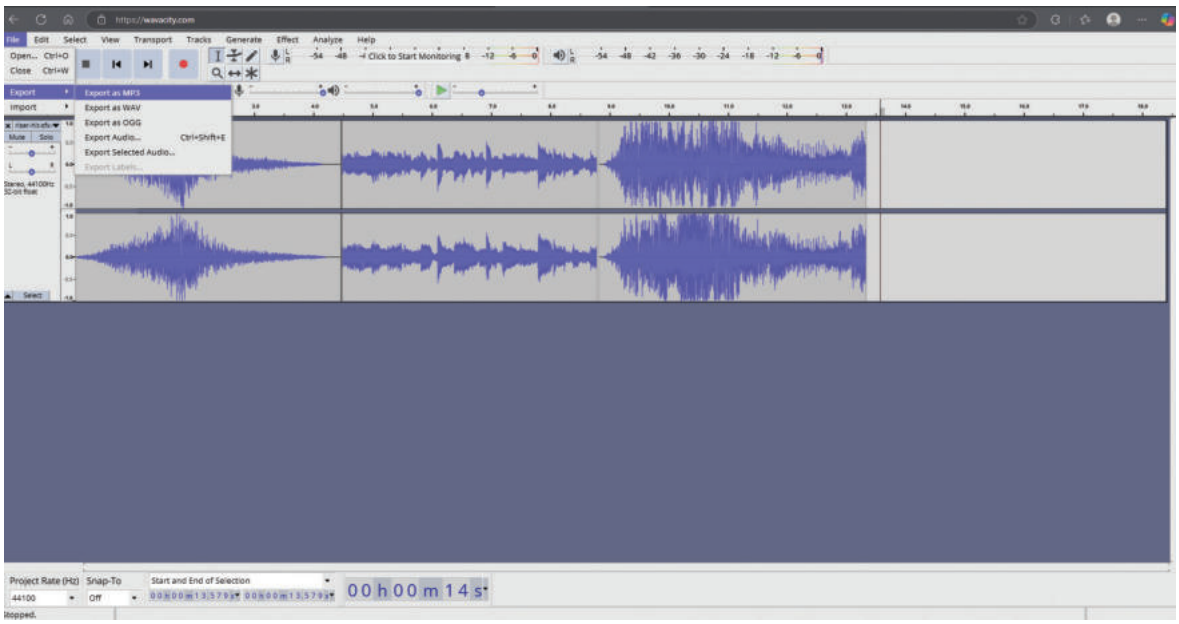
Gambar 5.48 Tampilan Saat Menyatukan Beberapa Sampel Audio di Dalam Program Wavacity
Sumber: Muhammad Ardhan Fadhlurrahman / Kemendikbudristek (2025)

kamu bisa melakukan export hasilnya setelah proses edit dilakukan. export yang kamu lakukan akan dimasukkan ke dalam game engine



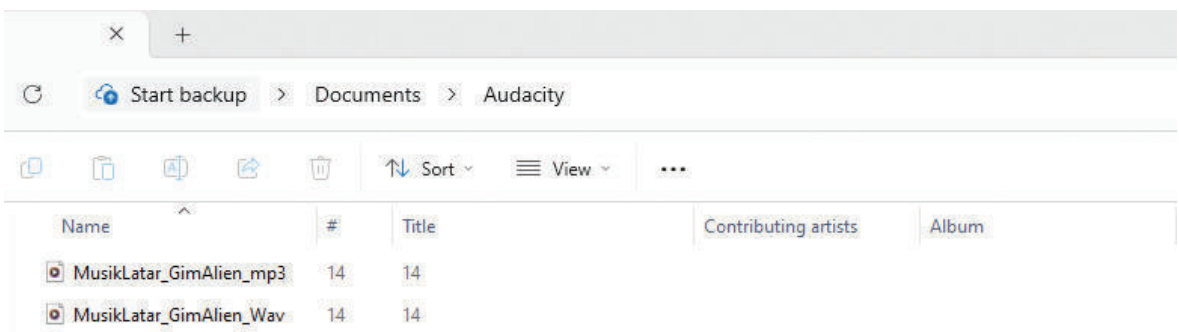
Gambar 5.49 Tampilan Saat Ingin Melakukan *Export*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

Setelah puas dengan hasil sampel, kamu bisa melakukan export hasilnya setelah proses edit di Wavacity seperti berikut



Gambar 5.50 Tampilan Saat Ingin Melakukan *Export di Wavacity*
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2025)

Terdapat dua jenis format yang di *export* yaitu wav dan mp3 yang keduanya merupakan format populer dari audio untuk musik latar. Hasil output akan memiliki format yang sama dengan berbagai aplikasi. Kamu akan melakukan *export* dari audacity dalam 2 format (mp3 dan wav) seperti gambar berikut:



Gambar 5.51 Contoh Hasil Audio Export dari Program Audio yang Digunakan
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

4. Pengujian Audio



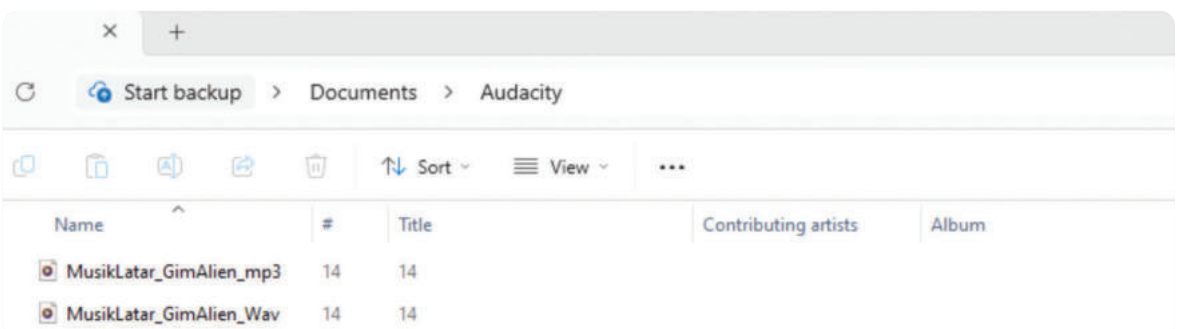
Uji semua efek suara dan musik dalam konteks *gameplay* untuk memastikan mereka berfungsi dengan baik dan tidak mengganggu pengalaman bermain. Uji dalam berbagai situasi dan kondisi untuk mendapatkan gambaran yang lebih jelas tentang bagaimana audio terdengar. Meyer et al. (2017) menyatakan bahwa pengujian dalam konteks sangat penting untuk mencapai kualitas audio yang optimal.

Kumpulkan umpan balik dari tim dan penguji untuk mengevaluasi bagaimana audio berkontribusi terhadap pengalaman keseluruhan. Lakukan penyesuaian berdasarkan umpan balik tersebut. Fowler & Turing (2019) menekankan pentingnya umpan balik dalam proses pengembangan audio untuk mencapai hasil yang diinginkan.

Pada tahap sebelumnya, kita sudah mengetahui cara untuk membuat suatu musik latar untuk gim Mr.Alien. sekarang bagaimana hasil dari musik latar tersebut masuk ke dalam gim. Berikut langkah pengujian aset audio yang bisa kamu lakukan:

a Menyiapkan File Musik Latar Gim

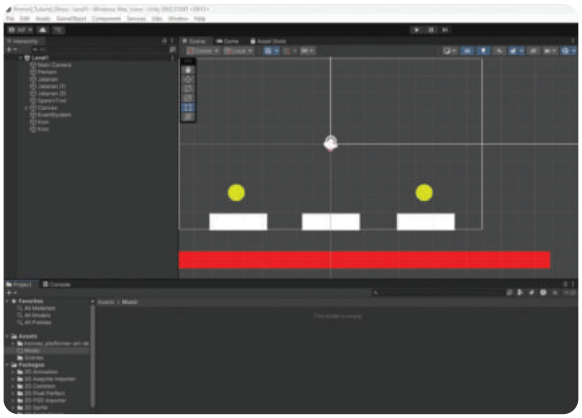
Berikut adalah satu jenis musik latar di gim Mr. Alien dengan dua format berbeda, kita ingin menguji apakah keduanya dapat masuk ke dalam *game engine* dari gim kita, pada kasus ini kita memakai Unity.



Gambar 5.52 Tampilan Ketika Membuka Hasil Audio pada Tahap Sebelumnya
Sumber: Muhammad Ardhan Fadhlurrahman/ Kemendikbudristek (2024)

b Buka Proyek di Dalam Game Engine

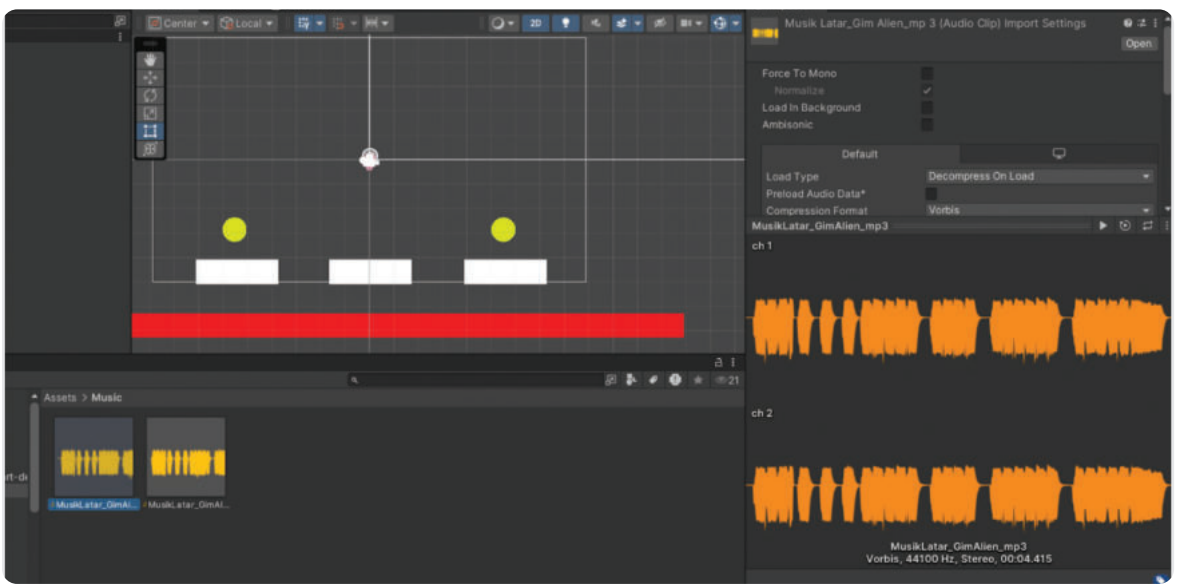
Berikut ini adalah proyek gim Mr. Alien yang akan kamu lengkapi dengan musik latar belakang. Sebelum memasukkan file audio, kamu perlu membuat folder khusus, misalnya *Assets/Music*, agar lebih mudah dalam pencarian dan pengelolaan file. Hal ini penting untuk kamu lakukan karena dalam pengembangan gim, kamu tidak bekerja sendiri, melainkan bersama tim. Dengan membuat struktur folder yang rapi dan terorganisasi, kamu akan membantu semua anggota tim menemukan file yang dibutuhkan dengan cepat dan efisien. Strukturisasi yang baik juga akan mempermudah proses *debugging*, pembaruan, maupun pengembangan lebih lanjut.



Gambar 5.53 Tampilan Proyek Gim Mr. Alien di Dalam *Unity*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

c Memasukan File Musik Latar Ke Game Engine

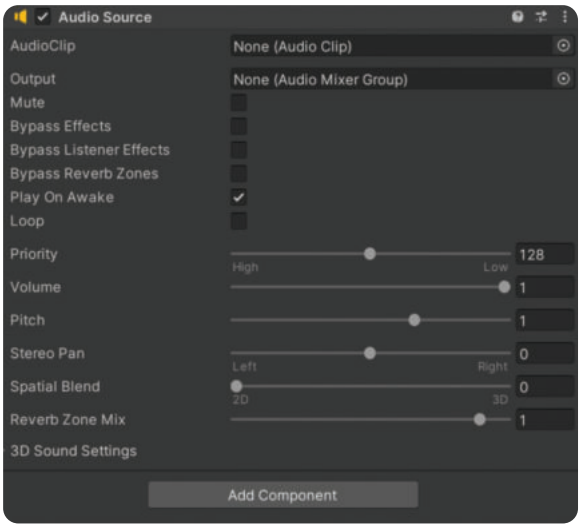
File musik latar gim sudah masuk dan bisa dilihat terdapat beberapa *setting* yang bisa dilakukan. Disini juga bisa melakukan testing suara dan memastikan apakah lebih bagus di format mp3 ataupun wav.



Gambar 5.54 File Musik Latar Gim Sudah Dimasukan Ke dalam Proyek Gim Mr.Alien
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

d Setting Audio Lebih Lanjut

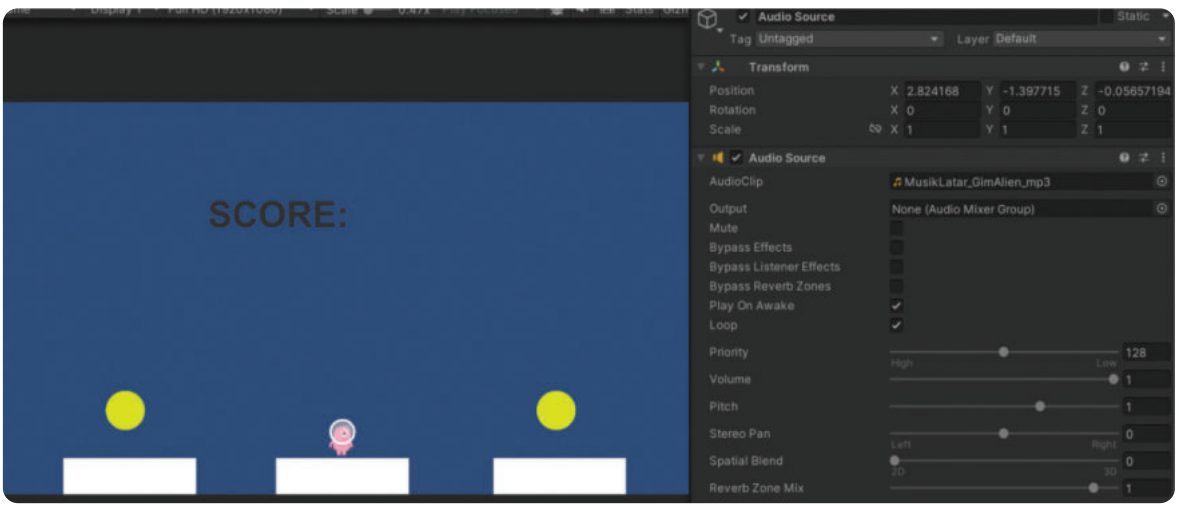
Kita akan memasukan file mp3 dan wav pada bagian *AudioClip.* Kita bisa melakukan *setting* lanjutan mengenai volume, *pitch* ataupun hal lain di dalam *AudioSource* ini.



Gambar 5.55 Komponen *Audiosource* untuk Memainkan Audio di dalam Gim.
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

e Uji Coba Musik Latar

Masukkan file musik latar ke dalam objek di gim Mr. Alien dengan memastikan bahwa file tersebut bisa diputar dan berjalan saat permainan dimulai. Untuk melakukannya, kamu bisa menggunakan komponen *Audio Source* yang ada di Unity. Pada bagian *Select Audio Clip*, pilih file musik yang sudah kamu simpan sebelumnya. Fungsi dari *Select Audio Clip* adalah untuk memasukkan file audio ke dalam variabel *AudioClip*, yang digunakan untuk memutar audio di dalam gim. Kamu bisa mengetes musik latar dengan memainkan gim dan memastikan *setting* volume dan hal lainnya sudah baik dan mendukung pengalaman bermain. Kita bisa mendengarkan *AudioClip* melalui AudioSource selama permainan dan ketika memainkan gimnya. Pada percobaan ini format wav mempunyai suara yang lebih baik dibandingkan mp3. maka dari itu kita akan menggunakan file wav untuk music latar.



