**BAB III**

**PERANCANGAN DAN ANALISA SISTEM**

Bab ini menjelaskan mengenai tahap analisa game dan rancangan system pada pembuatan game. Tahapan dilakukan mulai dari perencanaan, analisa, penentuan elemen-elemen yang ada didalam game hingga perancangan metode emotion detection yang digunakan, dan rancangan parameter yang akan digunakan didalam game horror 2D skenario Alas Tilas, Jawa Timur.

1. **Perencanaan**

Dalam perencanaan penulis menganalisa hal yang dibutuhkan untuk membangun game, yaitu sebuah game yang dapat digunakan sebagai sarana untuk melestarikan cerita rakyat Indonesia dan game menjadi menarik untuk dimainkan. Game yang akan dibangun dalam tugas akhir ini adalah sebuah game cerita rakyat yang mengambil tema horror Alas Tilas, Jawa Timur. Game yang dibangun menggunakan bahasa pemrograman C# dan Unity sebagai game enginenya.

Berdasarkan judul yang dibuat yaitu fokusnya kepada produk game yang memanfaatkan emosi, maka hasil akhir dari game ini adalah sebuah game dengan judul “Game Horror 2D Skenario Alas Tilas, Jawa Timur” yang dapat menyesuaikan tingkat kesulitan level berdasarkan ekspresi yang terdeteksi pada wajah pemain dengan tema horror dan cerita horror Indonesia. Dengan adanya game horror yang mengambil tema Alas Tilas, Jawa Timur dengan tingkat kesulitan yang dapat beradaptasi dengan pemain, game ini diharapkan mampu menarik minat penikmat game horror di Indonesia.

1. **Analisa Game Horror 2D Skenario Alas Tilas, Jawa Timur**

Analisa game didefinisikan sebagai penguraian dari suatu game yang utuh ke dalam bagian komponennya dengan tujuan untuk mengidentifikasi dan mengevaluasi permasalahan. Analisa bertujuan mendapatkan pemahaman secara keseluruh tentang game yang akan dibuat. Game horror ini diciptakan sebagai media hiburan bagi penikmat game horror dan pemain yang hendak mencoba bermain game dengan genre horror.

1. **Elemen Dalam Game**

Menurut Teresa Dillon terdapat 8 elemen-elemen dasar yang umumnya ada didalam game yaitu:

1. Peraturan Permainan

Peraturan permainan mencakup instruksi atau suatu aturan perintah, peran objek dan karakter dan metode penggunaanya didalam dunia permainan. Peraturan di game ini yaitu setelah mengklik tombol bermain pemain akan ditempatkan posisi awal pada hutan yang berbeda-beda setiap level. Pemain harus menginjak checkpoint untuk melanjutkan ke level selanjutnya.

1. Plot

Plot harus menguraikan dengan rinci aktivitas yang perlu dilakukan oleh pemain dalam game dan menjelaskan tentang tujuan yang harus dicapai. Dalam pembuatan game ini, menceritakan tentang seorang pendaki yang terpisah dari kerumunan pendaki lainnya dan terjebak di tengah hutan yang digambarkan sebagai lokasi Alas Tilas, Jawa Timur. Maka dari itu karakter utama harus berhasil bertahan dari rintangan dan gangguan dari makhluk supernatural serta mencari jalan keluar dari hutan belantara tersebut.

1. Objektifitas

Objektif merupakan sebuah hal yang penting dan biasanya digunakan dalam bentuk misi bagi pemain dan dituntut memecahkan masalah atau misi tersebut, sewaktu-waktu pemain dituntut punya keahlian dan pengetahuan untuk bisa memaninkannya. Pada game ini pemain dituntut untuk berhasil menyelesaikan game yang memiliki 3 level dengan melewati rintangan yang ada sepanjang level dan mencari jalan keluar hingga berhasil meloloskan diri dari hutan yang angker tersebut.

1. Karakter

Karakter disini yaitu karakter utama dan karakter lain-lain. Model permainan game horror ini yaitu single player. Pemain mengendalikan satu karakter yaitu sang pendaki itu sendiri. Karakter yang tidak bisa di kendalikan atau yang umumnya disebut NPC (Non Playable Character) berupa hantu-hantu yang mengganggu pemain selama menyelesaikan tiap rintangan yang diberikan oleh sistem dalam game.

1. Teks, grafik, dan suara

Pada umumnya pada game terdapat elemen grafik, suara, dan narasi. Hal tersebut dapat dimunculkan secara berkombinasi maupun terpisah. Meskipun mayoritas game yang beredar memenuhi seluruh aspek tersebut, kadangkala tidak selalu diperlukan elemen tersebut muncul secara bersamaan selama game dijalankan. Pada game ini unsur narasi terdapat pada banyak hal, salah satunya pada narasi pembuka dan penutup yang menceritakan kronologis karakter game.

