BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem ini merupakan cakupan dari analisis yang berhubungan dengan *game* yangakan dibangun diantaranya analisis masalah terhadap *game* yang sejenis, analisis kebutuhan fungsional, dan analisis non fungsional.

3.1.1 Analisis Masalah

Berdasakan latar belakang yang sudah diuraikan di atas, maka masalahnya yaitu *game Tower defense* yang bertemakan hutan yang ada saat ini masih kebanyakan dalam *gameplay*-nya hanya berfokus pada strategi penempatan *tower* saja, tidak ada strategi bagaimana menanam dan merawat pohon sampai besar serta menjaganya dari serangan musuh sehingga *gameplay* lebih *variatif*. Selain itu pada *gameplay*-nya masih jarang yang mengimplementasikan hutan dan karakter khas Indonesia yang menyisipkan pengetahuan mengenai kondisi Indonesia pada saat ini, serta musuh hanya bergerak pada jalurnya saja, tidak bisa berpindah ke jalur lain dan menyerang saat berhadapan dengan *Tower* sehingga pola penyerangan kurang optimal. Selanjutnya Perumusan masalah dari permasalahan di atas adalah sebagai berikut :

1. Bagaimana membuat *game Tower defense* pengendalian pada *gameplay*-nya tidak berfokus pada *tower* saja.
2. Bagaimana Membuat *game Tower defense* yang bertemakan hutan Indonesia dengan ciri khas karakter Indonesia yang memberikan pengetahuan mengenai kondisi hutan Indonesia saat ini.
3. Bagaimana membuat pergerakan karakter NPC lebih dinamis dengan bisa menyerang *tower* dan dapat berpindah jalur pada saat berhadapan dengan *tower*.

3.1.2 Analisis *Game* sejenis

3.1.2.1 *Game* The Utans



Gambar 3. 1 *Screenshoot Game* The Utans

Game The Utans merupakan *game Tower Defense* untuk PC yang di kembangkan oleh sebuah Studio Pengembang *game* asal Bandung yang bernama Ulingamework. *Game* ini di bangun menggunakan flash, sesuai dengan namanya tujuan utama dari *game* ini ialah untuk mempertahankan hutan dari serangan para makhluk asing.



Gambar 3. 2 *Gameplay* The Utans

Game ini menceritakan tentang kehidupan para penghuni planet Mavas yang di sebut Utans dimana planet ini memiliki karakteristik yang serupa dengan hutan di Kalimantan dan Sumatra. Diceritakan Saat ini planet Mavas sedang dalam

serangan oleh para pendatang yang menyerang tempat tinggal para Utans. Sehingga para utan harus bisa mempertahankan planet mereka dengan melawannya.

Pada *game* ini terdiri dari 12 *level* dengan 10 *wave* (gelombang serangan musuh) di tiap *level*-nya, tiap *level* kesulitannya makin tinggi. Pada *game* ini juga terdapat dua jenis orang utan yang bertugas sebagai *Tower*, pohon yang ditanam yang juga berfungsi sebagai *Tower* dan ranjau yang ditempatkan pada jalur musuh. Terdapat dua pilihan *upgrade Tower*, pertama *upgrade* standar yaitu *upgrade* secara langsung dan yang kedua *upgrade* kemampuan dari biji-bijian yang didapatkan dari hasil mengalahkan musuh. Setiap biji-bijian yang didapatkan ada juga yang berubah menjadi poin yang bisa digunakan untuk membeli *Tower* baru dan membeli tanaman.

Setiap *Tower* bisa ditempatkan pada sebuah pohon yang tersebar di pinggir jalur dan bisa juga ditempatkan di sepanjang jalur *defense* namun untuk tanaman hanya bisa ditempatkan di jalur *defense* saja.

Kelebihan *Game* The Utans

Kelebihan *game* ini antara lain :

1. Antarmuka yang menarik.
2. Karakter *Tower* yang unik.
3. Menampilkan unsur daerah dalam *gamplay*-nya.
4. Musuh bisa menyerang *Tower*.

Kekurangan *Game* The Utans

Kekurangan *game* ini antara lain :

1. Musuh tidak bisa berpindah jalur saat berhadapan dengan *Tower*.
2. *Gameplay* yang cukup sulit.
3. Musuh kurang cerdas sehingga penyerangan kurang optimal.

3.1.2.2 *Game* Field Runner 2



Gambar 3. 3 *Screenshoot Game* Field Runner 2

Game Field Runner 2 merupakan *game Tower defense* yang dikembangkan oleh Subatomic Studio. Field Runner2 merupakan *game* yang dikembangkan untuk beberapa *platform* seperti Windows, IOS, dan Android.



Gambar 3. 4 *Gameplay* Field Runner 2

Game ini terdiri dari 24 *level* dan 4 map *background* yang berbeda, dalam *game* ini terdapat 3 tingakatan kesulitan yaitu, mode *casual*, mode *tough*, dan mode *heroic*. Pada *Gameplay*-nya hampir sama dengan *game Tower defense* yang lainnya, yaitu pemain harus menyusun strategi penempatan *Tower*. Pada *game* ini pemain harus bisa bertahan melawan musuh yang terdiri dari 60 sampi 70 *waves* dengan batas hanya 20 musuh saja yang boleh masuk ke area pertahanan, apabila

lebih, maka permainan akan berakhir. Pada *game* ini terdapat lebih dari 2 jenis *tower* dan lebih dari 20 senjata lainnya.

Kelebihan *Game* Field Runner 2

Kelebihan *game* ini antara lain :

1. Antarmuka yang bagus menarik.
2. Laju *game* bisa dipercepat.
3. Pergerakan musuh sudah dinamis.

Kekurangan *Game* Field Runner 2

Kekurangan *game* ini adalah Musuh tidak bisa menyerang *Tower*.

3.1.2.3 *Game* Ace defense



Gambar 3. 5 *Screenshoot Game* Ace Defense

Ace Defense merupakan *Game Tower defense* yang misinya untuk menyelamatkan teknologi yang berusaha di curi oleh musuh. Pada *game* ini terdiri dari sepuluh *level* dengan bentuk jalur yang berbeda-beda. Musuh dalam *game* ini *variatif* dengan kekuatan yang berbeda beda. Musuh sendiri dibedakan dari warnanya dan kendaraan yang ditumpanginya. Musuh diharuskan untuk terus maju sampai basis pertahanan bangunan, ketika musuh tersebut sampai pada basis pertahanan maka nyawa yang tertera di atas bangunan akan berkurang seiring banyaknya musuh yang berhasil masuk ke dalam bangunan.



Gambar 3. 6 *Gameplay Game* Ace Defense

Pada *game* ini disediakan lima jenis *Tower* yang berbeda-beda baik dalam hal kekuatan serang maupun dalam jarak jangkauan serang terhadap musuh. Dan hanya dua *Tower* yang bisa digunkan pada awal-awal *level*. Setiap *Tower* mempunyai harga yang berbeda-beda, untuk memperoleh poin yang digunakan membeli *tower*, maka pemain harus membunuh setiap musuh yang ada, dan setiap musuh datang secara bergelombang (*wave*). Dalam *gameplay*-nya pengaturan datangnya *wave* bisa diatur otomatis ataupun manual dan *speed game* bisa diatur cepat atau sedang.

Kelebihan *Game* Ace Defense

Kelebihan *game* ini antara lain :

1. Laju *game* bisa dipercepat.
2. *Gameplay* relatif mudah dimainkan.
3. Variasi Musuh cukup banyak.

Kekurangan *Game* Ace Defense

Kekurangan game ini antara lain :

1. Musuh tidak bisa menyerang *tower*.
2. Gerakan musuh *statis*.
3. *Gameplaya* hanya *singline*, jadi musuh tidak bisa berpindah jalur.

