BAB 3
ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini menjelaskan mengenai analisis dan gambaran dari sistem yang ada dan bagaimana perancangan program aplikasi game yang akan dibangun.

3.1. Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sitem yang utuh kebagian-kebagian komponennya yang dimaksudkan untuk mengidentifikasi dan mengevaluasi segala macam permasalahan dan hambatan apa saja yang bisa terjadi dan kebutuhan-kebutuhan yang apa saja yang diharapkan dapat menjadi acuan untuk diusulkannya perbaikan-perbaikan.

3.1.1. Analisis Masalah

Berdasarkan analisis sistem yang dilakukan dengan melakukan evaluasi terhadap sistem aplikasi game yang akan dibangun. Beberapa masalah yang penulis analisis sebagai latar belakang dalam pembangunan aplikasi game survive from death adalah berikut.

3.1.2. Analisis Scoring

Aplikasi *game Survive from death* ini memiliki *score* yang diterapkan disetiap masing-masing *Level*nya dan tidak berhubungan antara *score* satu *Level* dengan *Level* lainnya

3.1.3. Analisis *Finite State Machine* (FSM)

Finite State Machine adalah suatu metode perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal yaitu *State* (Keadaan), *Event* (Kondisi), dan *Action* (Aksi). Adapun FSM yang terdapat di dalam aplikasi game ini akan di implementasikan pada NPC (*Non Playable Character*) untuk menentukan prilaku atau aksi yang akan dilakukan NPC. Adapun konfigurasi FSM yang akan digunakan untuk menentukan prilaku pada NPC dapat dinyatakan sebagai berikut :

Q = {q1,q2,q3,q4,q1}

∑ = {2,4,8,20}

S = q1

F = {q1}

Dimana pada *game Survive From Death* yang akan dibangun, Q merupakan himpunan state yang teridiri dari aksi-aksi terdapat pada NPC, q1 mewakili keadaan diam, q2 mewakili keadaan mulai mengejar, q3 mewakili keadaan menyerang, dan q4 mewakili keadaan menyerah. ∑ merupakan himpunan symbol input. Dalam hal ini jarak atau parameter dalah simbol input yang akan digunakan. Adapun simbol input itu adalah 2,3,4,8,20. S adalah state awal yaitu q1 dan F adalah state akhir yaitu, {q1}. Dan berikut adalah fungsi transisi dalam perancangan *game survive from death* :

δ (q1,2) = q3	δ (q2,2) = q3	δ (q3,2) = q3
δ (q1,4) = q2	δ (q2,4) = q2	δ (q3,4) = q2
δ (q1,8) = q4	δ (q2,8) = q4	δ (q3,8) = q4
δ (q1,20) = q1	δ (q2,20) = q1	δ (q3,20) = q1

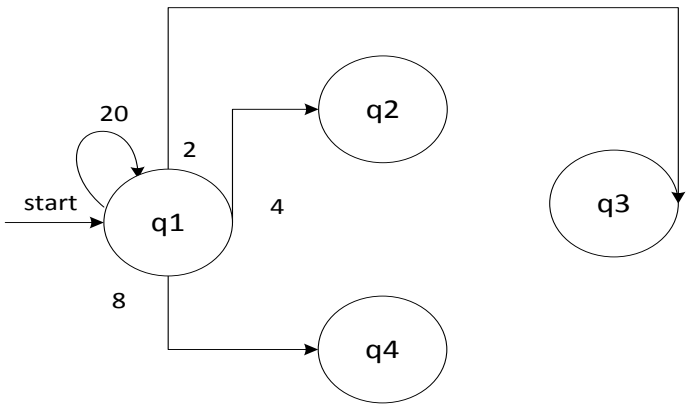
δ (q4,2) = q3

δ (q4,4) = q2

δ (q4,8) = q4

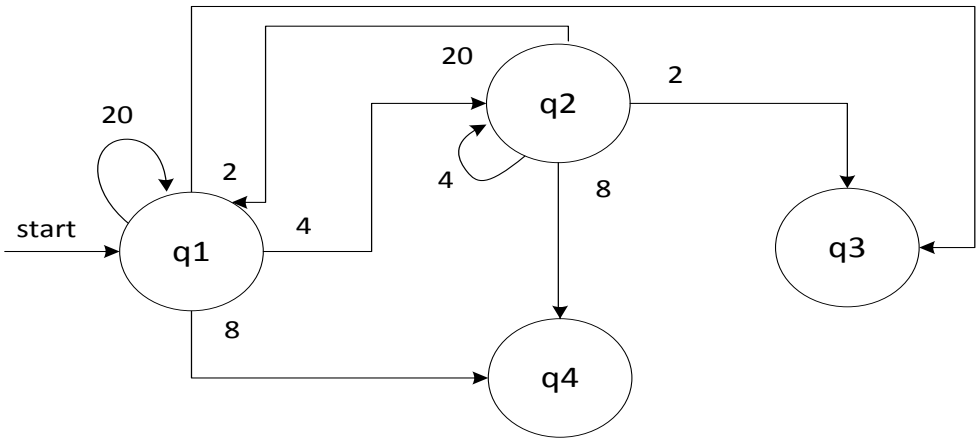
δ (q4,20) = q1

Fungsi transisi di atas kemudian dapat digambarkan dengan menggunakan *Finite State Diagram* (FSD) untuk q1 yang mewakili keadaan diam seperti pada gambar 3.1 berikut :



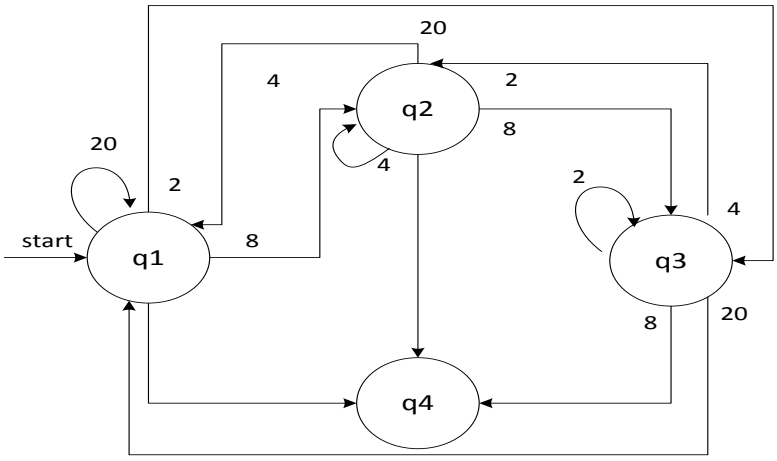
Gambar 3. 1 *Finite State Diagram* q1 untuk NPC

Kemudian q2 yang mewakili keadaan mulai mengejar yang digambarkan menggunakan *Finite State Diagram* (FSD) untuk q2 yang mewakili keadaan mulai mengejar seperti pada gambar 3.2 berikut :



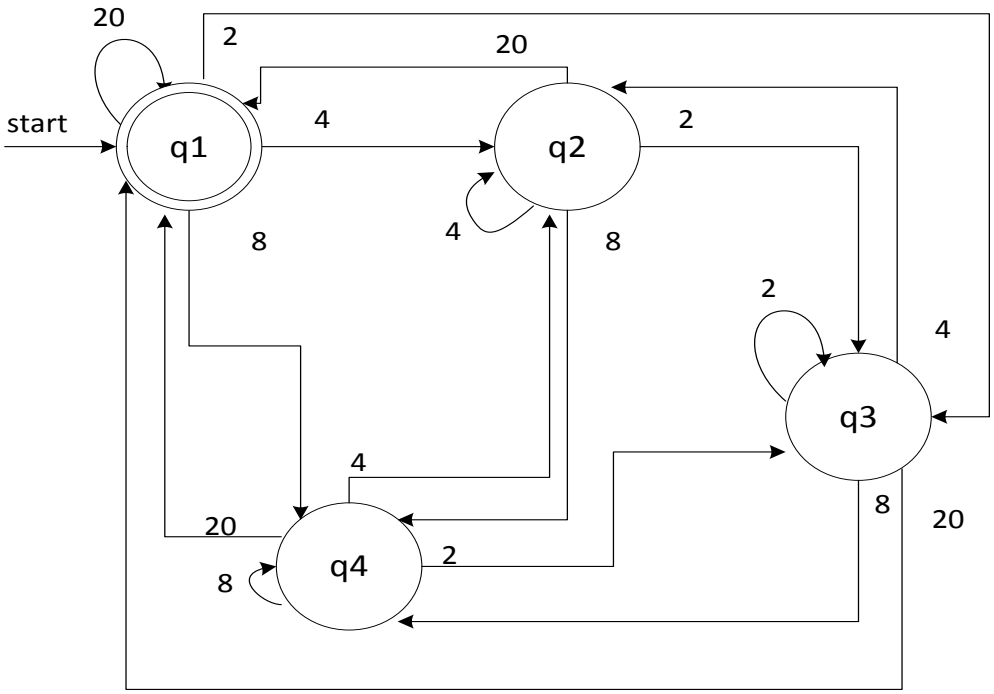
Gambar 3. 2 *Finite State Diagram* q2 untuk NPC

Selanjutnya q3 yang mewakili keadaan menyerang yang digambarkan menggunakan *Finite State Diagram* (FSD) untuk q3 yang mewakili keadaan menyerang seperti pada gambar 3.3 berikut :



Gambar 3. 3 *Finite State Diagram* q3 untuk NPC

Selanjutnya q4 yang mewakili keadaan menyerah dan gambaran seluruh fungsi transisi yang digambarkan menggunakan *Finite State Diagram* (FSD) seperti pada gambar 3.4 berikut :



Gambar 3. 4 *Finite State Diagram* untuk NPC pada *Survive From Death*

Lingkaran pada FSD menyatakan *state.* Dimana setiap *state* mewakili aksi NPC pada permainan. Lingkaran bergaris tunggal brarti *state* sementara dan lingkaran bergaris ganda berarti *state* akhir. Lingkaran diberi label sesuai dengan nama *state* tersebut. Anak panah menyatakan transisi yang terjadi. Label yang terdapat di anak panah menyatakan *event* yang membuat transisi dari suatu *state* ke *state* lain. Satu anak panah diberikan label *start* untuk menyatakan awal mula transisi dilakukan.

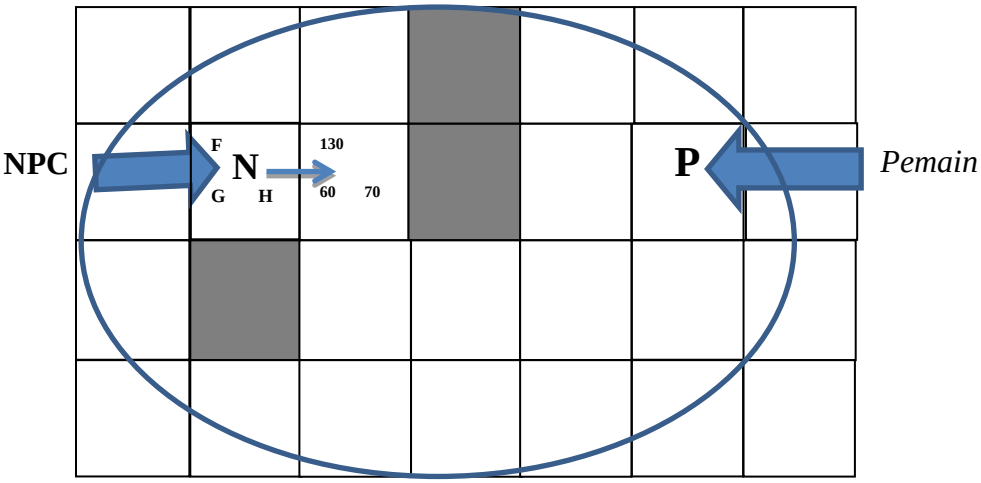
3.1.4. Analisis Algoritma A* (*A Star)*

Algoritma A* yang terdapat di dalam aplikasi game akan di implementasikan pada NPC (*Non Playable Character*) untuk menentukan jalur terpendek atau *pathfinding* menuju *target* yang telah ditentukan. Adapun istilah-istilah yang akan dibahas yaitu path, open list, closed list, nilai f,g dan n.

Algoritma A* menggunakan dua *list* yaitu *Open* dan *Closed. Open* adalah *list* yang digunakan untuk menyimpan simpul-simpul yang pernah dibangkitkan nilai heuristiknya telah dihitung tetapi belum terpilih sebagai simpul terbaik (*best* node). *Closed* adalah *list* untuk menyimpan simpul-simpul yang sudah perbnah dibangkitkan dan sudah pernah terpilih sebagai simpul terbaik.

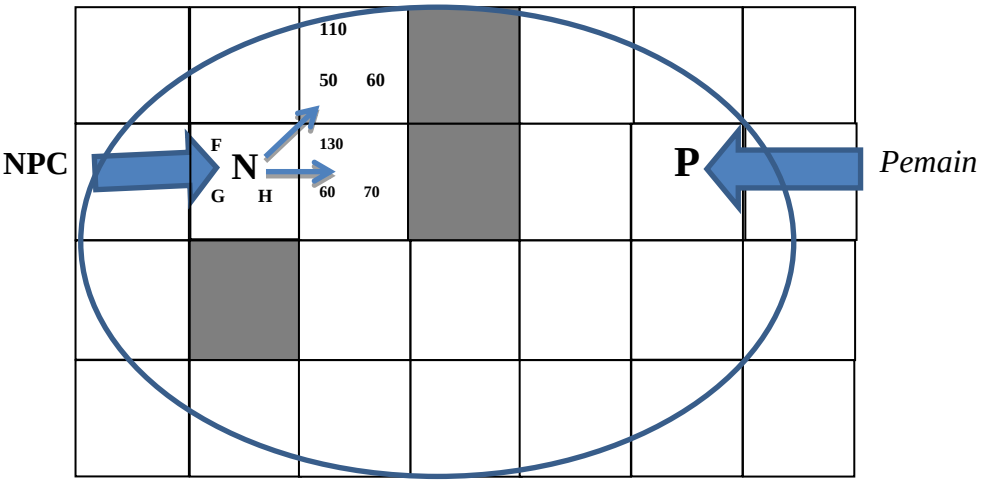
Berikut adalah penjelasan AI yang digunakan musuh dalam pengejaran pemain, dalam pengujian ini asumsi bahwa pemain terdeteksi oleh NPC dan pemain dalam keadaan diam ditempat. Dalam pengejaran ini, NPC menggunakan Alggoritma A* untuk pencarian rute terdekat agar pemain dapat diserang.

Apabila *target* telah keluar ataupun berada di luar dari parameter deteksi, maka NPC akan menghentikan pencarian terhadap target.Adapun cara kerja algoritma A* pada NPC didalam game lebih lengkapnya sebagai berikut.



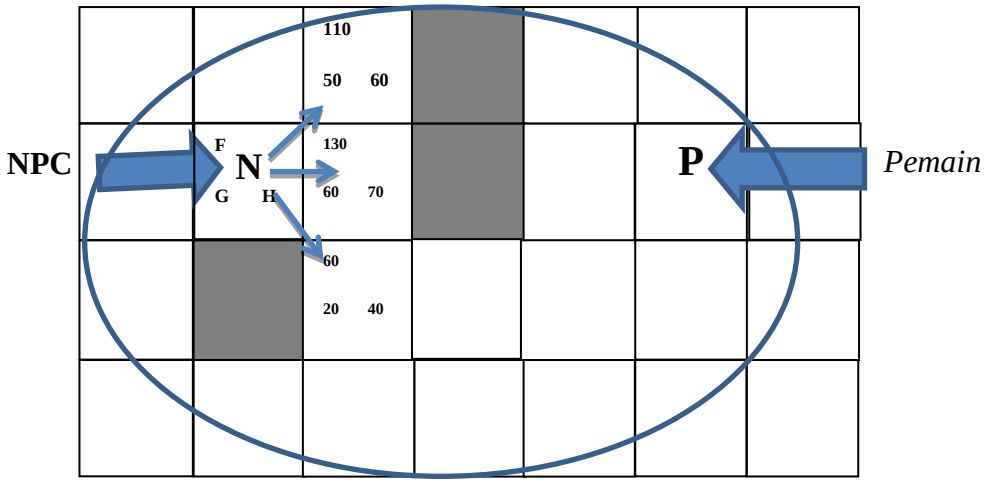
Gambar 3. 5 Pencarian Menggunakan A* Tahap 1

Dari NPC mencari secara *horizontal* dengan nilai G=60, dan H=70 maka dengan rumus F=G+H, dihasilkan nilai F= 130



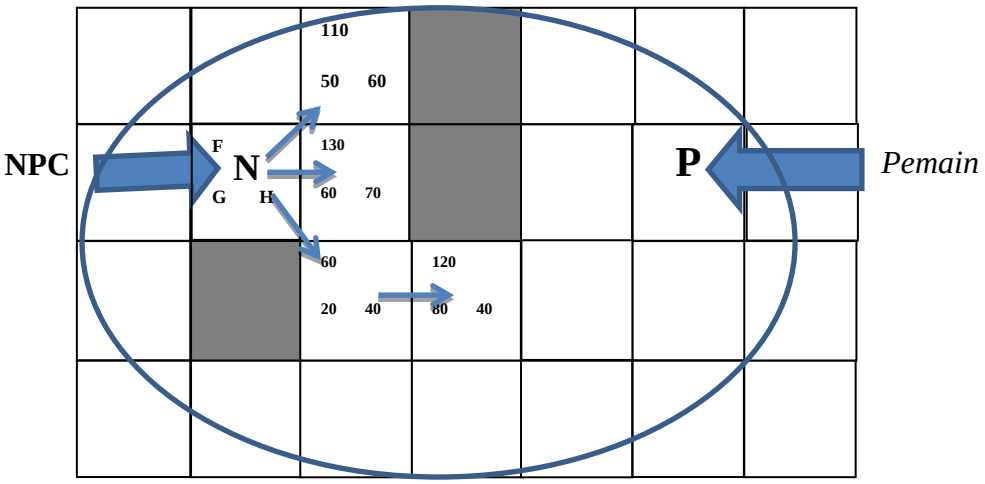
Gambar 3. 6 Pencarian Menggunakan A* Tahap 2

