BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Pada analisis sistem akan mencakup analisis game sejenis, analisis AI, analisis kebutuhan non-fungsional dan analisis kebutuhan fungsional.

3.1.1 Analisis Masalah

Berdasarkan tinjauan analisis sistem yang telah dilakukan dengan melakukan evaluasi terhadap *game* sejenis, berikut adalah beberapa masalah yang terdapat pada game sejenis :

1. Alur cerita *game* yang tidak jelas dan karakter NPC yang kurang bervariatif.
2. NPC masih kesulitan dalam melakukan pencarian jalan terhadap *player*.
3. *Enemy boss* yang hanya mempunyai kemampuan menyerang tanpa memiliki kemampuan seperti melarikan diri ataupun bertahan.

Beberapa permasalahan yang merupakan hasil dari analisis yang dijadikan sebagai latar belakang dalam pembangunan *game Action The Defend Island*.

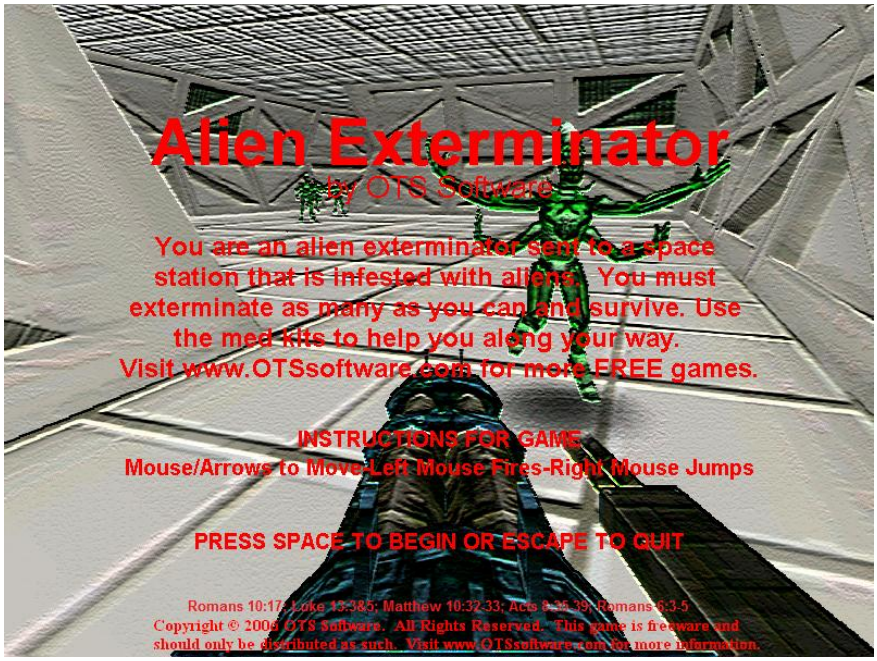
3.1.2 Analisis *Game* Sejenis

Pada analisis *game* sejenis, akan dilakukan observasi terhadap *game* yang memiliki kesamaan *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 dibahasa *game* sejenis yang dilihat dari jenis *genre game,* maupun dari kesamaan alur cerita yang diambil. *Game* yang menjadi acuan disini adalah *game Alien Exterminator* dan *game Island Defense*.

1. *Game Alien Exterminator*

Game Alien Exterminator merupakan *game* dengan *genre* FPS (*First Person Shooter*) yang mempunyai tampilan 3D (tiga dimensi).

Beberapa tampilan permainan dari *game Alien Exterminator* dapat dilihat pada gambar 3.1 dan gambar 3.2 :



Gambar 3.1 menu utama dari *game Alien Exterminator*.



Gambar 3.2 Layar Permainan *Alien Exterminator*

Gambar 3.2 merupakan tampilan layar permainan *level* satu pada *game Alien Exterminator*, pada layar permainan dapat dilihat status *lives*, *energy, score. lives* menunjukan jumlah nyawa yang dimiliki *player*, *energy* menunjukan jumlah *health* atau darah player dan *score* akan didapat ketika musuh terkena tembakan *player*.

2. *Gameplay Game Alien Exterminator*

Tabel 3.1 *Gameplay Game Alien Exterminator*

Level	**Deskripsi**
Level 1	Pada level pertama menceritakan seorang prajurit yang ditugaskan untuk mempertahankan pangkalan pesawat yang berada di tengah pulau dari serangan alien yang ingin menguasai pangkalan pesawat tersebut, pemain harus mengalahkan semua alien yang ada di pangkalan pesawat dan juga harus mempertahankan pulau dari invasi alien. Pemain harus masuk ke dalam gedung yang dipangkalan pesawat dimana para alien sudah berada di dalamnya dan pemain harus mengalahkan semua alien yang ada di dalam gedung pangkalan pesawat agar pangkalan pesawat tidak jatuh ke tangan *alien*. Ketika pemain terkena tembakan musuh maka darah atau *energy* pemain akan berkurang dan jika darah atau energy pemain habis, pemain akan mati. Akan tetapi di dalam arena permainan pemain dapat mencari healthpack yang berfungsi untuk menambah darah atau energy untuk pemain agar bisa bertahan hidup lebih lama. Pemain harus mengalahkan semua musuh yang ada di *level* 1 untuk bisa melanjutkan permainan ke *level* selanjutnya.
Level 2	Di level kedua pemain harus menghancurkan pesawat alien yang berada di pangkalan pesawat dan mengalahkan semua prajurit alien yang muncul, dan ketika prajurit alien sudah berhasil dikalahkan pemain akan menghadapi boss alien yang mempunyai *healthbar* lebih banyak dan damage tembakan yang lebih besar dari alien yang biasa sehingga sangat sulit untuk di kalahkan. Dan setiap level pemain

	mempunyai tiga nyawa sehingga pemain masih mempunyai kesempatan untuk bisa mengalahkan semua alien berserta boss.

3. Skenario *Game Alien Exterminator*

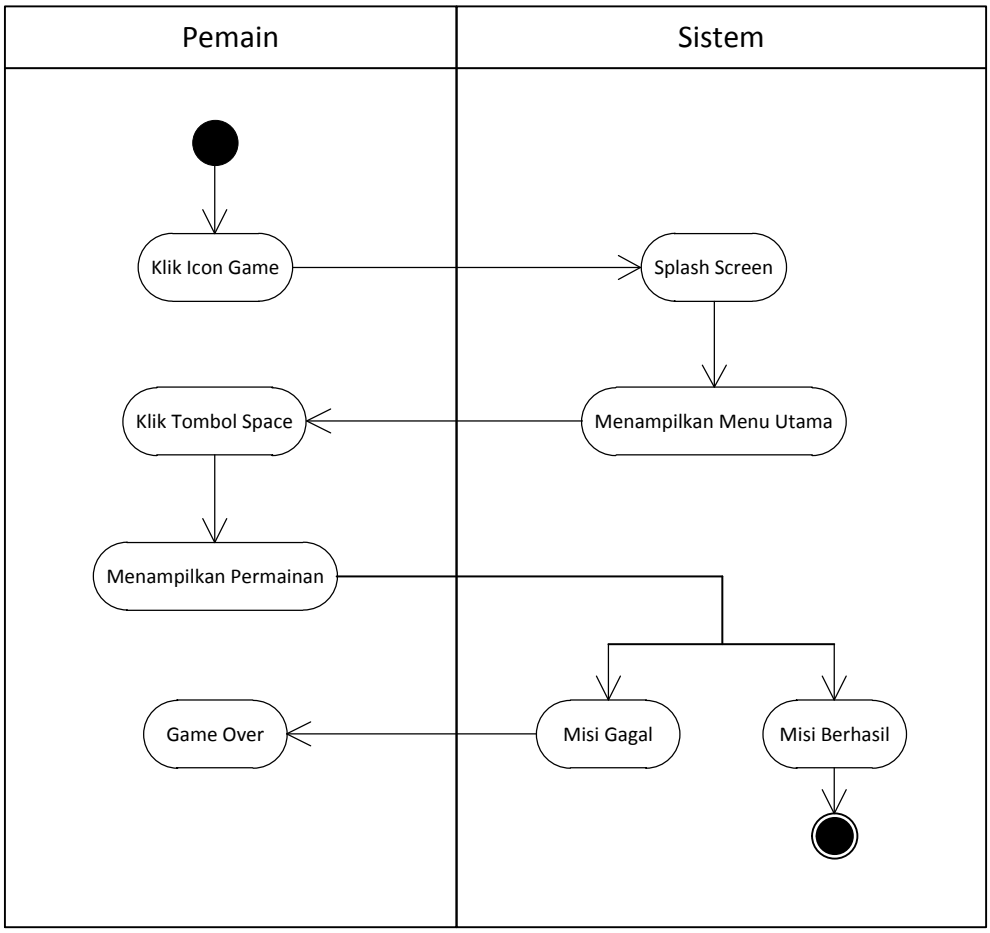
Skenario permainan dalam *game Alien Exterminator* disusun oleh beberapa hal sebagai berikut :

1. Alur Permaian

Alur permainan pada *game Alien Exterminator* adalah sebagai berikut :

a. Pemain ditempatkan pada *map* disebuah pulau yang ternyata sudah dihuni oleh para *Alien*.
b. Pemain akan mendapatkan *score* ketika *player* musuh terkena tembakan *player*.
c. Pemain harus mengalahkan semua musuh pada setiap *level* untuk bisa lanjut ke *level* selanjutnya.
d. Jika pemain terkena serangan oleh musuh, maka darah / *health point* akan berkurang dan apabila waktu habis maka permainan diulangi ke awal permainan.

Adapun alur sistem dari permainan *game Alien Exterminator* dapat di gambarkan melalui *activity diagram* pada Gambar 3.3:



Alur permainan pada *game Alien Exterminator*

Tabel 3.2 Kekurangan dan Kelebihnan *game alien exterminator*

kekurangan *game Alien Exterminator*
1. Algoritma Pencarian terlihat masih kurang optimal
2. Animasi pergerakan NPC masih kurang baik,
3. Grafis 3D masih kurang baik
4. Tidak ada mini *map*
5. Tidak ada alur cerita atau *prologue* tentang misi yang harus dilakukan.
6. Musuh kurang bervariatif
Kelebihan *game Alien Exterminator*
1. Terdapat *score*.
2. Terdapat *boss* pada setiap *level*-nya

4. *Game Island Defense*

Game Island Defense merupakan *game bergenre action* yang menceritakan seorang kapten kapal laut yang diberi misi untuk mempertahankan pulau dari musuh yang ingin menguasai pulau itu. *Game* ini mempunyai tampilan 3D. beberapa tampilan permainan *Island Defense* dapat dilihat pada gambar 3.3



Gambar 3.3 tampilan Menu Utama *game Island Defense*



Gambar 3.4 Tampilan Permainan *game Island Defense*

a. *Gameplay game Island Defense*

Tabel 3.3 *Gameplay Game Island Defense*

Level	**Deskripsi**
Level 1	*Gameplay* dari *game Island Defense* dimana seorang kapten kapal ditugaskan oleh pemerintah untuk mempertahankan pulau yang berada ditengah laut dari serangan musuh yang ingin merebut sumberdaya yang ada dipulau tersebut. Pemain harus berpatroli berkeliling pulau jika sewaktu-waktu kapal musuh mendekat ke pulau dan pemain harus mengalahkan semua musuh yang datang dan ingin merebut pulau. Di dalam game ini terdapat *score*, *item rocket* dan *healthbar*. Pemain harus mengalahkan semua kapal musuh untuk bisa melanjutkan ke level selanjutnya. yang membedakan level yang satu dengan level yang lain dari jumlah musuh yang harus dikalahkan.
Level 2	Pada level ini tidak beda jauh dengan level sebelumnya, dimana jumlah musuh lebih banyak dan tingkat kesulitan lebih sulit dari level sebelumnya dan harus mempertahankan dua pulau dimana dipulau tersebut terdapat sumberdaya alam yang diinginkan oleh musuh sehingga pemain harus bisa mempertahankan dan mengalahkan musuh yang berusaha masuk ke pulau itu.

Misi utama dari *game* ini adalah dimana seorang kapten kapal yang diberikan misi oleh pemerintah untuk mempertahankan pulau dari musuh yang berusaha ingin merebut pulau dan menguasai sumberdaya yang ada dipulau itu.

b. Skenario *Game*

Skenario dalam permainan *Island Defense* dapat dilihat alur permainannya sebagai berikut :

1. Pemain berada di tengah pulau
2. Pemain harus berpatroli dengan mengelilingi pulau jika tiba-tiba kapal musuh datang.
3. Pemain harus mengalahkan semua kapal musuh yang ingin merebut pulau disetiap *level*-nya untuk bisa melanjutkan ke *level* berikutnya.
4. Pemain akan mendapatkan skor ketika musuh terkena tembakan pemain.
5. Jika permain terkena tembakan musuh, maka *healthpoint* akan berkurang dan apabila healthpoint habis maka permainan berakhir.

Tabel 3.4 Kekurangan dan Kelebihan *Game Island Defense*

Kekurangan *game Island Defense*
1. Tidak ada alur cerita yang menjelaskan tentang misi yang akan dilakukan
2. Tidak ada *lives* atau nyawa
Kelebihan *game Island Defense*
1. Terdapat *score*
2. Terdapat waktu
3. Terdapat *items* pendukung dalam permainan

5. *Game* yang akan dibangun

Game The Defend Island yang akan dibangun yaitu berupa *game* 3D yang ber-*genre action*. *Game* ini mempunyai alur cerita yang sama dengan *game Alien Exterminator* yaitu harus mempertahankan pulau yang mereka tinggali dari alien yang ingin menguasai pulau dan sumber daya yang ada di pulau itu. Dimana tokoh utamanya berupa seorang mantan tentara yang berusaha untuk mempertahankan pulau dari alien yang ingin menguasai pulau beserta sumber daya yang ada di pulau.

Game dibangun dengan menggunakan sudut pandang orang pertama atau *first person shooter* sehingga karakter pemain tidak terlihat dalam permainan. Berikut ini adalah beberapa fitur yang ditawarkan di dalam *game* ini :

1. Grafik *game* 3D
2. *Game* memiliki 3 *Level*
3. Sistem *single player*
4. *Game* ber-*genre First Person Shooter*
5. Karater musuh lebih bervariatif dan memiliki dan diterapkannya AI pada NPC untuk pencarian dan untuk pengambilan keputusan pergerakan dalam kondisi tertentu.

Secara garis besar berikut ini adalah perbandingan *game* sejenis :

Tabel 3.5 Perbandingan *Game* Sejenis

	Game Alien Exterminator 3D	***Game Island Defense***	***Game* yang akan dibangun**
Genre Game	*First Person Shooter*	*First Person Shooter*	*First Person Shooter*
Desain Grafis	3D	3D	3D
Karakter Utama	Warga biasa	Seorang penjaga pulau	Seorang veteran tentara
Pergerakan Animasi	Pergerakan berupa berjalan dan melompat	Karakter hanya berdiam di atas mercusuar yang ada di pulau	Pergerakan karakter berupa, berjalan, berlari, melompat, merunduk dan *reload* senjata
Enemy	Terdapat dua karakter musuh	Terdapat dua karakter musuh	Terdapat tiga karakter musuh yang berbeda
Gameplay	Mempertahankan pangkalan	Pemain mendapatkan misi	Mempertahankan pulau dari alien

	pesawat luar angkasa yang sudah mulai dikuasai oleh alien dan setiap level memiliki laur cerita yang sama	untuk melindungi dan mempertahankan pulau dari serangan kapal musuh yang ingin masuk dan menguasai pulau tersebut dan akan saling tembak-menembak antara *player* dengan kapal musuh	yang ingin menguasai sumberdaya yang ada di pulau. Pemain harus mengalahkan semua musuh dan menghancurkan markas musuh
Items	*Item* darah	*Item* darah dan *item rocket, item bom*.	*Item* darah dan *item* peluru
Basis	*Desktop*	*Mobile*	*Desktop*

Dari tabel *game* sejenis diatas akan diambil poin-poin yang akan diperbaiki pada *game* yang akan dibangun dengan acuan sebagai berikut :

1. *Gameplay*.
2. Objek musuh yang lebih bervariatif.
3. AI pada musuh yang tidak hanya mempunyai kemampuan untuk pencarian jalan (*pathfinding*) tetapi kemampuan seperti menyerang, bertahan atau melarikan diri.

Dalam pembangunan aplikasi ini terdapat kesamaan atau korelasi antara *game* sejenis dengan *game* yang akan dibangun diantaranya adalah :

1. Memiliki kesamaan dari *gameplay*.
2. Memiliki kesamaan dari karakter utama.
3. Memiliki kesamaan dari misi yang harus diselesaikan.
4. Memiliki kesamaan dari genre yaitu FPS (*first person shooter*).

3.1.3 Analisis *Game* yang akan dibangun

3.1.3.1 *Storyline The Defend Island*

Tabel 3.6 *Storyline game The Defend Island*

Level	**Deskripsi**
	Game yang dibangun *single player* atau dengan satu pemain yang menjalankan seorang katakter utama berupa seorang veteran tentara yang bernama marko. *Game* ini menceritakan seorang veteran tentara yang tinggal disebuah desa terpencil di suatu pulau, yang mana konon kabarnya di pulau tersebut terdapat sumber daya yang melimpah. Tiba-tiba datanglah mahluk asing dengan beberapa pesawat ke pulau tersebut yang menginginkan sumber daya yang ada di pulau, dan ada warga desa yang mengetahui keberadaan pesawat asing itu kemudian melaporkan ke kepada desa tentang kejadian itu kemudian kepala desa memberikan misi kepada marko untuk mencari tahu tentang keberadaan pesawat tersebut dan mengusir mengusir keberadaan mahluk asing itu dari pulau.
Level 1	Permainan dimulai dengan cerita dimana marko yang seorang veteran tentara mempunyai misi yang diberikan oleh kepala desa untuk mencari tahu lokasi pesawat yang ada di tengah hutan dengan berinteraksi kepada warga dan juga pentujuk yang ada di dalam permainan. Untuk bisa sampai pada lokasi pesawat, marko harus melewati hutan dan juga goa yang mana ditengah hutan terdapat banyak sekali anjing liar (serigala) yang sering menyerang warga desa sehingga marko harus berhati-hati ketika melewati hutan agar anjing liar tidak menyerangnya dan bisa sampai dengan selamat ke lokasi pesawat asing tersebut. pada level ini yang menjadi tantangannya adalah marko harus bisa bertahan dari serangan anjing liar yang akan menyerangnya sewaktu-waktu.
Level 2	Pada level kedua marko harus masuk ke dalam pesawat *alien*

	dan harus membebaskan seorang karakter teman yang tenyata telah di sandera oleh *alien* di dalam ruangan dan dijaga ketat oleh robot-robot *alien*, sehingga marko harus mengalahkan satu per satu robot-robot penjaga yang menjaga ruangan tempat temannya di sandera. Untuk membebaskan teman marko harus mengalahkan semua robot alien penjaga dan harus mencari kunci agar bisa membuka ruangan tempat temannya disadera, kunci akan di dapat pemain setelah mengalahkan semua robot alien penjaga, kemudian setelah pemain berhasil mendapatkan kunci pemain harus membuka pintu yang tempat dimana temannya dikurung. Setelah membebaskan teman selanjutnya pemain harus menuju ke markas utama musuh bersama karakter teman yang barusan telah dibebaskan dan masih terdapat robot alien dan juga tower senjata yang harus dikalahkan untuk bisa menuju ke markas utama musuh.
Level 3	Setelah berhasil membebaskan karakter teman pemain harus melanjutkan misi berikutnya yaitu mengalahkan dan menghancurkan markas utama alien dengan dibantu oleh karakter teman yang telah di selamatkan pada *level* sebelumnya, pada tingkat akhir ini pemain harus bekerja sama dengan karakter teman untuk menghancurkan markas utama musuh (alien) yang mana pada level ini musuh robot alien memiliki kemampuan seperti menyerang, bertahan maupun melarikan diri sehingga tantangan pada level ini lebih menantang dan lebih sulit dari pada *level* sebelumnya. Dan juga terdapat musuh pesawat patroli alien yang bisa bergerak bebas dan juga menyerang pemain beserta karakter teman dari atas sehingga sangat sulit untuk dikalahkan karena pergerakannya yang sangat cepat juga mempunyai peluru yang tidak terbatas.
	Karakter utama akan mendapatkan *equipment* dan *weapon* yang pada awal permainan ketika kepada desa memberikan misi dan di dalam permainan terdapat beberapa *items* yang akan membantu

	pemain dalam menyelsaikan misi seperti *item healthpack* yang berfungsi untuk menambah darah pemain, dan juga pemain akan menemukan *bulletspack* yang berfungsi untuk menambah peluru pada senjata pemain.

