BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan dan hambatan-hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikanperbaikan. Pembangunan perangkat lunak terdiri dari beberapa tahap, salah satunya adalah tahap menganalisa sistem. Sistem yang akan dianalisis untuk membangun perangkat lunak *game "Haunted Hospital"* menggunakan metoda *Potential Field*. Analisis sistem yang akan dibahas mengenai cara kerja metode *potential field* yang diimplementasikan pada *game "Haunted Hospital".*

3.1.1 Analisis Masalah

Di Indonesia sendiri sebagian besar *gamer* yang menyenangi *game-game* bernuansa *horror*, selalu di terpa dengan suasana latar belakang tempat maupun budaya luar seperti Resident Evil, Doom, Silent Hill, dan lain-lain. Sedangkan perkembangan industri *game* yang menggunakan karakter-karakter lokal masih terlalu sedikit, dibandingkan dengan peminat *game* dalam negeri yang semakin meningkat dari tahun ke tahun. Oleh sebab itu dibutuhkan berbagai upaya dalam meningkatkan produksi *game* yang dapat memeperkenalkan karakter-karakter dan budaya dalam negeri.

3.1.2 Analisis *Game* Sejenis

Pada analisis *game* sejenis, akan dilakukan observasi terhadap *game* yang memiliki kesan *gameplay* dan *genre* dengan *game* yang akan dibangun. Tujuan dari analisis ini yaitu mendapatkan konsep permainan dan perbandingan dari *game* yang telah ada.

Dalam pembahasan *game* sejenis ini, akan dibahas *game* sejenis yang dilihat dari jenis *game*, maupun dari kesan alur cerita yang diambil. *Game* yang akan menjadi acuan disini adalah *game* Z-Virus.

1. Pengenalan *game* Z-Virus

   Game Z-Virus merupakan *game* berjenis *First Person Shooter* (FPS), yang dibuat oleh Fajrul Falakh seorang *game maker* dari Indonesia. *Game* ini dibuat mengguanakan *Blender Game Engine* (BEG) yang mempunyai *desain* 3D (3 Dimensi).

   Berberapa cuplikan permainan dari *game* Z-Virus dapat dilihat pada gambar 3.1 dan gambar 3.2 :



Gambar 3-1 menu utama *game* Z-Virus

Gambar 3.1 adalah tampilan menu utama dari *game* Z-Virus dimana terdapat pilihan *menu start game* untuk memulai permainan, *training* untuk latiahan menembak, *control* untuk *info controller*, *credit* untuk *info* pengembang, *intro* untuk melihat *splash screen intro*, dan *exit game* untuk keluar dari *game*.



Gambar 3-2 layar permainan *game* Z-Virus

Gambar 3.2 adalah tampilan layar permainan pada *game* Z-Virus, pada layar permainan dapat dilihat status jumlah peluru dan kondisi *hit point* (HP) dan juga terdapat *minimap* yang dapat dilihat pada sudut kanan atas layar permainan.

2. *Gameplay Game* Z-Virus

   Game Z-Virus ini menceritakan seorang prajurit yang harus berjuan membasmi semua makhluk hidup yang telah terjangkit virus Z. *Level* pertama musuh berupa laba-laba yang bergerak menyerang player dengan jarak dekat, pemain harus bisa sampai ke lokasi tujuan misi untuk lanjut ke *level* berikutnya. Pada *level* kedua musuh berupa makhluk bermata satu berwarna hijau yang biasa dikenal dengan nama *Ogre*. Untuk sukses dalam *level* dua ini pemain harus berhasil sampai ke *gate level* 2.

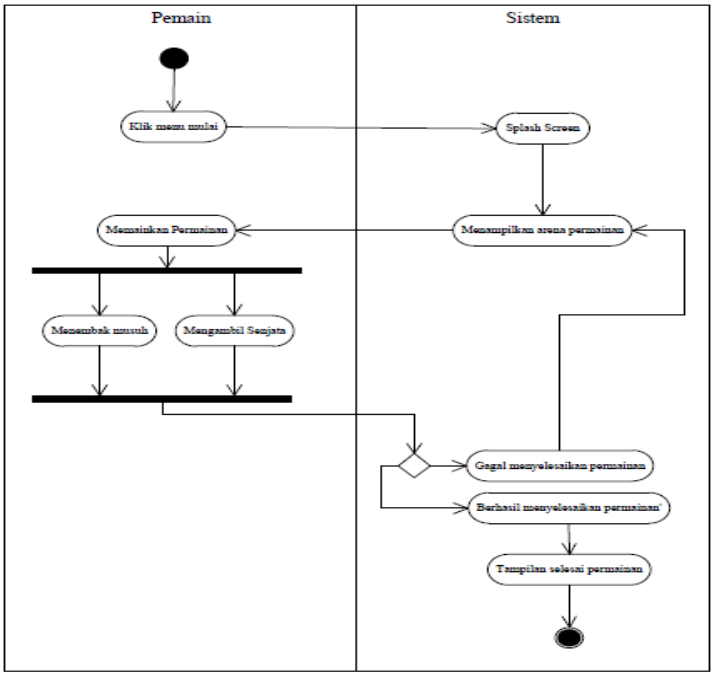
3. Skenario *game* Z-Virus

   Sekenario permainan dalam *game* Z-Virus disusun oleh beberapa alur permainan sebagai berikut :

   a. Pemain berada pada gurun pasir yang sangat luar
   b. Pemain harus menghabisi semua makhluk yang tertular virus Z yang menyerang mereka
   c. Pemain harus mampu bertahan hidup dari serangan makhluk tersebut dan harus dapat mememukan *game* jalan keluar dari setiap level.

d. Jika pemain terkena serangan oleh musuh, maka darah / *Health Poin* akan berkurang dan apabila habis maka pemain diulangi ke awal permainan.

Adapun alur sistem dari permainan *game* Z-Virus dapat di gambarkan melalui activity diagram pada gambar 3.3 :



Gambar 3-3 Alur permainan pada *game* Z-Virus

Game Z-Virus memiliki beberapa keterbatasan, yang didapatkan dari hasil *riview* dari beberapa *game maker* pada *website* tempat *game* ini dipublikasikan, yaitu :

1. Pada *game* ini keterangan tentang jalan cerita, target masing-masing *level* dan tujuan akhir *game* tidak dijelaskan didalam *game*, baik itu berupa tulisan maupun *video*.
2. Pada *game* ini NPC musuh *spawn* (muncul) ditempat yang sama, sehingga *player* dengan mudah dapat membunuh musuh dengan mengarahkan *crossfire* ketempat *spawn* musuh tersebut
3. Pada *game* ini NPC musuh hanya dapat menyerang *player*, tidak mempunyai kemampuan untuk menentukan pilihan berupa menyerang, bertahan atau melarikan diri.

Kelebihan *game* ini adalah :

1. *Game* Z-Virus memiliki *interface* (antarmuka) permainan yang bagus dan nyaman.
2. *Game* Z-Virus memiliki *desain* dan efek 3D yang sudah halus.

3.1.3 Analisis *Game*

Perkembangan indrustri *game* sekarang ini sangat pesat terutama dari segi grafis. *Game* dengan tipe *gameplay action* sudah tidak asing lagi dalam dunia *game*, karena *gameplay*-nya *action* membuat orang-orang tertarik untuk memainkannya, banyak *game* yang memakai sistem *gameplay action* ini. *Action Games* tidak ada aturan tetap, karena dalam permainan jenis ini setiap pembuat *game* mempunyai aturan atau cara yang berbeda-beda. *Action Games* terdiri dari dua jenis, yaitu *single player* atau *multiplayer*, dari segi kamera atau pandangan ”*Action Game”* tidaklah seperti *game Strategy Game, Simulator Games*, atau *Sport Games* yang diam pada satu sisi saja. Tetapi mempunyai dua katagori *First-Person Point-of- View* (FPS) dan *3rd First-Person Point-of-View* (3rd FPS). Pada *First-Person Point-of-View* (FPS) *game*, pandangan pemain hanya sebatas lengan tangan atau senjata saja. Contoh *game* FPS yang paling popular saat ini adalah *point blank, Counter Strike* dan *call of duty*. Pada *action game*s sebagian besar *game* yang dilihat dari segi *gameplay*-nya terutama dari segi AI, tetapi tidak menutup kemungkinan bahwa setiap *game* memiliki AI yang menerapkan metode *Potential Field* pada *Action Games.*

3.1.3.1 Pengenalan Aplikasi *Game Haunted Hospital*.

Jenis permainan yang dibuat adalah merupakan *First Person Shooter* (FPS) *game*, adalah *game* aksi yang mengandalkan tembak - menembak dan mengambil sudut pandang orang pertama, yaitu si pemain *game* itu sendiri. Dalam *game* FPS dilayar monitor tidak akan terlihat karakter utama yang kita mainkan. Yang terlihat biasanya hanya tangan yang memegang senjata. *Game* yang dibuat adalah bergenre *horor*, tokoh utama adalah seorang prajurit bernama sigit yang bertugas sebagai seorang polisi, yang harus menempuh *map* yang berisi beberapa rintangan dan musuh untuk mencapai tujuan akhirnya yaitu mecari penyebab dari

semua kejadian misterius. Dalam permainan ini terdapat beberapa settingan lokasi. setiap lokasi mempunyai beberapa perbedaan tingkat kesulitan.

Kemampuan yang dimiliki oleh sigit adalah berjalan, berlari, dan melompat. Untuk membunuh musuhnya, sigit diharuskan untuk menembak musuhnya dengan menggunakan senjata api yang telah di sediakan sebelumnya. Sigit juga mempunyai kotak darah (*blood*) yang berupa garis-garis putih yang berada pada kotak (*bar*). Setiap kali sigit terkena serangan dari musuh, maka Lerpz akan berkurang darahnya sesuai dengan seberapa kuat musuhnya, namun ada beberapa item penambah darah yang di letakan di beberapa tempat tertentu untuk menambah kotak darah yang dimiliki oleh sigit. Jika sigit kehabisan kotak darah pada *map* tertentu tertentu, maka sigit hanya akan mengulangi di *map* itu saja, kecuali jika pemain memilih tombol *New game* pada *menu* utama.

Pada *map* pertama pemain hanya akan di pandu menuju suatu pintu dari bangunan *"Haunted Hospital"* untuk memasuki pintu gerbang dari bangunan gedung utama nya. Menuju ke sebuah tempat yang bernama Apotik, sebagai tempat *Check point* selama tinggal pada bangunan *"Haunted Hospital"*.

3.1.3.2 *Storyline*

Kisah dari *game* ini dimulai pada 24 Juli 1990. Saat itu di *Flower City* terjadi banyak kasus misterius (lebih tepatnya aneh dan menjijikan) di daerah sekitar *"Haunted Hospital"*, seperti mayat wanita yang dimutilasi, atau sering adanya laporan mengenai kemunculan makhluk misterius yang berkeliaran.

J.A.G.U.R. divisi khusus Departemen Kepolisian *Flower City* , mengirimkan salah satu timnya, yaitu Pikitdro 1 yang dipimpin oleh Ario Deska untuk menyelidiki kasus ini. Namun mereka kehilangan kontak dengan para anggota Pikitdro 1 ini dan secara resmi dinyatakan hilang pada tanggal 28 Juli 1990. Kemudian Pikitdro 2 dikirim untuk mencari tim yang hilang secara misterius tersebut dan melanjutkan penyelidikan yang mereka lakukan.

Setelah tiba di *"Haunted Hospital"*, Pikitdro 2 menemukan helikopter yang digunakan oleh Pikitdro 1, namun tidak ada tanda-tanda kehidupan didalamnya.

Sebaliknya sekarang mereka malah diserang oleh sekelompok tuyul cebol. Dalam perjuangan mereka bertahan melawan tuyul-tuyul cebol tersebut Michael Stepanus, Salah satu anggota Pikitdro 2, terbunuh.

Dalam situasi yang menegangkan ini Ryan Heryanto, pilot dari pesawat yang ditumpangi oleh tim Pikitdro 2 dan juga dikenal dengan julukan "Buye", panik dan terbang meninggalkan anggota tim yang lainnya. Para anggota Pikitdro 2 kemudian bersembunyi di dalam sebuah Apotik tua milik Juragan Febri Ariansyah, yang mereka kira aman.Terjebak dalam sebuah Apotik yang (kelihatannya) cukup aman, para anggota Pikitdro 2 yang selamat adalah Faris Mutaqin, Sigit, Indra D.C, dan Arni Ardini. Mereka kemudian berpencar dan mencari petunjuk mengenai hilangnya Pikitdro 1 dan juga mengenai kasus misterius yang terjadi di sekitar *"Haunted Hospital"*.

Selang beberapa jam setelah mereka berpencar mereka pun di serang oleh para hantu yang ada di sekitaran *"Haunted Hospital"*. Dan hanya menyisakan satu anggota yang bernama Sigit.

3.1.3.3 *Gameplay*

Karakter pemain adalah anggota dari pasukan khusus yang terjebak di sebuah rumahsakit yang dihuni oleh makhluk berbahaya. Tujuan permainan ini adalah untuk mengungkap misteri yang ada di rumahsakit angker tersebut. Graphics pada *game* ini terdiri dari karakter polygonal 3D dan dari berbagai objek yang disusun sedemikian rupa sehingga dapat menjadi latar belakang agar bisa dilihat dari berbagai sudut kamera. Pemain mengontrol karakter pada *game* ini dengan menggunakan keyboard dan mouse sehingga karakter dapat bergerak maju-mundur serta bergerak ke arah kiri dan kanan.

Untuk menyelesaikan misi pada permainan ini, pemain harus menghabisi semua mahluk-mahluk halus yang ada pada setiap *map* pada permainan ini. karakter telah di persenjatai dengan senjata api untuk mempertahankan diri dari musuh, meskipun amunisi tersedia untuk senjata api, Pemain harus belajar untuk menghemat amunisi yang mereka miliki untuk situasi di mana mereka akan benar-benar membutuhkannya. Untuk mengembalikan kesehatan karakter, pemain

menggunakan pertolongan pertama dengan mengambil kotak P3K yang di sebar di tempat-tempat tertentu.