1. Tema

Tema game merupakan sesuatu yang pembuat kehendak atau tujuan yang mau disampaikan oleh pembuat. Tema game yang dipilih merupakan game horor dengan mengambil suasana hutan belantara dengan vegetasi Jawa Timur yang disertai dengan hantu yang kerap dibahas didalam mitos hantu Jawa salah satunya yaitu kuntilanak. Tema ini diambil karena penulis hendak mempromosikan game horror dengan tema kearifan lokal.

1. User Interface

User interface merupakan tampilan antarmuka yang mempermudah pemain berinteraksi dengan game. User interface dibuat selain membuat game menjadi lebih menarik, tujuan lain dibuatnya user interface yaitu memudahkan pemain untuk beradaptasi selama sesi permainan dijalankan. Game horror ini memiliki tampilan antarmuka pengguna yaitu dari menu utama, Sinopsis, Play, Cara Bermain, dan Keluar. Sedangkan saat didalam level, terdapat tampilan status pemain, item yang dibawa karakter, dan tombol interaksi pada objek.

1. Animasi

Animasi merupakan hal yang harus ada pada dunia game, khususnya untuk gerakan karakter - karakter yang ada dalam properti dari objek game. Dengan adanya animasi, pemain dimudahkan untuk memahami suatu situasi yang akan terjadi maupun apa yang sedang terjadi saat ini. Game ini memiliki banyak animasi pada karakter utama, objek yang ada dan hantu yang mengganggu sepanjang permainan.

1. **Analisa Kebutuhan**

Analisis kebutuhan digunakan untuk mempermudah menganalisis sebuah aplikasi atau game yang dibutuhkan, ada dua jenis kebutuhan sistem :

1. **Analisa Kebutuhan Hardware**

Spesifikasi hardware yang digunakan dalam pembuatan Game Alas Tilas, Jawa Timur sebagai berikut :

1. Processor : Ryzen 5 3500H
2. Memory : 8 GB
3. VGA : GTX1050 3GB Mobile
4. Camera : Camera HD Laptop
5. Perangkat Lainnya : Keyboard & Mouse

Sedangkan spesifikasi minimum pengguna agar dapat menjalankan game dengan lancar yaitu :

1. CPU : Core I3 7xxx atau lebih tinggi
2. Memory : 8 GB
3. VGA : GT710 atau Radeon HD5500
4. Camera : Minimum support resolusi 320p
5. Perangkat Lainnya : Keyboard & Mouse
6. **Analisa Kebutuhan Software**

Software yang digunakan selama pembuatan Game Alas Tilas, Jawa Timur yaitu :

1. Unity 2D
2. Visual Studio Code
3. Barracuda Unity SDK
4. MoodME Emotion Detector Unity
5. **Perancangan Game**

Pada subbab ini, akan memasuki perancangan game horror Alas Tilas, Jawa Timur. Subbab ini membahas detail-detail penting terkait perancangan game, termasuk konsep, plot, usecase, cara bermain, karakter, dan elemen-elemen lain yang akan disajikan kepada pemain sehingga pengalaman yang didapat menjadi unik dan menantang.

1. Konsep

Konsep yang menjadi cerita dari game menggunakan papan dasar yang diperoleh dari sumber berita misteri[[1]](#footnote-1). Jejak Gunung Lali Jiwo Arjuno, Jawa Timur, sebagai lokasi yang dilaporkan memiliki hal-hal mistis di mana ditemukan insiden hilangnya pendaki dan tim SAR pendaki kemudian hanya ditemukan tulang-benulangnya saja.

Pemain dituntut untuk berhasil keluar dari hutan belantara yang ada pada Gunung Lali Jiwo tersebut. Indikator keberhasilannya yaitu pemain berhasil melewati 3 level yang diberikan. Setiap level yang muncul digenerate secara otomatis oleh sistem dengan menyesuaikan dengan data pencatatan ekspresi wajah pemain serta performa pemain dalam melewati rintangan yang diberikan.