3.1.3.2 *Gameplay The Defend Island*

Tabel 3.7 *Gameplay Game The Defend Island*

Level	**Deskripsi**
	Pada awal permainan terdapat tampilan menu utama yang terdapat beberapa pilihan seperti *play game, controls, credits* dan *quit*. Ketika pemain memilih pemain memilih menu *controls* sistem akan menampilkan petunjuk permainan atau informasi kontrol, kemudian ketika pemain memilih menu *credits* sistem akan menampilkan informasi mengenai pengembang aplikasi dan ketika pemain memilih menu play game sistem akan menampilkan *scene video* atau *prologue*, setelah selsai menampilkan *prologue* pemain dapat menekan tombol spasi untuk menampilkan area permainan.
Level 1	Permainan dimulai dengan karakter utama bernama marko yang diberikan misi untuk mencari tahu lokasi markas *alien* yang ada dipulau dengan mencari petunjuk tentang lokasi pesawat *alien* yang ada dipulau dengan berinteraksi kepada warga setempat dan dikasih waktu, sehingga apabila waktu habis dan player tidak bisa menyelsaikan misi maka misi gagal. Dan *player* harus melewati goa karena lokasi markas dan pesawat berada di balik goa,untuk sampai ke *markas alien player* harus melewati goa yang terdapat di tengah hutan dan di dalam goa suka terdapat anjing liar sehingga *player* harus menghindar dari pengejaran anjing liar yang ada di goa. Di

	dalam permainan pemain akan menemukan *healthpack* yang berfungsi untuk menambah darah pemain.
Level 2	Pada *level* kedua pemain sudah berada di dalam pesawat *alien* yang mana didalam pesawat terdapat karakter teman yang disandera oleh *alien*, jadi pada *level* ini misi pemain harus membebaskan karakter teman yang di sandera di dalam sebuah ruangan di dalam pesawat alien, akan tetapi ruangan tersebut di jaga ketat oleh *robot alien* dan juga terdapat sentrygun disetiap pojok pesawat yang secara otomatis akan menembak pemain jika dalam jangkauannya, pemain harus mencari kunci untuk membebaskan karakter teman yang dikurung di dalam ruangan. Kunci akan di dapat ketika pemain berhasil mengalahkan semua *robot* penjaga dan sentrygun yang berada di ruangan tersebut, ketika pemain sudah berhasil mengalahkan *robot* penjaga maka pemain akan mendapatkan kunci dan dan harus membebaskan karakter teman yang di kurung. Kemudian pemain melanjutkan misinya menuju *level* 3 bersama karakter teman yang barusan sudah dibebaskan.
Level 3	Pada *level* terakhir pemain harus menghancurkan markas utama *alien* yang ada dipulau bersama karakter teman yang sudah di bebaskan pada *level* sebelumnya, pemain harus mengalahkan semua musuh yang muncul, pada *level* ini terdapat 3 musuh berupa *robot alien*, *sentrygun* dan pesawat *alien*. Dari ketiga jenis musuh tersebut mereka mempunyai peluru yang tidak terbatas dan tingkat kesulitan yang berbeda dari *level* sebelumnya. Setelah pemain berhasil mengalahkan semua musuh kemudian pemain harus menghancurkan markas utama *alien* supaya *alien* tidak akan kembali lagi ke pulau itu.
	Pada setiap *level* terdapat *items* seperti *item* darah, *item* peluru. *Item* darah berfungsi untuk menambah darah pada karakter dan *item* peluru berfungsi untuk menambah peluru pada senjata karakter. Dalam

	setiap *level* dibedakan dari jumlah musuh yang harus dikalahkan, ketika pemain terkena tembakan musuh maka darah atau *healthpoint* pemain akan berkurang, kemudian pemain harus mencari item darah supaya darah pemain bertambah, begitu juga ketika peluru dalam senjata pemain habis, pemain harus mencari item peluru untuk menambahkan peluru pada senjata pemain.

3.1.3.3 Deksripsi *Level*

Tingkat kesulitan dari tiap *level* akan di bedakan dari jumlah musuh yang harus dikalahkan dan terdapat tiga jenis musuh berupa *robot alien*, *sentrygun* dan pesawat otomatis *alien*.

Untuk *level* akan dibangun tiga *level*, pada *level* pertama misi yang harus diselsaikan oleh *player* adalah harus mencari tahu lokasi markas *alien* yang berada dipulau dengan mencari petunjuk dan berinteraksi dengan warga, dimana lokasi markas berada ditengah hutan di balik goa dan *player* harus masuk ke goa untuk bisa sampai ke markas musuh. Pada *level* kedua *player* sudah berada di dalam markas musuh dan *player* harus membebaskan karakter teman yang di sandera di sebuah ruangan yang di jaga ketat oleh *robot alien*, kemudian *player* harus mengalahkan semua musuh yang ada di ruangan tersebut untuk membebaskan karakter teman yang dikurung.

Pada *level* ketiga *player* harus mangalahkan semua musuh dan menghancurkan markas musuh dengan dibantu oleh karakter teman yang dibebaskan pada *level* sebelumnya. Tingkat kesulitan setiap *level* dibedakan dari jumlah musuh yang harus dikalahkan.

3.2 Analisis Algoritma

Pada analisis akan dijelaskan tentang algoritma *SMA** dan *fuzy Logic* yang akan digunakan di dalam *game*.

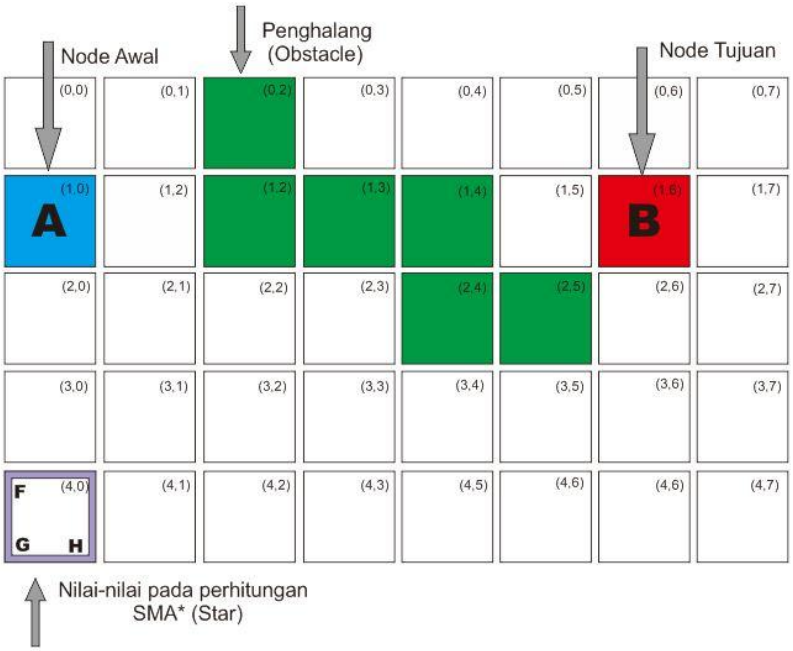
3.2.1 Analisis Algoritma SMA*

Algoritma SMA* di dalam *game* akan diterapkan pada NPC (*non playable character*). sebagai penentuan jalur terpendek atau *pathfinding* menuju *target.* Cara kerja algoritma SMA* di dalam *game The Defend Island* lebih jelasnya sebagai berikut.

Selanjutnya adalah cara melakukan perhitungan untuk menemukan jalan dengan cara menghitung *node-node* yang berdekatan dari *node* A sampai node B. Langkah-langkahnya adalah sebagai berikut :

1. Pertama, dimulai dari *node* A sebagai *node* awal dan *node* B sebagai *node* tujuan.
2. Hitung nilai *heuristic* (nilai sebenarnya) dari *node* A menuju *node* B, kemudian setelah didapat nilai *heuristic*-nya,
3. Bangkitkan semua *node* tetangga (*neighbor node),* kemudian masukan suksesor dari A ke dalam *open list*. Disini suksesor hanya bisa digenerate 2 *node*, karena hanya tersedia 2 *node* yang ada di *open list*.
4. Ketika semua suksesor dari A sudah dibangkitkan, ganti nilai f(A) ganti nilai f-*cost* terkecil dari nilai f-*cost* yang ada pada semua suksesornya.
5. Pilih *node* dengan nilai f-*cost* terkecil sebagai *node* yang akan dicek, *generate* suksesornya dan hitung nilai f-*cost* nya juga. Kemudian ulang langkah no 3 sampai dengan no 4.

Untuk lebih jelasnya algoritma SMA*, bisa dilihat dari contoh kasus. Seperti terlihat pada gambar 3.5. Di dalam *game* yang akan dibangun, karakter ingin bergerak dai *point* A ke *point* B, kedua *point* dipisahkan oleh penghalang. Untuk memudahkan setiap *node* diberi warna, warna biru muda adalah titik mulai A, dan warna merah adalah titik tujuan B, warna hijau tua adalah dinding atau *collider*, warna abu-abu adalah *node* yang berada di *closed list* dan warna kuning adalah *node* yang berada di *open list*, disini diasumsikan tersedia dua *node* untuk *open list*.



Gambar 3.5 Langkah 1 pencarian SMA*

Mulai pencarian dari point A dan tambahkan point A ke list. *Generate suksesor* dari A, kemudian hitung f(s) nya dengan rumus :

$$f(s) = MAX(f(n),g(s)+h(s)) \quad (3.1)$$

Dengan keterangan :

f(s) = fungsi evaluasi *node* yang dicari

f(n) = fungsi evaluasi *node parent*

g(s) = biaya yang sudah dikeluarkan

h(s) = estimasi biaya untuk sampai pada suatu tujuan

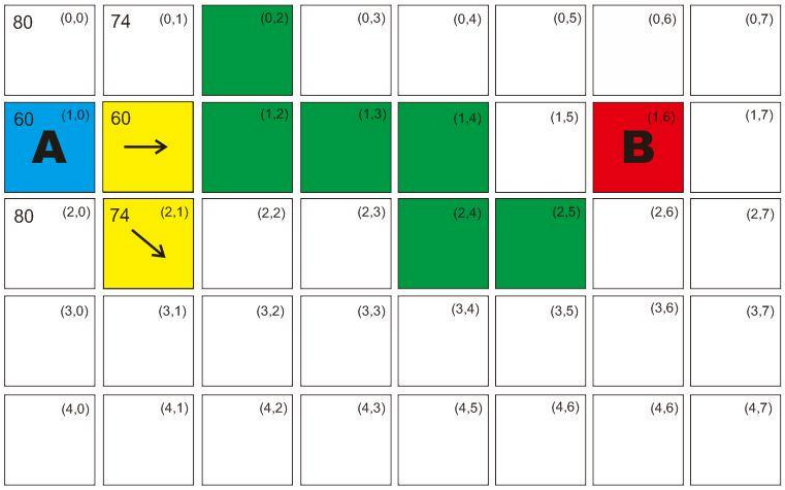
Untuk perhitungan nilai H menggunakan fungsi *heuristic*, metode yang digunakan didalam contoh ini adalah metode *Manhattan distance* yang dirumuskan dengan :

$$H = 10*(abs(currentX-targetX+ abs(currentY-targetY)) \quad (3.2)$$

Sedangkan untuk perhitungan nilai g dilakukan dengan ketentuan, untuk pergerakan *horizontal* dan *vertical*, *cost*-nya adalah 10, sedangkan untuk pergerakan diagonal memakan *cost* 14.

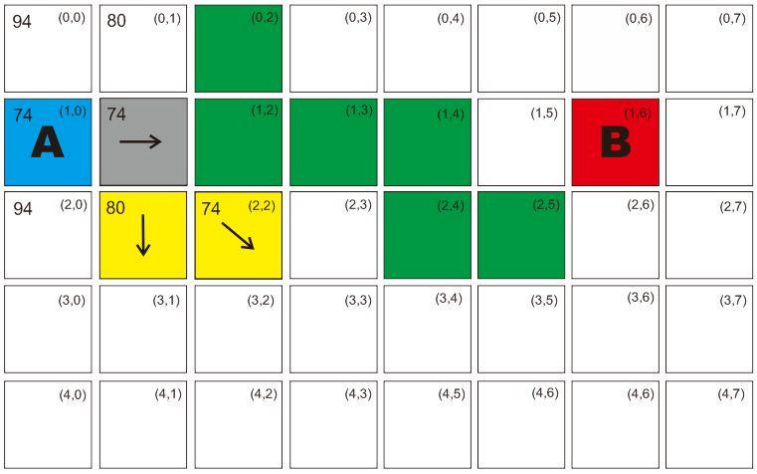
Dengan f(a) = h(a) = 60, karena g(n) = 0 biaya dari *initial state* ke *initial state* = 0, kemudian masukan suksesor ke dalam *open list*, disini suksesor hanya bisa digenerate 2 *node*, karena hanya tersedia 2 di *open list*. Ketika semua suksesor

dari A sudah dibangkitkan ganti nilai f(A) dengan nilai f-*cost* terkecil dari nilai f-*cost* yang ada pada semua suksesornya, karena nilai f-*cost* yang terkecil adalah 60 yang sama dengan f(A) jadi tidak perlu diganti, kemudian masukan A ke *closed list* seperti pada gambar 3.6



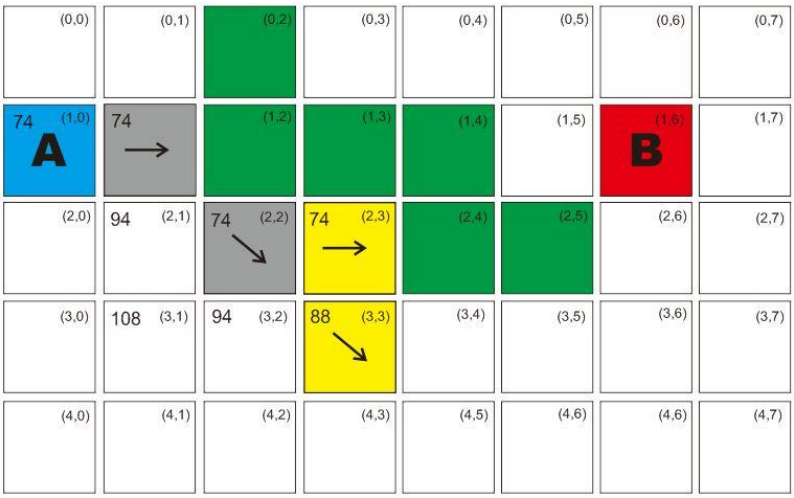
Gambar 3.6 Langkah 2 pencarian SMA*

Langkah selanjutnya adalah pilih *node* dengan *f-cost* terkecil sebagai *node* yang akan dicek, *generate* suksesornya dan hitung f-costnya juga, pada gambar 3.7 *node* terkecil adalah dengan nilai 74.



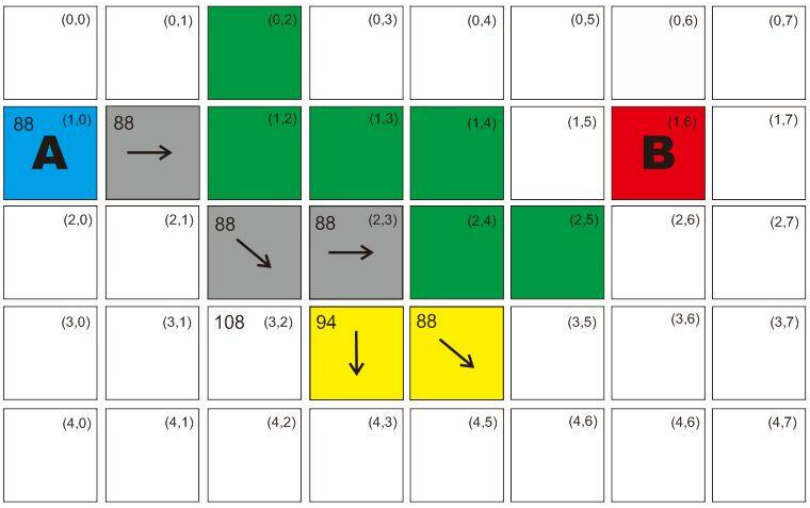
Gambar 3.7 Langkah ke 3 pencarian SMA*

Sama seperti langkah sebelumnya pilih f-*cost* yang terkecil yaitu 60, kemudian *generate* suksesor-nya, didapatkan dua suksesor baru, dan f(s) yang terkecil adalah 74, sehingga f(*best*) dari 60 diganti dengan 74, begitu juga dengan *ancestor*-nya seperti terlihat pada gambar 3.8.



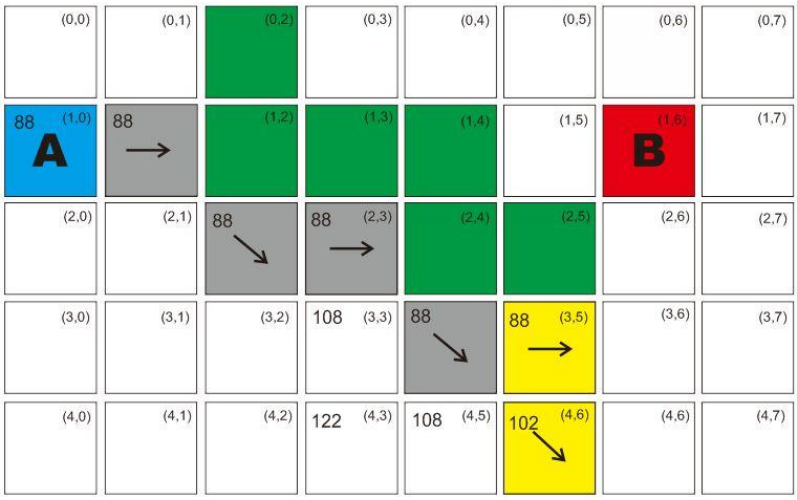
Gambar 3.8 Langkag ke 4 pencarian SMA*

Pilih f-*cost* terkecil 88, langkah selanjut sama seperti diatas, untuk lebih jelasnya bisa dilihat pada gambar 3.9



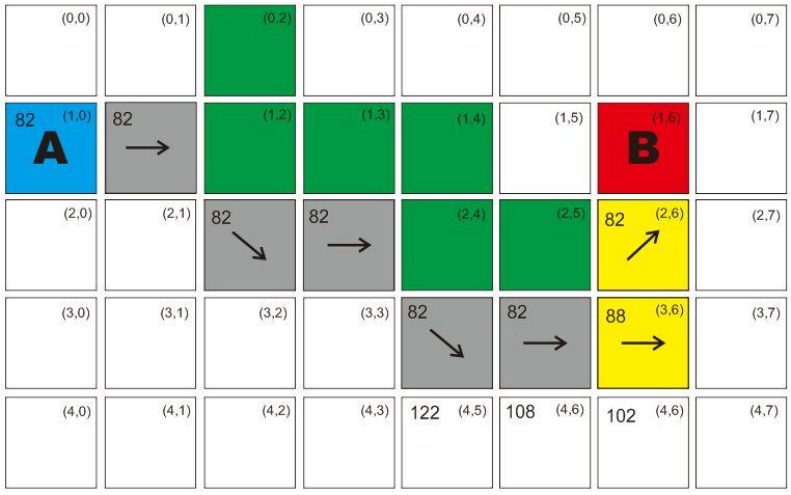
Gambar 3.9 langkag ke 5 pencarian SMA*

Dipilih f-*cost* tekecil 102, didapatkan dua suksesor baru dan ganti nilai f(*best*) dengan f(s) terkecil yaitu 88 seperti yang terlihat pada gambar 3.9.