Gambar 5.56 Tampilan *Game Engine* saat Melakukan *Testing*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

5. Optimasi Audio



Dalam mengembangkan gim, kamu perlu melakukan **optimasi audio** agar kualitas suara tetap baik tanpa mengorbankan performa. Salah satu hal yang perlu kamu perhatikan adalah pengaturan kualitas audio sesuai dengan platform. Untuk gim berbasis PC, kamu bisa menggunakan audio berkualitas tinggi agar pengalaman bermain lebih maksimal. Namun, jika gim ditujukan untuk perangkat seluler, sebaiknya kamu menggunakan versi audio yang lebih ringan atau sudah dikompresi, agar tidak membebani memori dan performa perangkat.

Selain itu, kamu juga harus **mengelola memori audio** dengan efisien. Audio yang tidak diatur dengan baik bisa membuat gim menjadi berat dan lambat. Oleh karena itu, penting untuk menggunakan teknik kompresi audio yang tepat dan menghindari memuat terlalu banyak aset audio secara bersamaan. Menurut Hahn (2016), manajemen memori yang baik sangat berperan dalam menjaga performa gim tetap optimal dan nyaman dimainkan.

6. Menyampaikan Emosi & Narasi



Dalam pengembangan gim, audio bukan hanya berfungsi sebagai pelengkap, tetapi juga berperan penting dalam **mendukung cerita** yang ingin kamu sampaikan. Musik dan efek suara dapat memperkuat elemen naratif, menambah kedalaman emosional, dan membantu pemain merasa lebih terhubung dengan alur cerita. Watt & Policarpo (2007) menjelaskan bahwa audio yang disusun dengan baik mampu memperkuat suasana dan membuat pengalaman bermain terasa lebih hidup dan menyentuh.

Salah satu contoh yang bisa kamu lihat adalah gim asal Indonesia berjudul *A Space for The Unbound*. Gim ini menggunakan musik latar yang sangat menyentuh hati sehingga pemain bisa merasakan keresahan dan perasaan mendalam dari karakter utama.

Selain untuk mendukung cerita, audio juga bisa kamu manfaatkan sebagai sarana umpan balik atau *feedback* kepada pemain. Misalnya, suara saat pemain mendapatkan item, menyelesaikan misi, atau ketika menerima kerusakan. Semua ini membantu pemain memahami apa yang sedang terjadi di dalam gim dan meningkatkan keterlibatan mereka. Kelley (2017) menyebutkan bahwa *feedback* audio yang tepat dapat meningkatkan pengalaman bermain secara keseluruhan.



Gambar 5.57 Gim *A Space for The Unbound*
Sumber: Mojiken Studio/ Toge Production (2022)

Dalam gim *Battle Royale*, misalnya, *feedback* audio sangat krusial. Suara langkah kaki, tembakan, atau ledakan bisa memberikan informasi penting tentang posisi musuh atau ancaman lainnya yang membuat pemain lebih waspada dan terlibat secara aktif dalam bertahan hidup.

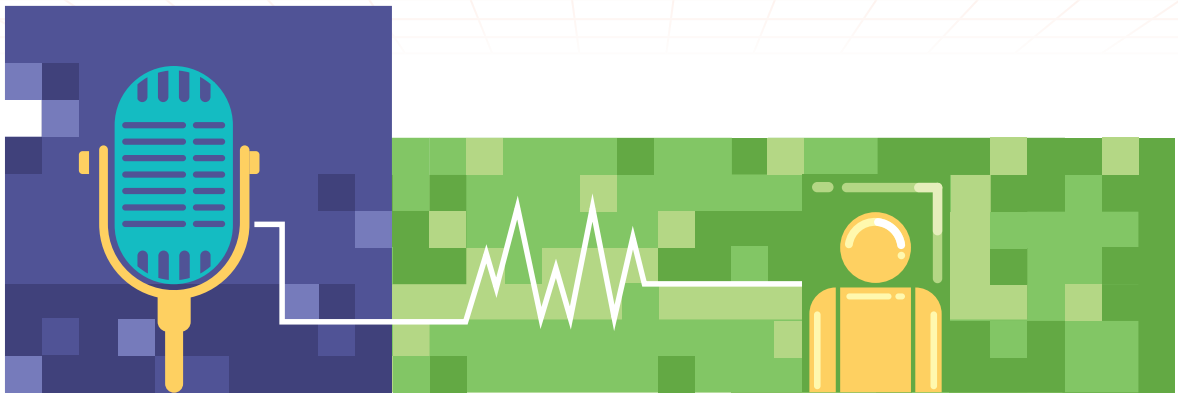


Gambar 5.58 Gim *Battle Royale*,
Sumber: PUBG Corporation/ KRAFTON , Inc. (2024)

Aktivitas 5.8 **Praktik Mandiri**

Sekarang saatnya kamu berkreasi! Buatlah aset audio originalmu sendiri dengan mengikuti langkah-langkah dari materi sebelumnya. Kamu bisa membuat efek suara, musik latar, atau dialog pendek untuk animasi atau gim. Gunakan kreativitasmu untuk menciptakan suasana dan pengalaman audio yang unik. Setelah selesai, dokumentasikan hasil kerjamu dengan tangkapan layar proses pembuatan dan tampilkan di depan kelas!

D. *Dubbing*



Perekaman suara atau *dubbing* adalah aspek penting dalam pengembangan gim yang berfungsi untuk memberikan suara pada karakter, narasi, dan dialog dalam permainan. Proses ini tidak hanya menambah kedalaman emosional, tetapi juga meningkatkan konteks cerita yang disampaikan kepada pemain. Dalam buku ini, kita akan membahas langkah-langkah penting dan faktor-faktor yang perlu diperhatikan ketika menerapkan perekaman suara dalam pengembangan gim.

1. Perencanaan *Dubbing*

Dalam proses *dubbing* untuk gim, hal pertama yang harus kamu siapkan adalah **skrip**. Skrip ini menjadi fondasi utama yang akan membimbing seluruh proses perekaman suara. Sebelum sesi dimulai, kamu perlu menyusun skrip yang jelas dan terperinci, mencakup dialog, instruksi teknis, serta catatan mengenai emosi atau nada suara yang diinginkan untuk setiap bagian. Hahn (2016) menyatakan bahwa skrip yang terstruktur dengan baik sangat penting untuk memberikan arahan yang jelas kepada pengisi suara, agar mereka dapat memahami konteks cerita dan menyampaikan nuansa dialog dengan tepat.

Setelah skrip siap, tahap penting berikutnya adalah ***casting***. Dalam tahap ini, kamu harus memilih pengisi suara yang sesuai dengan karakter yang akan diisi. Suara, gaya bicara, serta kemampuan akting menjadi pertimbangan utama untuk memastikan karakter terdengar meyakinkan dan hidup. Menurut Meyer et al. (2017), pemilihan pengisi suara yang tepat dapat meningkatkan keterikatan pemain terhadap karakter dalam gim, karena suara yang pas bisa memperkuat kepribadian dan emosi dari tokoh yang dimainkan.

2. Studio dan Peralatan

Untuk menghasilkan suara *dubbing* yang berkualitas tinggi, kamu perlu memperhatikan penggunaan studio rekaman yang memadai. **Studio** yang ideal harus memiliki akustik yang baik, bebas dari kebisingan luar, serta minim gema atau pantulan suara yang tidak diinginkan. Lingkungan perekaman yang bersih secara audio akan membantu pengisi suara menyampaikan dialog dengan jernih dan tanpa gangguan. Zhang et al. (2018) menekankan bahwa kualitas akustik dalam studio sangat memengaruhi hasil akhir rekaman, sehingga pemilihan studio tidak boleh dilakukan secara sembarangan.

Selain studio, kamu juga harus **berinvestasi pada peralatan perekaman** yang berkualitas. Mikrofon yang baik, *preamp*, dan perangkat lunak perekaman seperti **Pro Tools**, **Audacity**, atau **Adobe Audition** akan membantu kamu mendapatkan hasil audio yang bersih dan profesional. Reynolds (2015) menyebutkan bahwa pemilihan peralatan yang tepat tidak hanya meningkatkan kualitas perekaman, tetapi juga meminimalkan potensi masalah teknis yang bisa menghambat proses produksi.

3. Proses Perekaman

Sebelum kamu memulai sesi perekaman, langkah pertama yang harus dilakukan adalah memastikan semua **pengaturan suara telah dikonfigurasi dengan benar**. Periksa posisi mikrofon, sesuaikan level input agar tidak terjadi *clipping* atau distorsi, dan pastikan perangkat perekaman berjalan dengan lancar. Bennett (2015) mengingatkan bahwa pengaturan awal yang tepat akan sangat membantu menghindari masalah teknis saat proses penyuntingan, sehingga proses produksi bisa berjalan lebih efisien.

Setelah pengaturan teknis selesai, fokus berikutnya adalah **memberikan pengarahan kepada pengisi suara**. Kamu perlu menjelaskan konteks adegan, emosi yang harus disampaikan, serta intonasi yang sesuai dengan karakter yang mereka mainkan. Dengan pengarahan yang jelas, pengisi suara akan lebih mudah memahami karakter dan situasi yang sedang berlangsung. Fowler & Turing (2019) menekankan pentingnya pengarahan yang baik untuk memastikan hasil akhir yang sesuai dengan visi cerita.

Selama sesi berlangsung, kamu disarankan untuk **merekam beberapa pengambilan untuk setiap bagian dialog**. Hal ini akan memberimu lebih banyak opsi saat proses editing dan memungkinkan kamu memilih hasil terbaik dari tiap performa yang diberikan. McGuire et al. (2012) menyarankan agar tidak hanya mengandalkan satu pengambilan, karena variasi performa bisa memberikan hasil akhir yang lebih kuat dan meyakinkan.

4. Editing dan Pengolahan

Setelah proses perekaman selesai, langkah berikutnya yang harus kamu lakukan adalah **menyunting rekaman audio**. Pada tahap ini, kamu perlu menghapus kesalahan, jeda yang tidak perlu, atau suara latar yang mengganggu. Pastikan setiap transisi antara satu pengambilan ke pengambilan lain terdengar halus dan alami. Müller et al. (2018) menyatakan bahwa penyuntingan yang dilakukan dengan baik dapat secara signifikan meningkatkan kualitas keseluruhan dari audio yang digunakan dalam gim.

Setelah penyuntingan selesai, kamu bisa melanjutkan ke tahap **pengolahan suara**. Di tahap ini, kamu dapat menerapkan berbagai efek dan teknik pemrosesan, seperti kompresi untuk menyeimbangkan volume, *equalization* untuk memperjelas frekuensi tertentu, serta *reverb* untuk menciptakan nuansa ruang atau atmosfer yang sesuai dengan adegan. Kelley (2017) menjelaskan bahwa pengolahan suara yang tepat tidak hanya memperbaiki kualitas audio, tetapi juga memberikan dimensi tambahan yang membuat rekaman terasa lebih hidup dan mendalam.

5. Implementasi dalam Gim

Setelah proses editing selesai, langkah selanjutnya adalah **mengekspor file audio dalam format yang sesuai** dengan *game engine* yang kamu gunakan, seperti **WAV** atau **OGG**. Format **WAV** sering dipilih karena kualitasnya tinggi, sementara **OGG** lebih ringan dan cocok untuk penggunaan di perangkat seluler. Dalam proses ini, kamu perlu mempertimbangkan kualitas suara sekaligus ukuran file agar tidak membebani performa gim. Hahn (2016) menekankan bahwa memilih format file yang tepat sangat penting untuk menjaga keseimbangan antara kualitas audio dan efisiensi performa.

Setelah file diekspor, kamu bisa mulai **mengimplementasikan audio tersebut ke dalam *game engine*** seperti Unity atau Unreal Engine. Pastikan setiap file suara diatur dengan pemicu (*trigger*) yang tepat, sehingga suara dapat diputar secara sinkron dengan aksi atau peristiwa dalam *gameplay*—misalnya saat karakter berbicara, menerima misi, atau menyelesaikan tugas. Erleben et al. (2009) menyatakan bahwa pengelolaan audio yang baik tidak hanya membuat permainan terdengar lebih hidup, tetapi juga secara langsung meningkatkan kualitas pengalaman bermain bagi pemain.

6. Pengujian Audio

Setelah audio diimplementasikan ke dalam gim, kamu perlu melakukan **pengujian menyeluruh** untuk memastikan semuanya berjalan dengan baik dalam konteks *gameplay*. Uji setiap rekaman suara secara langsung di dalam permainan untuk memastikan bahwa audio tidak hanya terdengar jelas, tetapi juga mendukung pengalaman bermain secara keseluruhan. Lakukan pengujian dalam berbagai situasi—seperti saat dialog berlangsung, saat karakter bergerak, atau saat terjadi aksi penting—agar kamu bisa melihat apakah ada suara yang terlalu pelan, terlalu mengganggu, atau bahkan tidak muncul. Watt & Policarpo (2007) menekankan bahwa pengujian audio yang baik bisa membantu mengidentifikasi masalah-masalah kecil yang mungkin terlewatkan selama proses pengembangan.

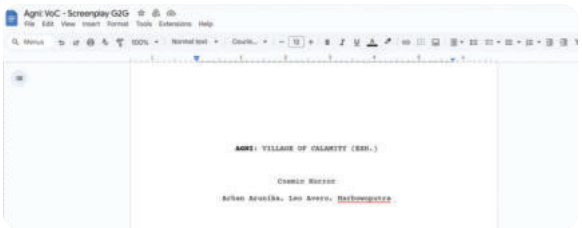
Selain pengujian teknis, kamu juga perlu **mengumpulkan umpan balik dari tim pengembang dan penguji**. Mintalah pendapat mereka tentang apakah suara sudah sesuai dengan karakter dan konteks cerita. Terkadang, suara yang terdengar bagus secara teknis bisa terasa kurang cocok secara emosional atau naratif. Berdasarkan umpan balik tersebut, kamu bisa melakukan penyesuaian ulang untuk mencapai kualitas audio yang optimal. Zhang et al. (2018) menyatakan bahwa umpan balik dari berbagai pihak sangat berharga dalam proses penyempurnaan kualitas *dubbing* dan pengalaman suara dalam gim.

Berikut ini kamu akan mempelajari contoh proses *dubbing* yang bisa dilakukan dalam pengembangan sebuah gim. Dalam contoh ini, proses *dubbing* dilakukan menggunakan aplikasi **Audacity** untuk perekaman dan penyuntingan suara, serta **Unreal Engine** sebagai platform implementasi ke dalam gim.

Dalam pengembangan gim *Agni: Village of Calamity*, terdapat kebutuhan untuk menambahkan suara dialog ke dalam beberapa *scene* penting. Untuk itu, berikut adalah langkah-langkah yang bisa kamu ikuti dalam proses *dubbing*:

1. Mempersiapkan Skrip

Langkah pertama yang perlu kamu lakukan adalah mempersiapkan skrip dialog. Skrip ini berisi percakapan antar karakter yang akan direkam. Selain itu, skrip juga dilengkapi dengan informasi penting seperti ekspresi, intonasi, dan petunjuk emosional, agar pengisi suara dapat lebih mudah menjiwai peran dan menyampaikan dialog sesuai konteks cerita.



Gambar 5.59 Dokumen Digital Skrip *AGNI Village of Calamity*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

2. Casting pengisi suara dan rekaman

Langkah berikutnya adalah memilih pengisi suara yang sesuai dan melakukan sesi rekaman. Pada tahap ini, pengisi suara biasanya tidak harus menghafal dialog, melainkan diberikan skrip untuk dibaca. Cara ini dilakukan agar mereka lebih fokus pada intonasi, penjiwaan, dan emosi dalam menyampaikan dialog sesuai dengan karakter yang diperankan.



Gambar 5.60 Pengisi Suara Sedang Melakukan *Take*
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

3. Editing dan Penyuntingan

Setelah proses rekaman selesai, kamu perlu melakukan penyuntingan untuk memastikan hasil suara sesuai dengan animasi di dalam gim. Proses ini mencakup penyesuaian agar suara selaras dengan gerakan bibir dan tubuh karakter dalam *cutscene*. Sinkronisasi yang baik akan meningkatkan kualitas naratif dan membuat pengalaman bermain terasa lebih imersif.