Dari NPC mencari secara diagonal ke atas dengan nilai G=50, dan H=60 maka dengan rumus F=G+H, dihasilkan nilai F=110



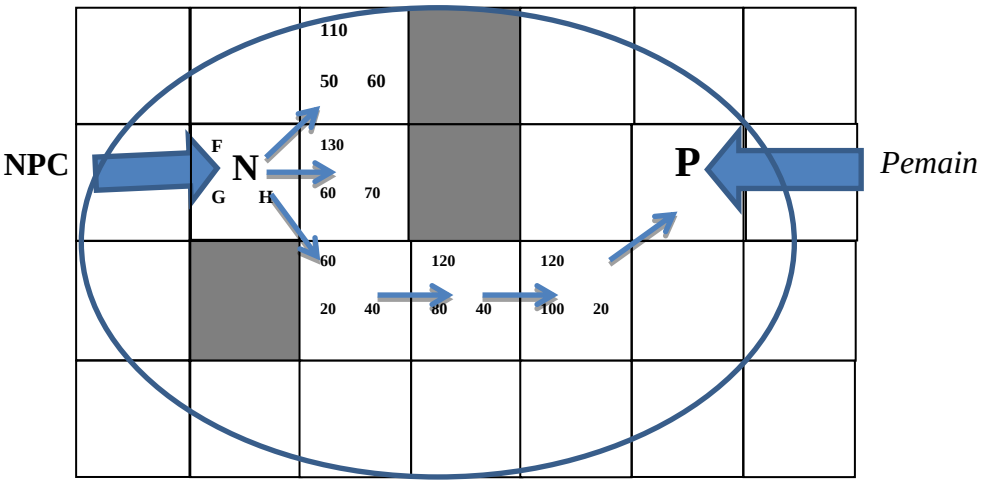
Gambar 3. 7 Pencarian Menggunakan A* Tahap 3

Dari NPC mencari secara diagonal ke bawah dengan nilai G=20, dan H=40 maka dengan rumus F=G+H, dihasilkan nilai F=60



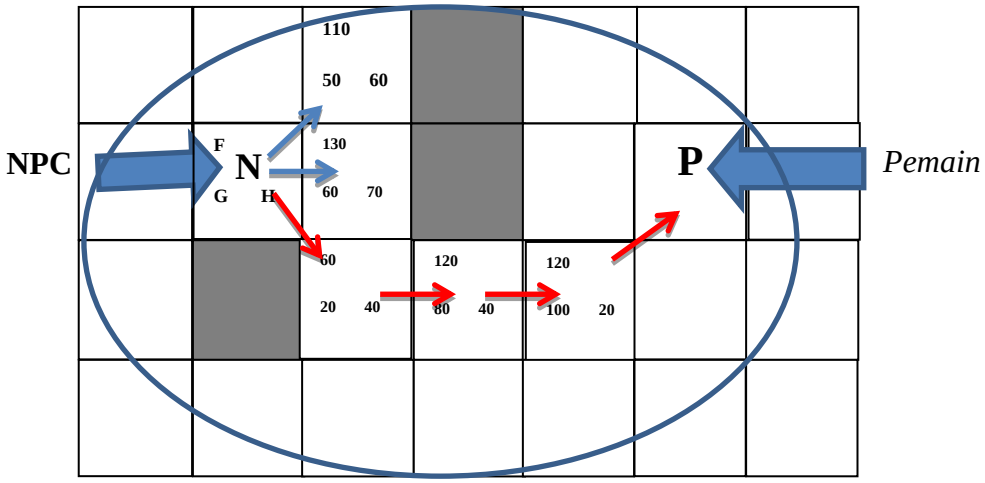
Gambar 3. 8 Pencarian menggunakan A* Tahap 4

Kemudian dari node yang benilai F=60, mencari secara Horizontal dengan nilai G =80 yang didapatkan dari tahap 1 ditambah dengan nilai G posisi diagonal yaitu 20, dan H=40 maka dengan rumus F=G+H, dihasilkanlah F=120



Gambar 3. 9 Pencarian Menggunakan A*Tahap 5

Kemudian dari node yang bernilai F=120, mencari secara horizontal dengan nilai G= 100 yang didapat dari tahap 4 ditambah dengan nilai G diagonal yaitu 20, dan H=20 maka dengan rumus F=G+H, dihasilkan nilai F=120.



Gambar 3. 10 Pencarian Menggunakan A* Tahap 6

Maka dihasilkan jalur optimum untuk pencarian NPC menuju Pemain menggunakan algoritma A*. Dengan rute tersebut NPC akan melakukan pengejaran terhadap karakter pemain.

3.1.5. Penerapan Artificial Intelligence pada Game

AI diterapkan pada musuh yaitu laba-laba dan zombie. Fungsi dari *Artificial Intelligence* yang diterapkan pada musuh ini adalah sebagai pathfinding atau untuk menentukan rute terpendek mengejar pemain dengan menggunakan algoritma A*.Adapun untuk menentukan kondisi atau prilaku pada musuh kapan untuk menyerang serta kapan untuk berhenti menyerang menggunakan FSM. Musuh diberikan suatu parameter atau jarak yang digunakan untuk menentukan prilaku atau kondisi musuh untuk menyerang dan mengejar pemain. Apabila pemain sudah ada dalam jarang pengejaran maka musuh ini akan mulai mengejar target dan akan berhenti pada jarang jangkauan yang telah ditentukan. Begitu pula apabila pemain sudah ada dalam jarak serang maka musuh akan menyerang sesuai kemampuan yang dimiliki.

3.1.6. Deskripsi Aplikasi Game

1. Pengenalan Game Survive from Death

Game Survive From death yang akan dibangun ini bergenre First Person Shooter yaitu salah satu jenis game (genre) yang digemari menggunakan sudut pandang orang pertama di mana pemain seolah-olah menjadi karakter utama dalam game yang berpusat pada permainan di sekitar senjata-senjata dan peluru tempur.

Pada game Survive From Death ini bahasa komunikasi nantinya akan menggunakan bahasa Indonesia baik teks maupun suara dan nantinya akan terdapat 4 *Level* permainan diawali dengan intro dan dilanjutkan kepertualangan player untuk mencapai misi nya yaitu bertahan hidup hingga akhir. Setiap *Level*nya memiliki tingkat kesulitan yang berbeda dimana disetiap *Level*nya pemain dibuat kebingungan serta berfikir untuk menemukan jalan keluar untuk melanjutkan ke *Level* berikutnya. Tidak seperti pada game Call of Duty dimana player dan musuh tidak seimbang, dimana lebih kuat karakter dibanding musuh yang ada namun pada game ini musuh akan dibuat seimbang dan agak sulit dilumpuhkan begitu juga musuh di*Level-Level* selanjutnya akan jauh lebih sulit. Pada game Doom hanya ada

senjata mesin namun pada game ini akan terdapat beberapa senjata unik yaitu senjata tradisional yang tidak akan dapat dijumpai di game genre fps lainnya. Musuh pada game ini tidak seperti House of The Dead yaitu hanya zombie dan monster saja, pada game ini terdapat 3 tipe musuh yaitu manusia zombie dan monster. Game ini juga dibumbui dengan animasi atau cinematik sebagai penguat alur cerita seperti pada game Call of Duty modern warfare sehingga membuat permainan lebih menarik dan tidak membosankan.

AI pada game ini yaitu FSM digunakan untuk memodelkan prilaku dari NPC dan algoritma A* digunakan untuk bagaimana cara NPC menemukan atau mengidentifikasi keberadaan player.

2. Storyline

Jalan cerita game ini diawali dari sang karakter utama / player yang terjebak disuatu pulau yang berada suatu pulau yang berada di indonesia dimana pulau tersebut adalah bekas tempat penelitian yang sudah terbengkalai dikarenakan kegagalan sistem dan objek penelitian yang mengamuk serta lepas kendali. Disini player harus berusaha bertahan hidup dan menemukan cara keluar dari pulau tersebut. Adapun misi utama yang harus dicapai player yaitu pergi ke bagian tengah pulau dimana terdapat tempat bekas penelitian yang terbengkalai dan goa untuk dapat keluar dari pulau tersebut dimana pulau tersebut telah dikelilingi berbagai objek hasil dari penelitian yang lepas kendali.

Pada *Level* pertama akan dibuka dengan intro, yaitu cerita awal mula player terjebak dipulau dan dilanjutkan dengan pertualangan karakter untuk mencari info bagaimana cara keluar dari pulau tersebut. Pada *Level* awal ini player akan berada disuatu perkampungan dimana terdapat rumah adat Sumatra utara yang sudah tidak berpenduduk dan player akan dihadapkan dengan musuh yaitu para laba – laba yang sudah terinfeksi dan bermutasi hasil dari penelitian di pulau tersebut serta hanya bermodalkan senjata kapak , disini player ditugaskan mencari tombol untuk membuka jalan ke tempat selanjutnya.

Di *Level* kedua player sudah berada jauh ke bagian dalam pulau dimana player akan berada disuatu goa yang telah dibangun jauh dan memiliki arsitektur yang cukup aneh disini player akan menemukan dan dapat memakai senjata baru yaitu pistol untuk melawan musuh dan senter untuk penerangan selama berada didalam goa. Musuh di *Level* kedua ini akan lebih sulit dibanding yang sebelumnya yaitu monster-monster hasil penelitian yang lepas kendali yaiut para zombie yang tidak lain adalah penduduk pulau yang terkena virus hasil penelitian dari fasilitas yang ada di pulau tersebut. Selain itu player akan dibuat kebingunan untuk menemukan jalan keluar dai gua tersebut karena terdapa alat teleportasi yang dapat memindahkan player secara random dan disisni player dituntut untuk dapat mencari teleportasi tabg benar dengan menemukan info atau petunjuk yang akan menuntun player agar dapat segera menyelesaikan misi dan keluar dari tempat tersebut.

Di *Level* ketiga player akan sampai disebuah fasilitas yang ternyata terdapat dulunya adalah perkampungan penduduk asli pulau dimana terdapat bangunan dan rumah adat yang berada di tempat tersebut. Pada *Level* ini player dituntut untuk mencari info dimana terdapat info bahwa ada sebuah alat transportasi di balik gua selanjutnya dimana terdapat teka-teki atau puzel untuk membuka jalan ke gua tersebut. Selain itu karakter juga akan dihadang dengan musuh yang jauh lebih sulit, dimana musuh akan lebih kuat menerima serangan-serangan dari player dan tidak mudah untuk dilumpuhkan.

Di*Level* terakhir player berada kembali disuatu gua namun berbeda daru gua sebelumnya disni player dinuntut untuk menemukan tempat terakhir dimana terdapat alat transprortasi untuk keluar dari pulau tersebut. Disini karakter akan dituntut untuk menemukan alat teleportasi yang benar yang mengarah ke tempat tujuan terakhir dimana alat teleportasi akan lebih sulit karena di kombinasikan dengan puzzle seperti pada *Level* 1 dan 2 serta dihadapkan melawan boss musuh yang sangat kuat, tebal serta sulit dikalahkan inilah rintangan terakhir karakter sebelum akhirnya dapat keluar dari pulau tersebut dengan selamat menggunakan helicopter.

3. *Gameplay*

Dalam *Game Survive From Death*, Pemain diharuskan untuk menghancurkan musuh yang ada di dalam *game* untuk dapat bertahan hidup guna melanjutkan kelevel berikutmya. Dimana disetiap *level*nya terdapat 2 jenis musuh yang berbeda.

Setiap musuh yang dihadapi memiliki cara yang berbeda untuk menyerang *Player*, berjalan mengejar *Player*, daya tahan terhadap serangan dari *Player*.

Tabel 3. 1 Penjelasan Musuh Pada Permainan

Model Karakter	Kemampuan
	Pada musuh ini, laba-laba mempunyai kemampuan untuk meloncat menyerang jika Pemain sudah berada dijarak jangkauan serangan laba-laba. Apabila Pemain menjauh dan sudah tidak ada pada jarang jangkauan laba-laba. Maka laba-laba akan berhenti menyerang dan mengejar target.
	Pada musuh ini, Zombie mempunyai kemampuan untuk menyerang dengan menerkam Pemain dengan tangan dan mulutnya jika Pemain sudah berada dijarak jangkauan serangan Zombie. Apabila Pemain menjauh dan sudah tidak ada pada jarang jangkauan Zombie. Maka Zombie akan berhenti

	menyerang dan mengejar target. Zombie memiliki daya tahan lebih tinggi dibandingkan laba-laba.

3.1.7. Analisis Kebutuhan Non-Fungsional

Analisis dan kebutuhan non-fungsional meliputi analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, analisis kebutuhan user, serta sasaran user dan kegunaan aplikasi game.

3.1.7.1. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perangkat yang dibutuhkan untuk mengaktifkan dan mengoptimalkan kinerja perangat keras. Adapun perangkat lunak yang digunakan adalah sebagai berikut :

Perangkat lunak yang dibutuhkan pengembang untuk membangun aplikasi ini adalah sebagai berikut :

1. Sistem Operasi Windows XP SP2.
2. Unity 4.0.1.
3. Autodesk 3ds Max.
4. Adobe Photoshop
5. DirectX 9

Perangkat lunak yang dibutuhkan pengguna untuk dapat menjalankan aplikasi ini adalah sebagai berikut :

1. Sistem Operasi Windows XP SP2.
2. DirectX 9

3.1.7.2. Analisis 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. Adapun perangkat keras yang digunakan untuk membangun aplikasi game ini adalah sebagai berikut :

Perangkat keras yang digunakan oleh pengembang untuk membangun aplikasi ini adalah sebagai berikut :

1. Processor 3,00Ghz
2. RAM : 4 GB.
3. VGA : 2 GB.
4. Harddisk 60 GB.
5. Monitor.
6. *Mouse* dan *Keyboard*.
7. Speaker.

Perangkat keras yang direkomendasikan bagi pengguna untuk dapat menjalankan aplikasi ini adalah sebagai berikut :

1. Processor 1,65 Ghz.
2. RAM : 1 GB.
3. VGA : 512 MB.
4. Harddisk 1 GB.
5. Monitor.
6. *Mouse* dan *Keyboard*.
7. Speaker

3.1.7.3. Analisis Kebutuhan User

Selain dibutuhkannya perangkat lunak dan perangkat keras, *user* juga sangat dibutuhkan dalam penggunakaan aplikasi game ini. Adapun spesifikasi *user* yang dibutuhkan:

1. Mengerti dalam mengoperasikan aplikasi.
2. Dapat menggunakan *mouse* dan *keyboard*.

3.1.7.4. Sasaran User dan Kegunaan

Aplikasi game yang akan dibangun ditujukan untuk user berusia 15 tahun dan kegunaan dari aplikasi game ini yaitu sebagai sarana hiburan serta melatih pola pikir user dalam menentukan strategi untuk menyelesaikan permainan serta dengan disisipkannya unsur kebudayaan didalam game diharapkan dapat menambah ilmu pengetahuan terhadap kebudayaan Indonesia salah satunya berupa pengetahuan tentang rumah adat.

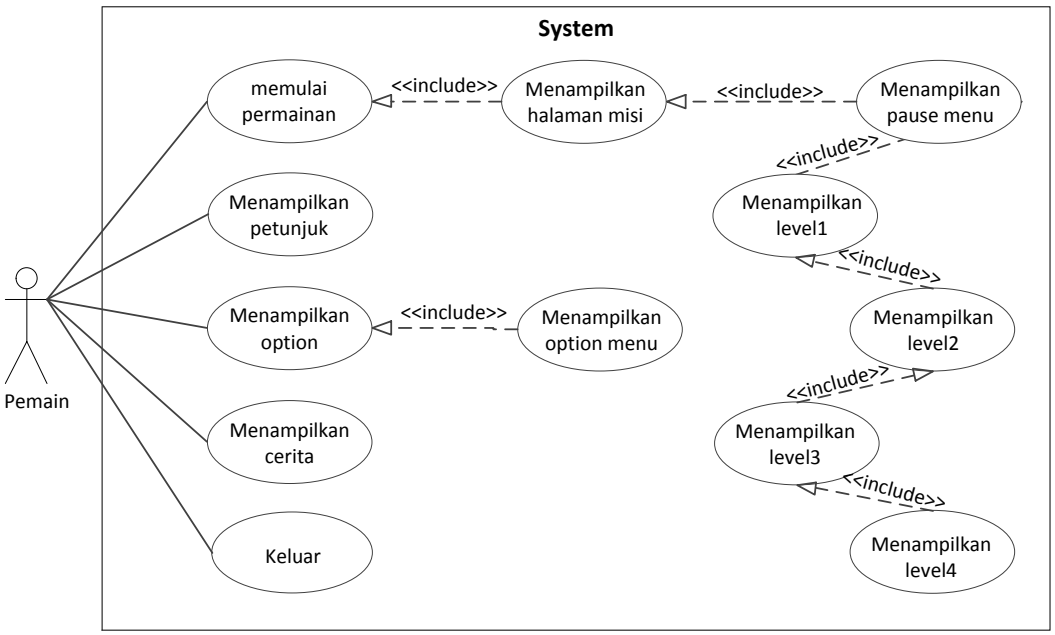
3.1.8. Analisis Kebutuhan Fungsional

3.1.8.1. Identifikasi Aktor

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

3.1.8.2. Use Case Diagram

Diagram *Use Case* menggambarkan hubungan antara aktor dan aplikasi yang dibangun. Berikut ini adalah perancangan proses-proses yang terdapat pada game yang akan dibangun, yang digambarkan dengan *Use Case* Diagram pada Gambar 3.12.