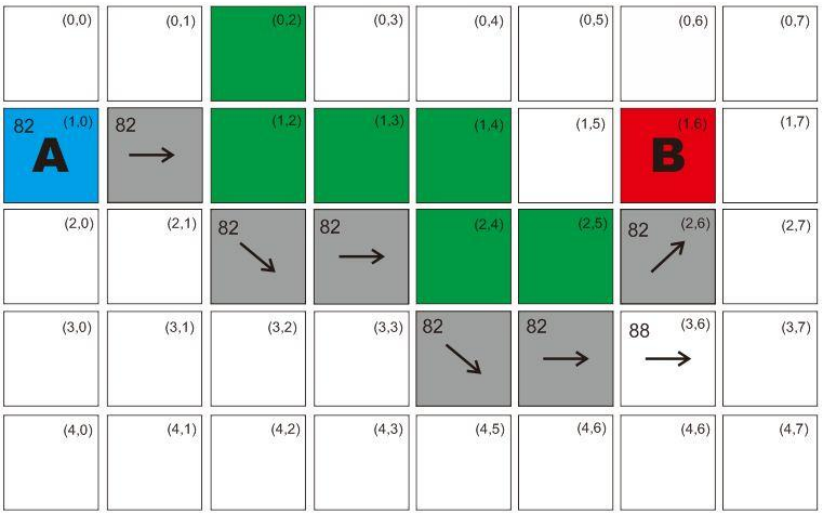
Gambar 3.10 langkag ke 6 pencarian SMA*

Terlihat pada gambar 3.11 dipilih f-*cost* terkecil 88, kemudian didapatkan dua suksesor baru lagi dan ganti nilai f(*best*) dengan nilai f(s) terkecil yaitu 82, seperti yang terlihat pada gambar 3.10.



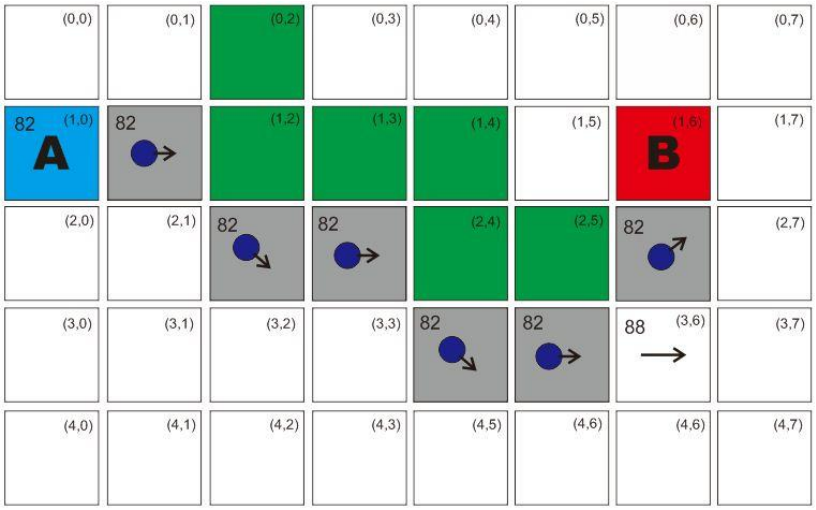
Gambar 3.11 langkag ke 7 pencarian SMA*

Paga gambar 3.11 terlihat nilai f-*cost* terkecil adalah 82 dan didapatkan dua suksesor baru.



Gambar 3.12 langkag ke 8 pencarian SMA*

Node selanjutnya yang terpilih merupakan *node* tujuan, sehingga didapatkan jalur terpendek dari titik A menuju ke titik B, terlihat pada gambar 3.13 yang diberi tanda warna hitam merupakan hasil dari pencarian jalur.



Gambar 3.13 langkag ke 9 pencarian SMA*

Ketika *node* tujuan sudah ditemukan maka tinggal ditarik *path* untuk menuju player dengan menggunakan *backtrack* dan mengikuti *pointer* sebagai penanda jalur terbaik yang sudah didapat dari hasil pencarian.

3.2.2 Analisis Algoritma *Fuzzy Logic*

Fuzzy Logic (logika fuzzy) di dalam *game* akan diterapkan pada NPC (*Non Playable Character*) untuk memberikan *abilities* dalam menentukan keputusan untuk menyerang, bertahan dan melarikan diri. Cara kerja *Fuzzy Logic* di dalam *game The Defend Island* lebih jelasnya sebagai berikut.

Logika fuzzy yang digunakan adalah metode sugeno karena menghasilkan keluaran yang konstan tegas sehingga mewakili nilai perilaku ayng telah dirancang. Analisis *fuzzy logic* tahap-tahapnya adalah sebagai berikut.

1. Variabel Fuzzy

Ada 3 variabel yang digunakan dalam fungsi fuzzy, yaitu sebagai berikut:

1. Variabel jarak sebagai variabel input.
2. Variabel kekuatan (darah/health) sebagai variabel input.
3. Variabel keputusan sebagai variabel output.

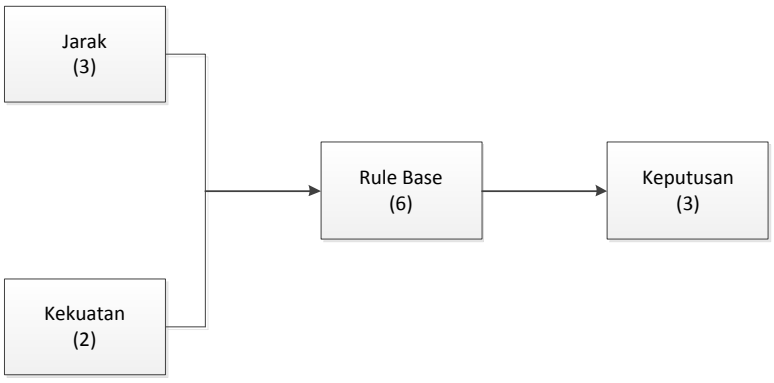
2. Nilai linguistik / *terma*

Dari tiga variabel yang digunakan, memiliki nilai linguistik sebagai berikut:

1. Jarak : Dekat, Sedang, Jauh
2. Kekuatan : Lemah, Sedang, Kuat
3. Keputusan : Melarikan diri, Bertahan, Menyerang.

3. Fuzzyfikasi

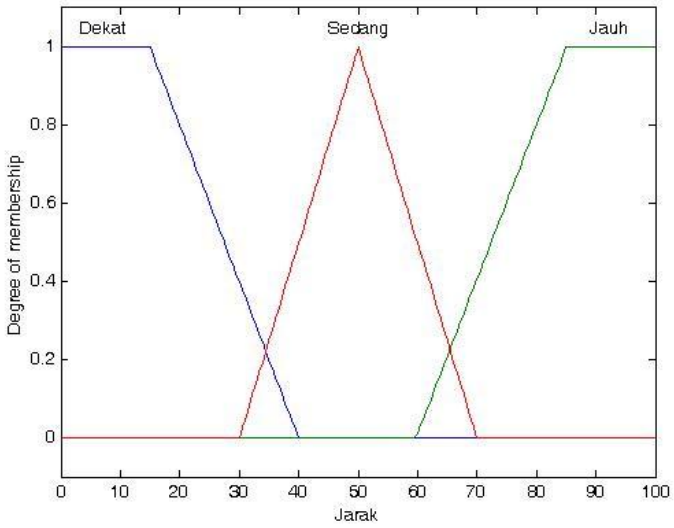
Fuzzyfikasi merupakan proses memetakan nilai *crisp* (numerik) ke dalam himpunan *fuzzy* dan menentukan derajat keanggotaannya. Secara garis besar pemetaan nilai *crisp* ke dalam himpunan *fuzzy* dijelaskan dengan gambar 3.13.



Gambar 3.14 *Fuzzy Interface System*

Berdasarkan *fuzzy interface system* di atas maka pemetaan *input-output* adalah sebagai berikut:

1. Variabel Jarak



Gambar 3.15 Fuzzyfikasi Variabel Jarak

Range nilai untuk variabel jarak diantara 0-10 dengan penjelasan setiap *terma* sebagai berikut

a) *Range* nilai Dekat antara : 0 - 40

b) *Range* nilai Sedang antara : 30 – 70

c) *Range* nilai Jauh antara : 60 – 100

Pada gambar 3.14 menunjukan sebuah grafik yang mempunyai *range* nilai dari 0 – 100, setiap nilai linguistik dari variabel jarak seperti dekat, sedang dan jauh mempunyai nilai fuzzyfikasi yang berbeda-beda. Perhitungan nilai fuzzyfikasi didapatkan dari beberapa fungsi, fungsi yang digunakan pada variabel jarak ada dua fungsi yaitu fungsi Trapesium dan Fungsi Segitiga.

- Variabel linguistik Dekat yang memiliki *range* nilai antara 0 – 40, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut adalah persamaannya :

$$\mu Dekat[x]\begin{cases}1; x \leq 15\\ \dfrac{40-x}{25}\\ 0; x \geq 40\end{cases} ; 15 \leq x \leq 40$$

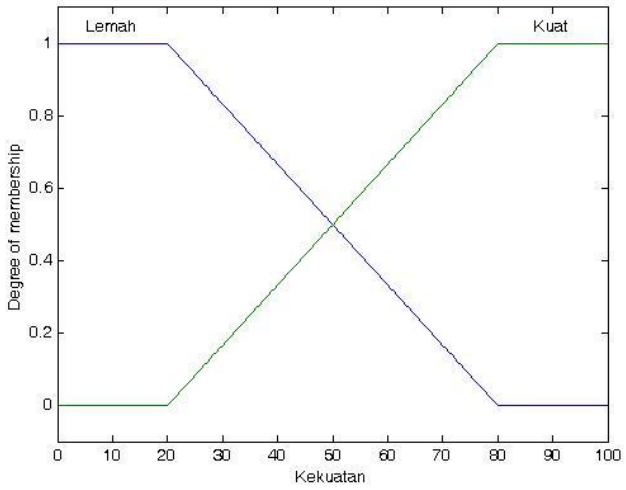
- Variabel linguistik Sedang yang memiliki nilai *range* nilai antara 30 – 70, nilai fuzzyfikasinya dihitung berdasarkan fungsi segitiga, beriktu adalah persamaannya :

$$\mu Sedang[x]\begin{cases}0; x \leq 30/x \geq 70\\ \dfrac{x-30}{20} ; 30 \leq x \leq 50\\ \dfrac{50-x}{20} ; 50 \leq x \leq 70\end{cases}$$

- Variabel linguistik Jauh yang memiliki *range* nilai antara 60 – 100, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut adalah persamaannya :

$$\mu Jauh[x]\begin{cases}0; x \leq 60\\ \dfrac{x-60}{25} ; 60 \leq x \leq 85\\ 1; x \geq 85\end{cases}$$

2. Variabel Kekuatan



Gambar 3.16 Fuzzyfikasi Variabel Kekuatan

Range nilai variabel kekuatan diantara 0 – 100 dengan penjelasan *terma* sebagai berikut:

a) *Range* nilai untuk Lemah antara 0 – 80
b) *Range* nilai untuk Kuat antara 20 – 100

Pada gambar 3.15 menunjukan sebuah grafik trapesium Kekuatan yang mempunyai *range* nilai antara 0 – 100, setiap nilai linguistik dari variabel kekuatan seperti lemah dan kuat mempunyai nilai *fuzzyfikasi* yang berbeda-beda.

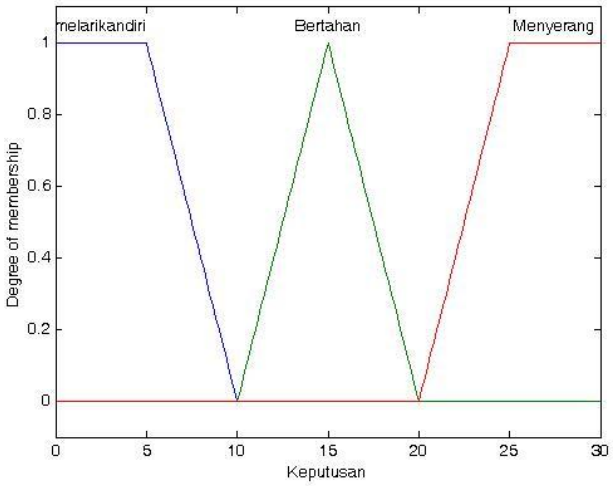
- Variabel linguistik Lemah yang memiliki *range* nilai antara 0 – 80, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut adalah persamaannya :

$$\mu Lemah[y]\begin{cases}1; y \leq 20\\ \dfrac{80-y}{60}\ ; 20 \leq y \leq 80\\ 0; y \geq 80\end{cases}$$

- Variabel linguistik Kuat yang memiliki *range* nilai antara 20 – 100, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut adalah persamaannya :

$$\mu Kuat[y]\begin{cases}0; y \leq 20\\ \dfrac{y-20}{60}\ ; 20 \leq y \leq 80\\ 1; y \geq 80\end{cases}$$

3. Variabel Keputusan



Gambar 3.17 Fuzzyfikasi Variabel Keputusan

Range nilai variabel keputusan diantara 0 - 30 dengan penjelasan *terma* sebagai berikut:

a. *Range* nilai untuk menyerang antara 0 – 10
b. *Range* nilai untuk bertahan antara 10 – 20
c. *Range* nilai unutk melarikan diri antara 20 – 30

Pada gambar 3.16 menunjukan sebuah grafik keputusan yang mempunyai *range* nilai antara 0 – 30, setiap nilai linguistik seperti melarikan diri, bertahan dan menyerang mempunyai nilai fuzzyfikasi yang berbeda-beda.

- Variabel linguistik Melarikan Diri yang memiliki *range* nilai antara 0 – 10, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut adalah persamaannya :

$$\mu MelarikanDiri[z]\begin{cases} 1; z \leq 6 \\ \dfrac{10-z}{4} \; ; 6 \leq z \leq 10 \\ 0; z \geq 10 \end{cases}$$

- Variabel linguistik Bertahan yang memiliki *range* nilai antara 10 – 20, nilai fuzzyfikasinya dihitung berdasarkan fungsi segitiga, berikut adalah persamaannya :

$$\mu Bertahan[z]\begin{cases} 0; z \leq 10/z \geq 20 \\ \dfrac{z-10}{5} \; ; 10 \leq z \leq 15 \\ \dfrac{15-z}{5} \; ; 15 \leq z \leq 20 \end{cases}$$

- Variabel linguistik Menyerang yang memiliki *range* nilai antara 20 – 30, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut adalah persamaannya :

$$\mu Menyerang[z]\begin{cases} 0; z \leq 20 \\ \dfrac{z-20}{4} \; ; 20 \leq z \leq 24 \\ 1; z \geq 24 \end{cases}$$

4. Kaidah *fuzzy* atau *Rule Based*

Kaidah *fuzzy (rules)* atau aturan-aturan yang diterapkan dalam penentuan perilaku Decision Maker ada enam *rules* yaitu:

Tabel 3.8 *Rule-Based* tindakan NPC

	Jarak		**Kekuatan**		**Keputusan**
IF	Dekat	AND	Lemah	THEN	Melarikan diri
IF	Dekat	AND	Kuat	THEN	Menyerang
IF	Sedang	AND	Lemah	THEN	Bertahan
IF	Sedang	AND	Kuat	THEN	Menyerang
IF	Jauh	AND	Lemah	THEN	Melarikan diri
IF	Jauh	AND	Kuat	THEN	Menyerang

5. Agregasi dan Defuzzyfikasi

Agregasi untuk menentukan keluaran semua *rules* yang dikombinasikan menjadi sebuah *fuzzy set* tunggal. *Defuzzyfikasi* merupakan proses pemetaan besaran dari himpunan fuzzy set yang dihasilkan ke dalam bentuk *crisp*.

Defuzzyfikasi pada metode Sugeno adalah dengan menghitung *center of single-ton* atau titik pusat nilai crisp dengan metode rata-rata (*average*).

$$z^* = \frac{\sum \mu i.zi}{\sum \mu i}$$

z^* = titik pusat nilai crisp.

μi = derajat keanggotaan masing-masing variabel.

zi = nilai domain pada variabel linguistik z.

Contoh kasus

Misal, apabila jarak memiliki nilai [35] dan Kekuatan memiliki nilai [65], maka tahapan-tahapan atau proses agregasi yang menentukan keluaran sesuai dengan *rules* yang sudah dibuat akan dijelaskan seperti berikut :

1. Fuzzyfikasi

Perhitungan *Fuzzyfikasi* untuk Jarak dari setiap variabel lingustik (Dekat, Sedang dan Jauh) apabila nilai jarak [35] adalah sebagai berikut :

$$\mu Dekat[35] = \frac{40-35}{25} = 0{,}2$$

$$\mu Sedang[35] = \frac{35-30}{20} = 0{,}25$$

$$\mu Jauh[35] = 35 \leq 60 = 0$$

Dari hasil perhitungan *fuzzifikasi* menggunakan fungsi trapesium untuk mencari nilai derajat keanggotaan, maka didapatkan nilai derajat keanggotaan dari masing-masing variabel lingustik yaitu :

1. Fungsi Derajat Keanggotaan Dekat = 0.2
2. Fungsi Derajat Keanggotaan Sedang = 0.25
3. Fungsi Derajat Keanggotaan Jauh = 0

Perhitungan *Fuzzyfikasi* untuk Kekuatan dari setiap variabel lingustik (Lemah dan Kuat) apabila nilai Kekuatan [65] adalah sebagai berikut :

$$\mu Lemah[65] = \frac{80-65}{60} = 0{,}25$$

$$\mu Kuat[65] = \frac{65-20}{60} = 0{,}75$$

Dari hasil perhitungan untuk mencari nilai fungsi kenaggotaan menggunakan fungsi trapesium, didapatkan nilai fungsi keanggotaan dari masing-masing variabel lingustik yaitu :

1. Fungsi Keanggotaan Lemah = 0,25
2. Fungsi Keanggotaan Kuat = 0,75

2. Implikasi

Setelah nilai derajat keanggotaan didapatkan, maka nilai-nilai derajat tersebut harus dibandingkan sesuai dengan *Rule* yang ada, tujuannya untuk mencari nilai minimum untuk setiap *rule* yang dihitung.

- *Rule* 1

  If (Jarak *is* Dekat) *and* (Kekuatan *is* Lemah) *then* (Perilaku NPC *is* Melarikan diri)

  → Jarak (Dekat) = 0.2 dan Kekuatan (Lemah) = 0.25

  Min (0.2 , 0.25) = 0.2

- *Rule* 2

  If (Jarak *is* Dekat) *and* (Kekuatan *is* Kuat) *then* (Perilaku NPC *is* Menyerang)

  → Jarak (Dekat) = 0.2 dan Kekuatan (Kuat) = 0.75

  Min (0.2 , 0.75) = 0.2

- *Rule* 3

  If (Jarak *is* Sedang) *and* (Kekuatan *is* Lemah) *then* (Perilaku NPC *is* Bertahan)

  → Jarak (Sedang) = 0.25 dan Kekuatan (Lemah) = 0.25

  Min (0.25 , 0.25) = 0.25

- *Rule* 4

  If (Jarak *is* Sedang) *and* (Kekuatan *is* Kuat) *then* (Perilaku NPC *is* Menyerang)

  → Jarak (Sedang) = 0.25 dan Kekuatan (Kuat) = 0.75

  Min (0.25 , 0.75) = 0.25

- *Rule* 5

  If (Jarak *is* Jauh) *and* (Kekuatan *is* Lemah) *then* (Perilaku NPC *is* Melarikan diri)

  → Jarak (Jauh) = 0 dan Kekuatan (Lemah) = 0.25

  Min (0 , 0.25) = 0

- *Rule* 6

  If (Jarak *is* Jauh) *and* (Kekuatan *is* Kuat) *then* (Perilaku NPC *is* Menyerang)

  → Jarak (Jauh) = 0 dan Kekuatan (Kuat) = 0.75

  Min (0 , 0.75) = 0

3. *Defuzzy*

- *Rule* 1 = Melarikan Diri

$$\frac{10 - z}{4} = 0.2$$

Z = 10 – 1 = 9,2

- *Rule* 2 = Menyerang

$$\frac{z - 20}{4} = 0.2$$

Z = 20 – 0,8 = 19,2

- *Rule* 3 = Bertahan

$$\frac{z - 10}{5} = 0.25$$

Z= 10 – 1.25 = 8.75

- *Rule* 4 =Menyerang

$$\frac{z - 20}{4} = 0.25$$

Z = 20 – 1 = 19

- *Rule* 5 = Melarikan Diri

$$\frac{10 - z}{4} = 0$$

Z = 10 – 0 = 10

- *Rule* 6 = Menyerang

$$\frac{z-20}{4}=0$$

Z = 20 – 0 = 20

$$z^*=\frac{\sum \mu i.zi}{\sum \mu i}$$

$$Z=\frac{0{,}2 \text{ x } 9{,}2+0{,}2 \text{ x } 19{,}2+0{,}25 \text{ x } 8{,}75+0{,}25 \text{ x } 19+0 \text{ x } 10+0 \text{ x } 20}{0{,}2+0{,}2+0{,}25+0{,}25+0+0}$$

$$=\frac{12{,}617}{0{,}9}$$

Z* = 14,0188

Jadi Keputusan yang dihasilkan berdasarkan dari nilai jarak [35] dan Kekuatan [65] adalah **bertahan**. Sesuai dengan nilai z* yang didapatkan yaitu **14,0188**.

