

BAB III

ANALISA DAN PERANCANGAN

III.1. Analisis Sistem

Analisis sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka.

Analisis sistem juga banyak orang yang mendefinisikan sebagai bagaimana memahami dan menspesifikasikan dengan detail apa yang harus dilakukan oleh sistem. Analisis sistem merupakan tahapan paling awal dari pengembangan sistem yang menjadi fondasi untuk menentukan keberhasilan sistem yang dihasilkan nantinya.

Adapun perancangan *game* lari halang rintang ini bertujuan sebagai sarana hiburan. Oleh sebab itu penulis merancang permainan yang sederhana namun tetap memberikan tantangan kepada pemain.

Kendala-kendala yang dihadapi pada sistem yang sedang berjalan yaitu:

1. *Game* yang dibuat dapat berjalan pada sistem operasi *Windows* dengan arsitektur x86 (32 bit).
2. Objek-objek yang digunakan dalam *game* ini menggunakan *sprite* berbasis 2D.
3. Tingkat kesulitan dalam *game* ini terdiri dari tiga, yaitu :

a. Mudah

Pada tingkat ini jarak antar halangan jauh, kecepatan lari lawan lambat.

b. Menengah

Pada tingkat ini jarak antar halangan sedang, kecepatan lari lawan juga sedang.

c. Sulit

Pada tingkat ini jarak antar halangan dekat, kecepatan lari lawan cepat.

III.2. Analisis Masalah

Dalam merancang sebuah aplikasi tidak luput dari beberapa masalah. Adapun analisa masalah di dalam *game* lari halang rintang ini adalah sebagai berikut:

1. Penentuan Tujuan dan Jenis *Game*

Penentuan tujuan *game* sangat berpengaruh terhadap perancangan *game* itu sendiri. Adapun tujuan dari *game* ini yaitu pemain berusaha berlari secepat mungkin agar dapat mengalahkan lawan untuk mendapatkan posisi juara.

2. Pembuatan Model *Game* dan Implementasi Model ke *Game*

Game ini menggunakan objek 2D berbasis *sprite*, dimana dalam pembuatannya menggunakan *Adobe Photoshop CS4*. Hasil dari pembuatan model tersebut akan di-*convert* ke dalam bentuk *.jpg* dan *.png*. Setelah pemodelan selesai dilakukan, *file-file* tersebut di-*export* ke *Unity Engine* agar model yang dibuat dapat dibaca oleh *Unity Engine*.

3. Penulisan *Script* Program dan Penggabungan *Script* ke Objek

Tahap ini ditujukan untuk mengatur setiap objek tertentu, antara lain pembentukan map secara prosedural, mengatur pergerakan karakter utama dan lawan, deteksi antar objek, dan pengaturan suara (*background music* dan *sound effect*).

III.3. Analisis Kebutuhan

Game ini dirancang menggunakan *software Unity Engine* versi 5.0.1, penulisan *listing* program (*scripting*) menggunakan *software MonoDevelop* versi 4.0.1, dan untuk pembuatan model objek menggunakan *Adobe Photoshop CS4*.

Adapun untuk menjalankan *game roguelike* ini menggunakan *smartphone Android* dengan spesifikasi minimum sebagai berikut:

1. *Processor* 1.2 Ghz
2. *Harddisk* 128 MB
3. RAM 512MB
4. Sistem operasi: *Android Jellybean* (4.1)

III.4. Penerapan Algoritma

Algoritma merupakan kumpulan instruksi atau langkah-langkah yang jelas untuk menyelesaikan suatu masalah. Algoritma memegang peranan penting dalam bidang pemrograman sehingga dapat dipahami konsep dasar algoritma program yang lebih efektif dan efisien. Dalam *game* ini terdapat algoritma yang diterapkan adalah Algoritma *fuzzy logic*.

Algoritma *fuzzy logic* diterapkan pada karakter lawan dimana mengatur kecepatan lari lawan. Adapun kecepatan lari lawan ada 3 *state* yaitu cepat (C), sedang (Sd) dan lambat (L), adapun parameter untuk kecepatan lari lawan yaitu :