3.1.3.4 Analisis Tingkat Kesulitan

Di dalam *game* ini terdapat 4 *Map* yang saling berhubungan, dimana *Map* 1 harus selesai sebelum ke *Map* 2, begigtu juga dengan Map 3 yang harus di selesaikan terlebih dahulu sebelum ke Map final. Adapun isi pembagian *Map* dapat di lihat seperti di bawah ini :

1. *Map* 1

- Pemain akan dihadapkan dengan semua musuh yang ada pada game, seperti tuyul, pocong, suster ngesod, dan kunti lanak.
- Pemain berjalan mencari jalan menuju Haunted Hospital.
- Pemain menembak musuh sebanyak mungkin yang menjadi rintangan di Map 1 yang berada di sekitar bangunan Haunted Hospital.
- Pemain mencari *icon* obat untuk menambah darah.
- *Icon* darah bisa diambil jika darah pemain berkurang.
- Pemain menggunakan 1 jenis senjata yang diberikan secara otomatis.
- Senjata Senapan mempunyai 30 peluru, hanya bisa di *reload* otomatis.

2. *Map* 2

- Pemain akan dihadapkan dengan semua musuh yang ada pada game, seperti tuyul, pocong, suster ngesod, dan kunti lanak.
- Musuh yang di hadirkan di map ini lebih lengkap dengan kuatan dan kemampuan lebih tinggi dari map sebelumnya.
- Musuh yang sering muncul di Map 2 hanya 3 jenis musuh yaitu, Tuyul, pcong dan Suster ngesod.
- Pemain harus mencari semua musuh yang ada di map 2.
- Pemain harus menghancurkan semua musuh yang ditemukan pada lokasi Map 2.
- Pemain mencari *icon* obat untuk menambah darah.
- *Icon* darah bisa diambil jika darah pemain berkurang.
- Pemain harus mencari pintu untuk menuju Map berikutnya
- Senjata Senapan mempunyai 30 peluru, *reload* secara otomatis

3. *Map* 3

- Pemain akan dihadapkan dengan semua musuh yang ada pada game, seperti tuyul, pocong, suster ngesod, dan kunti lanak.
- Musuh yang di hadirkan di map ini lebih lengkap dengan kuatan dan kemampuan lebih tinggi dari map sebelumnya.
- Pemain harus mencari semua musuh yang ada di map 3.
- Pemain harus menghancurkan semua musuh yang ditemukan pada lokasi Map 3.
- Pemain mencari *icon* obat untuk menambah darah.
- *Icon* darah bisa diambil jika darah pemain berkurang.
- Pemain harus mencari pintu untuk menuju Map berikutnya
- Senjata Senapan mempunyai 30 peluru, *reload* secara otomatis

4. *Map* Final

- Pemain akan dihadapkan dengan semua musuh yang ada pada game, seperti tuyul, pocong, suster ngesod, dan kunti lanak.
- Musuh yang di hadirkan di map ini lebih lengkap dengan kuatan dan kemampuan lebih tinggi dari map sebelumnya.
- Pemain harus mencari semua musuh yang ada di map final.
- Pemain harus menghancurkan semua musuh yang ditemukan pada lokasi Map final.
- Pemain harus mengalahkan 4 musuh terkuat yang menjadi bos dari setiap musuh yang ditampilkan pada Map yang sebelumnya telah diselesaikan.

Pemain mencari *icon* obat untuk menambah darah.

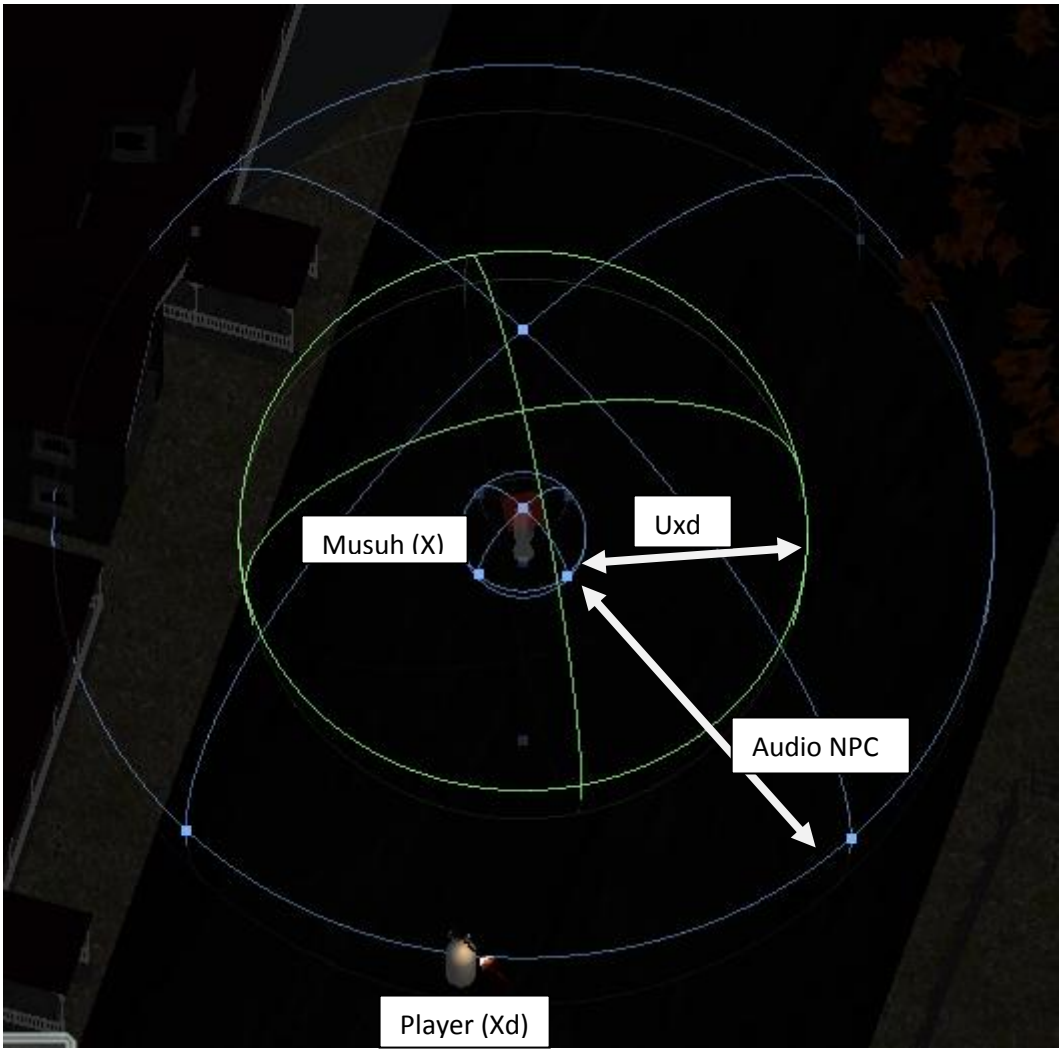
- *Icon* darah bisa diambil jika darah pemain berkurang.
- Senjata Senapan mempunyai 30 peluru, *reload* secara otomatis.

3.1.3.5 Analisa pemecahan masalah dengan menggunakan metode *potential field.*

Jarak jalur yang ditempuh karakter enemy (musuh) menuju nilai tujuan (palyer) dengan menggunakan metode potential field sebagai berikut :

Disini dimisalkan X sebagai nilai posisi *agent* (musuh), Xd sebagai nilai tujuan (*player*) jika karakter player sudah masuk pada nilai tujuan maka enemy akan mengejar karakter player, Kp sebagai nilai konstanta, Uxd sebagai total jarak

karakter enemy mengejar karakter player dari *potential field* attraktif jika melebihi dari nilai total jarak tersebut maka karakter enemy akan kembali posisi semula.



Gambar 3-4 potential field aktraktif pada *game*

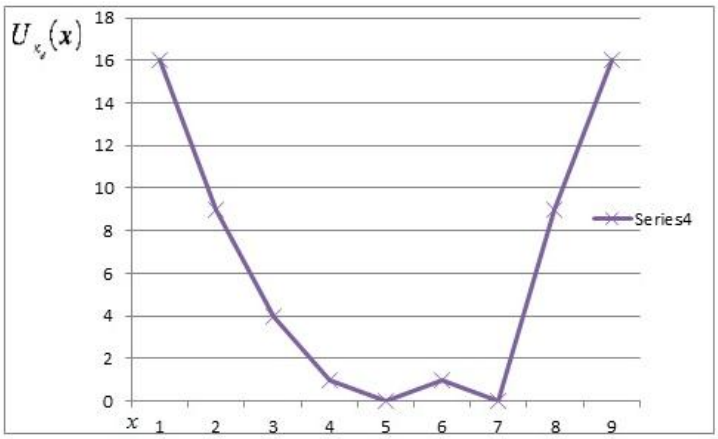
4. Menentukan nilai potential field aktraktif
    1. Potential field pada target location diberi nilai = 2 (*xd = 2*)
    2. Menentukan nilai konstanta (*kp = 2*)
    3. Mencari nilai potential field atraktif menggunakan rumus matematika.

$$U_{x_d}(\boldsymbol{x}) = \frac{1}{2} k_p (\boldsymbol{x} - \boldsymbol{x}_d)^2$$

Tabel 3.1 nilai potential field aktraktif

k_p	x_d	x	$U_{x_v}(x)$
2	2	6	16
2	2	5	9
2	2	4	4
2	2	3	1
2	2	2	0
2	2	1	1
2	2	0	0
2	2	-1	9
2	2	-2	16

Dapat dilihata pada grafik :



Gambar 3-5 grafik nilai potential field aktraktif

1. Total nilai Uxd(X) = 16 maka total nilai tersebut adalah jarak jalur karakter enemy menuju ke player. Jika karakter player melebih total nilai 16 dari karakter enemy sehingga karakter enemy tidak mengejar karakter player dan karakter enemy akan kembali ke posisi semula.
2. Total nilai Uxd(X) = 9 maka total nilai tersebut adalah jarak jalur karakter enemy menuju ke player. Jika karakter player melebih total nilai 9 dari karakter enemy sehingga karakter enemy tidak mengejar karakter player dan karakter enemy akan kembali ke posisi semula.

3. Total nilai Uxd(X) = 4 maka total nilai tersebut adalah jarak jalur karakter enemy menuju ke player. Jika karakter player melebih total nilai 4 dari karakter enemy sehingga karakter enemy tidak mengejar karakter player dan karakter enemy akan kembali ke posisi semula.
4. Total nilai Uxd(X) = 1 maka total nilai tersebut adalah jarak jalur karakter enemy menuju ke player. Jika karakter player melebih total nilai 1 dari karakter enemy sehingga karakter enemy tidak mengejar karakter player dan karakter enemy akan kembali ke posisi semula.
5. Total nilai Uxd(X) = 0 maka karakter enemy tidak mengejar karakter player dikarenakan total nilai = 0.

   Sehingga dari analisa diatas maka dalam *game* yang dibuat menggunakan total nilai potential field aktraktifnya adalah 16 dikarenakan luas jarak jalur menuju karakter player tidak terlalu besar untuk map dalam *game* tersebut. Luas jarak jalur menuju karakter player digambarkan pada *game* berbentuk sebuah lingkaran.

3.1.4 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non fungsional adalah sebuah langkah dimana seorang pembangun perangkat lunak menganalisis sumber daya yang akan digunakan perangkat lunak yang dibangun.

Analisis non fungsional tidak hanya menganalisis siapa saja yang akan menggunakan aplikasi tetapi juga menganalisis perangkat keras dan perangkat lunak, sehingga dapat menentukan kompabilitas aplikasi yang dibangun terhadap sumber daya yang ada. Setelah melakukan analisis non fungsional, maka dilanjutkan kelangkah berikutnya, yaitu menentukan kebutuhan non fungsional sistem yang akan dibangun untuk disesuaikan dengan fakta yang ada.

Apabila terjadi ketidak cocokan antara fakta dan kebutuhan, maka perlu adanya penyesuaian fakta terhadap kebutuhan yang ada. Apabila kebutuhan tidak dipenuhi maka sistem yang akan dibangun tidak akan berjalan baik sesuai dengan yang diharapkan.

3.1.4.1 Sasaran *User* dan Kegunaan Aplikasi

Game yang akan dibuat ditujukan untuk pengguna dengan usia 15 tahun ke atas, alasannya adalah pada usia ini lebih senang bermain dengan permainan yang lebih mempunyai karakter, *game*-nya mempunyai tujuan yang jelas, tidak terlalu sulit , maupun dari segi permainannya yang terlalu mudah.

3.1.4.2 Analisis dan Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintahyang diberikan kepada perangkat keras agar bisa saling berinteraksi diantara keduanya. Perangkat lunak yang dibutuhkan untuk membangun aplikasi *game superhero* pasta gigi vs *tooth germs* ini adalah sebagai berikut :

1. Sistem Operasi *Window XP*.
2. *Unity game* 3D
3. *Autodesk Maya* 3D
4. *Sketchup Pro*

3.1.4.3 Analisis dan Kebutuhan Perangkat Keras

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan suatu tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya. Pada aplikasi *game* ini, perangkat keras yang digunakan adalah sebagai berikut :

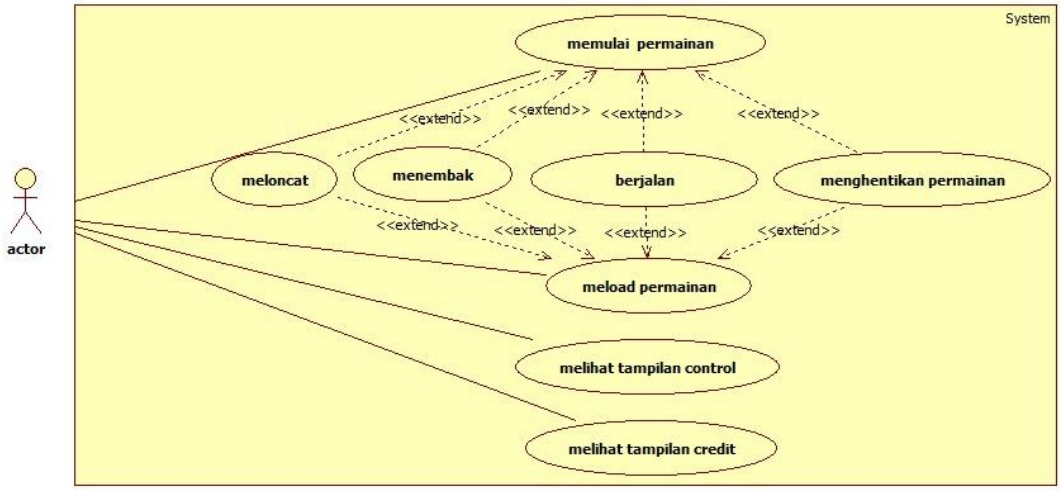
1. *Intel Core* 2 *Duo Processor* T6500
2. RAM : 1GB
3. VGA 256 MB.
4. *Harddisk* 320 GB.
5. Monitor .
6. *Mouse* dan *Keyboard*.