3.3 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional menggambarkan kebutuhan luar sistem yang diperlukan untuk menjalankan aplikasi yang dibangun. Adapun kebutuhan non-fungsional pada *game* FPS *The Defend Island* meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pemain sistem yang akan memakai aplikasi. Analisis kebutuhan non-fungsional bertujuan agar aplikasi yang dibangun dapat digunakan sesuai dengan kebutuhan.

3.3.1 Analisis Kebutuhan Perangkat Lunak

Game Action The Defend Island memiliki kebutuhan perangkat lunak sebagai berikut :

1. Aplikasi *game* yang dihasilkan berbasis *desktop*.
2. Pemodelan dan perancangan sistem dalam penulisan laporan ini menggunakan pemodelan berbasis objek dengan *tools* UML (*Unified modeling Language*).

3. Dikembangkan dengan menggunakan *tools game engine unity 3D*, dengan bahasa pemrograman berupa *C#*, *Javascripts*, dan *Boo.*
4. Rendering gambar 3D menggunakan 3ds Max, dan gambar 2D menggunakan *Adobe Photoshop CS5*, sedangkan suara/*sound effect* menggunakan *Adobe After Effect*.

Dari hasil pengamatan perangkat lunak, terdapat beberapa hal penting yang harus diperhatikan agar sistem yang dirancang dapat berjalan dengan baik dan sesuai dengan yang diharapkan. Pengguna diharapkan dapat memenuhi kebutuhan perangkat lunak yang telah direkomendasikan agar aplikasi dapat berjalan dengan baik.

3.3.2 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.

Spesifikasi minimum Perangkat keras untuk pengembang dalam membangun aplikasi adalah sebagai berikut :

Tabel 3.9 Analisis Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi / keterangan
1	*Processor*	Intel Pentium 4 1.8 GHz
2	*VGA*	256 MB
3	*RAM*	1 GB
4	*Harddisk*	3 GB
5	*Monitor*	10” (1024 x 768)
6	*Mouse*	Standar
7	*keyboard*	Standar

Perangkat keras untuk pengembang dalam membangun *game action The Defend Island* ini adalah sebagai berikut :

Tabel 3.10 Analisis Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi / keterangan
1	Processor	Dual Core 2.10 GHz
2	VGA	512 MB
3	RAM	4 GB
4	Harddisk	500 GB
5	Monitor	19” (1024 x 768)
6	Mouse	Standar
7	keyboard	Standar

3.3.3 Analisis Pengguna

Pada analisis *user* (pemakai) ini akan mencakup analisis beberapa parameter terhadap calon *user* dari aplikasi.

1. *User Knowledge and Experience* dari target user yang akan menggunakan *game*.

Game the defend island ini bisa digunakan oleh kalangan apapun, tetapi pengetahuan dan pengalaman akan memudahkan *user* dalam penggunaanya. Terutama pengetahuan dan pengalaman dalam memainkan *game action* atau *first person shooter*. Berikut ini klasifikasi *knowledge and experience* dari pengguna applikasi:

Tabel 3.11 Analisis klasifikasi *knowledge and experience*

Educational level	**Reading Level**	**Typing Skills**
Game ini bisa digunakan oleh berbagai kalangan, seperti pelajar, mahasiswa hingga	*Game* ini bisa digunakan oleh berbagai level pendidikan dengan reading level yang	*Game* ini tidak memerlukan typing skills yang tinggi

masyarakat awam	sedang	
Computer Literacy	**Task Experience**	**System Experience**
Game ini bisa digunakan oleh pengguna yang memiliki kemampuan komputer yang *moderate* (menengah)	*Game* ini bisa digunakan oleh dengan pengalaman penggunaan *computer* dan *game* yang sedang	*Game* ini bisa digunakan oleh dengan pengalaman penggunaan komputer dan *game* yang sedang
Application Experience	**Native Language**	**Use Of Other System**
Game ini bisa digunakan dalam semua sistem operasi *Windows xp/vista/7*	*Game* ini menggunakan dua bahasa, yakni Indonesia dan Inggris	*Game* ini bisa di jalankan tanpa perlu install aplikasi lain

2. *Users Physical Characteristic*

Keadaan fisik seseorang mungkin akan berpengaruh pada penggunaan aplikasi *game* ini. Ada hal-hal yang harus diperhatikan juga terhadap *user* dari karakteristik fisiknya untuk dapat menggunakan aplikasi ini yaitu, *Color Blind*, *Handednes*, dan *Gender*.

Tabel 3.12 Analisis *User Physical Charateristic*

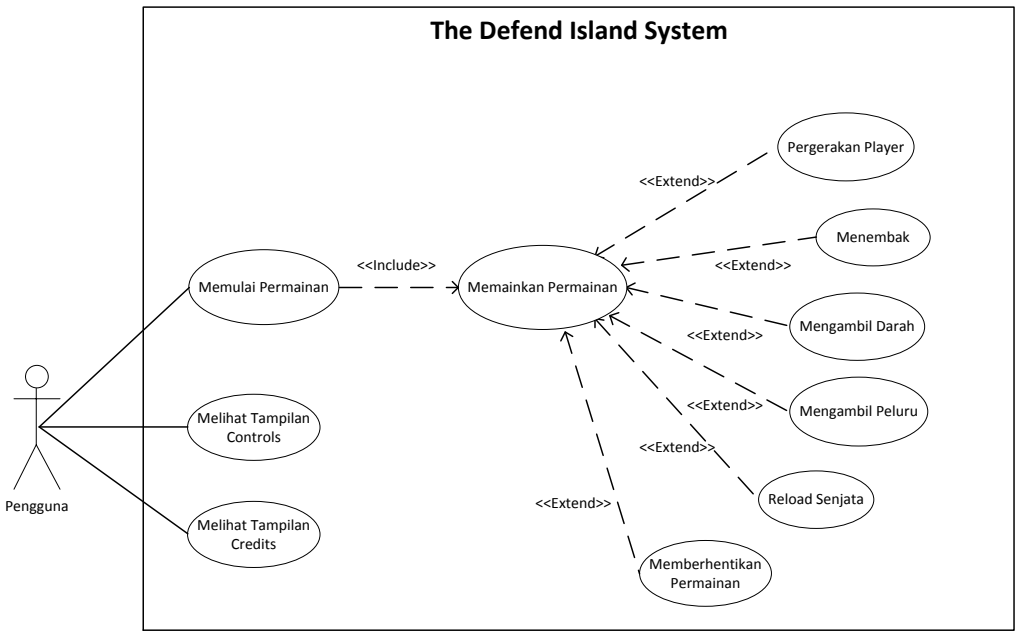
Age	17 tahun ke atas
Gender	Pria dan Wanita
Handedness	Kanan dan Kiri
Color Blind	*User* yang tidak bisa membedakan warna yang satu dengan yang lainnya (buta warna) masih mampu menggunakan aplikasi ini, karena tidak ada indicator warna-warna khusus yang membedakan antara fungsional yang satu dengan fungsional yang lainnya. Akan tetapi penggunaannya tidak akan optimal karena dalam *game* ini terdapat banyak sekali perbedaan warna yang menunjang interaksi dan ketertarikan dalam permainan.

3.4 Analisis 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.

3.4.1 *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 action The Defend Island* :



Gambar 3.17 *Use Case Diagram*

5. Definisi *Actor*

Definisi *Actor* berfungsi untuk menjelaskan *actor* yang terdapat pada *use case diagram*. Definisi *actor* diterangkan pada tabel 3.11.

Tabel 3.13 Definisi *Actor*

No	*Actor*	Deskripsi
1.	Pengguna	Pemain atau orang yang memainkan *game.*

6. Definisi *Use Case*

Definisi *use case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *use case*. Definisi *use case* diterangkan pada tabel 3.5 :

Tabel 3.14 *Definisi Use Case*

No.	*Use Case*	Deskripsi
1.	Memulai Mulai	Proses untuk memulai permainan.
2.	Memainkan Mainkan	Proses untuk Memainkan Permainan
3.	Melihat Tampilan Petunjuk	Proses untuk melihat petunjuk permainan dan informasi kontrol.
4.	Melihat Tampilan *Credits*	Proses untuk melihat informasi pembuat *game.*

5.	Pergerakan Player	Proses untuk menggerakkan karakter dengan animasi
6.	Menembak	Proses untuk menggerakkan karakter dengan animasi menembak
7.	Mengambil Darah	Proses untuk mengambil darah
8.	Mengambil Peluru	Proses untuk mengambil peluru
9.	*Reload* Senjata	Proses untuk mengisi peluru
10.	Memberhentikan Permainan	Proses untuk memberhentikan permainan

1. *Scenario Use Case*

Skenario setiap bagian pada *use case* menunjukkan proses apa yang terjadi pada setiap bagian di dalam *use case* tersebut, dimana pengguna memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada pengguna setelah pengguna memberikan perintah pada setiap bagian - bagian *use case*.

Tabel 3.15 *Scenario Use Case* Memulai Permainan

Identifikasi	
Nomor	1
Nama	Memulai Permainan
Tujuan	Memulai Permainan
Deskripsi	Proses Memulai Permainan
Aktor	Pemain atau Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di Menu Utama
Aksi Aktor	Reaksi Sistem
1. Pemain memilih *menu play game*	
	2. Menampilkan *scene video*
	3. Menampilkan informasi misi

	4. Menampilkan permainan *level* 1
Kondisi Akhir	Level 1 di mainkan.

Tabel 3.16 *Scenario Use Case* Melihat Tampilan Petunjuk

Identifikasi	
Nomor	2
Nama	Melihat Tampilan *Controls*
Tujuan	Menampilkan Tampilan *Controls* atau Petunjuk Permainan
Deskripsi	Proses Melihat Tampilan *Controls*
Aktor	Pemain atau Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di Menu Utama
Aksi Aktor	Reaksi Sistem
1. Pemain memilih *menu controls*	
	2. Sistem akan menampilkan halaman *controls* permainan
Kondisi Akhir	Menampilkan halamana *controls* atau petunjuk permainan

Tabel 3.17 *Scenario Use Case* Melihat Tampilan *Credits*

Identifikasi	
Nomor	3
Nama	Melihat Tampilan *Credits*
Tujuan	Menampilkan Tampilan *Credits*
Deskripsi	Proses Melihat Tampilan *Credits*
Aktor	Pemain atau Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di Menu Utama
Aksi Aktor	Reaksi Sistem
1. Pemain memilih *menu*	

credits	
	2. Sistem akan menampilkan halaman *credits*
Kondisi Akhir	Menampilkan halamana *credits*

Tabel 3.18 *Scenario Use Case* Memainkan Permainan

Identifikasi	
Nomor	4
Nama	Memainkan Permainan
Tujuan	Memainkan Permainan
Deskripsi	Proses Memainkan Permainan
Aktor	Pemain atau Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di Menu Utama
Aksi Aktor	Reaksi Sistem
	1. Menampilkan Arena Permainan
2. Memainkan Permainan	
	3. Menjalankan Permainan
	4. Gagal menyelsaikan misi permainan, pemain mengulang permainan
	5. Berhasil menyelsaikan misi permainan
	6. Tampilan misi selesai
Kondisi Akhir	Level 1 di mainkan.

Tabel 3.18 *Scenario Use Case* Pergerakan *Player*

Identifikasi	
Nomor	5
Nama	Pergerakan *Player*

Tujuan	Pemain bermain game dengan menggerakan player
Deskripsi	Proses Memainkan Permainan
Aktor	Pemain atau Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama

Aksi Aktor	Reaksi Sistem
1. Pemain menekan tombol W pada *keyboard*	
	2. Karakter akan bergerak ke depan
3. Pemain menekan tombol A pada *keyboard*	
	4. Karakter akan berkerak ke samping kiri
5. Pemain menekan tombol S pada *keyboard*	
	6. Karakter akan bergerak ke belakang
7. Pemain menekan tombol D pada *keyboard*	
	8. Karakter akan bergerak ke samping kanan
9. Pemain menekan tombol W + *SHIFT* pada *keyboard*	
	10. Karakter akan berlari ke depan
11. Pemain menekan tombol A + *SHIFT keyboard*	
	12. Karakter akan berlari ke samping kiri
13. Pemain menekan tombol S + *SHIFT keyboard*	

	14. Karakter akan berlari ke belakang
15. Pemain menekan tombol D + *SHIFT keyboard*	
	16. Karakter akan berlari ke samping kanan
Kondisi Akhir	Sistem pergerakan *player* menjalankan animasi berjalan ke depan, kiri, belakang, dan ke kanan sesuai dengan inputan pemain.

Tabel 3.19 *Scenario Use Case* Menembak

Identifikasi	
Nomor	6
Nama	Menembak
Tujuan	Menggerakan karakter dengan animasi menembak
Deskripsi	Proses menembak
Aktor	Pemain atau Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di dalam permainan
Aksi Aktor	Reaksi Sistem
1. Pemain menekan tombol kiri *mouse* atau menekan *CTRL* pada *keyboard*	
	2. Sistem akan menjalankan animasi menembak
Kondisi Akhir	Sistem akan menggerakan karakter dengan animasi menembak sesuai dengan inputan pemain

Tabel 3.20 *Scenario Use Case* Mengambil Darah

Identifikasi		
Nomor	7	
Nama	Mengambil darah atau *health*	
Tujuan	Menambahkan darah pada karakter	
Deskripsi	Proses menambahkan darah pada karakter	
Aktor	Pemain atau Pengguna	
Skenario Utama		
Kondisi Awal	Pemain berada di Menu Utama	
Aksi Aktor		Reaksi Sistem
1. Pemain menggerakan karakter dan berjalan menuju item darah		
		2. Sistem akan menambahakan darah untuk karakter
Kondisi Akhir	Sistem akan menampilkan darah karakter bertambah	

Tabel 3.21 *Scenario Use Case* Mengambil Peluru

Identifikasi		
Nomor	8	
Nama	Mengambil peluru	
Tujuan	Menambahkan peluru pada sejnata karakter	
Deskripsi	Proses menambahkan peluru pada senjata karakter	
Aktor	Pemain atau Pengguna	
Skenario Utama		
Kondisi Awal	Pemain berada di Menu Utama	
Aksi Aktor		Reaksi Sistem
1. Pemain menggerakan karakter dan berjalan menuju item peluru		

	2. Sistem akan menambahakan peluru pada senjata karakter
Kondisi Akhir	Sistem akan menampilkan peluru pada senjata karakter bertambah

Tabel 3.22 *Scenario Use Case* Mengisi Peluru

Identifikasi	
Nomor	9
Nama	Mengisi peluru
Tujuan	Menggerakan karakter dengan animasi mengisi peluru
Deskripsi	Proses mengisi peluru
Aktor	Pemain atau Pengguna
Skenario Utama	
Kondisi Awal	Pemain berada di Menu Utama
Aksi Aktor	Reaksi Sistem
1. Pemain menggerakan karakter dengan menekan tombol R pada *keyboard*	
	2. Sistem akan mengisi peluru atau *reload*
Kondisi Akhir	Sistem menggerakan karakter dengan animasi mengisi peluru sesuai inputan

Tabel 3.23 *Scenario Use Case* Memberhentikan Permainan

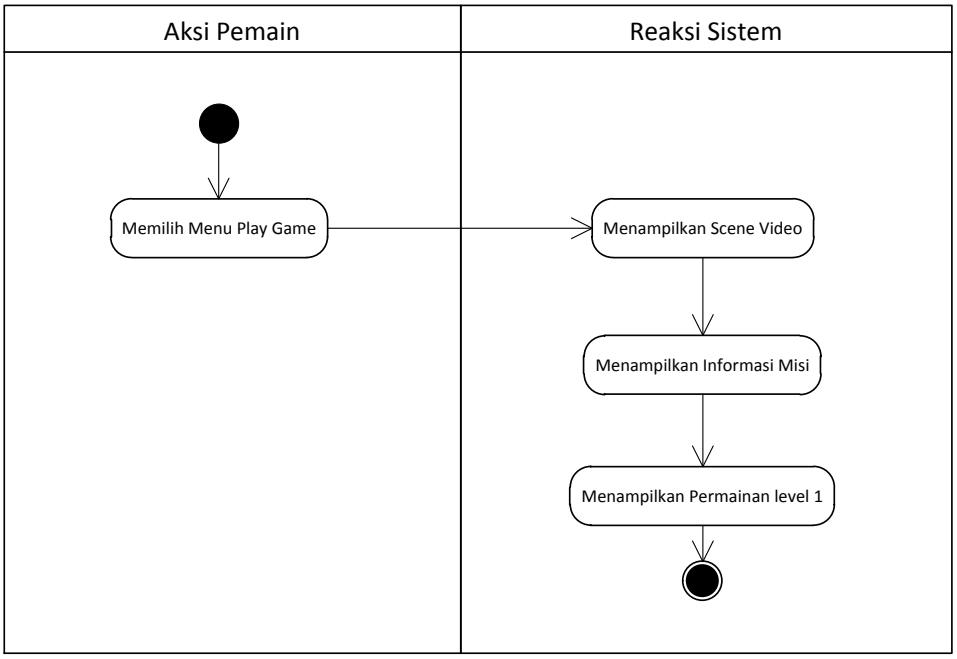
Identifikasi	
Nomor	10
Nama	Memperhentikan Permainan
Tujuan	Memberhentikan Permainan untuk sementara
Deskripsi	Proses memberhentikan permainan
Aktor	Pemain atau Pengguna

<table>
<tr><th colspan="3">Skenario Utama</th></tr>
<tr><td>Kondisi Awal</td><td colspan="2">Pemain berada di Menu Utama</td></tr>
<tr><td colspan="2">Aksi Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td colspan="2">1. Pemain menggerakan karakter dengan menekan tombol ESC pada keyboard</td><td></td></tr>
<tr><td colspan="2"></td><td>2. Sistem akan menampilkan halaman “pausemenu”.</td></tr>
<tr><td colspan="2">3. Pemain memilih menu “continue” untuk kembali ke arena permainan.</td><td></td></tr>
<tr><td colspan="2">4. Pemain memilih menu “restartlevel”</td><td></td></tr>
<tr><td colspan="2"></td><td>5. Sistem akan mengulang permainan pada level tersebut.</td></tr>
<tr><td colspan="2">6. Pemain memilih menu “mainmenu”</td><td></td></tr>
<tr><td colspan="2"></td><td>7. Sistem akan menampilkan menu utama.</td></tr>
<tr><td>Kondisi Akhir</td><td colspan="2">Sistem memberhentikan permainan akan menampilkan halaman menu jeda atau pausemenu. Jika pemain memilik menu continue, maka sistem akan kembali ke arena permainan, jika pengguna memilih menu restartlevel, maka sistem akan menampilkan ulang arena permainan dari awal, jika pemain memilih menu main menu, maka sistem akan menampilan menu utama permainan.</td></tr>
</table>

