BAB III
ANALISIS DAN PERANCANGAN SISTEM

III.1 Analisis Sistem

Pada bagian analisis sistem ini, akan dilakukan berbagai macam analisis yang berhubungan dengan *game* yang akan dibangun, mulai dari analisis terhadap *game* sejenis serta analisis berbagai kebutuhan baik fungsional maupun non-fungsional.

III.1.1 Analisis Masalah

Saat ini pendidikan sejarah mengalami penurunan, salah satu faktornya karena kurang menariknya media pembelajaran yang menerangkan tentang sejarah di Indonesia, termasuk di dalamnya adalah pemberian materi yang terkesan monoton karena hanya mengedepankan sisi menghafal isi materi dari pada memaknai apa kandungan materi tersebut [1].

Sejarah Pertempuran 10 November 1945 merupakan salah satu sejarah pertempuran besar yang terjadi di Indonesia. Pertempuran ini penuh dengan nilai-nilai *historis* atau sejarah yang menggambarkan keberanian dan kebulatan tekad para pahlawan untuk membela tanah air dan kemerdekaan [2].

Berdasarkan uraian di atas maka dapat disimpulkan terdapat beberapa masalah yaitu :

1. Bagaimana membangun *game* yang mampu menjadi media pembelajaran yang menarik dalam upaya memberikan informasi tentang pendidikan sejarah khususnya sejarah Pertempuran 10 November 1945?
2. Bagaimanan membangun *game* sebagai media pembelajaran yang tidak hanya memberikan materi sejarah yang penuh dengan hafalan?

III.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 akan menjadi acuan disini adalah *game* P1ONER, pertempuran 10 November dan *Game* Merdeka atau Mati.

1. P10NER 10 November 1945

a. Pengenalan

Game P10NER, merupakan *game* edukasi yang mempunyai *genre first person shooter* (FPS), yang dibuat oleh beberapa mahasiswa ITS Surabaya dalam rangka menyemarakkan pagelaran mahasiswa teknologi informasi dan komunikasi (GeMasTIK) 2010 di Surabaya. *Game* ini dibuat oleh 10 orang mahasiswa ITS Surabaya yang mempunyai tugas masing-masing yaitu 3 orang sebagai programer, 3 orang sebagai desin *game* 3D dan 4 orang sebagai desain konten.

Game P10NER, menceritakan pertempuran yang terjadi di Surabaya pada tanggal 10 November 1945. *Game* ini mempunyai 3 *level*, *level* pertama menceritakan kejadian pada tanggal 19 September 1945 di Hotel Yamato, *level* kedua menceritakan pertempuran dengan salah satu jendral tentara Inggris dan level terakhir merupakan pertempuran besar-besaran terhadap tentara Inggris pada tanggal 10 November 1945.

Pada gambar III.1 merupakan tampilan awal pada *game* 10 November 1945, terdapat 3 buah menu berupa mulai, *load* dan keluar.



Gambar III.1 *Screenshoot* tampilan awal permainan *game* P10NER

Pada gambar III.2 bisa dilihat salah satu tampilkan pertempuran yang terjadi di dalam Hotel Yamato.



Gambar III.2 *Screenshoot* tampilan permainan *game* P10NER

Pada gambar III.3 merupakan tampilkan pertempuran dengan tentara Inggris yang terjadi di atas Hotel Yamato.



Gambar III.3 *Screenshoot* tampilan permainan *game* P10NER

b. Gameplay

Pada *game* P10NER 10 November 1945, pemain harus mencari dan mengambil beberapa kunci yang disediakan oleh sistem disetiap *level* yang dimainkan, untuk digunakan pemain ketika akan membuka pintu yang membutuhkan kunci untuk membukanya.

c. Skenario *Game*

Skenario dalam permainan *game* P10NER disusun oleh beberapa hal sebagai berikut :

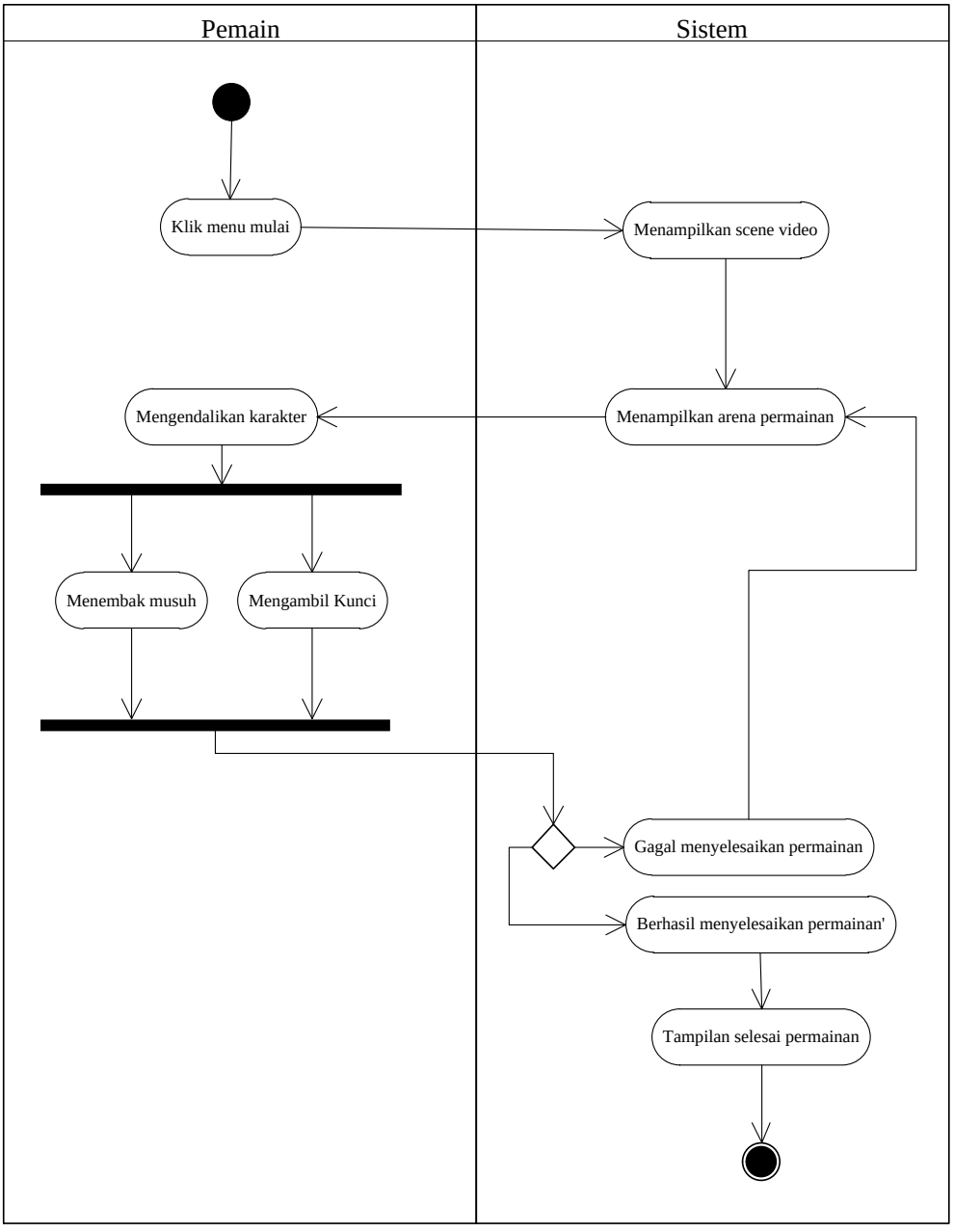
1. Alur Permaian

   Alur permainan pada *game* P10NER adalah sebagai berikut :

   a. Pemain berada di dalam hotel yamato yang dipenuhi oleh musuh.
   b. Pemain harus mencari kunci apabila hendak membuka sebuah pintu atau jalan utama ketika berada di dalam permainan.
   c. Pemain harus menghindari dan mematikan musuh ketika berada di arena permainan.

d. Jika pemain terkena tembakan oleh musuh, maka darah / *health point* akan berkurang dan apabila habis maka permainan diulangi ke awal permainnan.

Adapun alur sistem dari permainan *game* P10NER dapat di gambarkan melalui *activity diagram* berikut ini :



Gambar III.4 Alur permainan pada *game* P10NER

Game P10NER memiliki beberapa keterbatasan yaitu,

- *Game* ini merupakan *game* edukasi akan tetapi konten edukasi yang diperlihatkan pada *game* ini tidak terlalu terlihat.
- *Setting* tempat / *mapping* yang ada pada *game* ini kurang sesuai dengan fakta sejarah yang terjadi.
- Materi tentang sejarah Pertempuran 10 November 1945 hanya diberikan melalui *scene video*.

Disamping ada beberapa keterbatasan, *game* P10NER sendiri memiliki banyak kelebihan seperti :

- Terdapat fitur *save* dan *load game*
- Senjata yang dapat digunakan *player* beragam.

2. *Game* Merdeka atau Mati

a. Pengenalan

Merdeka atau Mati merupakan *game* yang mengangkat cerita sejarah pertempuran Indonesia, *game* yang menceritakan tentang perjuangan seorang pejuang yang berperan melawan penjajah Belanda dan Jepang. *Game* ini dibuat dengan grafis 2D, berbasis *mobile (Symbian)*.



Gambar III.5 *Game* Merdeka atau Mati

Pada gambar III.6 Merupakan salah satu cuplikan permainan dari Merdeka atau Mati.



Gambar III.6 *Screenshoot* tampilan permainan *game* Merdeka atau Mati

b. *Gameplay*

Pada *game* Merdeka atau Mati, pertama-tama pemain akan diberikan petunjuk permainan secara otomatis berupa tombol-tombol apa saja yang harus digunakan pemain ketika pemain memulai permainan. Setelah petunjuk permainan selesai, pemain akan langsung masuk ke dalam arena permainan. Pemain diberikan 2 senjata berupa 1 buah pistol dan 1 buah basoka yang digunakan untuk menembak musuh. Musuh yang terdapat di dalam *game* ini secara langsung akan menembak ketika pemain mendekati musuh. *Level* permainan berakhir ketika pemain telah mengalahkan semua musuh.

c. Skenario *Game*

Skenario dalam permainan *game* Merdeka atau Mati disusun oleh beberapa hal sebagai berikut :

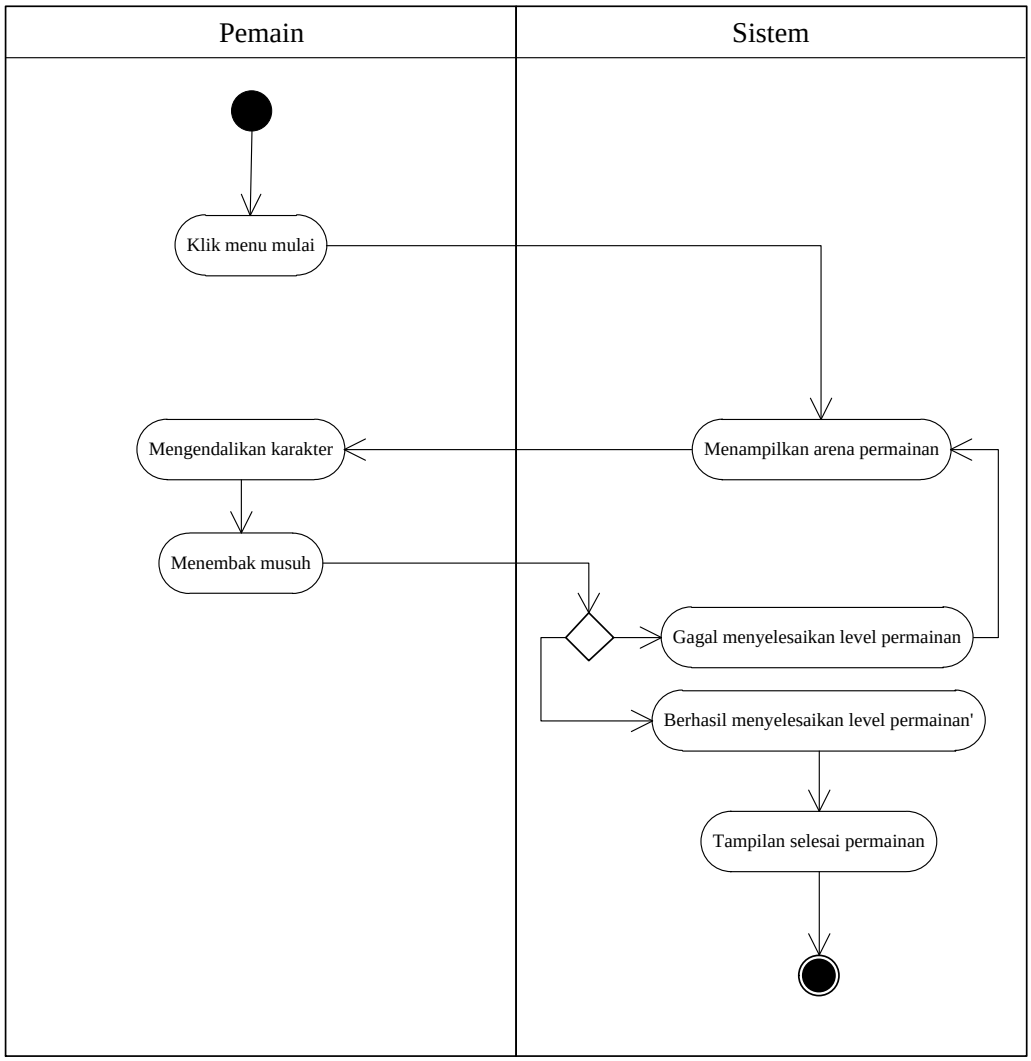
a. Alur Permaian

Alur permainan pada *game* Merdeka atau Mati adalah sebagai berikut :

a. Pemain berada di dalam arena permainan
b. Pemain diberikan senjata yang digunakan untuk menembak musuh
c. Pemain harus menembak mati seluruh musuh apabila ingin mengakhiri *level* yang sedang dimainkan.

d. Jika pemain tertembak musuh maka darah akan berkurang.

Adapun alur sistem dari permainan *game* Merdeka atau Mati dapat di gambarkan melalui *activity diagram* berikut ini :



Gambar III.7 Alur permainan pada *game* Merdeka atau Mati

Game Merdeka atau Mati memiliki beberapa kekurangan yaitu :

- Alur cerita yang tidak sesuai dengan fakta sejarah peperangan di Indonesia.
- Grafis yang kurang menarik karena masih dibuat dengan grafis 2D
- Setiap *level* yang dimainkan terlalu mudah dimainkan sehingga permainan cepat berakhir.

Berikut perbandingan *game* sejenis yang telah dijelaskan di atas.

Tabel III.1 Perbandingan *game* sejenis

Nama *Game*	*Genre Game*	Desain Grafis	*Storyline* dan *gameplay*	*Multi player*	Basis
P10NER, Pertempuran 10 November	*First Person Shooter*	Cukup, 3D tetapi lebih terkesan kartun dan penuh warna.	Cukup, alur cerita cukup sama dengan fakta sejarah pertempuran 10 November 1945	Tidak.	*Desktop*
***Game* Merdeka atau Mati**	*Platform*	Kurang, 2D masih belum mampu	Kurang, alur cerita tidak jelas sama sekali	Tidak.	*Mobile*

III.1.3 Analisis game yang dibangun

Game Pertempuran 10 November 1945 Surabaya yang akan dibangun yaitu *game* yang mempunyai grafis 3D yang ber-*genre first person shooter (FPS). Game* ini mempunyai alur cerita yang sama seperti sejarah Pertempuran 10 November 1945, dimana alur cerita pertempuran 10 November dimulai dari percobaan agresi militer oleh sekutu terhadap kota Surabaya.

Game dibangun dengan konsep sudut pandang orang pertama atau *first person shooter* dengan tujuan agar kesan pertempuran yang dihadirkan lewat game lebih terlihat nyata.

III.1.3.1 Storyline

Game ini akan menceritakan kronologis pertempuran 10 November 1945 Surabaya, di mulai dari Insiden Hotel Yamato, terbunuhnya Jend. Mallaby hingga Pertempuran besar-besaran terhadap tentara Sekutu pada tanggal 10 November 1945. Pemain merupakan seorang pejuang bernama Kuesno Wibowo yang

mengetahu jika pihak Belanda telah mengibarkan bendera mereka di atas Hotel Yamato, hal tersebut menimbulkan kemarahan rakyat Surabaya, tidak terkecuali Kuesno Wibowo seorang pejuang dari Surabaya. Kuesno Wibowo diharuskan untuk mencari Hotel Yamato dan menghadapi serangan dari tentara Belanda, lalu menurunkan bendara Belanda dengan menaiki atap Hotel Yamato.

Setelah Insiden Hotel Yamato berakhir, pemain akan menghadapi kemarahan dari tentara Inggris akibat dari terbunuhnya seorang Jendral mereka yang bernama Mallaby. Hal tersebut memicu terjadinya pertempuran pada tanggal 30 Oktober 1945, pemain diharuskan untuk betahan dari serangan tentara Inggris lalu berusaha untuk merebut sebagian wilayah kota Surabaya.

Pertempuran pada tanggal 30 Oktober 1945 memicu dibuatnya sebuah ultimatum dari pihak Inggris pada tanggal 9 November 1945 yang berisi tuntutan pertanggung jawaban pihak Indonesia terhadap terbunuhnya Jend. Mallaby dan menuntuk tentara Indonesia untuk menyerahkan diri dengan mengangkat tangan ke atas. Hal ini tentu saja ditolak mentah-mentah olah rakyat Surabaya hingga pada tanggal 10 November 1945 pecahlah pertempuran terbesar dalam sejarah pertempuran Indonesia. Pada insiden kali ini, pemain diharuskan untuk malakukan kontak senjata secara langsung terhadap pihak Sekutu dengan tujuan mengalahkanya dan menguasai wilayah Surabaya yang terpisahkan oleh Jembatan Merah.

III.1.3.2 Deskripsi Level

Dalam *game* ini terdapat *actor* dimana *actor* tersebut harus menyelesaikan 3 *level* yang tiap-tiap *level*nya memiliki rintangan yang berbeda-beda. Pada *level* pertama *actor* harus menurunkan bendera Belanda yang dikibarkan di Hotel Yamato oleh serdadu Belanda dan pada *level* ini juga *actor* akan dihadapkan dengan serangkaian-serangkaian pertempuran dengan tentara belanda. Pada *level* kedua, Ada satu kejadian dimana seorang jenderal Inggris yang bernama Jenderal Mallaby tewas ketika berkonfrontasi dengan rakyat surabaya, kematian jenderal tersebut membuat Inggris mencoba untuk menyerang tentara kita dan pada *level* ke dua ini *actor* harus berusaha untuk bertahan dari

serangan tentara inggris dan mengamankan wilayahnya. Pada *level* ketiga, *actor* akan dihadapkan pada pertempuran besar-besaran melawan tentara inggris dan ini merupakan akhir dari *game* ini.

III.1.3.3 Gameplay

Tugas utama pemain dalam memainkan *game* ini adalah menyelesaikan misi di setiap *level* yang dimainkan, setiap *level* yang dimainkan memiliki tingkat kesulitan yang berbeda-beda, di setiap *level* pemain akan diberikan beberapa petunjuk berupa *icon* panah yang apabila di ambil akan mengeluarkan informasi berupa teks dan suara. Petunjuk ini berguna apabila pemain kesulitan mencari informasi tentang apa yang harus dilakukan ketika memainkan *game* ini. Selain mendapatkan petunjuk berupa *icon* yang berisi informasi, pemain diberikan juga 2 *item* senjata yang berbeda, senjata pertama merupakan senjata laras panjang yang memiliki peluru tidak terbatas dan senjata kedua berupa roket yang memiliki peluru terbatas. Ketika berada di dalam arena permainan, pemain akan dihadapkan dengan beberapa musuh yang terdapat di beberapa *spot* / tempat yang berbeda.

a. *Level* 1

Misi yang harus diselesaikan adalah sebagai berikut :

- Menemukan Hotel Yamato
- Menurunkan bendera Belanda yang berada di atas Hotel Yamato.

b. *Level* 2

Misi yang harus diselesaikan adalah sebagai berikut :

- Menyelamatkan seorang tentara indonesia / karakter teman
- Bertahan dari serangan tentara Inggris dan menguasai wilayah yang dikuasai oleh tentara Inggris.
- Mencari sebuah rumah untuk menyelamatkan karakter utama sekaligus mengakhiri permainan pada level 2.

c. *Level* 3

Misi yang harus diselesaikan adalah sebagai berikut :

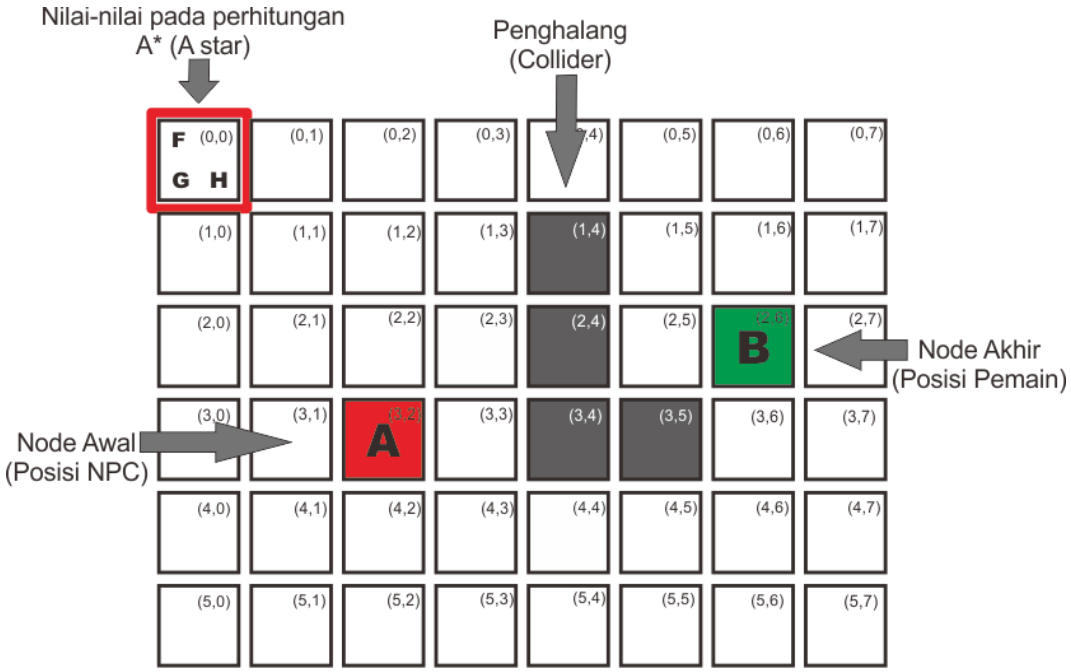
- Melakukan pertempuran secara besar-besaran terhadap tentara Inggris / musuh dengan tujuan untuk merebut wilayah yang dikuasai oleh

tentara Inggris / musuh yaitu wilayah yang terhalang oleh sebuah jembatan yaitu Jembatan Merah.

III.2 Analisis AI

III.2.1 Analisis A* (A Star)

Algoritma A* di dalam *game* akan diterapkan pada NPC (*Non Playable Character*) sebagai penentuan jalur terpendek atau *pathfinding* menuju *target.* Cara kerja algoritma A* di dalam *game first person shooter* 3D sejarah Pertempuran 10 November 1945 surabaya lebih jelasnya sebagai berikut.



Gambar III.8 Penjelasan A*

Pada gambar III.8 terdapat beberapa kotak yang memiliki koordinat dari (0,1) sampai (5,7). Kotak atau *node* yang masing-masing telah memiliki koordinat merupakan *node* yang digunakan untuk melakukan pencarian jalur terpendek / *pathfinding*. Terdapat beberapa *node* yang memiliki warna berbeda, berikut penjelasan dari masing -masing *node* tersebut :

- *Node* warna merah : Node awal dari posisi NPC (*Non Playbel Character*) yang berada pada node di koordinat (3,1)

- *Node* warna abu : Node penghalang (*collider*) yang berada pada *node* di koordinat (1,3), (2,3), (3,3) dan (3,4).
- *Node* warna hijau : *Node* dari posisi pemain atau *player* yang berada pada *node* di koordinat (2,5).