3.1.5 Analisa kebutuhan fungsional

Tahap pertama yang dilakukan dalam melakukan analisis berorientasi objek menggunakan UML adalah menentukan aktor atau pengguna sistem. Kata aktor dalam konteks UML, menampilkan peran (*rules*) pengguna atau sesuatu diluar sistem yang dikembangkan yang dapat berupa perangkat keras, end user, sistem yang lain , dan sebagainya.

3.1.5.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Berikut ini adalah perancangan proses-proses yang terdapat pada *game FPS 3D "Haunted Hospital":*



Gambar 3-6 Use Case Diagram menu utama

1. **Definisi *Actor***

Definisi *Actor* berfungsi untuk menjelaskan *actor* yang terdapat pada *use case diagram*. Definisi *actor* diterangkan pada tabel 3-2.

Tabel 3-2 Definisi *Actor*

No	*Actor*	Deskripsi
1	Pemain	Aktor utama yang mengakses aplikasi.

2. Definisi *Use Case*

Definisi *Use Case* adalah menggambarkan *external view* dari system yang akan kita buat modelnya.

Tabel 3-3 Definisi Use Case

No	***Use Case***	**Deskripsi**
1	Memulai Permainan	Proses untuk memulai permainan.
2	*Load Game*	Proses untuk melanjutkan permainan yang tertunda.
3	Melihat *Control*	Proses untuk menampilkan *control* permainan.
4	Melihat *Credit*	Proses untuk menampilkan tentang pembuat permainan.
5	Menembak	Proses untuk menampilkan aksi tembakan
6	Berjalan	Proses untuk menggerakan karakter sesuai dengan inputan keyboard
7	Meloncat	Proses untuk menggerakan karakter sesuai dengan inputan keyboard
8	Menghentikan Permainan	Proses berhenti ketika permainan sedang berlangsung (*Pause).*

3.1.5.2 Skenario *Use Case*

Skenario *Use Case* merupakan penggambaran dari alur penggunaan sistem dimana setiap skenario digambarkan dari sudut pandang *actor*, seseorang atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara.

1. Skenario *Use Case* Memulai Permainan

Tabel 3-4 di bawah ini adalah skenario *use case* memulai permainan pada *game*.

Tabel 3-4 Skenario *Use Case* Memulai Permainan

Identifikasi	
Nomor	1
Nama	Memulai permainan
Tujuan	Memulai permainan
Deskripsi	Proses untuk memulai permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Memilih menu New Game

Aksi Aktor	Reaksi Sistem
Memilih menu New Game.	
	Mengecek data *gameplay*.
Masuk ke tampilan *gameplay*.	
	Menampilkan *introduction game* berupa *text* dan gambar.
Memilih tombol *skip*.	
	Menampilkan tampilan karakter *player*, tampilan *map*, dan tampilan karakter *enemy*, data *health*, dan lain-lain.
Pemain memainkan karakter *player* dan melihat animasi *enemy*.	
	a. Mengecek data animasi. b. Menampilkan info tampilan karakter

	player. c. Menampilkan info tampilan karakter *enemy*.
Skenario tambahan	
Aksi aktor	Reaksi sistem
Tidak memilih tombol skip	
	Akan melanjutkan tampilan introduction sampai selesai.
Kondisi Akhir	Pemain masuk ke dalam arena permainan.

2. Skenario *Use Case* Load Game

Tabel 3-5 di bawah ini adalah skenario *use case* Load Game pada *game*.

Tabel 3-5 Skenario *Use Case* Load Game

Identifikasi	
Nomor	2
Nama	Load Game
Tujuan	Melanjutkan permainan yang tertunda sebelumnya.
Deskripsi	Proses untuk melanjutkan kembali permainan ketika permaian terhenti atau dihentikan.
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Memilih *menu* Load Game
Aksi Aktor	Reaksi Sistem
Memilih *menu* Load Game.	
	a. Mengecek data *player* data *map*, data karakter *player*. b. Menampilkan tampilan *map* yang terhenti sebelumnya.
Kondisi Akhir	Pemain masuk ke dalam arena permainan.

3. Skenario *Use Case* Melihat *Control*

Tabel 3-6 di bawah ini adalah skenario *use case* Melihat Control pada *game*.

Tabel 3-6 Skenario *Use Case* Melihat *Control*

Identifikasi		
Nomor	3	
Nama	Melihat Control	
Tujuan	Menampilkan *control game*	
Deskripsi	Proses menampilkan *control game* berupa gambar control dan *text control* permainan.	
Aktor	Pemain	
Skenario Utama		
Kondisi Awal	Pemain berada di tampilan *menu control.*	
Aksi Aktor		Reaksi Sistem
Memilih *menu control*		
		Menampilkan *info control game* yang beruapa gambar dan *text.*
Kondisi Akhir	Menampilkan info *control game.*	

4. Skenario *Use Case* Melihat *credit*

Tabel 3-7 di bawah ini adalah skenario *use case* Melihat *credit* pada *game*.

Tabel 3-7 Skenario *Use Case* Melihat Credit

Identifikasi		
Nomor	4	
Nama	Melihat Credit	
Tujuan	Menampilkan *credit game*	
Deskripsi	Proses menampilkan *credit game* berupa informasi gambar dan *text* tentang pembuat game.	
Aktor	Pemain	
Skenario Utama		
Kondisi Awal	Pemain berada di tampilan *menu credit.*	
Aksi Aktor		Reaksi Sistem
Memilih *menu control*		
		Menampilkan *info credit game* yang beruapa gambar dan *text.*
Kondisi Akhir	Menampilkan info tentang pembuat *game.*	

5. Skenario *Use Case* menembak

Tabel 3-8 di bawah ini adalah skenario *use case* menembak

Tabel 3-8 Skenario *Use Case* menembak

Identifikasi		
Nomor	5	
Nama	Menembak	
Tujuan	Membunuh character musuh	
Deskripsi	Proses untuk melakukan tembakan dalam permainan	
Aktor	Pemain menkan tombol mouse kiri	
Skenario Utama		
Kondisi Awal	Pemain menekan tombol kiri pada mouse	
Aksi Aktor		Reaksi Sistem
Menekan tombol kiri pada mouse untuk menembak		
		Mengeluarkan efek tembakan berupa partikel dan efek cahaya.
jika mengenai character maka darah character berkurang.		
		Peluru berkurang
Kondisi Akhir	Menampilkan aksi sesuai dengan inputan dari mouse	

6. Skenario *Use Case* berjalan

Tabel 3-9 di bawah ini adalah skenario *use case* berjalan

Tabel 3-9 Skenario *Use Case* berjalan

Identifikasi		
Nomor	6	
Nama	Berjalan	
Tujuan	Menggerakan karakter dengan inputan keyboard	
Deskripsi	Proses untukmenggerakan karakter atau tokoh utama yang dimainkan dalam permainan menggunakan tombol keyboard	
Aktor	Pemain	
Skenario Utama		
Kondisi Awal	Pemain menekan tombol keboard yang di sediakan di dalam permainan	
Aksi Aktor		Reaksi Sistem
Menekan tombol W		
		Karakter berjalan ke depan
Menekan tombol W dan A secara bersamaan		
		Karakter berjalan maju ke sebelah kiri
Menekan tombol W dan D secara bersamaan		
		Karakter berjalan maju ke kanan
Menekan tombol D		
		Karakter berjalan
Menekan tombol S		
		Karakter berjalan berjalan lurus kebelakang
Pemain menekan tombol S dengan D secara bersamaan		
		Karakter berjalan mundur ke kanan
Menekan tombol S dan A secara bersamaan		
		Karakter berjalan mundur ke kiri
Kondisi Akhir	Menampilkan aksi pergerakan karakter sesuai dengan inputan keyboard	

7. Skenario *Use Case* meloncat

Tabel 3-10 di bawah ini adalah skenario *use case* meloncat.

Tabel 3-10 Skenario *Use Case* meloncat

Identifikasi		
Nomor	7	
Nama	Meloncat	
Tujuan	Menggerakan charakter untuk bisa menloncati objek	
Deskripsi	Proses untuk menggerakan karakter atau tokoh utama yang dimainkan dalam permainan menggunakan tombol keyboard	
Aktor	Pemain	
Skenario Utama		
Kondisi Awal	Pemain menekan tombol keboard yang di sediakan di dalam permainan	
Aksi Aktor		Reaksi Sistem
Menekan tombol “SPACE”pada keyboard		
		Menampilkan karakter yang meloncat sesuai inputan dari pemain
Kondisi Akhir	Menampilkan aksi pergerakan karakter sesuai dengan inputan keyboard	

8. Skenario *Use Case* menghentikan permainan *(pause)*

Tabel 3-11 di bawah ini adalah skenario *use case* menghentikan permainan.

Tabel 3-11 Skenario *Use Case* menghentikan permainan

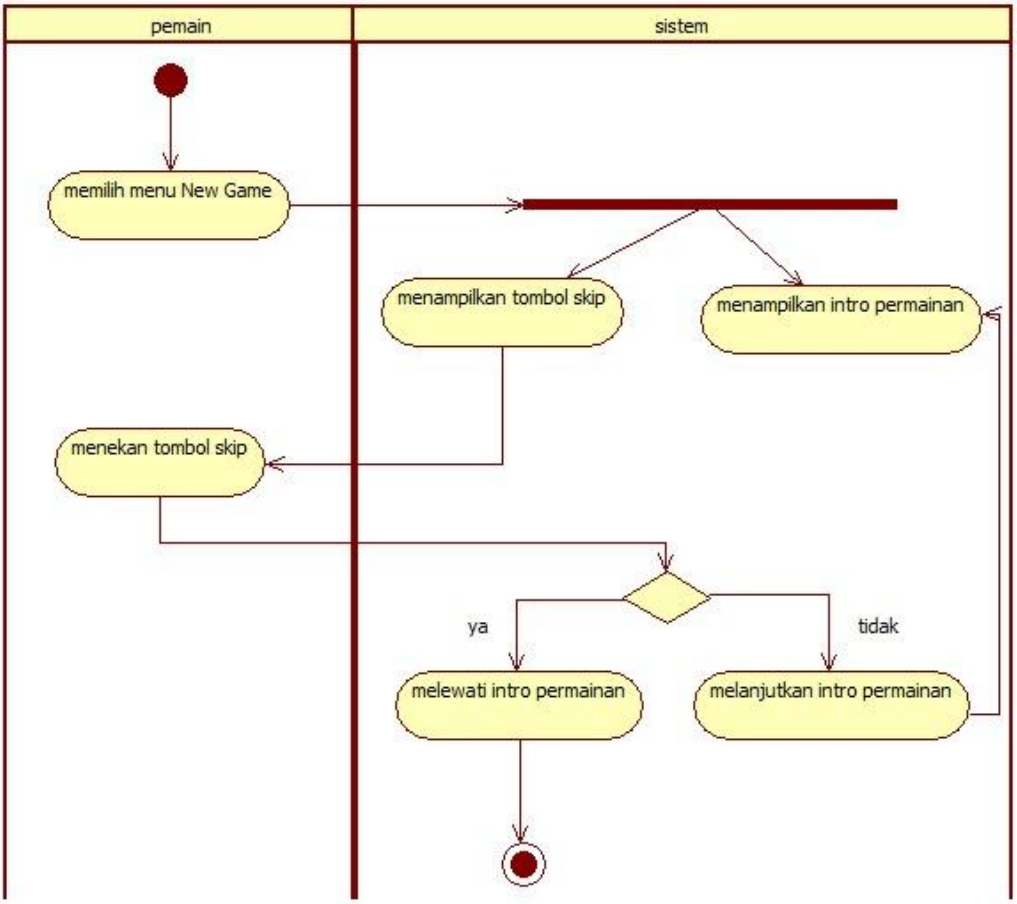
Identifikasi		
Nomor	8	
Nama	Menghentikan permainan	
Tujuan	Menghentikan permainan sementara	
Deskripsi	Proses memberhentikan sementara permainan	
Aktor	Pemain	
Skenario Utama		
Kondisi Awal	Pemain menkan tombol “ESC” pada keyboard	
Aksi Aktor		Reaksi Sistem
Menekan tombol “ESC” pada keyboard		
		Menampilkan informasi pause menu
Menekan tombol “ESC” untuk yang ke dua kali		
		Kembali ke tampilan permainan
Memilih opsi Main Menu pada opsi pause menu		
		Kembali ke tampilan menu utama
Memilih opsi Change Graphics Quality pada pause menu		
		Merubah tampilan grafik layar sesuai dengan opsi yang di pilih
Memilih opsi Quit Game pada pause menu		
		Keluar dari sistem
Kondisi Akhir	1. Menampilkan tampilan pause menu 2. Menjalan aksi sesuai opsi yang ada pada tampilan pause menu 3. Menghentikan permainan sementara	

3.1.5.3 Activity diagram

Activity diagram berisi penjelasan lebih lengkap mengenai urutan suatu proses yang terjadi antara pelaku dan sistem. Aktivitas yang pertama selalu dimulai dari pelaku, kemudian akan mendapat respon dari sistem yang nantinya akan menampilkan proses akhir dari proses tersebut.

***1. Activity diagram* memulai permainan**

Proses ini dimulai ketika pemain akan mulai memainkan game dengan mengakses menu utama, setelah itu kemudian pemain dapat memilih langsung menu New Game dengan cara meng-klik button New Game, setelah itu player dapat memulai permainan dengan diawali dengan intro game.