3.4.2 *Activity Diagram*

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang. Berikut ini beberapa *activity diagram* yang terdapat pada game ini yaitu :

A. *Activity Diagram* Proses Memilih Memulai Permainan

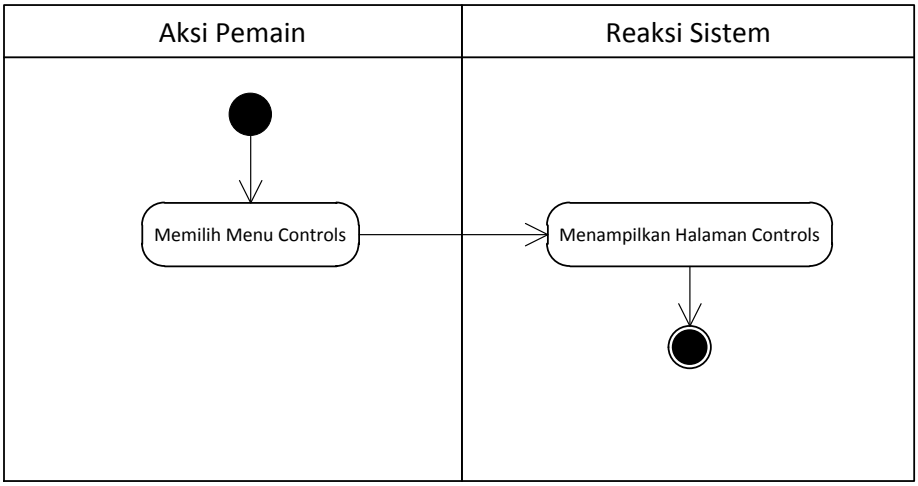


Gambar 3.18 *Activity Diagram* Proses Memulai Permainan

Tabel 3.24 Penjelasan Proses Memulai Permainan

Deskripsi	Menu *Start* merupakan menu yang digunakan untuk mulai permainan.
Normal *Flow*	1. Pengguna memilih menu *start Game* pada menu utama 2. Sistem menampilkan *scene video* pengenalan cerita sebelum masuk ke permainan 3. Pengguna memainkan permainan level 1 4. Sistem menjalankan permainan level 1

B. *Activity Diagram* Proses Melihat Tampilan *Controls*

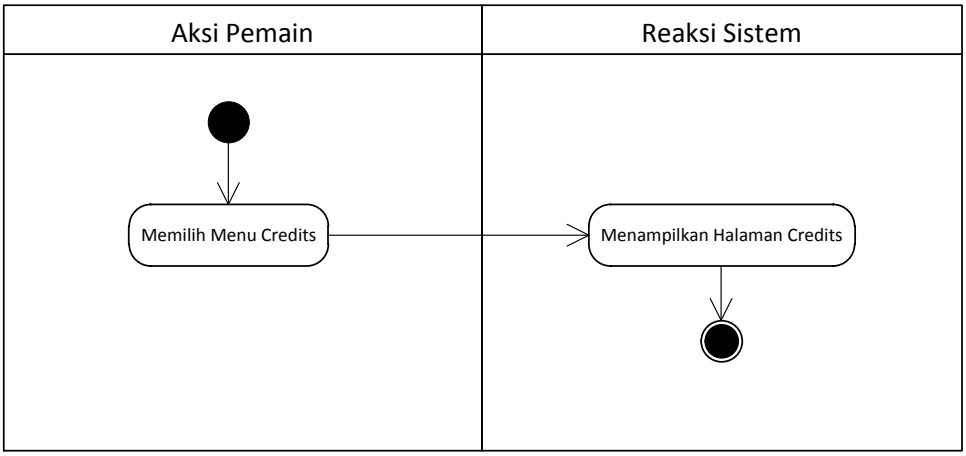


Gambar 3.19 *Activity Diagram* Proses Melihat Tampilan Pentunjuk

Tabel 3.25 Penjelasan Proses Tampilan Petunjuk

Deskripsi	Menu informasi control digunakan pemain untuk melihat petujuk permainan.
Normal Flow	1. Pengguna memilih informasi control 2. Sistem menampilkan halaman pentunjuk permainan

C. *Activity Diagram* Proses Melihat Tampilan *Credits*

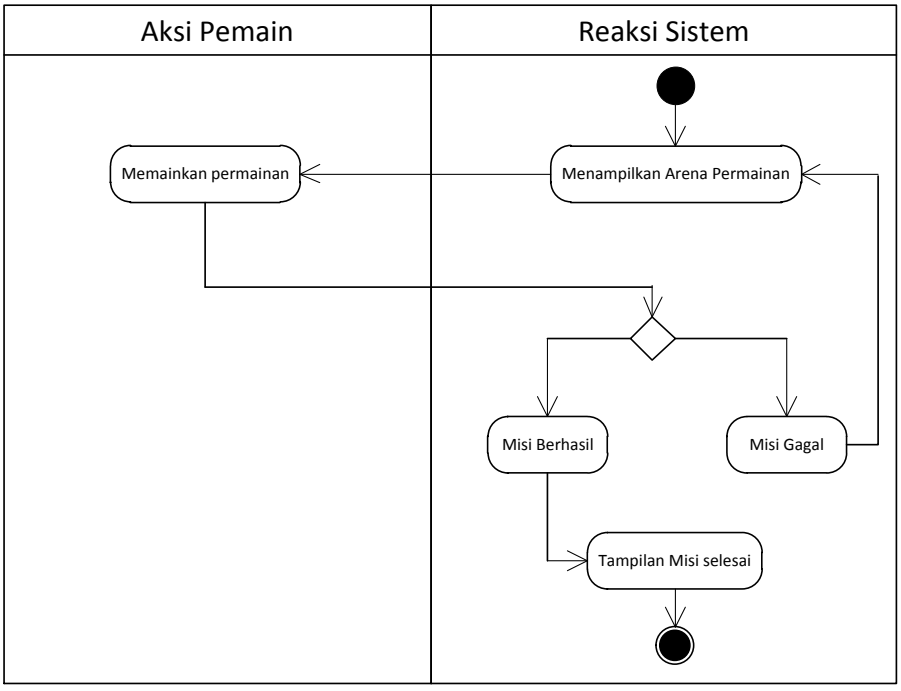


Gambar 3.20 *Activity Diagram* Proses Melihat Tampilan *Credits*

Tabel 3.26 Penjelasan Proses Melihat Tampilan *Credits*

Deskripsi	Menu Credits digunakan untuk melihat informasi pembuat game
Normal *Flow*	1. Pengguna memilih menu *credits* 2. Sistem menampilkan halaman *credits*.

D. *Activity Diagram* proses Memainkan Permainan

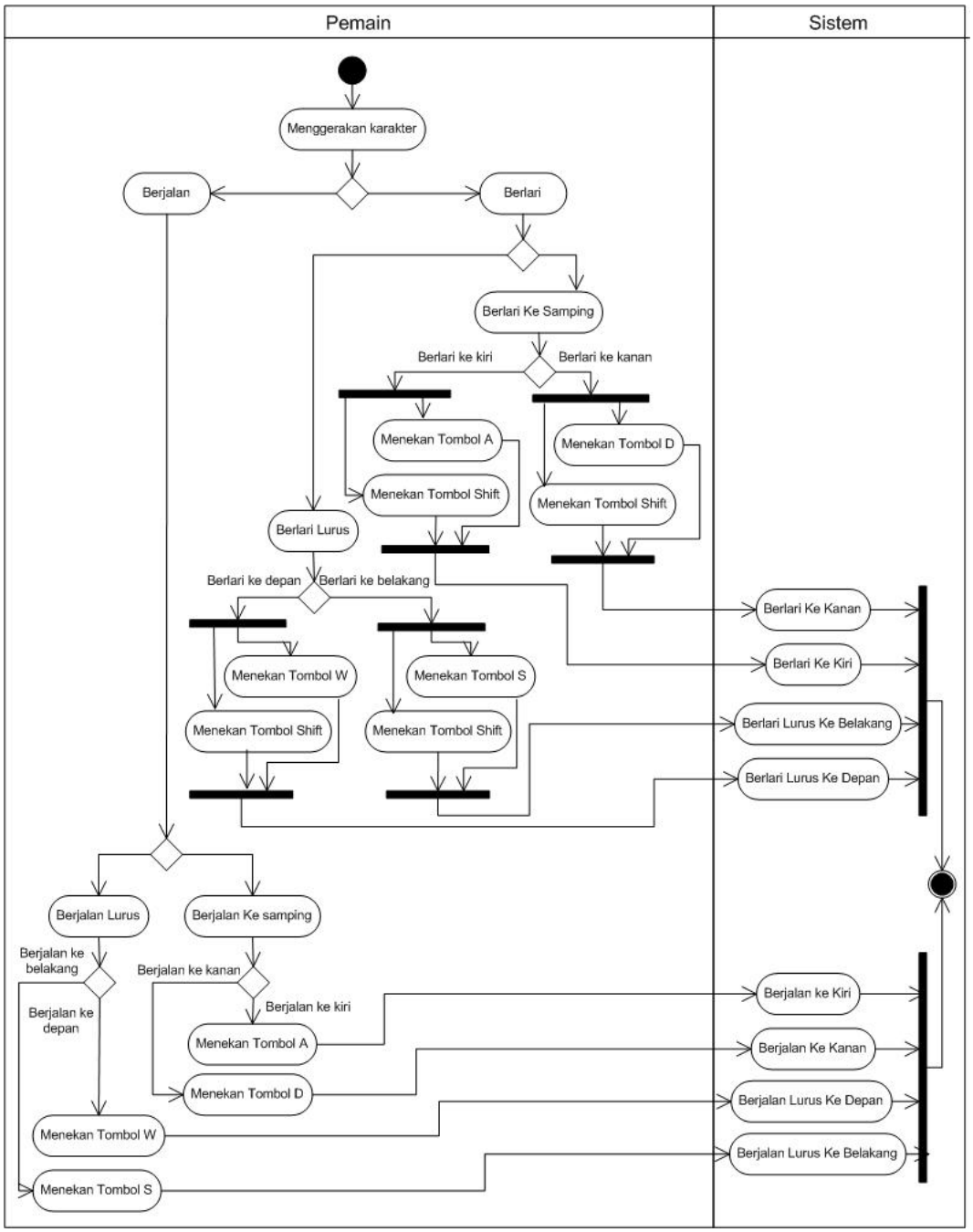


Gambar 3.19 *Acitvity Diagram* Proses Memainkan Permainan

Tabel 3.25 Penjelasan Proses Memainkan Permainan

Deskripsi	Menu *Start* merupakan menu yang digunakan untuk mulai permainan.
Normal *Flow*	1. Sistem menampilkan arena permainan 2. Pengguna Memainkan Permainan 3. Jika pemain gagal menyelsaikan misi, sistem akan menampilkan ulang permainan 4. Jika pemain berhasil menyelsaikan misi, sistem akan menampilkan tampilan misi selesai.

E. *Activity Diagram* Proses Pergerakan *Player*

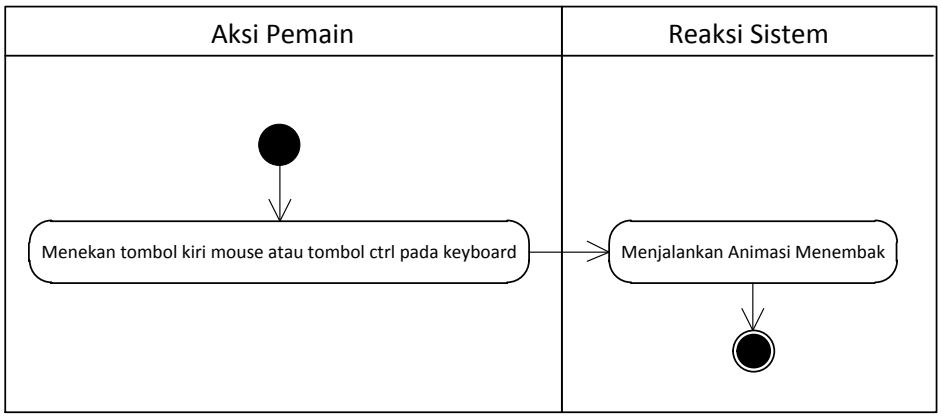


Gambar 3.22 *Activity Diagram* Proses Menggerakan *Player*

Tabel 3.27 Penjelasan Proses Menggerakan *Player*

Deskripsi	Proses untuk menggerakan karakter atau tokoh utama yang dimainkan dalam permainan menggunakan tombol *keyboard.*
Normal Flow	1. Pengguna menggerakan karakter dengan interaksi menggunakan *keyboard* menekan tombol W untuk berlari ke depan
	2. Pengguna menggerakan karakter dengan interaksi menggunakan *keyboard* menekan tombol A untuk berlari ke kiri
	3. Pengguna menggerakan karakter dengan interaksi menggunakan *keyboard* menekan tombol D untuk berlari ke kanan
	4. Pengguna menggerakan karakter dengan interaksi menggunakan *keyboard* menekan tombol S untuk berlari ke belakang
	5. Pengguna menggerakan karakter dengan interaksi menggunakan *keyboard* menekan tombol Shift dan W untuk berjalan ke depan
	6. Pengguna menggerakan karakter dengan interaksi menggunakan *keyboard* menekan tombol Shift dan A untuk berjalan ke kiri
	7. Pengguna menggerakan karakter dengan interaksi menggunakan *keyboard* menekan tombol Shift dan D untuk berjalan ke kanan
	8. Pengguna menggerakan karakter dengan interaksi menggunakan *keyboard* menekan tombol Shift dan S untuk berjalan ke belakang

F. *Activity Diagram* Proses Menembak

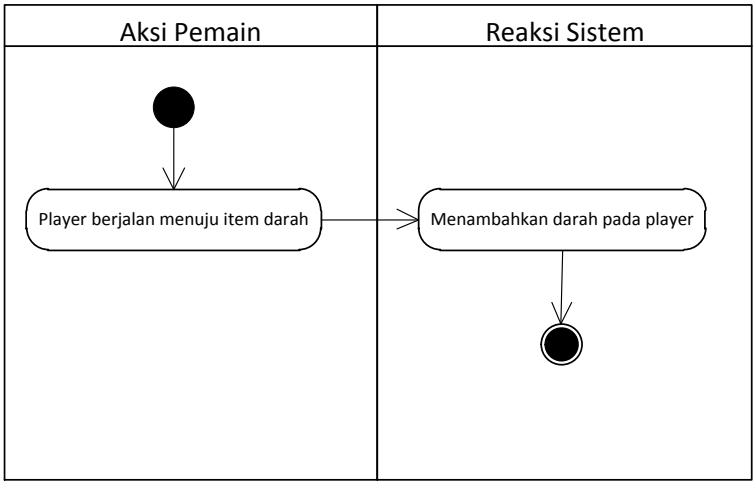


Gambar 3.23 *Activity Diagram* Proses Menembak

Tabel 3.28 Penjelasan Proses Menembak

Deskripsi	Proses untuk menggerakan karakter atau tokoh utama yang dimainkan dalam permainan menggunakan tombol *keyboard.*
Normal Flow	1. Pengguna menggerakkan karakter untuk animasi menembak dengan interaksi menggunakan tombol kiri *mouse* atau tombol CTRL pada *keyboard*. 2. Sistem menggerakkan karakter menjalankan animasi menembak.

G. *Activity Diagram* Proses Mengambil Darah

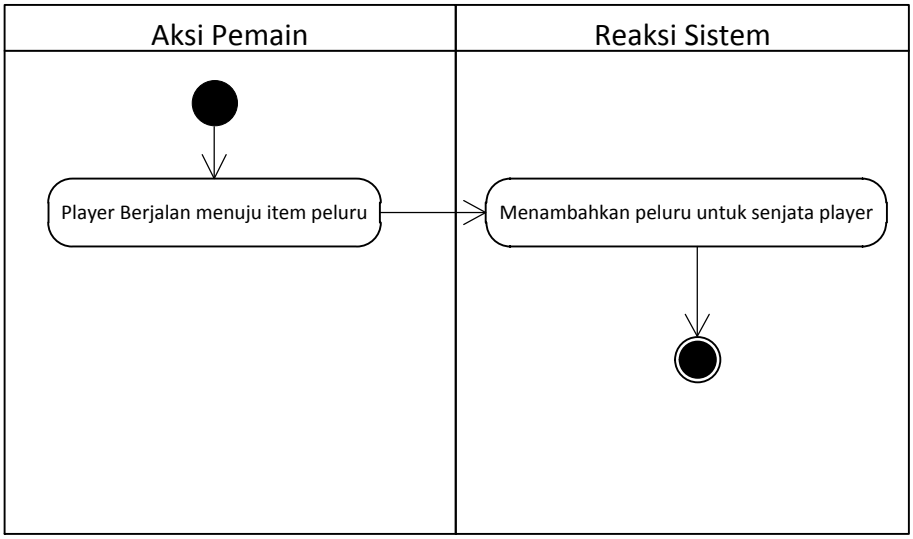


Gambar 3.26 *Activity Diagram* Proses Mengambil Darah

Tabel 3.29 Penjelasan Proses Mengambil Darah

Deskripsi	Proses mengambil darah untuk karakter
Normal Flow	1. Pengguna menggerakan karakter dan berjalan menuju item darah 2. Sistem menambahkan darah untuk karakter

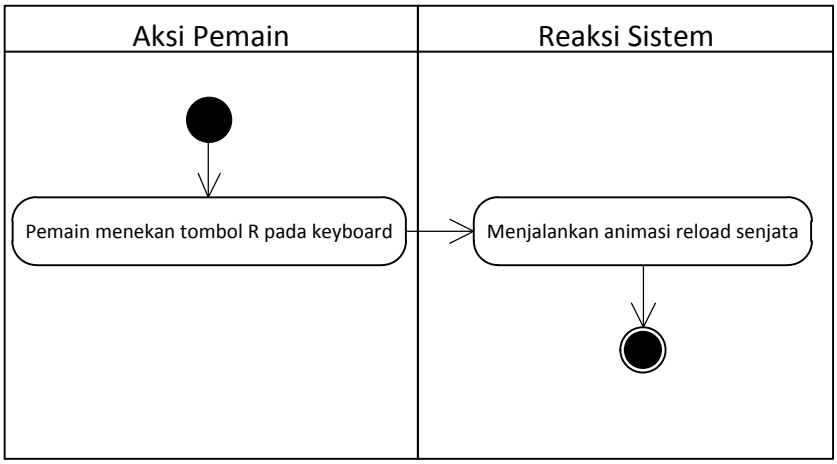
H. *Activity Diagram* Proses Mengambil Peluru



Tabel 3.30 Penjelasam Proses Mengambil Peluru

Deskripsi	Proses mengambil peluru untuk karakter
Normal Flow	1. Pengguna menggerakan karakter dan berjalan menuju item pelutu. 2. Sistem akan menambahkan jumlah peluru pada *player*.