Untuk nilai-nilai yang digunakan untuk perhitungan A star, terdapat 3 nilai yaitu :

- Nilai F, nilai yang diperoleh dari penjumlahan nilai G, jumlah nilai tiap simpul dalam jalur terpendek dari *starting point* ke *current node*, dan H, jumlah nilai perkiraan dari sebuah simpul ke *target point*.
- Nilai G, nilai atau biaya yang dikeluarkan dari *node* awal / *node* A ke *node* yang ada di sekitarnya.
- Nilai H, nilai *heuristic* yang digunakan untuk menghitung perkiraan biaya yang dikeluarkan dari *node* awal / *node* A menuju *node* tujuan / *node* B.

  Perhitungan nilai F didapat dari persamaan F= G+H.

Sedangkan untuk nilai G digunakan nilai 10 untuk gerakan *vertical* atau *horizontal*, angka 10 digunakan hanya untuk memudahkan perhitungan. Sedangkan untuk gerakan *diagonal* maka akan dikalikan dengan √2 yang berarti 10x√2 = 14,14. Untuk memudahkan perhitungan angka 14,14 dibulatkan menjadi 14.

Untuk perhitungan nilai H digunakan fungsi *heuristic*, metode yang digunakan di dalam contoh ini adalah metode *Manhattan* dimana perhitungan jumlah *node* hanya yang bergerak secara *vertical* dan *horizontal* menuju tujuannya serta mengabaikan penghalang atau *collider*, yang kemudian nilainya dikalikan dengan 10. Atau dirumuskan dengan:

H = 10*(abs(*current*X-*target*X) + abs(*current*Y-*target*Y))

Pada pencarian jalur terpendek oleh *Astar* ada beberapa istilah yang digunakan seperti *open list, closed list, parrent, current node* dan *pointer*.

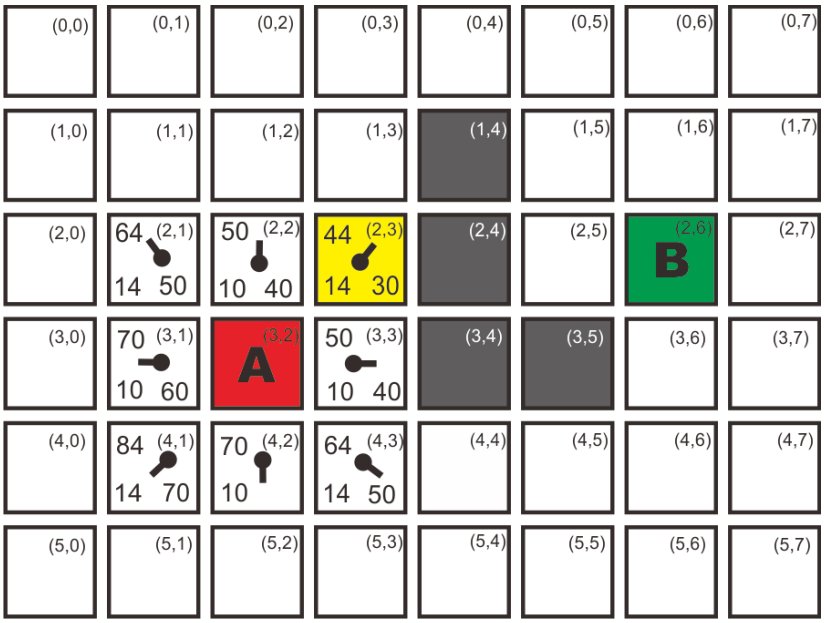
a. *Open list* adalah *list* untuk memeriksa *node-node*.
b. *Closed list* adalah *node-node* yang diabaikan atau *node-node* yang tidak perlu diperiksa.
c. *Parrent* adalah digunakan untuk menelusuri jalan.

d. *Current Node* adalah simpul yang sedang dijalankan dalam algoritma pencarian jalan terpendek.
e. *Pointer* adalah penunjuk jalan terhadap *Node* A *(back pointer*) ketika *node* B (target) telah ditemukan.

Langkah berikutnya ketika telah mendapatkan *node* awal adalah menentukan rute terpendek dengan cara memeriksa setiap *node* yang berdekatan dengan *node* A sampai *node* B. Di bawah ini merupakan penjelasan langkah-langkahnya.

1. Langkah pertama adalah memasukan *node* A kedalam *open list*.
2. Langkah kedua adalah memeriksa *node-node* yang berdekatan dengan *node* A, abaikan *node* dengan collision atau penghalang lalu tambahkan semua *node* tersebut kedalam *open list*. *Node* A sendiri nantinya akan menjadi "*parent*" dan digunakan untuk menelusuri jalan.
3. Langkah ketiga adalah menghapus *node* A dari *open list*, dan dimasukan ke dalam *closed list*.

Berikut ini perhitungan pada langkah kedua yang berada pada node dengan koordinat (2,3).



Gambar III.9 Penjelasan Astar pada langkah pertama

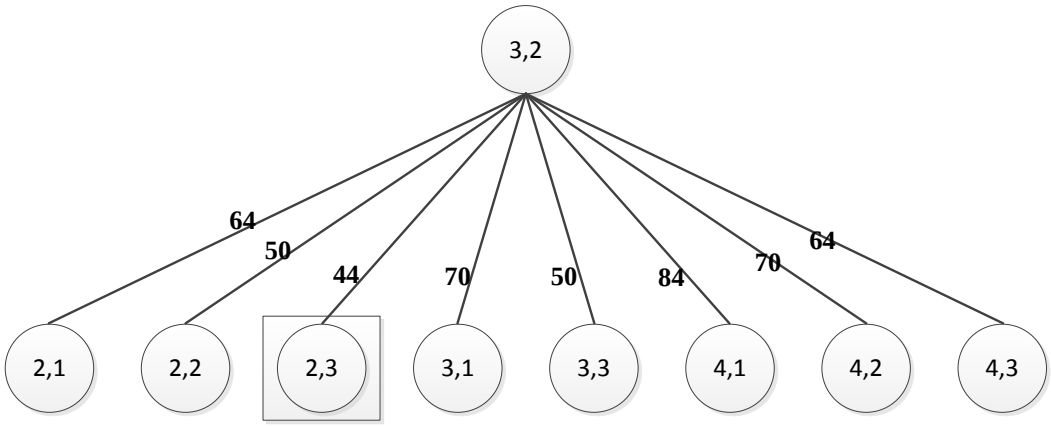
Pada langkah pertama, *node* (3,2) dimasukan ke dalam *open list* dan selanjutnya memeriksa *neighbor node* atau *node* yang berdekatan dengan *node* (3,2) yaitu, *node* (2,1), (2,2), (2,3), (3,1), (3,3), (4,1), (4,2), (4,3). Setelah diperiksa *node-node* tersebut dimasukan ke dalam *open list*. Hapus *node* (3,2) dari *open list* dan masukan *node* tersebut kedalam *closed list*. Lalu dari *node-node* yang berada di dalam *open list* pilih *node* dengan nilai F yang paling rendah, yaitu *node* (2,3) dengan nilai F = 44.

Langkah pertama menghasilkan *node-node* di dalam *open list* dan *closed list* dijelaskan dengan tabel sebagai berikut.

Tabel III.2 Tabel Perhitungan langkah pertama

Open List	(2,1), (2,2), (2,3), (3,1), (3,3), (4,1), (4,2), (4,3)
Closed list	(3,2)
Parent	(3,2)
Current node	(2,3)

Gambat III.10 merupakan penjelasan langkah pertama pencarian jalur terpendek / *pathfinding* menggunakan *tree*. *Node* (2,3) merupakan *current node* yang dihasilkan pada perhitungan langkah pertama.

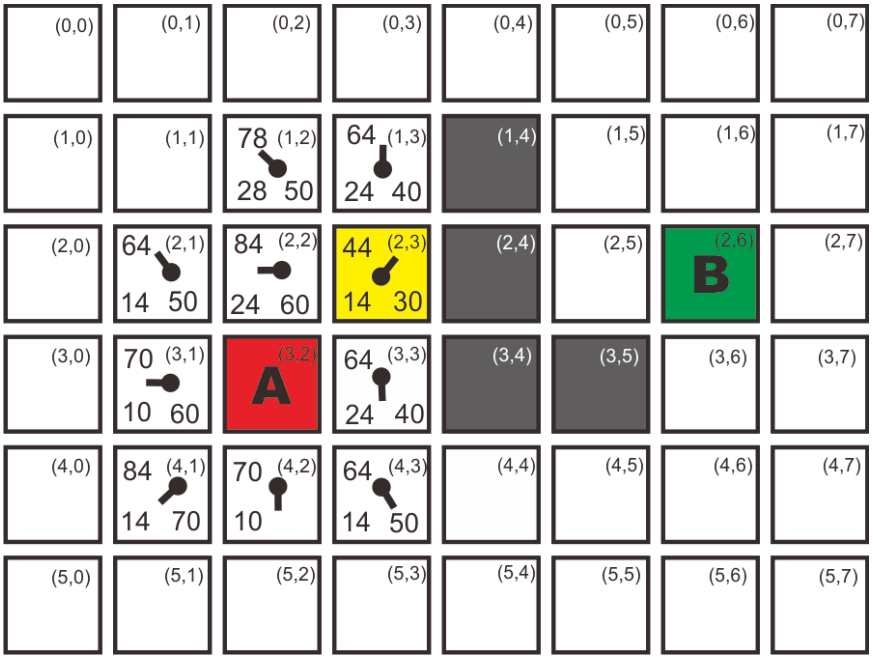


Gambar III.10 Penjelasan A* dengan *tree* pada langkah pertama

Proses yang dijelaskan di atas, dilakukan berulang kali atau *looping*, dimulai dari *open list* dan menemukan *node* dengan nilai F paling rendah. Langkahnya adalah sebagai berikut :

1. Hapus *current node* tersebut dari *open list* dan masukkan ke dalam *closed list.*
2. Periksa semua *node* yang berdekatan dengan *current node*, abaikan *node* yang *unwalkable* atau dalam contoh ini adalah *node* penghalang. Jika tidak ada pada *open list*, tambahkan ke dalam *open list.* Dan tambahkan "*parent*" untuk *node* tersebut.
3. Jika *node* sudah ada pada *open list,* periksa apakah ini jalan yang lebih baik dari *node* awal, dengan kata lain periksa apakah nilai G-nya lebih rendah apabila menggunakan *node* ini, jika lebih rendah maka hitung ulang nilai F dan G serta merubah arah *pointer,* jika tidak maka jangan lakukan apa-apa.

Berikut ini perhitungan pada langkah kedua yang berada pada *node* dengan koordinat (2,3).



Gambar III.11 Penjelasan *Astar* pada langkah kedua

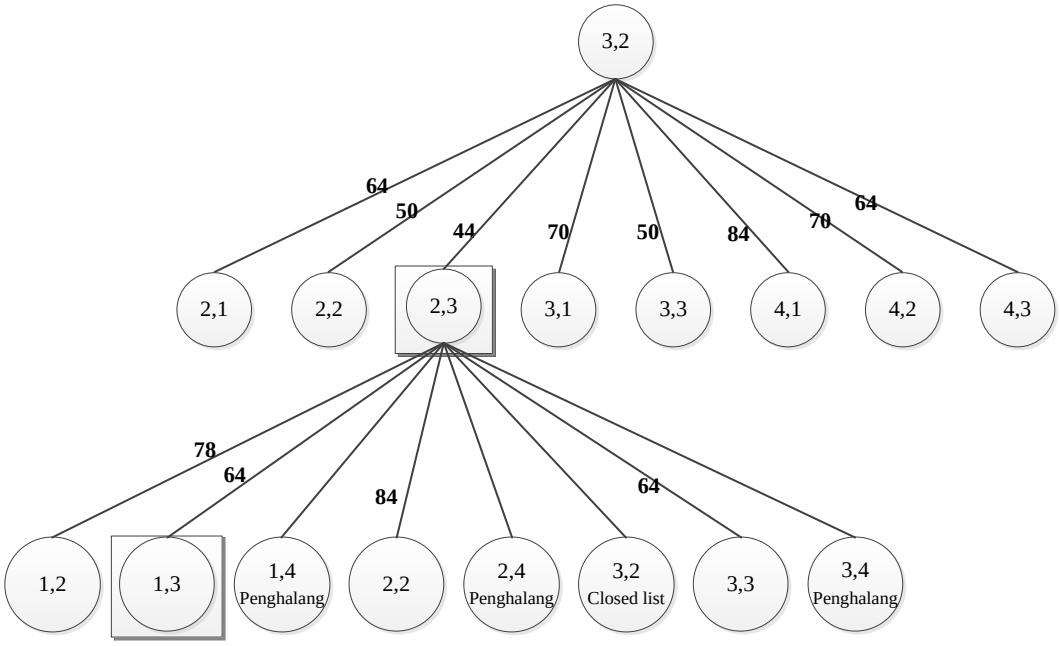
Pada gamber III.11 terlihat ada 4 *node* yang masuk ke dalam *open list* yaitu *node* (1,2), (1,3), (2,2) dan (3,3). Sedangkan untuk *node* (2,3), (3,2) merupakan *node* c*losed list* dan *node* (1,4), (2,4), (3,4) merupakan penghalang atau *collider*. Dari keempat *node* yang masuk ke dalam *open list* terdapat 2 node yang mempunyai nilai F yang sama yaitu *node* (1,3) dan (3,3). Jika kondisi seperti ini terjadi maka sistem akan memeriksa *node-node* yang masih berada di *open list* yaitu *node* (2,2) dan *node* (1,2) untuk di cek apakah nilai dari kedua *node* tersebut tidak lebih kecil dari *node* (1,3) dan (3,3). Karena nilai dari node (2,2) dan (1,2) tidak lebih kecil, maka sistem akan memilih *node* (1,3) sebagai *node* yang memiliki nilai F paling kecil meskipun nilai F *node* tersebut sama dengan *node* (3,3). Hal ini disebabkan sistem akan sistem akan menentukan nilai F pada *node* terakhir yang ditambahkan ke dalam *open list.*

Berikut ini penjelasan melalui tabel untuk pencarian jalan terpendek pada langkah kedua.

Tabel III.3 Tabel Perhitungan langkah kedua

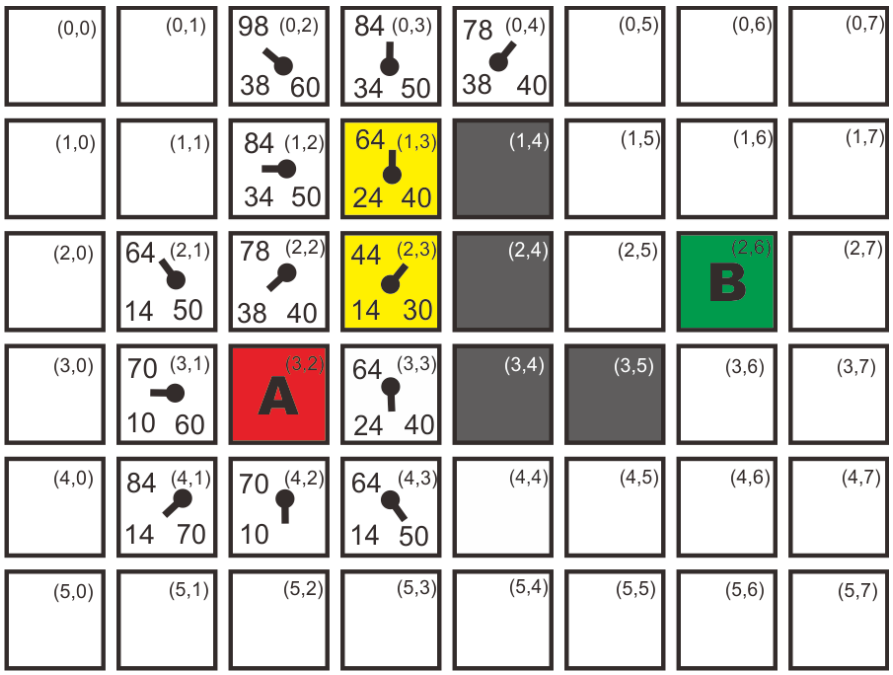
Open List	(1,2), (1,3), (2,2), (3,3)
Closed list	(3,2), (2,3)
Parent	(2,3)
Current node	(1,3)

Gambat III.12 merupakan penjelasan pencarian jalur terpendek / *pathfinding* menggunakan *tree*. *Node* (1,3) merupakan *current node* yang dihasilkan pada perhitungan langkah kedua.



Gambar III.12 Penjelasan A* dengan *tree* pada langkah kedua

Gambar III.13 menunjukan perhitungan pada langkah ketiga yang berada pada *node* dengan koordinat (1,3).



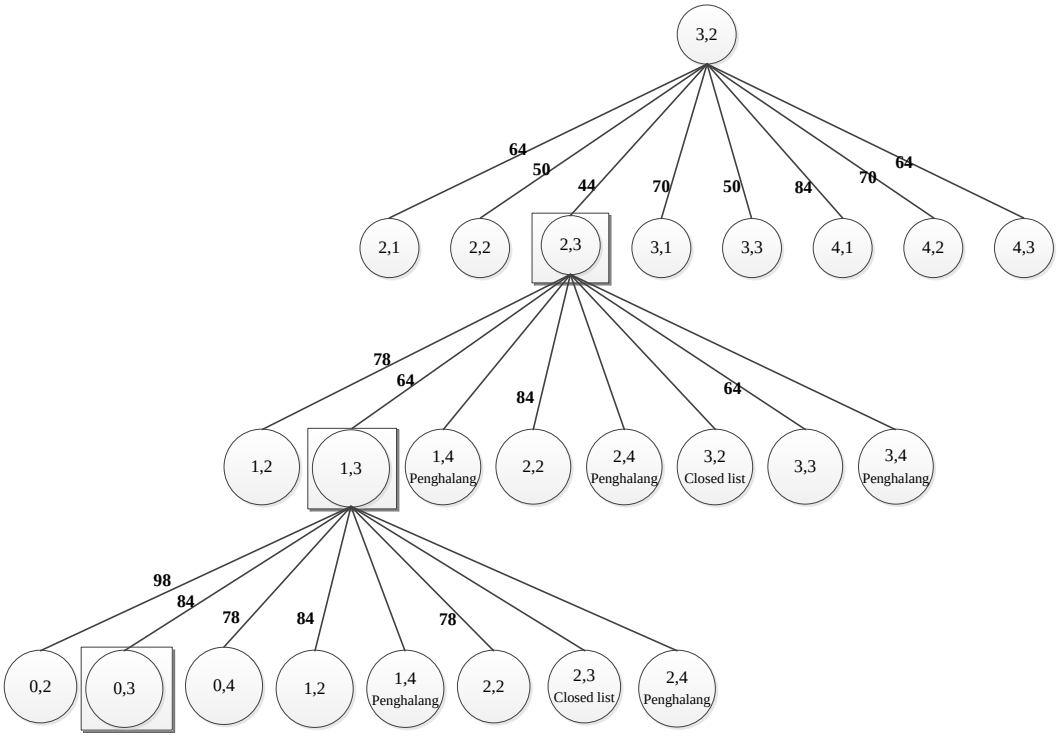
Gambar III.13 Penjelasan *Astar* pada langkah ketiga

Berikut ini penjelasan melalui tabel III.4 untuk pencarian jalan terpendek pada langkah ketiga.

Tabel III.4 Tabel Perhitungan langkah ketiga

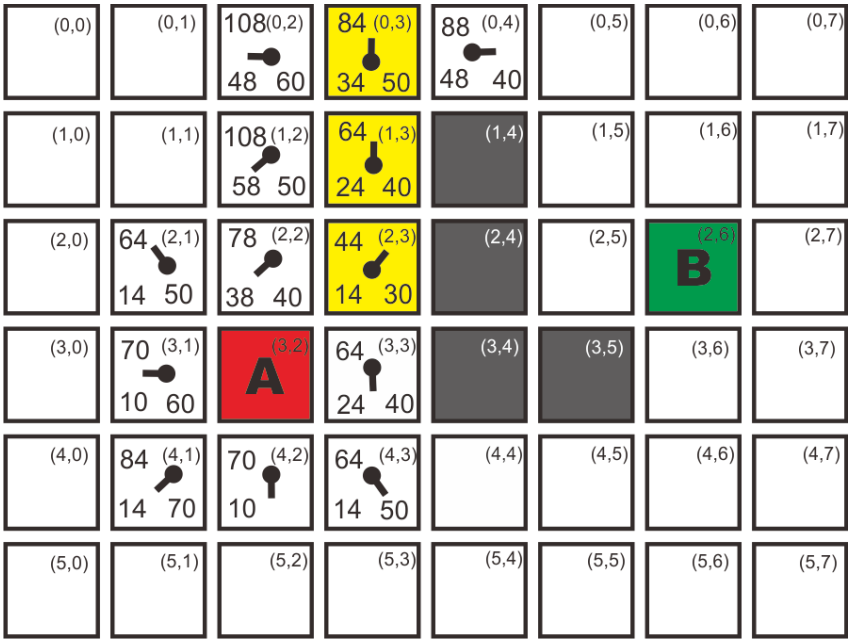
Open List	(0,2), (0,3), (0,4), (1,2), (2,2)
Closed list	(3,2), (2,3), (1,3)
Parent	(1,3)
Current node	(0,3)

Gambat III.14 merupakan penjelasan pencarian jalur terpendek / *pathfinding* menggunakan *tree*. *Node* (0,3) merupakan *current node* yang dihasilkan pada perhitungan langkah ketiga.



Gambar III.14 Penjelasan A* dengan *tree* pada langkah ketiga

Gambar III.15 menunjukan perhitungan pada langkah keempat yang berada pada *node* dengan koordinat (0,3).



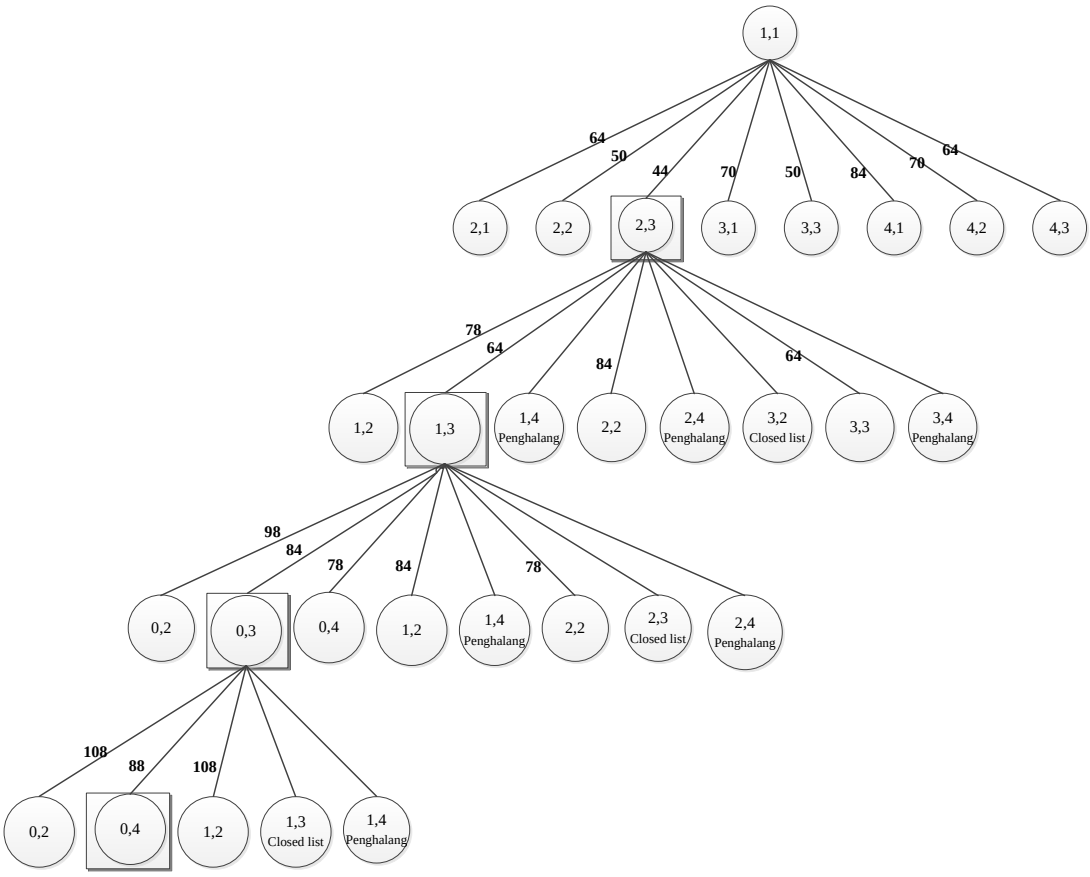
Gambar III.15 Penjelasan *Astar* pada langkah keempat

Berikut ini penjelasan melalui tabel untuk pencarian jalan terpendek pada langkah keempat.