3.1.3 Analisis *Game* yang akan dibangun

3.1.3.1 Pengenalan

Game yang akan dibangun berjudul Selamatkan Hutan Indonesia yang ber-*genre Tower Defense*. *Game* ini dibangun dengan menginterpretasikan hutan Indonesia sebagai tema dari permainan yang bertujuan untuk memperkenalkan hutan Indonesia sebagai salah satu hutan yang perlu dilindungi kedua setelah hutan amazon.

Batasan-batasan yang terdapat pada *game* Selamatkan Hutan Indonesia ini adalah sebagai berikut :

1. Game yang dibuat berbasis *desktop* dengan tampilan jenis 2D.
2. *Tools* yang digunakan untuk membangun *game* ini menggunakan bahasa Pemrograman Java dengan *software* pembangun Greenfoot.
3. *Game* berupa *single player game*.
4. *Game* berupa *offline game*.
5. Target penggunanya adalah untuk umum.
6. Media interaksi permainan *game* menggunakan mouse.
7. *Game* terdiri dari 3 *level* dengan map yang berbeda di tiap *level*-nya.
8. Sebelum *game* di mulai dijelaskan terlebih dahulu kondisi hutan, jumlah pembalakan dan jenis jenis pohon di pulau yang di pilih sebagai map tiap *level*.
9. Dalam *game* di implementasikan karakter dengan ciri khas daerah Indonesia.
10. Pemodelan perancangan sistem menggunakan UML.
11. Metode yang akan digunakan adalah Logika Fuzzy sebagai implementasi dari pergerakan musuh agar serangan yang di bangun lebih optimal dan dinamis.

3.1.3.2 Storyline

Game Selamatkan Hutan Indonesia menceritakan tentang menjaga hutan dari penebang liar (*ilegall loging*) dan melestarikan hutan dengan menanam tumbuhan. Indonesia adalah salah satu negara yang mempunyai area hutan yang

luas yang tersebar di seluruh kepulauan di Indonesia, juga terdiri dari berbagai jenis hutan dan jenis pohon yang berbeda di tiap daerahnya. Namun saat ini hutan di wilayah Indonesia dalam kondisi sangat kritis, yang diakibatkan oleh maraknya aktivitas Pembalakan Liar (*Ilegal Logging*), konservasi kawasan hutan menjadi perkebunan sawit, karet dan juga kebakaran hutan. Agar kondisi hutan Indonesia pada saat ini lebih diketahui oleh berbagai kalangan masyarakat dengan harapan meningkatkan keprihatinan terhadap kondisi hutan indonesia.

Pada suatu hari terdengar isu akan pembalakan hutan secara berskala besar di Indonesia, salah seorang penjaga hutan tersebut mendapatkan kabar dari salah seorang mantan pekerja daripenebang hutan liar tersebut. Kabar tersebut sempat mencemaskan terjadinya pembalakan yang akan terjadi secara diam-diam di Indonesia, hingga tidak lama kemudian seorang penjaga yang berasal dari pulau jawa tersebut mengisaratkan kepada rekan-rekan penjaga hutan di daerah Sumatra, Kalimantan, Sulawesi, dan Papua untuk segera berkumpul untuk membicarakan masalah itu.

Dalam pembicaraannya mereka berencana untuk membuat sebuah solusi yang dapat mengusir para penebang liar (*ilegall loging*) di titik hutan yang global, titik-titik tersebut adalah Sumatra, Jawa, Kalimantan, Sulawesi, dan Papua. Dimana titik-titik tersebut akan menjadi tujuan dari para penebang liar (*ilegall loging*). Para penjaga hutan harus dapat menahan terjadinya pembalakan hutan dan harus berhasil menanam kembali hutan yg tertebang karena para penebang liar akan secara terus menerus berdatangan.

3.1.3.3 *Gameplay*

Di awal permainan, pemain diberikan 5 *wave* yang harus diselesaikan pemain. Setiap *wave* memiliki rentang waktu selama 10 detik untuk keluar. Selama rentang waktu tersebut, pemain harus memasang setiap karakter untuk melawan musuh dan harus menanam tumbuhan hingga menjadi pohon. Setiap *wave* mempunyai tingkat kesulitan yang berbeda-beda. Dalam kemampuan musuhnya, salah satu musuh dapat menghindar untuk menerobos masuk hutan dan salah satu musuh lainnya pun dapat menyerang *tower*. Pemain diberikan

modal 50 point. *Point* di dapat pemain yang menanam pohon dan pemain yang berhasil membunuh penebang liar.

3.1.3.4 Skenario *Game*

Skenario pada *game Tower defense* Selamatkan Hutan Indonesia terdapat beberapa pengembangan skenario diantaranya :

A. Alur Permainan

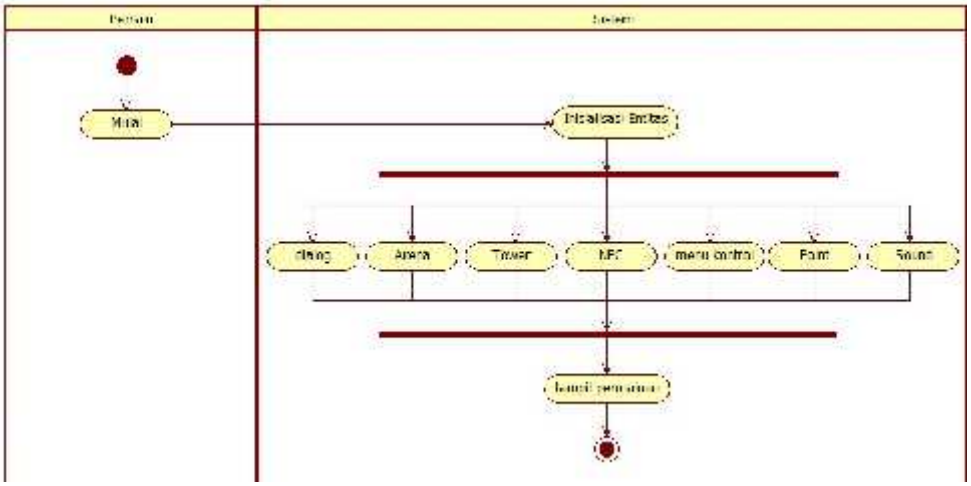
1. Alur Inisialisasi

Alur inisialisasi pada *game Tower defense* Selamatkan Hutan Indonesia dapat dilihat pada tabel 3.1.

Tabel 3. 1 Alur Inisialisasi

No	Alur Sistem
1	Pemain melakukan pilih menu yang akan dimainkan.
2	Sistem menginisialisasi seluruh entitas yang ada dalam permainan seperti, intro, arena permainan, pemain atau *tower*, musuh, menu kontrol, *point, sound.*
3	Menampilkan *entitas* permainan yang akan ditentukan oleh sistem.