I. *Activity Diagram* Proses *Reload* Senjata

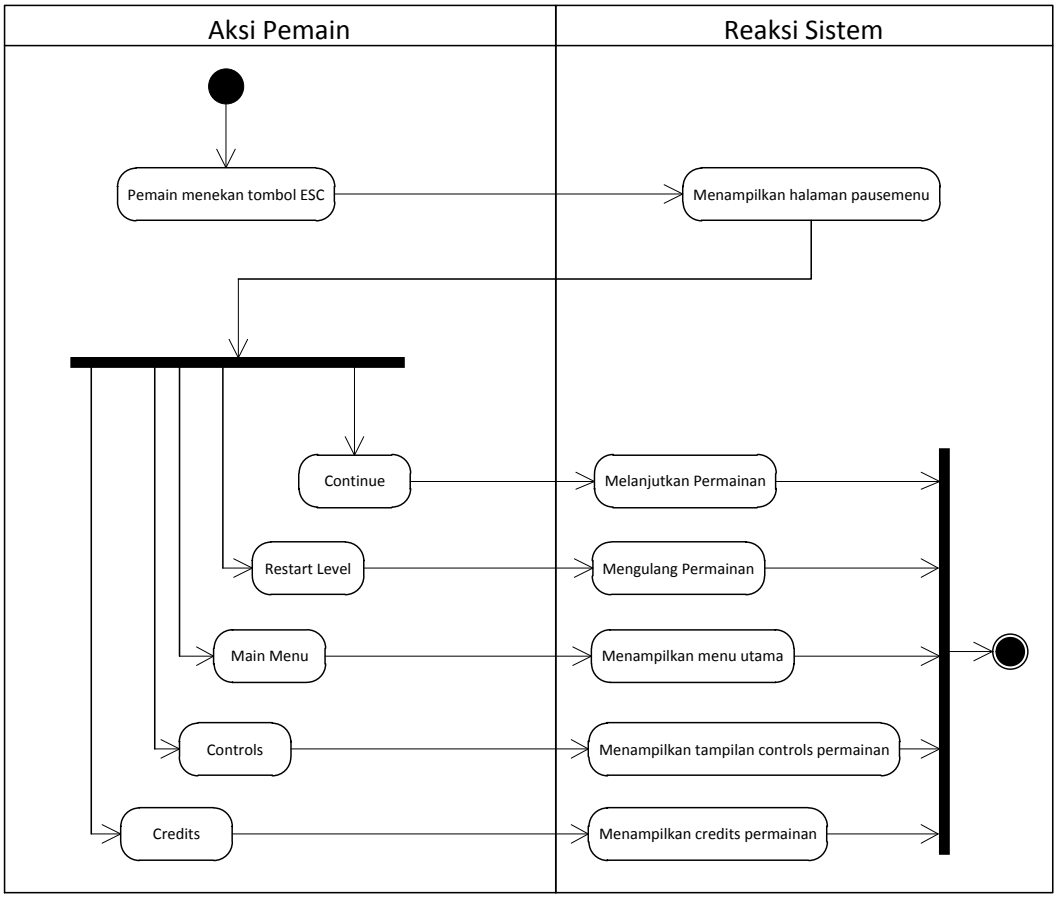


Gambar 3.27 *Activity Diagram* Mengisi peluru

Tabel 3.31 Penjelasan Proses Memilih *Reload* Senjata

Deskripsi	Proses reload senjata, sehingga peluru pada senjata player menjadi penuh
Normal Flow	1. Pengguna menggerakkan karakter menjalankan animasi *reload* senjata dengan interaksi menggunakan *keyboard* menekan tombol R 2. Sistem menampilkan karakter menjalankan animasi *reload* senjata

J. *Activity Diagram* Proses Memberhentikan Permainan



Gambar 3.28 *Activity Diagram* proses memberhentikan permainan

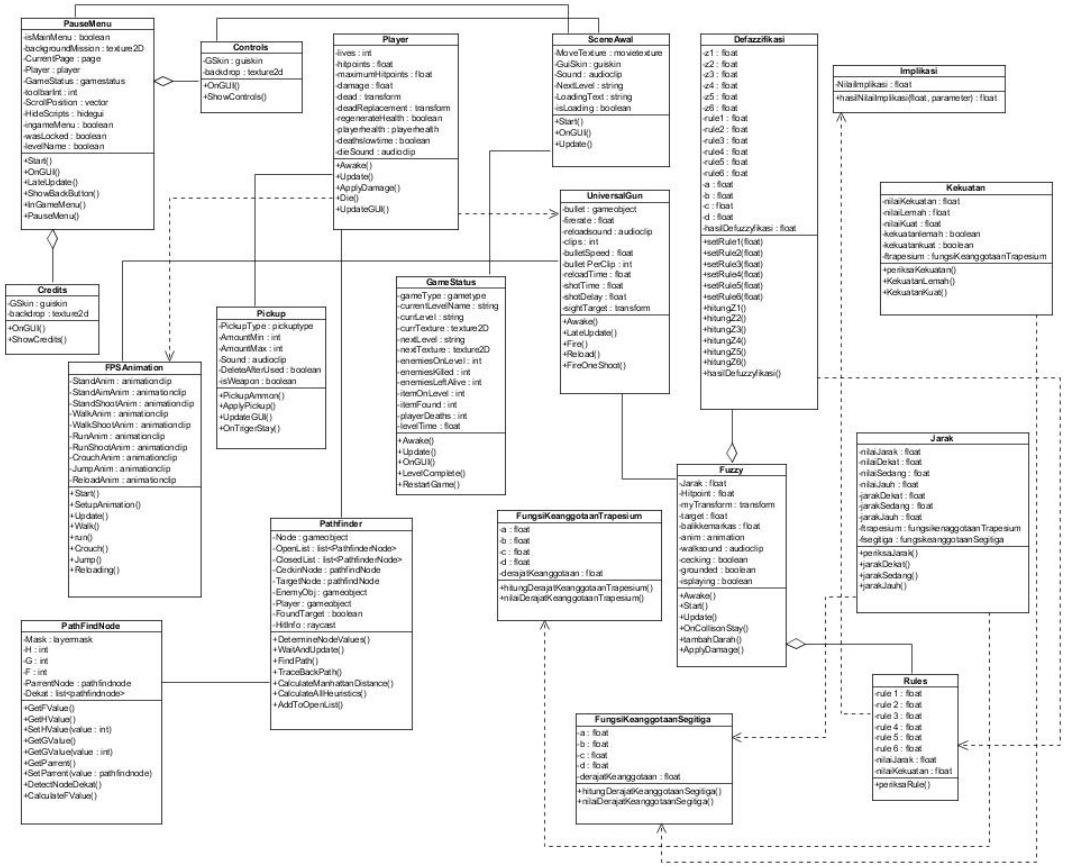
Tabel 3.32 Proses Memberhentikan Permainan

Deskripsi	Proses memberhentikan permainan
Normal Flow	1. Pemain menekan tombol ESC pada *keyboard*. 2. Sistem akan menampilkan *pausemenu*. 3. Pemain memilik tombol menu *continue*. 4. Sistem akan mengembalikan ke tampilan arena permainan. 5. Pemain memilih tombol menu *restartlevel*. 6. Sistem akan mengulang permainan ke tampilan awal. 7. Pemain memilih tombol menu *main* menu. 8. Sistem akan menampilkan tampilan menu utama permainan

	9. Pemain memilih tombol menu controls. 10. Sistem menampilkan halaman controls permainan. 11. Pemain memilih tombol menu credits. 12. Sistem menampilkan halaman credits permainan

3.4.3 *Class Diagram*

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

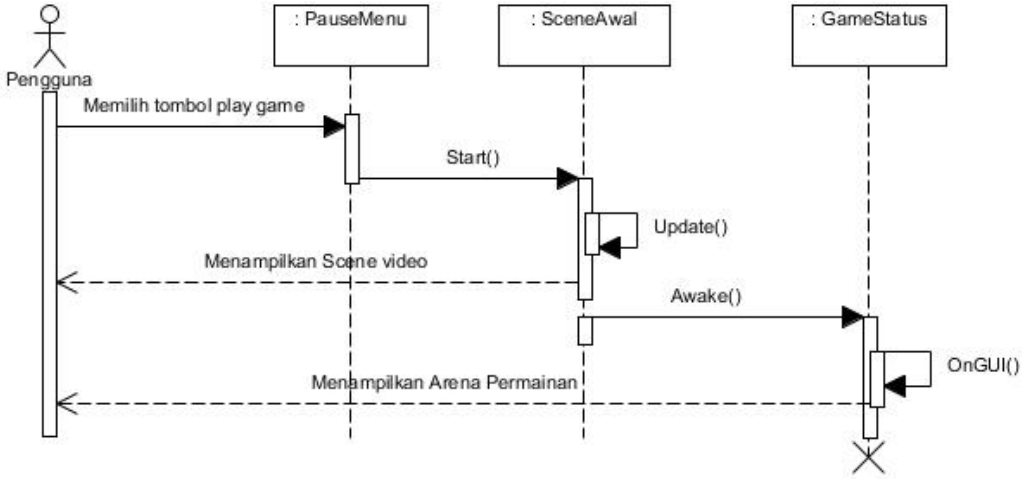


Gambar 3.27 *Class Diagram The Defend Island*

3.4.4 *Sequence Diagram*

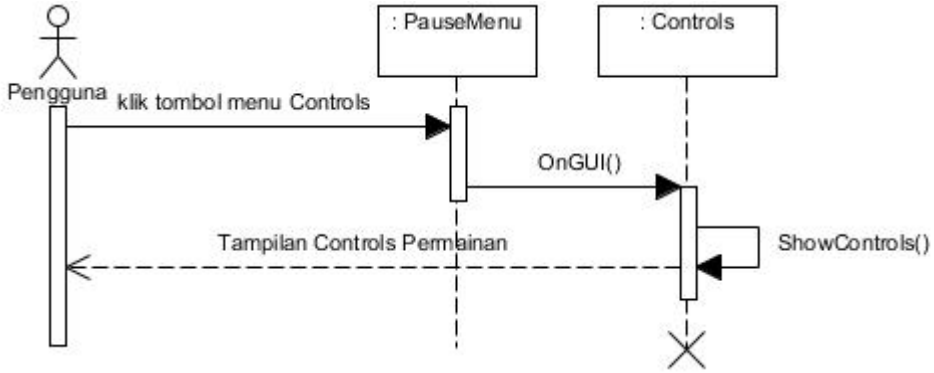
Sequencial diagram menggambarkan interaksi antar objek di dalam dan disekitar sistem berupa *message* yang digambarkan terhadap waktu. Berikut ini beberapa *sequencial diagram* yang terdapat pada *game The Defend Island* yaitu :

1. *Sequence Diagram* Mulai Permainan



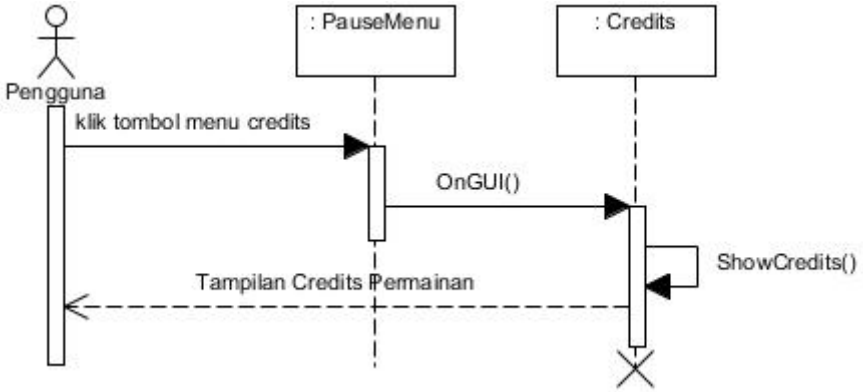
Gambar 3.28 *Sequence Diagram* Prose Memulai Permainan

2. *Sequence Diagram* Melihat Tampilan Petunjuk



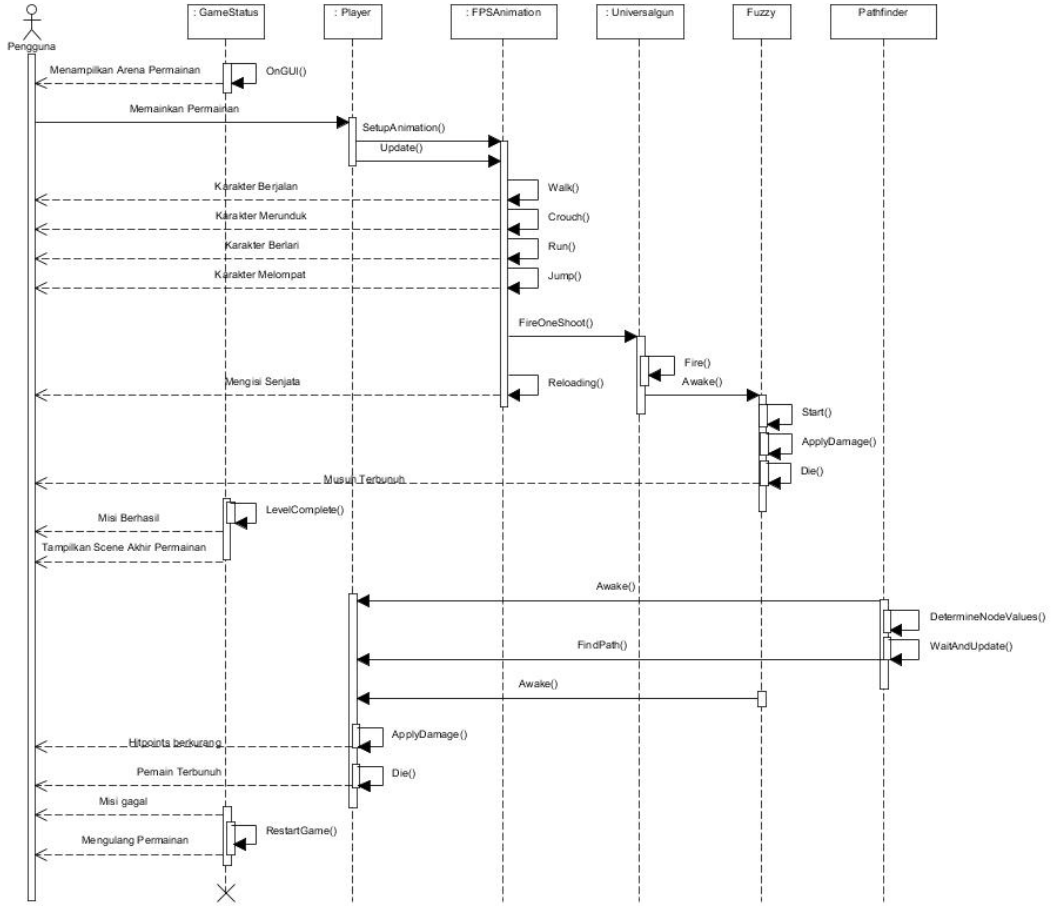
Gambar 3.29 *Sequence Diagram* Melihat Tampilan Petunjuk

3. *Sequence Diagram* Melihat Tampilan *Credits*



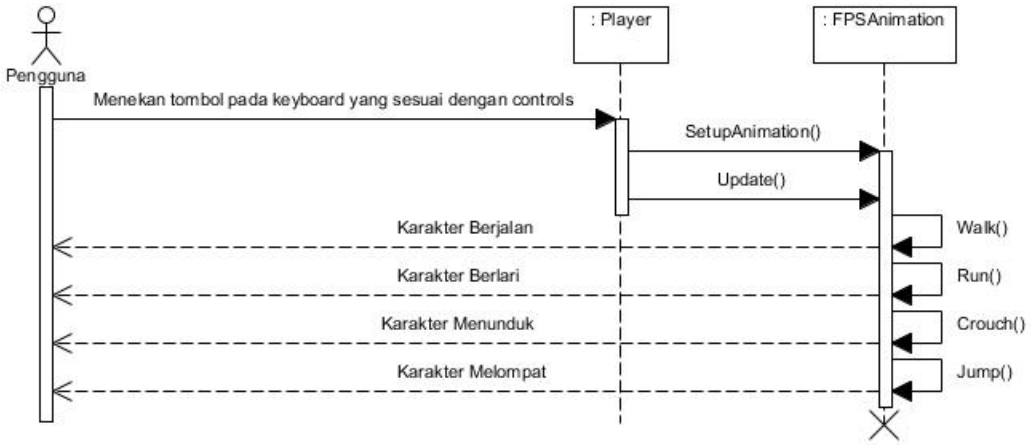
Gambar 3.30 *Sequence Diagram* Melihat Tampilan *Credits*

4. *Sequence Diagram* Memainkan Permainan



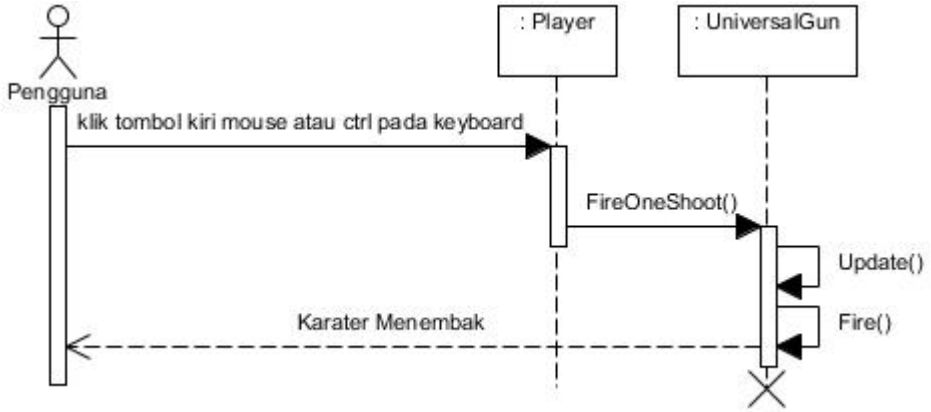
Gambar 3.31 *Sequence Diagram* Memainkan Permaian

5. *Sequence Diagram* Pergerakan *Player*



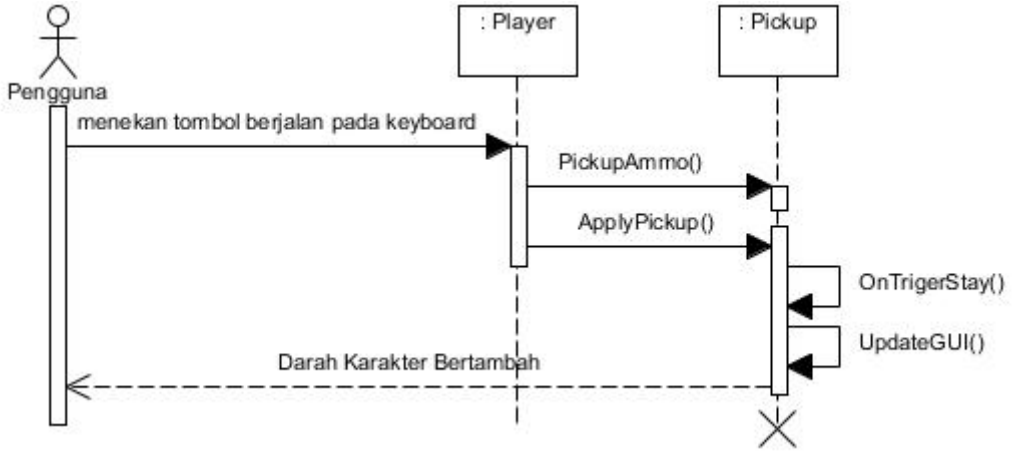
Gambar 3.32 *Sequence Diagram* Proses Pergerakan *Player*

6. *Sequence Diagram* Menembak



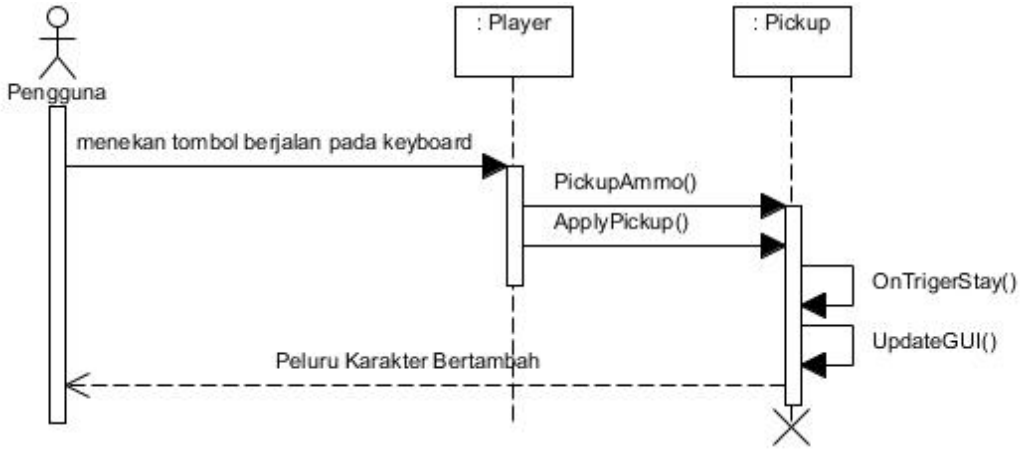
Gambar 3.33 *Sequence Diagram* Proses Menembak