Tabel III.5 Tabel Perhitungan langkah keempat

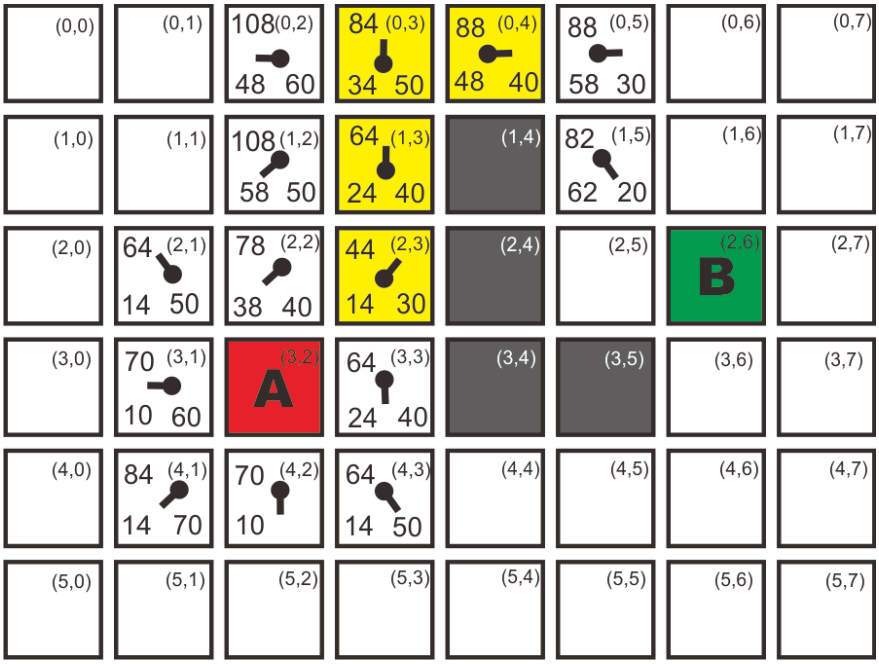
Open List	(0,2), (0,4), (1,2)
Closed list	(3,2), (2,3), (1,3), (0,3)
Parent	(0,3)
Current node	(0,4)

Gambat III.16 merupakan penjelasan pencarian jalur terpendek / *pathfinding* menggunakan *tree*. *Node* (0,4) merupakan *current node* yang dihasilkan pada perhitungan langkah keempat.



Gambar III.16 Penjelasan A* dengan *tree* pada langkah keempat

Gambar III.17 menunjukan perhitungan pada langkah kelima yang berada pada *node* dengan koordinat (0,4).



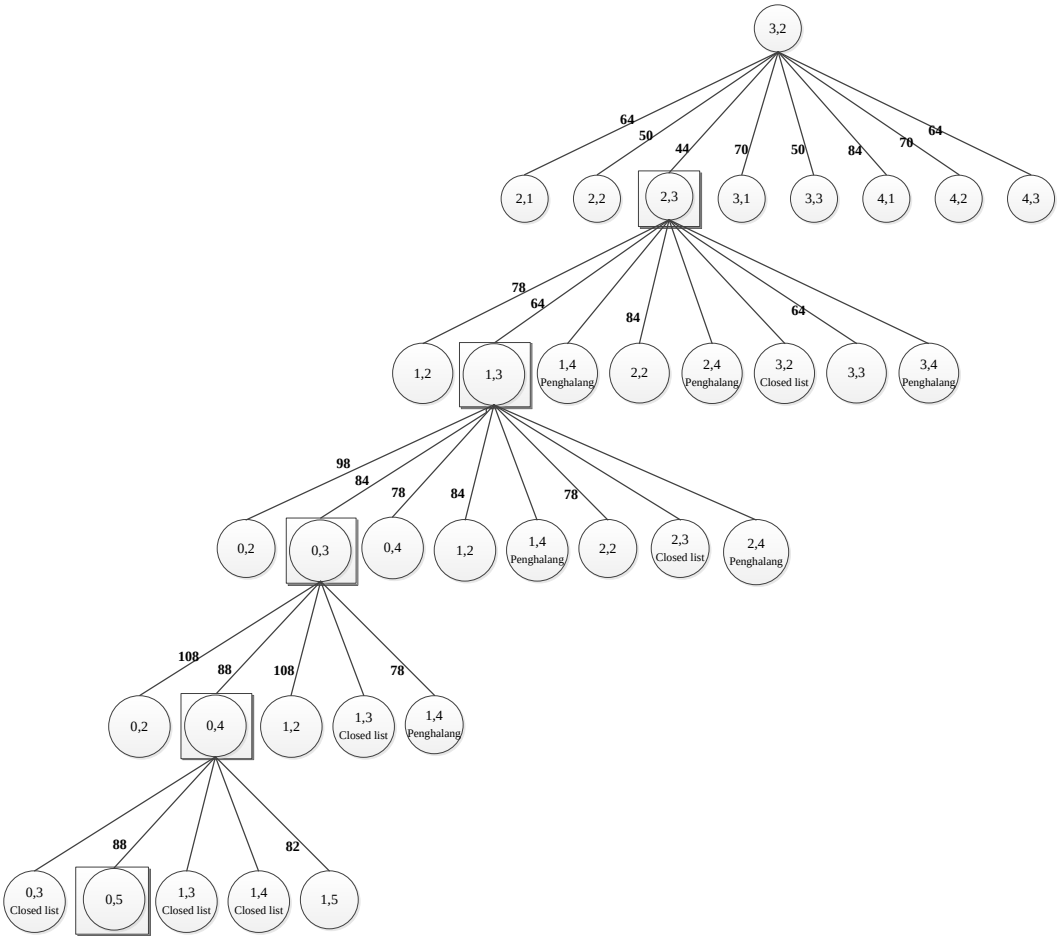
Gambar III.17 Penjelasan *Astar* pada langkah kelima

Berikut ini penjelasan melalui tabel untuk pencarian jalan terpendek pada langkah kelima.

Tabel III.6 Tabel Perhitungan langkah kelima

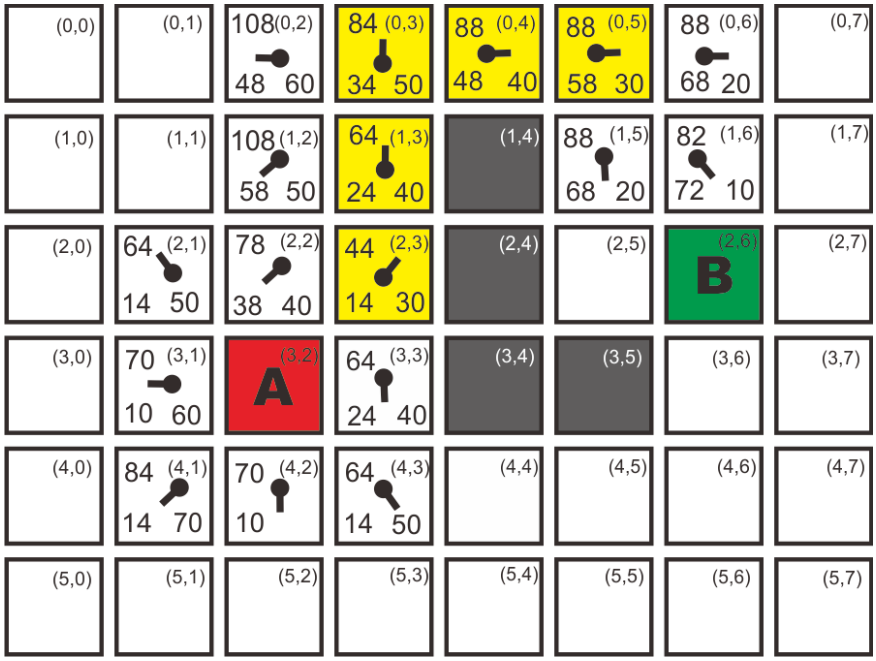
Open List	(0,5), (1,5)
Closed list	(3,2), (2,3), (1,3), (0,3), (0,4)
Parent	(0,4)
Current node	(0,5)

Gambat III.18 merupakan penjelasan pencarian jalur terpendek / *pathfinding* menggunakan *tree*. *Node* (0,5) merupakan *current node* yang dihasilkan pada perhitungan langkah kelima.



Gambar III.18 Penjelasan A* dengan *tree* pada langkah kelima

Gambar III.19 menunjukan perhitungan pada langkah kelima yang berada pada *node* dengan koordinat (0,5).



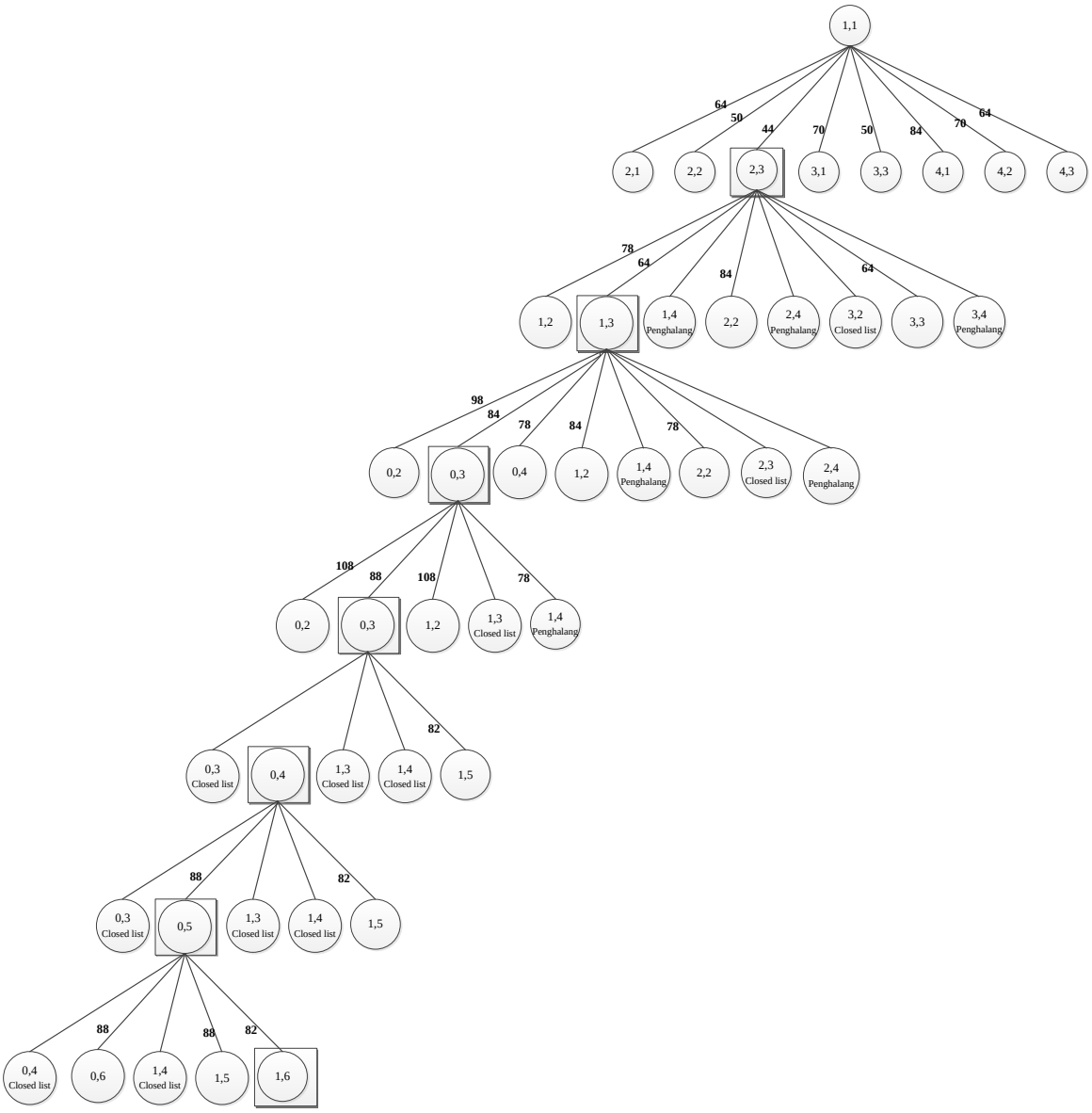
Gambar III.19 Penjelasan *Astar* pada langkah keenam

Berikut ini penjelasan melalui tabel untuk pencarian jalan terpendek pada langkah keenam.

Tabel III.7 Tabel Perhitungan langkah keenam

Open List	(0,6), (1,5), (1,6)
Closed list	(3,2), (2,3), (1,3), (0,3), (0,4), (0,5)
Parent	(0,5)
Current node	(1,6)

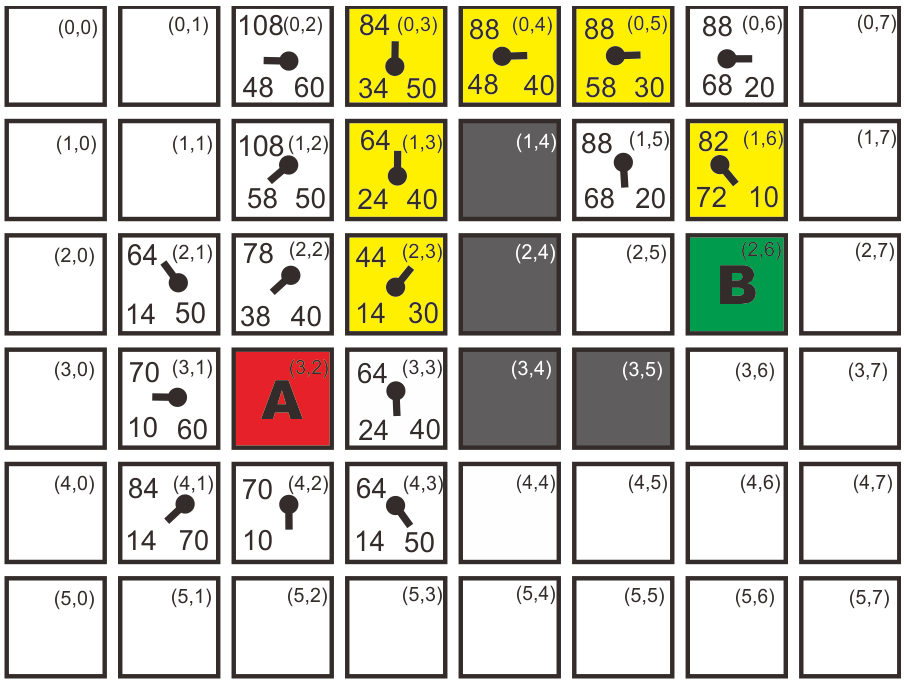
Gambat III.20 merupakan penjelasan pencarian jalur terpendek / *pathfinding* menggunakan *tree*. *Node* (1,6) merupakan *current node* yang dihasilkan pada perhitungan langkah keenam.



Gambar III.20 Penjelasan A* dengan *tree* pada langkah keenam

Penentuan jalan terbaik dengan melakukan pemeriksaan dimulai dari *target* atau *node* B, dalam ilustrasi adalah kotak berwarna hijau. Dengan mengikuti arah *pointer* yang berarti akan membawanya kembali ke *node* awal, maka akan didapatkan jalan terbaiknya, Maka jika diimplementasikan pada *game*, nantinya musuh / NPC akan melewati halangan dan akhirnya akan menemukan rute terpendek ke tujuan, lebih jelasnya seperti gambar berikut.

Berikut ini perhitungan pada langkah terakhir yang berada pada *node* dengan koordinat (1,6).

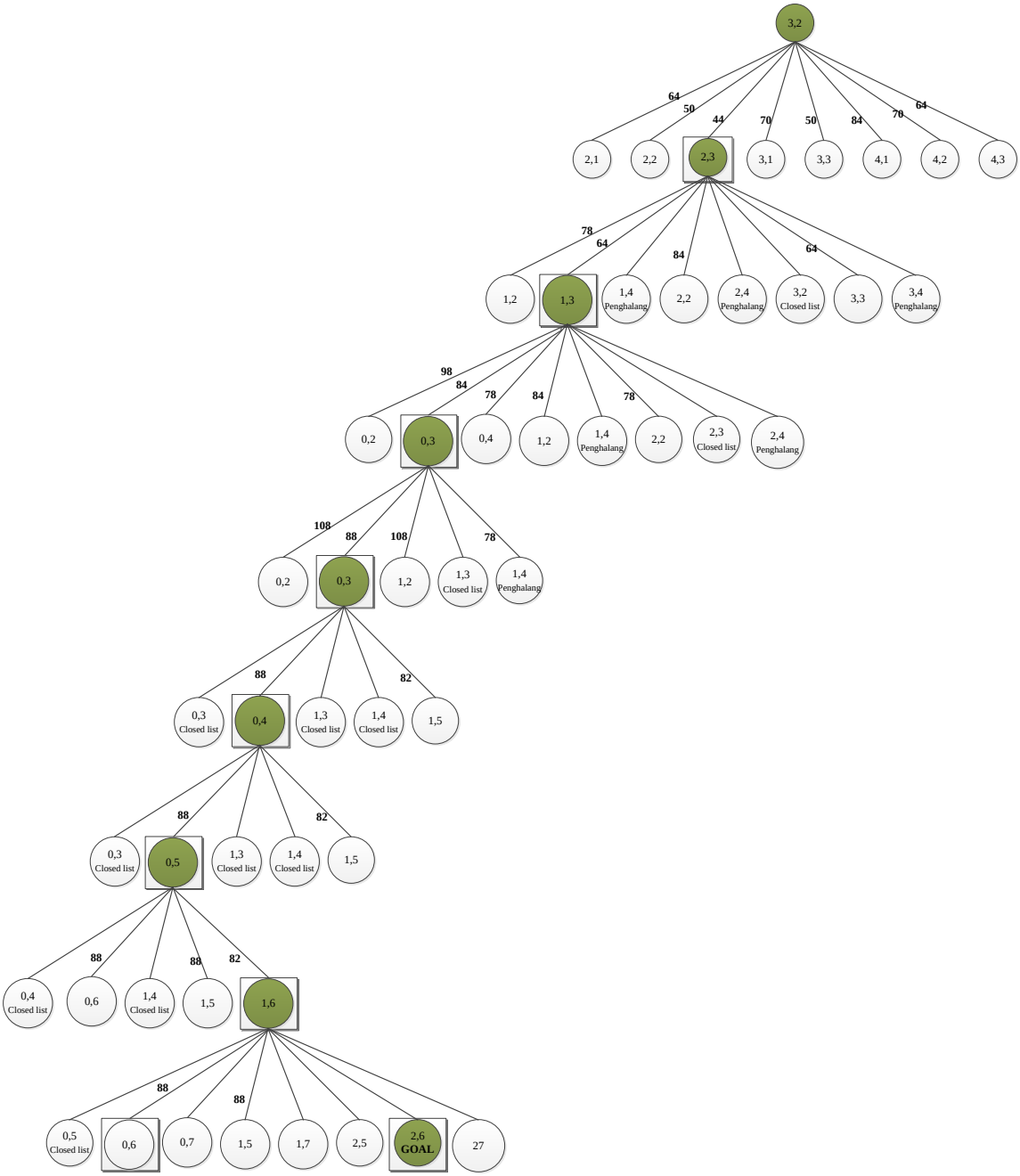


Gambar III.21 Penjelasan *Astar* pada langkah terakhir

Berikut ini penjelasan melalui tabel untuk pencarian jalan terpendek pada langkah terakhir.

Tabel III.8 Tabel Perhitungan langkah terakhir

Open List	(0,6), (0,7), (1,5), (1,7), (2,5), (2,7)
Closed list	(3,2), (2,3), (1,3), (0,3), (0,4), (0,5), (1,6)
Parent	(1,6)
Current node	(2,6)



Gambar III.22 Penjelasan A* dengan *tree* pada langkah terakhir

III.2.2 Analisis Fuzzy Logic

Pada *game* ini *fuzzy logic* digunakan untuk menentukan variasi prilaku yang dilakukan oleh NPC. Dengan adanya *fuzzy logic* masing-masing NPC dapat merubah perilaku berbasis perubahan variabel masukan. Metode *fuzy* yang digunakan adalah metode sugeno karena metode ini menghasilkan keluaran yang berupa konstan tegas, sehingga dapat mewakili nilai perilaku yang telah dirancang.

Untuk menghasilkan prilaku pada NPC ada dua variabel yang digunakan, yaitu Jarak (dekat, sedang dan jauh) dan Kekuatan (lemah dan kuat). Dengan menggunakan dua variabel diharapkan NPC mampu menentukan prilaku yang akan dilakukan.

Tahapan-tahapan dari analisis *fuzzy logic* 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 sebagai variabel *input*.
3. Variabel Keputusan sebagai variabel *output*.

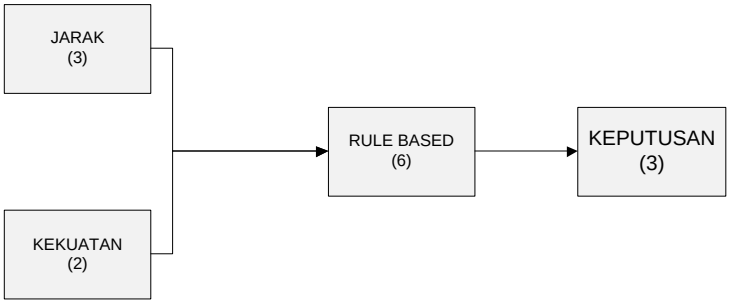
2. Nilai linguistik

Dari tiga variabel yang digunakan, maka nilai linguistiknya sebagai berikut:

1. Variabel Jarak, terbagi menjadi 3 *input* yaitu : Dekat, Sedang, Jauh.
    1. Dekat
    2. Sedang
    3. Jauh
2. Variabel Kekuatan, terbagi menjadi 2 *input* yaitu : Lemah dan Kuat.
    1. Lemah
    2. Kuat
3. Variabel Keputusan, terbagi menjadi 3 *output* yaitu :
    1. Menyerang
    2. Bertahan
    3. Melarikan Diri.

3. Fuzzyfikasi

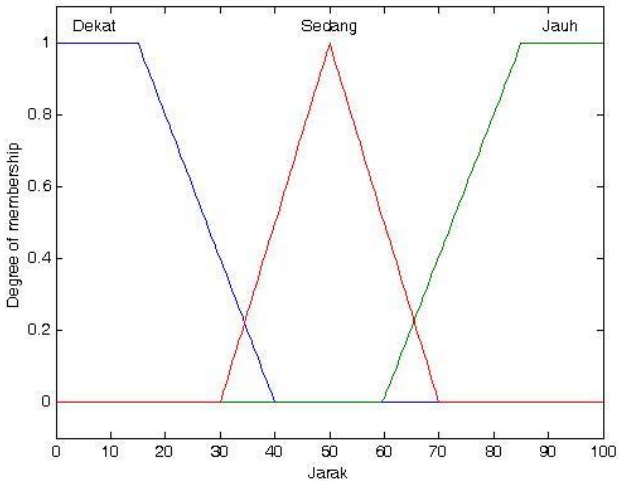
Fuzzyfikasi merupakan proses memetakan nilai *crisp* (numerik) ke dalam himpunan *fuzzy* dan menentukan derajat keanggotaannya. Secara garis besar pemetaan nilai *crisp* ke dalam himpunan *fuzzy* dijelaskan dengan gambar sebagai berikut.