Gambar 5.61 Proses Editing dan Penyuntingan Oleh Pengembang Gim
Sumber: Muhammad Ardhan Fadhlurrahman/Kemendikbudristek (2024)

4. Review, Revisi dan Final Delivery

Setelah proses penyuntingan selesai, kamu perlu melakukan review bersama tim untuk memastikan apakah kualitas *output* sesuai dengan yang dibutuhkan, jika tidak maka akan diperlukan proses revisi yaitu menjalankan proses kembali dari tahap pertama sampai dengan tahap ke empat. Ketika kualitas *output* sesuai dengan yang dibutuhkan, maka proses persiapan terakhir ada proses Final Delivery yaitu dengan memastikan nama file sesuai dengan yang dibutuhkan dan dimasukan ke dalam permainan.

Aktivitas 5.9 Praktik Tim Kerja

Dalam aktivitas ini, kamu akan bekerja dalam tim yang sama seperti pada aktivitas sebelumnya. Tugas kamu adalah melakukan proses *dubbing* bersama tim. Silakan tentukan siapa saja yang akan mengerjakan setiap tahapan sesuai dengan materi *dubbing* yang telah dibahas. Kemudian, lengkapi lembar kerja berikut ini:

1. Persiapan Skrip: Transkripsi dan adaptasi skrip untuk sinkron dengan gerakan bibir. Pastikan dialog sudah disesuaikan dengan konteks budaya.
2. *Casting* Pengisi Suara: Pilih aktor suara yang sesuai dengan karakter. Pertimbangkan intonasi dan aksen yang tepat.
3. Rekaman: Lakukan rekaman dialog. Pastikan semua aktor mengikuti arahan sutradara untuk mendapatkan intonasi dan emosi yang sesuai.
4. Edit dan Sunting: Edit hasil rekaman untuk menghilangkan kesalahan dan melakukan sinkronisasi dengan adegan visual.
5. *Mixing* dan *Mastering*: Gabungkan suara dialog dengan efek suara dan musik latar untuk menciptakan trek audio yang lengkap.
6. Reviu dan Revisi: Tonton dan dengarkan hasil akhir. Lakukan revisi jika diperlukan untuk memastikan kualitas.
7. *Final Delivery*: Kirimkan audio final dalam format yang diperlukan untuk distribusi atau penyiaran. Dalam hal ini kalian akan mengirimkan ke guru kalian masing-masing
8. Catat setiap langkah yang telah dilakukan, termasuk kendala yang dihadapi dan solusi yang diterapkan. Buat laporan singkat mengenai proses dubbing yang telah dilalui oleh tim. Setelah semua tahapan selesai dan laporan dilengkapi, kumpulkan hasil pekerjaan kalian sesuai dengan instruksi yang diberikan.

Aktivitas 5.10 Proyek Tim

Dalam bab ini kamu sudah mempelajari tentang aset audio visual dan penerapannya. Sekarang giliran kamu dan tim untuk mencoba menerapkan materi yang sudah dipelajari. Silahkan pelajari skenario berikut dan ikuti setiap tahapannya:

Skenario

Kalian sedang melaksanakan Praktek Kerja Lapangan atau PKL di sebuah industri gim. Kemudian kalian diberi tugas oleh pembimbing PKL untuk membuat sebuah aset *Audio visual* dari sebuah desain gim. Kalian diperbolehkan untuk membuat desain gim sendiri. Kemudian ikuti langkah kerja berikut ini :

1. Tuliskan ide desain gim yang akan tim kalian buat.
2. Buat daftar aset audio dan visual yang dibutuhkan untuk mendukung desain gim tersebut.
3. Lakukan simulasi gerak pada aset yang sudah kalian buat.
4. Rekam proses *dubbing* untuk beberapa dialog dalam gim kalian.
5. Setelah semua tugas selesai, presentasikan hasil kerja tim kalian kepada pembimbing PKL.

Kalian bisa melakukan ini di depan kelas bersama teman kelas yang lain. Semangat ya!



Uji Kompetensi

Untuk menguji dan mengasah keterampilan kamu lebih lanjut, sekarang saatnya menyusun sebuah portofolio. Berikut adalah langkah-langkah yang perlu kamu lakukan:

1. Buatlah aset visual 2D dan aset visual 3D dalam gim yang kamu kembangkan.
2. Buatlah aset audio yang dibutuhkan dalam desain yang kamu buat pada intruksi nomor 1.
3. Setelah itu lakukanlah proses *dubbing* untuk desain gim yang kamu kembangkan.
4. Selanjutnya, buatlah dokumen portofolio kamu dengan list berikut:

a. Aset visual 2D

b. Aset Visual 3D

c. Proses pembuatan Asset Audio yang dibutuhkan

d. Tahapan dubing yang kalian lakukan

5. Desainlah portofolio kamu sebagus mungkin, sehingga bisa kamu gunakan saat lulus nanti.



Pengayaan

Bagi kamu yang sudah memahami tentang materi dalam bab ini, kamu bisa melakukan eksplorasi lebih mendalam tentang pembuatan dan penerapan aset audio visual menggunakan aplikasi lain yang tidak dibahas dalam bab ini! dan kamu juga bisa mempelajari dan mempraktekkan materi yang ada dalam buku berikut ini untuk menambah wawasan dan kemampuan kamu. Kamu bisa membaca buku berjudul Programming 2D Games dari Charles Kelly (2012) .



Refleksi

Setelah kalian mempelajari bab ini, sekarang coba kalian melakukan refleksi diri dengan menjawab pertanyaan berikut ini :

1. Apa yang kamu rasakan setelah memahami materi Aset Audio Visual?
2. Apakah kamu semakin tertarik untuk mengembangkan gim terbaikmu?
3. Apakah kamu semakin yakin untuk terjun dalam dunia pengembangan gim ini?
4. Jika yakin, bagian mana dalam materi bab ini yang benar-benar kamu kuasai dan bagian mana yang perlu kamu pelajari lebih lanjut lagi?
5. Menurut kamu *soft skill* apa yang harus kamu asah dan kembangkan untuk bisa bekerja di dunia pengembangan gim ini?
6. Dan bagaimana caranya agar kamu bisa meningkatkan *soft skill* tersebut?

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2024
Pengembangan Gim
untuk SMK/MAK Kelas XI-XII
Penulis: Rini Melati dan Muhammad Ardhan Fadlurrahman
ISBN: 978-634-00-0360-4

BAB 6

Pengujian Gim



Tujuan Pembelajaran

Setelah mempelajari materi dalam bab ini, diharapkan kamu akan mampu menerapkan pengujian perangkat lunak meliputi *debugging*, menerapkan optimasi kinerja gim, dan pembaharuan.

Kata Kunci

Debug Optimasi Bugs

Crash *Breakpoint*

Soft Skill

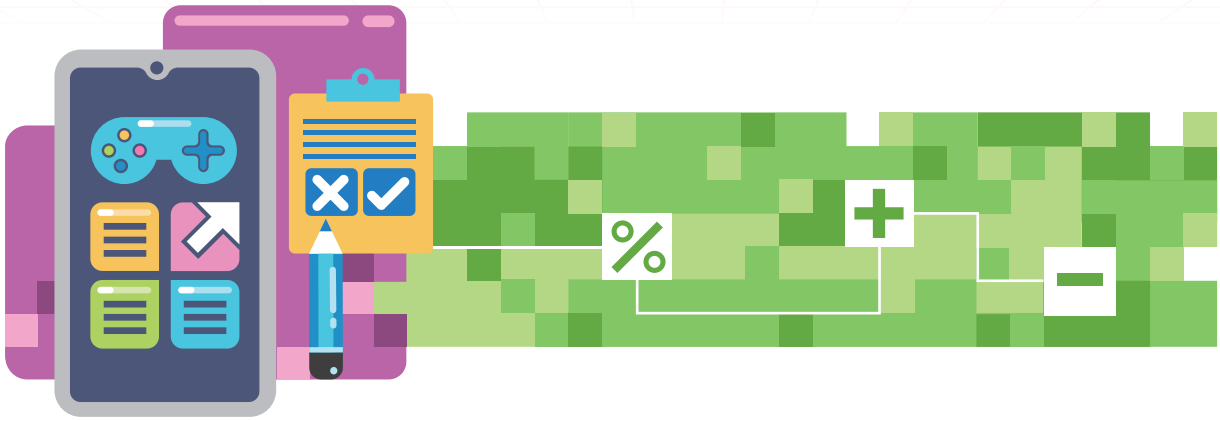
- **Kooperatif dalam bekerjasama** antar tim kerja untuk mengembangkan gim
- **Kooperatif dalam bekerja** Ketika melakukan *debugging* dan optimasi gim
- **Kolaboratif** dalam menyelesaikan tugas tim terkait pengujian gim
- **Kreatif** dalam menuangkan ide dan alternatih untuk melakukan pengujian gim
- **Kritis** dalam memilih engine yang akan digunakan
- **Teliti** dalam menyelesaikan tugas terkait dengan identifikasi bug pemrograman
- **Tanggung jawab dan disiplin** dalam menyelesaikan tugas mandiri

Peta Materi



Saat kamu berhasil membuat suatu produk atau aplikasi, bagaimana caranya agar kamu tahu bahwa produk atau aplikasi yang kamu buat sudah sesuai dengan kebutuhan yang diinginkan atau tujuan yang akan dicapai? Untuk mengetahuinya, kamu bisa melakukan pengujian terhadap produk atau aplikasi yang kamu buat tersebut. Tapi, sebelum kamu melaukan pengujian. Kamu perlu tahu apa saja yang harus disiapkan saat melakukan pengujian? Bagaimana melakukannya dan seperti apa hasil dari pengujian tersebut? Kamu bisa mengetahui dan mempelajarinya dalam bab ini ya.

A. *Debugging* Gim



Aktivitas 6.1 **Literasi Mandiri**

Sebelum kamu memulai pembelajaran pada bab ini, coba kamu cari tahu melalui internet atau buku di perpustakaan tentang alasan mengapa perlu dilakukan pengujian terhadap sebuah gim dan bagaimana cara melakukan pengujiannya. Setelah itu, kamu lengkapi tabel berikut ini, ya.

Alasan	
Cara Pengujian Gim	

Setelah itu, silahkan kamu presentasikan hasil yang kamu dapatkan di depan kelas untuk bisa saling berbagi informasi dengan teman yang lainnya.

Aktivitas yang kamu lakukan di atas bertujuan untuk mengetahui alasan mengapa sebuah gim perlu dilakukan pengujian. Pengujian gim atau disebut juga dengan *game testing* adalah proses yang dilakukan untuk menemukan kesalahan atau penyimpangan dari tujuan awal gim, serta memastikan agar gim telah mencapai hasil yang diinginkan.

Pengujian dan iterasi adalah proses penting dalam pengembangan gim yang berkontribusi pada peningkatan kualitas produk akhir. Menerapkan pengujian perangkat lunak dan *debugging* merupakan langkah krusial untuk memastikan bahwa gim berfungsi dengan baik dan memberikan pengalaman yang memuaskan bagi pemain. Dalam bab ini, kita akan membahas berbagai aspek terkait pengujian gim.

1. Tujuan Pengujian Perangkat Lunak

a Menjamin Kualitas

Salah satu alasan utama mengapa kamu perlu melakukan pengujian adalah untuk menjamin kualitas gim yang kamu buat. Dengan melakukan pengujian, kamu bisa memastikan bahwa gim berjalan sesuai harapan, tidak ada *bug* yang mengganggu, dan tetap stabil saat dimainkan. Kualitas yang baik sangat penting agar pemain merasa puas dan ingin terus memainkan gimmu (Nash, 2016).



b Meningkatkan Pengalaman Pemain

Selain itu, pengujian juga bisa meningkatkan pengalaman pemain. Saat kamu menemukan dan memperbaiki masalah sebelum gim dirilis, kamu membantu pemain mendapatkan pengalaman bermain yang lebih nyaman dan lancar. Pengujian yang dilakukan dengan baik bisa mencegah gangguan saat gameplay berlangsung (Bach, 2015)

2. Jenis Pengujian

Dalam mengembangkan sebuah gim, kamu perlu melakukan beberapa jenis pengujian agar hasilnya benar-benar siap dimainkan. Setiap jenis pengujian memiliki fokus yang berbeda dan penting untuk dilakukan secara menyeluruh.

a Pengujian Fungsional

Pengujian ini dilakukan untuk memeriksa apakah semua fitur dalam gim seperti kontrol karakter, misi, dan antarmuka pengguna berfungsi dengan baik. Cohen (2018) menekankan pentingnya pengujian fungsional untuk memastikan fitur dasar berfungsi dengan baik. Contohnya, kalau kamu membuat gim platformer, fitur utama dari gim tersebut adalah fitur melompat yang akan dipakai secara berulang sepanjang permainan. Maka, kamu perlu memastikan saat fitur melompat dapat berjalan dengan lancar saat pengujuan fungsional.

b Pengujian Non–Fungsional

Jenis pengujian ini berfokus pada aspek yang tidak langsung terlihat dari fitur utama, seperti kecepatan (*frame rate*), waktu loading, kenyamanan saat digunakan, atau keamanan sistem. Perry (2019) menyebut bahwa pengujian ini penting untuk menjaga pengalaman pengguna tetap baik. Misalnya, jika gim kamu akan dirilis di platform seperti Xbox atau PlayStation, maka kamu perlu memastikan gim berjalan dengan *frame rate* minimal 60fps atau 120fps, sesuai standar. Untuk itu, kamu melakukan pengujian non-fungsional untuk mengecek kinerjanya.

c Pengujian Beta

Pengujian beta dilakukan dengan cara mengundang orang lain, biasanya bukan dari tim pengembang untuk mencoba gim sebelum resmi dirilis. Dari sini, kamu bisa mendapatkan masukan langsung dari pemain. O'Neil (2021) menyebut bahwa pengujian beta sangat berguna karena memberi sudut pandang pengguna akhir yang mungkin belum terpikirkan oleh pengembang. Contohnya, jika kamu ingin tahu apakah gim buatanmu sudah cocok untuk target pemain tertentu, kamu bisa mengadakan pengujian beta. Masukan dari pemain akan membantumu melakukan perbaikan sebelum gim dirilis ke pasar.

3. Proses Pengujian

a Rencana Pengujian

Langkah pertama yang perlu kamu lakukan adalah membuat dokumen rencana pengujian. Di dalamnya berisi jenis pengujian apa yang akan dilakukan, alat apa saja yang akan digunakan, dan skenario pengujian yang sesuai dengan fitur gim. Sommerville (2016) menjelaskan bahwa rencana pengujian yang jelas dan lengkap sangat penting untuk memandu jalannya proses pengujian dari awal hingga akhir.



b Pengembangan *Test Cases*

Setelah rencana siap, kamu perlu menulis ***test case***, yaitu petunjuk langkah demi langkah untuk menguji suatu fitur. *Test case* ini menjelaskan apa yang harus kamu lakukan dan hasil seperti apa yang diharapkan. Kaner et al. (2016) menyebutkan bahwa *test case* yang terperinci akan membuat proses pengujian jadi lebih efisien dan terarah.

c Pelaksanaan Pengujian

Jika semua *test case* sudah siap, kamu bisa mulai melakukan pengujian sesuai dengan rencana. Catat semua hasil yang kamu temukan, termasuk *bug* atau masalah yang muncul. Gunakan alat pelaporan bug agar semua masalah bisa terdokumentasi dengan baik. Jorgensen (2018) menekankan bahwa dokumentasi yang baik sangat penting dalam proses pengujian perangkat lunak, karena akan memudahkan saat memperbaiki dan mengembangkan gim lebih lanjut.

Berikut ini adalah proses pengujian yang dilakukan dalam gim Mr. Alien:

1. **Menentukan Objektif dari Pengujian**

   Dalam proses ini, kita menentukan objektif dari pengujian yang akan dilakukan. Apakah kontrol dirasa cukup baik dan apakah level terlalu sulit bagi pemain. Kedua objektif tersebut sangat penting untuk diketahui agar dapat menjadi referensi dalam pengembangan lanjutan dari pengujian tersebut. Selain itu, kamu juga harus menentukan berapa banyak orang (sampel) yang akan ikut dalam pengujian ini.

2. **Menentukan Metode Pengujian**

   Metode yang dipakai dalam pengujian ini adalah kuesioner dan observasi bermain. Diawali dengan pemain memainkan level 1 dari gim Mr.Alien tersebut. Pengembang mencoba merekam dan memperhatikan bagaimana pemain tersebut memainkan gimnya, dari raut ekspresi maupun dari dari bagaimana cara mereka memainkan gim yang diberikan. Setelah memainkan gim tersebut mereka akan mengisi kuesioner yang berisi beberapa pertanyaan terkait dari objektif pengujian.