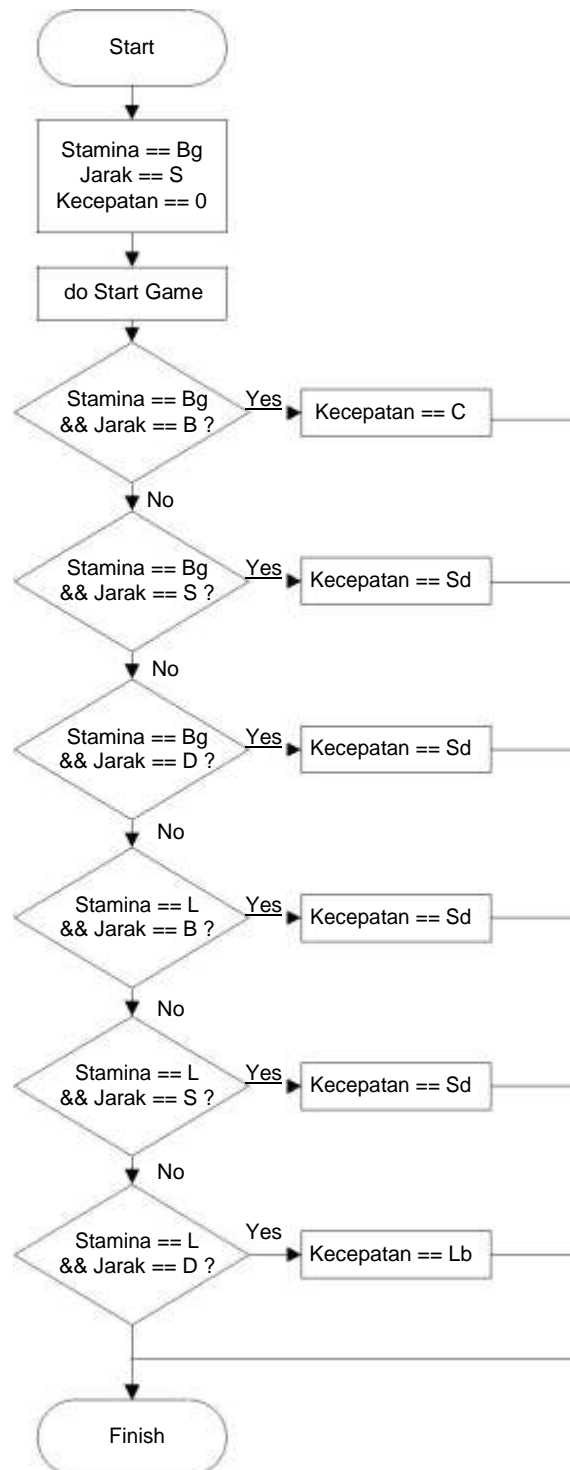
- a. Jarak lawan dengan *player*, terdiri dari 3 *state* :
 - 1) Belakang (B) : Posisi lawan berada di belakang pemain.
 - 2) Depan (D) : Posisi lawan berada di depan pemain.
 - 3) Sejajar (S) : Posisi lawan sejajar dengan pemain.
- b. Stamina, terdiri dari 2 *state* :
 - 1) Bugar (Bg) : Jika stamina lawan bugar kemungkinan kecepatan lawan berada posisi cepat itu lebih besar.
 - 2) Lelah (L) : Jika stamina lawan lelah kemungkinan kecepatan lawan berada posisi lambat itu lebih besar.

Aturan fuzzy untuk kecepatan lawan pada tabel dibawah:

Tabel III.1. Tabel Aturan Fuzzy Logic Untuk Kecepatan pada Karakter Lawan

J A R A K				
	VARIABEL	B	S	D
STAMINA	Bg	C	Sd	Sd
	L	Sd	Sd	Lb

Adapun untuk penjelasan dari algoritma ini dapat dilihat pada *flowchart* pada gambar III.1:



Gambar III.1. Flowchart Fuzzy Logic pada Karakter Lawan

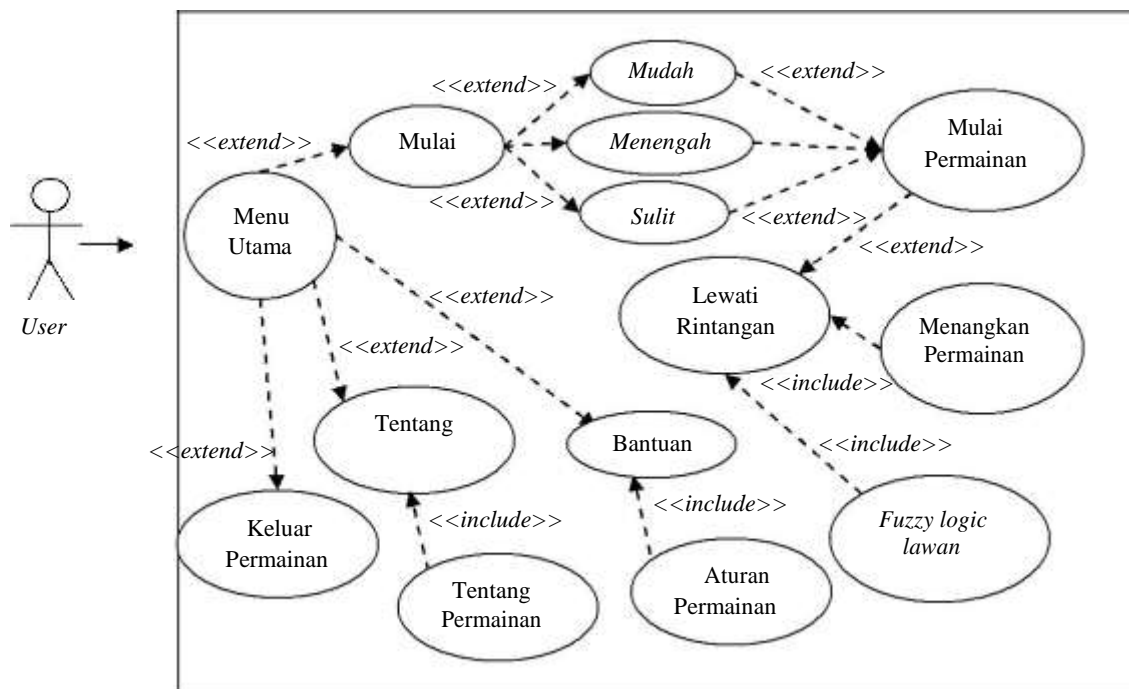
III.5. Desain Sistem

Metode perancangan yang digunakan untuk merancang *game* lari halang rintang ini adalah metode perancangan terstruktur menggunakan UML. UML pada dasarnya merupakan konsep perancangan untuk mengetahui proses dan jalannya aliran data serta penyimpanan data yang merupakan konsep perancangan termudah dengan pendekatan *top down*.

III.5.1. Use Case Diagram

Use case diagram adalah sebuah gambaran dari fungsi sistem yang dipandang dari sudut pandang pemakai. *Actor* adalah segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi (Whitten, 2004: 258).

Berikut perancangan aplikasi yang digambarkan menggunakan *use case diagram*:










Gambar III.2. Use Case Diagram

III.5.2. Storyboard

Storyboard adalah area berseri dari sebuah gambar sketsa yang digunakan sebagai alat perencanaan untuk menunjukkan secara visual bagaimana aksi dari sebuah cerita berlangsung. *Storyboard* sangat diperlukan dalam hal menjelaskan proses-proses yang terjadi pada *game* yang dirancang. Berikut adalah *storyboard* dari *game* lari halang rintang:

Tabel III.2. Storyboard Game Lari Halang Rintang

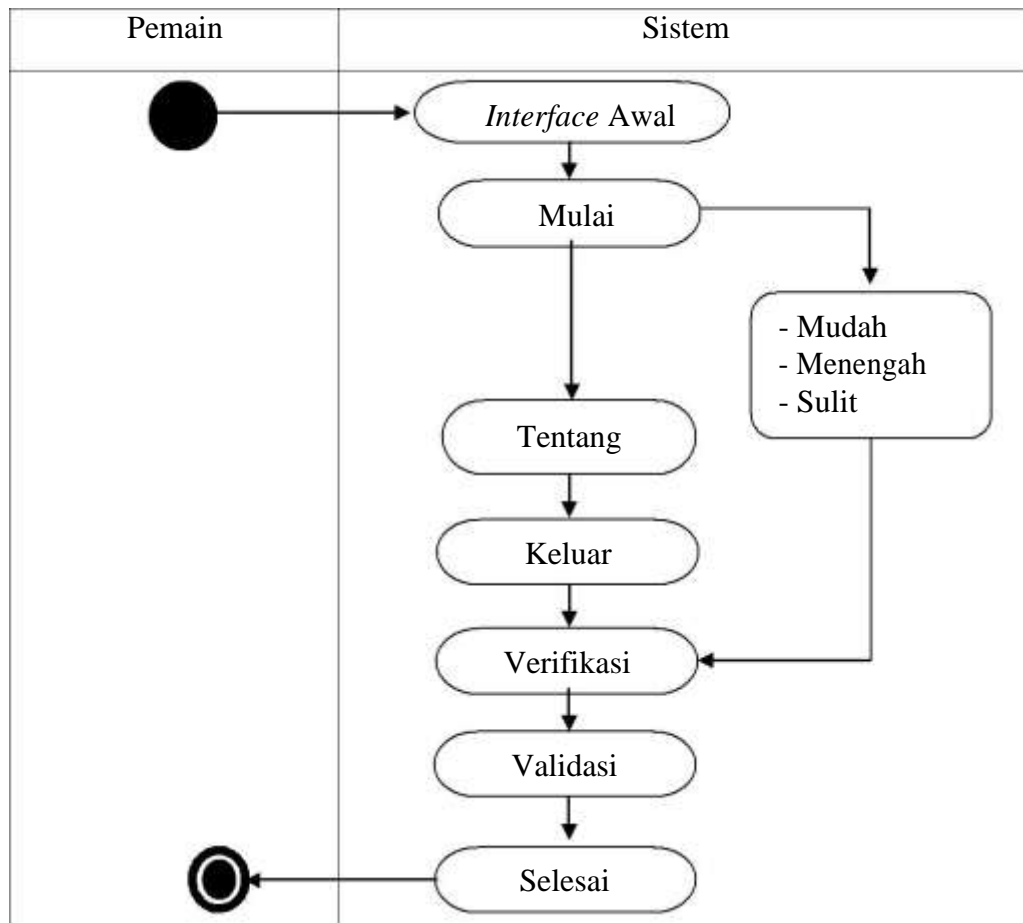
Nama Objek	Gambar	Deskripsi
Karakter utama		Karakter yang digerakkan oleh pemain untuk mengalahkan lawan menuju garis <i>finish</i> dengan melewati rintangan-rintangan.
Karakter lawan		Karakter yang akan bergerak menuju garis <i>finish</i> yang bertujuan mengalahkan karakter utama.
Rintangan		Objek yang terdapat pada <i>game</i> yang berfungsi sebagai rintangan dimana letaknya berurut di sepanjang pertandingan.
Pause		<i>Button pause</i> yang terdapat pada <i>game</i> berfungsi sebagai tombol jeda pada <i>game</i> lari halang rintang ini.
Lintasan		Lintasan yang berfungsi sebagai arena lintasan <i>game</i> .