Gambar 3. 11 Diagram *Use Case* Game Survive From Death

3.1.8.3. Definisi Use Case

Definisi *Use Case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *Use Case.*

Tabel 3. 2 Definisi Use Case

No	***Use Case***	**Deskripsi**
1	Mulai	Proses untuk memulai permainan
2	Petunjuk	Proses untuk melihat petunjuk cara bermain
3	Option	Proses untuk melihat option permainan
4	Option menu	Proses untuk melihat menu option permainan
5	Cerita	Proses untuk melihat cerita permainan
6	Halaman misi	Proses untuk melihat halaman misi permainan
7	Pause menu	Proses untuk mengistirahatkan permainan
8	*Level*1	Tingkatan *Level* 1
9	*Level*2	Tingkatan *Level* 2

10	*Level*3	Tingkatan *Level* 3
11	*Level*4	Tingkatan *Level* 4
12	Keluar	Proses untuk menutup permainan

3.1.8.4. Use Case Scenario

Skenario *Use Case* menunjukkan proses apa yang terjadi pada setiap bagian didalam *Use Case* tersebut, dimana *user* memberikan perintah pada sistem dan sistem akan merespon sesuai perintah yang diminta oleh *user*.

1. Skenario *Use Case* Memulai Permainan

Tabel 3. 3 Skenario *Use Case* Memulai Permainan

Identifikasi		
Nomor	1	
Nama	Mulai	
Tujuan	Memulai permainan	
Deskripsi	Proses untuk memulai permainan	
Aktor	Pemain/Pengguna	
Skenario Utama		
Kondisi Awal	Pemain berada di menu awal/utama permainan	
Aksi Aktor		**Reaksi Sistem**
Memilih menu Main		Masuk ke sistem dan menampilkan halaman misi permainan
Kondisi Akhir	Menampilkan halaman misi permainan	

2. Skenario *Use Case* Petunjuk Permainan

Tabel 3. 4 *Use Case* Petunjuk Permainan

Identifikasi		
Nomor	2	
Nama	Petunjuk	
Tujuan	Melihat petunjuk cara penggunaan aplikasi game	
Deskripsi	Proses untuk melihat petunjuk cara dan tips memainkan permainan	
Aktor	Pemain/Pengguna	
Skenario Utama		
Kondisi Awal	Pemain berada di menu utama permainan	
Aksi Aktor		**Reaksi Sistem**
Memilih menu Petunjuk		Menampilkan petunjuk permainan
Memilih menu kembali		Menutup petunjuk permainan dan Menampilkan menu utama
Kondisi Akhir	Menampilkan layar petunjuk permainan	

3. Skenario *Use Case* Option Game

Tabel 3. 5 Skenario *Use Case* Option Game

Identifikasi		
Nomor	3	
Nama	Option	
Tujuan	Melihat menu option aplikasi game	
Deskripsi	Proses untuk melihat option permainan	
Aktor	Pemain/Pengguna	
Skenario Utama		
Kondisi Awal	Pemain berada di menu utama permainan	
Aksi Aktor		**Reaksi Sistem**

Memilih menu option	Menampilkan option menu game
Memilih menu kembali	Menutup halaman option
Kondisi Akhir	Menampilkan layar option menu game

4. Skenario *Use Case* Option Menu

Tabel 3. 6 Skenario *Use Case* Option Menu

Identifikasi	
Nomor	4
Nama	Option Menu
Tujuan	Melihat option menu pada aplikasi game
Deskripsi	Proses untuk melihat option menu permainan
Aktor	Pemain/Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di option permainan
Aksi Aktor	**Reaksi Sistem**
Memilih menu volume	Menampilkan konfigurasi volume suara permainan
Memilih menu grafik	Menampilkan konfigurasi grafik permainan
Memilih menu Status	Menampilkan Status Permainan
Memilih menu kembali di layar option menu	Menutup halaman option menu
Kondisi Akhir	Menampilkan layar option menu permainan

5. Skenario *Use Case* Cerita

Tabel 3. 7 Skenario *Use Case* Cerita

Identifikasi	
Nomor	5

Nama	Cerita
Tujuan	Melihat cerita tentang permainan
Deskripsi	Proses untuk melihat informasi cerita permainan
Aktor	Pemain/Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama permainan
Aksi Aktor	**Reaksi Sistem**
Memilih menu cerita	Menampilkan layar cerita permainan
Memilih menu kembali	Menampilkan layar menu utama
Kondisi Akhir	Menampilkan layar cerita permainan

6. Skenario *Use Case* Halaman Misi

Tabel 3. 8 Skenario *Use Case* Halaman Misi

Identifikasi	
Nomor	6
Nama	Halaman misi
Tujuan	Melihat halaman misi permainan
Deskripsi	Proses untuk melihat misi permainan
Aktor	Pemain/Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama permainan
Aksi Aktor	**Reaksi Sistem**
Memilih menu mulai	Masuk ke sistem permainan dan menampilkan halaman misi
Memilih menu lewati	Menggenerate Permainan *Level* 1 dan Menampilkan layar pause menu
Kondisi Akhir	Menampilkan layar pause menu

7. Skenario *Use Case* Pause Menu

Tabel 3. 9 Skenario *Use Case* Pause Menu

Identifikasi		
Nomor	7	
Nama	Pause menu	
Tujuan	Melihat pause menu pada permainan	
Deskripsi	Proses untuk melihat pause menu pada permainan	
Aktor	Pemain/Pengguna	
Skenario Utama		
Kondisi Awal	Pemain berada halaman misi permainan	
Aksi Aktor		**Reaksi Sistem**
Memilih menu mulai main		Sistem memulai permainan
Memilih menu keluar		Menampilkan layar menu utama
Kondisi Akhir	Menampilkan layar pause menu permainan	

2. Skenario *Use Case Level*1

Tabel 3. 10 Skenario *Use Case Level*1

Identifikasi		
Nomor	8	
Nama	*Level*1	
Tujuan	Melihat *Level* 1 permainan	
Deskripsi	Proses untuk menampilkan *Level* 1 permainan	
Aktor	Pemain/Pengguna	
Skenario Utama		
Kondisi Awal	Pemain berada di pause menu permainan	
Aksi Aktor		**Reaksi Sistem**
Memilih menu main		Sistem menampilkan *Level* 1

	permainan
Menekan tombol *keyboard* ‘W’	Menggerakkan karakter pemain ke depan
Menekan tombol *keyboard* ‘A’	Menggerakkan karakter pemain ke kiri
Menekan tombol *keyboard* ‘S’	Menggerakkan karakter pemain ke belakang
Menekan tombol *keyboard* ‘D’	Menggerakkan karakter pemain ke kanan
Menekan *mouse* ‘Klik Kiri’	Menggerakkan karaktek pemain untuk melakukan serangan
Menggerakkan *mouse* ke kiri	Menggerakkan pandangan karakter pemain ke kiri
Menggerakkan *mouse* ke kanan	Menggerakkan pandangan karakter pemain ke kanan
Menekan tombol *keyboard* ‘Space’	Menggerakkan karakter pemain untuk melompat
Menekan tombol *keyboard* ‘1’	Karakter menggunakan senjata utama
Menekan tombol *keyboard* ‘2’	Karakter menggunakan senjata kedua
Menekan tombol *keyboard* ‘*Esc*’	Menampilkan layar pause menu
Kondisi Akhir	Menampilkan *Level* 1 permainan

3. Skenario *Use Case Level*2

Tabel 3. 11 Skenario *Use Case Level*2

Identifikasi	
Nomor	9
Nama	*Level*2

Tujuan	Melihat *Level* 2 permainan	
Deskripsi	Proses untuk menampilkan *Level* 2 permainan	
Aktor	Pemain/Pengguna	
Skenario Utama		
Kondisi Awal	Pemain berada di pause menu permainan	
Aksi Aktor		**Reaksi Sistem**
Memilih menu main		Sistem menampilkan *Level* 2 permainan
Menekan tombol *keyboard* ‘W’		Menggerakkan karakter pemain ke depan
Menekan tombol *keyboard* ‘A’		Menggerakkan karakter pemain ke kiri
Menekan tombol *keyboard* ‘S’		Menggerakkan karakter pemain ke belakang
Menekan tombol *keyboard* ‘D’		Menggerakkan karakter pemain ke kanan
Menekan *mouse* ‘Klik Kiri’		Menggerakkan karaktek pemain untuk melakukan serangan
Menggerakkan *mouse* ke kiri		Menggerakkan pandangan karakter pemain ke kiri
Menggerakkan *mouse* ke kanan		Menggerakkan pandangan karakter pemain ke kanan
Menekan tombol *keyboard* ‘Space’		Menggerakkan karakter pemain untuk melompat
Menekan tombol *keyboard* ‘1’		Karakter menggunakan senjata utama
Menekan tombol *keyboard* ‘2’		Karakter menggunakan senjata kedua
Menekan tombol *keyboard* ‘*Esc*’		Menampilkan layar pause menu
Kondisi Akhir	Menampilkan *Level* 2 permainan	

4. Skenario *Use Case Level*3

Tabel 3. 12 Skenario *Use Case Level*3

Identifikasi	
Nomor	10
Nama	*Level*3
Tujuan	Melihat *Level* 3 permainan
Deskripsi	Proses untuk menampilkan *Level* 3 permainan
Aktor	Pemain/Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di pause menu permainan

Aksi Aktor	**Reaksi Sistem**
Memilih menu main	Sistem menampilkan *Level* 3 permainan
Menekan tombol *keyboard* ‘W’	Menggerakkan karakter pemain ke depan
Menekan tombol *keyboard* ‘A’	Menggerakkan karakter pemain ke kiri
Menekan tombol *keyboard* ‘S’	Menggerakkan karakter pemain ke belakang
Menekan tombol *keyboard* ‘D’	Menggerakkan karakter pemain ke kanan
Menekan *mouse* ‘Klik Kiri’	Menggerakkan karaktek pemain untuk melakukan serangan
Menggerakkan *mouse* ke kiri	Menggerakkan pandangan karakter pemain ke kiri
Menggerakkan *mouse* ke kanan	Menggerakkan pandangan karakter pemain ke kanan
Menekan tombol *keyboard* ‘Space’	Menggerakkan karakter pemain untuk melompat

Menekan tombol *keyboard* ‘1’	Karakter menggunakan senjata utama
Menekan tombol *keyboard* ‘2’	Karakter menggunakan senjata kedua
Menekan tombol *keyboard* ‘*Esc*’	Menampilkan layar pause menu
Kondisi Akhir	Menampilkan *Level* 3 permainan

5. Skenario *Use Case Level*4

Tabel 3. 13 Skenario *Use Case Level*4

Identifikasi	
Nomor	11
Nama	*Level*4
Tujuan	Melihat *Level* 4 permainan
Deskripsi	Proses untuk menampilkan *Level* 4 permainan
Aktor	Pemain/Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di pause menu permainan
Aksi Aktor	**Reaksi Sistem**
Memilih menu main	Sistem menampilkan *Level* 4 permainan
Menekan tombol *keyboard* ‘W’	Menggerakkan karakter pemain ke depan
Menekan tombol *keyboard* ‘A’	Menggerakkan karakter pemain ke kiri
Menekan tombol *keyboard* ‘S’	Menggerakkan karakter pemain ke belakang
Menekan tombol *keyboard* ‘D’	Menggerakkan karakter pemain ke kanan
Menekan *mouse* ‘Klik Kiri’	Menggerakkan karaktek pemain

	untuk melakukan serangan
Menggerakkan *mouse* ke kiri	Menggerakkan pandangan karakter pemain ke kiri
Menggerakkan *mouse* ke kanan	Menggerakkan pandangan karakter pemain ke kanan
Menekan tombol *keyboard* ‘Space’	Menggerakkan karakter pemain untuk melompat
Menekan tombol *keyboard* ‘1’	Karakter menggunakan senjata utama
Menekan tombol *keyboard* ‘2’	Karakter menggunakan senjata kedua
Menekan tombol *keyboard* ‘*Esc*’	Menampilkan layar pause menu
Kondisi Akhir	Menampilkan *Level* 4 permainan

6. Skenario *Use Case* Keluar

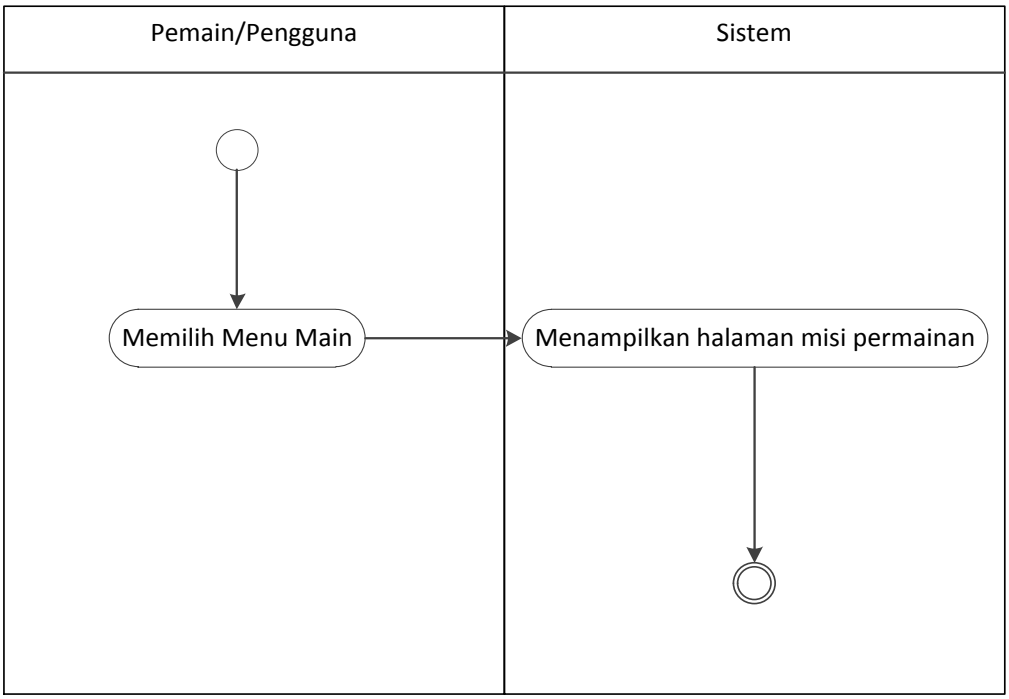
Tabel 3. 14 Skenario *Use Case* Keluar

Identifikasi	
Nomor	12
Nama	Keluar
Tujuan	Menutup aplikasi permainan
Deskripsi	Proses untuk menutup aplikasi permainan
Aktor	Pemain/Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama
Aksi Aktor	**Reaksi Sistem**
Memilih menu keluar	Sistem menutup aplikasi permainan
Kondisi Akhir	Menutup aplikasi permainan

3.1.8.5. *Activity* Diagram

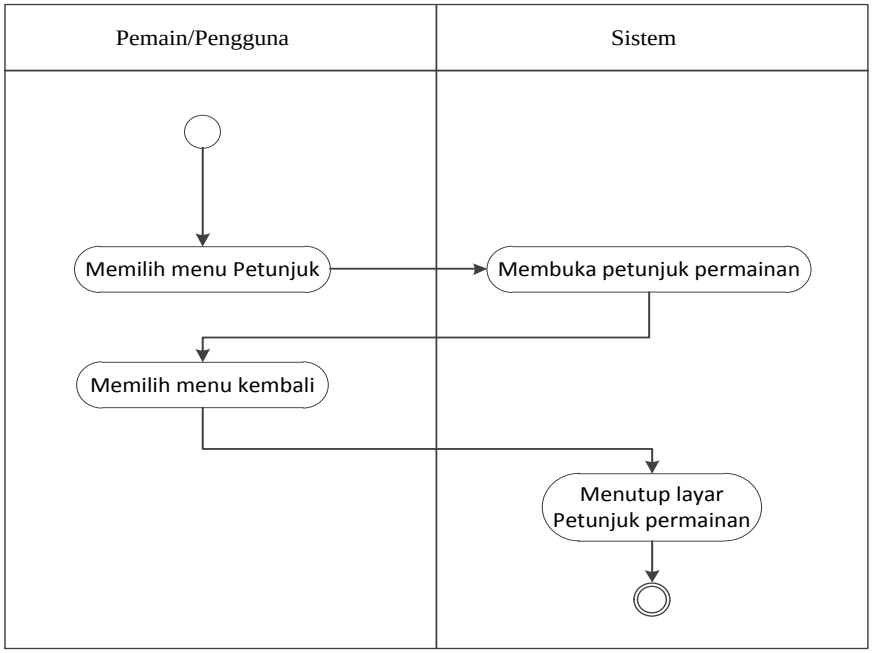
Diagram *Activity* menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti pada sistem yang sedang dirancang. Bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