Gambar III.23 *Fuzzy Interface System*

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

1. Variabel Jarak, terbagi menjadi 3 input yaitu : Dekat, Sedang dan Jauh. *Range* nilai untuk variabel Jarak antara 0-100 akan dijelaskan sebagai berikut :
    1. Dekat = 0 - 40
    2. Sedang = 30 - 70
    3. Jauh = 60 -100



Gambar III.24 Grafik *input* jarak

Pada gambar III.24 Menunjukan sebuah grafik Jarak 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 Variable Jarak ada 2 yaitu fungsi Trapesium dan Fungsi Segitiga.

- Variabel linguistik Dekat yang memiliki *range* nilai antara 0 – 40 nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut 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 *range* nilai antara 30 – 70 nilai *fuzzyfikasi*nya dihitung berdasarkan fungsi segitiga, berikut persamaannya :

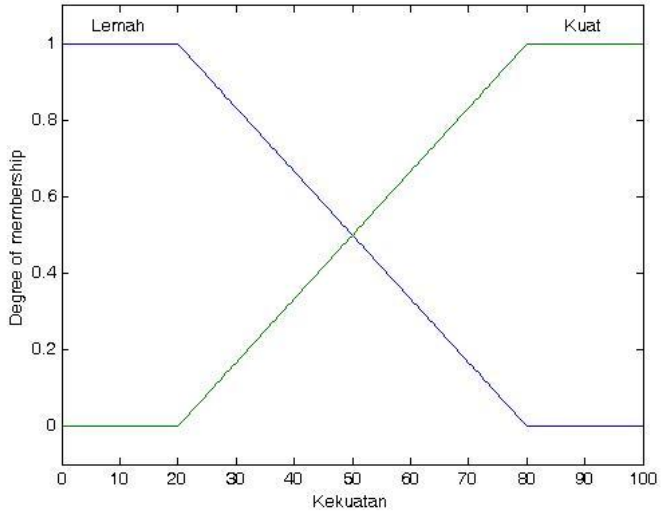
$$\mu Sedang[x]\begin{cases}0; x \leq 30/x \geq 70\\ \dfrac{x-30}{20}; 30 \leq x \leq 50\\ \dfrac{50-x}{20}; 50 \leq x \leq 70\end{cases}$$

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

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

2. Variabel Kekuatan, terbagi menjadi 2 input yaitu : Lemah dan Kuat. *Range* nilai variabel Kekuatan diantara 0 – 100 akan dijelaskan sebagai berikut :

1. Lemah = 0 - 80
2. Kuat = 20 – 100



Gambar III.25 Grafik *input* kekuatan

Pada gambar III.25 Menunjukan sebuah grafik trapesium Kekuatan yang mempunyai range nilai dari 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 *fuzzyfikasi*nya dihitung berdasarkan fungsi trapesium, berikut 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 *fuzzyfikasi*nya dihitung berdasarkan fungsi trapesium, berikut persamaannya :

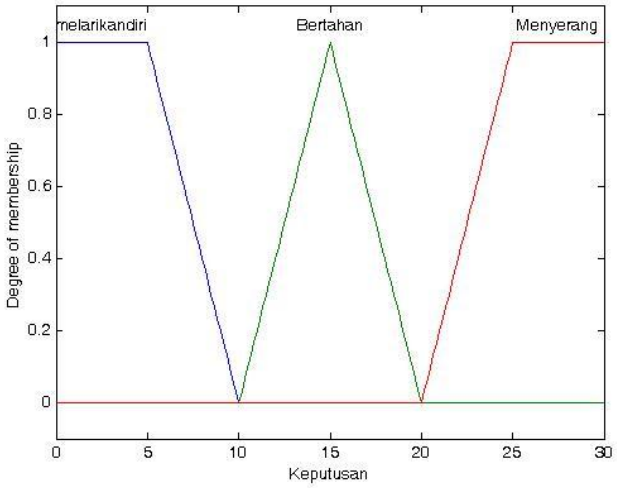
$$\mu Kuat[y]\begin{cases}0; y \leq 20\\ \dfrac{y-20}{60}\\ 1; y \geq 80\end{cases} ; 20 \leq y \leq 80$$

3. Variabel Keputusan, yaitu Menyerang, Bertahan dan Melarikan Diri. *Range* nilai variabel keputusan diantara 0-30, akan dijelaskan sebagai berikut :

   1. Melarikan diri = 0 -10

   2. Bertahan = 10 - 20

   3. Menyerang = 20 - 30



Gambar III.26 Grafik Keputusan

Pada gambar III.26 Menunjukan sebuah grafik keputusan yang mempunyai *range* nilai dari 0 – 30, setiap nilai lingustik seperti melarikan diri, bertahan dan menyerang mempunyai nilai *fuzzyfikasi* yang berbeda-beda.

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

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

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

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

4. Kaidah *fuzzy* atau *fuzzy's rules*

Kaidah *fuzzy (rules)* atau aturan-aturan yang diterapkan dalam penentuan perilaku NPC berjumlah enam *rules* yaitu:

1. *If* (Jarak *is* Dekat) *and* (Kekuatan is Lemah) *then* (Perilaku NPC *is* Melarikan diri)
2. *If* (Jarak *is* Dekat) *and* (Kekuatan *is* Kuat) *then* (Perilaku NPC *is* Menyerang)
3. *If* (Jarak *is* Sedang) *and* (Kekuatan *is* Lemah) *then* (Perilaku NPC *is* Bertahan)
4. *If* (Jarak *is* Sedang) *and* (Kekuatan *is* Kuat) *then* (Perilaku NPC *is* Menyerang)
5. *If* (Jarak *is* Jauh) *and* (Kekuatan *is* Lemah) *then* (Perilaku NPC *is* Melarikan diri)

6. *If* (Jarak *is* Jauh) *and* (Kekuatan *is* Kuat) *then* (Perilaku NPC *is* 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 nilai *crisp.* *Defuzzifikasi* 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. Fuzzifikasi

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

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

$$\mu Sedang[35] = \frac{35-30}{20} = 0{,}2$$

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

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 Keanggotaan Dekat = 0,2
2. Fungsi Keanggotaan Sedang = 0,25
3. Fungsi 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

1. 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

2. 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**

III.3 Analisis dan Kebutuhan Non-Fungsional

Analisis non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

Pada analisis kebutuhan sistem non fungsional ini dijelaskan analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras dan analisis pengguna.

III.3.1 Analisis dan Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi diantara keduanya.

Perangkat lunak yang dibutuhkan pengembang untuk pembangunan *Game* ini adalah :

1. Sistem operasi *Windows Seven*.
2. *Unity* versi 3.5.0
3. *3ds Max* 2011
4. *Graphic card*
5. *DirectX* 9

Perangkat lunak yang dibutuhkan pengguna untuk memainkan *game* ini adalah :

1. Sistem operasi minimal *Windows Xp*
2. *Graphic card*
3. *DirectX* 9

III.3.2 Analisis dan Kebutuhan Perangkat Keras

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan suatu tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya.

Perangkat keras minimum bagi pengembang untuk membangun *game* ini adalah sebagai berikut :

1. *Processor Core 2 Duo* 2,13 Ghz
2. RAM 512 MB
3. VGA 512 MB
4. *Harddisk* 10 GB
5. *Mouse* dan *Keyboard*

Perangkat keras minimum bagi pengguna untuk memainkan *game* ini adalah sebagai berikut :

1. *Dual Core* 1,80 Ghz
2. RAM 256 MB
3. VGA 512 MB
4. *Harddisk* 1 GB
5. *Mouse* dan *Keyboard*

III.3.3 Analisis Pengguna

Pada analisis pengguna ini akan mencakup analisis beberapa parameter terhadap calon pemain dari applikasi.

Game yang akan dibangun ini dapat dimainkan oleh berbagai kalangan akan tetapi karena game ini mengangkat salah satu cerita tentang perjuangan rakyat surabaya, pertimbangan tersebut membuat *game* ini ditujukan secara khusus untuk siswa Sekolah Menengah Pertama kelas IX, karena pada kelas IX materi tentang sejarah pertempuran di Surabaya diberikan.

Ada 2 hal yang akan memudahkan pemain ketika memainkan *game* ini yaitu pengetahuan dan pengalaman, terutama pengetahuan dan pengalaman dalam memainkan *game first person shooter*.

1. *User Knowledge and Experience*

Berikut ini klasifikasi *knowledge adn experience* dari pengguna aplikasi.

Tabel III.9 Analisis klasifikasi *knowledge and experience*

Education Level	Bisa digunakan oleh berbagai kalangan, seperti pelajar, mahasiswa hingga masyarakat awam
Reading Level	Bisa digunakan oleh berbagai *level* pendidikan dengan reading *level* yang sedang
Typing Skills	*Average* (40 wpm) Tidak memerlukan *typing skills* yang tinggi
Computer Literacy	*Moderate* (menengah)
Task Experience	Bisa digunakan oleh dengan pengalaman penggunaan computer dan *game* yang sedang
System Experience	Bisa digunakan oleh dengan pengalaman penggunaan komputer dan *game* yang sedang
Application Experience	Bisa digunakan dalam semua sistem operasi *Windows xp / vista/ 7*
Native Language	Menggunakan satu bahasa, yakni Indonesia
Use Of Other System	Bisa di jalankan tanpa perlu *install* applikasi 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 III.10 Analisis *Users Physical Characteristic*

Age	16 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.

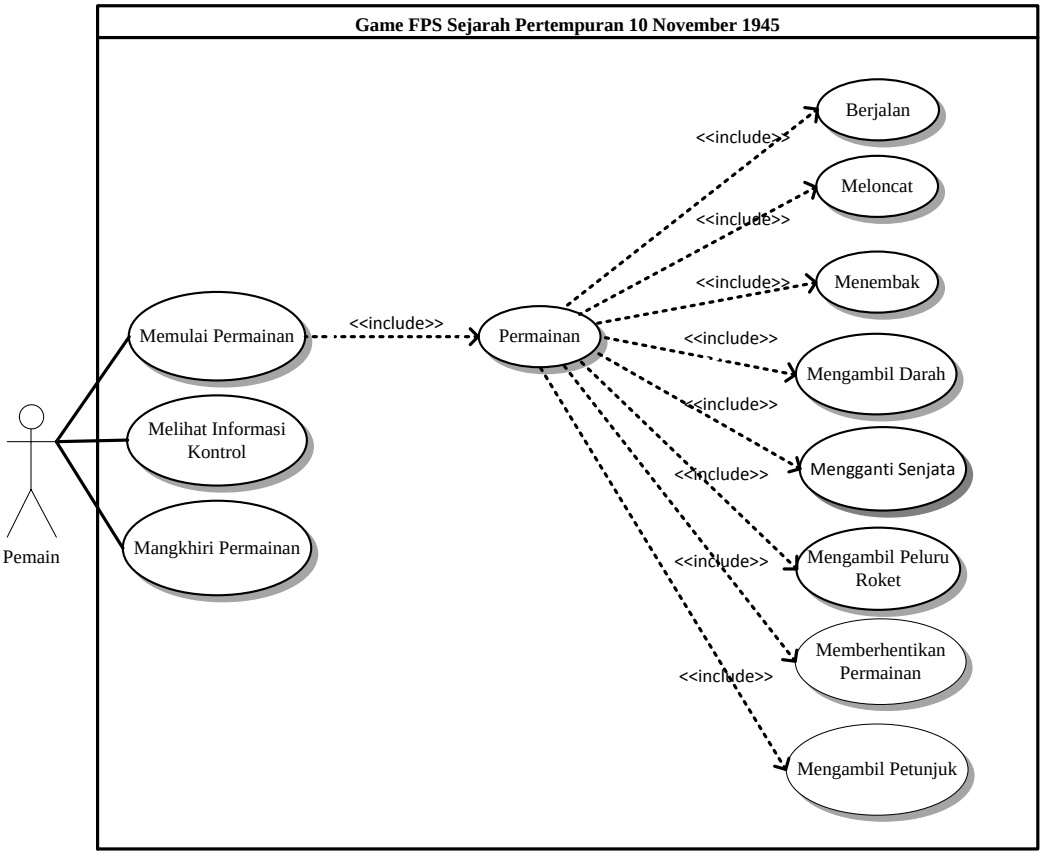
III.4 Analisis dan Kebutuhan Fungsional

Identifikasi aktor dapat dilakukan dalam analisis berorientasi objek dengan menggunakan UML yaitu menentukan aktor atau pengguna sistem. Aktor dalam konteks UML menampilkan peran pengguna atau sesuatu diluar sistem yang dikembangkan dapat berupa perangkat keras, *end user*, sistem yang lain dan sebagainya.

III.4.1 Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk prilaku *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Di bawah ini adalah perancangan *use case* yang terdapat pada *game first person shooter* 3D sejarah pertempuran 10 November Surabaya.



Gambar III.27 *Use Case Diagram*

III.4.1.1 Definisi Use Case

Definisi *use case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *use case*. Berikut di bawah ini tabel dari definisi *use case* yaitu :

Tabel III.11 Definisi *Use Case*

No	*Use Case*	Deskripsi
1	Memulai Permainan	Proses untuk memulai permainan
2	Melihat Informasi Kontrol	Proses untuk meilhat petunjuk permainan dan informasi kontrol
3	Mengakhiri Permainan	Proses untuk keluar dari permainan atau mengakhiri permainan
4	Permainan	Proses ketika pemain telah masuk ke dalam permainan.
5	Berjalan	Proses untuk menggerakan pemaian melalui interaksi dengan *keyboard*
6	Meloncat	Proses untuk membuat pemain melakukan tindakan meloncat melalui interaksi dengan *keyboard*
7	Menembak	Proses untuk menembakan peluru yang terdapat pada senjata
8	Mengambil Darah	Proses untuk menambah darah / *healt* ketika darah tersisa sedikit lagi
9	Mengganti Senjata	Proses untuk menukar senjata
10	Mengambil Peluru Roket	Proses untuk menambah peluru roket
11	Memberhentikan Permainan	Proses untuk menghentikan sementara permainan
12	Mengambil Petunjuk	Proses ketika pemain mengambil petunjuk permainan

III.4.1.2 Skenario Use Case

Skenario *use case* merupakan bagian pada *use case* yang menunjukkan proses apa saja yang terjadi pada setiap bagian di dalam *use case*, 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*. Berikut ini beberapa skenario *use case* berdasarkan *use case* yang ada yaitu :

Tabel III.12 Skenario *Use Case* Memulai Permainan

Identifikasi		
Nomor	1	
Nama	Memulai Permainan	
Tujuan	Melihat *scene* / cerita permainan dan masuk ke dalam permainan	
Deskripsi	Proses untuk melihat *scene* / cerita permainan dan masuk ke dalam permainan	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di menu utama	
Aksi Aktor		Reaksi Sistem
Memilih menu mainkan		
		Menampilkan *scene* / cerita tentang latar belakang terjadinya pertempuran
		Menampilkan permainan
Kondisi Akhir	Sistem menampilkan permainan	

Tabel III.13 Skenario *Use Case* Melihat Informasi Kontrol

Identifikasi	
Nomor	2
Nama	Melihat Informasi Kontrol
Tujuan	Menampilkan informasi kontrol
Deskripsi	Proses untuk menampilkan informasi kontrol
Aktor	Pemain

Skenario Utama		
Kondisi awal	Pemain berada di menu utama	
Aksi Aktor		Reaksi Sistem
Memilih menu petunjuk		
		Menampilkan informasi tentang kontrol *keyboard* dan *mouse*.
Kondisi Akhir	Sistem akan menampilkan informasi kontrol	

Tabel III.14 Skenario *Use Case* Mengakhri Permainan

Identifikasi	
Nomor	3
Nama	Mengakhiri Permainan
Tujuan	Keluar dari permainan
Deskripsi	Proses untuk mengakhiri permainan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	Reaksi Sistem
Memilih menu keluar	
	Keluar dari permainan atau mengakhiri permainan
Kondisi Akhir	Sistem akan keluar dari permainan

Tabel III.15 Skenario *Use Case* Permainan

Identifikasi	
Nomor	4
Nama	Permainan
Tujuan	permainan

Deskripsi	Proses ketika pemain telah masuk ke dalam permainan	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Sistem menampilkan permainan	
Aksi Aktor		Reaksi Sistem
		Sistem menampilkan permainan
Pemain mulai memainkan permainan		
Pemain berhasil menyelesaikan misi yang ada di dalam permainan		
		Sistem menampilkan tampilan misi yang telah selesai
Pemain gagal menyelesaikan misi yang ada di dalam permainan		
		Sistem kembali ke awal permainan
Kondisi Akhir	Sistem akan menampilkan tampilan misi selesai jika pemain berhasil menyelesaikan permainan dan sistem akan kembali ke awal permainan jika pemain gagal menyelesaikan misi permainan	

Tabel III.16 Skenario *Use Case* Berjalan

Identifikasi	
Nomor	5
Nama	Berjalan
Tujuan	Menggerakan karakter atau tokoh utama
Deskripsi	Proses untuk menggerakan karakter atau tokoh utama yang dimainkan dalam permainan menggunakan tombol

	keyboard.
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain menekan tombol pada *keyboard*

Aksi Aktor	Reaksi Sistem
Pemain menekan tombol W	
	Sistem menggerakan karakter ke depan
Pemain menekan tombol W dan A secara bersamaan	
	Sistem menggerakan karakter maju ke kiri
Pemain menekan tombol W dan D secara bersamaan	
	Sistem menggerakan karakter maju ke kanan
Pemain menekan tombol D	
	Sistem menggerakan karakter ke kanan
Pemain menekan tombol A	
	Sistem menggerakan karakter ke kiri
Pemain menekan tombol S	
	Sistem menggerakan karakter ke bawah
Pemain menekan tombol S dan D secara bersamaan	
	Sistem menggerakan karakter mundur ke kanan
Pemain menekan tombol S dan A secara bersamaan	
	Sistem menggerakan karakter mundur ke

	kiri
Kondisi Akhir	Sistem akan menampilkan karakter atau tokoh utama melakukan pergerakan berupa berjalan.

Tabel III.17 Skenario *Use Case* Meloncat

Identifikasi	
Nomor	6
Nama	Meloncat
Tujuan	Menggerakan karakter atau tokoh utama
Deskripsi	Proses untuk menggerakan karakter atau tokoh utama yang dimainkan dalam permainan menggunakan tombol *keyboard.*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain menekan tombol spasi pada *keyboard*
Aksi Aktor	Reaksi Sistem
Pemain menekan tombol spasi pada *keyboard*	
	Karakter atau tokoh utama dalam permainan akan meloncat sesuai dengan inputan dari pemain
Kondisi Akhir	Sistem akan menampilkan karakter atau tokoh utama melakukan loncatan

Tabel III.18 Skenario *Use Case* Menembak

Identifikasi	
Nomor	7
Nama	Menembak
Tujuan	Melakukan tembakan

Deskripsi	Proses untuk melakukan tembakan	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain menekan tombol kiri *mouse*	
Aksi Aktor		Reaksi Sistem
Pemain menekan tombol pada kiri *mouse*		
		Sistem mengeluarkan peluru dari senjata yang digunakan
Kondisi Akhir	Pemain menembak dengan menggunakan tombol kiri *mouse*	

Tabel III.19 Skenario *Use Case* Mengambil Darah

Identifikasi		
Nomor	8	
Nama	Mengambil Darah	
Tujuan	Menambah darah ketika darah tinggal tersisa sedikit lagi	
Deskripsi	Proses dimana pemain menambahkan darah ketika darah tersisa sedikit lagi	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain mengambil darah	
Aksi Aktor		Reaksi Sistem
Pemain mengambil darah		
		Sistem menampilkan *icon* darah
		Sistem melakukan pengecekan
		Apabila darah pemain penuh, sistem tidak mengalami perubahan apapun
		Apabila darah pemain tidak penuh, sistem menampilkan *bar* darah pemain

	yang bertambah
	Apabila darah pemain tidak penuh, sistem menghilangkan *icon* darah yang telah diambil oleh pemain
Kondisi Akhir	Sistem menampilkan *bar* darah yang bertambah atau tidak dan menampilkan *icon* darah yang menghilang atau tidak menghilang.

Tabel III.20 Skenario *Use Case* Mengganti Senjata

Identifikasi	
Nomor	9
Nama	Mengganti Senjata
Tujuan	Mengganti senjata dengan senjata laras panjang atau senjata roket
Deskripsi	Proses mengganti senjata dengan pilihan senjata laras panjang atau senjata roket
Aktor	Pemain
Skenario Utama	
Kondisi awal	Sistem menampilkan permainan
Aksi Aktor	Reaksi Sistem
	Sistem menampilkan permainan
Pemain memainkan permainan	
Pemain menggunakan senjata	
Pemain menekan tombol 1 jika ingin menggunakan senjata laras panjang	
	Sistem menampilkan karakter dengan senjata laras panjang
Pemain menekan tombol 2 jika ingin menggunakan senjata roket	
	Sistem menampilkan karakter dengan

	senjata roket
Kondisi Akhir	Sistem menampilkan senjata yang dipilih *player*

Tabel III.21 Skenario *Use Case* Mengambil Peluru Roket

Identifikasi	
Nomor	10
Nama	Mengambil Peluru Roket
Tujuan	Menambah peluru roket ketika peluru tinggal tersisa sedikit atau telah habis
Deskripsi	Proses dimana pemain menambahkan peluru roket ketika peluru roket tersisa sedikit atau habis
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain mengambil darah
Aksi Aktor	Reaksi Sistem
Pemain mengambil peluru roket	
	Sistem menampilkan *icon* peluru roket
	Sistem melakukan pengecekan
	Apabila peluru roket pemain penuh, sistem tidak mengalami perubahan apapun
	Apabila peluru roket pemain tidak penuh, sistem menampilkan bar peluru roket pemain yang bertambah
	Apabila peluru roket pemain tidak penuh, sistem menghilangkan *icon* peluru roket yang telah diambil oleh pemain
Kondisi Akhir	Sistem menampilkan *bar* peluru roket yang bertambah atau tidak dan menampilkan *icon* peluru roket yang menghilang atau tidak menghilang.