7. *Sequence Diagram* Mengambil Darah



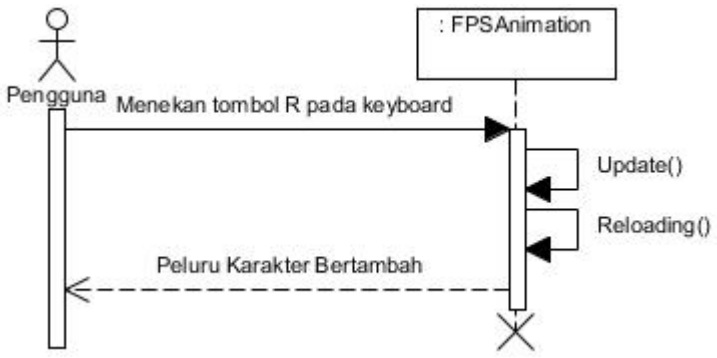
Gambar 3.33 *Sequence Diagram* Proses Mengambil Darah

8. *Sequence Diagram* Mengambil Peluru



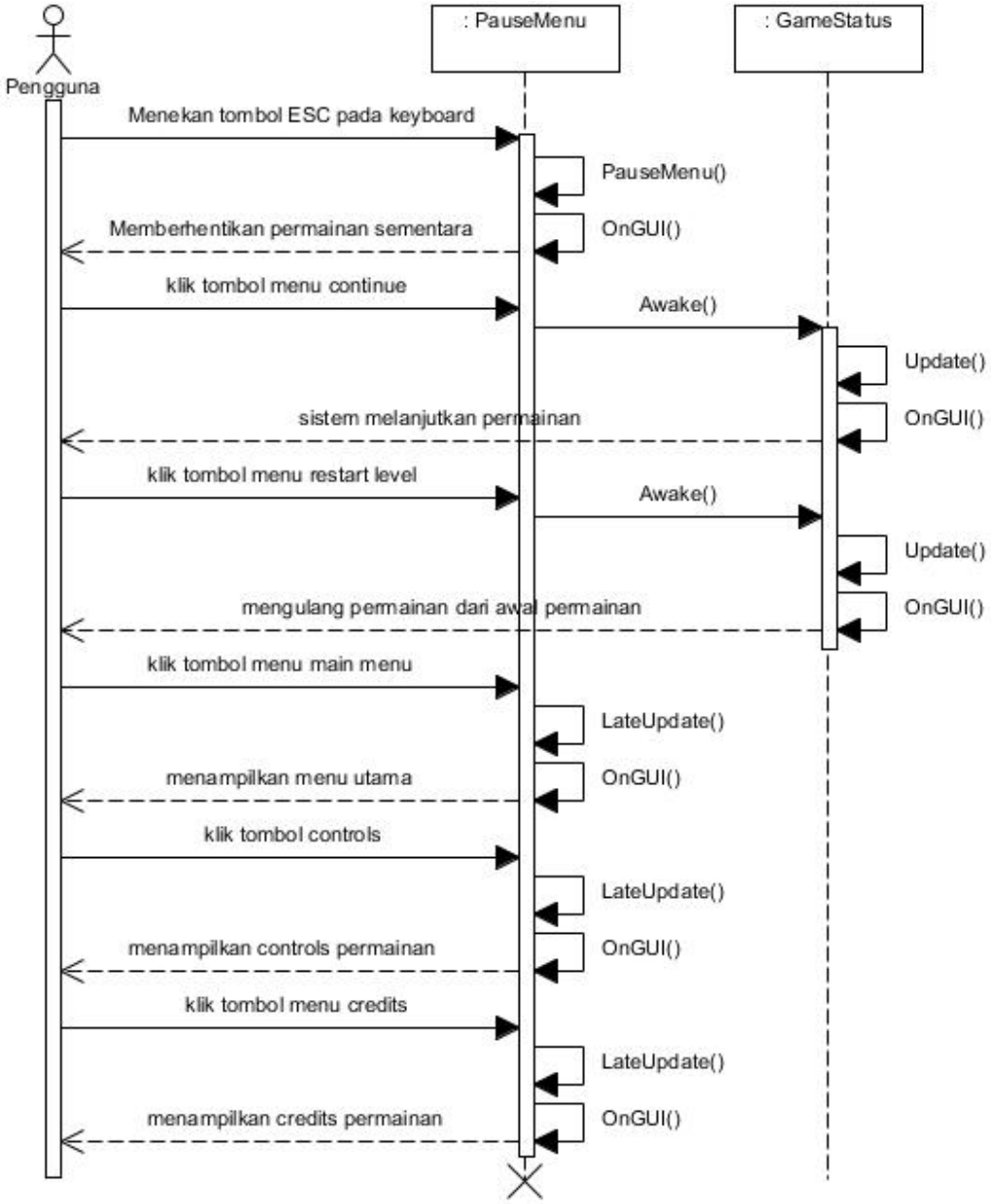
Gambar 3.34 *Sequence Diagram* Proses Mengambil Peluru

9. *Sequence Diagram* Reload Senjata



Gambar 3.35 *Sequence Diagram Reload* Senjata

10. *Sequence Diagram* Memberhentikan Permainan



Gambar 3.36 *Sequence Diagram* Memberhentikan Permainan

3.5 Perancangan Sistem

Perancangan adalah suatu bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan setelah tahapan untuk memberikan gambaran secara terperinci. Dalam tahap ini digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean ke dalam suatu bahasa pemrograman.

3.5.1 Karakter Utama

Karakter utama dalam game ini bernama Marko. Karakter ini memiliki seragam pasukan khusus yang dirancang untuk melawan alien yang berwarna coklat tua. Marko juga memakai sepatu bot khusus dan juga dilengkapi dengan *helm* pengaman. Dengan Semua peralatan yang ada Marko siap untuk menghadapi seluruh serangan alien.

Game The Defend Island adalah *game* yang *bergenre* FPS *(First Person Shooter)* yang mempunyai sudut pandang *camera* dalam *game* yang tidak memperlihatkan pergerakan karakter utama secara keseluruhan. Oleh sebab itu tubuh, kepala, kaki dan tangan dari karakter utama tidak diperlihatkan tetapi hanya memperlihatkan bagian moncong senjatanya saja akan. Pada saat di dalam game *player* hanya akan melihat senjata yang dipakai dan *crossfire* yang berada di tengah layar.

Tabel 3.33 Propertis Karakter Utama

Karakter Utama	
Nick Name	Marko
***Hit Points* (HP)**	250
Lives	2
Abilities	Berjalan, Menunduk, Melompat, Menembak
Guns	*Machinegun*

3.5.2 Senjata / *Weapon*

Hanya terdapat satu senjata jenis *machine gun* yang digunakan oleh Marko, akan tetapi senjata *machine gun* yang digunakan oleh Marko memiliki dua jenis peluru (*bullet*). Machine gun dengan peluru biasa memiliki *damage* 20 yang berarti ketika ditembakan ke emeny akan mengurangi hitpoint *enemy* sebanyak 20, sedangkan *machine gun* dengan peluru *bom* memiliki damage 50. *Machine gun* dengan peluru biasa memiliki 5 pack yang berisi 30 butir peluru. Sedangkan untuk peluru *bom* memiliki 3 pack yang didalamnya berisi 8 butir peluru.



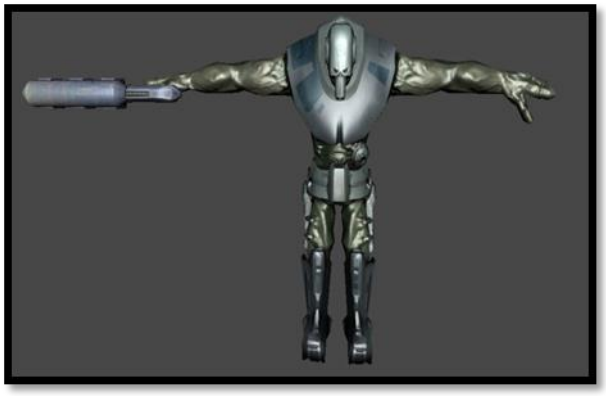
Gambar 3.34.Senjata *Machine Gun*

Tabel 3.34 Propertis Senjata

Senjata *Machine Gun*		
Jenis Peluru	Peluru Biasa	Peluru Boom
Damage	20	50
Bullet Pack	30 x 5	8 x 5
Reload Time	0.5 second	1 second

3.5.3 Musuh atau *Enemy*

Di dalam *game* ini terdapat tiga jenis musuh berupa *robot alien*, pesawat *alien* dan *sentrygun*. *Robot alien* yang memiliki tinggi tidak beda jauh dengan karakter utama, *robot alien* hanya memiliki satu senjata dengan peluru tidak terbatas dan damage senjata yang cukup besar. AI yang diterapkan pada NPC (*non playable character*) diharapkan dapat merespon pergerakan karakter terhadap *enemy* sehingga bisa menghasilkan keputusan sesuai dengan keadaan dan kondisi tertentu yang sudah dipertimbangkan. *Robot alien* terdapat pada *level* dua dan *level* tiga, sedangkan untuk pesawat *alien* dengan *sentygun* terdapat pada *level* tiga, pada *level* ketiga pesawat akan mengejar *player* secara otomatis ketika *player* sudah masuk dalam jangkauannya dan akan menembakinya dari atas dengan senjata laser dan peluru yang tidak terbatas.



Gambar 3.35 *Enemy Robot Alien*



Gambar 3.36 Pesawat *Alien*



Gambar 3.37 *Sentrygun*

Berikut ini adalah penjelasan perbedaan dari ketiga *enemy* :

Tabel 3.35 Propertis Perbandingan *Enemy*

Properties	***Robot Alien***	***Pesawat***	***Sentry gun***
Speed	3	5	-
Range Shoot	15	30	25
Delay Shoot	0.5	0.4	0.1
Damage	20	30	25
HP	250	300	150

3.5.4 Karakter Teman

Karakter teman yang terdapat dalam *game* ini hanya satu orang teman, karakter teman ini yang nantinya akan membantu karakter utama yang bernama marko untuk mengalahkan musuh (*alien*) dan menghancurkan markas musuh yang ada ditengah hutan.



Gambar 3.38 Karakter Teman

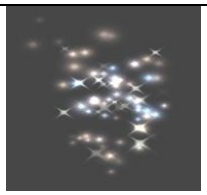
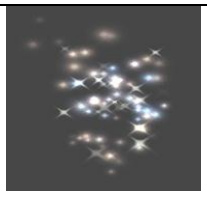
Tabel 3.36 *Properties* Karakter *Ray*

Karakter Teman (Shane)	
Nick Name	Ray
***Hit Points* (HP)**	200
Lives	1
Abilities	Menembak
Guns	*Machine gun*

3.5.5 *Items*

Berikut ini adalah beberapa *items* yang terdapat di dalam game dan yang berhubungan dengan jalan cerita game itu sendiri beserta penjelasannya.

Tabel 3.37 Properties *Items*

Gambar *Items*	**Nama *Items***	**Keterangan**
	*Sparkle*	*Target Point*, menunjukkan tempat yang harus dicapai untuk menyelesaikan misi.
	*Health*	*Item* yang bisa diambil oleh karakter utama untuk menambah darah atau *hitpoints*.
	*Bullets*	*Item* yang bisa di ambil karakter utama untuk menambah jumlah peluru pada senjata karakter
	*Player Respawn*	Tempat karakter utama hidup kembali, apabila *hitpoints* = 0 dan *lives* lebih dari 0.

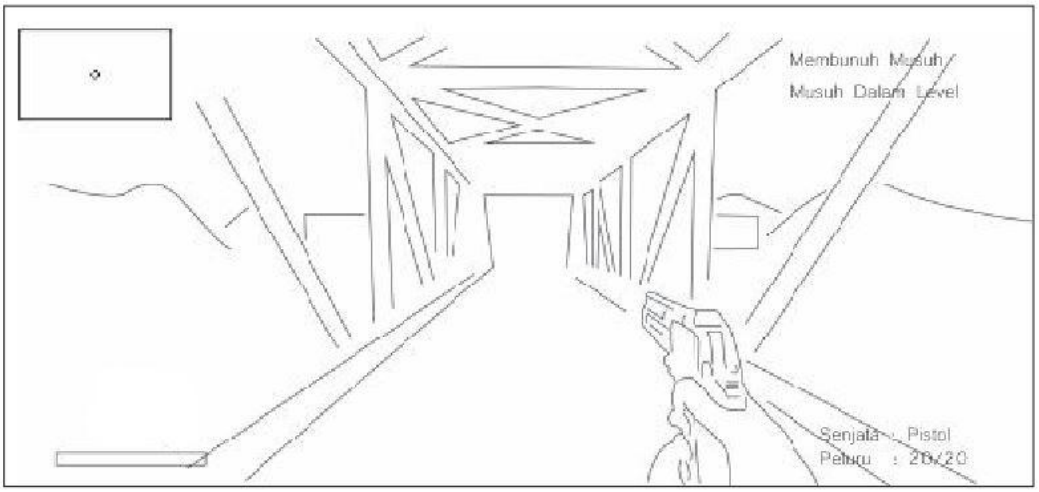
3.5.6 *Storyboard*

Storyboard menceritakan alur cerita dari sistem dengan gambar atau sketsa sehingga orang lain dapat dengan mudah untuk memahai apa yang ada pada sistem hanya dengan melihat gambar.



Gambar 3.39 *Storyboard Level1*

Pada *level* pertama ini *player* harus mencari petunjuk dengan berinteraksi kepada warga desa untuk mendapatkan informasi tentang keberadaan temannya yang disandra oleh mahluk asing.



Gambar 3.40 *Storyboard Level 2*

Pada *level* dua ini *player* sudah masuk ke markas *alien* dan harus mencari kunci untuk membebaskan temannya yang dikurung, kunci akan didapat ketika *player* berhasil mengalahkan semua *alien* yang berjaga didaerah itu.



Gambar 3.41 *Storyboard Level 3*

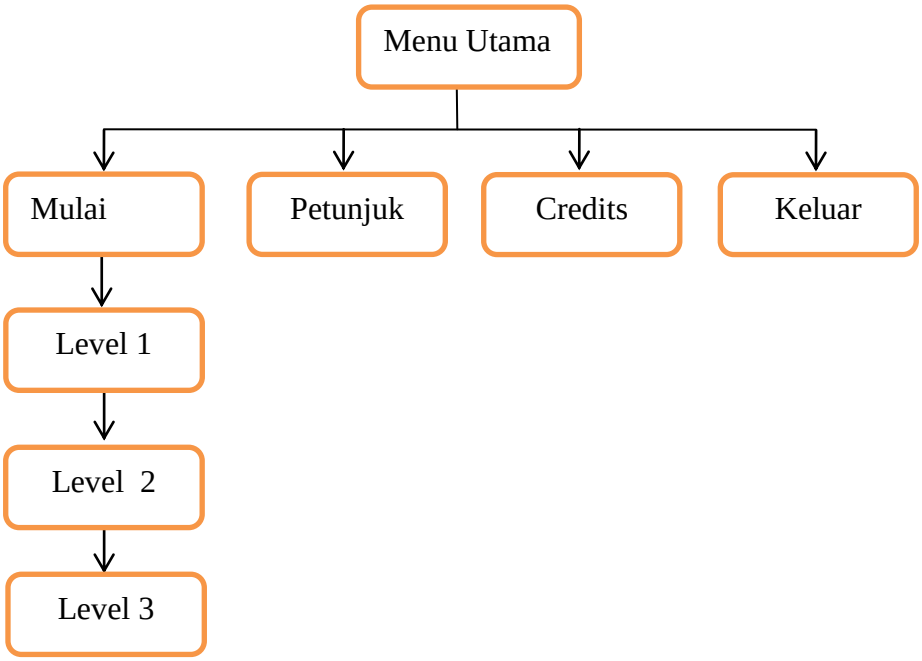
Pada *level* ini *player* harus mengalahkan semua musuh menghancurkan markas *alien* dengan kerjasama bersama teman yang dibebaskan pada *level* dua.

3.6 Perancangan Arsitektur

Setelah melakukan perancangan data pada sistem yang dibangun, maka dilakukanlah perancangan arsitektur. Perancangan arsitektur yang telah dibuat meliputi beberapa perancangan diantaranya struktur menu, perancangan antarmuka,

3.6.1 Perancangan Struktur Menu

Perancangan struktur menu dirancang untuk menyediakan fungsi-fungsi yang akan digunakan dalam *game The Defend Island* .

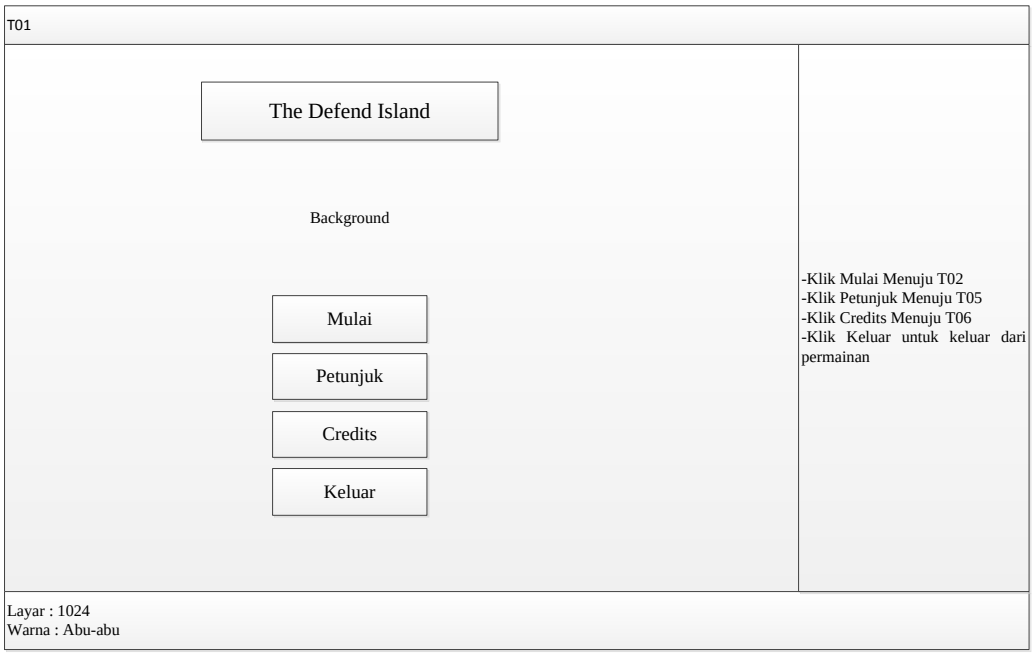


Gambar 3.42 Struktur Menu Utama

3.6.2 Perancangan Antarmuka

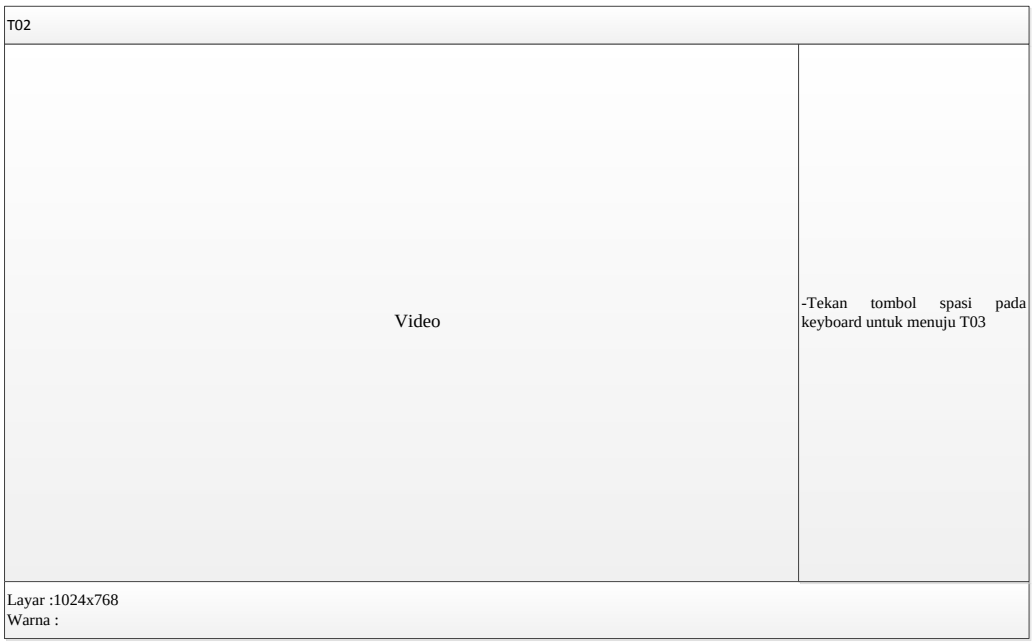
Perancangan antarmuka dibutuhkan untuk mawakili keadaan sebenarnya dari aplikasi yang akan dibangun. Berikut ini beberapa perancangan antarmuka dari aplikasi yang akan dibangun yaitu :