1. Diagram *Activity* Mulai



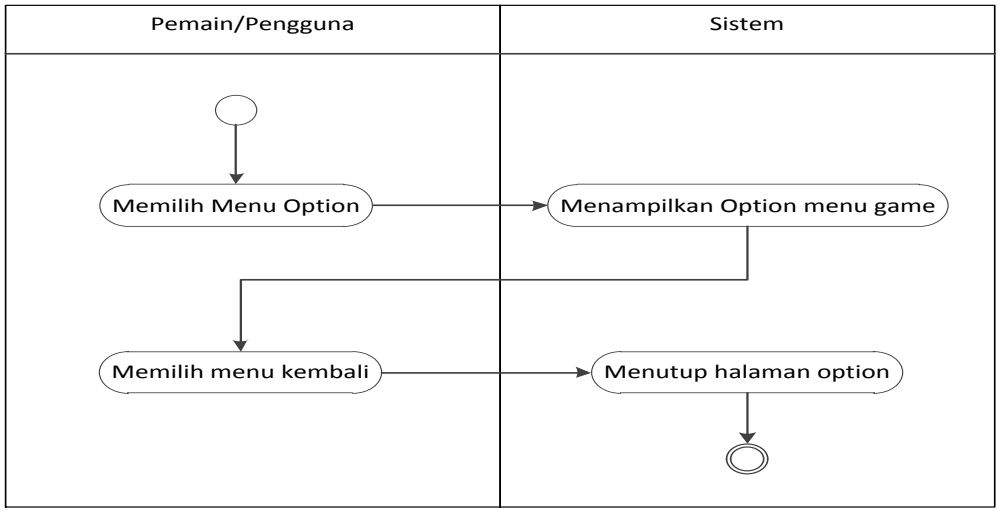
Gambar 3. 12 Diagram *Activity* Mulai

2. Diagram *Activity* Petunjuk



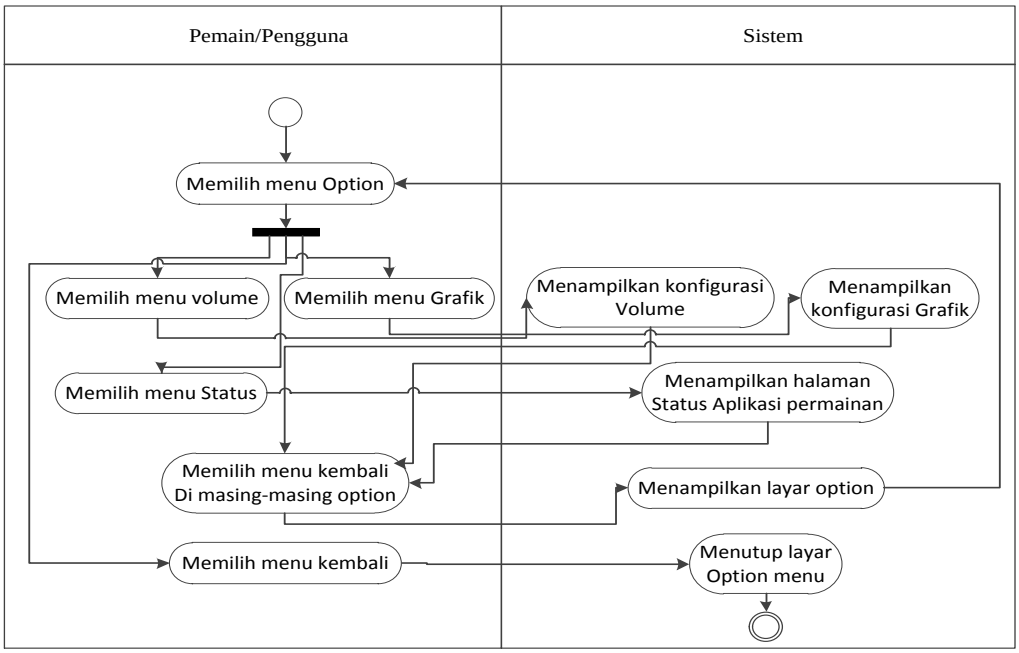
Gambar 3. 13 Diagram *Activity* Petunjuk

3. Diagram *Activity* Option



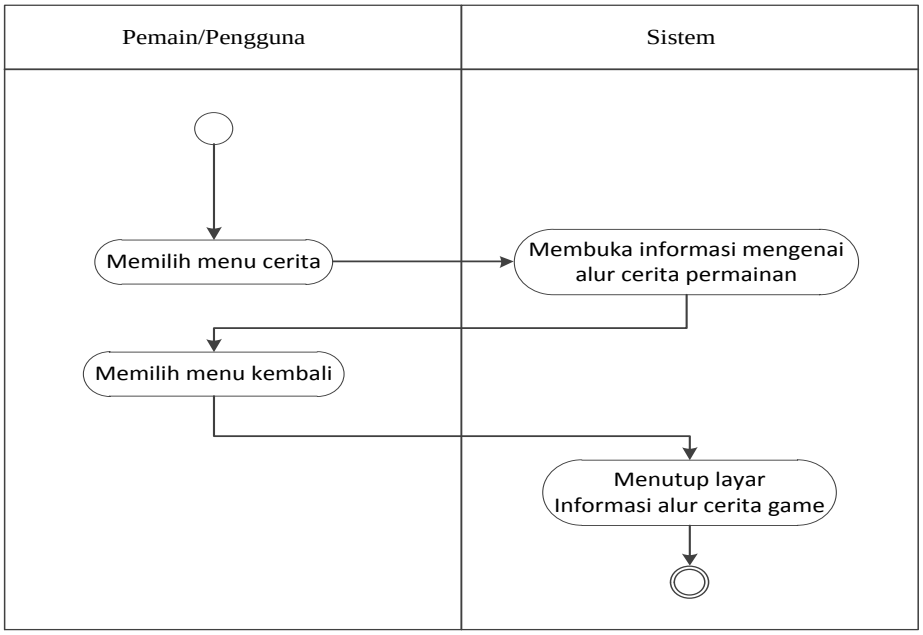
Gambar 3. 14 Diagram *Activity* Option

4. Diagram *Activity* Option Menu



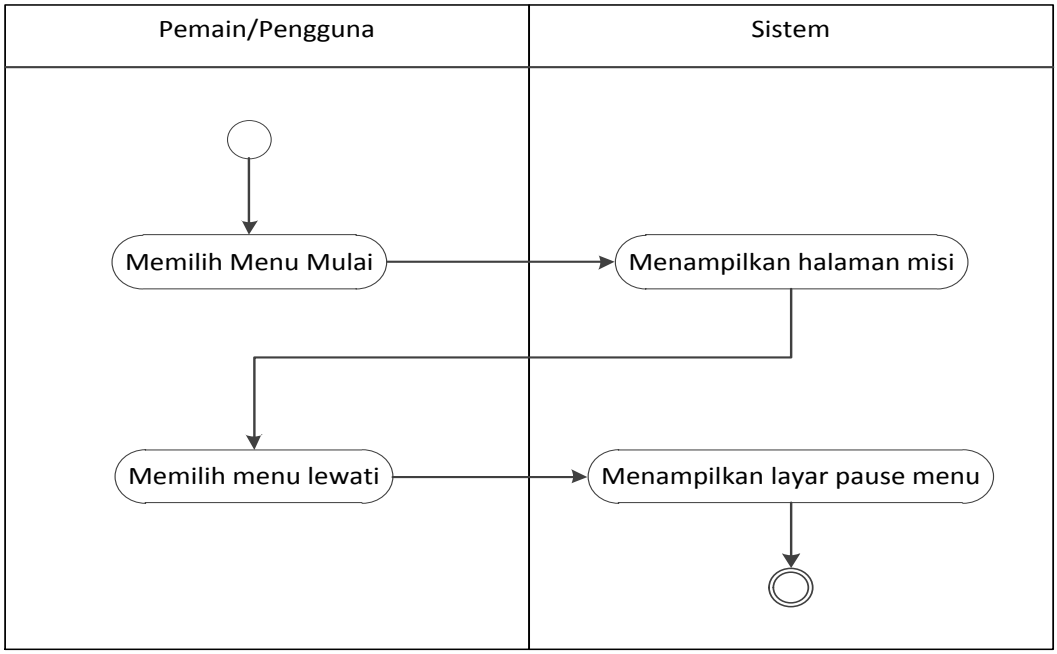
Gambar 3. 15 Diagram *Activity* Option

5. Diagram *Activity* Cerita



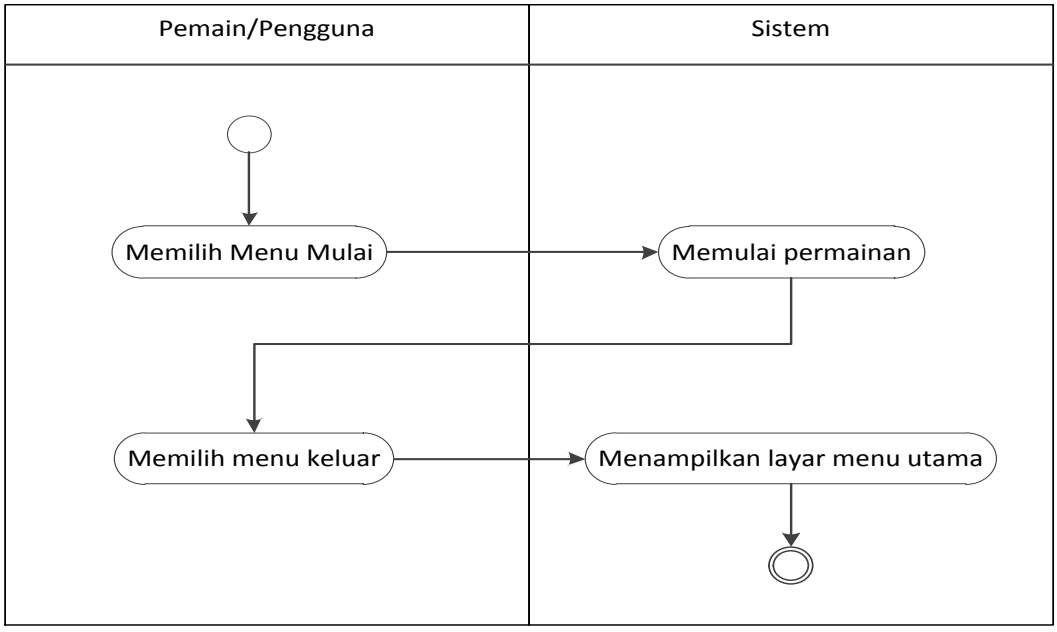
Gambar 3. 16 Diagram *Activity* Cerita

6. Diagran *Activity* Misi



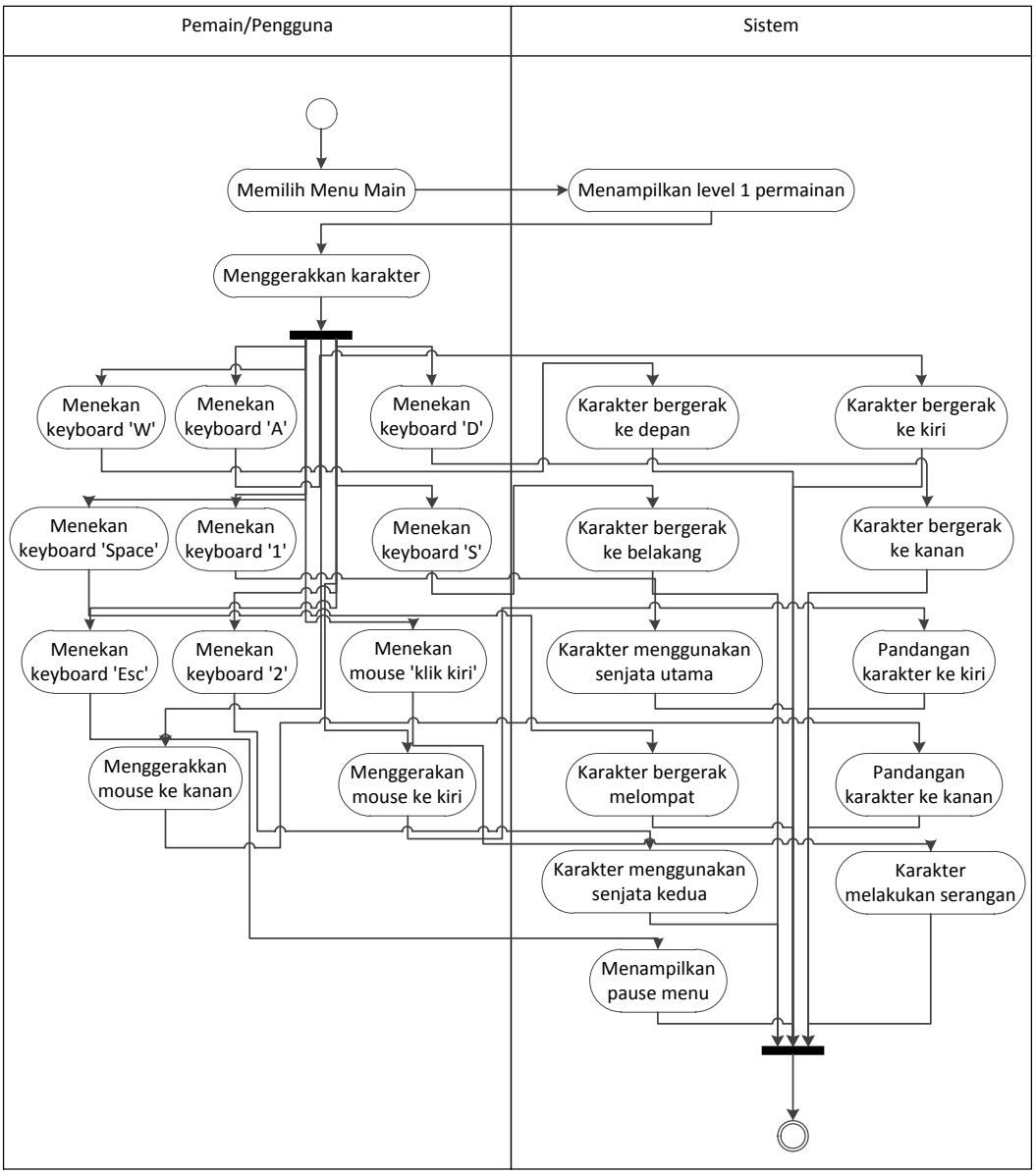
Gambar 3. 17 Diagram *Activity* Misi

7. Diagram *Activity* Pause Menu



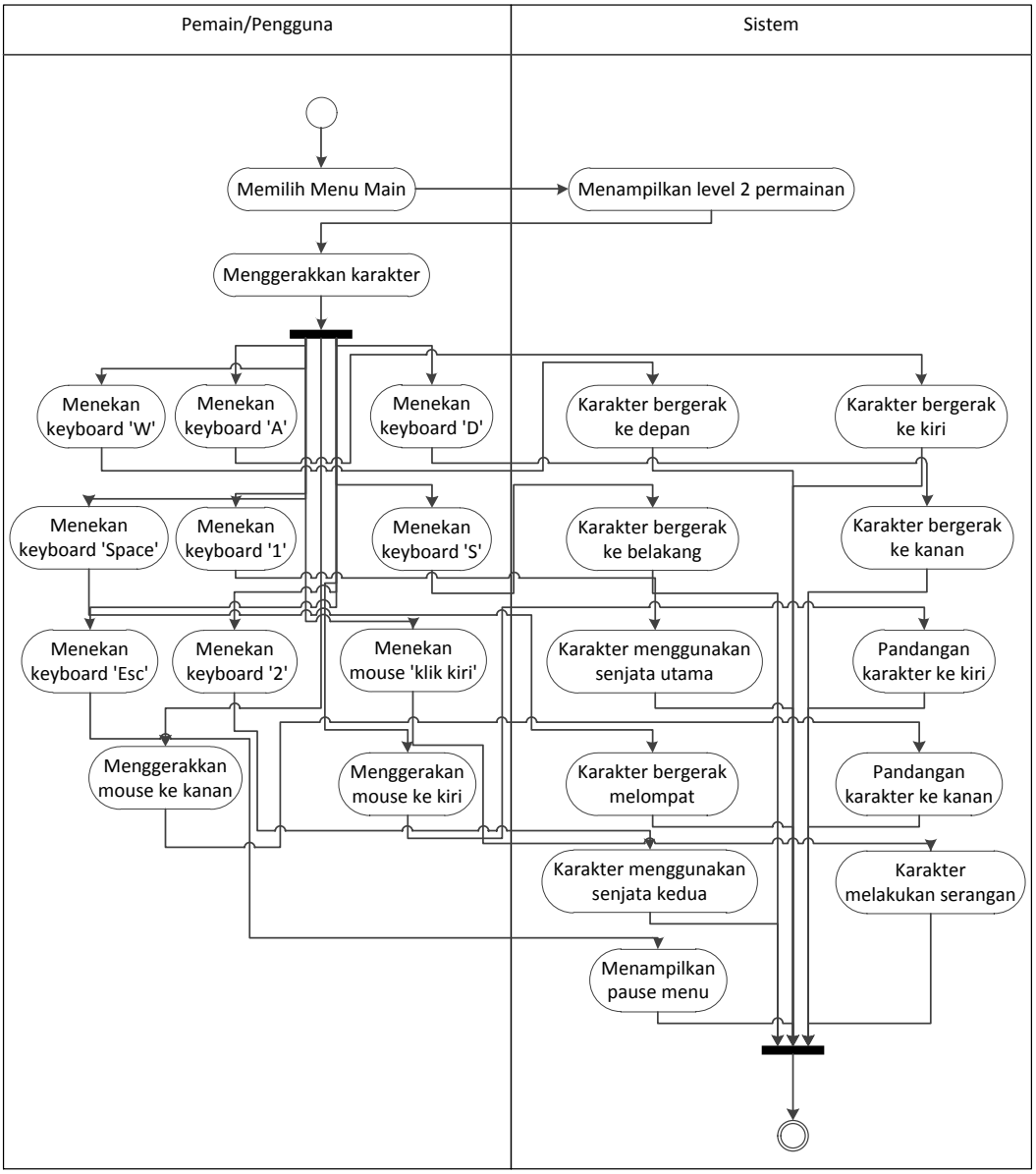
Gambar 3. 18 Diagram *Activity* Pause Menu