3. **Melakukan Pengujian**

   Setelah jumlah sampel dan motode ditentukan, kita langsung melakukan pengujian.

4. **Merekap dari Hasil Pengujian**

   Ketika menggunakan kuesioner online, biasanya data sudah dirangum secara otomatis. Kita langsung dapat melihat berapa persen dari partisipan yang memilih A, B, C, ataupun D. Data tersebut akan menjadi bahan dalam membuat rangkuman.

5. **Membuat Rangkuman dari Hasil Rekap**

   Setelah rekap tersedia, tugas selanjutnya adalah merangkai sebuah narasi dari hasil masing masing rekap pertanyaan yang diberikan. Rangkuman ini akan menjelaskan kondisi spesifik gim yang kamu uji. Misalnya, kita memiliki 10 pertanyaan dari 30 partisipan, dalam rekapan sudah terdapat 10 rekapan untuk masing masing pertanyaan.

6. **Membuat Rekomendasi dan Laporan dari Hasil Rangkuman**

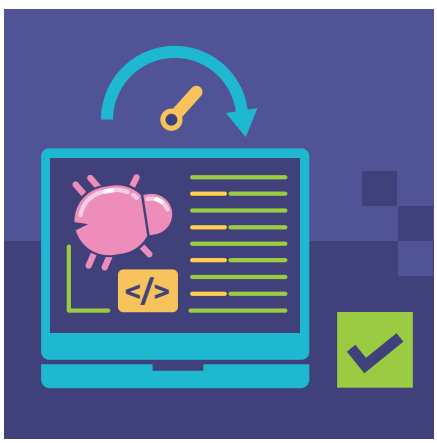
   Dari rangkuman yang kita buat, bandingkan dengan tujuan awal pengujian. Apakah hasilnya sesuai? Dari situ, kita bisa menyusun rekomendasi yang berisi apa saja yang sebaiknya diperbaiki. Laporan ini nantinya akan menjadi dasar untuk pengembangan gim selanjutnya.

7. **Memperbaiki Gim Berdasarkan Hasil Pengujian**

   Setelah laporan selesai, kita bisa mempresentasikan hasilnya kepada tim pengembang. Bersama-sama, tim bisa berdiskusi tentang apa yang harus ditambahkan, dikurangi, atau diperbaiki berdasarkan masukan dari pengujian. Dengan begitu, gim buatanmu akan semakin baik dan siap untuk dimainkan oleh lebih banyak orang.

4. *Debugging* dalam Pengembangan Gim

Debugging adalah proses menemukan, mendiagnosis, dan memperbaiki *bug* dalam gim. Bug dapat berkaitan dengan masalah kode, logika, atau performa. McConnell (2004) menyatakan bahwa *debugging* adalah keterampilan krusial dalam pengembangan perangkat lunak. Berikut tahapannya:



a Reproduksi Bug

Cobalah untuk mereproduksi *bug* dalam kondisi yang sama seperti yang dilaporkan, sehingga kamu dapat memahami penyebabnya. Shaw (2019) menekankan pentingnya reproduksi *bug* untuk mendiagnosis masalah dengan efektif.

b Analisis Kode

Telusuri kode yang relevan, fokus pada area yang mungkin menyebabkan masalah. Gunakan alat *debugging* untuk memeriksa variabel dan alur eksekusi. Fowler (2018) menyoroti bahwa analisis kode yang teliti dapat membantu menemukan kesalahan yang sulit dideteksi.

c Menambahkan *Logging*

Sisipkan pernyataan *logging* untuk melacak nilai variabel dan status program, membantu Anda mengidentifikasi titik di mana kesalahan terjadi. Wang (2020) menunjukkan bahwa *logging* yang efektif dapat mempercepat proses *debugging*.

d Menggunakan Alat *Debugging*

Manfaatkan *debugger* dalam lingkungan pengembangan seperti Unity, Unreal Engine, atau IDE lainnya. Alat ini memungkinkan kamu untuk menelusuri kode secara langsung, memeriksa kondisi *runtime*, dan mengubah nilai variabel. Johnson (2021) menggarisbawahi pentingnya penggunaan alat *debugging* untuk efektivitas yang lebih besar.

e Perbaikan *Bug*

Setelah menemukan sumber masalah, lakukan perbaikan. Pastikan kamu menguji perbaikannya untuk memastikan *bug* benar-benar teratasi. Perry (2019) mencatat bahwa pengujian pasca-perbaikan adalah langkah penting dalam proses *debugging*.

f Pengujian Ulang

Setelah perbaikan, lakukan pengujian ulang untuk memastikan bahwa *bug* telah diatasi dan tidak ada efek samping baru yang muncul. Cohen (2018) mengingatkan bahwa pengujian ulang membantu menjaga kualitas perangkat lunak.

Berikut ini adalah contoh *debugging* yang dilakukan dalam gim Mr. Alien:

Misalkan terdapat logika pemrograman di dalam gim Mr.Alien seperti ini untuk mencetak *score* di dalam script FungsiPemain menggunakan sebuah komponen UIText bernama UIScore dan juga variable int bernama score, seperti berikut:

Ketika dimainkan dan pemain mencetak score lebih besar dari 3, tidak terdapat perubahan dari komponen UIScore, dan text tetap stagnan di “Score = 3”. hal tersebut adalah sebuah bug. Lalu kita bisa perbaiki menjadi seperti ini:

```
public class FungsiPemain : MonoBehaviours
{
       private Text UIScore;
       private int score;

      void Start()
      {
             UIScore = GetComponents<Text>();
      }

     void Update()
      {
           if(score == 0)
           {
                UIScore.text = “Score = 0”;
           }
          else if(score == 1)
          {
               UIScore.text = “Score = 1”;
          }
           if(score == 2)
           {
                UIScore.text = “Score = 2”;
           }
          else if(score == 3)
          {
               UIScore.text = “Score = 3”;
          }
     }
}
```

```
public class FungsiPemain : MonoBehaviours
{
       private Text UIScore;
       private int score;

      void Start()
      {
             UIScore = GetComponents<Text>();
      }

     void Update()
      {
          UIScore.text = “Score : “ + score;
     }
}
```

dengan perubahan ini kita bisa memastikan semua jenis score akan bisa di cetak.

Aktivitas 6.2 **Praktik Mandiri**

Dalam aktivitas ini kamu diminta untuk mempraktikkan *debug* dengan kode program untuk gim yang kamu buat, kamu boleh menggunakan *engine* yang sudah kamu pahami sebelumnya. Selanjutnya presentasikan hasil kerja kamu di depan teman-temanmu.

5. Pengujian Otomatis Dalam Gim



Pengujian otomatis adalah proses penting yang bisa kamu manfaatkan untuk memastikan gim berjalan dengan baik, tanpa harus menguji semuanya secara manual. Dengan bantuan alat dan sistem otomatis, kamu bisa menghemat waktu dan menjaga konsistensi hasil pengujian.

a. *Automated Testing*

Kamu bisa menggunakan *framework* untuk mengotomatiskan pengujian. Misalnya, *unit testing* untuk logika permainan dan *UI testing* untuk antarmuka pengguna. Pengujian otomatis dapat menghemat waktu dan memastikan pengujian dilakukan secara konsisten. *Meyer (2017)* menekankan manfaat pengujian otomatis dalam meningkatkan efisiensi proses pengujian. Sebagai contoh, bayangkan kamu membuat *mobile* gim yang memiliki 1000 level. Jika ada satu komponen inti yang berubah, tentu akan merepotkan jika kamu harus menguji setiap level satu per satu. Nah, dengan pengujian otomatis, kamu bisa membuat program yang mensimulasikan bagaimana pemain berinteraksi di dalam gim. Program ini dapat menjalankan pengujian terhadap semua level secara otomatis, bahkan hingga 10.000 level, dengan waktu yang jauh lebih singkat.

b. *Continuous Integration/Continuous Deployment* (CI/CD)

Kamu juga bisa mengintegrasikan pengujian otomatis ke dalam sistem *Continuous Integration/ Continuous Deployment* atau CI/CD. Sistem ini memungkinkan pengujian dijalankan secara otomatis setiap kali ada perubahan kode dalam proyek. Dengan begitu, kamu bisa mendeteksi bug lebih awal dan memperbaikinya sebelum masalah menjadi lebih besar. Duvall et al. (2015) menjelaskan bahwa CI/CD membantu pengembang merespons masalah dengan lebih cepat dan efisien.

Contohnya, jika biasanya kamu harus menjalankan *automated testing* secara manual, maka dengan CI/CD proses ini akan berjalan otomatis setiap kali kamu menyimpan atau mengunggah perubahan kode. Hal ini sangat berguna, karena jika pengujian dilakukan secara otomatis setiap terjadi perubahan, kamu bisa langsung tahu bagian kode mana yang menyebabkan bug. Dengan begitu, proses debugging akan menjadi lebih cepat dan lebih terarah.

6. Dokumentasi dan Pelaporan



Agar proses pengujian yang kamu lakukan bisa memberikan manfaat jangka panjang, kamu perlu mencatat dan melaporkan semua hasilnya dengan baik. Dokumentasi dan laporan yang rapi akan sangat membantu dalam proses pengembangan berikutnya.

a Dokumentasikan Temuan

Saat kamu menemukan bug atau masalah saat pengujian, catat dengan lengkap. Tuliskan langkah-langkah untuk mengulang masalah tersebut, status perbaikannya, dan hasil dari pengujian ulang setelah diperbaiki. Kaner et al. (2016) menyatakan bahwa dokumentasi yang baik dapat membantu menjaga transparansi dalam proses pengembangan dan menjadi referensi penting untuk masa mendatang.

Contohnya, saat kamu menguji sebuah gim dan menemukan bahwa 80% dari 100 pemain menganggap gim tersebut terlalu sulit, kamu perlu mencatat lebih dari sekadar hasil akhirnya. Catat juga langkah-langkah yang dilakukan oleh pemain hingga mereka merasa kesulitan. Dokumentasi ini kemudian bisa kamu berikan kepada tim pengembang untuk dianalisis. Dari sana, tim bisa memutuskan bagian mana dari gim yang perlu disesuaikan atau diperbaiki.

b Laporan Pengujian

Setelah dokumentasi selesai, kamu perlu menyusunnya dalam bentuk laporan pengujian. Laporan ini berisi ringkasan hasil pengujian, daftar bug yang ditemukan, serta rekomendasi untuk perbaikan. Jorgensen (2018) menjelaskan bahwa laporan yang baik akan membantu tim pengembang memahami kondisi gim secara menyeluruh dan fokus pada area yang perlu ditingkatkan.

Berikut ini adalah bagian-bagian yang bisa kamu sertakan dalam laporan pengujian:

Latar Belakang Pengujian

Latar belakang menjelaskan tujuan dari diadakannya pengujian ini secara singkat dan ringkas, agar pembaca laporan memiliki pengertian lebih dalam ketika membaca hasil pengujian.

Metode Pengujian

Metode pengujian menjelaskan tentang bagaimana proses dalam melakukan pengujian, apakah dengan sampel atau *questionnaire* ataupun cara lain bisa kamu deskripsikan disini.

Ringkasan Hasil Pengujian

Sajikan data utama dari pengujian. Misalnya, persentase pemain yang merasa kesulitan, fitur yang sering bermasalah, atau grafik dari hasil pilihan responden.

Rekomendasi dari Ringkasan Hasil Pengujian

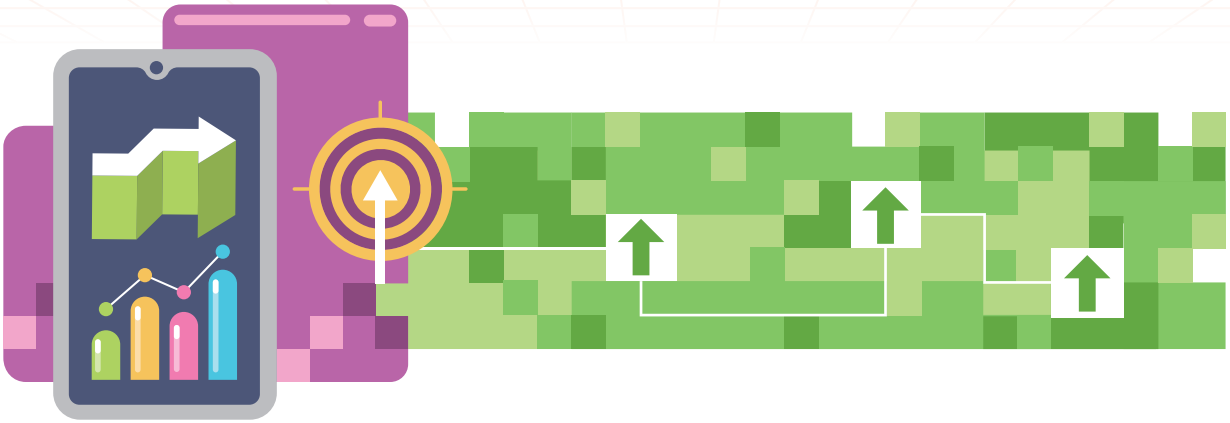
Berdasarkan data tersebut, berikan saran atau langkah konkret. Misalnya, menurunkan tingkat kesulitan musuh pada level tertentu, memperbaiki desain tombol, atau mengatur ulang urutan misi.

Dengan menerapkan proses pengujian dan debugging yang terstruktur, kamu dapat memastikan produk akhir yang lebih stabil dan berkualitas tinggi, memberikan pengalaman yang lebih baik bagi pemain.

Aktivitas 6.3 Praktik Tim Kerja

Kalian tentu sudah mengetahui tentang pengujian gim. Nah, coba kalian praktikkan bagaimana pengujian sebuah gim dilakukan? Lalu bentuklah sebuah tim yang terdiri dari 4 orang, kemudian coba kalian buat sebuah simulasi pengujian gim pada gim yang pernah kalian buat!

B. Optimasi Gim dan Pembaruan



Menerapkan optimasi kinerja gim dan pembaruan dalam pengembangan gim adalah aspek krusial untuk memastikan pengalaman bermain yang mulus dan memuaskan. Berikut adalah penjelasan rinci mengenai kedua aspek tersebut:

1. Optimasi Kinerja Gim



a Tujuan Optimasi

1. Meningkatkan Frame Rate

Memastikan permainan berjalan dengan *frame rate* yang stabil dan tinggi untuk mengurangi lag. *Perry (2019)* menggarisbawahi bahwa frame rate yang stabil sangat penting untuk pengalaman bermain yang baik.

2. Mengurangi Waktu Muat

Mempercepat waktu muat level dan aset. Nash (2016) mencatat bahwa waktu muat yang lama dapat mengganggu alur permainan.

3. Efisiensi Sumber Daya

Mengurangi penggunaan CPU, GPU, dan memori untuk membuat gim dapat berjalan pada berbagai perangkat. Cohen (2018) menekankan pentingnya efisiensi sumber daya dalam pengembangan gim.

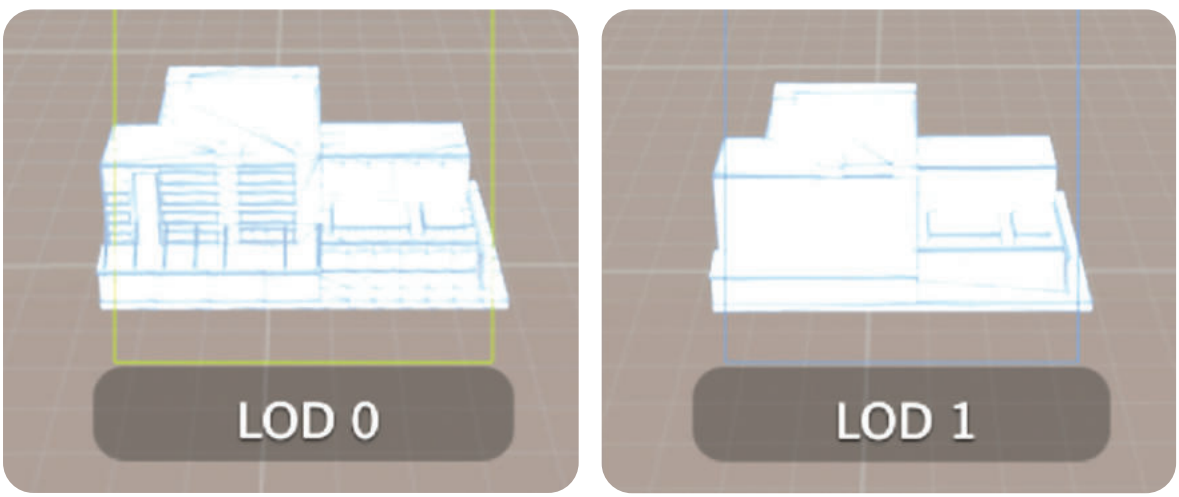
b Aspek yang Perlu Dioptimalkan

1. Grafik dan Visual

a. LOD (Level of Detail)

Gunakan model dengan detail yang berbeda tergantung pada jarak dari kamera. Model yang lebih jauh dapat memiliki detail lebih sedikit. Meyer (2017) menunjukkan bahwa LOD dapat meningkatkan kinerja tanpa mengorbankan kualitas visual.