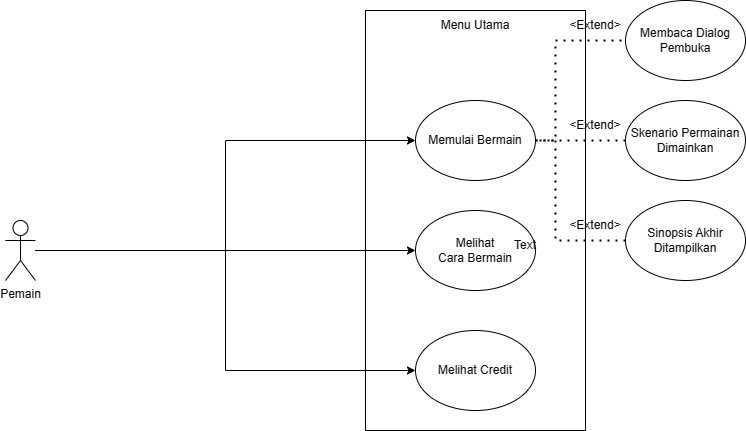
1. Plot Cerita

Plot cerita yang dirancang menceritakan tentang seorang pendaki yang terpisah dari kerumunan pendaki lainnya dan terjebak di tengah hutan yang digambarkan sebagai lokasi Alas Tilas, Jawa Timur. Pendaki mencoba untuk melarikan diri dari hutan belantara yang dilaluinya dengan selamat dengan melalui banyak rintangan dan hantu yang menghuni dan mengganggu manusia yang tersesat di dalam hutan tersebut.

Nahasnya meskipun pendaki merasa berhasil kabur dari hutan belantara yang dipenuhi hantu tersebut, rupanya itu merupakan halusinasi pendaki tersebut yang diakibatkan dari kekurangan darah setelah kecelakaan saat mencari jalan keluar dari hutan tersebut. Game ini mengambil cerita regional agar akrab dengan kehidupan sehari-hari pemain dan diharapkan memberikan efek manifestasi berlebihan pada pemain nantinya.

1. Use *Case* Diagram

Diagram use case adalah alat visual dalam rekayasa perangkat lunak yang berfungsi untuk menggambarkan interaksi antara pengguna dengan sistem yang akan dikembangkan. Dengan use case diagram, pengembang dapat mengidentifikasi fungsi atau tindakan yang akan dilakukan oleh pengguna dalam konteks sistem game. Ini adalah komponen penting analisis sistem yang membantu pengembangan memahami kebutuhan pengguna dan merencanakan fungsionalitas sistem dengan lebih baik. Berikut merupakan usecase diagram game yang dibuat.



Gambar 3.1

Diagram Use CaseGame

* + - 1. Skenario UseCaseMemulai Bermain

Aktor : Pemain

Nama UseCase : Memulai bermain

Deskripsi : Pemain akan memulai bermain dari level 1 hingga game berhasil diselesaikan.

* + - 1. Skenario UseCase Mempelajari Bagaimana Cara Bermain

Aktor : Pemain

Nama Use Case : Mempelajari Cara Bermain

Deskripsi : Memunculkan halaman yang memberi tahu tombol arah gerak dan hal-hal yang dapat dilakukan oleh pemain.

* + - 1. Skenario Use Case Membaca Credits

Aktor : Pemain

Nama Use Case : Membaca Credits

Deskripsi : Memunculkan halaman yang memberi siapa saja yang berkontribusi selama pembuatan game ini.

* + - 1. Skenario Use Case Membaca Latar Belakang Karakter Pemain

Aktor : Pemain

Nama Use Case : Membaca Dialog Pembuka

Deskripsi : Menampilkan dialog atau narasi pembuka yang memberikan latar belakang cerita sekaligus berfungsi sebagai tutorial kepada pemain.

* + - 1. Skenario Use Case Permainan Dimainkan

Aktor : Pemain

Nama Use Case : Skenario Permainan Dimainkan

Deskripsi : Pemain berinteraksi dengan permainan setelah membaca dialog pembuka dan memulai permainan.

* + - 1. Skenario Use Case Penampilan Sinopsis Akhir Karakter Pemain

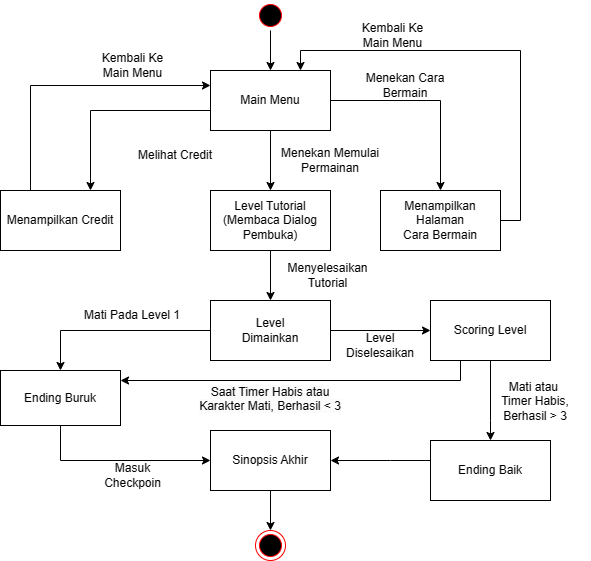
Aktor : Pemain

Nama Use Case : Sinopsis Akhir Ditampilkan

Deskripsi : Menampilkan sinopsis akhir atau ringkasan cerita karakter setelah pemain menyelesaikan permainan.