8. Diagram Activity Level 1



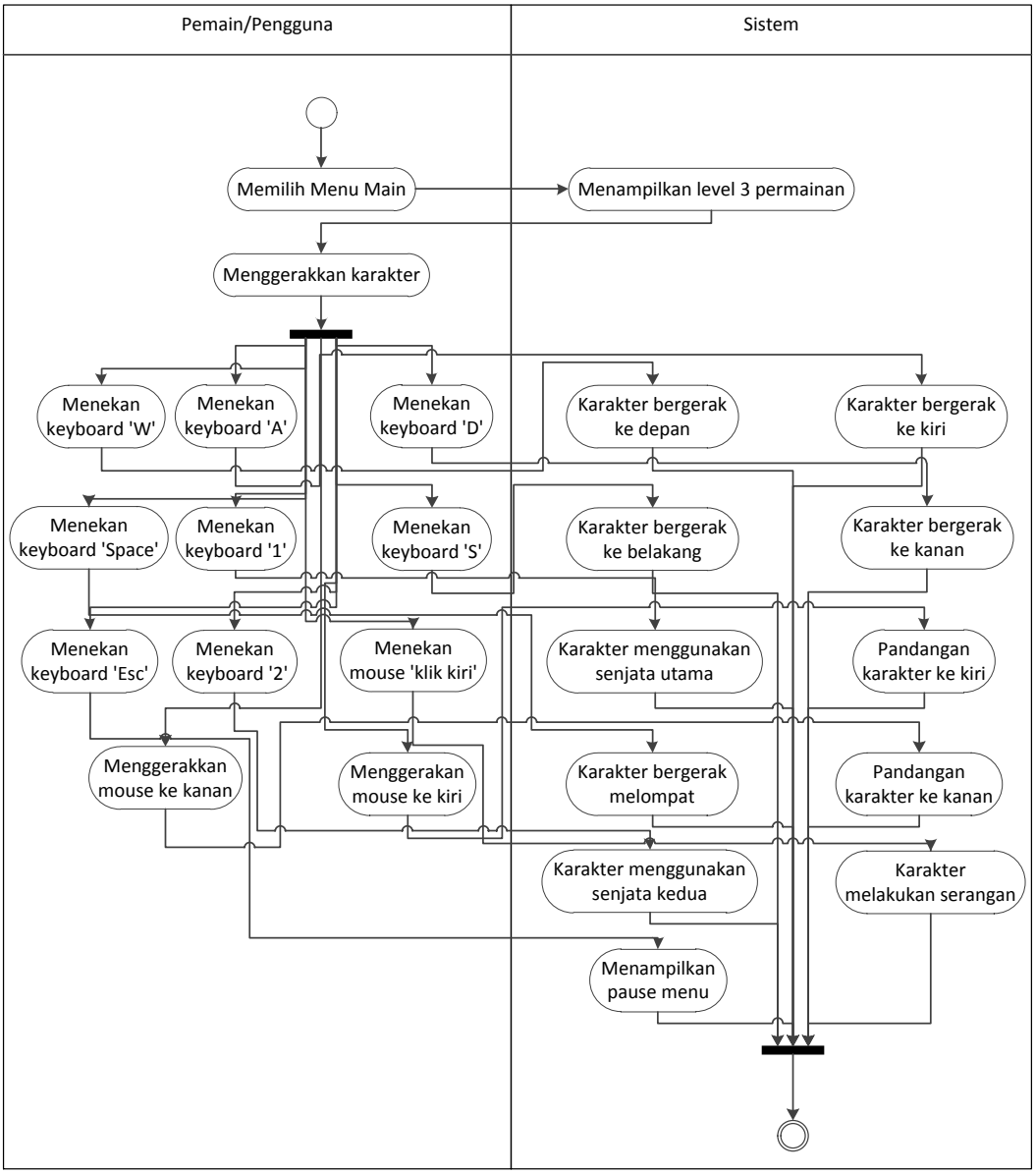
Gambar 3. 19 Diagram *Activity Level* 1

2. Diagram Activity Level 2



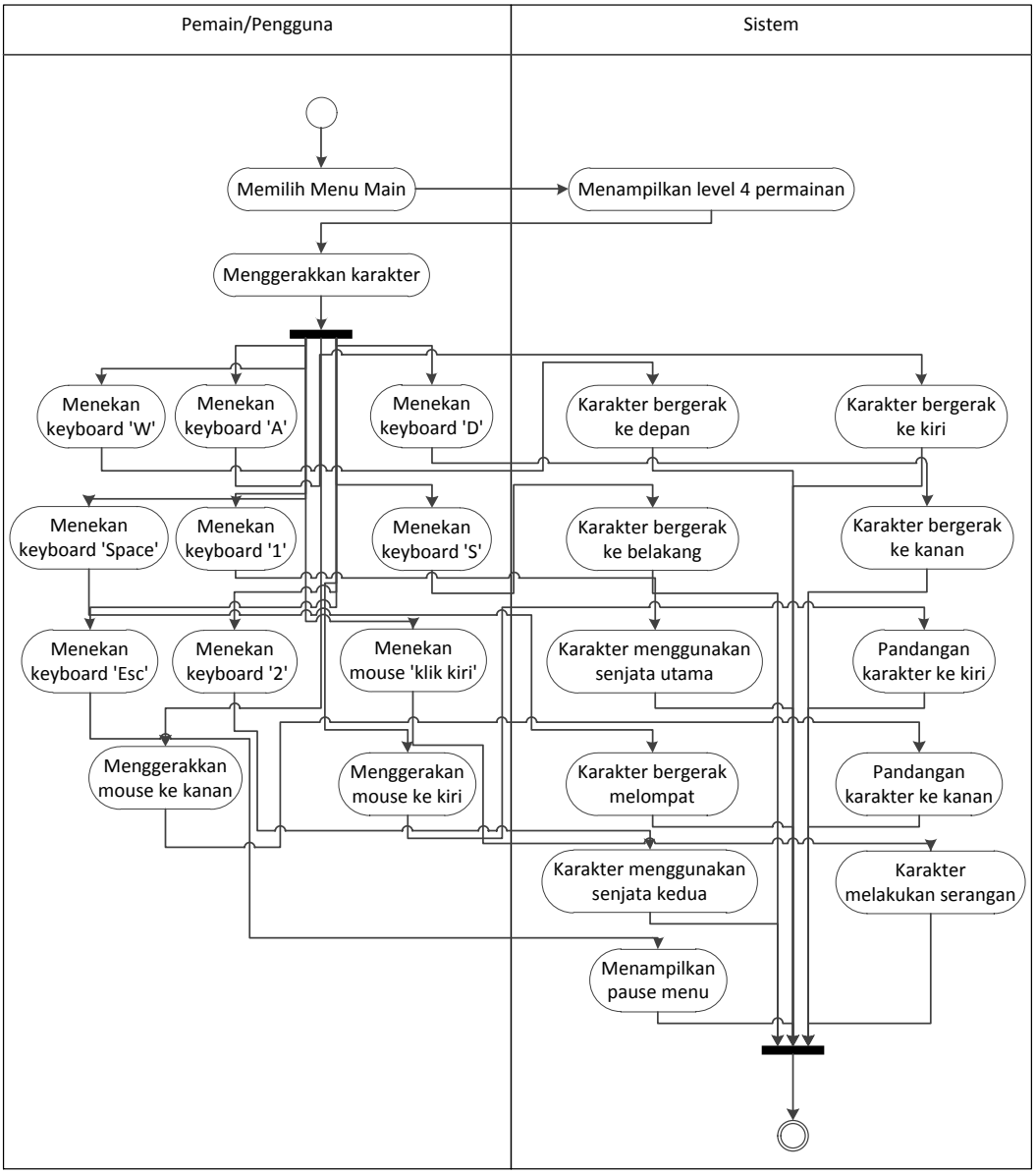
Gambar 3. 20 Diagram *Activity Level* 2

3. Diagram Activity Level 3



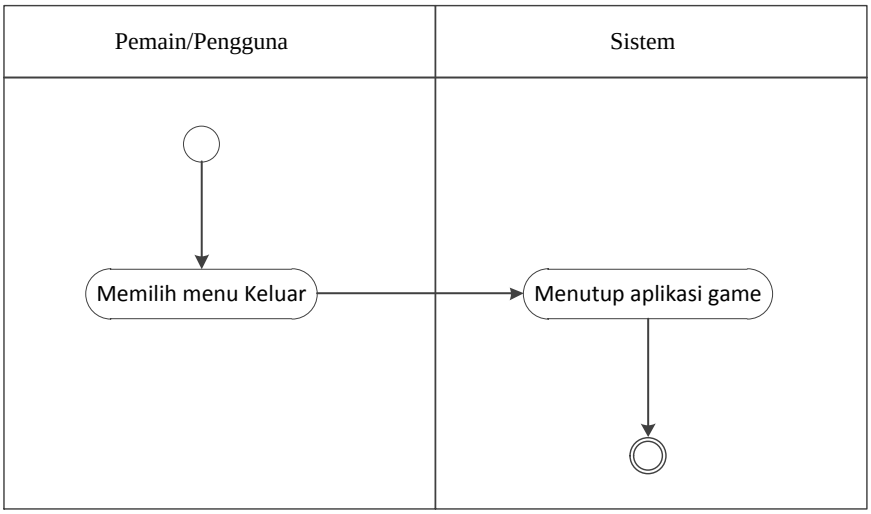
Gambar 3. 21 Diagram *Activity Level* 3

4. Diagram Activity Level 4



Gambar 3. 22 Diagram *Activity Level* 4

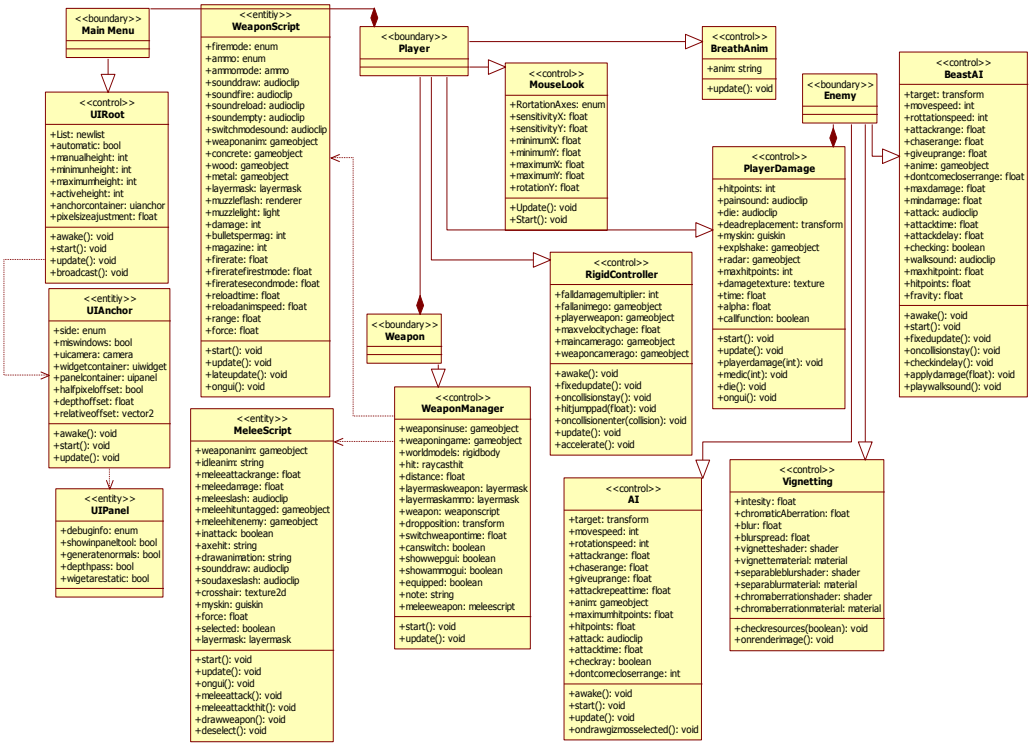
5. Diagram *Activity* Keluar



Gambar 3. 23 Diagram *Activity* Keluar

3.1.8.6. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Gambar 3.25 berikut adalah class diagram dari aplikasi game.

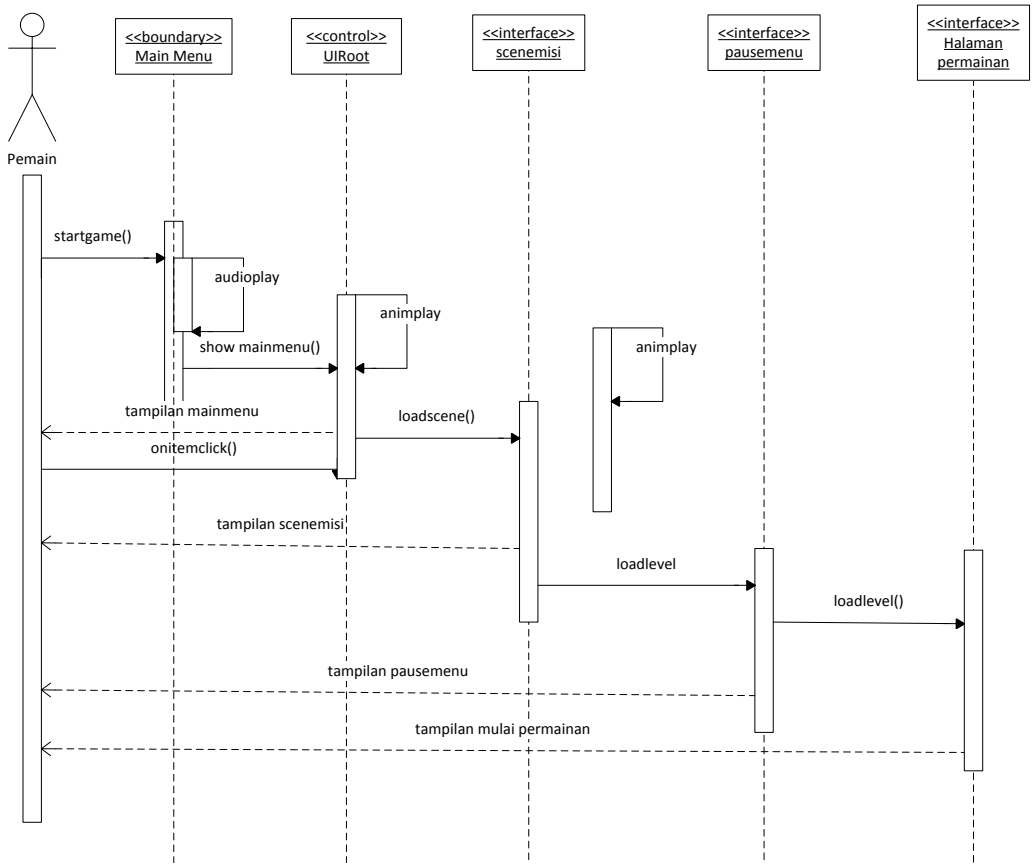


Gambar 3. 24 Class Diagram Game Survive from death

3.1.8.7. Sequence Diagram

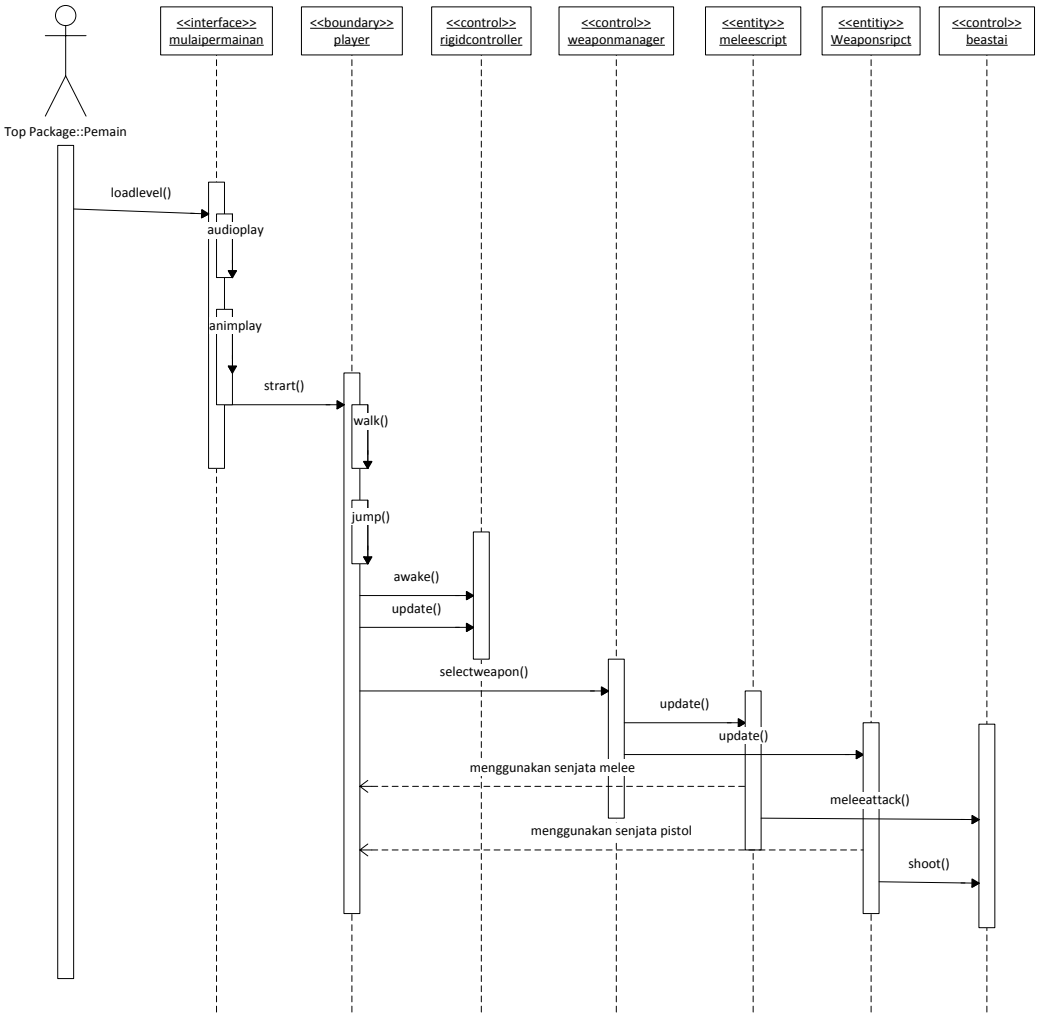
Sequence Diagram merupakan penggambaran dari interaksi antar objek di dalam dan disekitar sistem (termasuk pengguna, *display*, dan sebagainya). *Sequence Diagram* terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait)