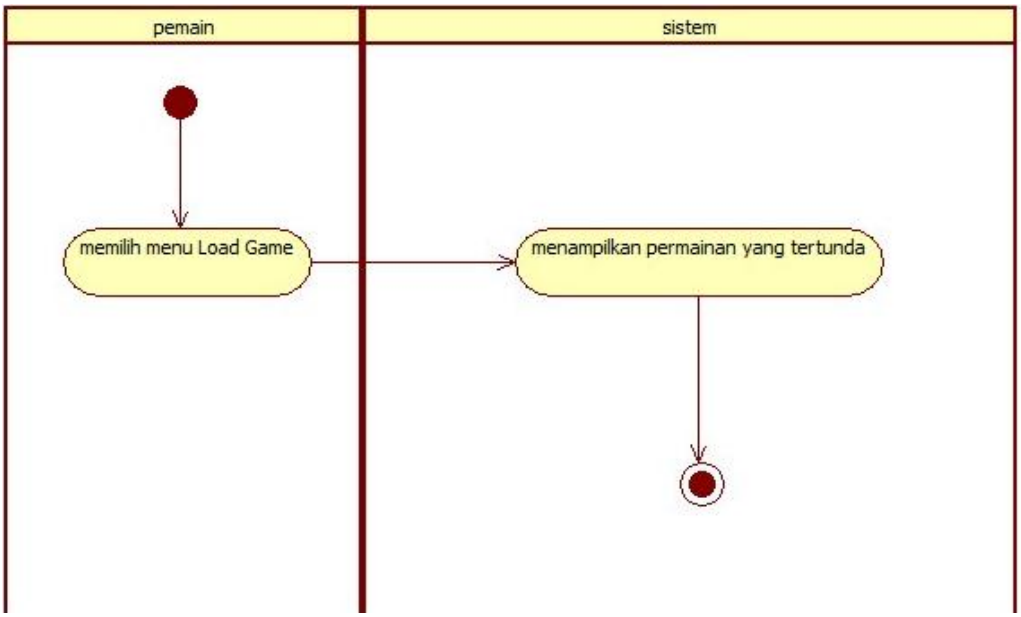


Gambar 3-7 *Activity diagram* memulai permainan

2. Activity diagram Load Game

Proses untuk meload atau melanjutkan kembali permainan yang sempat tertuda sebelumnya.

Proses ini dimulai ketika si pemain memilih menu load game pada menu utama, sistem langsung menampilkan tampilan permainan pada misi yang sempat tertunda sebelumnya.

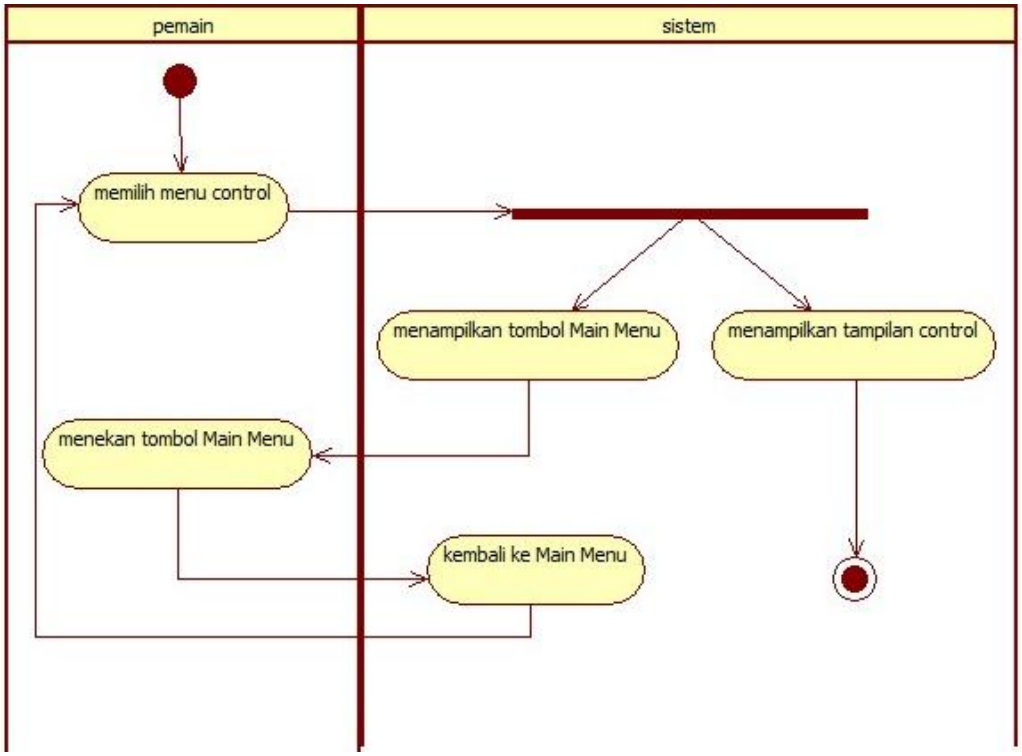


Gambar 3-8 *Activity diagram* load game

3. Activity diagram melihat *Control* permainan

Proses untuk melihat tampilan control pada permainan.

Proses ini akan dijalankan jika pemain memilih menu control untuk melihat control dari permainan, dan sistem akan memproses untuk menampilkan halaman control.

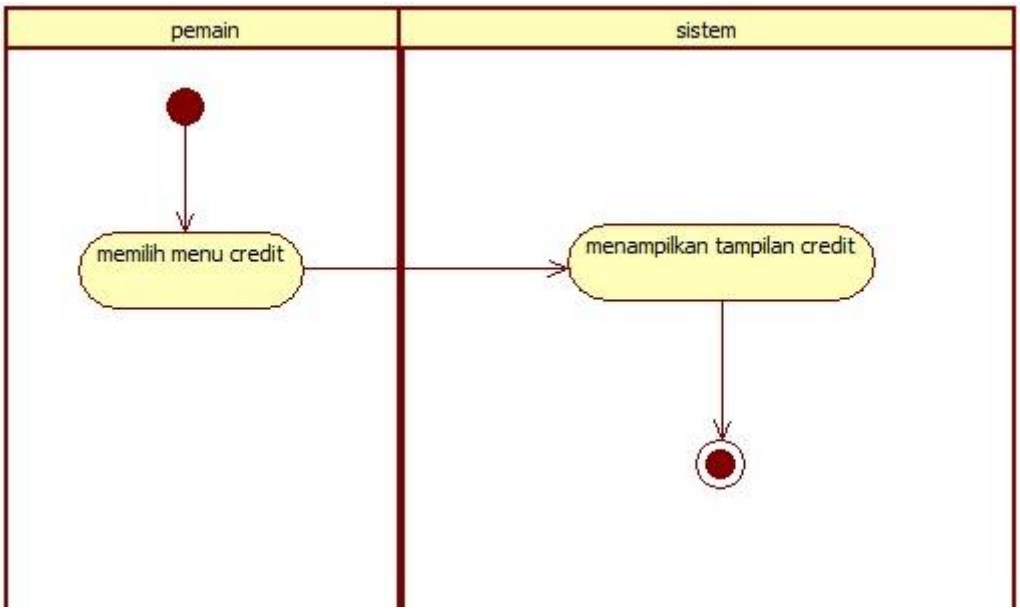


Gambar 3-9 *Activity diagram* melihat control

4. Activity diagram melihat *Credit* permainan

Proses untuk melihat tampilan *credit* pada permainan.

Proses ini dimulai ketika pemain memilih menu *credit* pada halaman utama. Sistem kemudian menampilkan halaman *scene* semacam *credit*, tentang pembuat aplikasi *game* kepada pemain.

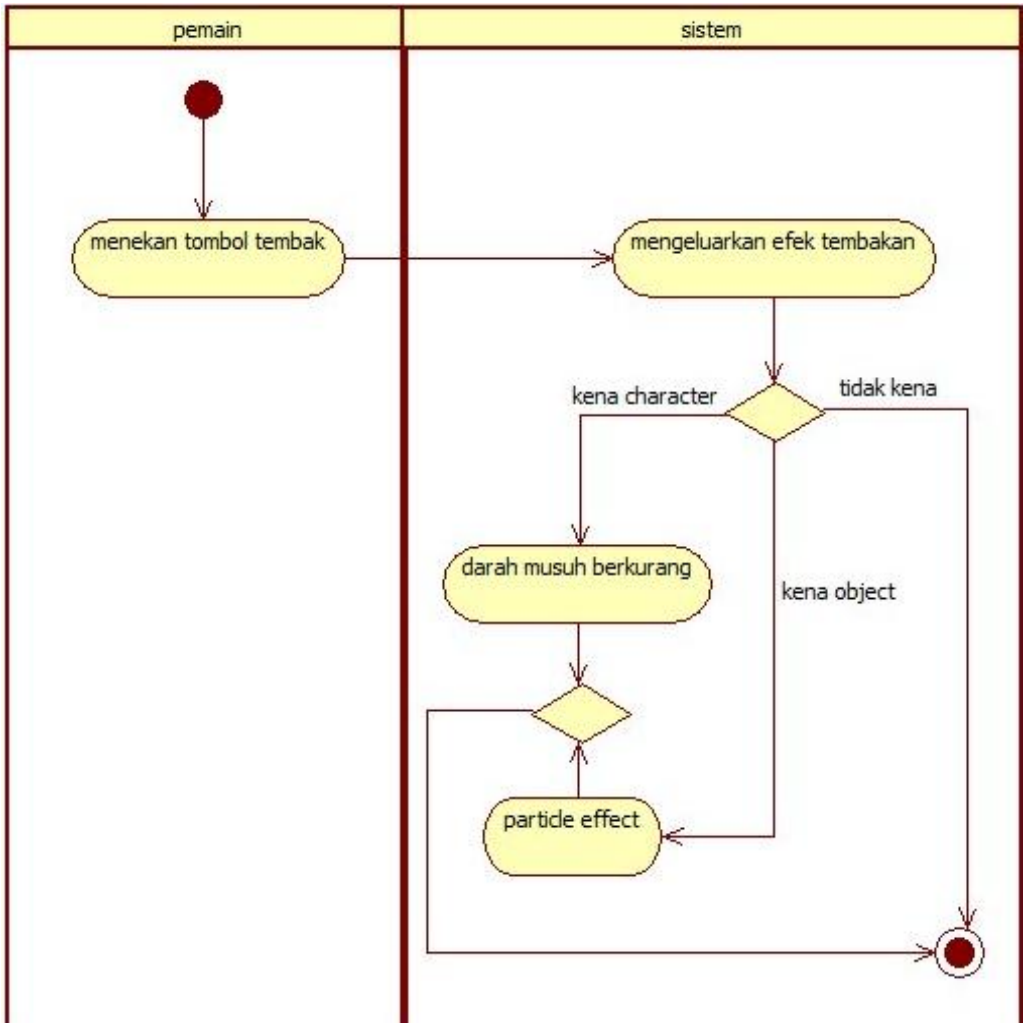


Gambar 3-10 *Activity diagram* melihat credit

5. Activity diagram menembak

Proses untuk melakukan tembakan dalam permainan

Proses ini dimulai ketika pemain menekan tombol kiri pada mouse untuk menembak, sistem mengeluarkan efek berupa suara, dan patikel tembakan. Jika mengenai character maka darah *character* berkurang jika mengenai objek menghasilkan *particle* api, Peluru berkurang.

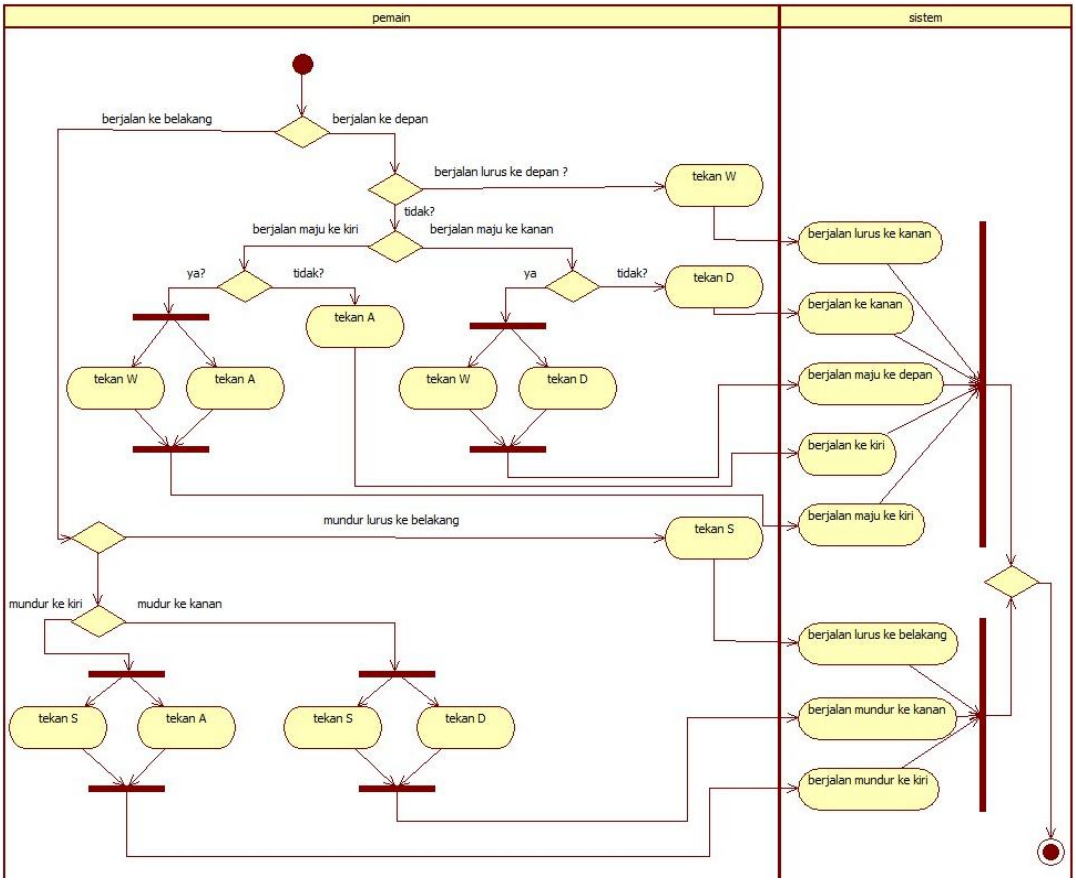


Gambar 3-11 *Activity diagram* menembak

6. Activity diagram berjalan

Proses untuk menggerakan karakter atau tokoh utama yang dimainkan dalam permainan menggunakan tombol *keyboard.*

Proses ini dimulai ketika pemain menekan tombol W, sistem menggerakan karakter berjalan ke depan. Pemain menekan tombol W dengan A secara bersamaan, sistem menggerakan karakter berjalan maju ke sebelah kiri. Pemain menekan tombol A, sistem menggerakan karakter berjalan ke kiri. Pemain menekan tombol W dengan D secara bersamaan, sistem menggerakan karakter berjalan maju ke kanan. Pemain menekan tombol D, sistem menggerakan karakter berjalan ke kanan. Pemain menekan tombol S, sistem menggerakan karakter berjalan lurus mundur. Pemain menekan tombol S dengan D secara bersamaan, sistem menggerakan karakter berjalan mundur ke kanan. Pemain menekan tombol S dengan A secara bersamaan, sistem menggerakan karakter berjalan mundur ke kiri