1. Antarmuka Menu Utama



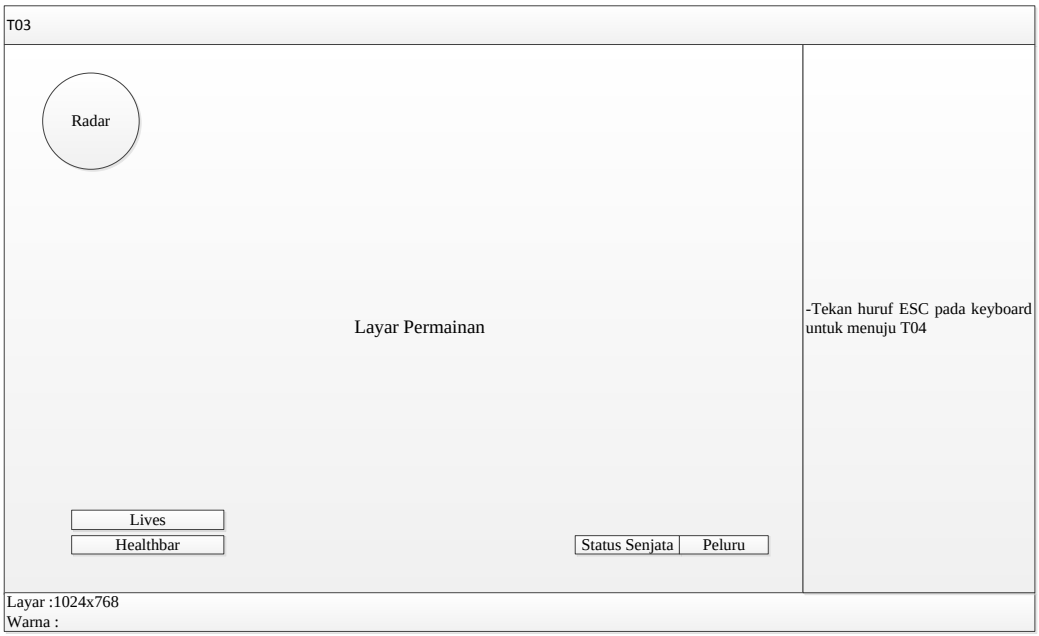
Gambar 3.43 Antarmuka Menu Utama

2. Antarmuka *Scene Video*



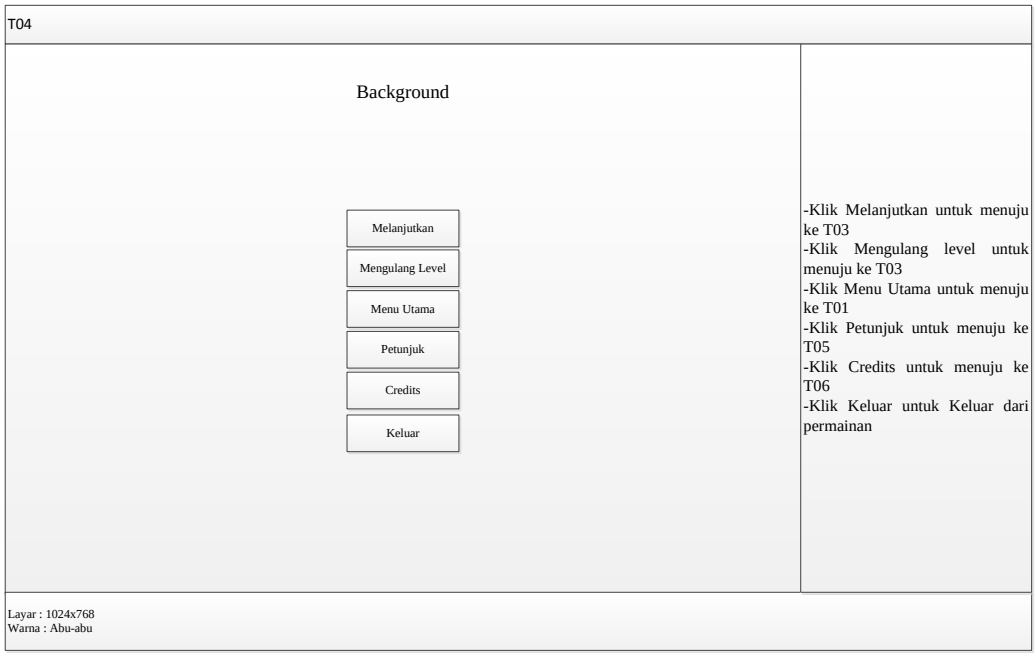
Gambar 3.44 Antarmuka *Scene Video*

3. Antarmuka Permainan



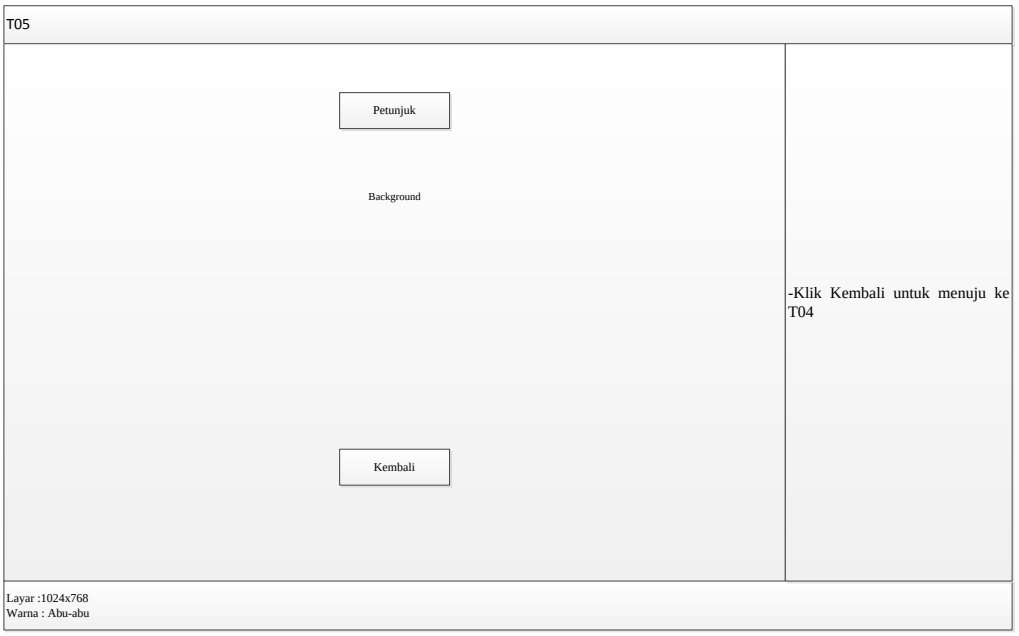
Gambar 3.45 Antarmuka Permainan

4. Antarmuka *Pause* Menu



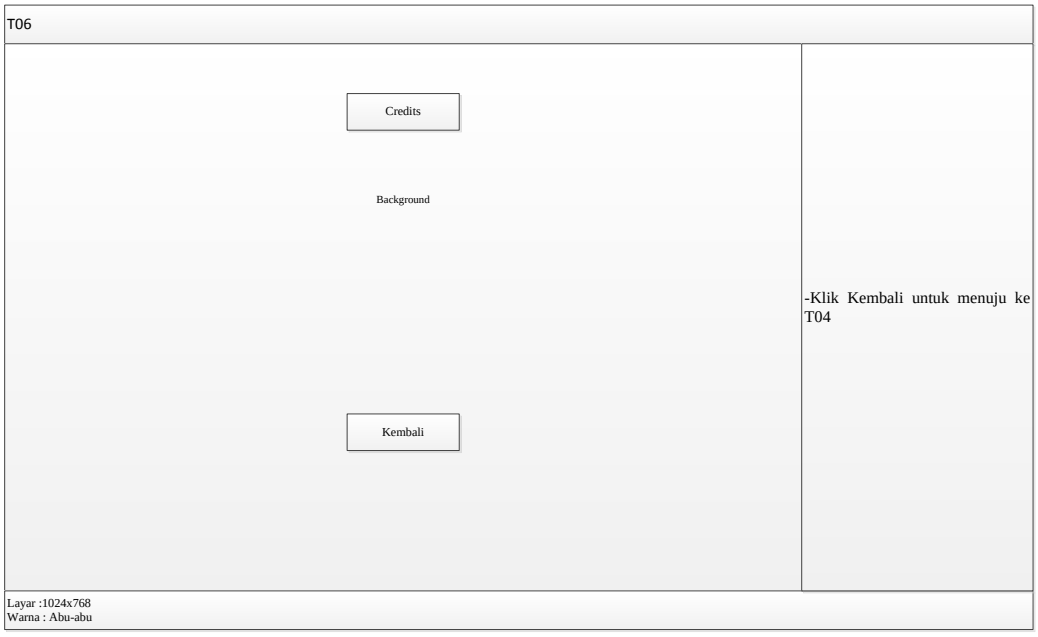
Gambar 3.46 Antarmuka *Pause Menu*

5. Antarmuka Petunjuk



Gambar 3.47 Antarmuka *Controls*

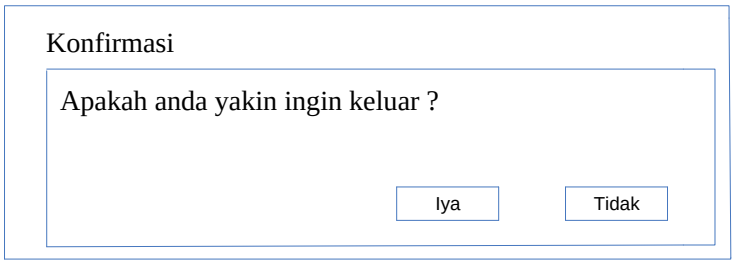
6. Antarmuka *Credits*



Gambar 3.48 Antarmuka *Credits*

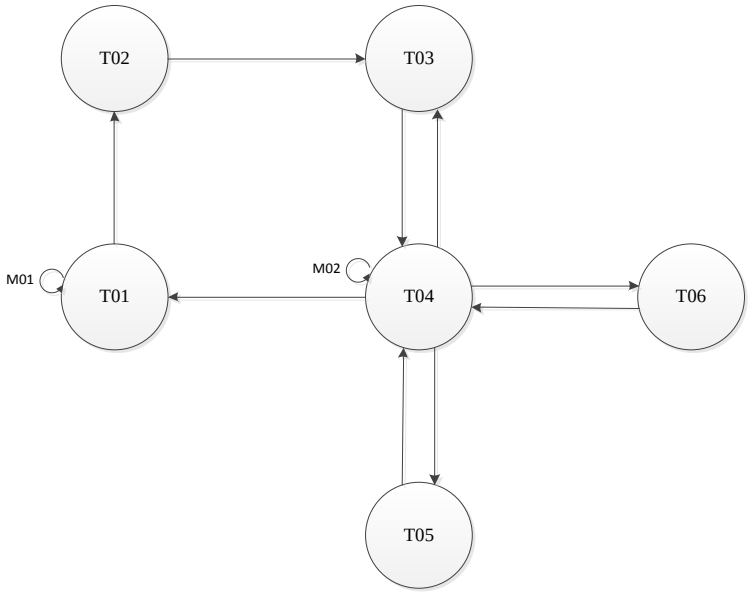
3.6.3 Perancangan Pesan

Perancangan pesan muncul sebagai informasi kepada pengguna dalam mengoperasikan sistem.



Gambar 3.49 Perancangan Pesan M01 dan M02

3.6.4 Jaringan Semantik



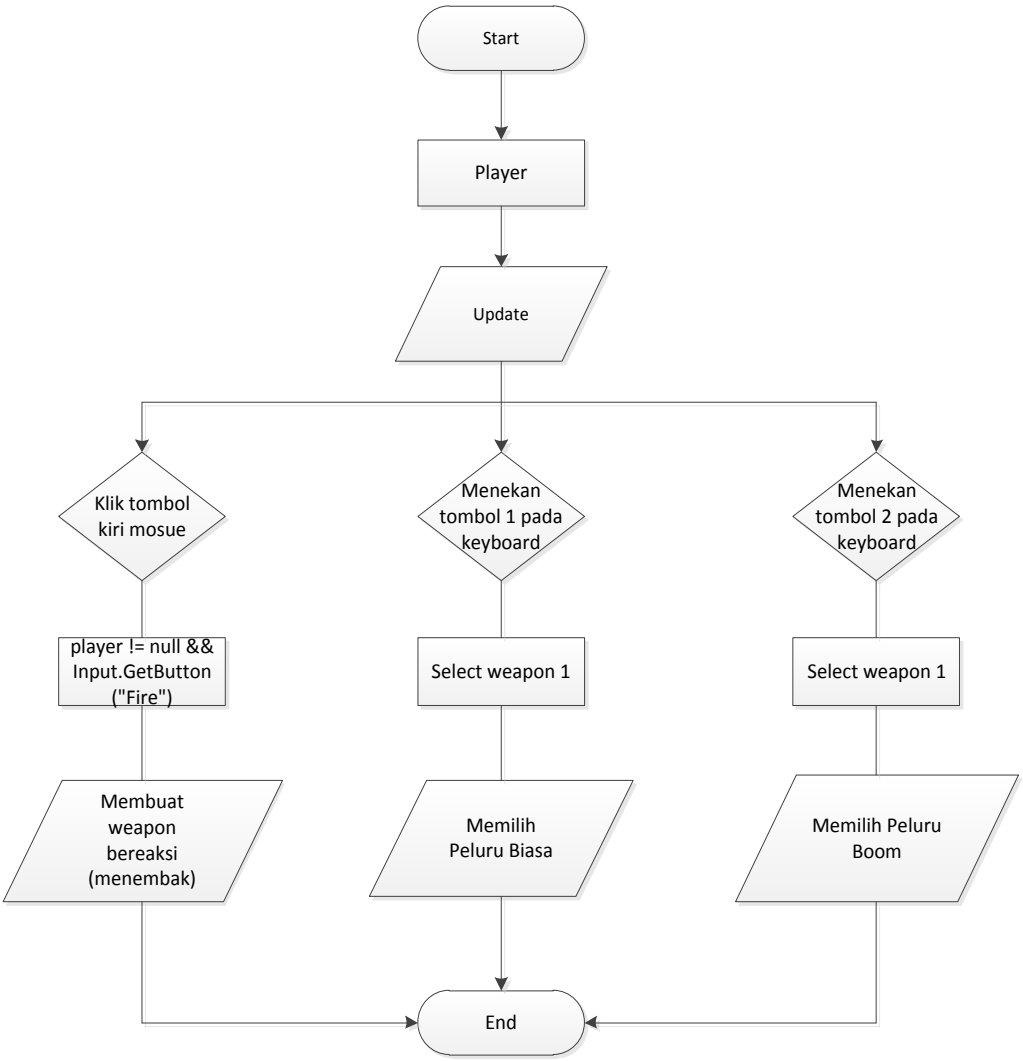
Gambar 3.50 Jaringan Semantik

3.7 Perancangan *Method*

Perancangan method merupakan perancangan yang berfungsi untuk mendeskripsikan method-method yan berada di dalam aplikasi. Methode dapat dipanggil dengan menyetakan variabel, baik hanya satu variabel, banyak variabel atau bahkan tidak ada sama sekali. Adapun method-method yang terdapat dalam game action The Defend Islan yang akan dibangun dijelaskan melalui flowchart sebagai berikut :

3.7.1 *Method Player Weapon*

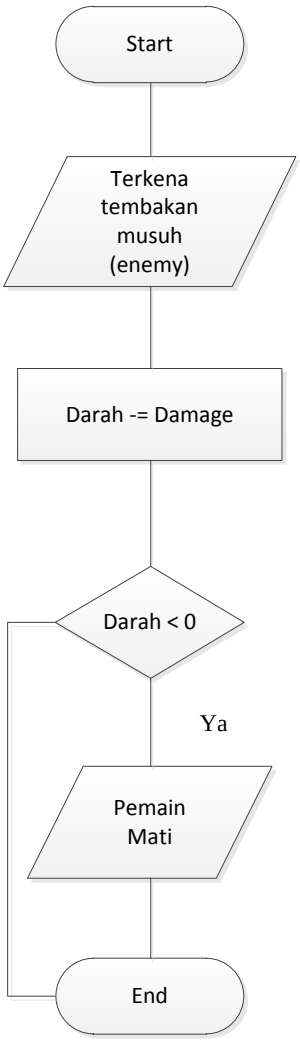
Methode player weapon digunakan untuk menampung samua atribut-atribut senjata yang ada pada *player*. Untuk lebih jelasnya dapat dilihat pada gambar *flowchart* dibawah ini.



Gambar 3.51 *Methode Player Weapon*

3.7.2 *Method Character Damage*

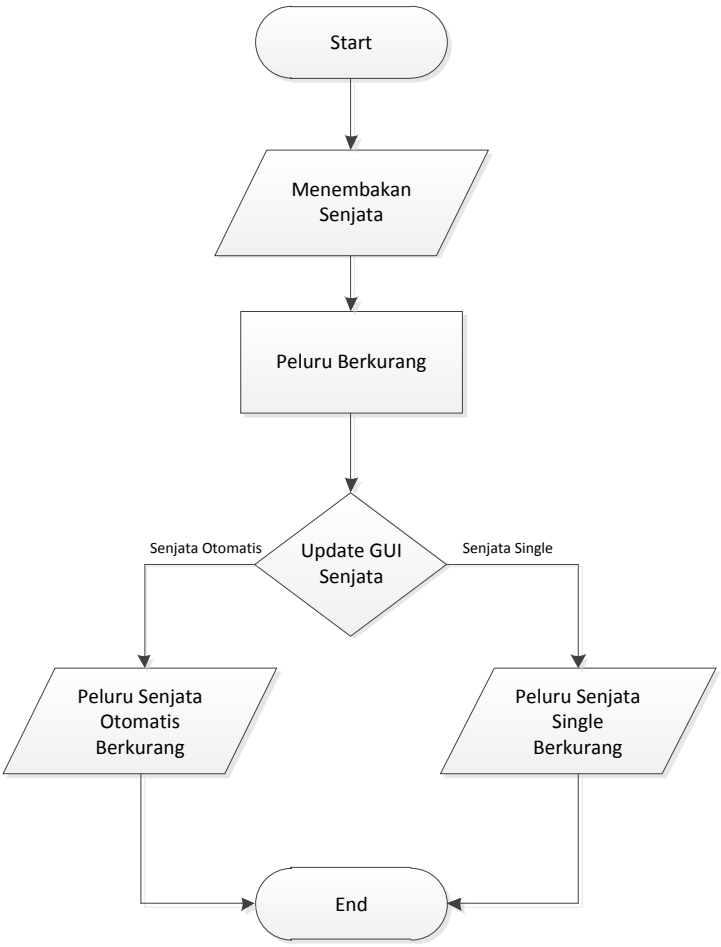
Methode character damage digunakan untuk membuat character menerima damage yang akan mengurangi *hitpoints* (darah). Untuk lebih jelasnya dapat dilihat pada gambar flowchart dibawah ini.



Gambar 3.52 *Method Character Damage*

3.7.3 *Method UpdateGUI*

Metode ini akan melakukan perubahan terhadap GUI berupa pengurangan peluru ketika player menambak, baik baik menembak menggunakan senjata *machinegun* otomatis maupun senjata *machinegun single*.



Gambar 3.53 *Method UpdateGUI*