1. **Sequence Diagram Mulai Main**



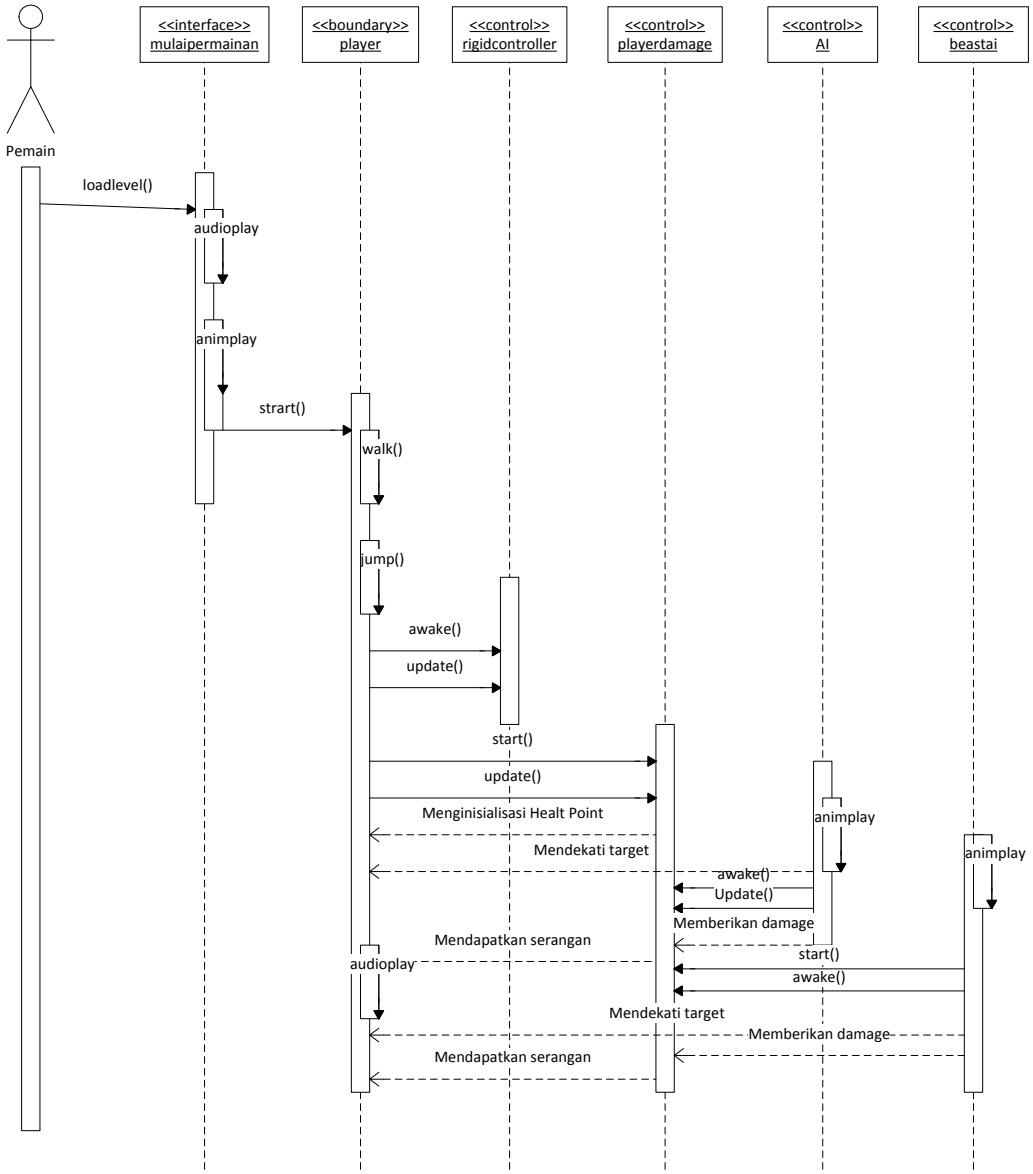
Gambar 3. 25 *Sequence Diagram* Mulai Main

2. Sequence Diagram Permainan



Gambar 3. 26 *Sequence Diagram* Permainan

3. Sequence Diagram Permaian (AI)



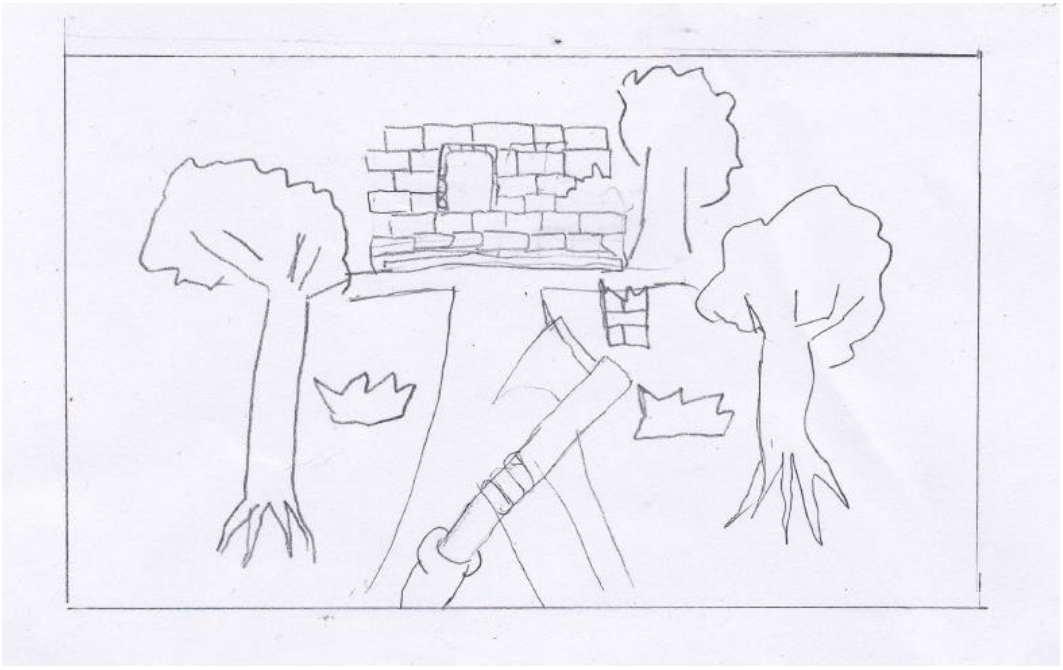
Gambar 3. 27 *Sequence Diagram* Permainan (AI)

3.2. Perancangan Sistem

Perancangan adalah suatu bagian dari metodologi pengembangan pembangunan suatu perangkat lunak yang dilakukan setelah tahapan untuk memberikangambaran secara terperinci. Berdasarkan uraian diatas perancangan sistem merupakan tahapan dari siklus pengembangan sistem yang didefinisikan dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk, yang dapat berupa penggambaran, perancangan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satukesatuan yang utuh dan berfungsi, juga menyangkut konfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem.

3.2.1. Storyboard

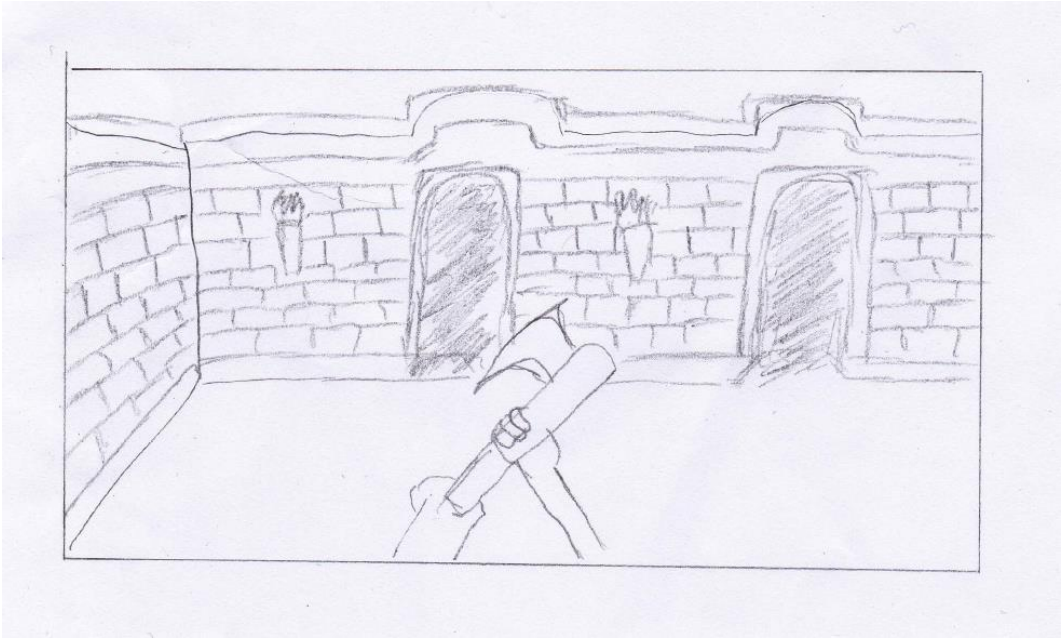
Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan storyboard kita dapat menyampaikan ide cerita kepada orang lain dengan lebih mudah, karena dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji, sehingga menghasilkan persepsi yang sama pada ide cerita. Berikut merupakan storyboard dari game Survive From Death:



Gambar 3. 28 Storyboard Map *Level* 1

Tabel 3. 15 Penjelasan Objek Map *Level* 1

No	Jenis Objek	Nama Objek	Keterangan
1	Item	Hp Aid	Untuk Memulihkan Hp
		Senter	Alat bantu pemain untuk penerangan
2	Weapon	Axe	Kapak untuk menyerang musuh

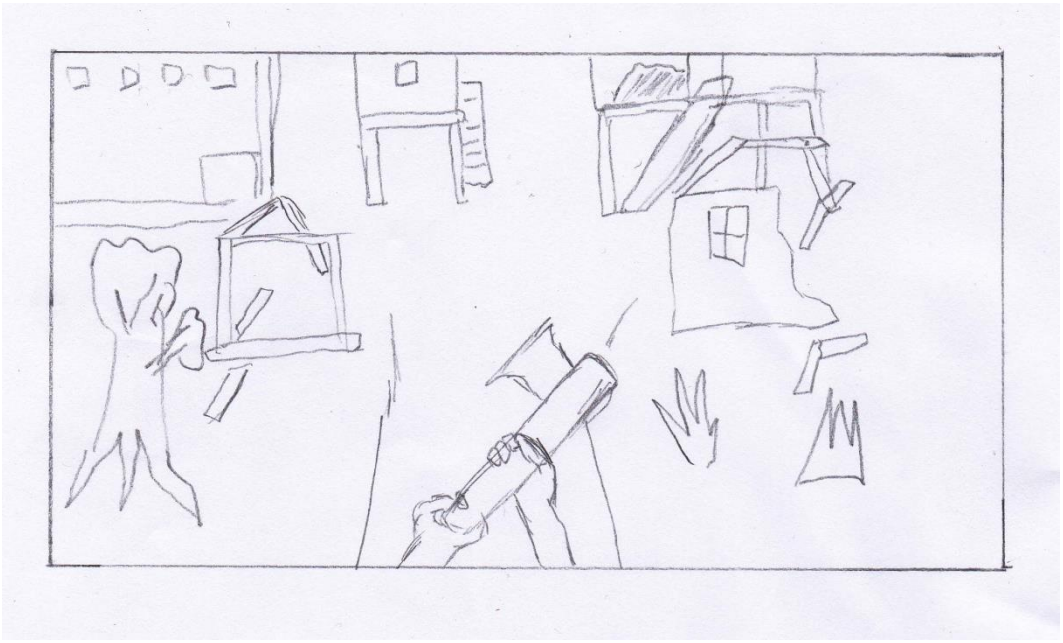


Gambar 3. 29 Storyboard Map *Level* 2

Tabel 3. 16 Penjelasan Objek Map *Level* 2

No	Jenis Objek	Nama Objek	Keterangan
1	Item	Hp Aid	Untuk Memulihkan Hp
		Senter	Alat bantu pemain

			untuk penerangan
2	Weapom	Pistol	Senjata api untuk menyerang musuh
		Axe	Senjata jarak dekat untuk menyerang musuh

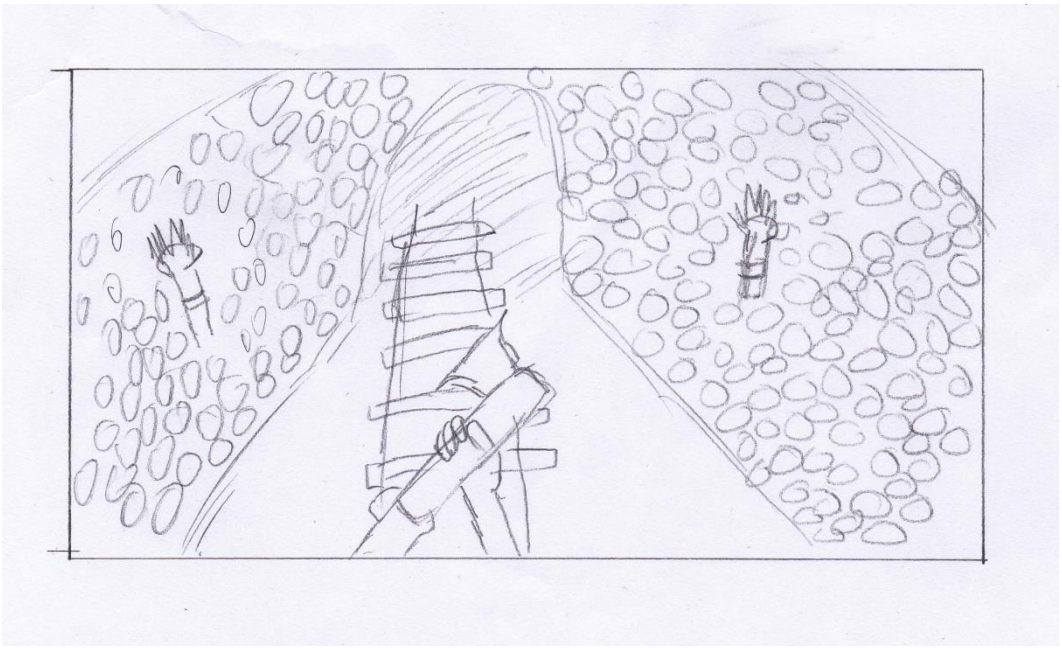


Gambar 3. 30 Storyboard Map *Level* 3

Tabel 3. 17 Penjelasan Objek Map *Level* 3

No	Jenis Objek	Nama Objek	Keterangan
1	Item	Hp Aid	Untuk Memulihkan Hp
		Senter	Alat bantu pemain untuk penerangan
2	Weapon	Axe	Senjata jarak dekat untuk menyerang musuh

		Pistol	Senjata api untuk menyerang musuh



Gambar 3. 31 Storyboard Map *Level* 4

Tabel 3. 18 Penjelasan Objek Map *Level* 4

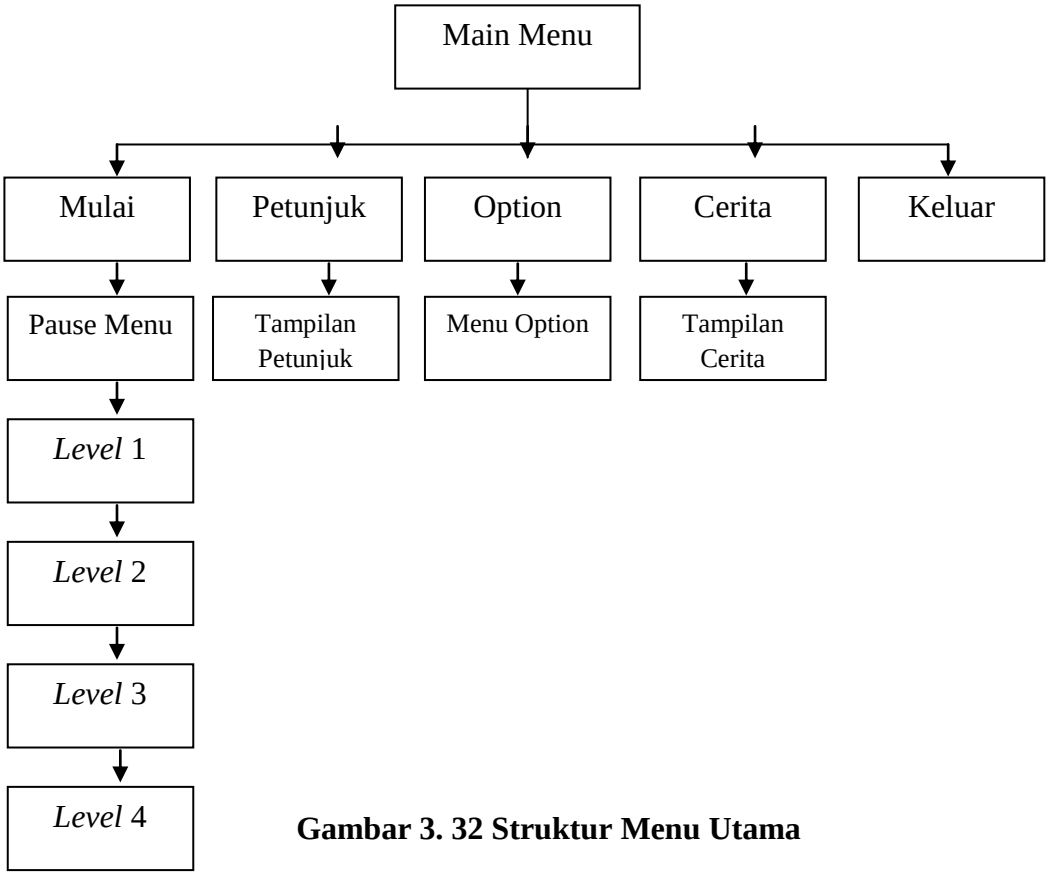
No	Jenis Objek	Nama Objek	Keterangan
1	Item	Hp Aid	Untuk Memulihkan Hp
		Senter	Alat bantu pemain untuk penerangan
2	Weapon	Axe	Senjata jarak dekat untuk menyerang musuh
		Pistol	Senjata api untuk menyerang musuh

3.2.2. Perancangan Arsitektur

Perancangan arsitekur yang dibuat meliputi beberapa perancangan diantaranya struktur menu, perancangan antarmuka.