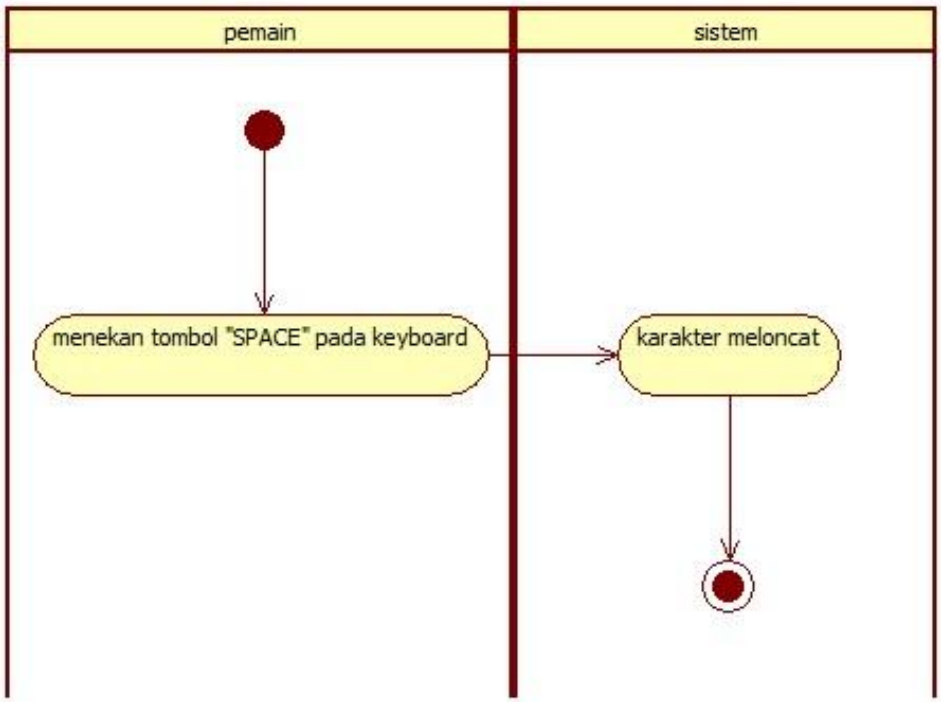


Gambar 3-12 *Activity diagram* berjalan

7. Activity diagram meloncat

Proses untuk menggerakan karakter atau tokoh utama yang dimainkan dalam permainan menggunakan tombol *keyboard.*

Proses ini dimulai ketika pemain menekan tombol space pada *keyboard*, kemudian sistem menampilkan karakter yang meloncat sesuai inputan pemain.

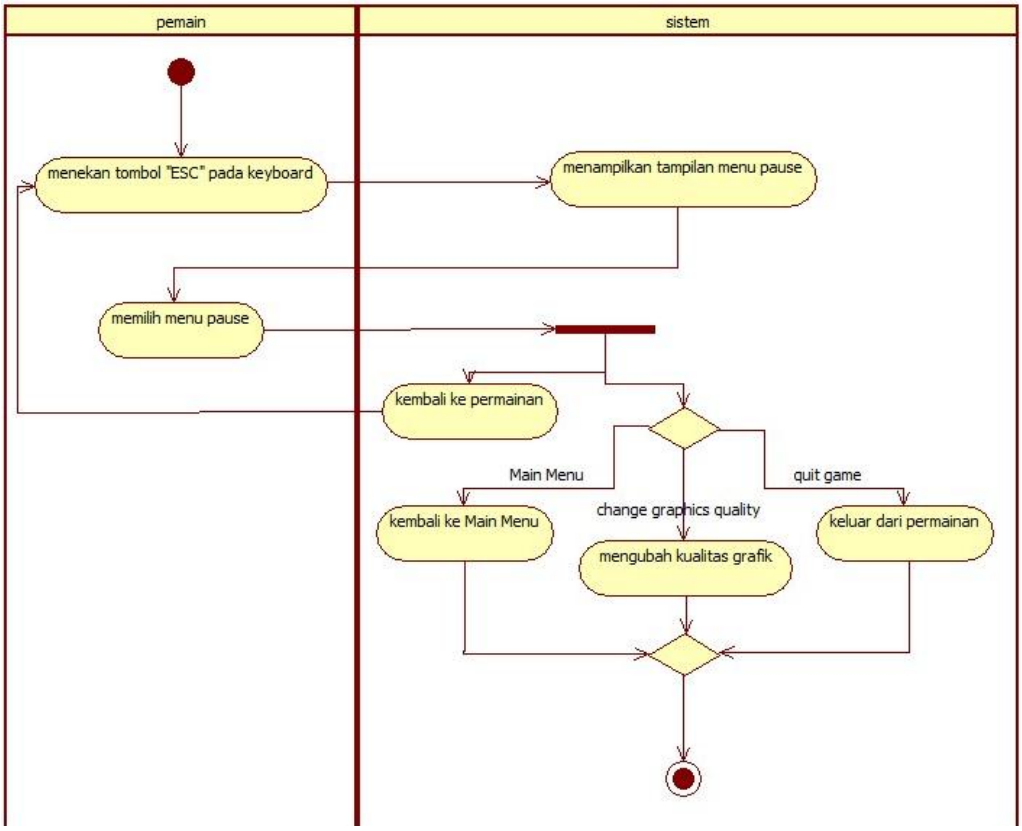


Gambar 3-13 *Activity diagram* meloncat

8. Activity diagram menghentikan permainan *(pause)*

Proses ketika pemain ingin memberhentikan sejenak permainan atau kembali *ke start menu*

Proses ini dimulai ketika pemain menekan tombol “ESC” pada keyboard, sistem menampilkan informasi *pause menu*. Sistem akan kembali ke permainan apabila pemain tombol “ESC” sekali lagi. Sistem akan keluar permainan dan kembali ke *main menu* apabila pemain memilih opsi *main menu.* Sistem dapat merubah qualitas grafik dengan menekan opsi *change graphics quality.* Sistem dapat langsung keluar dari sistem dengan menekan opsi *quit game.*

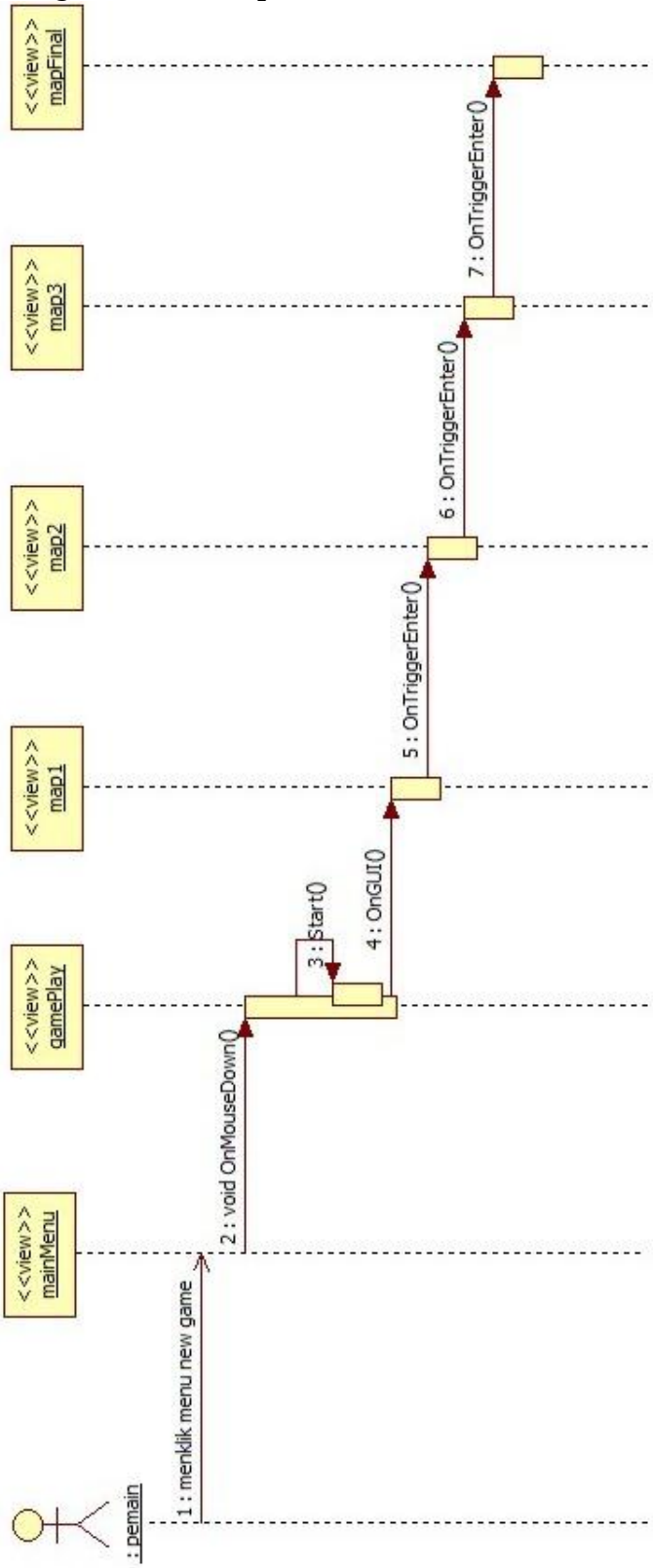


Gambar 3-14 *Activity diagram* menghentikan permainan *(pause)*

3.1.5.4 Sequence Diagram

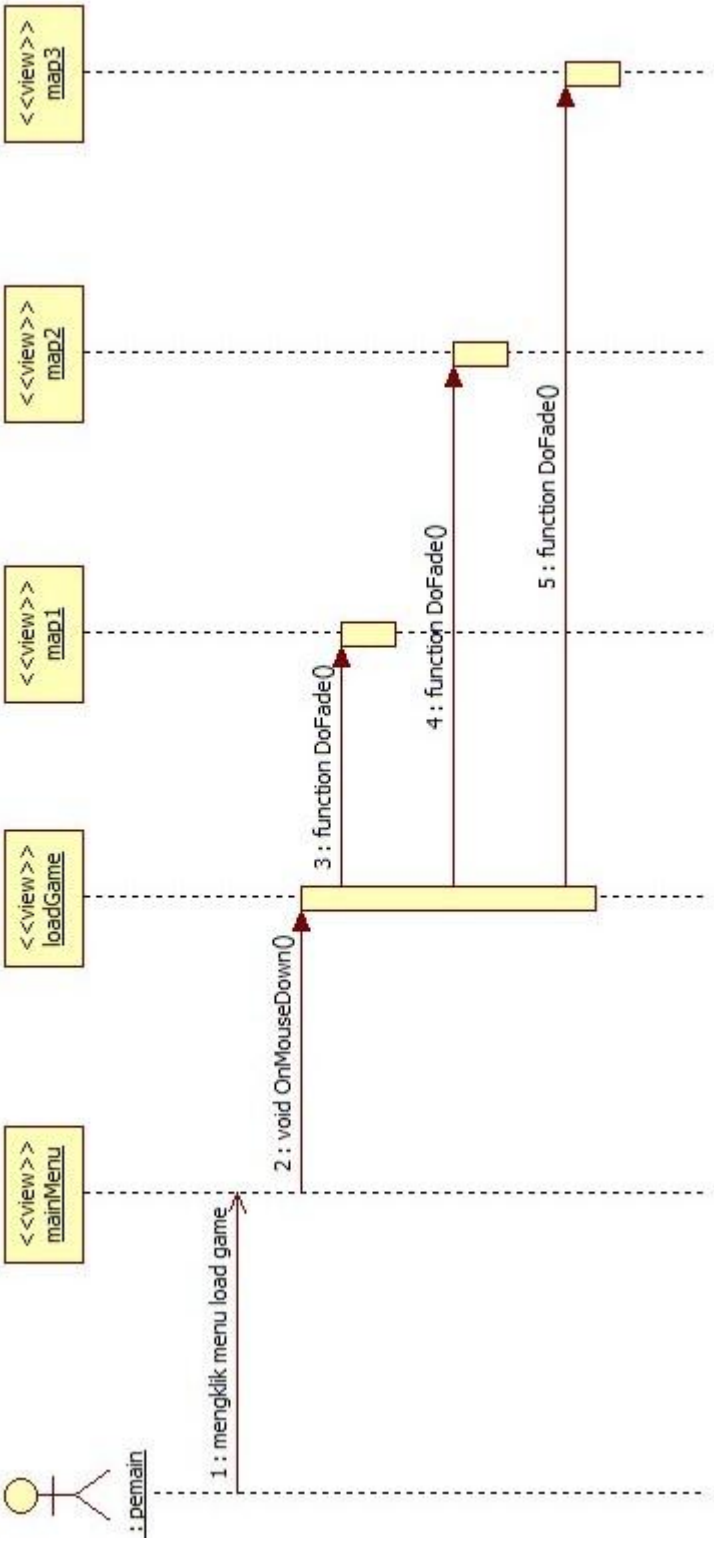
Sequencial diagram menggambarkan interaksi antar objek di dalam dan disekitar sistem berupa *message* yang digambarkan terhadap waktu. Berikut ini beberapa *sequencial diagram* yang terdapat pada *game Hauted Hospital* yaitu :

1. ***Sequence Diagram*** **memulai permainan**



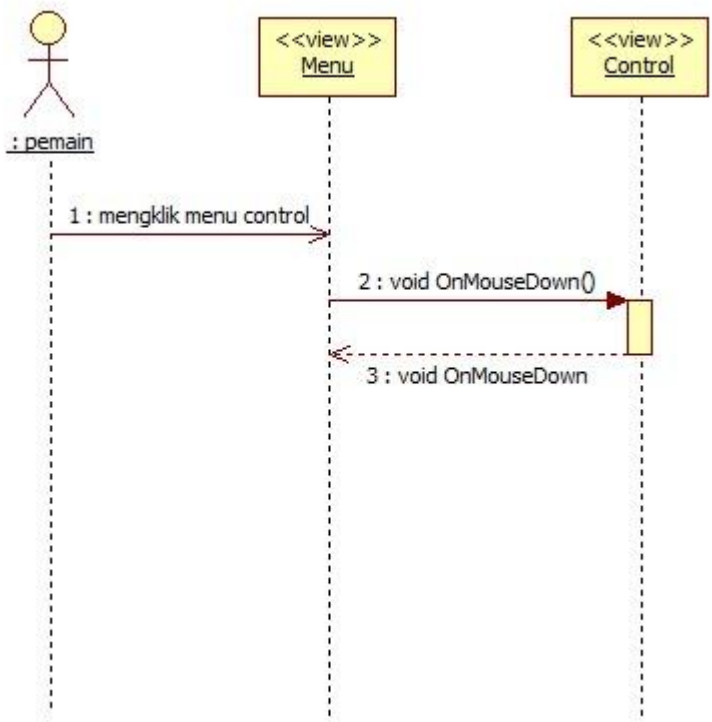
Gambar 3-15 ***Sequence Diagram*** **memulai permainan**