Adapun hasil alur inisialisasi dari *game Tower defense* Selamatkan Hutan Indonesia tersusun dari beberapa aktivitas yang akan digambarkan melalui *activity diagram* berikut :



Gambar 3. 7 Alur Inisialisasi

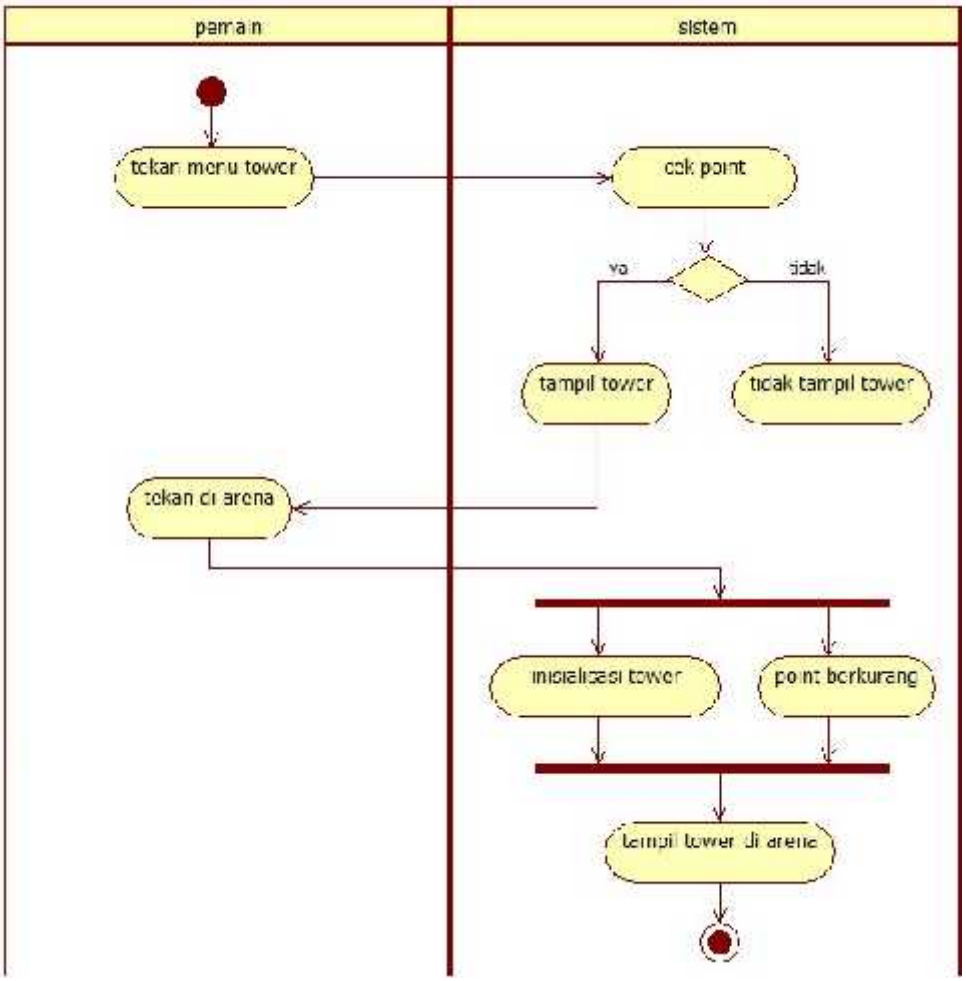
2. Alur kontrol membuat *tower Game Tower defense* Selamatkan Hutan Indonesia

Alur kontrol membuat *tower* pada *game Tower defense* Selamatkan Hutan Indonesia dapat dilihat pada tabel 3.2.

Tabel 3. 2 Alur Kontrol Membuat *Tower*

No	Alur sistem
1	Pemain memilih menu *tower*
2	Sistem menge-cek *gold* yang dimiliki pemain
3	Sistem menampilkan *tower*
4	Pemain memilih letak *tower* di arena
5	Sistem menampilkan *tower* di arena
6	Sistem mengurangi jumlah *gold* yang dimiliki pemain
7	Sistem menginisialisasi entitas *tower*

Adapun hasil alur kontrol membuat *tower* dari *game Tower defense* Selamatkan Hutan Indonesia tersusun dari beberapa aktivitas yang akan digambarkan melalui *activity diagram* berikut :



Gambar 3. 8 Alur Kontrol Membuat *Tower*

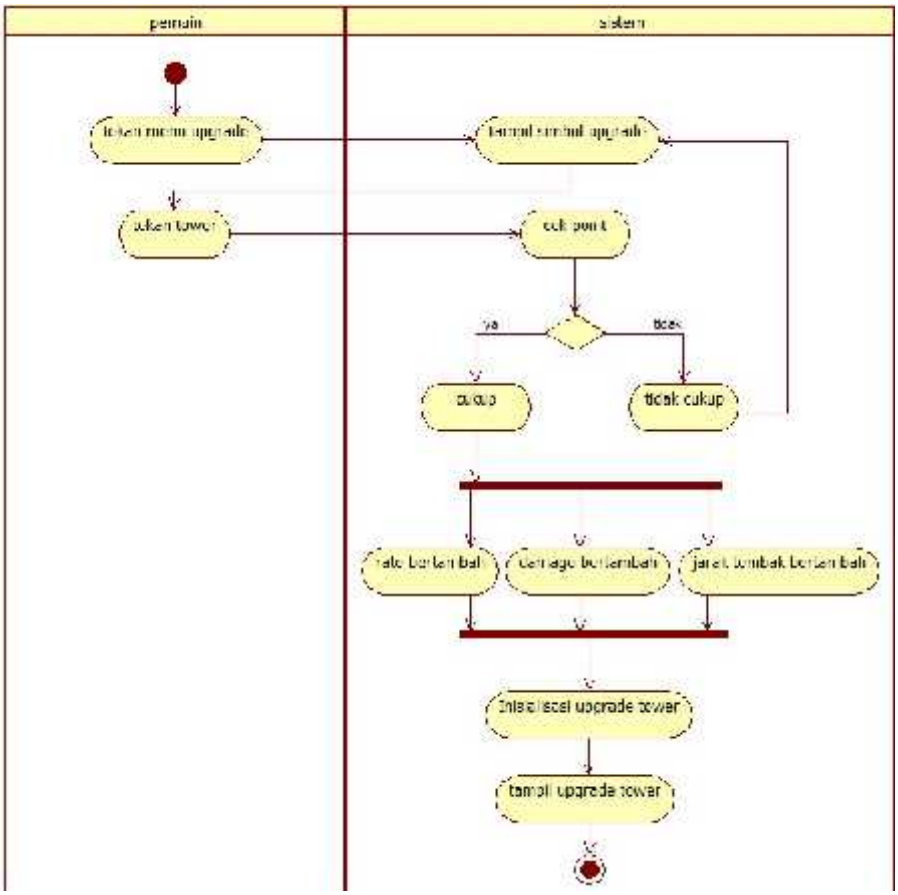
3. Alur *upgrade tower Game Tower defense* Selamatkan Hutan Indonesia

Alur *upgrade tower* pada *game Tower defense* Selamatkan Hutan Indonesia dapat dilihat pada tabel 3.3.