Tabel III.22 Skenario *Use Case* Memberhentikan Permainan

Identifikasi		
Nomor	11	
Nama	Memberhentikan Permainan	
Tujuan	Melihat petunjuk permainan dan informasi kontrol atau keluar dari permainan	
Deskripsi	Proses ketika pemain ingin melihat petunjuk permainan dan informasi kontrol atau keluar dari permainan	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Sistem menampilkan permainan	
Aksi Aktor		Reaksi Sistem
Pemain menekan tombol P		
		Sistem menampilkan *menu* jeda
		Sistem menampilkan petunjuk untuk keluar dari permainan dan kembali ke menu utama
Pemain menekan kiri *mouse* di sembarang arena permainan		
		Sistem kembali ke permainan
Pemain menekan tombol P		
		Sistem kembali ke menu utama
Kondisi Akhir	Sistem menampilkan menu jeda	

Tabel III.23 Skenario *Use Case* Mengambil Petunjuk

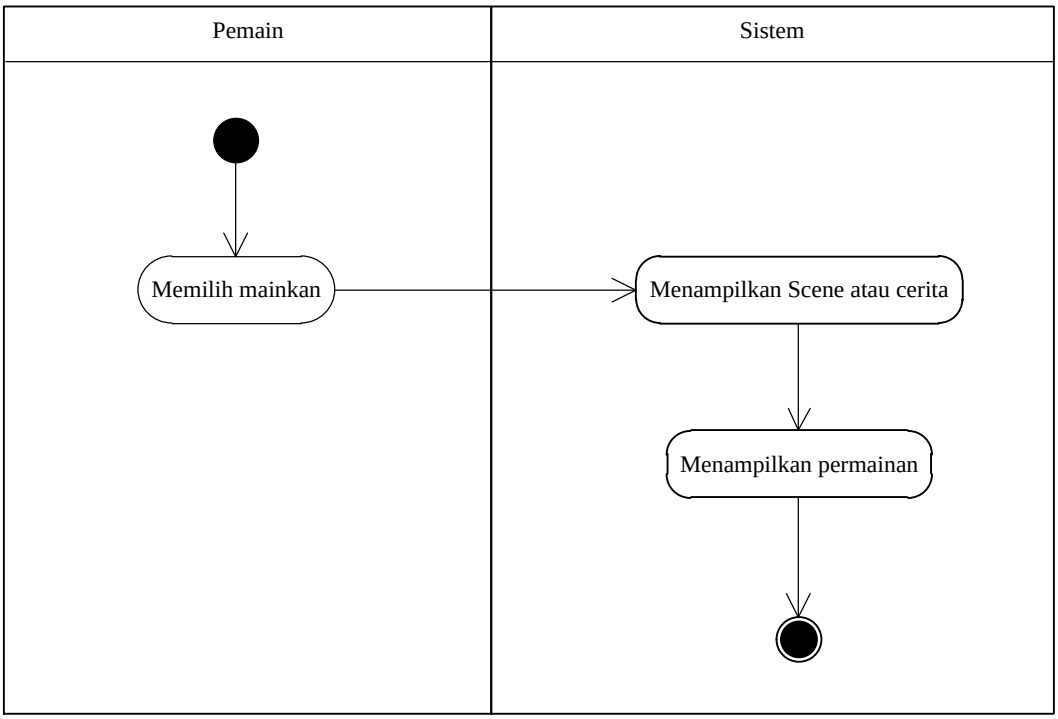
Identifikasi	
Nomor	12
Nama	Mengambil Petunjuk
Tujuan	Mengambil infromasi atau petunjuk permainan

Deskripsi	Proses ketika pemain mengambil informasi atau petunjuk permainan	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain mengambil petunnjuk	
Aksi Aktor		Reaksi Sistem
Pemain mengambil petunjuk		
		Sistem menampilkan icon petunjuk
		Sistem menampilkan informasi / petunjuk permainan
Kondisi Akhir	Sistem menampilkan informasi / petunjuk permainan	

III.4.2 Activity diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah action dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelum (*internal processing*).

A. *Activity Diagram* Proses Memulai Permainan

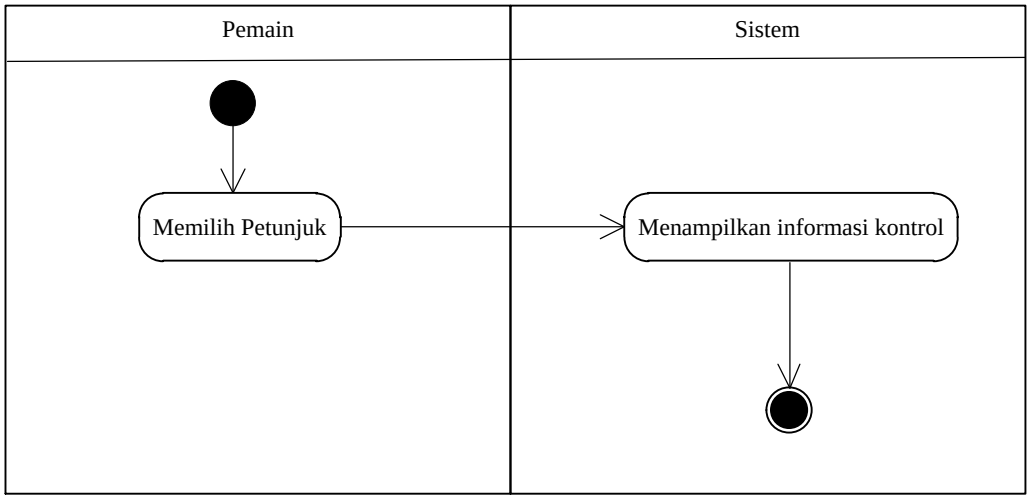


Gambar III.28 *Activity Diagram* Memulai Permainan

Tabel III.24 Penjelasan Proses Memulai Permainan

Deskripsi	Menu mainkan merupakan menu yang digunakan untuk mulai memainkan permainan.
Normal Flow	1. Pemain memilih menu mainkan di menu utama 2. Sistem menampilkan *scene* atau cerita sebelum masuk ke permainan 3. Sistem menampilkan permainan

B. *Activity Diagram* Proses Melihat Informasi Kontrol

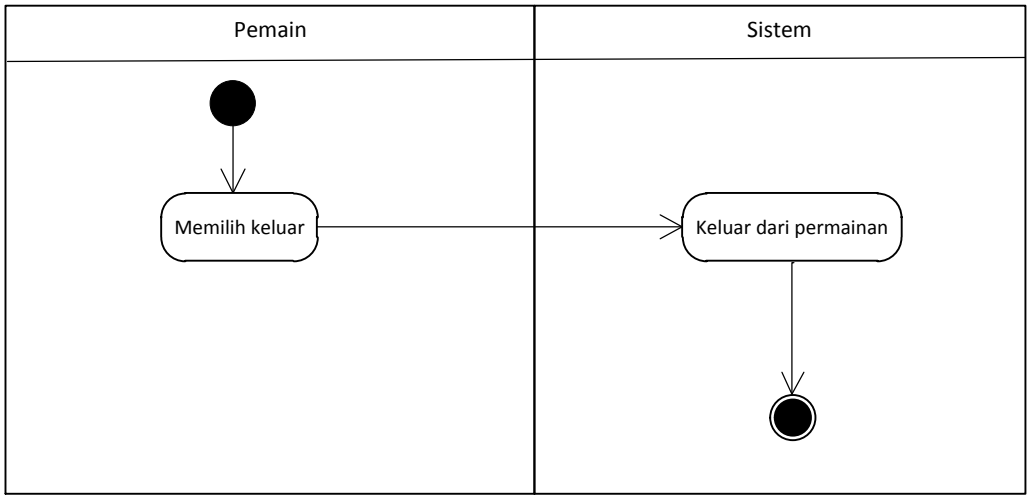


Gambar III.29 *Activity Diagram* Melihat Informasi Kontrol

Tabel III.25 Penjelasan Proses Melihat Informasi Kontrol

Deskripsi	Menu petunjuk digunakan pemain untuk melihat petunjuk permainan
Normal Flow	1. Pemain memilih Menu petunjuk di menu utama 2. Sistem menampilkan informasi kontrol

C. *Activity Diagram* Mengakhiri Permainan

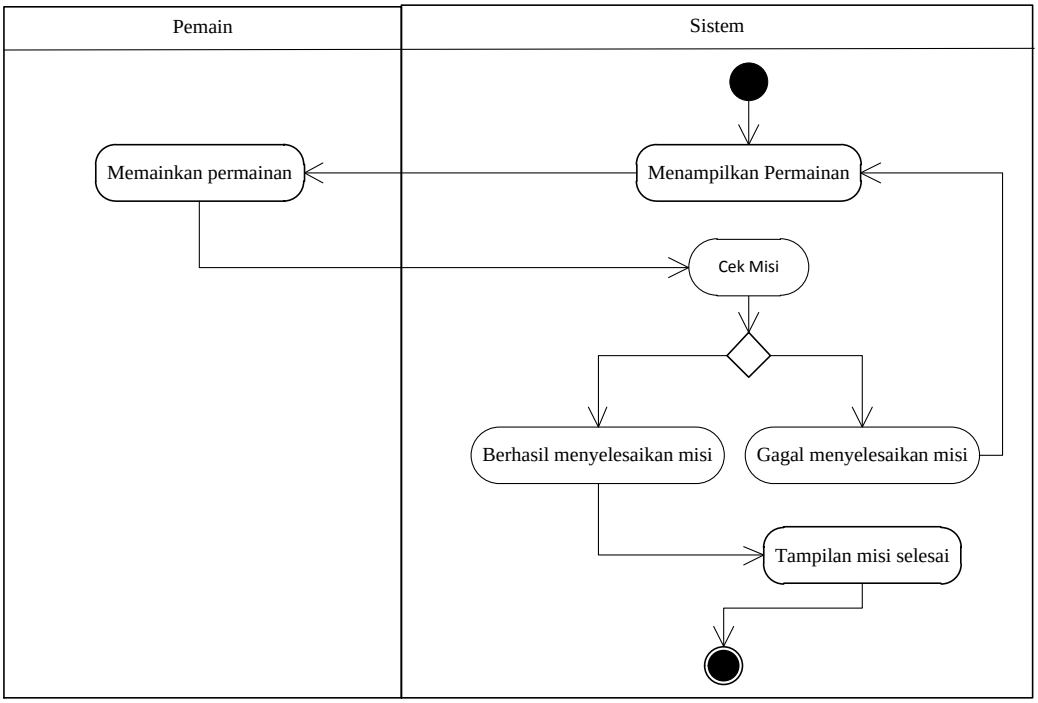


Gambar III.30 *Activity Diagram* Mengakhiri Permainan

Tabel III.26 Penjelasan Proses Mengakhiri Permainan

Deskripsi	Menu keluar merupakan menu yang digunakan untuk keluar dari permainan
Normal Flow	1. Pemain memilih menu keluar pada menu utama 2. Sistem akan mengakhiri permainan

D. Activity *Diagram* Permainan

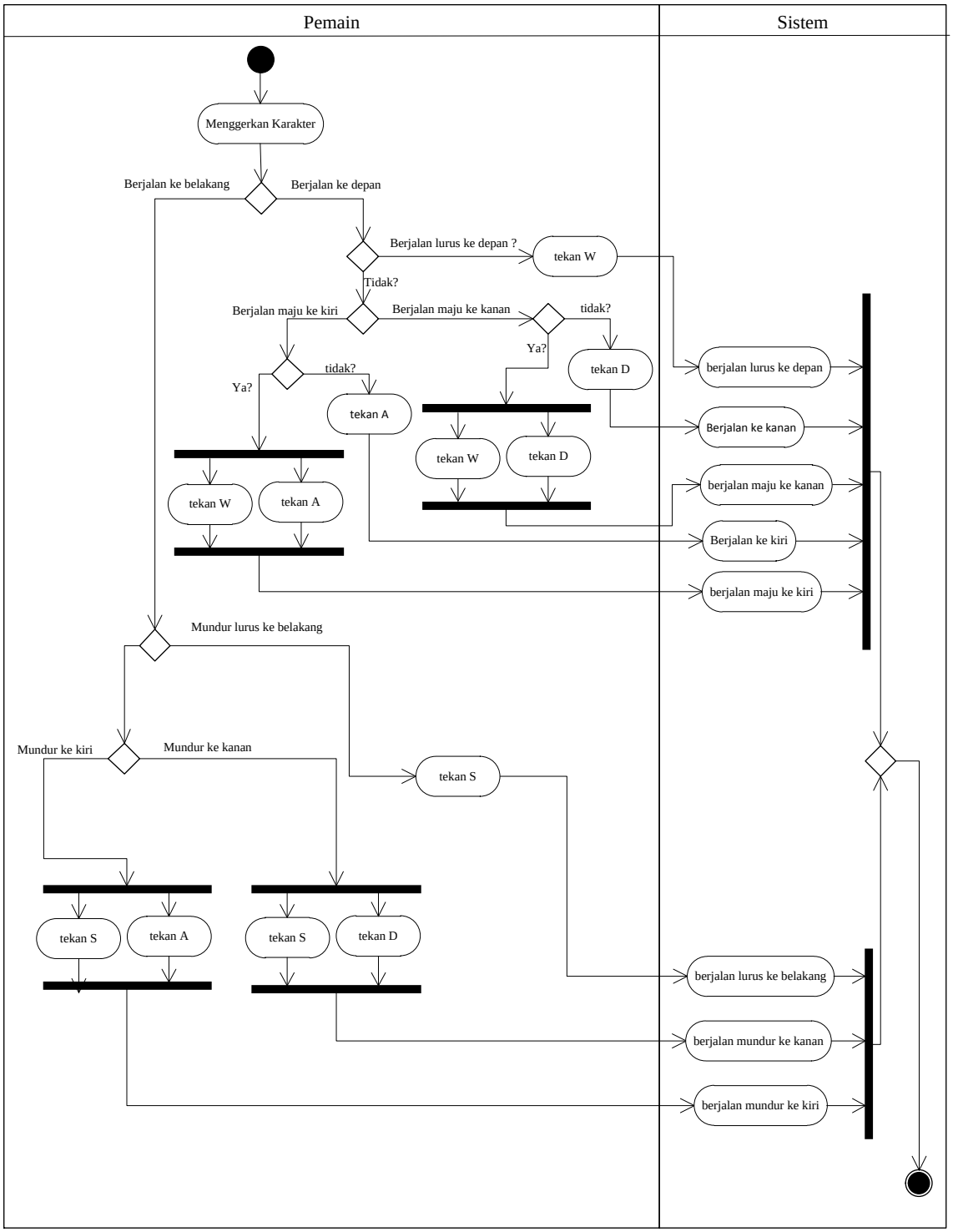


Gambar III.31 *Activity Diagram* Permainan

Tabel III.27 Penjelasan Proses Permainan

Deskripsi	Permainan merupakan kondisi pemain ketika telah berada di dalam permainan.
Normal Flow	1. Sistem menampilkan permainan 2. Pemain mulai memainkan permainan 3. Pemain berhasil menyelesaikan misi yang dihadapi 4. Sistem menampilkan tampilan misi selesai 5. Pemain gagal menyelesaikan misi 6. Sistem kembali ke awal permainan

E. *Activity Diagram* Berjalan

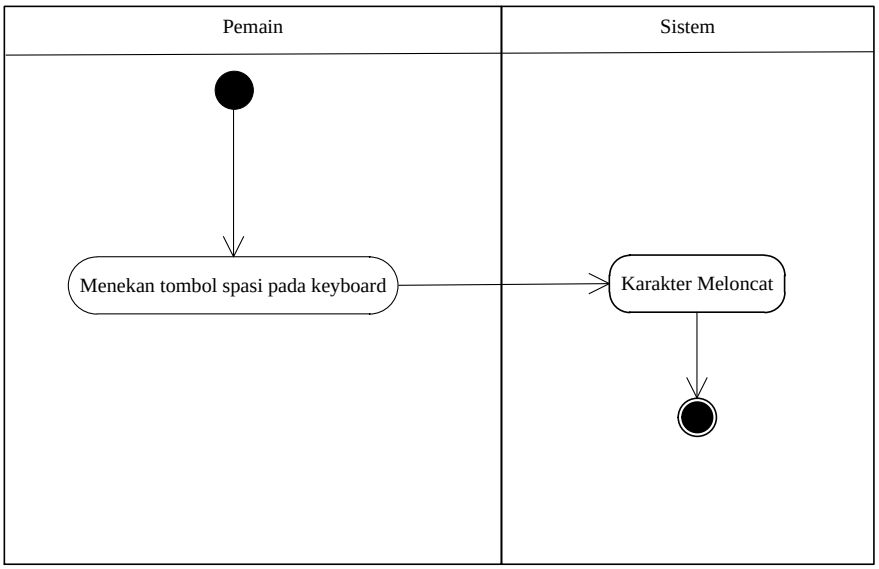


Gambar III.32 *Activity Diagram* Berjalan

Tabel III.28 Penjelasan Proses Berjalan

Deskripsi	Proses untuk menggerakan karakter atau tokoh utama yang dimainkan dalam permainan menggunakan tombol *keyboard*.
Normal Flow	1. Pemain menggerakan karakter 2. Pemain menekan tombol W 3. Karakter berjalan ke depan 4. Pemain menekan tombol W dengan A secara bersamaan 5. Karakter berjalan maju ke sebelah kiri 6. Pemain menekan tombol A 7. Karakter berjalan ke kiri 8. Pemain menekan tombol W dengan D secara bersamaan 9. Karakter berjalan maju ke kanan 10. Pemain menekan tombol D 11. Karakter berjalan ke kanan 12. Pemain menekan tombol S 13. Karakter berjalan lurus mundur 14. Pemain menekan tombol S dengan D secara bersamaan 15. Karakter berjalan mundur ke kanan 16. Pemain menekan tombol S dengan A secara bersamaan 17. Karakter berjalan mundur ke kiri

F. *Activity Diagram* Meloncat

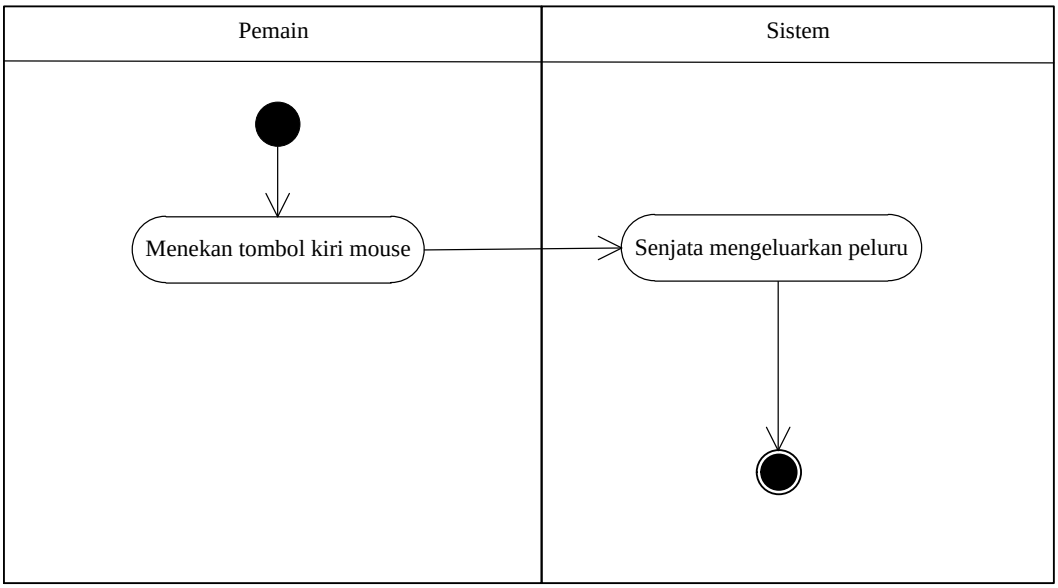


Gambar III.29 *Activity Diagram* Meloncat

Tabel III.29 Penjelasan Proses Meloncat

Deskripsi	Proses untuk menggerakan karakter atau tokoh utama yang dimainkan dalam permainan menggunakan tombol *keyboard.*
Normal Flow	1. Pemain menekan tombol spasi pada *keyboard* 2. Sistem menampilkan karakter yang meloncat sesuai inputan pemain

G. *Activity Diagram* Menembak

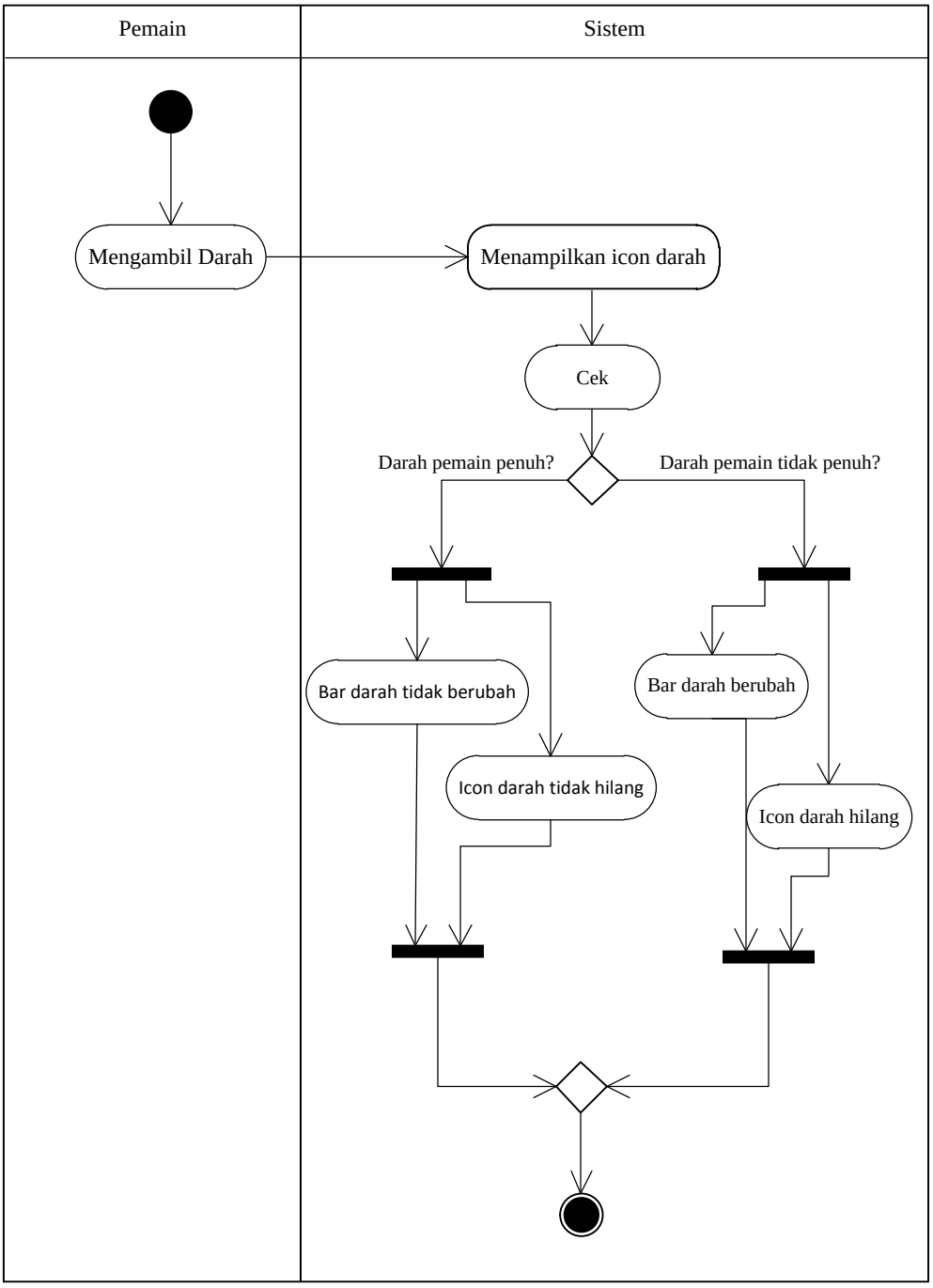


Gambar III.34 *Activity Diagram* Menembak

Tabel III.30 Penjelasan Proses Melakukan Tembakan

Deskripsi	Proses untuk melakukan tembakan
Normal Flow	1. Pemain menekan tombol kiri pada *mouse* 2. Sistem menampilkan senjata yang mengeluarkan peluru