Unity menjelaskan bahwa penggunaan LOD adalah untuk melakukan efisiensi dari penggunaan *Graphical Processing Unit* (GPU) tanpa mengurangi kualitas (Unity, 2024). Ketika objek berada di tempat jauh, maka detail *object* tidak terlalu terlihat maka lebih baik ketika LOD 1 digunakan karena detail object pada LOD 0 akan memakan lebih banyak memori dalam *performance* tanpa adanya peningkatan kualitas yang signifikan.



Gambar 6.1 Contoh perbedaan Level of Detail sebuah objek ketika dilihat dari dekat (LOD 0) dan dari jauh (LOD 1)
Sumber: Unity (2024)

b. Baking Lighting

Gunakan teknik pencahayaan yang di-*Baked* untuk mengurangi beban perhitungan pencahayaan *real-time*. Perry (2019) menyebutkan bahwa pencahayaan yang dipanggang dapat mengurangi penggunaan CPU. Di dalam *game engine Unity* terdapat tiga tipe pencahayaan yang dijelaskan oleh Unity Documentation yaitu:

Baked

Baked adalah proses di mana game engine, dalam hal ini Unity, melakukan perhitungan pencahayaan terlebih dahulu dan menyimpannya sebagai data cahaya. Data ini kemudian akan digunakan kembali sepanjang permainan (Unity, 2024). Karena perhitungan cahaya yang kompleks sudah dilakukan sebelum permainan dimulai, performa gim saat dijalankan tidak akan terlalu berat. Hal ini juga mengurangi beban dalam proses *rendering* bayangan.

Pencahayaan *baked* biasanya kamu gunakan untuk sumber cahaya yang tidak banyak berubah, seperti latar belakang atau lingkungan yang statis. Namun, teknik ini memiliki keterbatasan. Objek dinamis, seperti karakter atau benda yang bergerak, tidak akan menerima cahaya atau bayangan dari hasil *baked* tersebut (Unity, 2024).

Realtime

Sebaliknya, *realtime lighting* adalah pencahayaan yang dihitung secara langsung saat permainan berlangsung. Unity akan menghitung pencahayaan setiap saat selama permainan aktif. Jenis pencahayaan ini cocok untuk menciptakan suasana yang dinamis, seperti cahaya dari lampu yang meredup, api, atau cahaya yang berubah tergantung pada pergerakan objek di dalam gim.

Cahaya *realtime* sangat berguna saat kamu ingin menciptakan interaksi visual yang realistis dengan objek dinamis. Namun, kekurangannya adalah membutuhkan sumber daya yang lebih besar dibandingkan *baked lighting*, terutama pada perangkat dengan spesifikasi rendah.

2. *Teksturing*

a. Atlas Tekstur

Menurut Unity Documentation, Atlas Tekstur ini disebut juga *Sprite Atlas* merupakan satu tekstur yang menggabungkan beberapa tekstur menjadi satu. Unity hanya akan memanggil satu kali yaitu tekstur ini dibandingkan memanggil beberapa kali tekstur yang berbeda, hal ini menyebabkan terdapat pengurangan jumlah pemanggilan dan meningkatkan performa dari gim tersebut (Unity, 2024).

Misalkan pada gim Mr. Alien keseluruhan tekstur yang kita perlukan adalah 34 tekstur untuk seluruh komponen di dalam gimnya seperti karakter, item, *environment*, musuh, dan sebagainnya. Pada umumnya Unity akan memanggil spesifik 34 tekstur tersebut secara bergantian sehingga membuat pemanggilan tersebut tidak efisien. Lalu untuk mengurangi jumlah pemanggilan, maka kamu melakukan klasifikasi dari 34 tekstur tersebut menjadi beberapa klasifikasi seperti karakter, item, *environment*, musuh, dan item lain. Berarti di dalam gim Mr. Alien kita akan membuat 5 Atlas Tekstur untuk merepresentasikan 34 tekstur yang diperlukan. 5 Atlas Tekstur itu akan dibagi berdasarkan komponen seperti Atlas Tekstur Karakter, Atlas Tekstur Item, Atlas Tekstur *Environment*, Atlas Tekstur Musuh, dan Atlas Tekstur Item Lain. Dengan menggunakan Atlas Tekstur maka kita telah melakukan optimisasi dari performa gim yang kita buat.

b. Tekstur Kompresi

Gunakan format tekstur terkompresi untuk mengurangi ukuran file dan memori yang digunakan. Duvall et al. (2015) menekankan bahwa kompresi tekstur dapat meningkatkan kinerja. Unity menjelaskan bahwa saat kamu menggunakan sebuah gambar, misalnya dalam format JPEG atau PNG, gambar tersebut memang tampil seperti biasa di dalam editor. Namun, saat gim dijalankan, ternyata hardware grafis (GPU) tidak menggunakan format asli tersebut. Sebaliknya, gambar akan dikonversi ke format khusus yang telah dikompresi dan disesuaikan dengan platform tempat gim dijalankan. Hal ini dilakukan untuk mengoptimalkan performa grafis dan mempercepat proses rendering selama permainan berlangsung. Hal ini menyebabkan adanya perbedaan kompresi suatu aset ke platform *mobile* dengan kompresi suatu aset ke platform PC (Unity, 2024).

Komponen di dalam tekstur yang sangat berpengaruh dalam penentuan performa adalah *Bits per pixel*. *Bits per pixel (bpp)* ini adalah jumlah ukuran file yang diperlukan untuk menyimpan satu pixel di dalam tekstur. Semakin tinggi bppnya, maka semakin tinggi kualitas visual yang diberikan, tetapi berpengaruh kepada besarnya ukuran sebuah aplikasi, menambah waktu *loading* dan menambah penggunaan memori ketika menjalankan permainan (Unity, 2024).

Unity juga memberikan *referensi* terkait beberapa contoh format tekstur yang dapat digunakan di dalam *game engine* Unity, seperti:

Tabel 6.1 Contoh format teksture dalam game engine Unity

Format Tekstur	Deskripsi	Bits per pixel	Ukuran
RGB(A) *Compressed* BC7	Kompresi dari RGB/RGBA	8	1
RGBA 64 bit	RGBA yang tidak terkompresi dengan kualitas tinggi	64	8
RGBA 32 bit	RGBA yang tidak terkompresi dengan kualitas normal	32	4
RGB *Compressed* DXT1	Kompresi RGB (BC1)	4	0.5

*Ukuran untuk 1024x1024 tekstur dalam MB

Kita bisa melihat perbedaan yang cukup signifikan dari sisi ukuran antara RGBA 32 Bit dengan DXT1, di mana kompresi DXT1 memiliki ukuran 1/8 dari ukuran sebelum di kompresi pada format 32 bit.

3. Pengelolaan Memori

a. *Pooling* Objek

Gunakan teknik *pooling* untuk mengelola objek yang sering dibuat dan dihancurkan, mengurangi *overhead* dari alokasi memori. Unity menjelaskan di dalam dokumentasi manualnya bahwa objek *pooling* adalah sebuah teknik untuk melakukan optimasi dari sisi pemrograman di mana secara inti, dibandingkan melakukan pembuatan dan penghapusan permanen secara berulang seperti fungsi Create() dan Destroy() pada unity, lebih baik menggunakan kembali GameObject yang sudah ada tanpa perlu membuat maupun menghapus sebuah Game Object (Unity ,2024).

Misalkan di dalam gim Mr.Alien terdapat sebuah terowongan yang dilewati oleh beberapa kendaraan tanpa henti. Pada iterasi awal kita membuat fungsi Create() untuk selalu membuat *GameObject* ketika kendaraan mulai berjalan di awal terowongan dan membuat fungsi Destroy() ketika berada di akhir terowongan. Hal ini menyebabkan sebuah penurunan performa ketika kendaraan yang dibuat sebesar 30 mobil dan harus melakukan *destroy* sebanyak 30 mobil secara berulang.

Dengan objek *Pooling*, pada awal permainan sistem akan melakukan pembuatan 30 mobil dengan Create() dan menaruhnya di awal terowongan dan menonaktifkan semuanya, menunggu permainan dimulai. Lalu setelah permainan dimulai, maka *GameObject* kendaraan akan diaktifkan satu per satu di awal terowongan dan berjalan menuju akhir terowongan di mana pada akhir terowongan *GameObject* akan dinonaktifkan dan dikembalikan posisinya pada awal terowongan.

Dengan pendekatan seperti itu maka tidak perlu adanya fungsi Create() dan Destroy() ketika permainan dimulai, di mana meningkatkan performa gim secara signifikan. Kalian bisa mencoba melakukan eksplorasi pendekatan yang serupa untuk meningkatkan performa dari gim yang kalian buat.

b. Garbage Collection

Minimalkan penggunaan objek yang tidak perlu untuk mengurangi frekuensi pengumpulan sampah (*garbage collection*) yang dapat mengganggu kinerja. Meyer (2017) mencatat bahwa pengelolaan memori yang baik penting untuk kinerja yang optimal. Hal yang bisa kalian lakukan adalah melakukan beberapa optimisasi di dalam kode.

Misalkan pada kasus ini kalian mempunyai sebuah *array* berbentuk string dengan nama stringArray yang berisi [“A”, “B”, “C”, “D”, dan “E”] lalu kalian ingin menggabungkan keseluruhan isi *array* menjadi sebuah kalimat, lalu kalian membuat kode seperti ini:

```
using UnityEngine;
public class ContohBilangan: MonoBehaviiour
{
string ConcatExample (string[] stringArray)
{
       string result = “”;
       for (int i = 0; i < stringArray.Length; i++)
       {
              result += stringArray[i];
}
return result;
}
}
```

Pada kasus ini, memang betul pada akhirnya akan terdapat hasil dimana string menjadi "ABCDE", tetapi terdapat beberapa string lain yang tidak perlu dibuat dan tidak digunakan seperti "A", "AB", "ABC", "ABCD". Hal ini akan secara exponential bertambah juga jumlah huruf yang dimasukan lebih banyak. Maka dari itu lebih baik mengganti "result += stringArray[i]" menjadi 'stringBuilder.append(stringArray[i]"

Dengan perubahan tersebut maka operasi di dalam sistem langsung ke "ABCDE" karena melakukan appends bukan diulang menjadi 5 result "A", "AB", "ABC","ABCD","ABCDE" yang masing-masing . Ini adalah contoh dalam memanfaatkan sebuah pendekatan yang lebih teroptimalisasi untuk mengurangi sebuah operasi pemrograman yang berpengaruh kepada aspek *Garbage Collection*.

4. Sistem *Physics*

Kurangi jumlah objek yang terkena simulasi fisika secara *real-time*, atau gunakan simulasi fisika yang lebih sederhana untuk objek yang tidak terlalu penting. O'Neil (2021) mengingatkan bahwa penyederhanaan simulasi fisika dapat meningkatkan kinerja.

5. Kode dan Logika Permainan

Gunakan alat *profiling* untuk mengidentifikasi bagian dari kode yang memakan waktu lama dan perlu dioptimalkan. Cohen (2018) menyatakan bahwa *profiling* membantu menemukan *bottlenecks* dalam kode. Di dalam dokumentasi resmi Unity, dijelaskan bahwa game engine ini memiliki fitur *profiler* yang bisa kamu gunakan untuk memantau berbagai aspek performa secara detail. Melalui alat ini, kamu bisa melihat bagaimana kode pemrograman berjalan selama gim dimainkan, serta bagaimana aset digunakan di dalam gim.

Dengan memanfaatkan alat ini, kamu bisa mengetahui aset mana yang memakan banyak sumber daya dan mungkin menjadi penyebab performa gim menurun. Dari sini, kamu bisa mengambil keputusan untuk memperbaiki atau mengoptimalkan bagian tersebut agar gim berjalan lebih ringan dan lancar di berbagai perangkat.



Gambar 6.2 Contoh Program Profiler
Sumber: Unity (2024)

C Pengujian Kinerja

1. Benchmarking

Lakukan pengujian kinerja untuk mengukur *frame rate*, waktu muat, dan penggunaan sumber daya. Uji di berbagai perangkat untuk memastikan performa yang konsisten. Johnson (2021) mencatat bahwa *benchmarking* membantu memastikan kinerja yang baik di berbagai platform.

2. Feedback Pengguna

Selain pengujian teknis, kamu juga perlu mendengarkan umpan balik dari pemain, khususnya saat kamu melakukan pengujian beta. Terkadang, ada masalah kinerja yang tidak terlihat saat kamu melakukan pengujian sendiri di lingkungan pengembangan. Duvall et al. (2015) menyarankan bahwa umpan balik dari pengguna akhir sangat berharga dalam proses pengujian.

2. Pembaruan Pengembangan Gim



Dalam pengembangan gim, pembaruan adalah bagian penting yang perlu kamu lakukan secara berkala. Tujuannya tidak hanya untuk memperbaiki kesalahan, tetapi juga untuk menjaga minat pemain dan memastikan gim tetap menyenangkan dimainkan:

a Tujuan Pembaruan

1. Memperbaiki *Bug*

Mengatasi masalah dan bug yang ditemukan setelah peluncuran. Perry (2019) menyebutkan bahwa pembaruan penting untuk menjaga kualitas gim.

2. Menambah Konten Baru

Memperkenalkan fitur baru, karakter, atau level untuk menjaga minat pemain. O'Neil (2021) menekankan bahwa konten baru dapat menarik kembali pemain yang sudah lama tidak bermain.

3. Meningkatkan Kinerja

Mengoptimalkan aspek-aspek yang mungkin mempengaruhi pengalaman bermain. Cohen (2018) menunjukkan bahwa pembaruan dapat membantu menjaga pengalaman bermain tetap segar.

b Proses Pembaruan

1. Pengumpulan Umpan Balik

Mendengarkan umpan balik pemain setelah rilis untuk mengidentifikasi area yang perlu diperbaiki atau ditingkatkan. Shaw (2019) menekankan bahwa umpan balik pemain sangat berharga untuk pengembangan berkelanjutan. Dalam gim Mr.Alien pengumpulan umpan balik memiliki tujuan untuk memastikan apakah gim memiliki tingkat kesulitan dan kejelasan instruksi. Hasil dari pengumpulan umpan balik adalah laporan rekomendasi dari hasil pengujian dengan penjelasan objektif dari pengujian tersebut.

2. Perencanaan Pembaruan

Rencanakan pembaruan berdasarkan prioritas, mulai dari *bug* kritis hingga fitur baru yang diinginkan oleh pemain. Johnson (2021) mencatat bahwa perencanaan yang baik membantu mengarahkan fokus tim. Dalam pengembangan gim Mr.Alien perencanaan pembaruannya adalah menambah musuh di dalam level sehingga level menjadi lebih menantang dibanding sebelumnya.

3. Pengujian Sebelum Rilis

Uji semua perubahan dalam pembaruan secara menyeluruh untuk memastikan tidak ada *bug* baru yang diperkenalkan dan semua fitur berfungsi dengan baik. Duvall et al. (2015) mengingatkan bahwa pengujian sebelum rilis sangat penting untuk kualitas. Contohnya, setelah menambah musuh di dalam level dilakukan pengujian kembali dari sisi performa gim, serta apakah terdapat *error* maupun *bug* sebelum diluncurkan ke publik.