1. StateDiagram

State diagram merupakan diagram yang digunakan untuk menggambarkan aliran atau perpindahan keadaan dari suatu sistem. Diagram ini terdiri dari serangkaian keadaan, serta transisi antara keadaan yang digambarkan dengan panah. Dengan menggunakan state diagram, pengembang dapat dengan jelas memahami alur kerja sistem, memperkirakan kemungkinan keadaan yang terjadi, dan merancang logika yang sesuai. Berikut merupakan state diagram game yang dibuat.



Gambar 3.2

Diagram StateGame

Diagram diatas menggambarkan alur interaksi pengguna dengan permainan. Setiap kotak mewakili sebuah "state" atau kondisi dalam aplikasi/permainan, dan panah menunjukkan transisi antara state tersebut berdasarkan tindakan pengguna. Penjelasan setiap state akan dijabarkan dibawah ini :

1. Main Menu

Ini adalah titik awal ketika pengguna memulai game. Dari sini, pengguna dapat memilih untuk memulai permainan, melihat cara bermain, atau melihat credit dari aplikasi/permainan.

1. Menampilkan Halaman Cara Bermain

Jika pengguna memilih untuk melihat cara bermain, game akan menampilkan halaman yang menjelaskan aturan dan mekanisme permainan.

1. Menampilkan Credit

Jika pengguna memilih untuk melihat credit, game akan menampilkan informasi tentang siapa saja yang terlibat dalam pembuatan game.

1. Level Tutorial (Membaca Dialog Pembuka)

Jika pengguna memilih untuk memulai permainan, mereka akan diarahkan ke level tutorial di mana mereka dapat membaca dialog pembuka yang menjelaskan cerita dari karakter pemain.

1. Level Dimainkan

Setelah melewati tutorial, pengguna akan memasuki fase di mana mereka mulai bermain level yang sebenarnya.

1. Scoring Level

Setelah menyelesaikan level, game akan menampilkan skor atau penilaian berdasarkan performa pengguna selama bermain.

1. Ending Buruk

Ini adalah layar atau adegan yang ditampilkan jika pemain mati pada level 1 atau pada saat timer habis, jumlah level yang diselesaikan pemain dibawah 3.

1. Ending Baik

Jika pengguna berhasil menyelesaikan 3 atau lebih level permainan pada waktu habis atau mati pada level 3 atau lebih. Pemain akan ditampilkan adegan karakter berhasil kabur dari hutan.

1. Sinopsis Akhir

Jika pengguna berhasil menyelesaikan level atau seluruh permainan, mereka akan disajikan dengan sinopsis atau rangkuman dari cerita atau hasil dari permainan.

1. **Cara Bermain**

Game ini dirancang untuk dimainkan pada platform windows dengan memanfaatkan perangkat keyboard. Tombol-tombol pada keyboard digunakan untuk menggerakan karakter ke kanan dan ke kiri, loncat, dan menunduk. Berikut merupakan penjelasan tombol-tombol yang digunakan sebagai kontrol pemain berserta dengan penjelasannya :

* Tombol (A,S,D)

Tombol A digunakan untuk menggerakkan karakter ke kiri sedangkan Tombol D digunakan untuk menggerakkan karakter ke kanan. Tombol S digunakan untuk membuat karakter bergerak secara menunduk atau merangkak.

Kombinasi tombol ini digunakan dikarenakan agar tangan kiri fokus terhadap pergerakan karakter. Tombol yang dipilih juga dekat dengan tombol kontrol Shift, E, dan Space. Dengan pengaturan tombol sedemikian rupa pemain dapat melakukan pergerakan dalam game secara leluasa dan lebih lincah. Selain itu game serupa juga menerapkan kombinasi tombol ini sehingga pemain tidak kesulitan untuk beradaptasi dengan kontrol pada game ini.

* Tombol (Shift)

Tombol Shift digunakan untuk membuat karakter bergerak lebih cepat dengan cara berlari. Tombol Shift dipilih karena posisi Tombol Shift yang strategis dengan posisi jari kelingking tangan kiri, sehingga diperoleh ergonomic yang baik. Selain itu pada game serupa Tombol Shift juga digunakan secara *default* sebagai Tombol untuk membuat karakter berlari.

* Tombol (Space)

Tombol Space digunakan untuk membuat karakter melakukan aksi loncat. Tombol untuk loncat ini krusial dalam game karena digunakan agar pemain dapat bermanuver melewati rintangan dan jalan yang tidak rata. Tombol Space dipilih karena posisi Tombol Space yang dekat dengan jari jempol tangan kiri, sehingga tidak mengganggu jari-jari tangan kiri. Selain itu pada game serupa juga menerapkan Tombol Space secara *default* sebagai Tombol untuk membuat karakter melompat.