2. *Sequence Diagram* Load Game



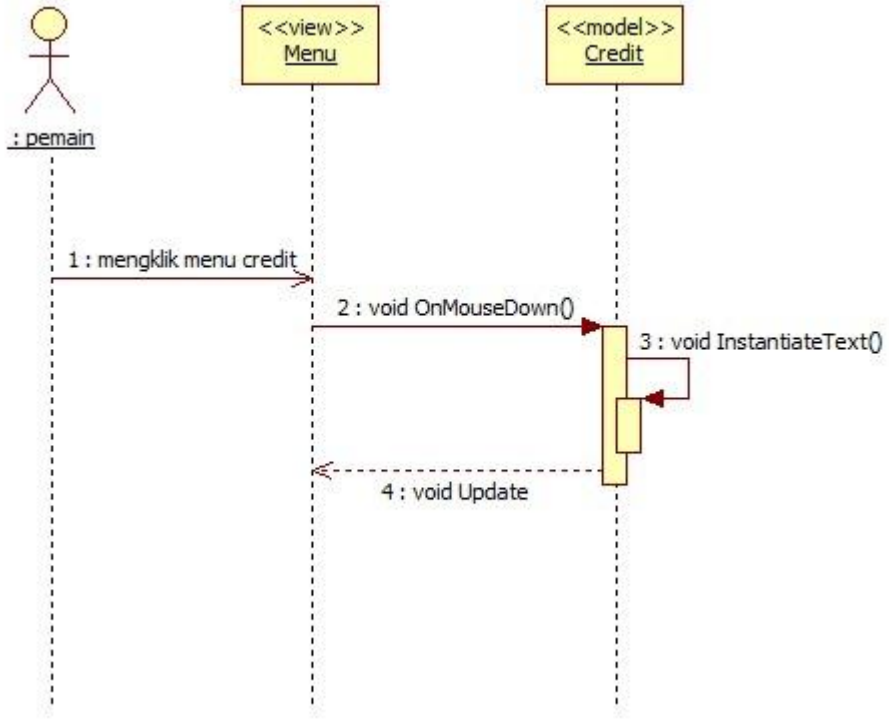
Gambar 3-16 *Sequence Diagram load* game

3. *Sequence Diagram* melihat *Control*



Gambar 3-17 *Sequence Diagram* melihat *control*

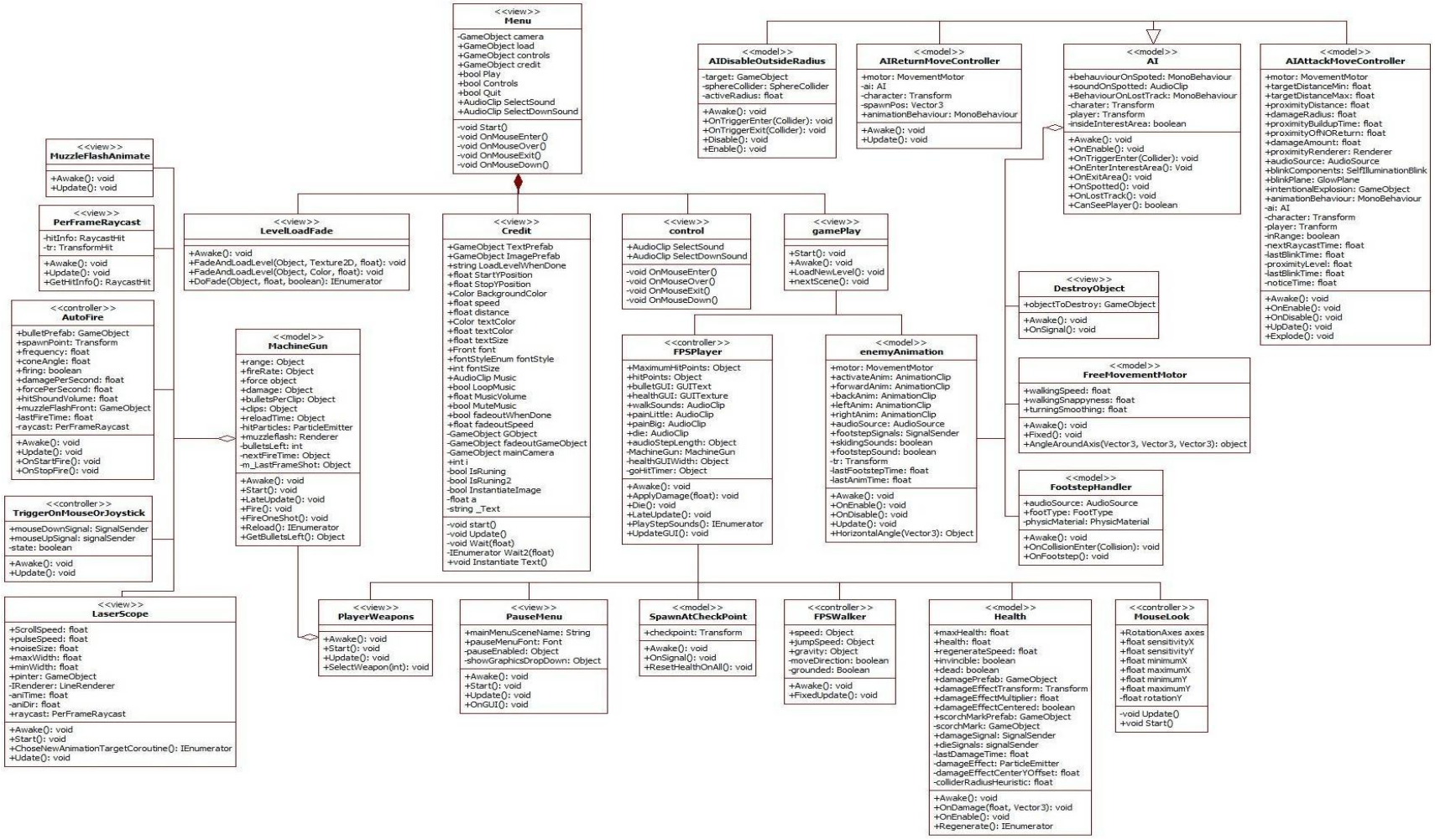
4. *Sequence Diagram* melihat *Credit*



Gambar 3-18 *Sequence Diagram* melihat *Credit*

3.1.5.5 Class diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Berikut adalah *class diagram* pada *game Haunted Hospital* :



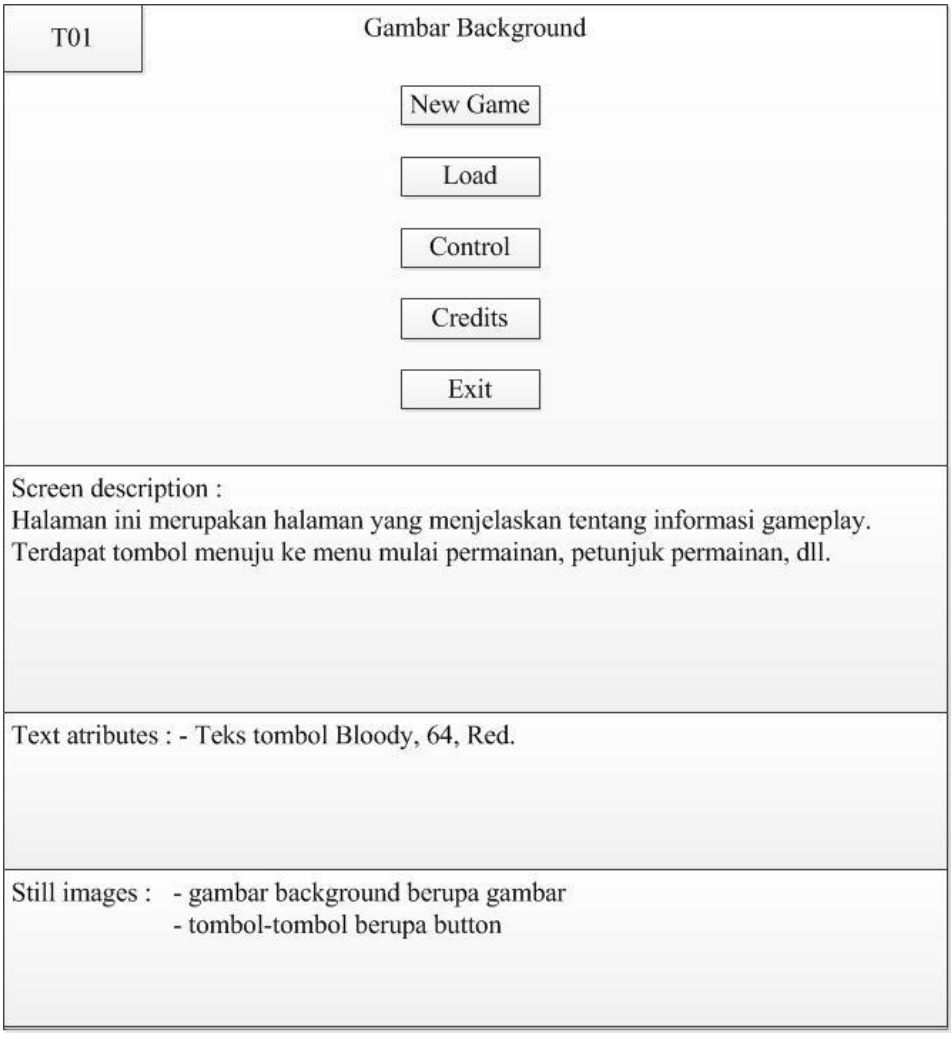
Gambar 3-19 *class diagram*

3.2 Perancangan Sistem

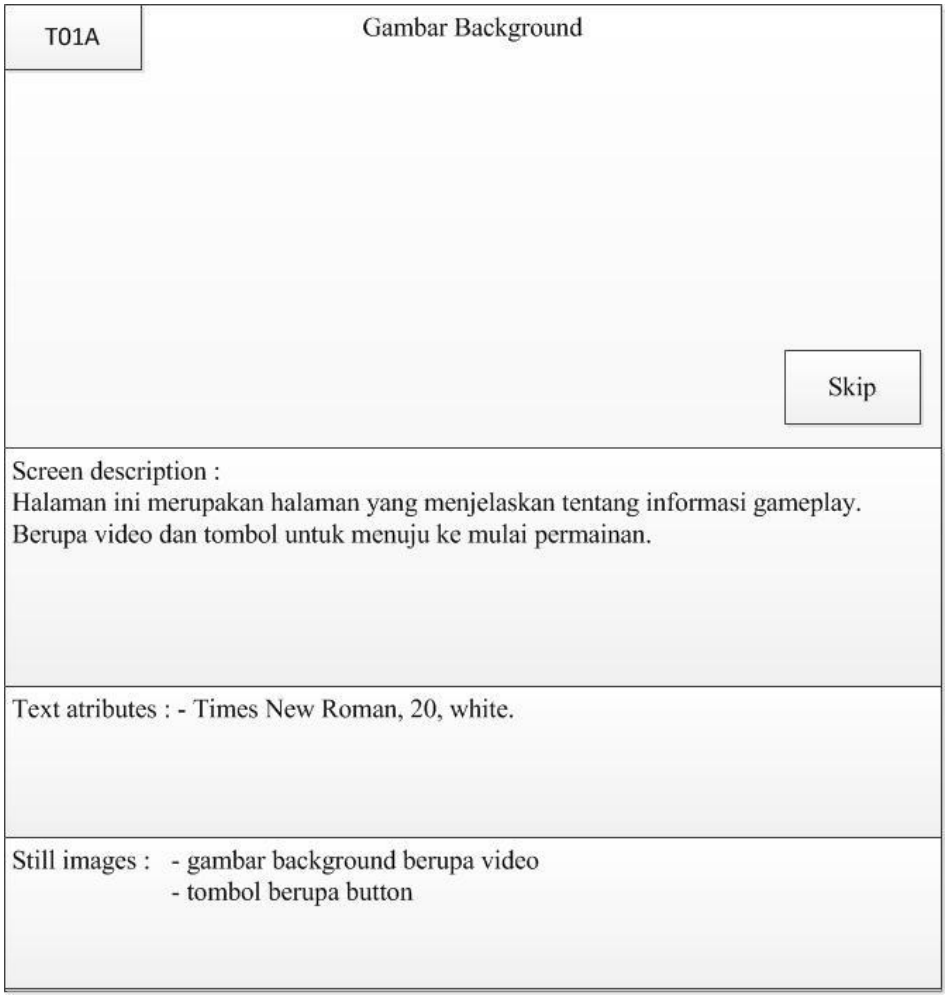
Perancangan ini bertujuan untuk menerapkan solusi pemecahan masalah yang telah diajukan pada analisis sistem