4. Peluncuran Pembaruan

Setelah pengujian selesai, luncurkan pembaruan dengan menginformasikan pemain tentang perubahan yang dilakukan dan manfaatnya. Meyer (2017) menyatakan bahwa komunikasi yang jelas dengan pemain penting saat meluncurkan pembaruan. Pada peluncuran ini, kita harus bisa melakukan definisi dari *key performance index* yang akan di monitoring pada Pasca-Pembaruan.

5. Monitoring Pasca–Pembaruan

Pantau kinerja gim setelah pembaruan untuk memastikan tidak ada masalah baru yang muncul dan tanggapi umpan balik pemain. Cohen (2018) menekankan bahwa monitoring pasca-pembaruan membantu menjaga kualitas. Setelah diluncurkan, maka secara berkala kita akan mengecek apakah terdapat peningkatan dari *key performance index*, jika tidak terdapat peningkatan maka proses harus diulangi kembali sampai dengan ada peningkatan dari *key performance index.*

Dengan menerapkan pengujian dan *debugging* yang terstruktur serta optimasi dan pembaruan secara sistematis, pengembang gim dapat menciptakan pengalaman yang lebih baik dan berkelanjutan bagi pemain, serta meningkatkan kualitas produk secara keseluruhan.



Uji Kompetensi

Coba kamu jawab pertanyaan di bawah ini untuk mengetahui apakah kamu sudah memahami materi tentang Pengujian Gim dalam bab ini!

1. Bagaimana cara kamu melakukan *debugging* pada sebuah gim?
2. Bagaimana cara kamu melakukan optimasi pada sebuah gim dan apa saja yang harus di perhatikan saat kamu melakukan sebuah optimasi gim?



Pengayaan

Bagi kamu yang sudah memahami tentang materi dalam bab ini, kamu bisa melakukan eksplorasi lebih mendalam tentang pengujian gim menggunakan aplikasi lain yang tidak dibahas dalam bab ini! Kamu juga bisa mempelajari dan mempraktekkan materi yang ada dalam buku "*Game Coding Complete, Fourth Edition*" oleh Mike "MrMike" McShaffry dan David Rez Graham.



Refleksi

Setelah kalian mempelajari bab ini, sekarang coba kalian melakukan refleksi diri dengan menjawab pertanyaan berikut ini:

1. Apa yang kamu rasakan setelah memahami materi Pengujian Gim?
2. Apakah kamu semakin tertarik untuk menyelesaikan pengembangan gim terbaikmu?
3. Apakah kamu semakin yakin untuk terjun dalam dunia pengembangan gim ini?
4. Jika yakin, bagian mana dalam materi bab ini yang benar-benar kamu kuasai dan bagian mana yang perlu kamu pelajari lebih lanjut lagi?
5. Menurut kamu *soft skill* apa yang harus kamu asah dan kembangkan untuk bisa bekerja di dunia pengembangan gim ini?
6. Dan bagaimana caranya agar kamu bisa meningkatkan *soft skill* tersebut?

Glosarium . . .

Aset Audio	:	Elemen suara dalam permainan, seperti musik latar, efek suara, dan dialog.
Aset Visual	:	Elemen grafis yang digunakan dalam permainan, termasuk karakter, latar, objek, dan antarmuka pengguna.
Breakpoint	:	Titik dalam kode di mana eksekusi program berhenti sementara untuk analisis.
Bugs	:	Kesalahan dalam kode perangkat lunak yang menyebabkan program tidak berfungsi dengan benar.
Crash	:	Keadaan di mana perangkat lunak atau permainan berhenti berfungsi mendadak.
Cutscenes	:	Potongan video yang menyajikan bagian penting dari cerita secara sinematik.
Dubbing	:	Proses merekam suara baru untuk menggantikan suara asli dalam media.
Debug	:	Proses mencari dan memperbaiki kesalahan dalam kode perangkat lunak.
Environmental Storytelling	:	Teknik menggunakan lingkungan untuk menceritakan cerita tanpa kata-kata.
Feedback	:	Informasi yang diberikan kepada pemain mengenai dampak dari tindakan mereka.
Game Design Balancing	:	Proses penyesuaian elemen dalam permainan untuk pengalaman bermain yang adil dan menyenangkan.

Game Design Document (GDD)	:	Dokumen yang merinci semua aspek dari sebuah gim.
Game Design Prototype	:	Versi awal permainan untuk menguji konsep dan mekanika.
Game Engine	:	Perangkat lunak yang menyediakan alat untuk membuat dan menjalankan permainan.
Game Loop	:	Siklus dasar yang menjadi fondasi permainan, mencakup Input, Update, dan Render.
Game Papan	:	Permainan yang dimainkan di atas papan menggunakan komponen fisik.
Game Video	:	Permainan yang dimainkan melalui perangkat elektronik.
Iterasi	:	Pengulangan terus menerus dari Game Loop selama permainan berlangsung.
Key Concept Art	:	Ilustrasi visual yang menggambarkan ide awal tentang elemen dalam game.
Level Gim (Game Level Concept)	:	Bagian terpisah dari gim yang menandakan kemajuan pemain.
Mekanika Gim	:	Elemen dasar yang menentukan bagaimana sebuah permainan berfungsi.
Game Mechanic Concept	:	Aturan dan sistem yang mengatur interaksi dalam gim.
Objek Dinamis	:	Elemen yang dapat bergerak atau berubah dalam permainan.
Objek Statis	:	Elemen yang tidak berubah posisi atau sifatnya selama permainan.
Optimasi	:	Upaya meningkatkan kinerja perangkat lunak.

Pencapaian/Piala (Achievements/Trophies)	:	Penghargaan untuk keberhasilan pemain.
Poin/Skor	:	Nilai yang diperoleh pemain setelah menyelesaikan tugas.
Poin Pengalaman (Experience Points–XP)	:	Poin yang memungkinkan karakter naik level.
Progression Loop	:	Siklus fokus pada peningkatan karakter/level.
Property Design	:	Proses menciptakan objek statis dalam game.
Puppeteer	:	Teknik untuk mengendalikan dan menganimasi karakter dalam permainan.
Replayability	:	Kemampuan permainan untuk menarik pemain untuk memainkannya kembali.
Rigging	:	Proses membuat kerangka untuk model 3D agar bisa di animasi.
Struktur Data	:	Cara mengorganisir dan menyimpan data dalam program.
System Mechanics	:	Kumpulan mekanik permainan yang saling terhubung.
Texturing	:	Proses menerapkan gambar pada permukaan objek 3D.
UI/UX	:	UI merujuk pada tampilan, UX berfokus pada pengalaman pengguna secara keseluruhan.
Umpan Balik Pemain	:	Informasi dari pemain mengenai pengalaman mereka.
Kemajuan Cerita (w)	:	Bagian dari alur cerita yang terungkap melalui misi.

Daftar Pustaka . . .

Adams, Ernest, dan Andrew Rollings. *Game Mechanics: Advanced Game Design*. New York: Addison-Wesley, 2014.

Amsden, J. *Game Testing and Debugging*. 2020.

Bach, J. *Effective Game Testing Techniques*. 2015.

Baker, J. *Game Engine Architecture: The Definitive Guide to the Development of Game Engines*. New York: Game Engine Press, 2019.

Baraff, D., dan A. Witkin. "Large Steps in Cloth Simulation." *ACM Transactions on Graphics* 17, no. 3 (1998): 43-54.

Barrera, Ray, et al. *Unreal Engine 4 for Design Visualization*.

Bates, Kevin. *Game Design: Theory and Practice*.

Birn, Jeremy. *Digital Lighting and Rendering*.

Black, R. "Iterative Design in Game Development." 2022.

Block, Bruce. *The Visual Story*.

Bogost, Ian. *Play Anything*. New York: Basic Books, 2016.

Brandon, Alexander. *Audio for Games*.

Brathwaite, Brenda, dan Ian Schreiber. *Challenges for Game Designers*. Boston: Cengage Learning, 2009.

CD Projekt Red. *The Art of The Witcher 3*.

Chan, John L. F. *Creating Interactive Characters in 3D*.

Colavito, Anna G. *The Art of Costume Design*.

Crossley, Kevin. *Character Design from the Ground Up*.

Dark Horse Books. *The Art of The Legend of Zelda: Breath of the Wild*.

Davis, J. *Dynamic Objects in Game Development*. New York: Game Design Press, 2019.

Davis, K. *Character Dialogues and Narrative in Games*. 2022.

Dyer, J. L. *Game Marketing Essentials*.

Eberly, D. *3D Game Engine Architecture: A Practical Approach to Real-Time Rendering*. Boston: Morgan Kaufmann, 2003.

Egenfeldt-Nielsen, Simon, et al. *Understanding Video Games: The Essential Introduction*. London: Routledge, 2013.

Ericson, Christer. *Real-Time Collision Detection*.

Ferrone, Harrison. *Learning C# by Developing Games with Unity 2021*.

Fenlon, J. *Environmental Interactivity in Game Design*. 2018.

Fischer, L. *Character Design in Video Games*. 2021.

Fischer, J. "Web-Based Prototyping for Interactive Games." *Journal of Game Development* 8, no. 4 (2021): 56-72.

Foley, J. D., et al. *Computer Graphics: Principles and Practice*. Boston: Addison-Wesley, 1996.

Ford, L. *Iterative Design: Principles and Practices*. San Francisco: Design Books, 2020.

Friedman, A. *Balancing Act: Game Design for Fair Play*. London: Game Balance Press, 2019.

Fullerton, Tracy. Game Design Workshop: Designing, Prototyping, and Playtesting Games. 4th ed. CRC Press, 2004.

Gaffney, Frank L. *The Future of Game Design*.

Graham, R. *Game Development Essentials: A Comprehensive Guide*. New York: Game Development Press, 2016.

Graham, R. *Game Mechanics: Balancing Static and Dynamic Elements*. London: Game Mechanics Publishing, 2020.

Gregory, Jason. *Game Engine Architecture*.

Greer, D. *Documenting the Design Process*. 2021.

Greene, R. "Creating Engaging Game Experiences." *Game Studies* 15, no. 1 (2015): 45-62.

Greene, S. *Market Research in Game Development*. 2015.

Gurney, James. *Color and Light*.

Hargreaves, Kim B. *Game Mechanics: Describing the Play Experience*.

Hawkins, R. *Post-Release Content Updates*. 2023.

Higgins, J. *Understanding Game Genres and Their Audio Needs*. 2014.

Holt, A. *Environmental Storytelling in Games*. 2019.

Howard, L. *Game Development Best Practices*. 2023.

Himekawa, Akira. *The Art of The Legend of Zelda: Breath of the Wild*.

Hegeman, R. *Vehicle Dynamics in Gaming*. Toronto: Vehicle Gaming Publishers, 2016.

Hegeman, R. *Game Physics: Principles and Techniques*. San Francisco: Game Physics Press, 2016.

Iversen, P. *Enhancing Player Experience through Audio*. 2022.

Isbister, K. *How to Playtest: A Guide for Game Developers*. Seattle: Game Testing Press, 2016.

Johnson, M. *Integrating Visuals in Game Design*. 2020.

Johnson, R. *Debugging and Performance Optimization in Game Development*. 2021.

Kallmann, M. *AI Techniques for Game Development*. San Francisco: AI Game Press, 2015.

Katz, J. *Audio Engineering for Video Games*. New York: Audio Press, 2015.

Kelley, J. *The Art of Game Lighting*. Toronto: Game Light Press, 2017.

Kelley, J. *Audio Quality in Game Development*. 2017.

Kessler, A. *Color Psychology in Game Design*. 2016.

Kessler, R. *The Art of Game Balancing: Strategies and Techniques*. Chicago: Game Design Press, 2016.

Kavanagh, T. R. *Substance Painter*.

Kearney, A. J. *Visual Effects for Games*.

Klok, Sander H. L. *The Audio of Game Design*.

Klug, C., dan M. McShaffry. *Introduction to Game Development*. Course Technology, 2010.

Koster, Raph. *A Theory of Fun for Game Design*. Scottsdale: Paraglyph Press, 2004.

Klososky, Scott. *Documentation for Designers*.

Lannon, John M. *Technical Communication*.

Li, Y. *Curating Visual References for Games*. 2023.

Lucas, H. *Testing Audio in Gameplay Contexts*. 2021.

Martin, A. *Guidelines for Audio Asset Integration*. 2022.

Martin, R. *Licensing Music for Game Development*. 2023.

Mason, T. *User Interface Design for Games*. 2021.

McFarlane, Tom. *The Art of Iteration*.

McGuire, M., et al. *Integrating Audio Middleware in Game Development*. 2012.

Meyer, J., et al. *Game Development: Shading Techniques and Applications*. Los Angeles: Game Press, 2017.

Meyer, J., et al. *Game Audio Design: Principles and Practices*. New York: Game Press, 2017.

Meyer, L. "Collecting and Analyzing Player Feedback." *Journal of Game Research* 11, no. 3 (2018): 12-29.

Meyer, S. *Concept Art in Game Development*. 2018.

Meyer, T. *Interactivity in Gaming: The Role of Dynamic Objects*. Chicago: Interactive Media Books, 2017.

Miller, T. *Emotional Design in Games*.

Moore, K. *Game Music Composition*.

Novak, Jeannie. *Game Development Essentials: An Introduction*. Cengage Learning, 2011.

Nielsen, J. *User Interface Design for Games*. Boston: UX Design Press, 2015.

Nelson, B. *Accessibility in Game Design*.

Nelson, T. *2D/3D Prototyping in Game Design*. Toronto: Graphics in Games Press, 2019.

Pallister, Kim. *Game Mechanics: Describing the Rules of Play*.

Parker, J. *Sound Effects and Realism in Gaming*. 2019.

Peirce, S. D. *The Future of Game Design*.

Perron, Bernard. *Game Design: A Practical Approach*.

Picard, C. *Game Design Iteration: How to Iterate Effectively*. San Francisco: Design Press, 2018.

Picard, L. *Visual Reference in Game Development*.

Phelps, J. *Innovation in Game Development: Creating Engaging Experiences*. New York: Game Innovators, 2019.

Phelps, M. *User Experience in Gaming*.

Reed, E. *Budgeting for Sound Engineers*. 2019.

Rich, F. *Documenting Audio Production Processes*. 2018.

Roberts, David L. *3D Game Modeling*.

Rodriguez, C. *Visual Effects in Games*. 2019.

Rogers, David L. *Creating Game Art for 3D Engines*.

Rogers, S. *Game Design Fundamentals*. 2014.

Salen, K., dan E. Zimmerman. *Rules of Play: Game Design Fundamentals*. Cambridge: MIT Press, 2004.

Schimmel, Timothy P. *Effective Documentation for Game Development*.

Schiller, D. *Narrative Through Dialogue in Games*. 2021.

Sheldon, Lee. *The Multiplayer Classroom: Designing Coursework as a Game*. Boston: Cengage Learning, 2011.

Steele, B. *Audio Performance and Format Considerations*. 2020.

Stevens, A. *Music and Game Mood*. 2020.

Stevens, R. *Tools for Game Prototyping: From Paper to Digital*. New York: Game Tools Press, 2020.

Swink, Steve. *Game Feel: A Game Designer's Guide to Virtual Sensation*. Boca Raton: CRC Press, 2009.

ter Hennepe, Sarah B. P. J. W. *Environmental Storytelling in Games*.

Tobias, T. *Multiplayer Game Development: A Comprehensive Guide*. Seattle: Multiplayer Press, 2018.

Torrence, J. *Researching Visual Styles for Games*.

Turner, S. *Globalization of Game Audio*. 2023.

Walsh, B. *Visual Design in Games: A Comprehensive Guide*. San Francisco: Game Design Press, 2018.

Weaver, A. *The Psychology of Video Games*. Cambridge: MIT Press, 2018.

Williams, D. S. *Game Programming for Beginners*. Cengage Learning, 2010.

Williams, P. *Ambient Soundscapes in Gaming*. 2022.

Wong, Anthony. *Animating Characters in 3D*.

Wright, T. *Detailing Audio Effects in Games*. 2018.

Young, J. *Diversity in Game Music*.

Zhang, Q. *Performance Optimization for Game Audio*. 2022.

Zhang, Y., et al. *Physically Based Rendering: Principles and Techniques*. New York: Graphics Press, 2018.

Bennett, P. *Audio Design Essentials for Games*, 2015.

Erleben, K., et al. *Game Development with Unity: An Introduction to Game Engine Architecture*. 2009.