Start		Tempat posisi awal karakter memulai pertandingan pada <i>game</i> lari halang rintang.
Finish		Tempat posisi akhir yang dituju oleh karakter untuk memenangkan <i>game</i> lari halang rintang.

III.5.3. Activity Diagram

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* merupakan *state diagram* khusus, yang sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu, *activity diagram* tidak menggambarkan perilaku internal sebuah sistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

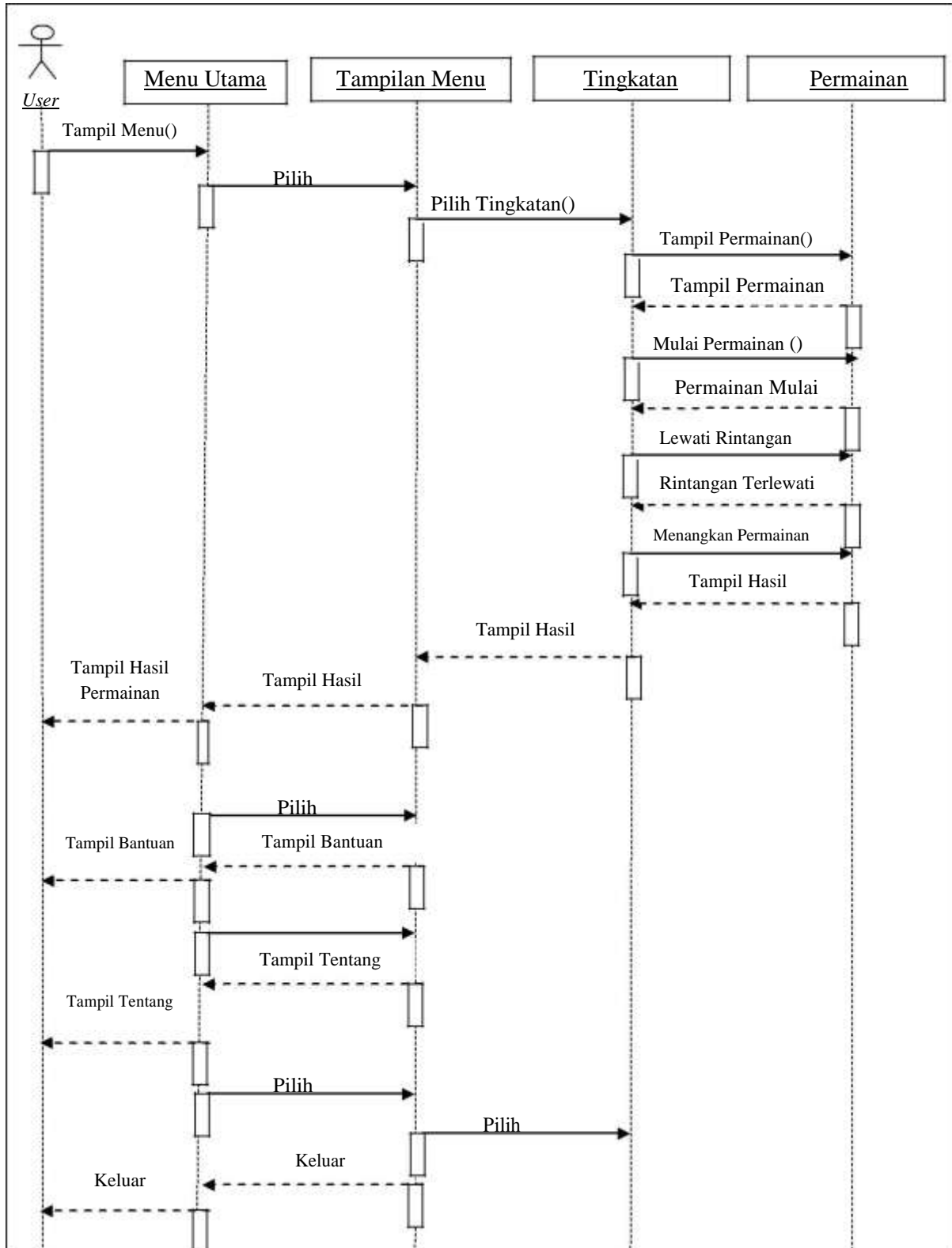
Adapun *activity diagram* pada *game lari halang rintang* ini adalah:



Gambar III.3. Activity Diagram

III.5.4. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* biasa digunakan untuk menggambarkan rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

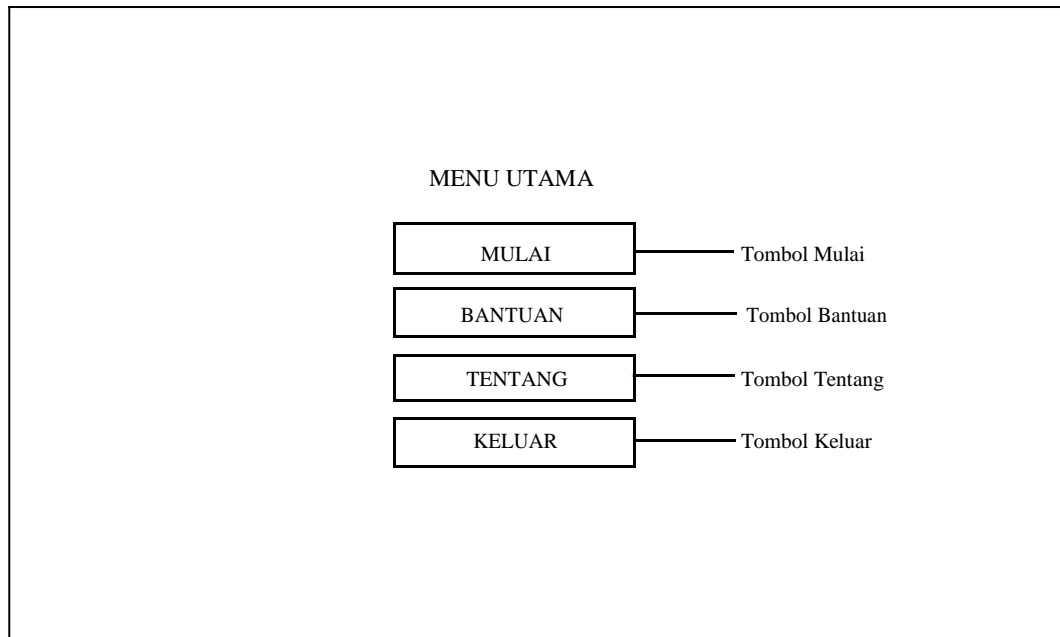


Gambar III.4. Sequence Diagram

III.6. Desain *User Interface*

1. Tampilan Menu Utama

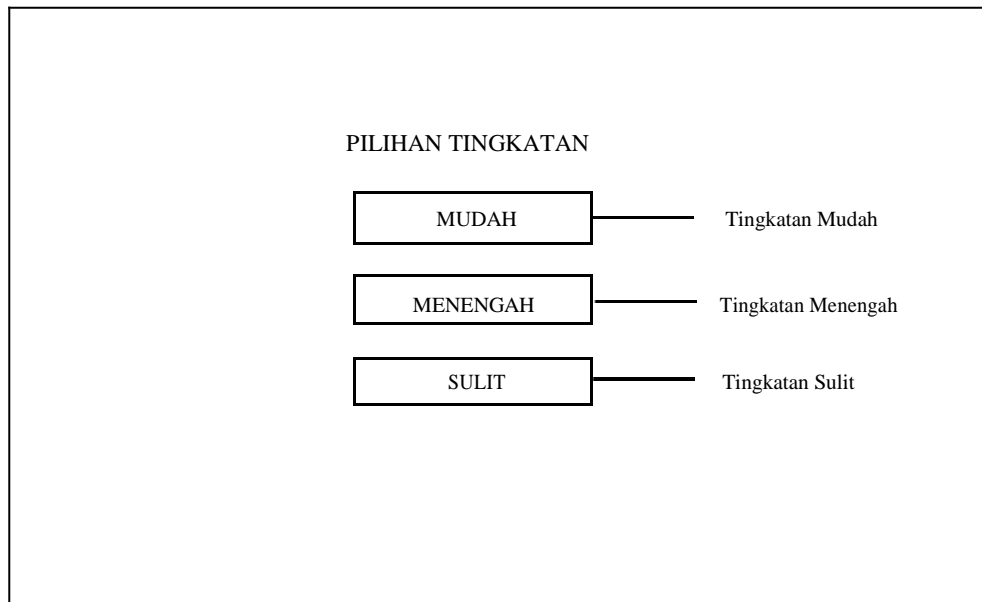
Tampilan awal ketika pemain memulai *game* ini adalah tampilan layar Menu Utama yang menunjukkan tombol Mulai, Bantuan, Tentang, dan Keluar .