H. *Activity Diagram* Mengambil Darah

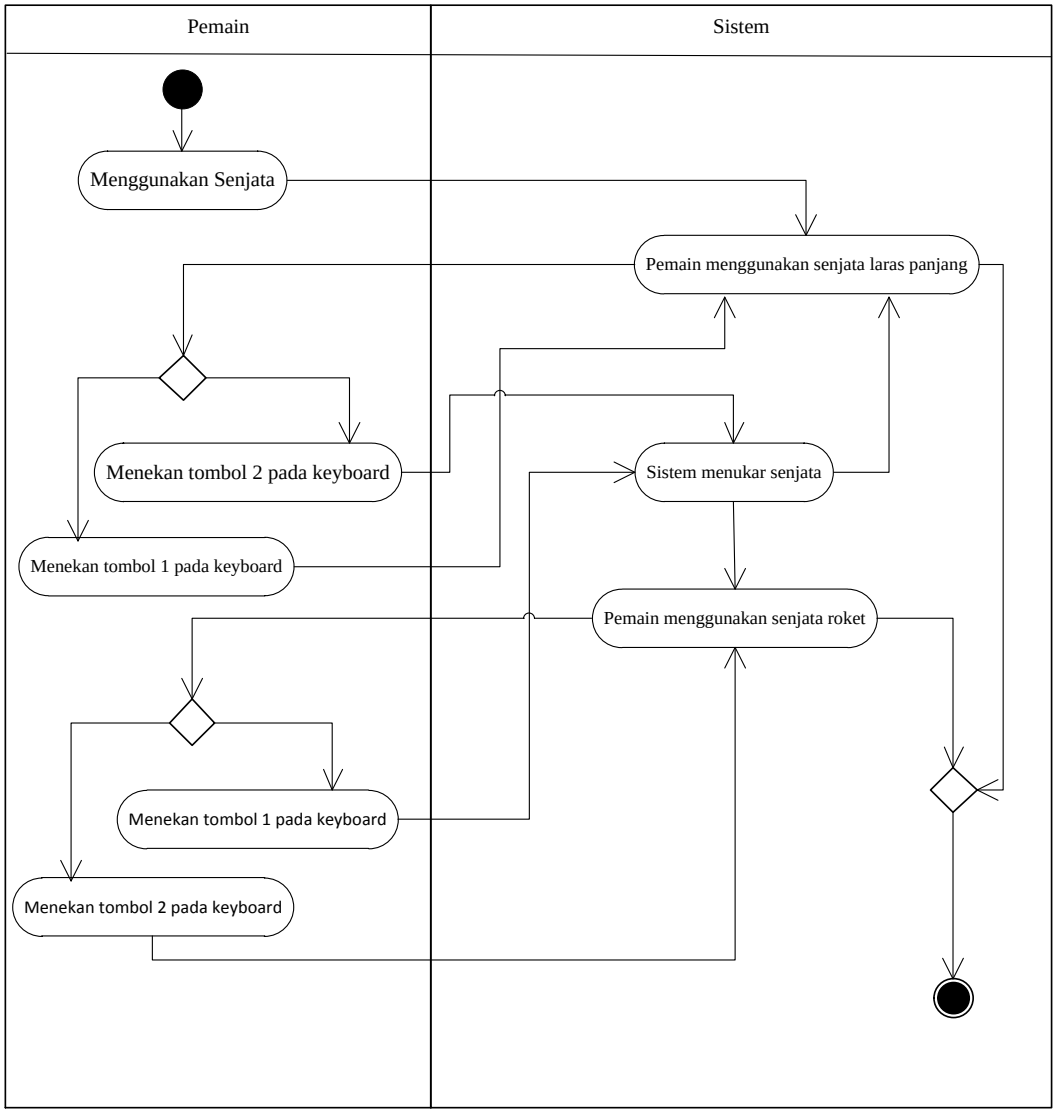


Gambar III.35 *Activity Diagram* Mengambil Darah

Tabel III.31 Penjelasan Proses Mengambil Darah

Deskripsi	Proses dimana pemain mengambil darah
Normal Flow	1. Pemain mengambil darah 2. Sistem menampilkan *icon* darah 3. Sistem melakukan pengecekan 4. Apabila darah pemain penuh, sistem menampilkan *bar* darah yang tidak berubah 5. Apabila darah pemain penuh, sistem menampilkan *icon* darah yang tidak menghilang 6. Apabila darah pemain tidak penuh, sistem menampilkan *bar* darah pemain yang bertambah 7. Apabila darah pemain tidak penuh, sistem menghilangkan *icon* darah yang telah diambil oleh pemain

I. *Activity Diagram* Mengganti Senjata



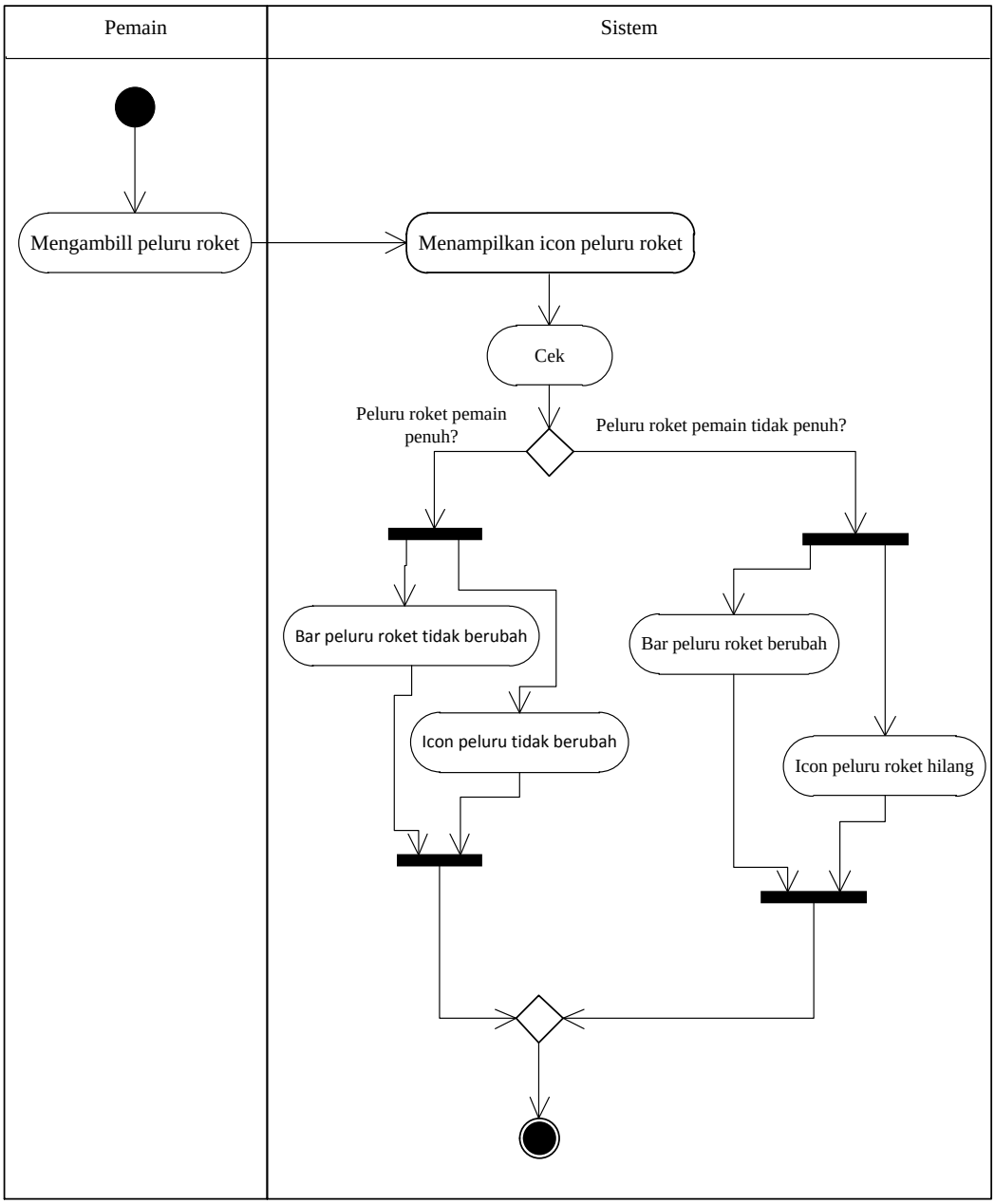
Gambar III.36 *Activity Diagram* Mengganti Senjata

Tabel III.32 Penjelasan Proses Mengganti Senjata

Deskripsi	Proses mengganti senjata dengan pilihan senjata laras panjang atau senjata roket
Normal Flow	1. Pemain menggunakan senjata 2. Sistem menampilkan pemain menggunakan senjata laras panjang 3. Pemain menekan tombol 1 pada *keyboard*

	4. Sistem tidak mengganti senjata 5. Pemain menekan tombol 2 pada *keyboard* 6. Sistem menukar senjata 7. Sistem menampilkan pemain menggunakan senjata roket 8. Pemain menekan tombol 2 pada *keyboard* 9. Sistem tidak mengganti senjata 10. Pemain menekan tombol 1 pada *keyboard* 11. Sistem menukar senjata 12. Sistem menampilkan pemain menggunakan sejata laras panjang

J. *Activity Diagram* Mengambil Peluru Roket

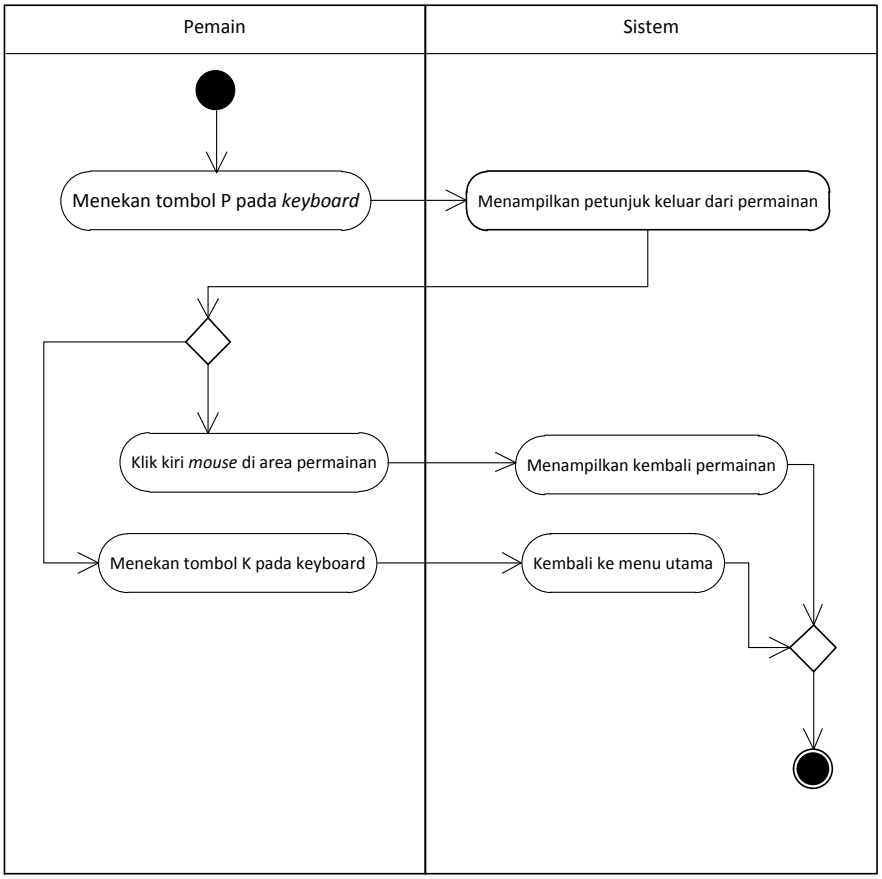


Gambar III.37 *Activity Diagram* Mengambil Peluru Roket

Tabel III.33 Penjelasan Proses Mengambil Peluru Roket

Deskripsi	Proses dimana pemain mengambil darah
Normal *Flow*	1. Pemain mengambil peluru roket 2. Sistem menampilkan *icon* peluru roket 3. Sistem melakukan pengecekan 4. Apabila peluru roket pemain penuh, *bar* peluru tidak bertambah 5. Apabila peluru roket pemain penuh, *icon* peluru roket tidak menghilang 6. Apabila peluru roket pemain tidak penuh, sistem menampilkan bar peluru roket pemain yang bertambah 7. Apabila peluru roket pemain tidak penuh, sistem menghilangkan icon peluru roket yang telah diambil oleh pemain

K. *Activity Diagram* Memberhentikan Permainan

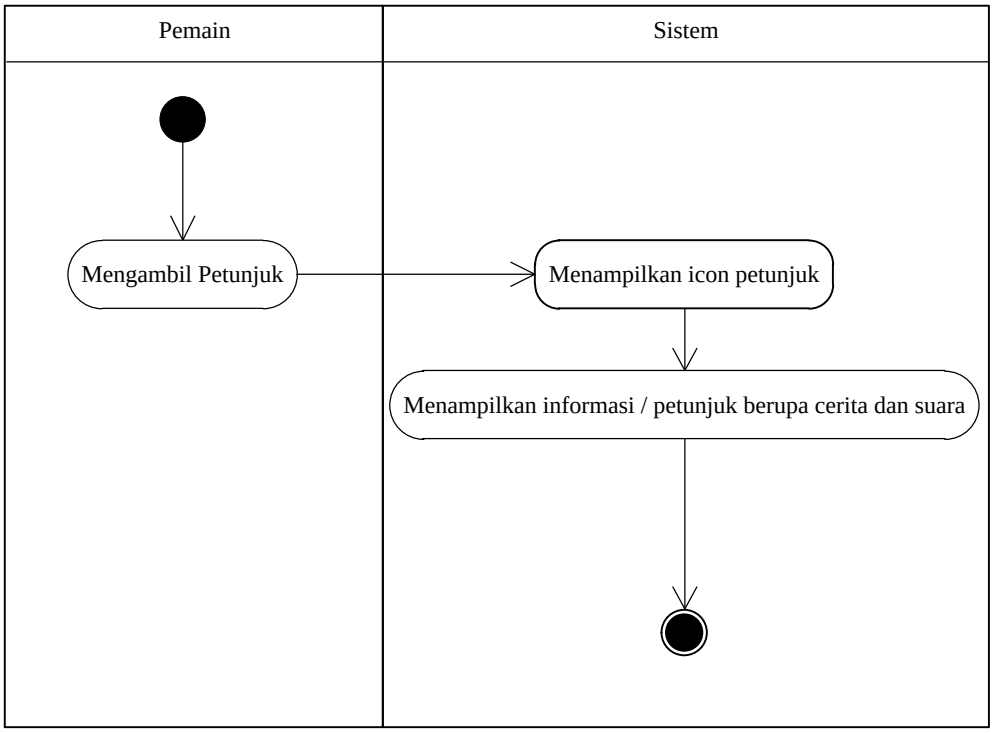


Gambar III.38 *Activity Diagram* Memberhentikan Permainan

Tabel III.34 Penjelasan Proses Memberhentikan Permainan

Deskripsi	Proses ketika pemain ingin melihat informasi kontrol atau keluar dari permainan
Normal Flow	1. Pemain menekan tombol P 2. Sistem menampilkan menu jeda berisi petunjuk untuk keluar dari permainan 3. Pemain menekan kiri *mouse* pada arena permainan 4. Sistem akan kembali menampilkan permainan 5. Pemain menenkan tombol K pada *keyboard* 6. Sistem akan keluar permainan dan kembali ke menu utama

L. *Activity Diagram* Mengambil Petunjuk



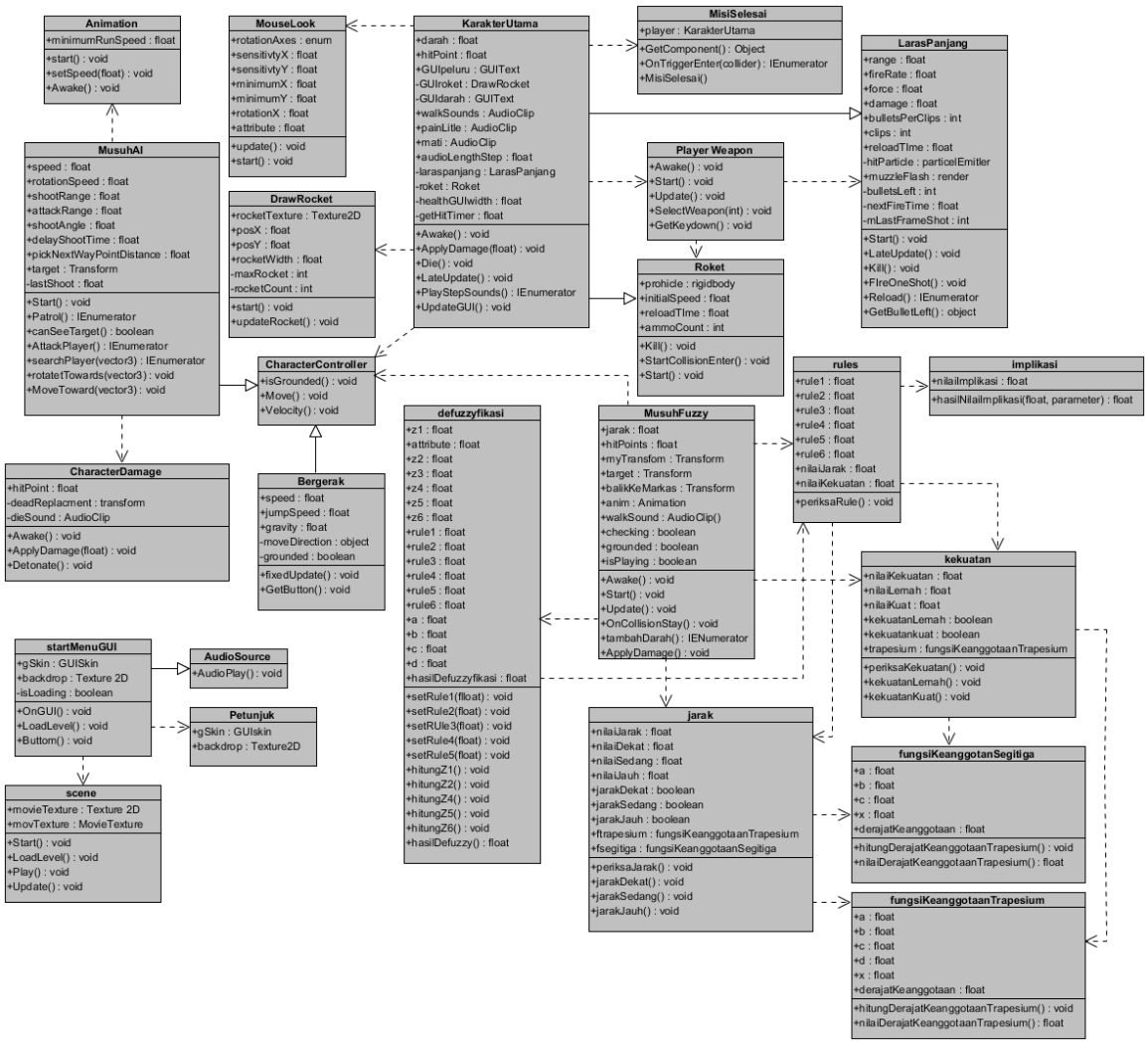
Gambar III.35 *Activity Diagram* Mengambil Petunjuk

Tabel III.35 Penjelasan Proses Mengambil Petunjuk

Deskripsi	Proses dimana pemain mengambil darah
Normal Flow	1. Pemain mengambil petunjuk 2. Sistem menampilkan *icon* petunjuk 3. Sistem menampilkan informasi / petunjuk berupa cerita dan suara

III.4.3 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian *class-class* yang dibuat untuk membangun sistem. Berikut adalah class diagram pada *game* Sejarah Pertempuran 10 November 1945 :



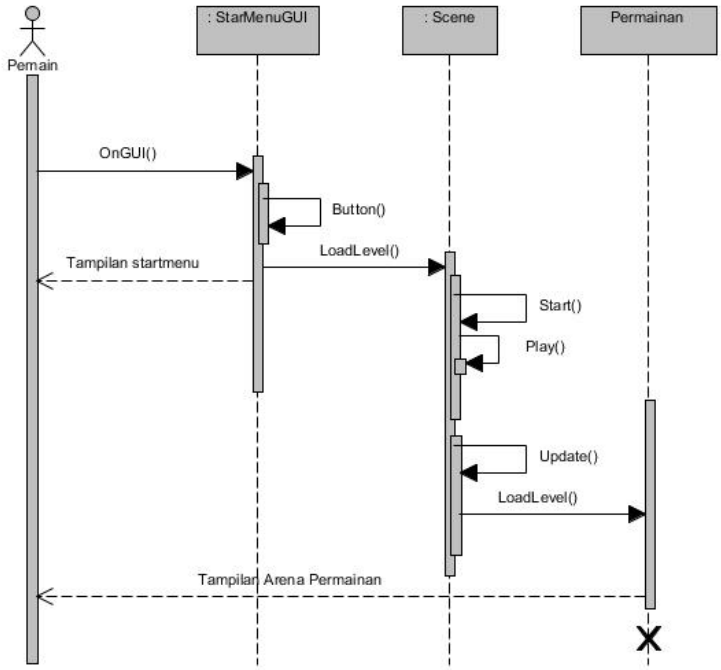
Gambar III.40 *Class Diagram* Pertempuran 10 November 1945

III.4.4 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

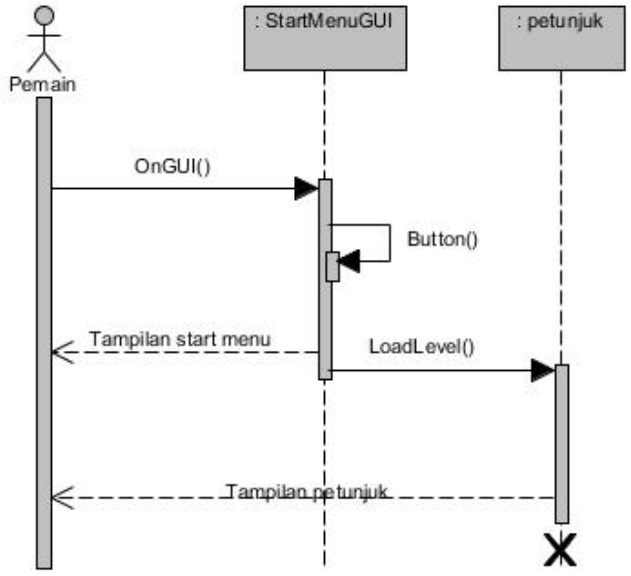
Sequence diagram biasa digunakan untuk menggambarakan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

A. *Sequence Diagram* Memulai Permainan



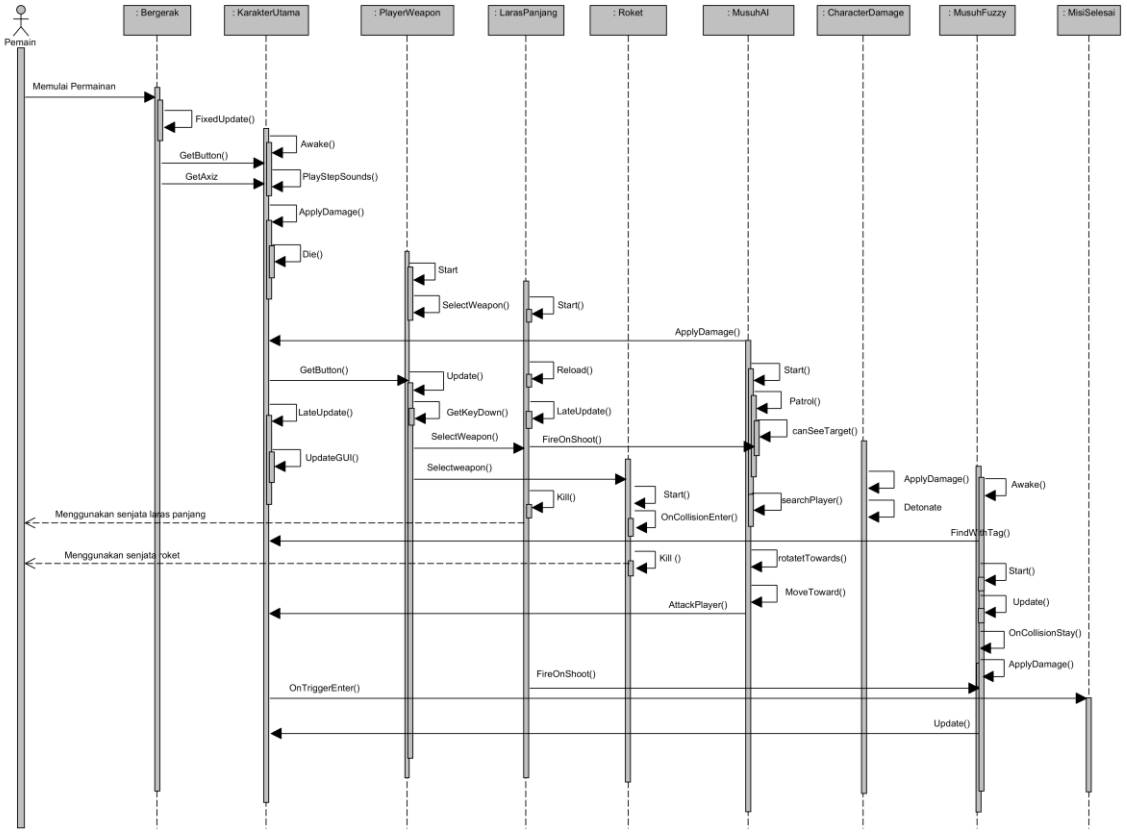
Gambar III.41 *Sequence Diagram* Memulai Permainan