3.2.3. Perancangan Struktur Menu

Perancangan menu dibuat dibuat agar pengguna dapat menggunakan aplikasi Game dengan mudah, sehingga pengguna dapat dengan mudah memilih menu aplikasi yang sedang berjalan. Stuktur menu dapat dilihat pada gambar 3.33.

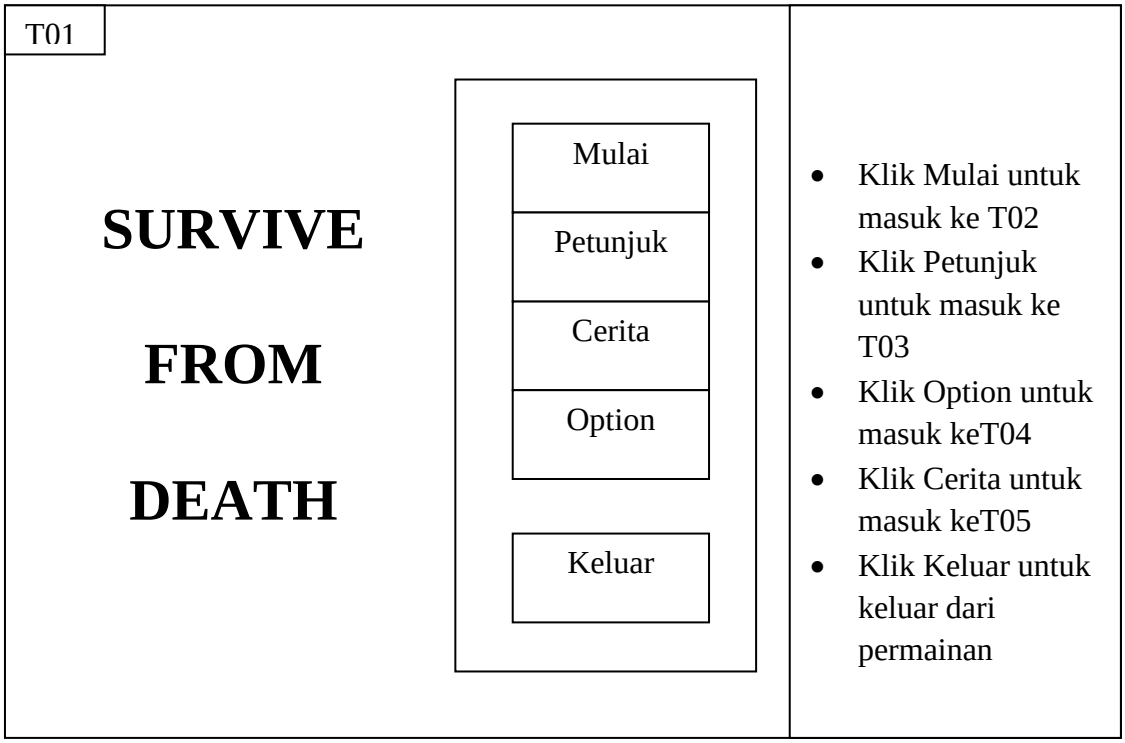


Gambar 3. 32 Struktur Menu Utama

3.2.4. Perancangan Antar Muka

Perancangan antarmuka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun, berikut ini adalah perancangan antarmuka dari aplikasi yang akan dibangun.

1. Antarmuka Menu Utama



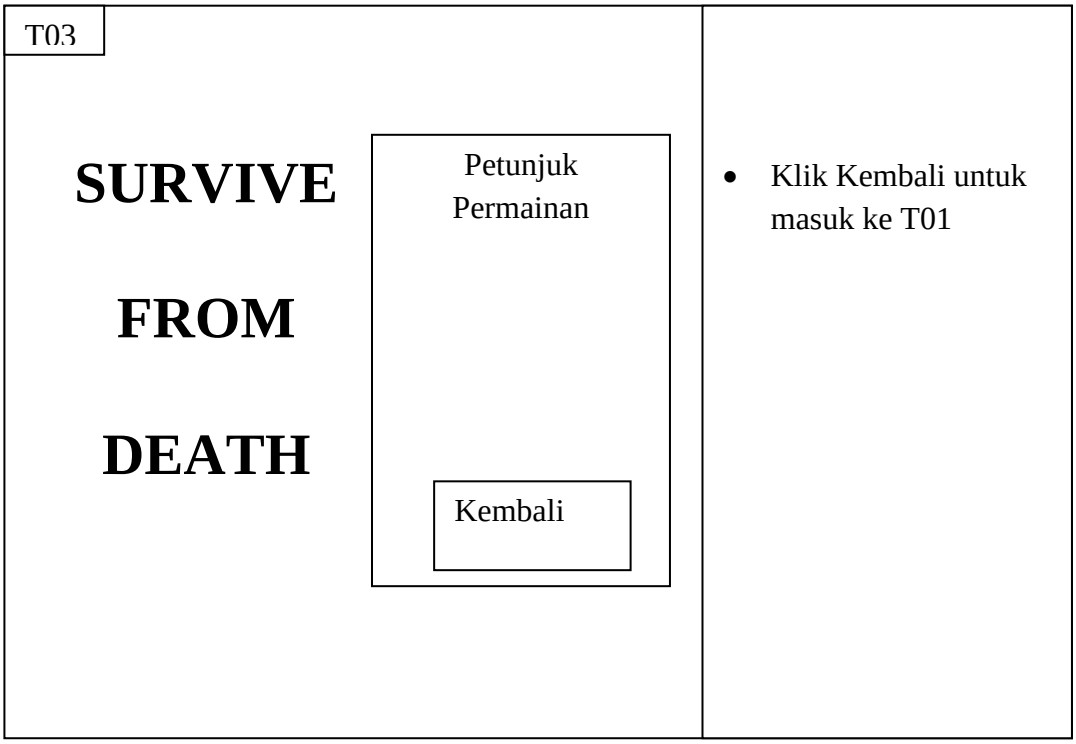
Gambar 3. 33 Antarmuka Tampilan Menu Utama

Tabel 3. 19 Penggunaan Tombol Menu Utama

No	Jenis Objek	Nama Objek	Keterangan
1	Button	Mulai	Untuk masuk memulai permainan
		Petunjuk	Untuk menampilkan halaman petunjuk
		Option	Untuk menampilkan halaman option
		Cerita	Untuk

			menampilkan halaman Cerita
		Keluar	Untuk menutup aplikasi

2. Antarmuka Petunjuk

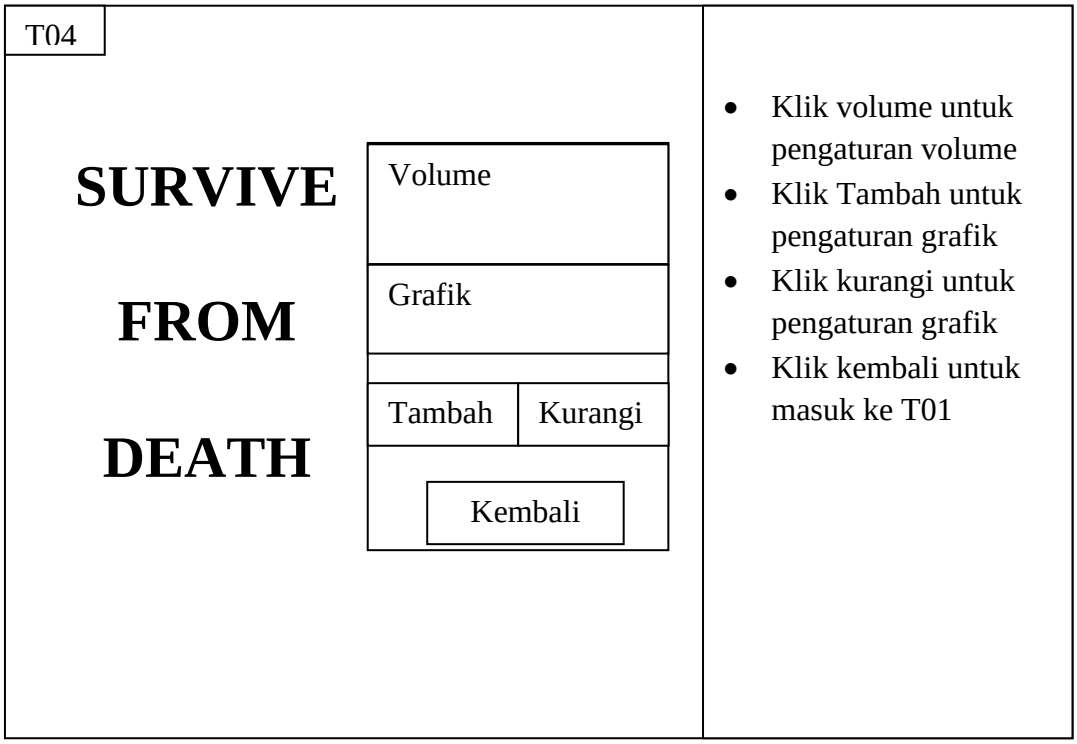


Gambar 3. 34 Antarmuka Tampilan Petunjuk

Tabel 3. 20 Penggunaan Tombol Petunjuk

No	Jenis Objek	Nama Objek	Keterangan
1	Button	Kembali	Untuk Kembali ke menu utama

3. Antarmuka Option

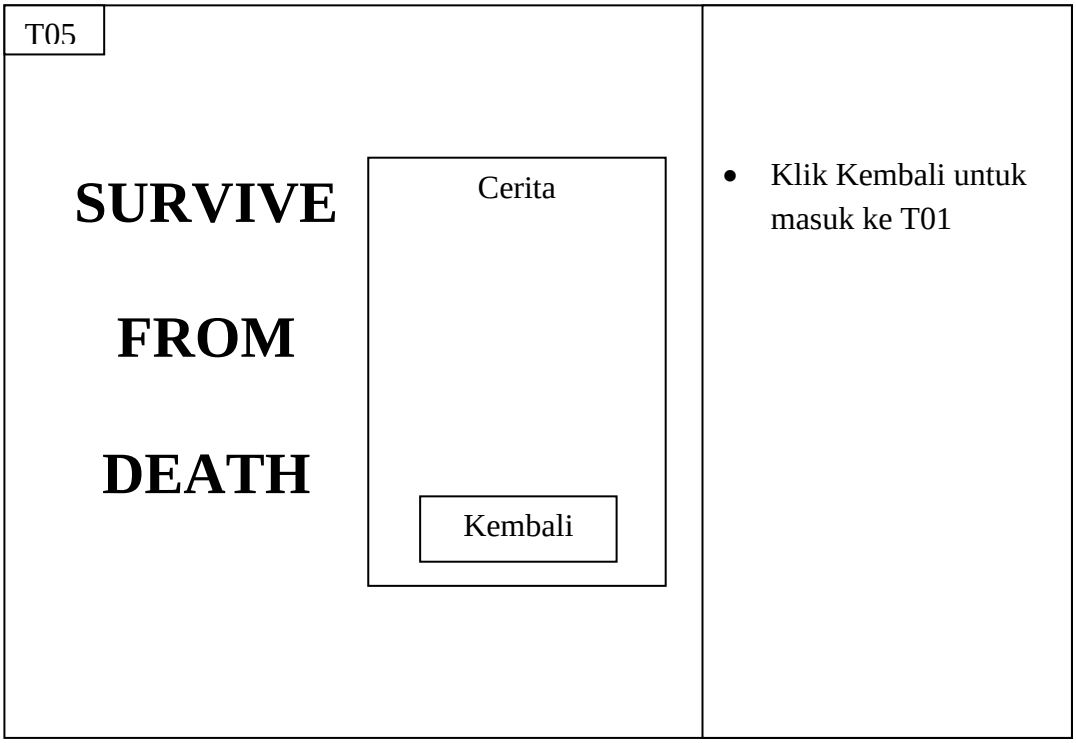


Gambar 3. 35 Antarmuka Tampilan Option

Tabel 3. 21 Penggunaan Tombol Option

No	Jenis Objek	Nama Objek	Keterangan
1	Button	Volume	Untuk pengaturan volume
		Tambah	Untuk pengaturan grafik
		Kurangi	Untuk pengaturan grafik
		Kembali	Untuk Kembali ke menu utama

4. Antarmuka Cerita



Gambar 3. 36 Antarmuka Tampilan Cerita

Tabel 3. 22 Penggunaan Tombol Cerita

No	Jenis Objek	Nama Objek	Keterangan
1	Button	Kembali	Untuk Kembali ke menu utama

5. Antarmuka Halaman Misi

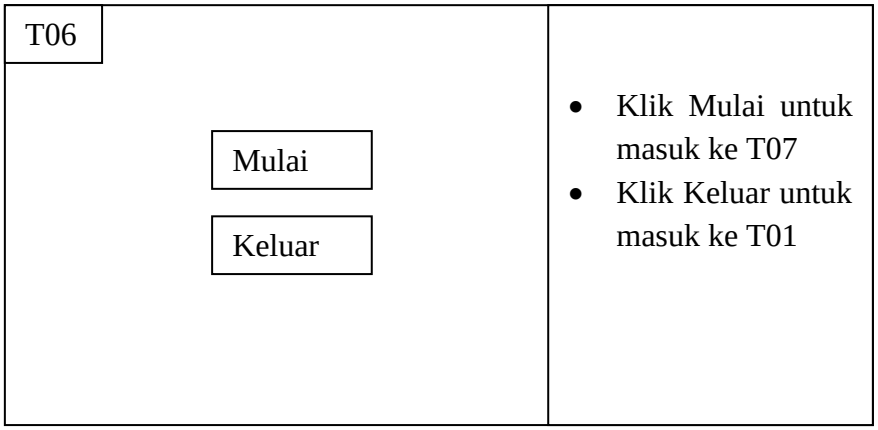


Gambar 3. 37 Antarmuka Halaman Misi

Tabel 3. 23 Penggunaan Tombol Halaman Misi

No	Jenis Objek	Nama Objek	Keterangan
1	Keyboard	Esc	Untuk melewati halaman misi dan langsung ke pause menu

6. Antar Muka Pause Menu

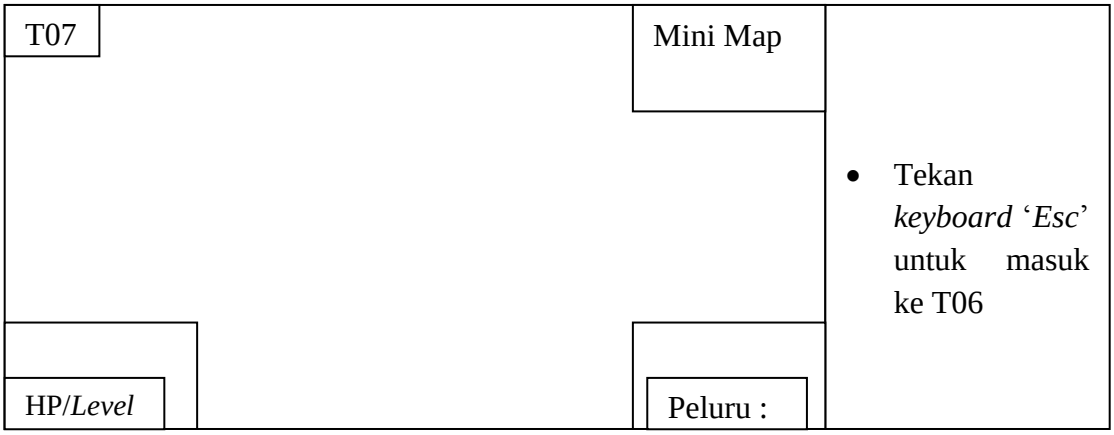


Gambar 3. 38 Antarmuka Tampilan Pause Menu

Tabel 3. 24 Penggunaan Tombol Pause Menu

No	Jenis Objek	Nama Objek	Keterangan
1	Button	Mulai	Untuk memulai permainan
		Keluar	Untuk ke Menu Utama