* Tombol (E)

Tombol E digunakan untuk mengambil item yang telah disebar di sepanjang map. Pemilihan tombol ini merujuk pada game serupa yang menerapkan kontrol Tombol E sebagai Tombol untuk mengambil item maupun melakukan aksi tertentu didalam game.

* Tombol (O)

Tombol O digunakan untuk melakukan aksi memukul oleh karakter. Pemilihan tombol ini merujuk pada game serupa yang menerapkan kontrol action untuk jari kanan sebagai Tombol untuk menggunakan melakukan aksi menghancurkan rintangan didalam game.

* Tombol (1,2, dan 3)

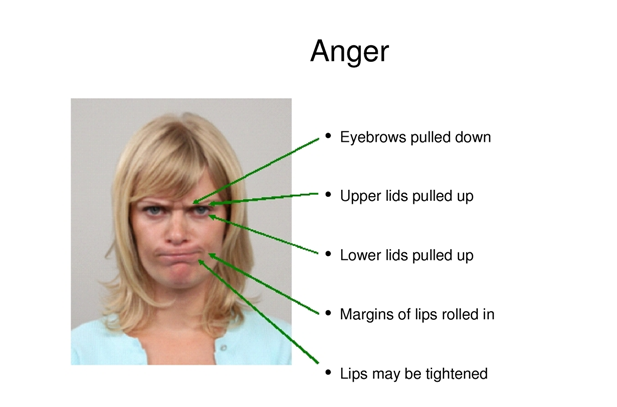
Tombol 1,2, dan 3 digunakan untuk mengkonsumsi atau menggunakan item yang telah dikumpulkan oleh karakter. Pemilihan tombol ini merujuk pada game serupa yang menerapkan kontrol Tombol 1, 2, dan 3 sebagai Tombol untuk menggunakan item didalam game.

1. **Klasifikasi Emosi**

Paul Ekman adalah seorang psikolog yang terkenal dengan penelitiannya tentang ekspresi wajah dan emosi. Menurut teori Ekman, ada enam emosi dasar yang diakui secara universal, yaitu: kebahagiaan, kesedihan, kejutan, rasa jijik, kemarahan, dan ketakutan[[2]](#footnote-2). Ekman dan rekan-rekannya melakukan serangkaian penelitian yang menunjukkan bahwa ekspresi wajah untuk emosi dasar ini konsisten lintas budaya[[3]](#footnote-3).

Berikut adalah deskripsi singkat tentang emosi manusia marah, takut, dan netral (biasa saja) berdasarkan penelitian Ekman:

1. Marah (Anger): Emosi marah ditandai dengan kontraksi otot-otot tertentu di wajah, seperti mengernyitkan dahi, menatap tajam, mengecilkan mata, dan kadang-kadang menunjukkan gigi atau mengetatkan mulut. Ekman menemukan bahwa ekspresi kemarahan memiliki fungsi adaptif, seperti mempersiapkan individu untuk bertindak melawan ancaman atau ketidakadilan.



**Gambar 3.3**

**Karakteristik Wajah Marah**

1. Takut (Fear): Ketakutan ditandai dengan ekspresi wajah yang melibatkan mengangkat alis dan memperlebar mata, seringkali untuk meningkatkan persepsi visual dan mempersiapkan tubuh untuk bereaksi terhadap potensi bahaya. Bibir bisa tertarik ke belakang atau sedikit terbuka sebagai bagian dari respons ini.



**Gambar 3.4**

**Karakteristik Wajah Takut**

1. Netral (Neutral): Ekspresi netral bukanlah emosi khusus, tetapi lebih merupakan keadaan wajah tanpa aktivitas otot wajah yang signifikan. Wajah netral sering digunakan sebagai titik referensi dalam penelitian untuk membandingkan dengan ekspresi emosi yang lebih aktif.



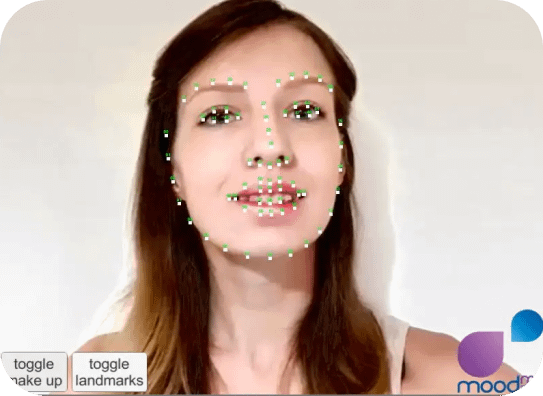
**Gambar 3.5**

**Ilustrasi Wajah Netral**

Dengan memahami ekspresi wajah untuk emosi manusia yang telah diteliti oleh Paul Ekman, penulis dapat mengintegrasikan teknologi deteksi emosi MoodMe ke dalam pengalaman bermain game.