Fowler, G., dan A. Turing. *Audio in Games: Testing and Feedback Strategies*. 2019.

Hahn, R. *Game Audio Implementation: A Practical Guide*.

Jorgensen, P. *Software Testing and Quality Assurance*. 2018.

McConnell, S. *Code Complete: A Practical Handbook of Software Construction*. Redmond: Microsoft Press, 2004.

Nash, K. *Game Quality Assurance: A Comprehensive Guide*.

O'Neil, J. *User-Centric Game Design and Testing*. 2021.

Perry, D. *Game Performance Optimization*. 2019.

Shaw, C. *Game Development: Insights on Player Feedback*. 2019.

Wang, H. *Effective Logging for Debugging in Game Development*. 2020.

Zhang, Q., et al. *Advancements in Spatial Audio for Interactive Experiences*. 2018.

Rouse, Richard. *Gamedesign:theory&practice* , WordwarePublishing,Inc., 2004

Schell, Jesse, The Art of Game Design, Morgan Kaufmann Publishers, 2008

Adams, Ernest, Fundamental of Game Design Third Edition, Pearson Education, Inc., 2014

McCabe, Patrick, Create Computer Game Design And Bulid Your Own Game, Wiley, 2018

Penton, Ron , Data Structures For Game Programmers, Premier Press, 2003

Artikel:

https://www.kominfo.go.id/content/detail/8440/kemkominfo-luncurkan-situs-indonesia-game-rating-system-igrs/0/berita_satker

https://www.interaction-design.org/literature/topics/games-user-research?srsltid=AfmBOopNAKzF06sK83tpskl75ma97xKjygh510-RsgvsBDeEZsiT9LvS

Daftar Sumber Gambar . . .

Gambar 1 : https://assets.nintendo.com/image/upload/ar_16:9,c_lpad,w_656/b_white/f_auto/q_auto/ncom/software/switch/70010000001130/c42553b4fd0312c31e70ec7468c6c9bccd739f340152925b9600631f2d29f8b5

Gambar 2 : https://www.gameuidatabase.com//uploads/uxthumbs/Achtung-Cthulhu-Tactics_25347.jpg

Gambar 3 : https://www.gameuidatabase.com/uploads/The-Legend-of-Zelda-Tears-of-the-Kingdom06242023-043651-36219_thumb.jpg

Gambar 4 : https://www.gameuidatabase.com/uploads/Yoshis-New-Island05282021-102302-83345.jpg

Gambar 5 : https://www.gameuidatabase.com/uploads/KartRider-Rush01262021-102941-94186.jpg

Gambar 6 : https://www.gameuidatabase.com/uploads/Need-for-Speed-Heat07152020-071740-76874_thumb.jpg

Gambar 7 : https://play-lh.googleusercontent.com/rmpkoPEeqnyr3CA-MGSTY8Glm3xX-r5KQEDnv-vnYLyu-ujERnZ1B62s3YpJ9W3ItEA=w526-h296-rw

Gambar 8 : https://www.gameuidatabase.com/uploads/Agents-of-Mayhem06242021-104952-81847.jpg

Gambar 9 : https://www.gameuidatabase.com/uploads/Assassins-Creed-Valhalla05182021-093036-66867.jpg

Gambar 10 : https://shared.cloudflare.steamstatic.com/store_item_assets/steam/apps/1158160/ss_197d35ede9bd2a4f0ad74ba085c54295cac36416.600x338.jpg?t=1726190192

Gambar 11 : https://upload.wikimedia.org/wikipedia/commons/thumb/7/78/Monopoly_board_on_white_bg.jpg/375px-Monopoly_board_on_white_bg.jpg

Gambar 12 : https://www.gameuidatabase.com/uploads/DRIVE-Switch06022023-102122-47857.jpg

Gambar 13 : https://www.gameuidatabase.com/uploads/Atelier-Ryza-Ever-Darkness-amp-the-Secret-Hideout05262020-035750_thumb.jpg

Gambar 14: https://www.gameuidatabase.com/uploads/Agents-of-Mayhem06242021-104956-86383.jpg

Gambar 15: https://www.gameuidatabase.com/uploads/Luigis-Mansion-206132021-111502-7829.jpg

Gambar 16: https://www.gameuidatabase.com/uploads/Borderlands-The-Pre-Sequel06252021-125559-17951.jpg

Gambar 17: https://shared.cloudflare.steamstatic.com/store_item_assets/steam/apps/3151360/extras/Agni_Hargai_4.png?t=1728834323

Gambar 18: https://images.sftcdn.net/images/t_app-cover-l,f_auto/p/51fa7896-21df-4a34-829e-60b1e4cbc3ec/650223266/rocky-rampage-wreck-em-up-screenshot.png

Gambar 19: https://www.gameuidatabase.com/uploads/Chivalry-206202021-111323-92707.jpg

Gambar 20: Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 21: https://shared.cloudflare.steamstatic.com/store_item_assets/steam/apps/570940/ss_626cc310dc9ac7fb146011582c864a35e5f3e381.116x65.jpg?t=1726158603

Gambar 22: https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/2447110/ss_55f29bcf0db4a6f417eb3e1c501a866a806d8e1f.116x65.jpg?t=1724079552

Gambar 23: https://images.uesp.net/thumb/d/d8/SR-menu-Skills_Menu.jpg/400px-SR-menu-Skills_Menu.jpg

Gambar 24: https://shared.cloudflare.steamstatic.com/store_item_assets/steam/apps/1888930/ss_c2853eee7ca824c42e008e83f3f45ad42d9e8547.116x65.jpg?t=1728603021

Gambar 25: https://shared.cloudflare.steamstatic.com/store_item_assets/steam/apps/8870/ss_e98deaf0e334206b84c2462276aee98107fa20d0.116x65.jpg?t=1728673518

Gambar 26: https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/1265920/ss_87e900d93e82074d18aab3083387791e0268a6f4.116x65.jpg?t=1724158974

Gambar 27:https://www.gameuidatabase.com/uploads/Super-Mario-Bros01292024-122745-37291_thumb.jpg

Gambar 28 : https://www.gameuidatabase.com/uploads/The-Legend-of-Zelda-Breath-of-the-Wild09302022-055539-19142_thumb.jpg

Gambar 29 : https://www.gameuidatabase.com/uploads/Metroid-Dread10172021-092917-23218_thumb.jpg

Gambar 30: https://www.gameuidatabase.com/uploads/Hitman(2016)04142020-111453_thumb.jpg

Gambar 31: https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/620/ss_6a-4f5afdaa98402de9cf0b59fed27bab3256a6f4.116x65.jpg?t=1725487730

Gambar 32: https://www.gameuidatabase.com/uploads/Overwatch01202022-121228-93564_thumb.jpg

Gambar 33: https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/287700/ss_01482dcddff4bb9d39650cd92e416d6263dccaf1.116x65.jpg?t=1727849445

Gambar 34: https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/504230/ss_4b0f0222341b64a37114033aca9994551f27c161.116x65.jpg?t=1714089525

Gambar 35: https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/1888930/ss_3f1805ecddafacee7f61f87cb8e4624435a83ee3.116x65.jpg?t=1728603021

Gambar 36:`https://www.gameuidatabase.com/uploads/Mirrors-Edge05202021-061709-5958_thumb.jpg

Gambar 37: Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 38: https://assets.nintendo.com/image/upload/ar_16:9,b_auto:border,c_lpad/b_white/f_auto/q_auto/dpr_1.5/c_scale,w_700/ncom/software/switch/70010000001130/1839d571921e3fb19ef48da64c145cb8ce573b07d7390c6350f15291b3905048

Gambar 39: `https://www.gameuidatabase.com/uploads/KartRider-Rush01262021-102941-94186_thumb.jpg

Gambar 40: The Legend of Zelda: Art & Artifacts (2017), Oregon, Dark Horse Books.

Gambar 41: Dark Souls Design Works (2014), Tokyo, FromSoftware.

Gambar 42: The Witcher 3: Wild Hunt Artbook (2015), Warwick, CD Projekt Red.

Gambar 43: Hollow Knight - Wanderer’s Journey (2019), Adelaide, Team Cherry

Gambar 44: Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 45: Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 46: Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 47: Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 48: Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 49: Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 50: Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 51: Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 52: Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 53: Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 54: Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 55: https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/3151360/ss_a121750e9a9ebf39391b9371c01f822369e70799.600x338.jpg?t=1728834323

Gambar 56: https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/257045640/movie.184x123.jpg?t%3D1723768219&sa=D&source=docs&ust=1730132247794912&usg=AOvVaw1kGhajiIiCcByojX42XeV6

Gambar 57 https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/257045640/movie.184x123.jpg?t%3D1723768219&sa=D&source=docs&ust=1730132247796115&usg=AOvVaw0tkOf_FcoCc4ZbPAUyxkUo

Gambar 58. Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 59. Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 60. Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 61. https://pbs.twimg.com/media/GVQ8TwCbAAA1XFM?format%3Djpg%26name%3Dlarge&sa=D&source=docs&ust=1730137299231120&usg=AOvVaw2td9o4Q88MXx1QJhv4mN5a

Gambar 62. https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/2479770/ss_583fb7e78a0a71fb7331bffc0ac88e9cab6500fa.600x338.jpg?t=1729192028

Gambar 63`https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/658500/ss_5fc533275decb44d1dbd507dc6a0b1ed227c6a68.1920x1080.jpg?t=1725270283

Gambar 64 https://drive.google.com/file/d/1eVsruMAjuXwGJBvt5qSDRuKvoAIZ2qCi/view?usp=sharing

Gambar 65 : https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/658500/ss_f393f4c6adda9a6bf8f3e19f5187f3b6236e6c2a.1920x1080.jpg?t=1725270283

Gambar 66 : https://assetstorev1-prd-cdn.unity3d.com/package-screenshot/05e23faa-5067-44a2-a811-c93f6ab76250.webp, https://assetstore.unity.com/packages/vfx/particles/trail-particle-system-196334

Gambar 67 : https://media.fab.com/image_previews/gallery_images/3a8a05df-bc9c-40a6-98ca-5480ddd62394/c1176958-f0f9-47f8-a375-1ecfecd2ae33.jpg
https://www.fab.com/listings/93369071-1075-4fbd-9b01-232b9b9fd28d

Gambar 68 : https://assetstorev1-prd-cdn.unity3d.com/key-image/d32e8afb-4be8-42bb-9905-d8681eca2564.webp
https://assetstore.unity.com/packages/tools/physics/toolkit-for-unity-physics-ballistics-251081

Gambar 69 : https://assetstore.unity.com/packages/tools/behavior-ai/ai-tree-behavior-trees-for-unity-229578
Gambar 70 : https://www.fab.com/listings/17f1bf1f-1125-4cc2-b42b-dbc620e44063
Gambar 71 : https://www.fab.com/listings/87d53674-265e-41ca-8634-5d782ed4abf0
Gambar 72 : https://assetstore.unity.com/packages/tools/animation/rukhanka-ecs-animation-system-241472
Gambar 73 : https://assetstore.unity.com/packages/tools/audio/the-audio-lab-287612
Gambar 74 : https://www.fab.com/listings/3dc8efb5-0b13-4925-a8cb-2ff052281298
Gambar 75 : https://www.fab.com/listings/9755f1dd-2287-4831-a2bb-125bce6a07a3
Gambar 76 : https://assetstore.unity.com/packages/3d/gui/modular-3d-text-in-game-3d-ui-system-247241
Gambar 77 : https://assetstore.unity.com/packages/tools/utilities/performance-tools-80534
Gambar 78 : https://www.fab.com/listings/184a32a0-c550-4103-b537-32be703c3210
Gambar 79 : https://www.fab.com/listings/b79e9fa5-2c21-4cdc-817d-a05d7468e434
Gambar 80 : https://assetstore.unity.com/packages/tools/network/photon-fusion-267958
Gambar 81 : https://assetstore.unity.com/packages/tools/level-design/fabgrid-level-editor-175642
Gambar 82 : https://www.fab.com/listings/17322224-0061-439c-9c90-6b9d92c7199c
Gambar 83 : https://www.fab.com/listings/5ec9302e-639b-4ff9-aff6-cf0816530388
Gambar 84 : https://assetstore.unity.com/packages/tools/utilities/fast-script-reload-239351
Gambar 85 : https://assetstore.unity.com/packages/tools/utilities/profiler-memory-plus-28888
Gambar 86 : https://www.fab.com/listings/1691bfe6-25bd-46cd-973b-7fc9ddde3592
Gambar 87 : https://www.fab.com/listings/1540b36c-79b9-48bb-9123-3ea4e1669450
Gambar 88 : https://assetstore.unity.com/packages/tools/integration/really-simple-steam-integration-248700
Gambar 89 : Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 90 : Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 91 : Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 92 : Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 93 : Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 94 : Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 95 : Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 96 : Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 97 : Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 98 : Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 99 : https://assetstore.unity.com/packages/essentials/starter-assets-thirdperson-updates-in-new-charactercontroller-pa-196526
Gambar 100: Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 101: Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 102: Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 103: Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 104: Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 105 :https://assetstore.unity.com/packages/essentials/starter-assets-thirdperson-updates-in-new-charactercontroller-pa-196526
Gambar 106 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 107 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 108 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 109 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 110 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 111 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 112 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 113 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 114 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 115 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 116 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 117 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 118 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 119 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 120 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 121 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 122 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 123 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 124 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 125 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 126 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 127 Penulis, Muhammad Ardhan Fadhlurrahman
Gambar 128 Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 129 Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 130 Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 131 Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 132 Penulis, Muhammad Ardhan Fadhlurrahman

Gambar 133 Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 134 Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 135 Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 136 Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 137 Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 138 Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 139 Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 140 Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 141 Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 142 Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 143 https://docs.unity3d.com/uploads/Shaders/Shader-NormalDiffuse.jpg

Gambar 144 https://docs.unity3d.com/uploads/Main/StandardShaderNormalMapAircraftSurface.jpg

Gambar 145 https://docs.unity3d.com/uploads/Shaders/Shader-NormalSpec.jpg

Gambar 146 https://docs.unity3d.com/Packages/com.unity.probuilder@4.0/manual/images/UVExamples_Translate.png

Gambar 147 https://cdn-ak.f.st-hatena.com/images/fotolife/h/halya_11/20180326/20180326225943.png

Gambar 148 https://cdn2.unrealengine.com/controlrig-screenshot-01-1918x1055-190322936.png?resize=1&w=1920

Gambar 149. Penulis, Muhammad Ardhan Fadhlurrahman.

Gambar 150 Penulis, Muhammad Ardhan Fadhlurrahman.

Indeks . . .

A

Achievements 13, 306

Aset Audio , vi, vii, 74, 75, 92, 93, 94, 95, 96, 106, 219, 220, 221, 258, 278, 305

Aset Visual , vii, 220, 221, 222, 226, 257, 277, 305

B

Breakpoint 280, 305

Bugs 280, 305

C

Crash 280, 305

Cutscenes 39, 305

D

Debug 160, 164, 280, 305

Debugging , viii, 281, 282, 287, 288, 307, 309, 312

Desain Gim v, 2, 3, 29, 30, 59, 72

Desain Karakter x, 75, 78, 82, 99, 102, 103, 105

Desain Properti 75, 88, 90, 99, 102, 105

dokumen desain artistik 97, 98, 101, 105

Dubbing , vii, 220, 221, 270, 305

E

Environmental Storytelling 39, 305, 309, 311

Environment Design 74

F

Feedback 300, 305, 310, 312

G

Game Design Balancing 108, 305

Game Design Document v, 2, 3, 60, 61, 71, 72, 186, 305

Game Design Prototype vi, 108, 110, 305

Game engine 194, 218

Game Loop 36, 305, 306

Game Mechanic Concept 306

Game Object 196, 297

Gim Papan x, 25, 26, 27

Gim Video ix, 20, 21, 26, 27

I

Imbalan ix, 11, 12, 13, 14, 15, 69

Iterasi 63, 113, 306

K

Karakter x, xi, xiii, xiv, xv, 37, 38, 50, 61, 66, 68, 75, 78, 82, 83, 84, 99, 102, 103, 104, 105, 177, 202, 203, 221, 223, 224, 225, 235, 239, 243, 296

Kemajuan Cerita ix, 14, 306

keseimbangan gim 108, 109, 130, 146, 147, 148, 149, 153

Key Concept Art , vi, 74, 76, 104, 306, 321

L

Level Gim 2, 3, 31, 49, 59, 72, 143, 184, 306

M

Mekanika Gim 2, 17, 31, 35, 36, 42, 59, 72, 306

N

Naratif Gim 2, 40, 59

O

Objek Dinamis xi, xii, 156, 177, 178, 179, 183, 190

Objek Statis xi, 156, 173, 174, 176, 179

Optimasi viii, 255, 267, 280, 281, 293, 306, 322

Optimasi Gim , viii, 281, 293

P

Pemrograman Gim , vii, 155, 157

Pencapaian 13, 306

Pencapaian/Piala 306

Poin ix, 11, 12, 145, 306

Poin Pengalaman ix, 12, 306

Poin/Skor ix, 11, 306

Progression Loop 306

Property Design 74, 306

Prototipe digital 121, 127, 129

Puppeteer xiv, 220, 221, 222, 225, 306

R

Replayability 306

Rigging xiv, 220, 221, 232, 238, 306

S

Skor ix, 11, 65, 306

Story Progression 14, 306

Struktur Data , vii, 156, 157, 159, 172, 218, 306

System Mechanics 42, 306

T

Texturing 220, 221, 226, 306

U

UI/UX 99, 102, 103, 156, 157, 158, 216, 218, 306

Umpan Balik Pemain 306

Profil Pelaku Perbukuan . . .