Tabel 3. 3 Alur *Upgrade Tower*

No	Alur sistem
1	Pemain memilih menu *upgrade tower*
2	Sistem menampilkan simbol *upgrade*
3	Pemain memilih *tower* yang berada di arena
4	Sistem mengecek *gold* yang dimiliki pemain
5	Sistem menambah jumlah tembakan *tower*
5	Sistem menambah *damage* tembakan *tower*
6	Sistem menambahkan jarak tembak *tower*

Adapun hasil Alur *upgrade tower* dari *game Tower defense* Selamatkan Hutan Indonesia tersusun dari beberapa aktivitas yang akan digambarkan melalui *activity diagram* berikut :



Gambar 3. 9 Alur *Upgrade Tower*

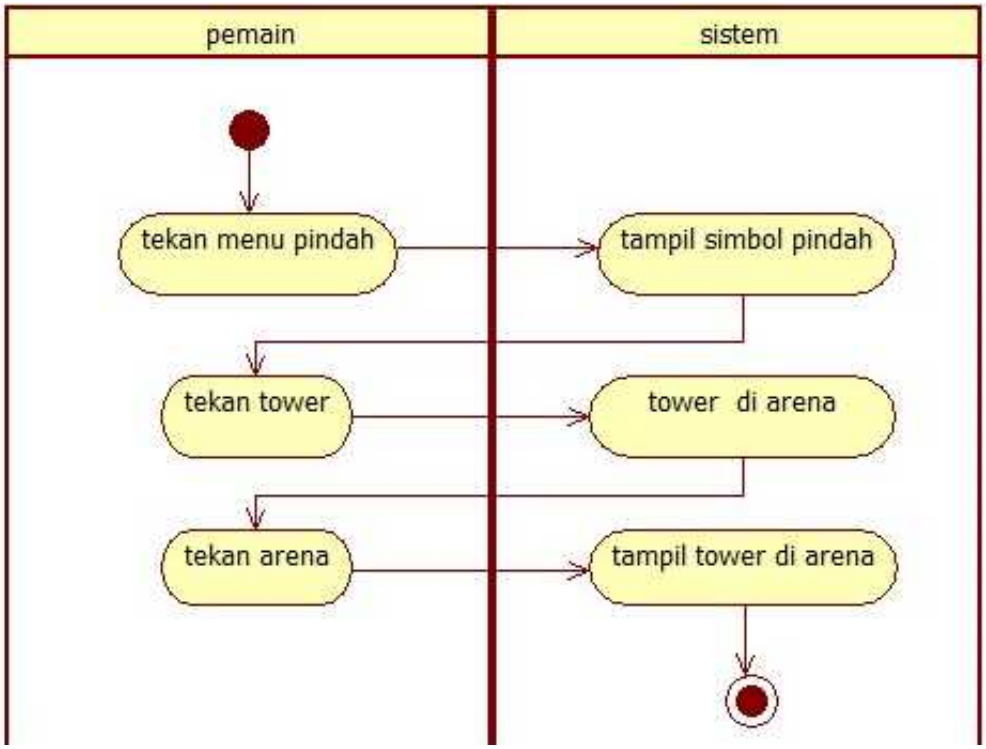
4. Alur Pindah *tower* pada *Game Tower defense* Selamatkan hutan Indonesia

Alur *Pindah tower* pada *game Tower defense* Selamatkan Hutan Indonesia dapat dilihat pada tabel 3.4.

Tabel 3. 4 Alur Pindah *Tower*

No	Alur sistem
1	Pemain memilih menu Pindah
2	Sistem menampilkan simbol Pindah
3	Pemain memilih *tower* yang berada di arena
4	Sistem memindahkan *tower* di arena

Adapun hasil Alur pindah *tower* dari *game Tower defense* Selamatkan Hutan Indonesia tersusun dari beberapa aktivitas yang akan digambarkan melalui *activity diagram* berikut :



Gambar 3. 10 Alur Pindah *Tower*

B. Misi Permainan

Misi permainan pada *game Tower defense* Selamatkan Hutan Indonesia dapat dilihat pada Tabel 3.5 berikut :

Tabel 3. 5 Misi Permainan

Misi permainan
Pemain harus menempatkan *tower* di dalam arena yang berupa seorang penjaga hutan dan pohon-pohon. Ketika pemain menempatkan *tower*, secara bersamaan musuh pun datang mendekati untuk mencoba menerobos masuk kedalam hutan. Musuh tersebut adalah para penebang liar yang menggunakan alat-alat tebang yang bermacam-macam. Pemain bertujuan untuk menjaga hutan-hutan yang akan ditebang oleh penebang hutan tersebut yaitu hutan di pulau Sumatra, Jawa, dan Kalimantan.

3.1.3.4 Analisis Algoritma *Fuzzy* pada NPC

Analisis *fuzzy* yang digunakan pada tugas akhir ini digunakan sebagai pengendali untuk menentukan jalur musuh yang aman. Dalam kaidah *fuzzy*, sistem dapat melakukan penalaran dengan prinsip serupa seperti manusia melakukan penalaran dengan nalurinya.

3.1.3.4.1 Sistem Inferensi Fuzzy (*Fuzzy Inference system*)

Sistem inferensi *fuzzy* yang digunakan dalam perancangan sistem untuk mengambil keputusan dalam menentukan jalur teraman. Sistem inferensi *fuzzy* mempunyai salah satu metode yaitu mamdani, dimana mekanismenya sistem inferensi *fuzzy* bekerja dalam 4 tahap yaitu, pembentukan himpunan *fuzzy* (*fuzzyfikasi*), fungsi implikasi (aturan), komponen aturan, dan penegasan (*defuzzyfikasi*).

1. Pembentukan Himpunan *Fuzzy*

   Pada metode mamdani, baik variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan *fuzzy*. Pada proses tersebut disebut juga proses *fuzzyfikasi*. *Fuzzyfikasi* adalah proses mengubah masukan eksak menjadi masukan berupa derajat keanggotaan.

   A. Variabel *Fuzzy*

      Variabel yang digunakan dalam pembangunan *game Tower defense* Selamatkan Hutan Indonesia adalah jumlah *tower* , jarak dari jalur, dan *level tower*.

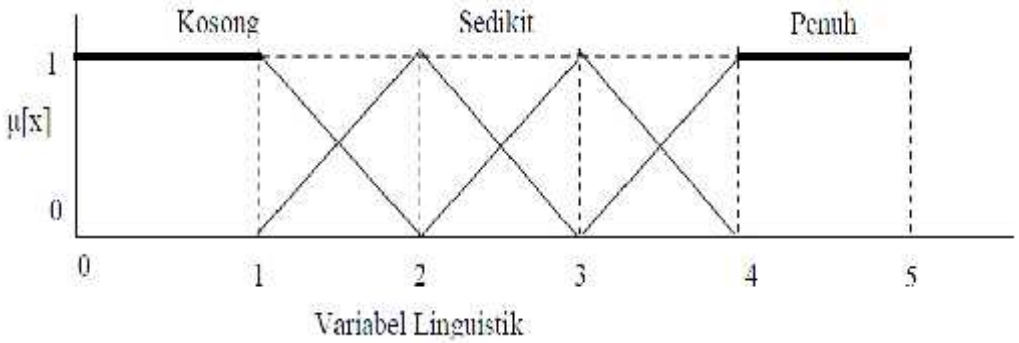
   B. Himpunan *Fuzzy*

Dalam *game Tower defense* Selamatkan Hutan Indonesia ini, himpunan *fuzzy* untuk variabel jumlah *tower* dibagi menjadi 3 kategori, yaitu sebagai berikut :

Kosong	Jumlah *tower* < 1
Sedikit	1 < jumlah *tower* <= 4
Penuh	4< jumlah *tower* <= 5

Tabel 3. 6 Variabel Himpunan *Fuzzy* Untuk Jumlah *Tower*

Variabel	Kosong	Sedikit	Penuh
Jumlah *tower* jalur 1	0	1 – 4	5
Jumlah *tower* jalur 2	0	1 – 4	5
Jumlah *tower* jalur 3	0	1 – 4	5
Jumlah *tower* jalur 4	0	1 – 4	5