1. **Pendeteksian Emosi MoodMe**

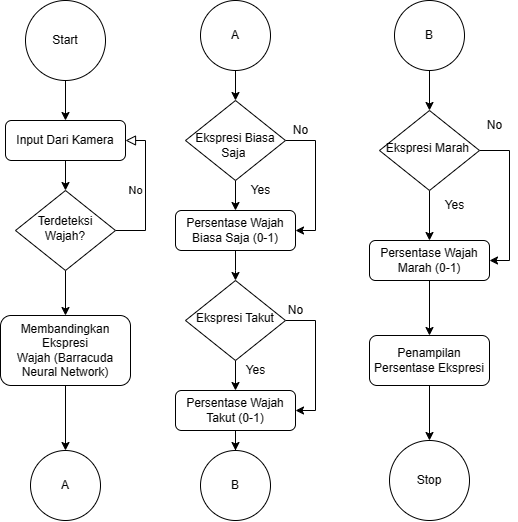
Pada game ini emosi yang dideteksi ada 3 yaitu biasa saja atau netral, marah, dan takut. Parameter ekspresi diambil hanya itu saja dikarenakan keterbatasan pada Library MoodMe yang digunakan, yang hanya mendeteksi emosi Marah, Takut, Jijik, dan Netral.



Gambar 3.6

Ilustrasi Penangkapan Wajah Oleh MoodMe

MoodMe mengklasifikaskan emosi yang terdeteksi dari data Neural Network Barracuda. Data yang sudah ada pada Neural Network Barracuda diambil oleh MoodMe dan diklasifikasikan berdasarkan label-label yang sudah dibuat oleh tim pengembang MoodMe.



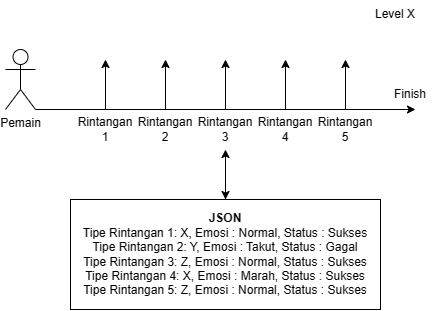
Gambar 3.7

**Alur Kerja Pendeteksian Emosi MoodMe**

MoodMe mengambil tampilan dari standard Webcam Texture Unity class yang mensupport format RGB, RGBA, BGRA, YUV, YUY2. Setelah tampilan didapatkan maka akan dibandingkan dengan label-label yang sudah dibuat oleh MoodME. Hasilnya berupa indicator emosi ber-range 0-100.

1. **Pengaturan Parameter DDA Rintangan Berdasarkan Pendeteksian Emosi MoodMe**

Cara meningkatkan pengalaman bermain salah satunya melalui penyesuaian tingkat kesulitan berdasarkan data pemain[[4]](#footnote-4). Didalam game terdapat beberapa variabel dinamis yang disiapkan oleh penulis. Variabel ini disiapkan untuk diubah oleh penulis berdasarkan data emosi yang sudah ditangkap menggunakan MoodMe. Variabel diolah untuk meningkatkan pengalaman bermain pemain game.



Gambar 3.8

Ilustrasi Pengolahan Parameter Menggunakan MoodME

Pengolahan parameter dilakukan oleh DDA dengan sistem *dynamic scripting*. Konsep *weight clipping* diterapkan pada sistem *dynamic scripting* dengan menentukan optimalisasi nilai tertinggi dari suatu variable yang dapat dicapai. Nilai maksimum yang tinggi memungkinkan bobot meningkat menjadi nilai yang tinggi, sehingga nilai yang paling efektif hampir selalu dipilih. Demikian pula, nilai rendah untuk nilai maksimum menghambat pertumbuhan bobot. Ini menciptakan variasi besar dalam skrip yang dihasilkan, beberapa diantaranya mungkin ada yang tidak optimal.

Dengan konsep ini game berjalan tidak terlalu mudah bagi pemain yang sudah mahir dan tidak terlalu sulit untuk dimainkan bagi pemain baru. Konsep *weight clipping* juga memudahkan pengembang untuk menentukan tingkat kesulitan dengan memberi pembatasan. Konsep ini diterapkan melalui beberapa parameter yang akan diolah kedalam DDA dan yang akan disesuaikan berdasarkan dari ekspresi wajah yang dideteksi oleh MoodMe. Uraian pengaturan parameter DDA rintangan berdasarkan ekspresi wajah pemain akan dibahas pada subbab di bawah.