B. *Sequence Diagram* Melihat Informasi Kontrol



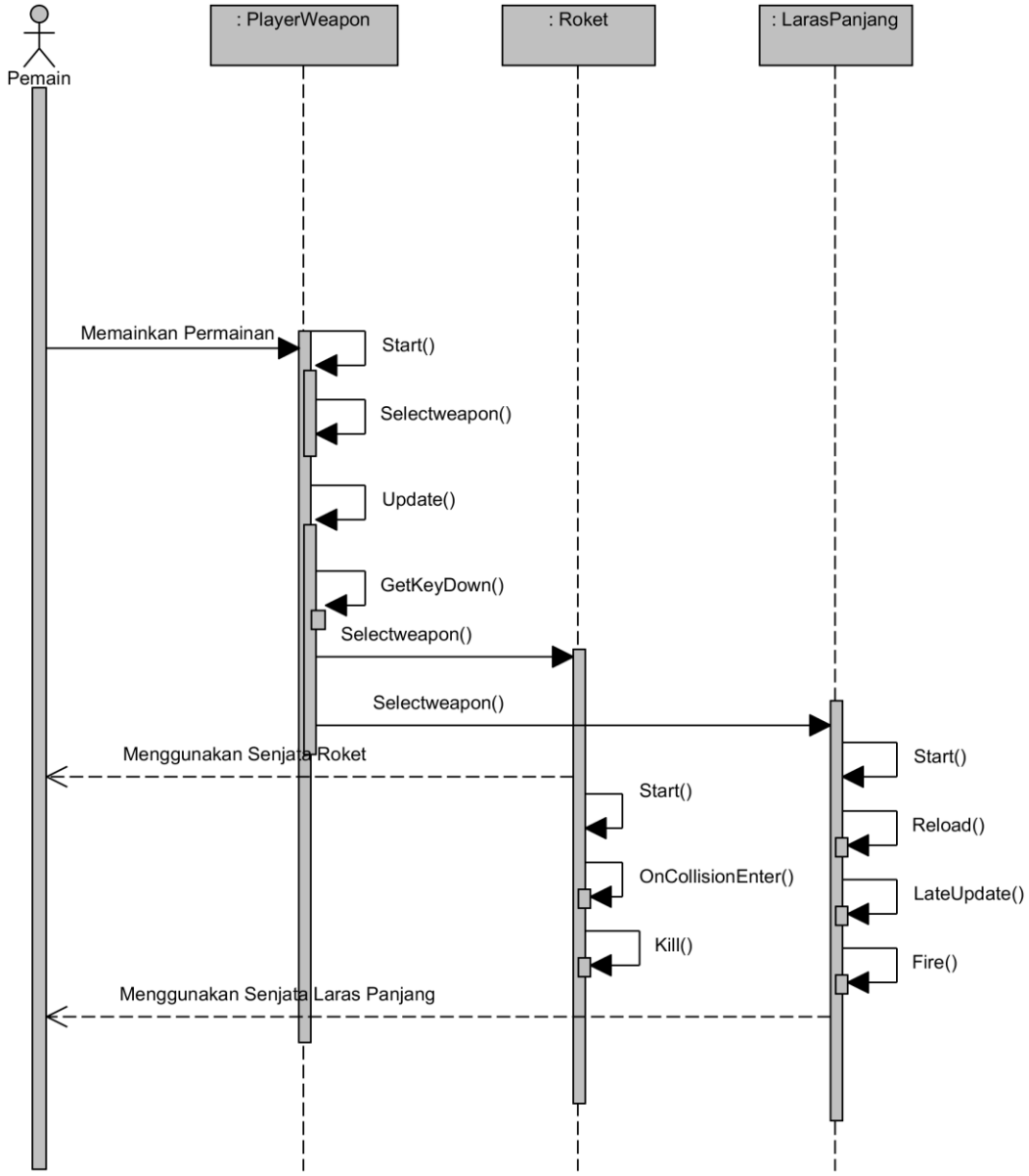
Gambar III.42 *Sequence Diagram* Melihat Informasi Kontrol

C. *Sequence Diagram* Permainan



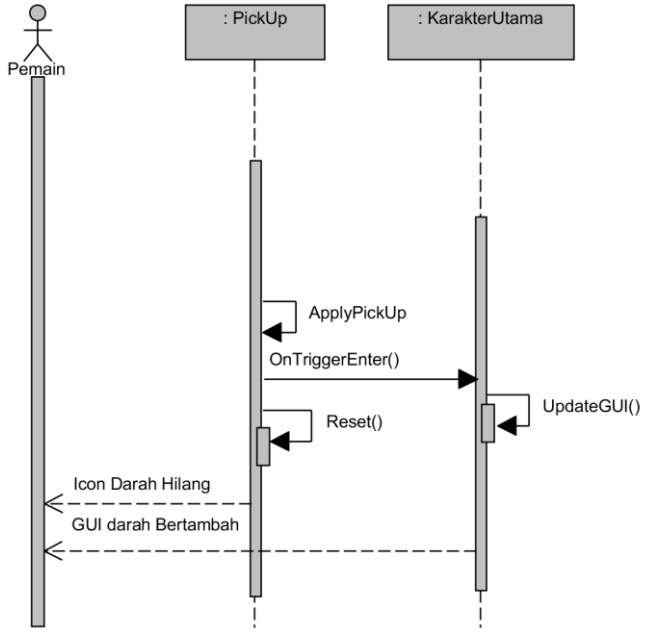
Gambar III.43 *Sequence Diagram* Permainan

D. *Sequence Diagram* Mengganti Senjata



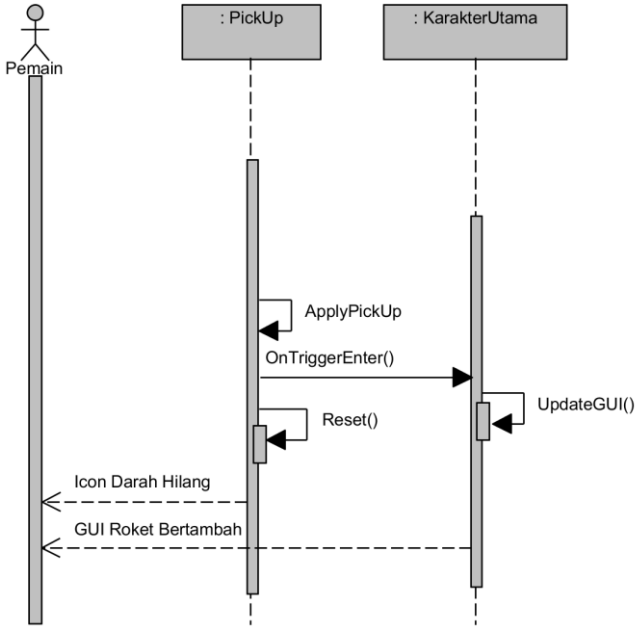
Gambar III.44 *Sequence Diagram* Mengganti Senjata

E. *Sequence Diagram* Mengambil Darah



Gambar III.45 *Sequence Diagram* Mengambil Darah

F. *Sequence Diagram* Mengambil Peluru Roket



Gambar III.46 *Sequence Diagram* Mengambil Peluru Roket

III.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.

III.5.1 Perancangan Komponen Permainan

III.5.1.1 Karakter

Karakter utama dalam *game* ini bernama Kuesno Wibowo seorang tentara atau pejuang yang nantinya akan melawan para tentara Inggria. Kuesno dilengkapi dua senjata, Pertama senjata laras panjang dan kedua senjata roket. Karena *game* ini merupakan *game* yang mempunyai *genre first person shooter* yang mempunyai sudut pandang orang pertama, maka karakter jatmiko seperti tubuh, kepala, kaki dan tangan tidak diperlihatkan tetapi hanya memperlihatkan bagian moncong senjatanya saja.

Tabel III.36 Properties Karakter Utama

Nama	Kuesno Wibowo
Darah	200
Kemampuan	Berlari, melompat, menembak
Senjata	Laras panjang dan Roket

A. Senjata

Senjata yang digunakan oleh Kuesno Wibowo ada dua macam, pertama adalah senjata laras panjang yang memiliki peluru tidak terbatas dan yang kedua adalah roket yang mempunyai peluru terbatas sebanyak 40.



Gambar III.47 Senjata Laras Panjang



Gambar III.48 Senjata Roket

Tabel III.37 Perbandingan properties senjata

Properti	Laras Panjang	Roket
Damage	5	30
Jarak	50	100
Reload	0.5	1
Peluru	Unlimited	20

B. Musuh

Karakter Musuh yang terdapat pada game ini hanya ada satu, yaitu tentara Inggris, akan tetapi karakter musuh ini memiliki 3 senjata yang berbeda, pertama tentara Inggris yang menggunakan senjata Laras Pnajang, kedua tentara Inggris yang menggunakan senjata pistol dan ketiga tentara Inggris yang menggunakan senjata Roket. Karakter musuh ini memiliki tingkat kesulitan yang berbeda pada setiap level yang dimainkan. Karakter musuh

lainnya adalah sebuah senapan otomatis yang akan menembak apabila target sudah berada pada jarak yang sesuai.



Gambar III.49 Karakter Musuh Tentara Inggris senjata laras panjang



Gambar III.50 Karakter Musuh Tentara Inggris senjata Pistol



Gambar III.51 Karakter Musuh Tentara Inggris senjata Roket



Gambar III.52 Senapan Otomatis

Tabel III.38 Perbandingan propertis Musuh

Propertis	**Tentara Inggris (Laras Panjang)**	**Tentara Inggris (Pistol)**	**Tentara Inggris (Roket)**	**Senapan Otomatis**
Kecepatan	3	3	3	-
Jarak Menembak	15	10	15	25
Jeda Menembak	0.35	0.35	0.5	0.1
Damage	7	5	20	5
Darah	100			100

C. Teman

Karakter teman pada game ini terdapat satu yang bernama Jatmiko, karakter teman ini akan secara otomatis mengikuti ketika game dimainkan pada level 2.



Gambar III.53 Karakter Teman

Tabel III.39 Perbandingan propertis Teman

Propertis	Karakter Teman
Kecepatan	5
Jarak Menembak	-
Jeda Menembak	-
Damage	-
Darah	200

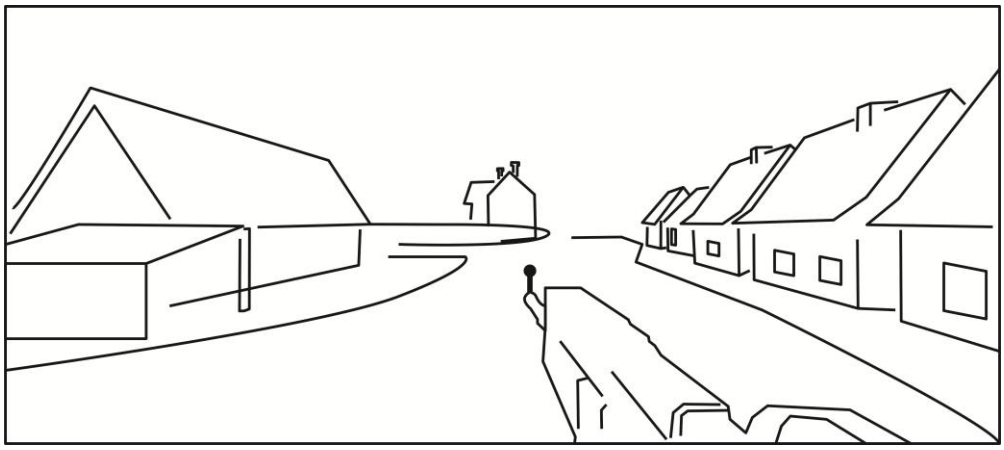
III.5.1.2 Storyboard

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan menggunakan *storyboard* dapat menyampaikan ide cerita kepada orang lain dengan lebih mudah karena dapat dengan *storyboard* dapat mengimajinasikan khayalan yang sesuai dengan gambar-gambar yang ada sehingga dapat menghasilkan persepsi yang sama pada ide cerita yang dibuat.

Berikut beberapa *storyboard* dari *game first person shooter* 3D sejarah pertempuran 10 november 1945 surabaya :

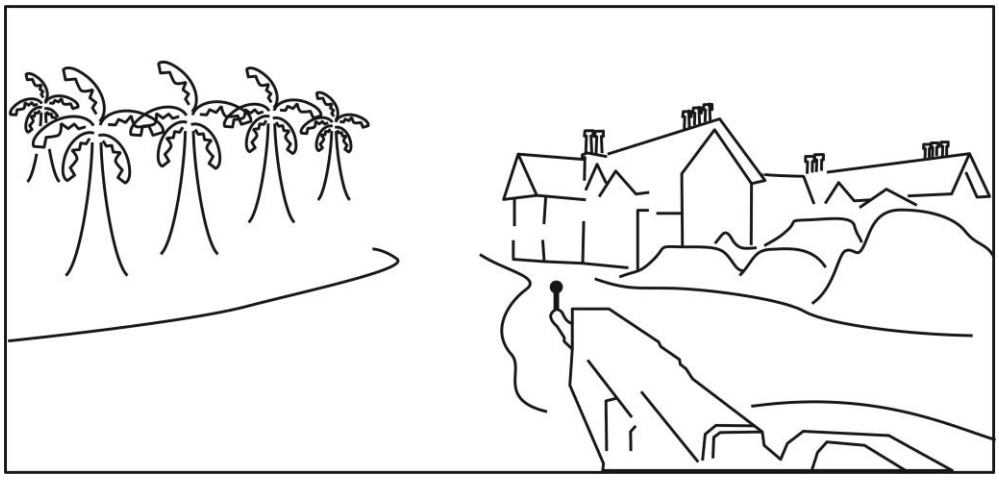
1. Tampilan *Storyboard* di level 1

Storyboard pada gambar III.54 Menceritakan pemain / *player* ketika memulai permainan. Terdapat beberapa rumah dan sebuah jalan dengan beberapa persimpangan yang harus di lewati oleh pemain.



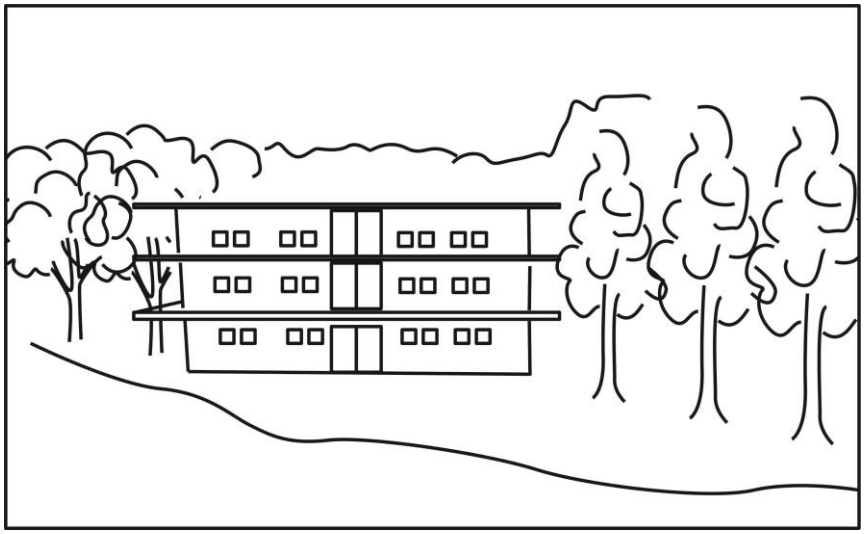
Gambar III.54 *Storyboard level* 1-1

Storyboard pada gambar III.54 terdapat beberapa rumah dan pohon serta sebuah jalan lurus yang nantinya akan mengarahkan pemain / *player* menuju hotel Yamato.



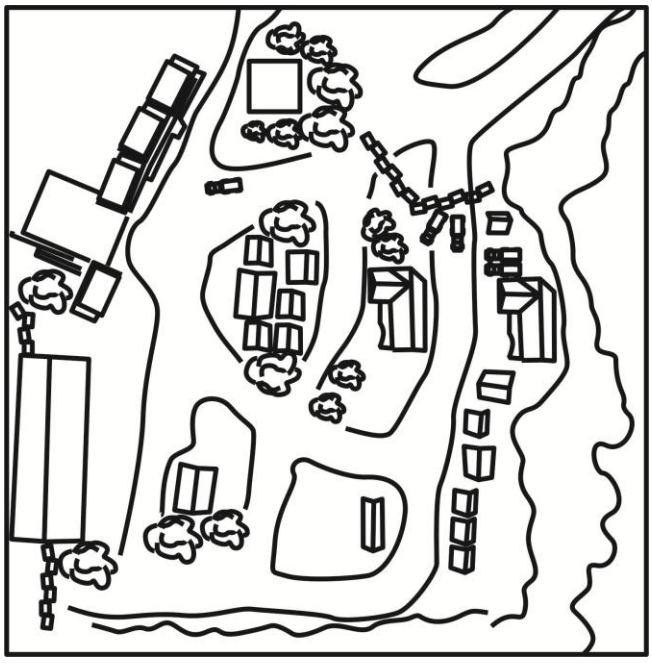
Gambar III.55 *Storyboard level* 1-2

Storyboard pada gambar III.56 menceritakan pemain yang telah berhasil menemukan hotel Yamato. Hotel Yamato sendiri merupakan tujuan dari pemain memainkan level pertama, pemain akan menyelesaikan level pertama ketika masuk ke dalam hotel tersebut.



Gambar III.56 *Storyboard level* 1-3

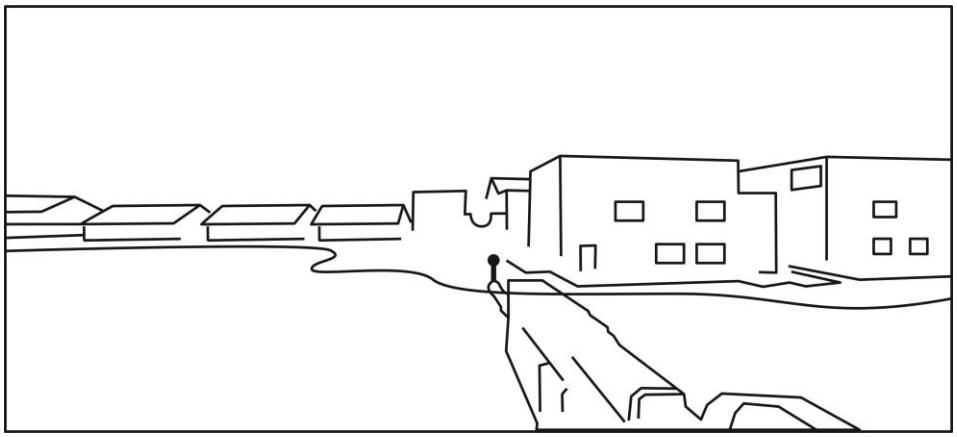
Storyboard pada gambar III.57 Memperlihatkan *mapping* secara keseluruhan permainan pada level pertama, pada level ini terdapat bermacam-macam bangunan, pepohonan, tank, truk dan jalan yang disertai dengan persimpangan-persimpangan serta gundukan-gundukan tanah yang menyerupai bukit.



Gambar III.57 *Storyboard level* 1-4

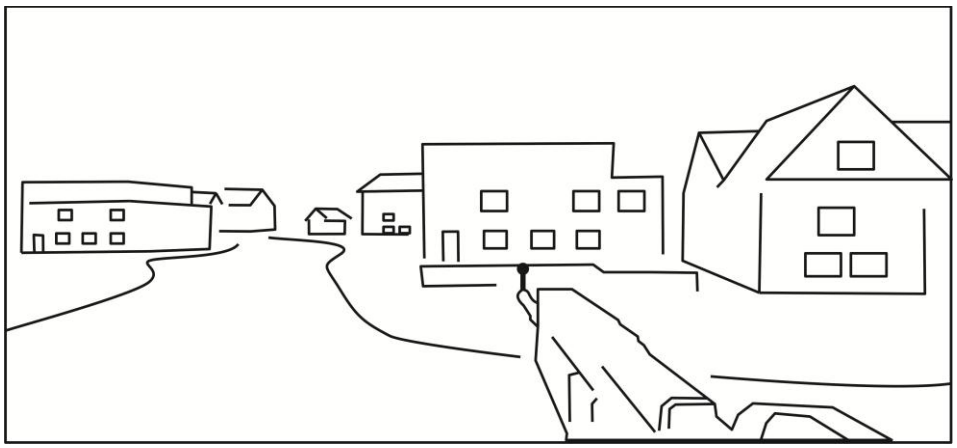
2. Tampilan *storyboard* di level 2

Storyboard pada gambar III.58 memperlihatkan tampilan pertama ketika pemain masuk ke level 2. Pada level ini masih terdapat beberapa bangunan-bangunan, seperti rumah, gedung kosong dan gudang.



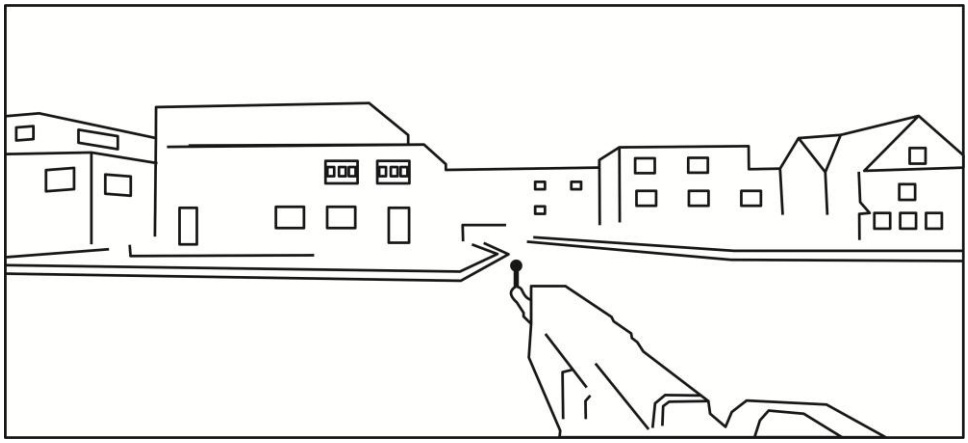
Gambar III.58 *Storyboard level* 2-1

Storyboard pada gambar III.59 menampilkan sebuah persimpangan jalan yang diapit oleh beberapa rumah.



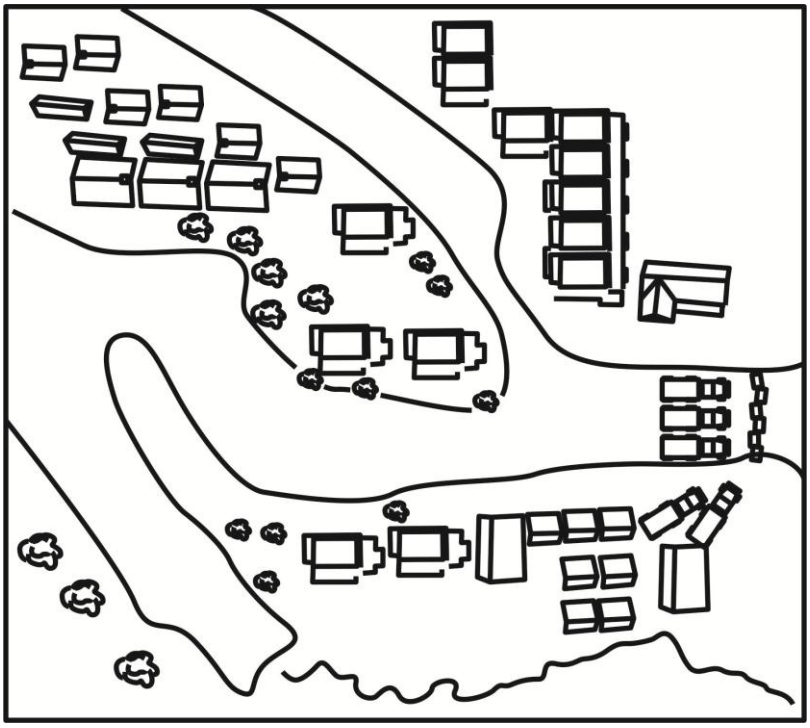
Gambar III.59 *Storyboard level* 2-2

Storyboard pada gambar III.60 Memperlihatkan pemain yang sedang berjalan untuk melakukan perlawanan terhadap tentara Inggris. Pada level ini pemain harus bertahan dari seranga tentara inggris serta berupaya untuk menyelesaikan misi berupa mengalahkan semua musuh yang terdapat di level kedua dan menyelamatkan seorang teman.