Penulis | Rini Melati, S. Kom

: rinzie1010@gmail.com

: SMK Negeri 11 Bandung

: Jalan Budi Cilember Kota Bandung

: Rekayasa Perangkat Lunak

Riwayat Pekerjaan/Profesi (10 Tahun Terakhir):

1. Guru di SMK Negeri 11 Bandung tahun 2008 sampai sekarang
2. Narasumber Bimbingan Teknis Training of Trainer (ToT) Pembelajaran Koding dan Kecerdasan Artifisial Jenjang Pendidikan Menengah Batch 5 Juni 2025
3. Narasumber Bimbingan Teknis Training of Trainer (ToT) Pembelajaran Koding dan Kecerdasan Artifisial Jenjang Pendidikan Menengah Batch 3 Mei 2025

Riwayat Pendidikan Tinggi dan Tahun Belajar:

S1: Sistem Informasi STMIK "AMIKBANDUNG" (2000 - 2005)

Judul Buku dan Tahun Terbit (10 Tahun Terakhir):

1. Buku Dasar – Dasar Perangkat Lunak dan Gim SMK/MAK Kelas X, penerbit Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, tahun 2023
2. Buku Pemrograman Dasar untuk SMK Kelas X, penerbit Sarana Pancakarsa Nusa , tahun 2019

Penulis | Muhammad Ardhan Fadhlurrahman, S.Kom M.A



: ardhanfadh@gmail.com

: Separuh Interactive

: Business Park Kb. Jeruk, Ruko,
Jl. Meruya Ilir Raya No.88 blok I-9,
RT.1/RW.5, Meruya Utara, Kec. Kembangan,
Kota Jakarta Barat, Daerah Khusus
Ibukota Jakarta 11620

: Pengembangan Gim

Riwayat Pekerjaan/Profesi (10 Tahun Terakhir):

1. Tim Pengajar, Asisten Dosen, Fakultas Ilmu Komputer, Universitas Indonesia 2016 - 2017
2. UX Researcher, Harmonix Teknologi Peentar, 2017 - 2018
3. Product Manager, Gamification and Retention, Pahamify 2020 - 2022
4. Product Manager, Digital Payments, Tokopedia 2022
5. Regional Game Publishing Manager, The Iterative Collective 2022 - 2024
6. Tenaga Ahli Tim Percepatan Gim Nasional, Ministry of Creative Economy, 2024.
7. Deputi Relasi Industri, Asosiasi Game Indonesia, 2019 - 2025
8. Co-Founder, Separuh Interactive, 2024 - 2025
9. Dosen, Prodi Produksi Media Konsentrasi Pengembangan Gim, Vokasi Universitas Indonesia 2024 - 2025.

Riwayat Pendidikan Tinggi dan Tahun Belajar:

1. S1: Ilmu Komputer, Universitas Indonesia 2012 - 2016
2. S2: MA Games Enterprise, University of South Wales, UK 2017 - 2018



Penelaah | Dr. Asep Wahyudin, S.Kom., M.T.

: away@upi.edu

: UniversitasPendidikanIndonesia

: Information System, Software Engineering, IT for Management, IS/IT Strategic Plan, Information Management, IS/IT Audit, IS/IT Governance, Business Process Management Human Computer Interaction, IT for Education.

Riwayat Pekerjaan :

1. Kepala KBK Software Engineering and Information Management Laboratory, Program Studi Ilmu Komputer UPI, 2021-Sekarang
2. Auditor Mutu Internal UPI, 2025-sekarang
3. Dosen Tetap Program Studi S2 Pendidikan Ilmu Komputer, 2006-sekarang
4. Asesor Lembaga Akreditasi Mandiri (LAM) INFOKOM, 2022-sekarang
5. Kepala Divisi Pengembangan Sistem Informasi pada Direktorat TIKUPI, 2007-2019

Riwayat Pendidikan

1. S1 Sistem Informasi, STMIK Bandung (2000)
2. S2 Teknik Informatika, Institut Teknologi Bandung (2005)
3. S3 Ilmu Komputer, Universitas Indonesia (2019)

Judul Buku dan Tahun Terbit (10TahunTerakhir):

1. Dasar-Dasar Pengembangan Perangkat Lunak dan Gim, untuk SMK/MAK Kelas X, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, 2023
2. Informatika, untuk SMK/MAK Kelas X, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, 2023
3. Dalam Pembelajaran Informatika, Direktorat Jenderal Guru dan Tenaga Kependidikan, Kemendikbudristek, 2023-2024
4. Modul Modul Keilmuan Informatika, Literasi Digital, dan Berpikir Komputasional Dalam Pembelajaran Informatika, Direktorat Jenderal Guru dan Tenaga Kependidikan, Kemendikbudristek, 2023-2024

Informasi Lain dari Penelaah:

1. Author ID Google scholar: https://scholar.google.com/citations?user=9wZGvEEAAAAJ&hl=id
2. Author ID SINTA ID: 5991982
3. Author ID SCOPUS ID: 25646715100



Penelaah | Dr. Ratna Wardani, S.Si., M.T

: ratna@uny.ac.id

: Universitas Negeri Yogyakarta

: Jalan Kolombo No. 1 Yogyakarta

: Rekayasa perangkat lunak, UI/UX, Koding dan Kecerdasan Artifisial, Design Thinking & Computational Thinking, Vocational Education.

Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Dosen
2. Instruktur Koding dan Kecerdasan Artifisial
3. Instruktur Program Profesi Guru
4. Master Tainer Design Thinking & Computational Thinking
5. Asesor LAMDIK
6. Asesor UKIN PPG
7. Asesor Sertifikasi Dosen
8. Asesor BKD

Riwayat Pendidikan

1. S1 FMIPA/Ilmu Komputer, Universitas Gadjah Mada (1995)
2. S2 Fakultas Teknik/Teknologi Informasi, Universitas Gadjah Mada (2003)
3. S3 Fakultas Teknik/Teknologi Informasi, Universitas Gadjah Mada (2011)

Pengalaman Penelaahan/Review Buku dalam (5 Tahun Terakhir)

1. Penilai Buku Teks Pelajaran Teknologi Informasi dan Komunikasi (2009)
2. Penilai Buku Teks Pendamping (BTP) Jenjang SMK/MAK Fase E dan Fase F (2023-2024)
3. Reviewer Pedoman Mata Pelajaran Teknik jaringan Komputer dan Telekomunikasi

Informasi Lain dari Penelaah:

1. Karya Tulis: “Computational Thinking dalam Pembelajaran” M odul PPG Prajabatan bagi Dosen
2. Author ID Google Scholar: https://scholar.google.com/citations?user=aq7_950AAAAJ&hl=id
3. Author ID Sinta: https://sinta.kemdiktisaintek.go.id/authors/profile/6006627/?view=iprs#!



Ilustrator | Yol Yulianto

: yolyulianto@gmail.com

: -

: Taman Rembrandt Blok R.04 No.88

: Ilustrasi

Riwayat Pekerjaan :

1. Ilustrator Majalah Anak Ina tahun 1999 - 2000
2. Ilustrator Gramedia Majalah Kelompok Kompas Gramedia tahun 2001 - 2010
3. Ilustrator Majalah Anak Superkid Junior tahun 2013 - 2016
4. Tim Teknis Ilustrasi Direktorat PAUD Kemdikbud tahun 2017 - 2019
5. Ilustrator dan desainer Modul Pembelajaran SD Pusmenjar Kemdikbud tahun 2020
6. Ilustrator Buku Mapel Kurikulum Merdeka Pusbuk Kemdikbud tahun 2021-2024

Riwayat Pendidikan Tinggi dan Tahun Belajar:

FT Arsitektur Undip Semarang tahun belajar 1991-1996

Judul Buku dan Tahun Terbit (10 Tahun Terakhir):

1. Cerita Rakyat Nusantara (BIP Gramedia) tahun 2010
2. Seri Komik Islami Rumah Ajaib (Elexmedia Komputindo) Tahun 2012
3. Siri Cerita Berirama (PTS Malaysia) tahun 2014
4. Seri Cergam Komilag (Diretorat Paud Kemdikbud) Tahun 2017
5. Seri Aku Anak Cerdas (BIP Gramedia) tahun 2018
6. Seri Cergam Tangguh Bencana (Diretorat Paud Kemdikbud) Tahun 2019
7. Seri Cerita Berima (BIP Gramedia) tahun 2019
8. Seri Aktivitas Seru (BIP Gramedia) tahun 2020
9. Buku Matematika kelas 1 dan 2 (Pusbuk) tahun 2022
10. Buku Tata Busana kelas 10 (Pusbuk) tahun 2022
11. Buku Akutansi dan Kelembagaan (Pusbuk) tahun 2022
12. Buku Pancasila kelas 6 (Pusbuk) tahun 2023
13. Buku Pegangan Guru Seni Teater kelas 1 (Pusbuk) tahun 2023



Editor | Khofifa Najma Iftitah, S.Pd.

: Khofifa.najma.iftitah@brin.go.id

: Pusat Riset Pendidikan, BRIN

: Jl. Gatot Subroto No.10, Kuningan Bar., Kec. Mampang Prpt., Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta

: Teknologi Pendidikan, Kurikulum Sekolah dan Pelatihan,

Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Peneliti, Pusat Riset Pendidikan BRIN (2022 s.d. sekarang)
2. Peneliti, Pusat Perbukuan, Kemendikbudristek (2021)
3. Pengembang Perbukuan, Pusat Kurikulum dan Perbukuan, Kemendikbud (2018 s.d 2020)

Riwayat Pendidikan

S1 Teknologi Pendidikan, Universitas Negeri Malang (2017)

Pengalaman Editor Buku dalam (5 Tahun Terakhir)

1. Buku Sosiologi SMA Kelas XI (Kemendikbudristek, 2022)
2. Panduan Guru Pengembangan Keterampilan Komunikasi dan Interaksi Sosial bagi Peserta Didik dengan Hambatan Intelektual (Kemendikbudristek, 2023)
3. Modul Pendidikan Jarak Jauh Perubahan Iklim, Unit 1 Perubahan Iklim (Direktorat Pendidikan Khusus dan Pendidikan Layanan Khusus, Kementerian Pendidikan Dasar dan Menengah, 2025)
4. Modul Pendidikan Jarak Jauh Perubahan Iklim, Unit 2 Karyaku untuk Lingkungan (Direktorat Pendidikan Khusus dan Pendidikan Layanan Khusus, Kementerian Pendidikan Dasar dan Menengah, 2025)

Informasi Lain dari Editor:

1. Author ID Google Scholar: https://scholar.google.com/citations?user=aq7_950AAAAJ&hl=id
2. Author ID Scopus: 57947181300
3. Author ID Sinta: 6852436
4. Author ID GARUDA: 5433358
5. Author ID Orcid: https://orcid.org/0000-0003-3067-0921

Editor | Erlina Indarti



: erlina.indarti@gmail.com

: Pusat Perbukuan Kemendikbudristek

: Pusat Perbukuan
Kompleks Kemdikbudristek Jalan RS.
Fatmawati, Cipete, Jakarta Selatan

: Pengembang Perbukuan

Riwayat pekerjaan:

1. Editor di Pusat Perbukuan (2024)
2. Analis Pengawasan di Pusat Perbukuan (2022-2024)
3. Pengembang Perbukuan di Pusat Kurikulum dan Perbukuan (Tahun 2015-2021); Pusat Perbukuan Kemendikbudristek (Tahun 2021 – 2022)
4. Pengembang Kurikulum di Pusat Kurikulum Kemendikbudristek (Tahun 2005 – 2011), Pusat Kurikulum dan Perbukuan (Tahun 2011-2015)

Riwayat pendidikan terakhir:

S2 Institut Teknologi Bandung, 2013

Pengalaman mengedit buku dan terbitan lainnya:

1. Panduan Guru: Dasar-Dasar Literasi, Matematika, Sains, Teknologi, Rekayasa, dan Seni – Edisi Revisi (2023)
2. Buku Teks Pelajaran Dasar-Dasar Teknik Kimia Industri, Kemdikburistek (2023)
3. Buku Teks Pelajaran Dasar-Dasar Teknik Furnitur, Kemdikburistek (2023)
4. Buku Panduan Guru Pendidikan Khusus bagi Peserta Didik Disabilitas Fisik Disertai Hambatan Intelektual, Kemdikbudristek (2022)
5. Buku Teks Pelajaran Dasar-Dasar Teknik Konstruksi dan Perumahan, Kemdikburistek (2022)
6. Buku Teks Pelajaran Dasar-Dasar Teknik Otomotif, Kemdikburistek (2022)
7. Buku Teks Pelajaran Dasar-Dasar Manajemen Perkantoran dan Layanan Bisnis, Kemdikburistek (2022)



Editor Visual | Ilhamsyah

: ideamaniacs@telkomuniversity.ac.id

: Telkom University

: Jl. Telekomunikasi No. 1,
Bandung Terusan Buahbatu - Bojongsoang,
Sukapura, Kec. Dayeuhkolot,
Kabupaten Bandung, Jawa Barat

: Art Directing, Graphic Design, Illustration

Pekerjaan/Profesi:

1. Dosen Tetap Telkom University, Kampus Pusat, Bandung
2. Konsultan Desain dan Perancangan Media Komunikasi, Informasi dan Edukasi Palang Merah Indonesia 2011-2022.
3. Konsultan Brand Strategy Kultura Creative Studio, Bandung 2020-2021

Riwayat Pendidikan:

Magister Desain (S2), Fakultas Seni Rupa Desain, Institut Teknologi Bandung, 2013.

Rekam Jejak Desain Buku atau Karya Lainnya:

Pengantar Strategi Kreatif Advertising Era Digital, 2021.



Desainer | Kiata Alma Setra

: Kiatayaki2023@gmail.com

: Praktisi

: Depok

: Graphic Design/Layout,
Content Writing & Social Media Specialist

Riwayat Pekerjaan/Profesi (10 Tahun Terakhir):

1. Penata Letak/Desainer (2015 - Sekarang)
2. Tim Penyelaras Akhir Kegrafikaan Buku, Pusat Perbukuan (2020 - Sekarang)
3. Penilai Kegrafikaan Buku Teks Pendamping, Pusat Perbukuan (2023 - Sekarang)
4. Editor Visual Buku Pendidikan Khusus, Buku Teks Utama, dan Buku Non Teks Kurasi, Pusat Perbukuan (2025)
5. Penulis konten dan Spesialis Sosial Media (2015 - Sekarang)

Riwayat Pendidikan Tinggi dan Tahun Belajar:

D3 – Jurusan Penerbitan – Politeknik Negeri Media Kreatif Jakarta (Polimedia)

Judul Buku dan Tahun Terbit (10 Tahun Terakhir)

1. Menulis berbagai buku proyek konstruksi nasional, Kementerian Pekerjaan Umum dan Perumahan Rakyat (2020 - sekarang)
2. Mendesain berbagai Buku Panduan Guru dan Buku Teks Pelajaran di Pusat Kurikulum dan Perbukuan (2015 - sekarang)

Informasi Lain dari Desainer:

Portofolio : linkedin.com/in/kiatayaki/