1. Parameter DDA Pada Rintangan Hantu

Parameter yang dipakai pada rintangan hantu terdapat 4 variabel parameter yang berpengaruh terhadap jumlah rintangan hantu pada level game. Beberapa variable tersebut adalah akan dijelaskan pada table dibawah ini.

**Tabel 3.1**

**Parameter DDA Pada Rintangan Hantu**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Tipe Data** | **Keterangan** |
| jumlahJalanHantu | int | Jumlah blok spawn hantu. |
| ctr\_dikejar\_hantu | int | Counter pemain dikejar hantu. |
| emosi | int | Poin emosi yang terdeteksi. |
| status | bool | Cek kegagalan pemain. |

Setiap pemain mengaktifkan trigger gameObject rintangan, maka secara otomatis akan memanggil function pencatat log. Semua variabel yang telah ditentukan dicatat dalam bentuk log berupa JSON. Setelah level selesai, log yang tercatat akan dibaca oleh pembentuk level.

Value parameter yang akan digunakan pada level berikutnya disesuaikan berdasarkan dengan patokan keputusan DDA. Dengan menerapkan konsep Weight Clipping pada DDA spawn hantu, diterapkan value minimum dan maksimum kemunculan blok spawn hantu. Penjelasan keputusan DDA untuk spawn hantu diuraikan dalam table dibawah.

**Tabel 3.2**

**Tabel Keputusan DDA Rintangan Spawn Hantu**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Parameter** | | | | **DDA** | | |
| **JalanHantu** | **Dikejar** | **Status** | **Emosi** | **Skor** | **Label** | **Output** |
| 4 | 2 | 1,0 | 1+2/dikejar = 1,5 (Marah) | Normal  (1 sukses, 1 gagal) | Normal | Jumlah = Rand (Weight.Clip Normal) |

Kemudian level dibuat secara otomatis dengan mengadopsi keputusan dari parameter yang telah diolah DDA. Level dibentuk oleh pembentuk level sesuai dengan suplay data baru yang diolah DDA. Pengolahan dilakukan untuk meningkatkan kualitas pengalaman bermain pemain, sehingga terasa tidak terlalu mudah dan tidak terlalu mudah yang dibuktikan melalui kuesioner.

1. Parameter DDA Pada Rintangan Duri

Parameter yang dipakai pada rintangan duri terdapat 3 variabel parameter yang berpengaruh terhadap jumlah rintangan duri pada level game. Beberapa variabel tersebut adalah akan dijelaskan pada table dibawah ini.

**Tabel 3.3**

**Parameter DDA Pada Rintangan Duri**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Tipe Data** | **Keterangan** |
| jumlahJalanJebakan | int | Jumlah blok rintangan duri. |
| emosi | int | Poin emosi yang terdeteksi. |
| status | int | Cek kegagalan pemain. |

Setiap pemain mengaktifkan trigger gameObject rintangan, maka secara otomatis akan memanggil function pencatat log. Semua variabel yang telah ditentukan dicatat dalam bentuk log berupa JSON. Setelah level selesai, log yang tercatat akan dibaca oleh pembentuk level.

Value parameter yang akan digunakan pada level berikutnya disesuaikan berdasarkan dengan patokan keputusan DDA. Dengan menerapkan konsep Weight Clipping pada DDA rintangan duri, diterapkan value minimum dan maksimum kemunculan rintangan duri. Penjelasan keputusan DDA untuk rintangan duri diuraikan dalam table dibawah

**Tabel 3.4**

**Tabel Keputusan DDA Rintangan Duri**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | | | **DDA** | | |
| **JalanJebakan** | **Status** | **Emosi** | **Skor** | **Label** | **Output** |
| 4 | 0,0,1,1 | 2+2+2+2/  jalanJebakan= 2 (Netral) | Normal (2 kena) | Mahir (Normal & Netral) | Jumlah = Rand (Weight.Clip Mahir) |

Kemudian level dibuat secara otomatis dengan mengadopsi keputusan dari parameter yang telah diolah DDA. Level dibentuk oleh pembentuk level dengan suplay data baru yang telah diolah DDA. Jumlah rintangan duri yang muncul pada level berikutnya telah disesuaikan dengan keputusan DDA.

1. Parameter DDA Pada Tempat Bersembunyi

Parameter yang dipakai pada tempat bersembunyi terdapat 3 variabel parameter yang berpengaruh terhadap jumlah tempat bersembunyi pada level game. Beberapa variable tersebut adalah akan dijelaskan pada table dibawah ini.