Gambar 3. 11 Himpunan Fuzzy Pada Variabel Jumlah *Tower*

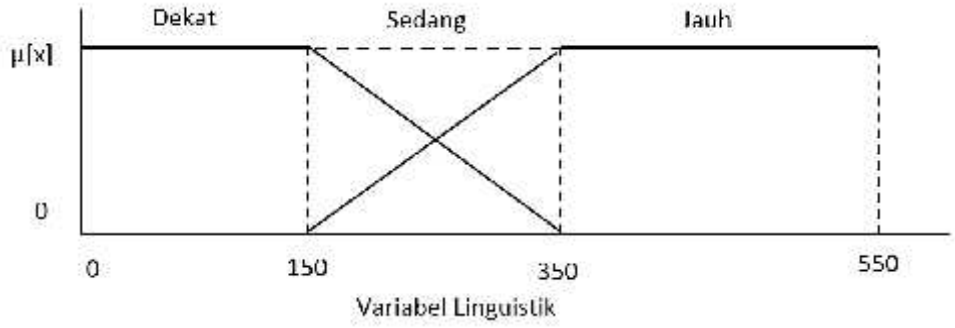
Untuk variabel jarak dari jalur yaitu dekat, sedang dan jauh, Perbandingan variabelnya adalah sebagai berikut :

Dekat	Jarak dari jalur <= 150
Sedang	150 < jarak dari jalur <550
Jauh	jarak dari jalur >= 550

Tabel 3. 7 Variabel Himpunan *Fuzzy* Untuk Jarak Dari Jalur

Variabel	Dekat	Sedang	Jauh
Jarak dari jalur 1	0 – 150	150 – 450	550
Jarak dari jalur 2	0 – 150	150 – 450	550
Jarak dari jalur 3	0 – 150	150 – 450	550

Jarak dari jalur 4	0 – 150	150 – 450	550



Gambar 3. 12 Himpunan *Fuzzy* Pada Variabel Jarak Dari Jalur

Untuk variabel *level tower* yaitu kuat dan lemah, Perbandingan variabelnya adalah sebagai berikut :

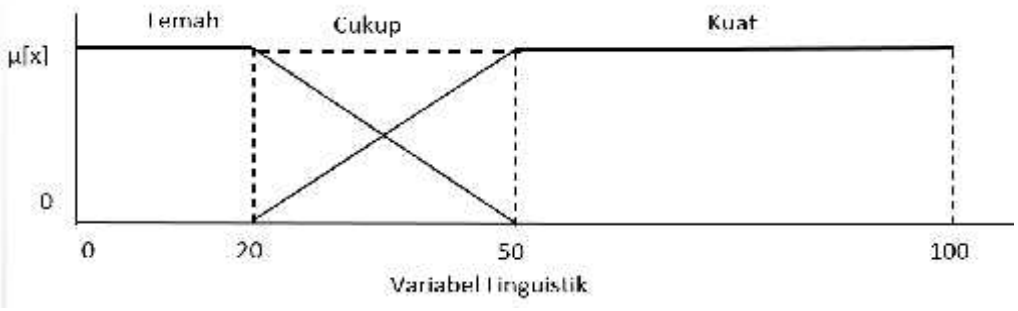
Lemah *level tower* <= 20

Cukup 20 <= *level tower* <= 50

Kuat *level tower* >= 50

Tabel 3. 8 Variabel Himpunan *Fuzzy* Untuk *Level Tower*

Variabel	Lemah	Cukup	Kuat
Level tower 1	0 – 20	20 – 50	50
Level tower 2	0 – 20	20 – 50	50
Level tower 3	0 – 20	20 – 50	50
Level tower 4	0 – 20	20 – 50	50



Gambar 3. 13 Himpunan *Fuzzy* Pada Variabel *Level Tower*

2. Fungsi Implikasi

Tiap-tiap aturan aturan pada basis *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*. Variabel yang digunakan dalam sistem ini terdiri dari 12 *input* (jumlah *tower* rute 1, jumlah *tower* rute 2, jumlah *tower* rute 3, jumlah *tower* rute 4, jarak dari jalur 1, jarak dari jalur 2, jarak dari jalur 3, jarak dari jalur 4, *level tower* 1, *level tower* 2, *level tower* 3, *level tower* 4) dan *output* (tentukan jalur teraman). Variabel jumlah *tower* terdiri dari 3 himpunan *fuzzy* yaitu, kosong, sedikit, penuh. Variabel jarak dari jalur memiliki 3 himpunan *fuzzy* yaitu, dekat sedang, jauh. Variabel *level tower* memiliki 3 himpunan yaitu, lemah, cukup, kuat.

3. Komponen Aturan

Tiap-tiap aturan pada basis pengetahuan *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Metode yang digunakan dalam melakukan inferensi sistem *fuzzy* yaitu Metode Max (*Maximum*). Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *fuzzy*, dan mengaplikasikannya ke *output*.

Aturan yang diberikan di dalam implementasi *fuzzy* pada *game* ini adalah sebagai berikut :

1) **IF** jarak dekat **AND** jumlah kosong **AND** *level* lemah **THEN** Aman.
2) **IF** jarak dekat **AND** jumlah sedikit**AND** *level* cukup**THEN** Sedang.
3) **IF** jarak dekat **AND** jumlah penuh **AND** *level* kuat **THEN** Berbahaya.
4) **IF** jarak sedang **AND** jumlah kosong **AND** *level* lemah **THEN** Aman.
5) **IF** jarak sedang **AND** jumlah sedikit **AND** *level* cukup **THEN** Sedang.
6) **IF** jarak sedang **AND** jumlah penuh **AND** *level* kuat **THEN** Berbahaya.
7) **IF** jarak jauh **AND** jumlah kosong **AND** *level* lemah **THEN** Aman.
8) **IF** jarak jauh **AND** jumlah sedikit **AND** *level* cukup **THEN** Sedang.
9) **IF** jarak jauh **AND** jumlah penuh **AND** *level* kuat **THEN** Berbahaya.

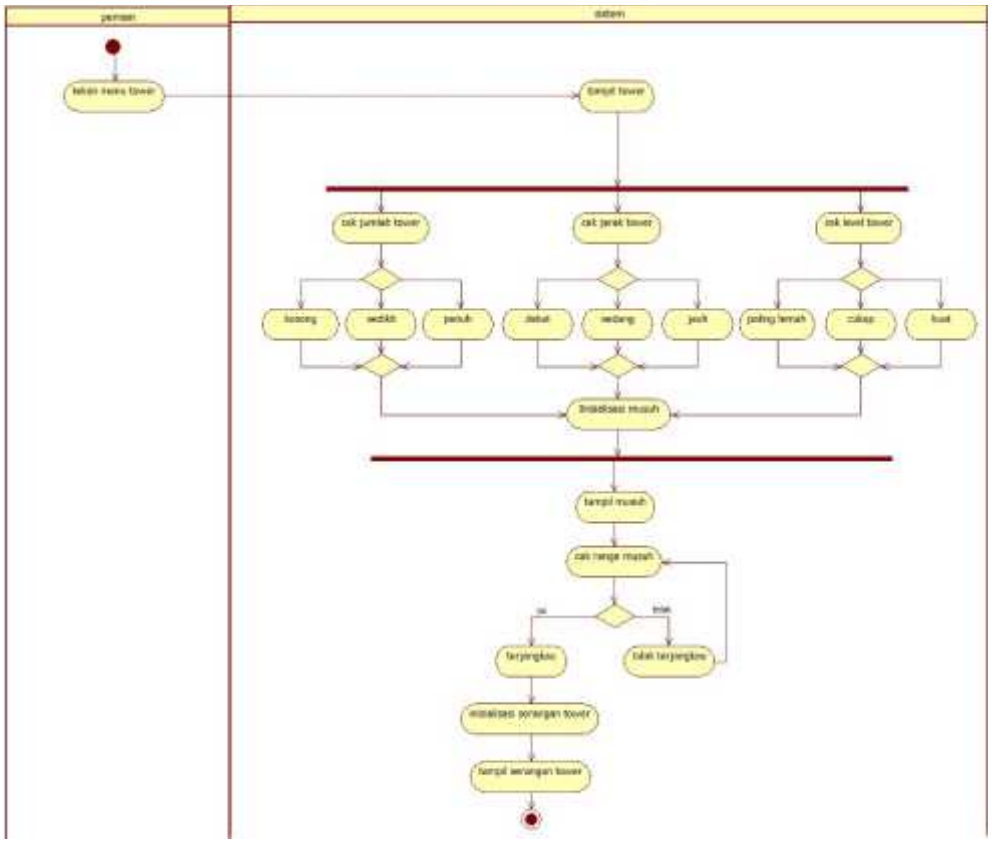
4. Keluaran (*Defuzzyfikasi*)