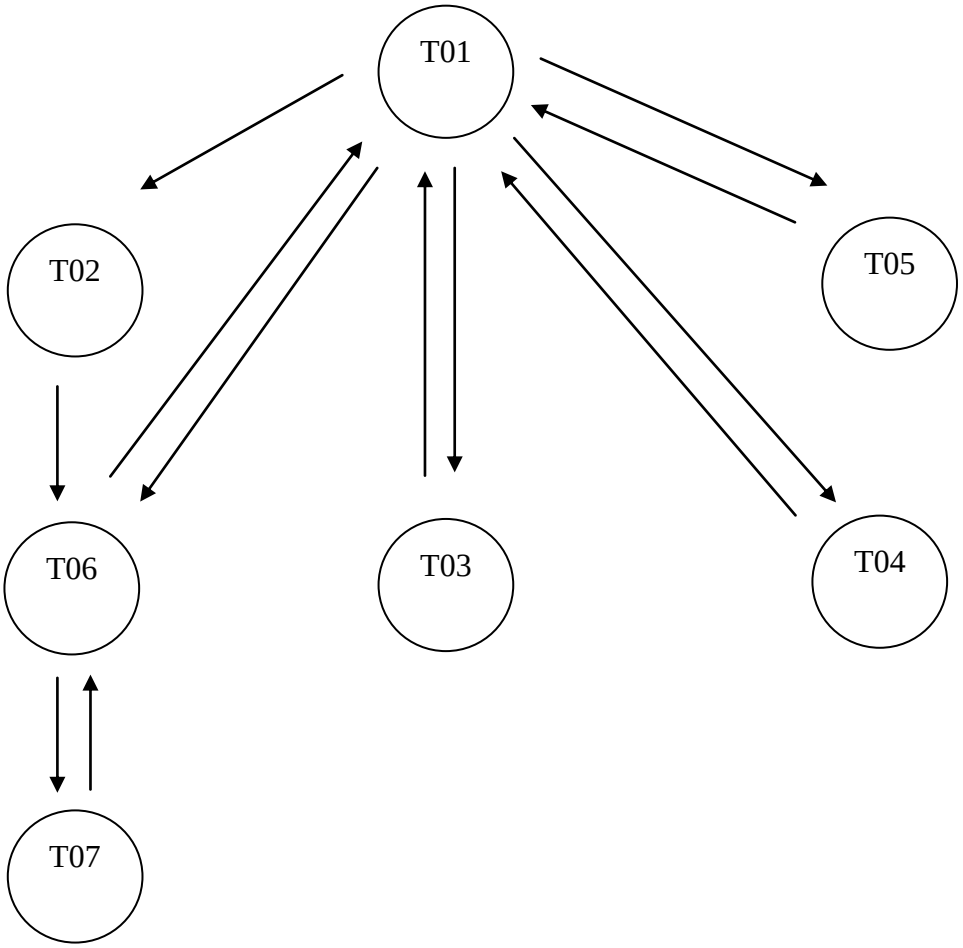
7. Antarmuka Permainan *Level* 1



Gambar 3. 39 Antarmuka Permainan *Level* 1

3.2.5. Jaringan Semantik

Struktur aplikasi ini dibuat secara moduler, yaitu program dipecah menjadi modul-modul kecil yang mudah dibuat, mudah dites, dan mudah dimodifikasi. Dalam pembuatan game ini menggunakan alat bantu bagan structural yang disebut juga jaringan semantik dapat dilihat pada gambar 3.41.



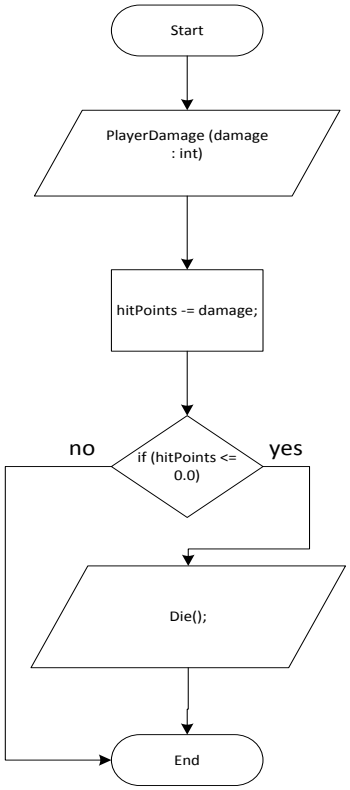
Gambar 3. 40 Jaringan Semantik

3.2.6. Perancangan Method

Perancangan *method* merupakan perancangan yang berfungsi untuk mendeskripsikan *method-method* yang berada di dalam aplikasi. *Method* dapat dipanggil dengan menyertakan variable, baik hanya satu variable, banyak variabel atau bahkan tidak ada sama sekali. Adapun *method-method* yang terdapat dalam game FPS Survive from death yang akan dibangun dijelaskan melalui *flowchart* sebagai berikut.

3.2.6.1. Method playerdamage

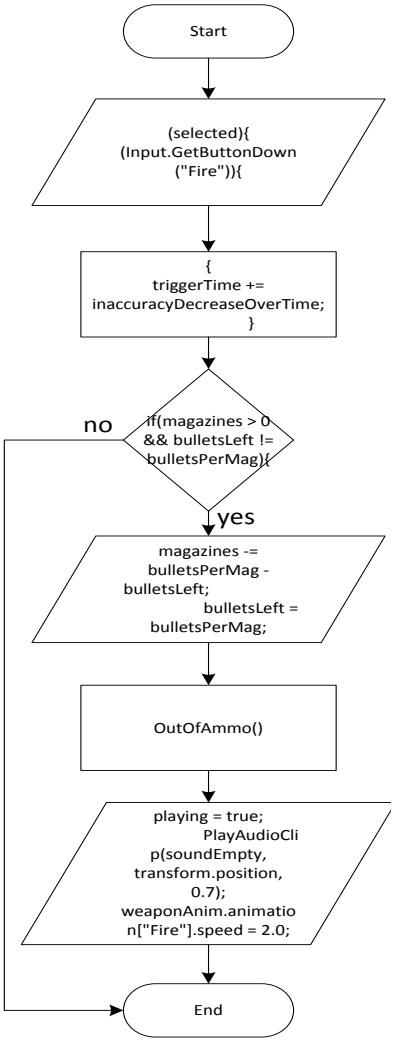
Method ini melakukan pengecekan terhadap setiap serangan/*damage* dari musuh yang mengenai pemain seperti pada gambar 3.42.



Gambar 3. 41 *Flowchart* Method playerdamage

3.2.6.2. Method *onGui* (WeaponScript)

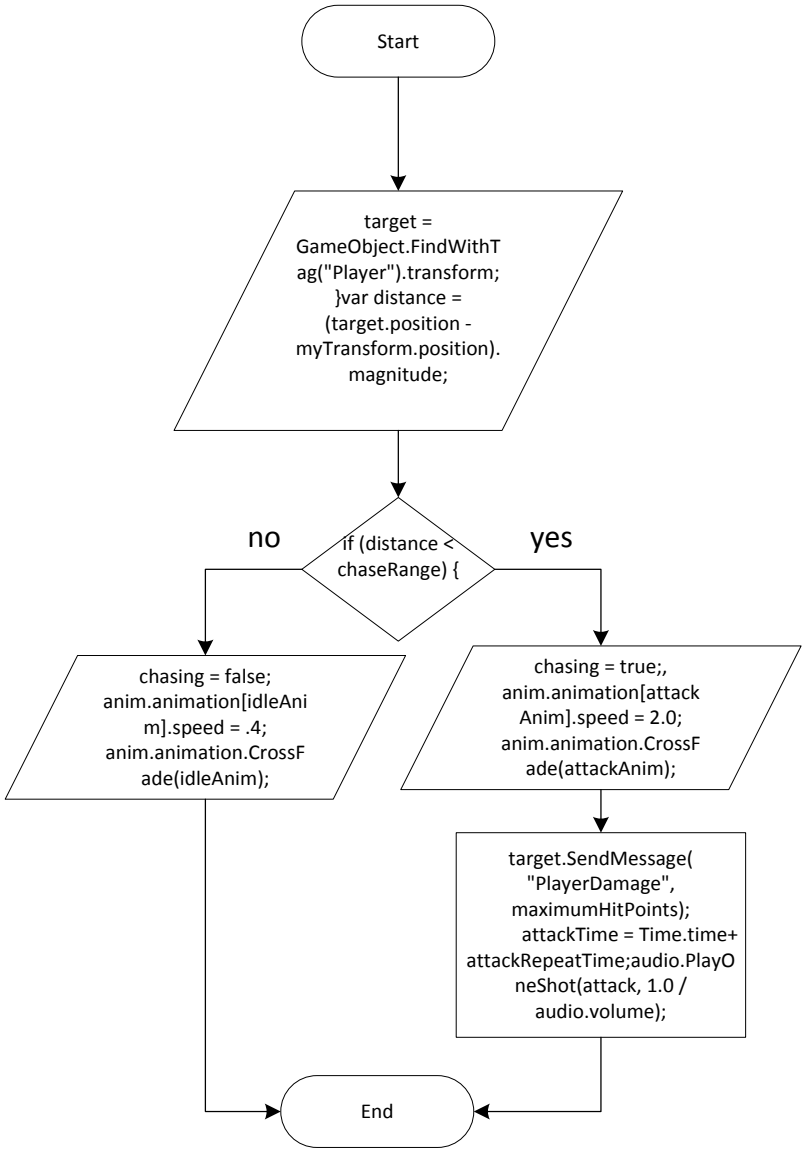
Method ini melakukan perubahann terhadap setiap peluru/*ammo* yang digunakan untuk menyerang musuh menggunakan senjata pistol seperti pada gambar 3.43.



Gambar 3. 42 *Flowchart* Method onGui

3.2.6.3. Method *awake* (*AI*)

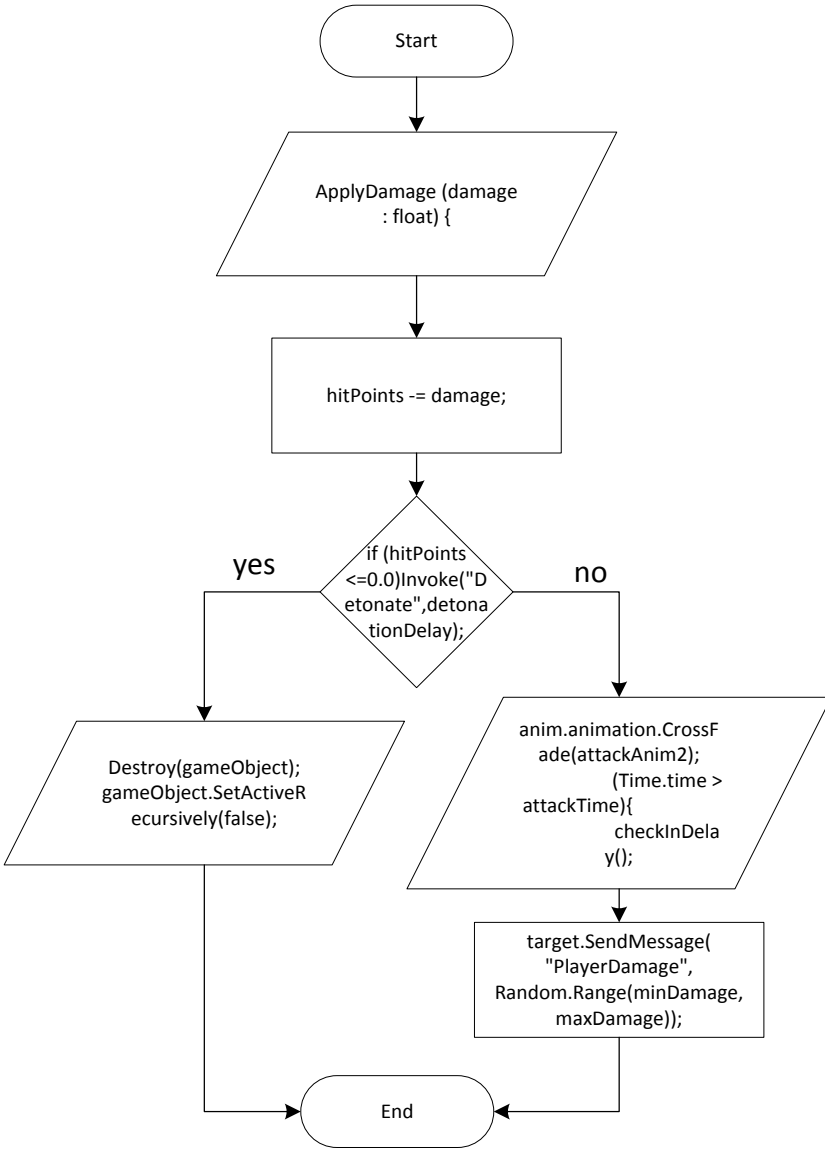
Method awake pada AI ini melakukan penyerangan/memberikan damage terhadap target untuk mengurangi jumlah healtpoint target sesuai damage yang diberikan seperti pada gambar 3.44



Gambar 3. 43 *Flowchart* Method awake(AI)

3.2.6.4. Method applydamage (BeastAI)

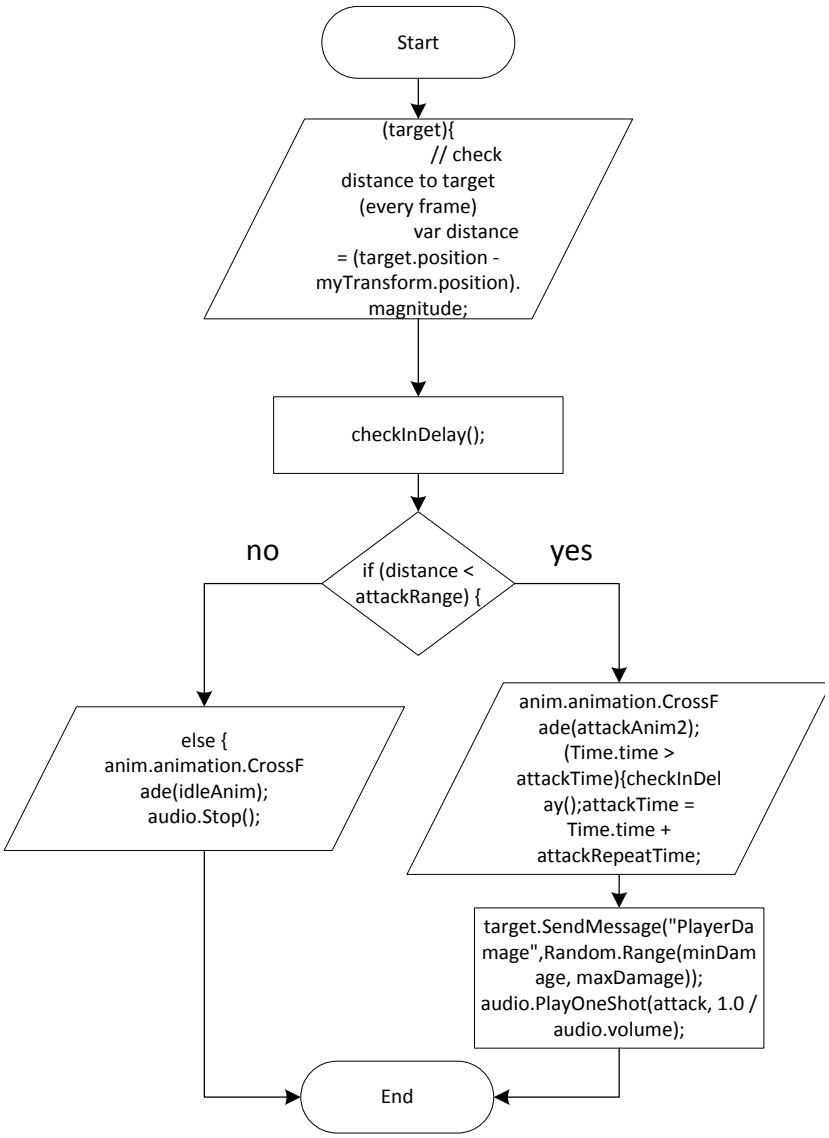
Method applydamage pada BeastAI ini mendapatkan serangan/menerima damage dari target untuk mengurangi jumlah healtpoint sesuai damage yang diterima seperti pada gambar 3.45.



Gambar 3. 44 *Flowchart* Method applydamage (*BeastAI*)

3.2.6.5. Method awake (BeastAI)

Method awake pada BeastAI ini melakukan penyerangan/memberikan damage terhadap target untuk mengurangi jumlah healtpoint target sesuai damage yang diberikan seperti pada gambar 3.46.



Gambar 3. 45 *Flowchart* Method awake(BeastAI)