3.2.1 Perancangan Antar Muka

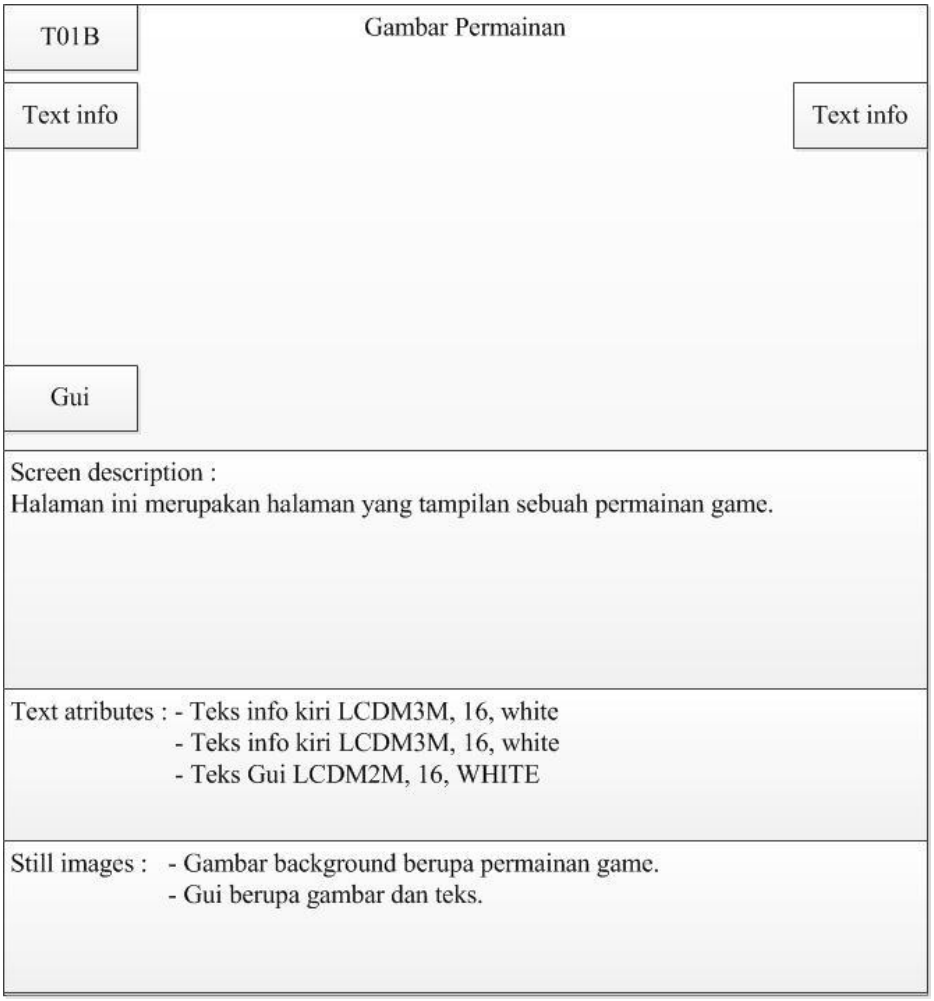
Perancangan antarmuka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun, berikut akan disajikan perancangan antarmuka dari aplikasi yang akan dibangun:



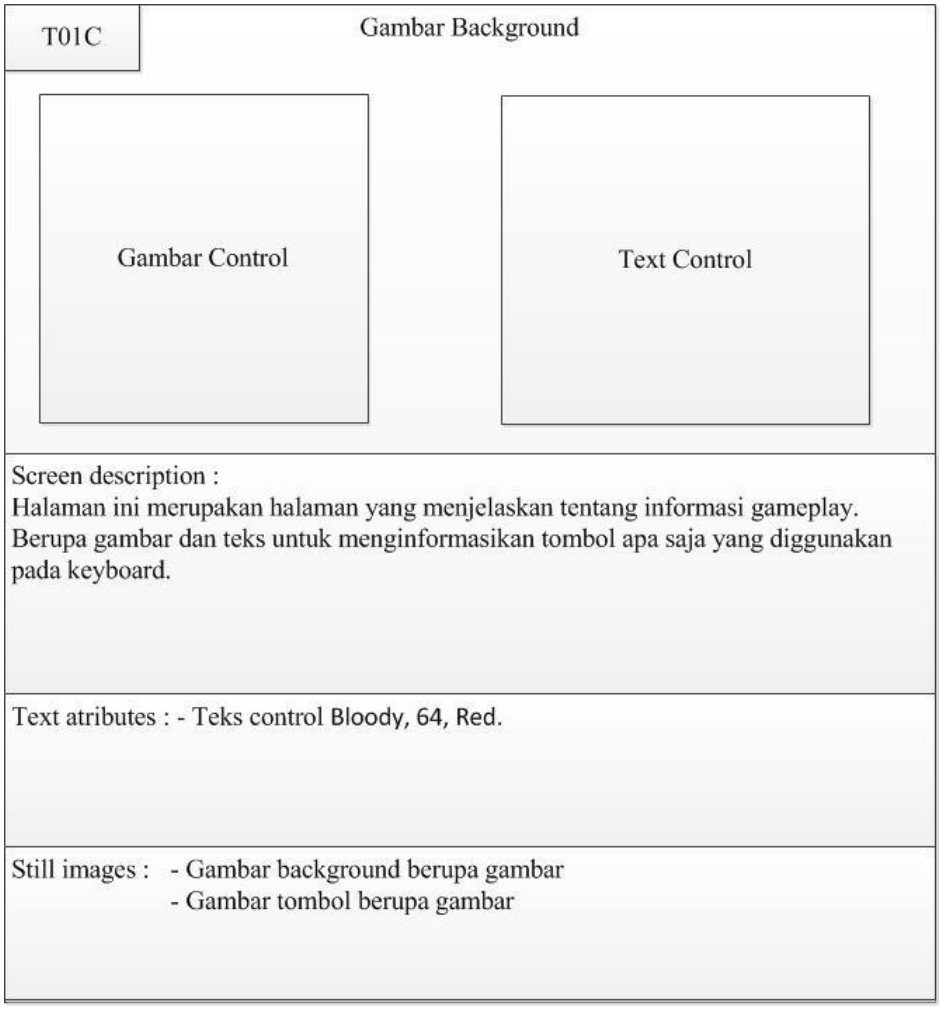
Gambar 3-20 tampilan antar muka menu utama



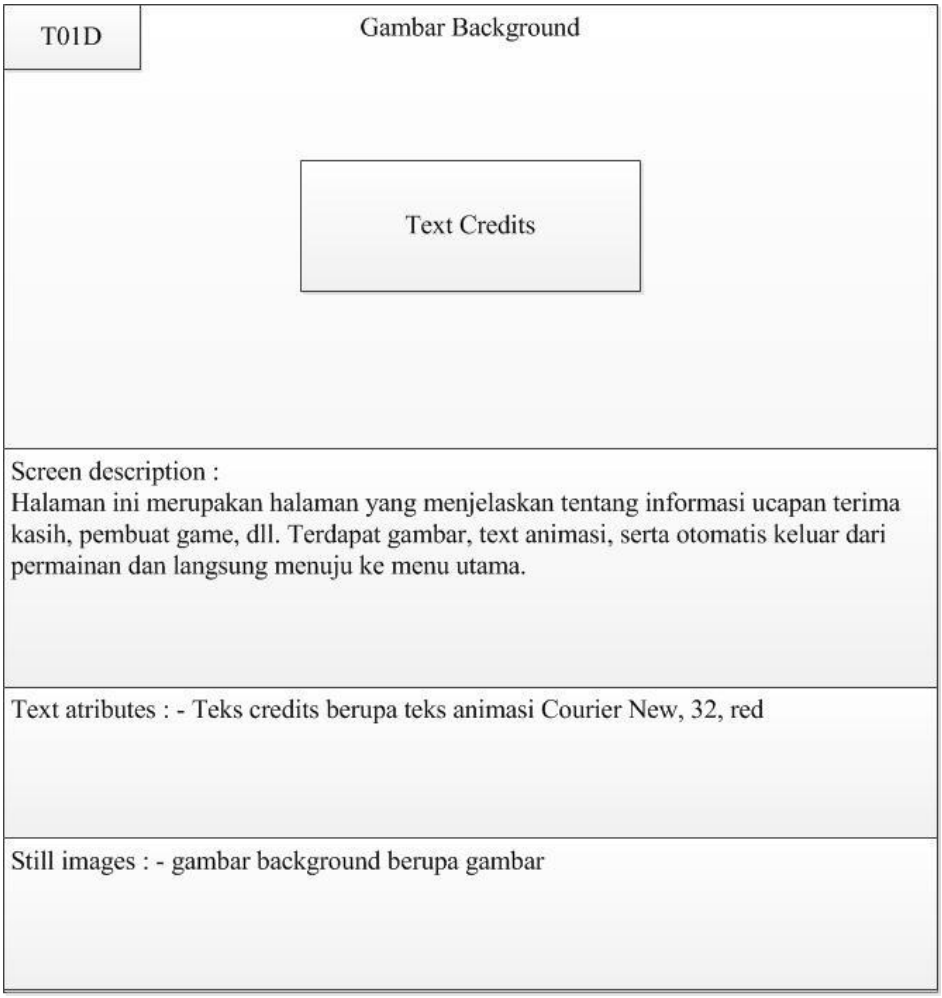
Gambar 3-21 tampilan antarmuka new *game*



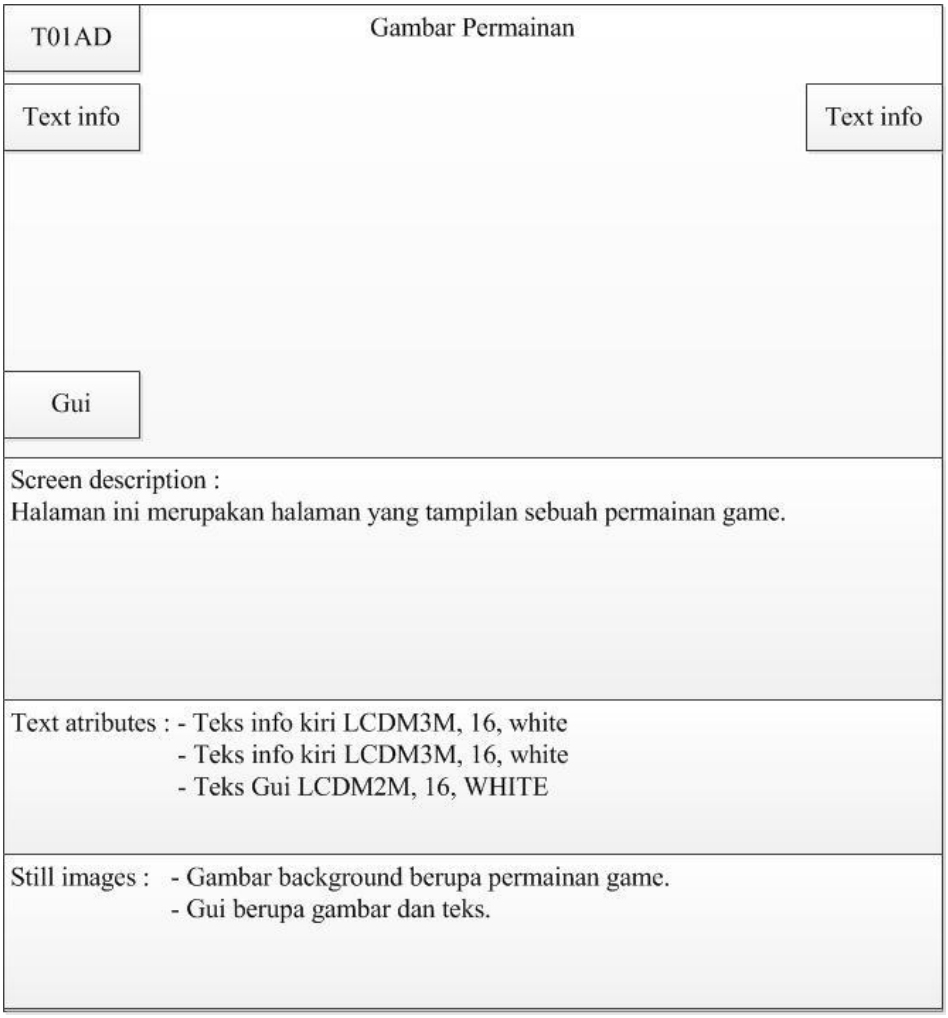
Gambar 3-22 tampilan antar muka load *game*



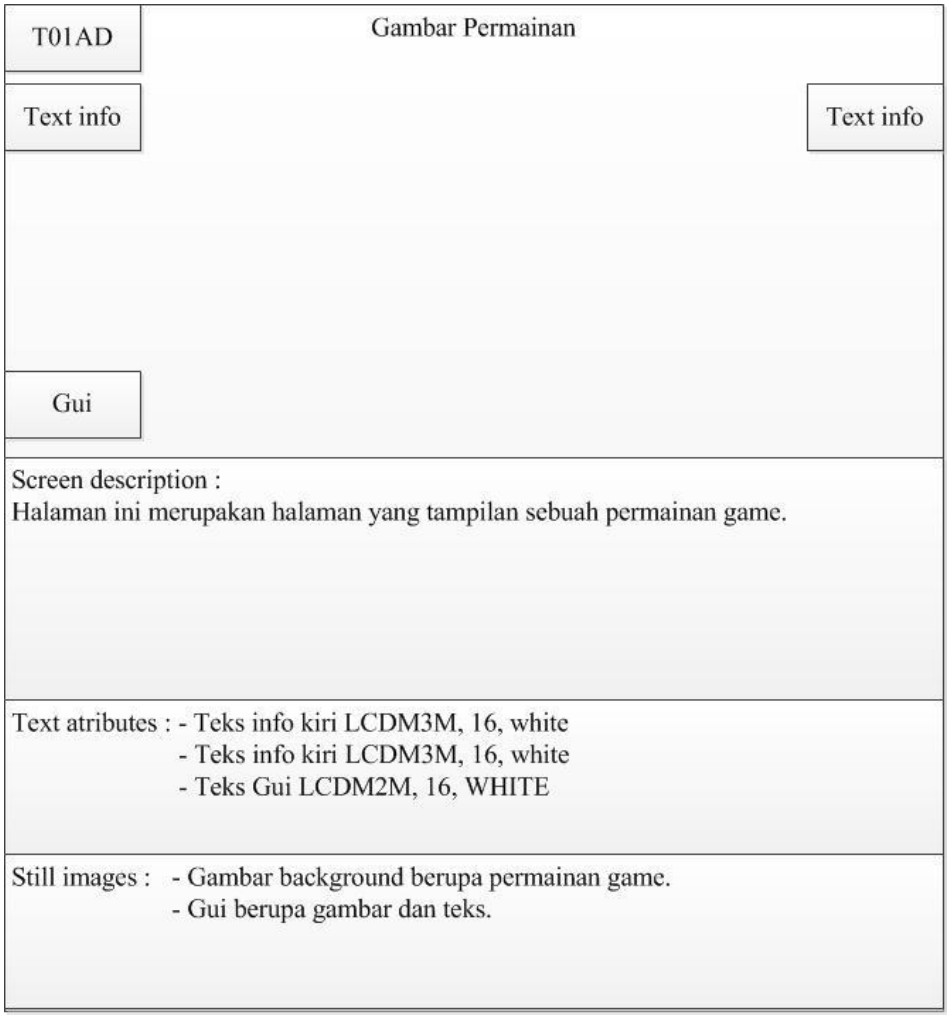
Gambar 3-23 tampilan antar muka menu control