Defuzzyfikasi merupakan proses pengubahan besaran *fuzzy* yang disajikan dalam bentuk himpunan-himpunan *fuzzy* keluaran dengan fungsi keanggotaan. Keluaran yang dihasilkan merupakan suatu bilangan pada himpunan *fuzzy*.

Input dari proses *defuzzyfikasi* adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada himpunan *fuzzy* tersebut.

3.1.3.4.2 Analisis metode pada NPC

Metode yang diterapkan pada NPC dalam *game Tower defense* Selamatkan Hutan Indonesia digambarkan melalui *Diagram activity* berikut :



Gambar 3. 14 Penerapan *Fuzzy Logic* Pada NPC

3.1.3.4.3 Contoh Kasus

Berikut ini adalah contoh kasus dalam penentuan jalur teraman.

Kondisi permainan :

Tabel 3. 9 Kondisi Permainan

Komponen Variabel	Nilai
Jumlah *tower* jalur 1	0
Jumlah *tower* jalur 2	2
Jumlah *tower* jalur 3	3
Jumlah *tower* jalur 4	5
Jarak *tower* jalur 1	150
Jarak *tower* jalur 2	250
Jarak *tower* jalur 3	350
Jarak *tower* jalur 4	550
Kuat *tower* jalur 1	20
Kuat *tower* jalur 2	30
Kuat *tower* jalur 3	40
Kuat *tower* jalur 4	50

1. Penentuan Status

Pembentukan himpunan *fuzzy* untuk menentukan status pengambilan jalur teraman.

Tabel 3. 10 Himpunan *Fuzzy* Penentuan Status Jalur

Hubungan variabel	Aman	Sedang	Berbahaya
Jumlah *tower* jalur 1, jarak *tower* jalur 1 dengan kuat *tower* jalur 1	Jumlah *tower* **kosong** = 0, jarak *tower* **dekat** = 150, kuat *tower* **lemah** = 20	Jumlah *tower* **sedikit** = 1 – 4 , jarak *tower* **dekat** = 150, kuat *tower* **sedang** = 20 – 40	Jumlah *tower* **penuh** = 5, jarak *tower* **dekat** = 150, kuat *tower* **kuat** = 50
Jumlah *tower* jalur 2, jarak *tower* jalur 2 dengan kuat *tower* jalur 2	Jumlah *tower* **kosong** = 0, jarak *tower* **sedang** = 150 – 350, kuat *tower* **lemah** = 20	Jumlah *tower* **sedikit** = 1 – 4, jarak *tower* **sedang** = 150 – 350, kuat *tower* **sedang** = 20 – 40	Jumlah *tower* **penuh** = 5, jarak *tower* **sedang** = 150 – 350, kuat *tower* **kuat** = 50
Jumlah *tower* jalur 3, jarak *tower* jalur 3 dengan kuat *tower* jalur 3	Jumlah *tower* **kosong** = 0, jarak *tower* **sedang** = 150 – 350, kuat *tower* **lemah** = 20	Jumlah *tower* **sedikit** = 1 – 4, jarak *tower* **sedang** = 150 – 350, kuat *tower* **sedang** = 20 – 40	Jumlah *tower* **penuh** = 5, jarak *tower* **sedang** = 150 – 350, kuat *tower* **kuat** = 50
Jumlah *tower* jalur 4, jarak *tower* jalur 4 dengan kuat *tower* jalur 4	Jumlah *tower* **kosong** = 0, jarak *tower* **jauh** = 550, kuat *tower* **lemah** = 20	Jumlah *tower* **sedikit** = 1 – 4 , jarak *tower* **jauh** = 550, kuat *tower* **sedang** = 20 – 40	Jumlah *tower* **penuh** = 5, jarak *tower* **jauh** = 550, kuat *tower* **kuat** = 50

2. Komposisi Aturan *Fuzzy*

Pada solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *fuzzy*, dan mengaplikasikannya ke output dengan menggunakan operator AND (*union*). Jika semua proposisi telah dievaluasi, maka output akan berisi suatu himpunan fuzzy yang merefleksikan konstribusi dari tiap-tiap proposisi. Berdasarkan himpunan *fuzzy* di atas dapat terlihat derajat keanggotaan dari setiap komponen nilai berdasarkan aturan (*rule*) yang sudah dijelaskan sebelumnya.

3. Aplikasi Fungsi Implikasi

Berdasarkan komposisi aturan yang sudah dibuat

Sebelumnya, kondisi permainan yang diuji pada contoh kasus ini dapat dikelompokan seperti terlihat pada tabel 3.11.

Tabel 3. 11 Himpunan *Fuzzy* pada Fungsi Implikasi

Komponen Variabel	**Aman**	**Sedang**	Berbahaya
Jumlah *tower* jalur 1	0	0	0
Jumlah *tower* jalur 2	0	2	0
Jumlah *tower* jalur 3	0	3	0
Jumlah *tower* jalur 4	0	0	5
Jarak *tower* jalur 1	150	0	0
Jarak *tower* jalur 2	0	250	0
Jarak *tower* jalur 3	0	350	0
Jarak *tower* jalur 4	0	0	550
Kuat *tower* jalur 1	20	0	0
Kuat *tower* jalur 2	0	30	0
Kuat *tower* jalur 3	0	40	0
Kuat *tower* jalur 4	0	0	50

4. Penegasan (*defuzifikasi*)

Berdasarkan kondisi tersebut maka menghasilkan penentuan status yang terdapat didalam tabel 3.12.

Tabel 3. 12 Hasil Penentuan Status

Komponen Variabel	Status
Jumlah *tower* jalur 1	AMAN
Jumlah *tower* jalur 2	SEDANG
Jumlah *tower* jalur 3	SEDANG
Jumlah *tower* jalur 4	BERBAHAYA
Jarak *tower* jalur 1	AMAN
Jarak *tower* jalur 2	SEDANG
Jarak *tower* jalur 3	SEDANG
Jarak *tower* jalur 4	BERBAHAYA
Kuat *tower* jalur 1	AMAN
Kuat *tower* jalur 2	SEDANG
Kuat *tower* jalur 3	SEDANG
Kuat *tower* jalur 4	BERBAHAYA

3.1.4 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen yang dibutuhkan untuk sistem yang akan dibangun sampai sistem tersebut dapat diterapkan. Analisis kebutuhan ini dapat menentukan spesifikasi inputan yang diperlukan sistem, keluaran yang akan dihasilkan, dan proses yang dibutuhkan untuk mengolah inputan sehingga menghasilkan keluaran yang diinginkan.