Gambar III.5. Tampilan Awal *Game* (Menu Utama)

2. Tampilan Tombol Mulai

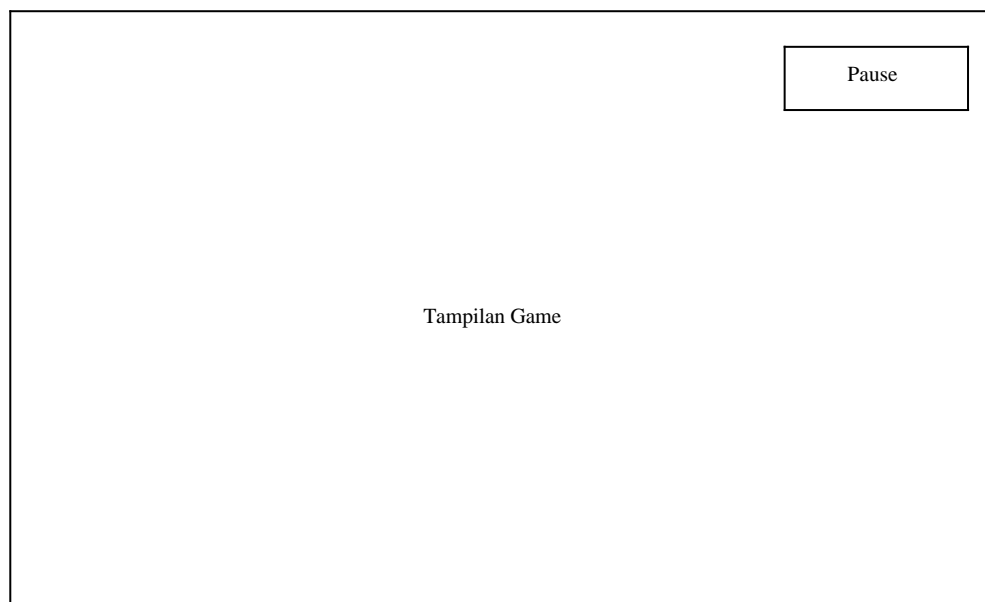
Pada Tampilan Tombol Mulai pada *game* ini terdapat pilihan untuk tingkatan permainan: Mudah, Menengah, dan Sulit.



Gambar III.6. Tampilan Pilihan Tingkatan

3. Tampilan *Game*

Desain tampilan ini pada pembuatan *game* lari halang rintang ini terdapat tampilan *game* dan juga terdapat tombol *Pause* yang berfungsi untuk menjeda permainan.



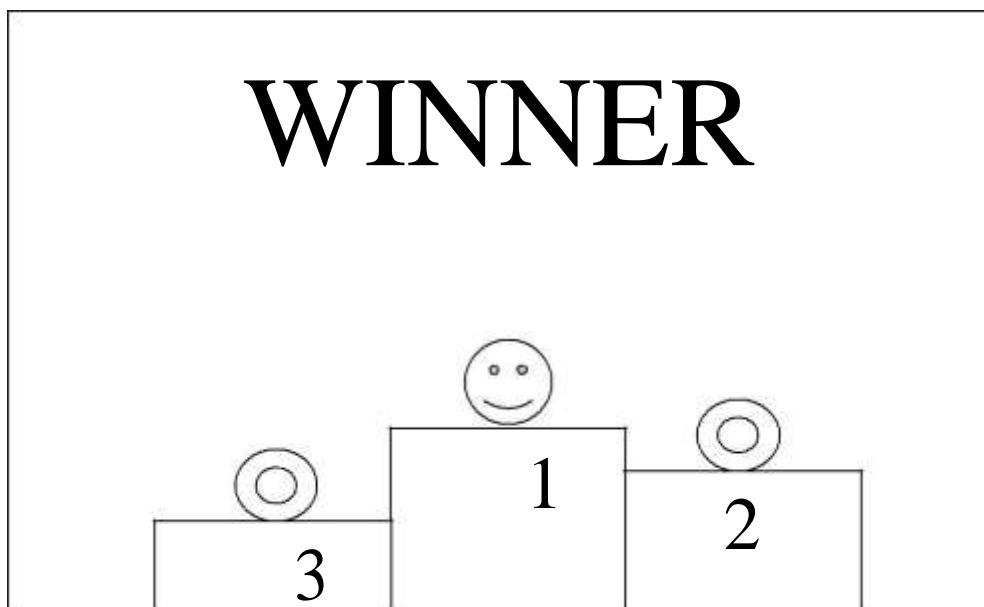
Gambar III.7. Tampilan *Game*

4. Tampilan *Finish*

Pada tampilan *finish* adalah tampilan akhir pertandingan yang terdapat terdapat 3 kondisi, yaitu: Juara 1, Juara 2, Juara 3.