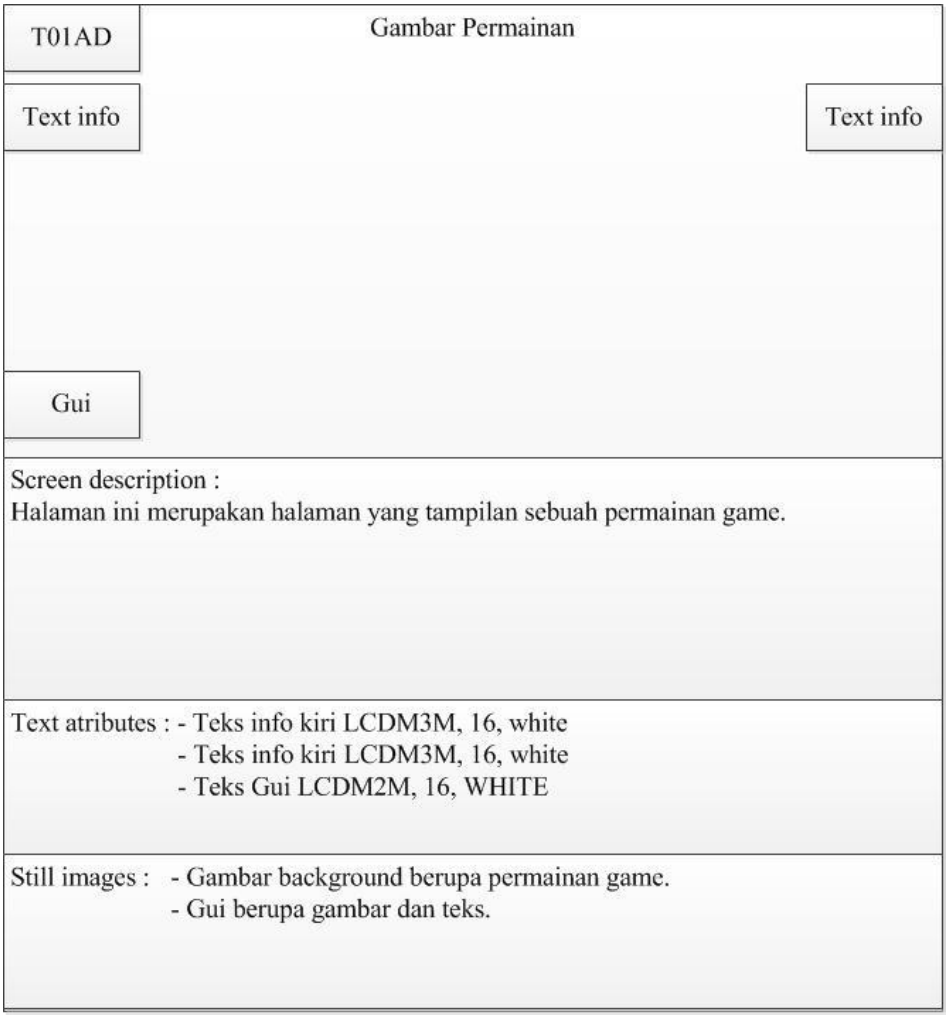
Gambar 3-24 tampilan antar muka menu credits



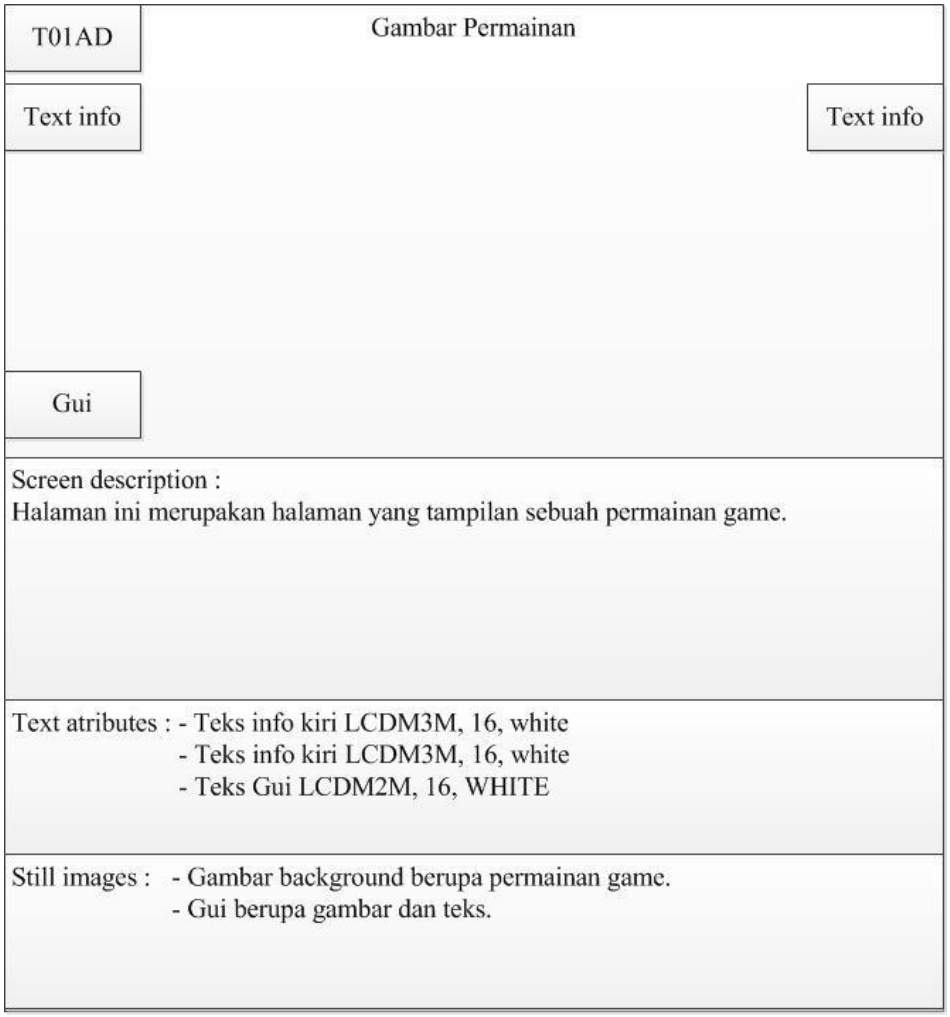
Gambar 3-25 tampilan antar muka *map* 1 (bangunan utama)



Gambar 3-26 tampilan antar muka *map 2* (lantai 1)



Gambar 3-27 tampilan antar muka *map 3* (ruang pasien)



Gambar 3-28 Tampilan antar muka *map 4 (apotik)*

3.2.2 Storyboard



Gambar 3-29 tampilan *map* 1

***Map* :** Gedung utama *"Haunted Hospital"*

Terdiri dari bangunan utama yang menampilkan kekokohan bangunan *"Haunted Hospital"*

Misi :

Di sini si pemain di haruskan menghabisi semua musuh yang berada di sekitar halaman gedung sebelum memasuki bangunan*"Haunted Hospital"*



Gambar 3-30 tampilan *map* 2

***Map* :** Lantai 1

Terdiri dari ruangan – ruangan yang dulunya di fungsikan sebagai ruang utama pada gedung

Misi :

Disini si Pemain harus menghabisi semua hantu yang akan sering muncul untuk menghalangi si karakter menuju *map* selanjutnya, di sini juga si pemain akan dihadapkan dengan penampakan hantu tuyul, pocong dan beberapa potongan tubuh manusia.



Gambar 3-31 tampilan *map* 3

***Map* : Lantai 3 (**Ruangan Pasien)

Terdiri dari ruangan – ruangan yang gelap yang dulunya di gunakan sebagai ruang untuk menginap para pasien

Misi :

Pemain akan di kejutan dengan penampakan semua hantu – hantu yang menghuni gedung *"Haunted Hospital"*. Serta pemain akan dibawa untuk mengetahui apa penyebab semua kejadian ini sebelumnya.



Gambar 3-32 tampilan *map* 4

***Map* :** Apotik

Terdiri dari peralatan–peralatan medis dan beberapa rak – rak yang dulunya di fungsikan untuk menyimpan obat-obatan.

Misi :

Disini si pemain dituntut untuk mengalahkan musuh terkuatnya yaitu Dr. Zombie sebelum mencari sebuah komputer yang menyimpan semua bukti kejahatan yang menyebabkan bangunan rumah sakit tersebut menjadi sangat angker.

3.2.3 Deskripsi karakter

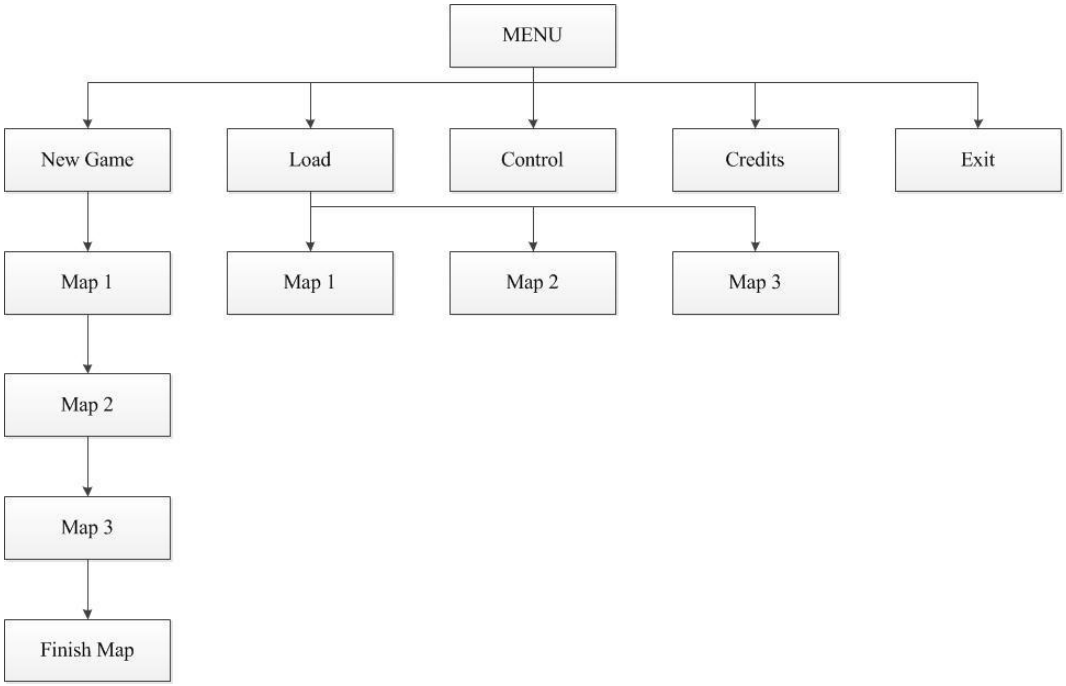
Tabel 3.12 nilai potential field aktraktif

No	Nama NPC	Gambar NPC	Kemampuan NPC	Deskripsi NPC
1	Tuyul Cebol		Menyerang secara bergerombol	Memiliki tubuh serperti bayi, dapat dilu
2	Suster Ngesot		Menyerang menggunakan pisau	Menyerupai wanita muda, namun tidak bisa berjalan dikarenakan kaki-kakinya hancur pada saat kecelakaan
3	Pocong		menyerang dengan cara memukul dengan menggunakan tubuhnya	Menyerupai mayat yang telah di bungkus oleh kain kapan. Bergerak dengan cara melompat.

4	Nyai. Kunti		Menyerang dengan cara mencarkar	Menyerupai wanita muda berambut panjang serta berpenampilan serba panjang.

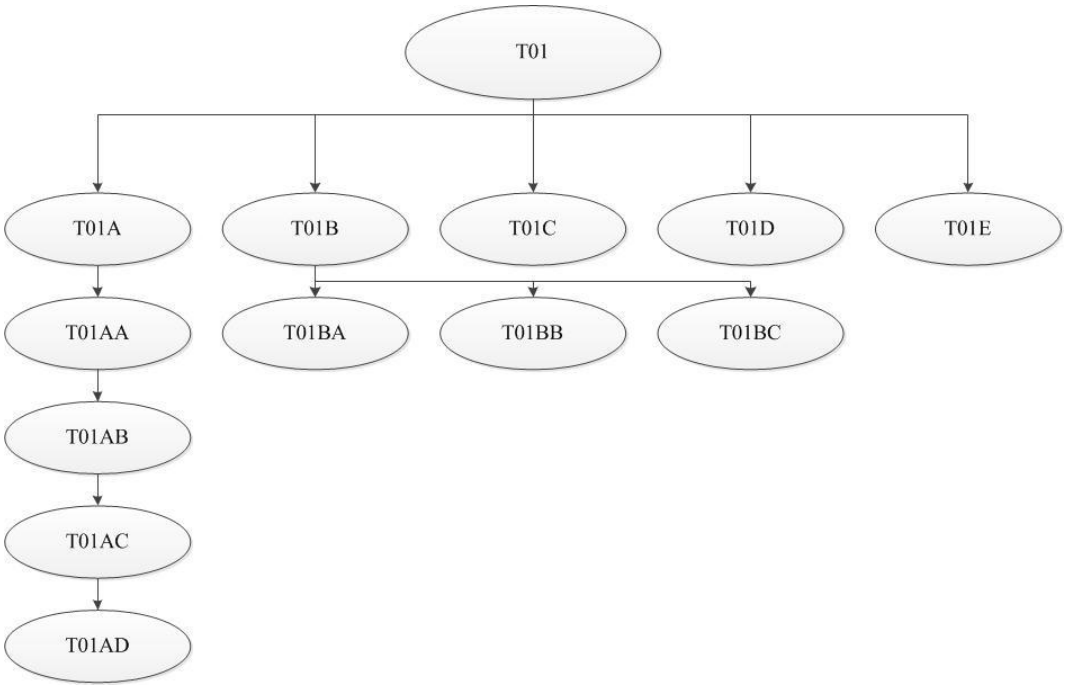
3.2.4 Struktur menu permainan

Berikut pada gambar 3-19 akan digambarkan mengenai struktur menu pada permainan :



Gambar 3-33 struktut menu permainan

3.2.5 Jaringan semantik



Gambar 3-34 jaringan semantik permainan