3.1.4.1 Kebutuhan Perangkat Lunak (Pembangun)

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah yang diberikan kepada perangkat keras agar dapat saling berinteraksi diantaranya. Perangkat lunak yang dibutuhkan untuk pembangunan *game Tower defense* Selamatkan Hutan Indonesia ini adalah sebagai berikut :

Tabel 3. 13 Spesifikasi Perangkat Lunak (Pembangun)

Sistem operasi	Windows 7 *Ultimate* 32-bit
Compiler	Greenfoot v2.2.1
Development toolkit	JDK 7 (*Java Development Kit*)
Design tools	Adobe Photoshop CS 3
Design analysis	StarUML

3.1.4.2 Kebutuhan Perangkat Keras (Pembangun)

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi kepada perangkat keras untuk melakukan tugas tertentu. Adapun perangkat keras yang digunakan dalam membangun *game Tower defense* Selamatkan Hutan Indonesia adalah sebagai berikut :

Tabel 3. 14 Spesifikasi Perangkat Keras (Pembangun)

Nama Perangkat	**Spesifikasi**
Processor	Dual Core
RAM	1024 MB
Harddisk	40 GB
Display	Resolusi 800 x 600
Keyboard	Standar
Mouse	Standar

3.1.4.3 Kebutuhan Pengguna

Spesifikasi minimal bagi pengguna yang dapat menggunakan *game* ini adalah sebagai berikut :

Tabel 3. 15 Spesifikasi Pengguna

Peran	Pemain
Usia	7 tahun ke atas
Tingkat Keterampilan	Dapat menggunakan computer
Pengalaman	Pernah memainkan aplikasi permainan lainnya seperti *plan versus zombie*, *fieldrunner*, *the utan's*, dan lain-lain.

3.1.4.4 Kebutuhan Perangkat Lunak (Pengguna)

Pada sisi pengguna, perangkat lunak yang dibutuhkan untuk menjalankan *game* ini adalah sebagai berikut :

Tabel 3. 16 Spesifikasi Perangkat Lunak (Pengguna)

Sistem Operasi	Windows 7
Perangkat Lunak	JDK 7 (*Java Development Kit*)

3.1.4.5 Kebutuhan Perangkat Keras (Pengguna)

Pada sisi pengguna, perangkat keras yang dibutuhkan untuk menjalankan *game* ini adalah sebagai berikut :

Tabel 3. 17 Spesifikasi Perangkat Keras (Pengguna)

Nama perangkat	Spesifikasi
Processor	Minimum Pentium 4
RAM	Minimum 512 MB
Harddisk	Minimum 500 MB
Display	Resolusi 800 x 600
Keyboard	Standar
Mouse	Standar

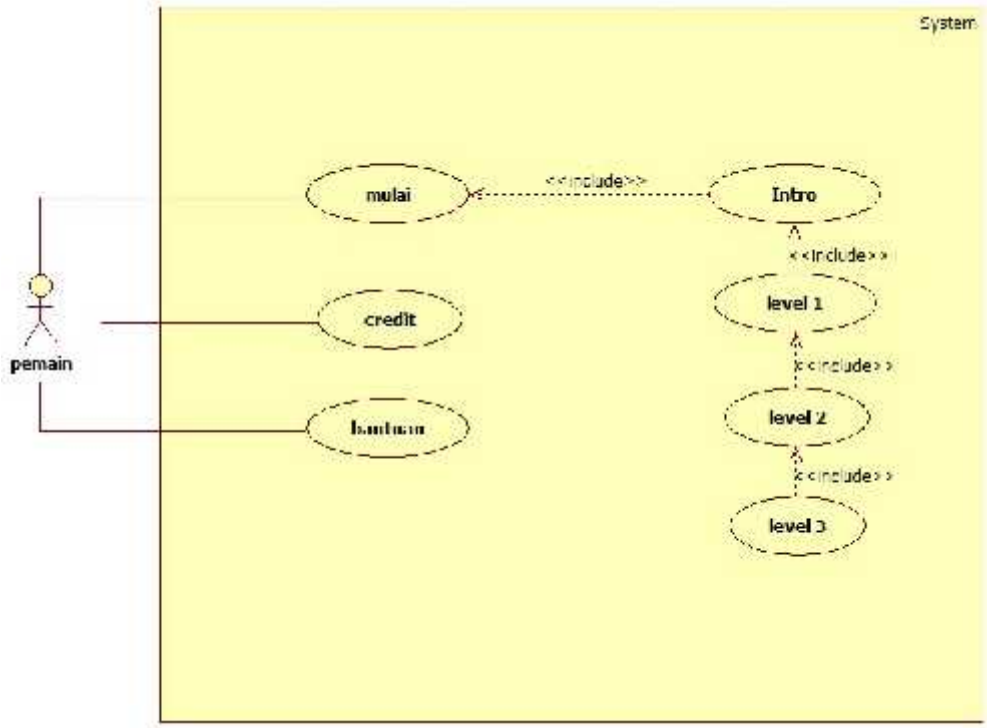
3.1.5 Analisis Kebutuhan Fungsional

Tahap pertama yang pertama dalam melakukan analisis berorientasi objek menggunakan UML adalah menentukan aktor atau pengguna sistem, dimana aktor menampilkan peran (*roles*) yang dilakukan pada sistem sebagai aksi dan respon dari sistem.

3.1.5.1 *Use Case* Diagram

Use case diagram merupakan gambaran fungsional dari sebuah sistem sehingga pengguna sistem mengerti mengenai kegunaan sistem yang akan dibangun.

Berikut ini adalah interaksi antara pengguna dengan *game Tower defense* Selamatkan Hutan Indonesia yang akan digambarkan sebagai berikut :



Gambar 3. 15 *Use Case* Diagram

3.1.5.2 Definisi *Use Case*

Definisi *use case* berfungsi untuk menjelaskan fungsi dari *use case* yang terdapat pada diagram *use case*. Definisi *use case* dijelaskan pada tabel berikut:

Tabel 3. 18 Definisi *Use Case*

No	*Use Case*	Keterangan
1	Mulai	Proses untuk memulai permainan
2	*Credit*	Proses untuk menampilkan *credit* pemain
3	Bantuan	Proses menampilkan petunjuk permainan
4	Intro	Proses menampilkan intro
5	*Level* 1	Proses permainan pada *level* 1
6	*Level* 2	Proses permainan pada *level* 2
7	*Level* 3	Proses permainan pada *level* 3

3.1.5.3 Skenario *Use Case*

Skenario pada diagram *use case* merupakan penjelasan mengenai alur kejadian yang di dalamnya menjelaskan urutan interaksi antara aktor dengan *use case* tersebut. Adapun skenario *use case* yang terdapat pada *game Tower defense* Selamatkan Hutan Indonesia adalah sebagai berikut :

Tabel 3. 19 Skenario *Use Case* Mulai

Identifikasi		
Nomor	1	
Nama	Mulai	
Tujuan	Masuk ke halaman intro	
Deskripsi	Proses untuk memulai permainan	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di menu utama *game Tower defense* Selamatkan Hutan Indonesia	
Aksi Aktor		**Reaksi Sistem**
1. Memilih menu mulai		
		2. Masuk ke tampilan Intro
3. Menekan tombol lanjut		
		4. Menampilkan Intro selanjutnya
Kondisi Akhir	Masuk ke tampil Intro	