Gambar III.60 *Storyboard level* 2-3

Storyboard pada gambar III.61 Memperlihatkan *mapping* secara keseluruhan permainan pada level kedua, pada level ini terdapat bermacam-macam bangunan, pepohonan, tank, truk dan jalan yang disertai dengan persimpangan-persimpangan serta gundukan-gundukan tanah yang menyerupai bukit.

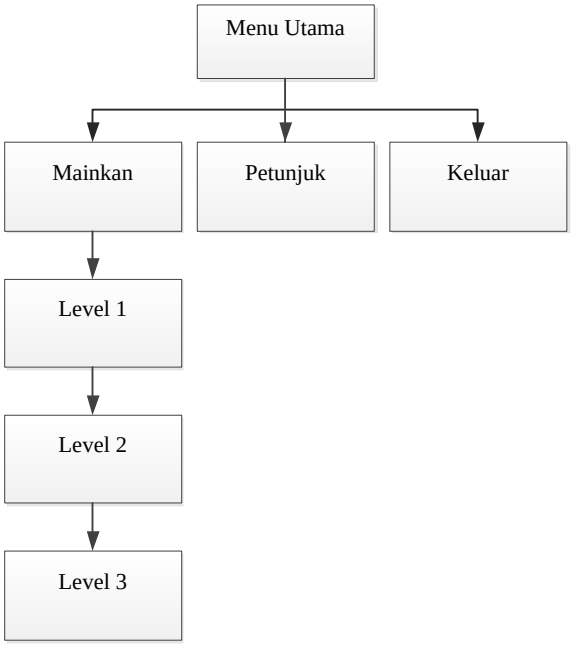


Gambar III.61 *Storyboard level* 2-4

III.5.2 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.

III.5.2.1 Perancangan Struktur Menu



Gambar III.62 Struktur menu utama

III.5.3 Perancangan Antarmuka

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.

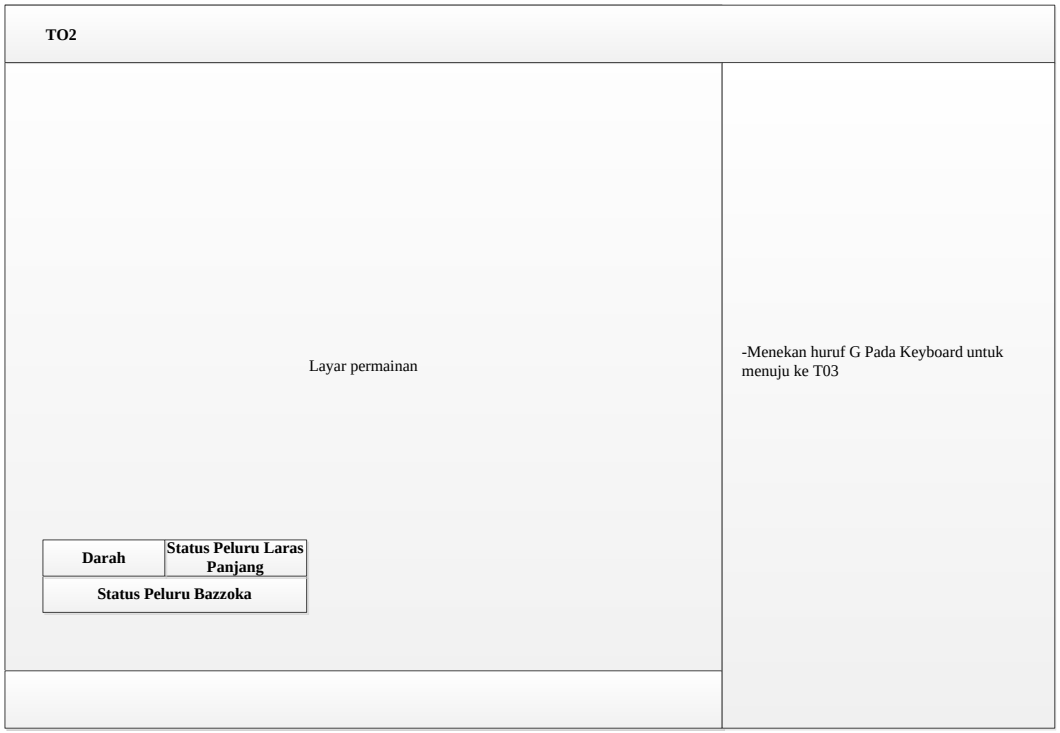
1. Menu Utama

Berikut dapat dilihat perancangan antarmuka menu utama dari *game* pertempuran 10 November 1945.



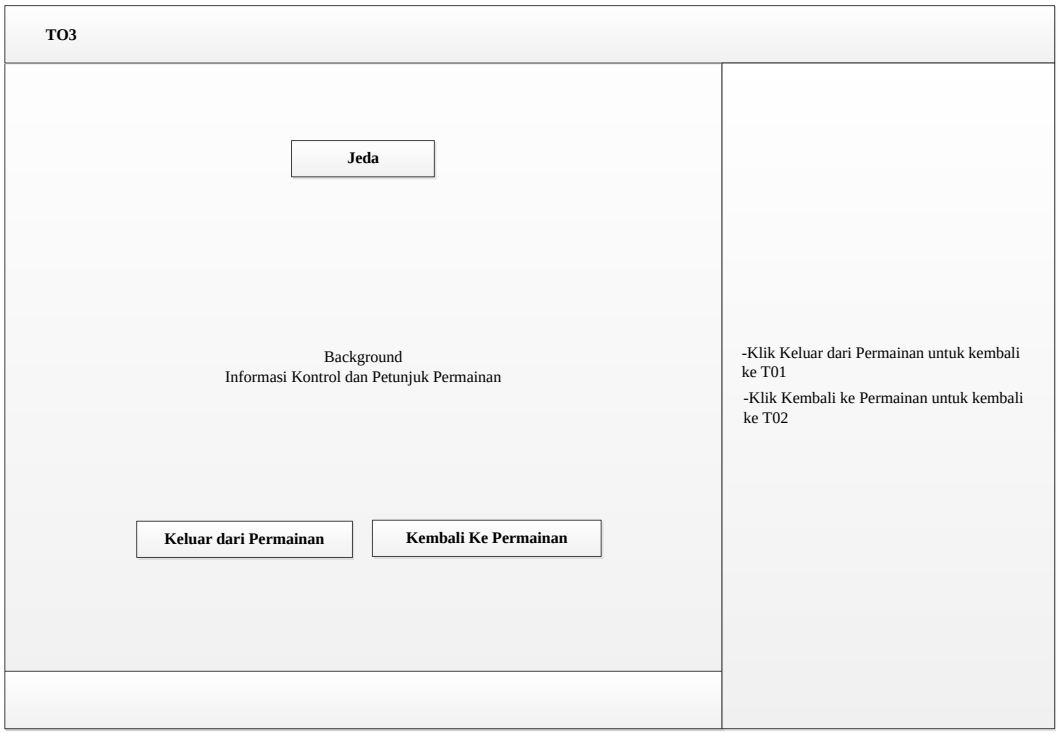
Gambar III.63 Tampilan Menu Utama

2. Permainan



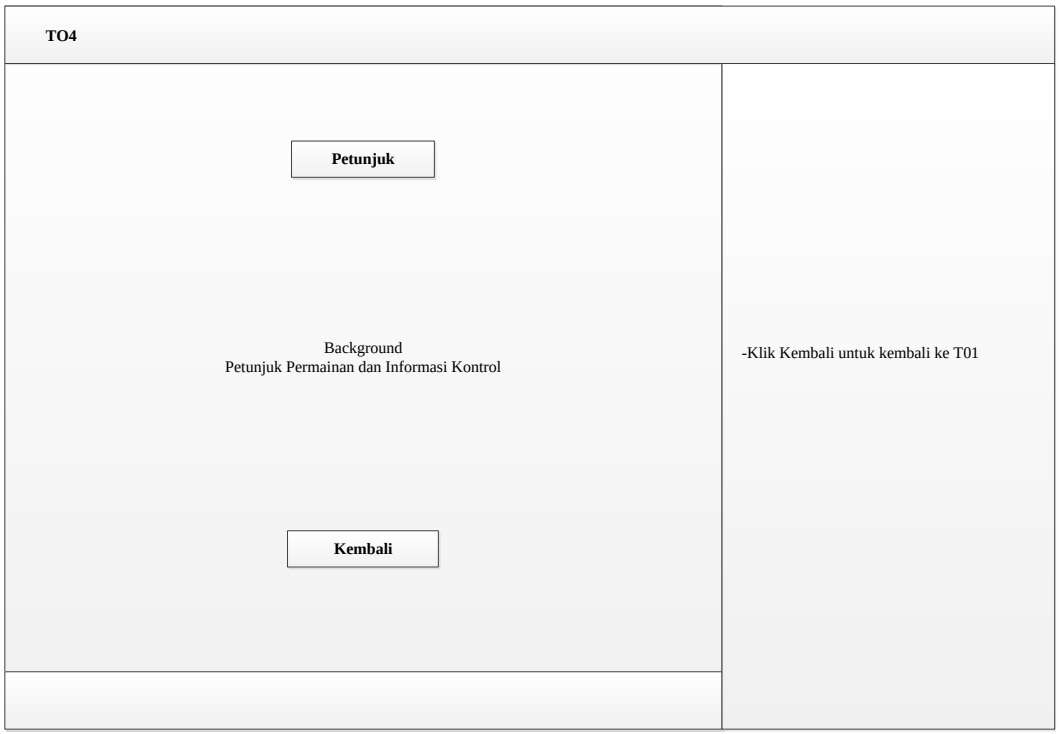
Gambar III.64 Tampilan Permainan

3. Jeda



Gambar III.65 Tampilan Jeda

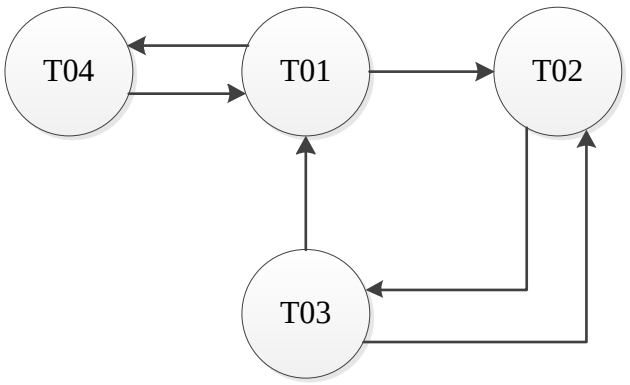
3. Petunjuk



Gambar III.66 Tampilan Petunjuk

III.5.3.1 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 struktural yang disebut juga jaringan semantik.



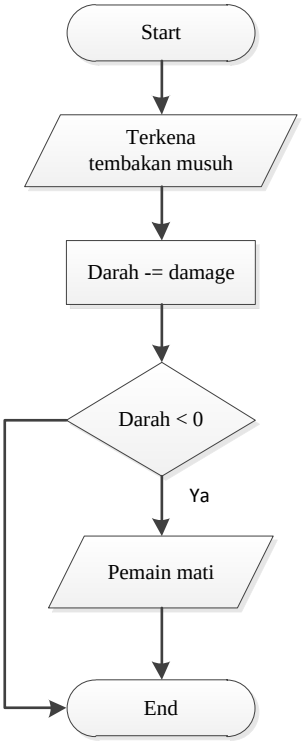
Gambar III.67 Jaringan Semantik

III.5.4 Perancangan Method

Perancangan *method* merupakan perancangan yang berfungsi untuk mendeskripsikan *method-method* yang berada di dalam aplikasi. *Method* dapat dipanggil dengan menyertakan variabel, baik hanya satu variabel, banyak variabel atau bahkan tidak ada sama sekali. Adapun *method-method* yang terdapat dalam *game* FPS Sejarah Pertempuran 10 November 1945 yang akan dibangun dijelaskan melalui *flowchart* sebagai berikut.

1. *Method ApplyDamage*

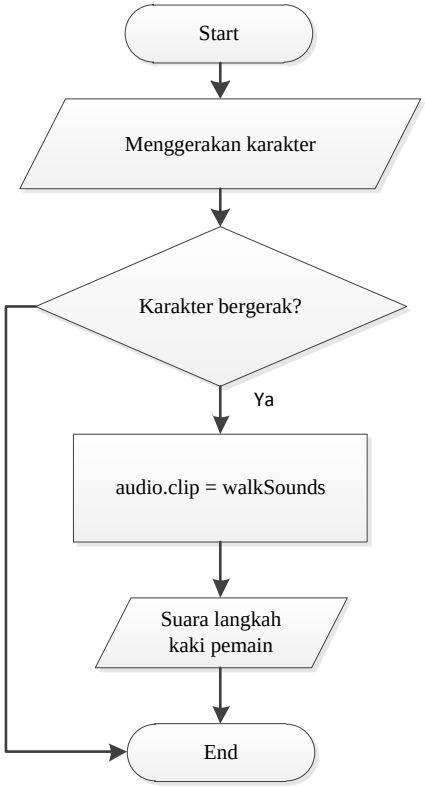
Method ini akan melakukan pengecekan terhadap setiap tembakan / *damage* dari musuh yang mengenai pemain.



Gambar III.68 *Flowchart Method ApplyDamage*

2. *Method PlayStepSounds*

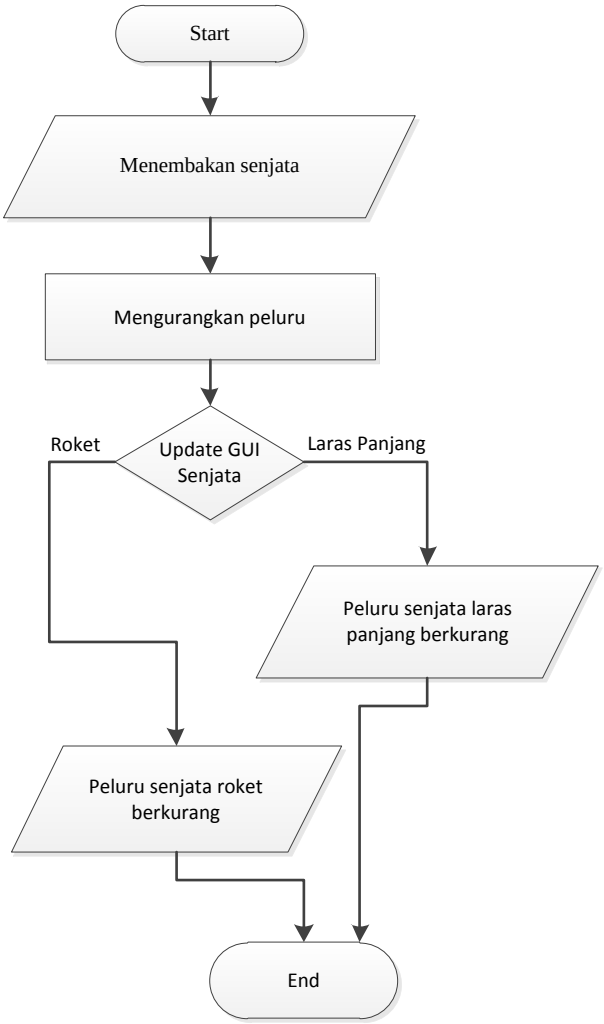
Method ini akan memberikan efek suara langkah kaki terhadap karakter ketika pemain menggerakan karakter.



Gambar III.69 *Flowchart Method ApplyDamage*

3. *Method UpdateGUI*

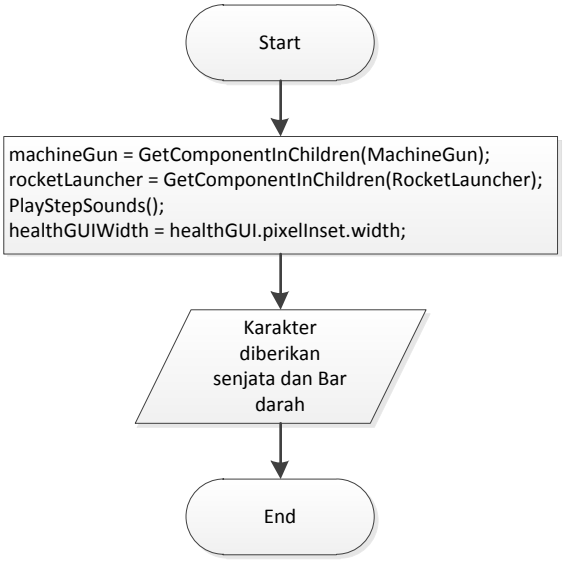
Method ini akan melakukan perubahan terhadap GUI berupa pengurangan peluru ketika pemain menembak, baik menembak menggunakan senjata laras panjang ataupun senjata roket.



Gambar III.70 *Flowchart Method UpdateGUI*

4. *Method Awake*

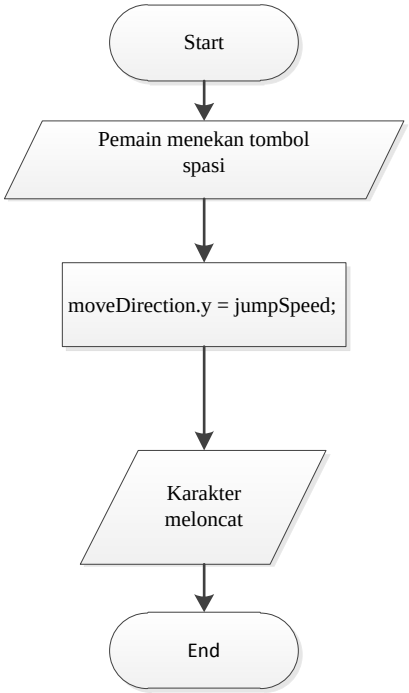
Method ini akan memberikan senjata dan GUI darah / health terhadap karakter pada saat permaian dimulai



Gambar III.71 *Flowchart Method Awake*

5. *Method FixedUpdate*

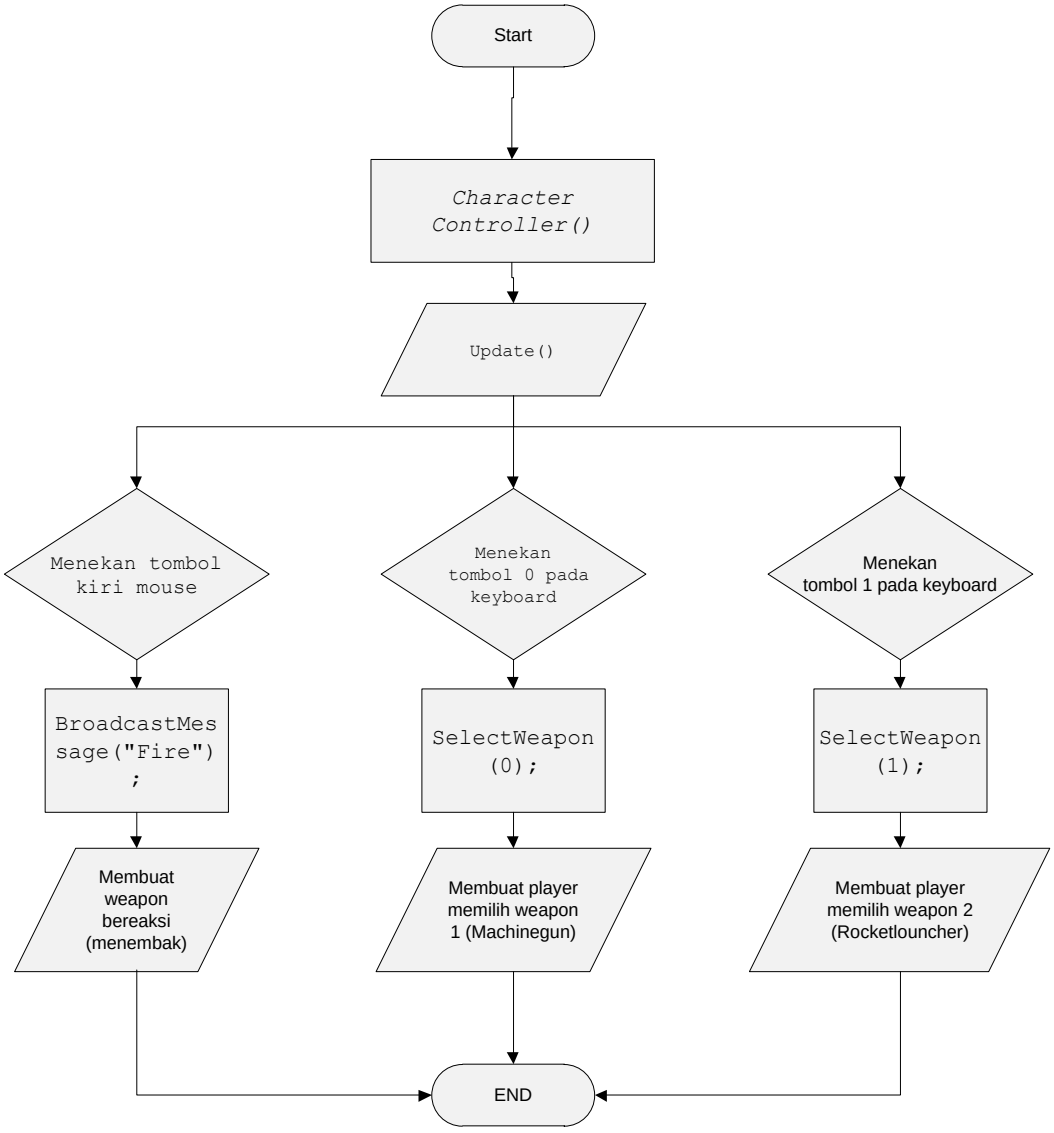
Method ini akan melakukan update berupa gerakan meloncat terhadap karakter yang diawali dari inputan yang dilakukan pemain



Gambar III.72 *Flowchart Method FixedUpdate*

6. *Method Player Weapon*

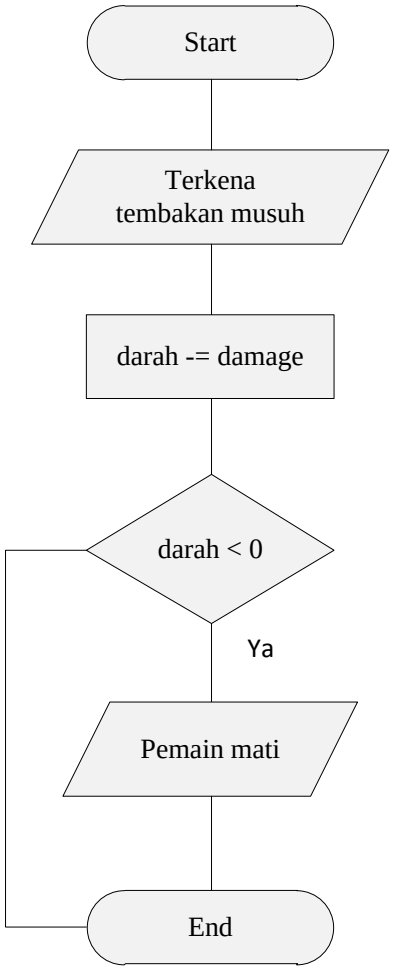
Method Player Weapon digunakan untuk menampung semua atribut-atribut senjata dari yang ada pada *FPS Player*. Untuk lebih jelasnya dapat dilihat pada gambar *flowchart* dibawah ini:



Gambar III.73 *Flowchart Method PlayerWeapon*

7. *Method Character Damage*

Method character damage digunakan untuk membuat character menerima damage yang akan mengurangi *hitpoints*. Untuk lebih jelasnya dapat dilihat pada gambar *flowchart* dibawah ini:



Gambar III.74 *Flowchart Method CharacterDamage*