a. Tampilan Juara 1

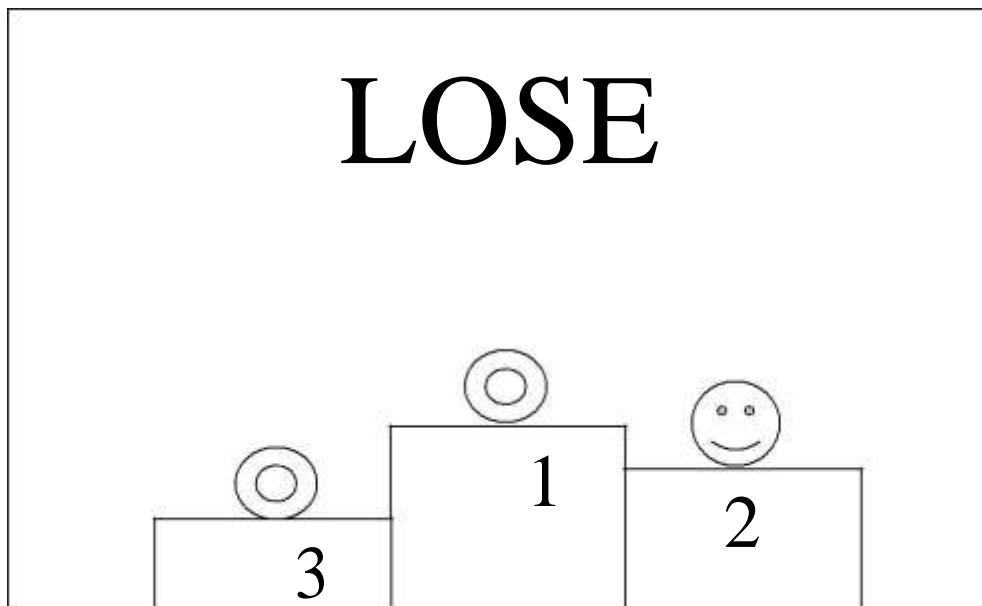
Tampilan ini akan muncul ketika posisi pemain berada pada posisi pertama.



Gambar III.8. Tampilan *Finish* Juara 1

b. Tampilan Juara 2

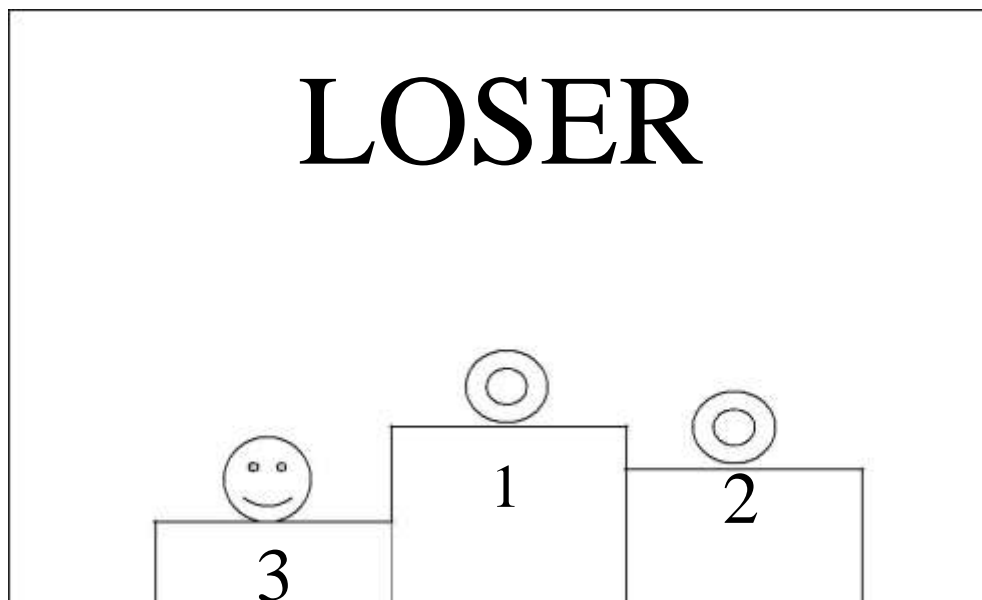
Tampilan ini akan muncul ketika posisi pemain berada pada posisi kedua.