**Tabel 3.5**

**Parameter DDA Pada Tempat Bersembunyi**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Tipe Data** | **Keterangan** |
| jumlahJalanBambu | int | Jumlah blok bambu. |
| masuk | int | Hitung jumlah pemain bersembunyi. |
| emosi | int | Poin yang diberikan kepada emosi yang terdeteksi |

Setiap pemain sedang dikejar hantu kemudian masuk dalam area trigger gameObject bambu dan durasi dikejar sudah habis, maka secara otomatis akan memanggil function pencatat log Semua variabel yang telah ditentukan dicatat dalam bentuk log berupa JSON. Setelah level selesai, log yang tercatat akan dibaca oleh pembentuk level.

Dengan menerapkan konsep Weight Clipping pada DDA tempat bersembunyi, diterapkan value minimum dan maksimum kemunculan tempat bersembunyi. Penjelasan keputusan DDA untuk tempat bersembunyi diuraikan dalam table dibawah

**Tabel 3.6**

**Tabel Keputusan DDA Tempat Bersembunyi**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | | | **DDA** | | |
| **JalanBambu** | **Masuk** | **Emosi** | **Skor** | **Label** | **Output** |
| 4 | 1 | 0 | Pemula (1) | Pemula (Pemula & Takut) | Jumlah = Rand (Weight.Clip Pemula) |

Kemudian level dibuat secara otomatis dengan mengadopsi keputusan dari parameter yang telah diolah DDA. Level dibentuk oleh pembentuk level dengan suplay data baru yang telah diolah DDA. Jumlah tempat bersembunyi yang muncul pada level berikutnya telah disesuaikan dengan keputusan DDA.

1. Parameter DDA Pada Jumpscare Penampakan

Parameter yang dipakai pada jumpscare penampakan terdapat 3 variable parameter yang berpengaruh terhadap jumpscare penampakan yang muncul pada level game. Beberapa variable tersebut adalah akan dijelaskan pada table dibawah.

**Tabel 3.7**

**Parameter DDA Pada Jumpscare Penampakan**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Tipe Data** | **Keterangan** |
| jumlahJalanJumpscare | int | Jumlah blok jumpscare. |
| status | int | Status pemain menginjak kuburan. |

**Tabel 3.7**

**(Lanjutan)**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Tipe Data** | **Keterangan** |
| emosi | int | Poin yang diberikan kepada emosi yang terdeteksi. |

Setiap pemain mengaktifkan trigger gameObject rintangan secara otomatis akan memanggil function pencatat log. Semua variabel yang telah ditentukan dicatat dalam bentuk log berupa JSON. Setelah level selesai, log yang tercatat akan dibaca oleh pembentuk level.

Value parameter yang akan digunakan pada level berikutnya disesuaikan berdasarkan dengan patokan keputusan DDA. Dengan menerapkan konsep Weight Clipping pada DDA jumpscare penampakan, diterapkan value minimum dan maksimum kemunculan jumlah jumpscare penampakan. Penjelasan keputusan DDA untuk jumpscare penampakan diuraikan dalam table dibawah

**Tabel 3.8**

**Tabel Keputusan DDA Jumpscare Penampakan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | | | **DDA** | | |
| **JalanJumpscare** | **Status** | **Emosi** | **Skor** | **Label** | **Output** |
| 6 | 3 | 1/status=  0,16 (Takut) | Normal (3) | Pemula (Normal & Takut) | Jumlah = Rand (Weight.Clip Pemula) |

Kemudian level dibuat secara otomatis dengan mengadopsi keputusan dari parameter yang telah diolah DDA. Level dibentuk oleh pembentuk level dengan suplay data baru yang telah diolah DDA. Jumlah blok jumpscare penampakan yang muncul pada level berikutnya telah disesuaikan dengan keputusan DDA.

1. Ihwal.Id, “*Mengungkap Misteri Gunung Arjuna, 3 Penyebab Pendaki Mudah Tersesat dan Hilang di Alas Lali Jiwa*”, (https://www.ihwal.id/info/68211238086/mengungkap-misteri-gunung-arjuna-3-penyebab-pendaki-mudah-tersesat-dan-hilang-di-alas-lali-jiwa/, Diakses pada 20 September 2022) [↑](#footnote-ref-1)
2. Ekman, P. *Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life*. Times Books. (2003). [↑](#footnote-ref-2)
3. Ekman, P., & Friesen, W. V. *Constants across cultures in the face and emotion*. Journal of Personality and Social Psychology. Vol 17(2), (1971), hlm 124–129. [↑](#footnote-ref-3)
4. Constant, T., & Levieux, G, *Dynamic difficulty adjustment impact on players' confidence*, In Proceedings of the 2019 CHI conference on human factors in computing systems, 2019, hlm 1-12. [↑](#footnote-ref-4)