Gambar III.9. Tampilan *Finish Juara 2*

c. Tampilan Juara 3

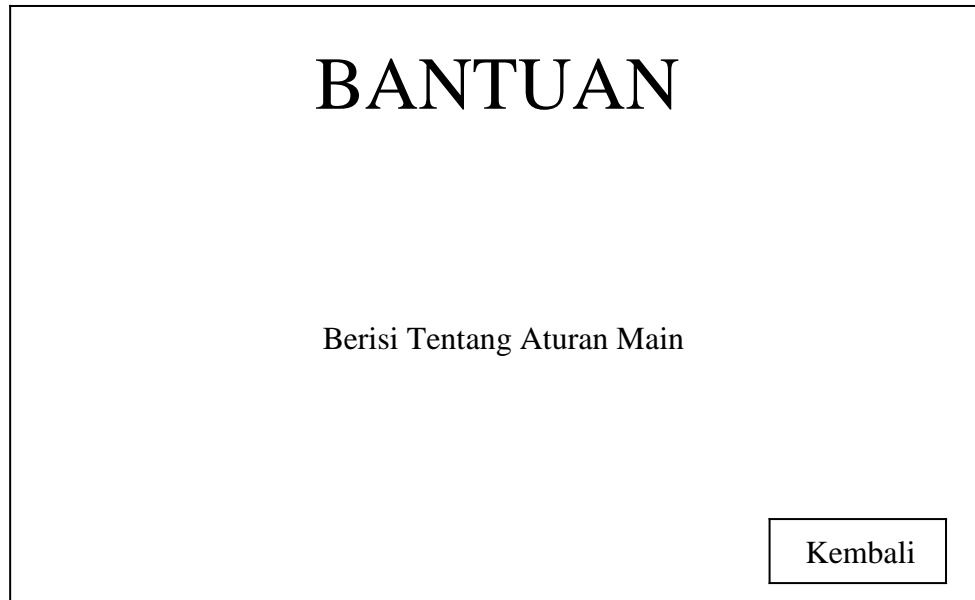
Tampilan ini akan muncul ketika posisi pemain berada pada posisi akhir



Gambar III.10. Tampilan *Finish Juara 3*

5. Tampilan Bantuan

Pada tampilan bantuan berisi petunjuk atau aturan main pada *game* ini.



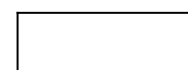
Gambar III.11. Tampilan Menu Bantuan

6. Tampilan Tentang

Pada tampilan bantuan berisi petunjuk cara memainkan atau aturan main pada *game* ini.

TENTANG

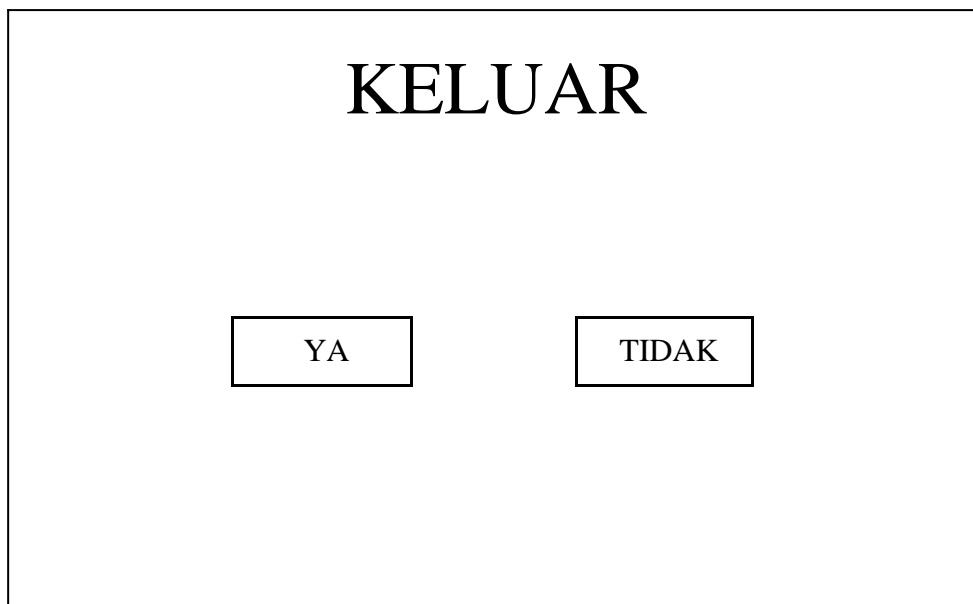
Berisi Tentang Biodata Pembuat



Gambar III.12. Tampilan Menu Bantuan

7. Tampilan Keluar

Pada tampilan keluar terdapat pilihan yang berisikan pesan Ya dan Tidak. Jika pemain memilih Ya berarti keluar dari permainan dan jika tidak kembali ke menu utama.



Gambar III.13. Tampilan Keluar