Tabel 3. 20 Skenario *Use Case Credit*

Identifikasi		
Nomor	2	
Nama	*Credit*	
Tujuan	Menampilkan *credit* permainan	
Deskripsi	Proses untuk menampilkan *credit* permainan	
Aktor	Pemain	
Skenario Utama		
Kondisi Awal	Pemain berada di menu utama *game Tower defense* Selamatkan Hutan Indonesia	
Aksi Aktor		**Reaksi Sistem**
1. Memilih menu *credit*		
		2. Menampilkan *credit* pemain
3. Menekan tombol kembali		
		4. Menampilkan halaman utama
Kondisi Akhir	Masuk ke tampilan *credit* permainan	

Tabel 3. 21 Skenario *Use Case* bantuan

Identifikasi	
Nomor	3

Nama	Bantuan
Tujuan	Menampilkan petunjuk permainan
Deskripsi	Proses untuk menampilkan petunjuk permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama *game Tower defense* Selamatkan Hutan Indonesia
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu bantuan	
	2. Menampilkan petunjuk permainan
3. Menekan tombol kembali	
	4. Menampilkan halaman utama
Kondisi Akhir	Masuk ke tampilan petunjuk permainan

Tabel 3. 22 Skenario *Use Case* Intro

Identifikasi	
Nomor	4
Nama	Intro
Tujuan	Menampilkan Intro permainan
Deskripsi	Proses untuk menampilkan Intro permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di halaman Intro permainan
Aksi Aktor	**Reaksi Sistem**
1. Menekan tombol lanjut	
	2. Menampilkan intro selanjutnya
3. Menekan tombol kembali	
	4. Menampilkan intro sebelumnya
5. Menekan tombol selesai	
	6. Menampilkan awal permainan
Kondisi Akhir	Masuk ke tampilan awal permainan

Tabel 3. 23 Skenario *Use Case Level* 1

Identifikasi	
Nomor	5
Nama	Tampil *Level* 1

Tujuan	Menampilkan permainan *level* 1	
Deskripsi	Proses untuk memainkan permainan *level* 1	
Aktor	Pemain	
Skenario Utama		
Kondisi Awal	Pemain berada di tampilan permainan *level* 1	
Aksi Aktor		**Reaksi Sistem**
1. Menekan tombol pohon		
2. Menekan arena permainan		
		3. Tampil tumbuhan di arena permainan
4. Menekan tombol siram		
5. Menekan gambar tumbuhan		
		6. Tampil pohon
		7. Poin bertambah selama 10 detik
8. Menekan tombol *tower*		
9. Menekan *tower* di arena permainan		
		10. Tampil *tower* di arena permainan
11. Memilih *tower* di arena permainan		
12. Menekan tombol *upgrade*		
		13. Tampil *tower* yang telah di upgrade
14. Memilih *tower* di arena permainan		
15. Menekan tombol pindah *tower*		
		16. Tampil tower di kursor
17. Menekan tower di arena		
		18. *Tower* pindah di arena permainan
Kondisi Akhir	Pemain menyelesaikan permainan *level* 1	

Tabel 3. 24 Skenario *Use Case Level* 2

Identifikasi	
Nomor	6
Nama	Tampil *Level* 2
Tujuan	Menampilkan permainan *level* 2
Deskripsi	Proses untuk memainkan permainan *level* 2
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di tampilan permainan *level* 2

Aksi Aktor	Reaksi Sistem
1. Menekan tombol pohon	
2. Menekan arena permainan	
	3. Tampil tumbuhan di arena permainan
4. Menekan tombol siram	
5. Menekan gambar tumbuhan	
	6. Tampil pohon
	7. Poin bertambah selama 10 detik
8. Menekan tombol *tower*	
9. Menekan *tower* di arena permainan	
	10. Tampil *tower* di arena permainan
11. Memilih *tower* di arena permainan	
12. Menekan tombol upgrade	
	13. Tampil *tower* yang telah di upgrade
14. Memilih *tower* di arena permainan	
15. Menekan tombol pindah *tower*	
	16. Tampil tower di kursor
17. Menekan tower di arena	
	18. *Tower* pindah di arena permainan
Kondisi Akhir	Pemain menyelesaikan permainan *level* 2

Tabel 3. 25 Skenario *Use Case level* 3

Identifikasi	
Nomor	7
Nama	Tampil *Level* 3
Tujuan	Menampilkan permainan *level* 3
Deskripsi	Proses untuk memainkan permainan *level* 3
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di tampilan permainan *level* 3

Aksi Aktor	Reaksi Sistem
1. Menekan tombol pohon	
2. Menekan arena permainan	
	3. Tampil tumbuhan di arena permainan

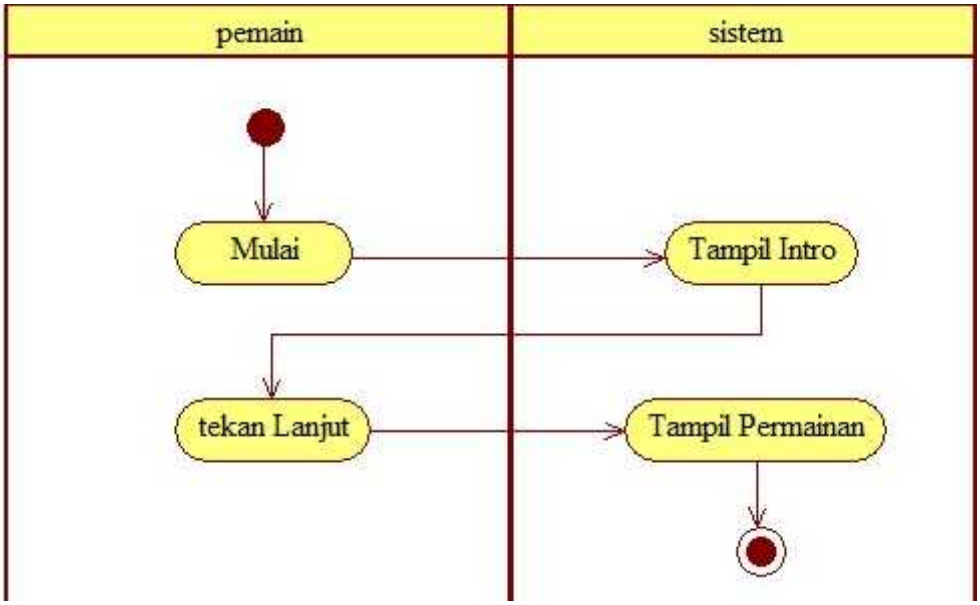
4. Menekan tombol siram	
5. Menekan gambar tumbuhan	
	6. Tampil pohon
	7. Poin bertambah selama 10 detik
8. Menekan tombol *tower*	
9. Menekan *tower* di arena permainan	
	10. Tampil *tower* di arena permainan
11. Memilih *tower* di arena permainan	
12. Menekan tombol upgrade	
	13. Tampil *tower* yang telah di upgrade
14. Memilih *tower* di arena permainan	
15. Menekan tombol pindah *tower*	
	16. Tampil *tower* di kursor
17. Menekan *tower* di arena	
	18. *Tower* pindah di arena permainan
Kondisi Akhir	Pemain menyelesaikan permainan *level* 3

3.1.5.4 *Activity* Diagram

Activity diagram mempunyai penjelasan lebih mengenai urutan suatu proses yang terjadi antara aktor dan sistem. Aktivitas yang pertama dimulai dari aktor, kemudian akan mendapat respon dari sistem yang nantinya akan menampilkan hasil dari proses tersebut.

1. *Activity* Diagram Mulai

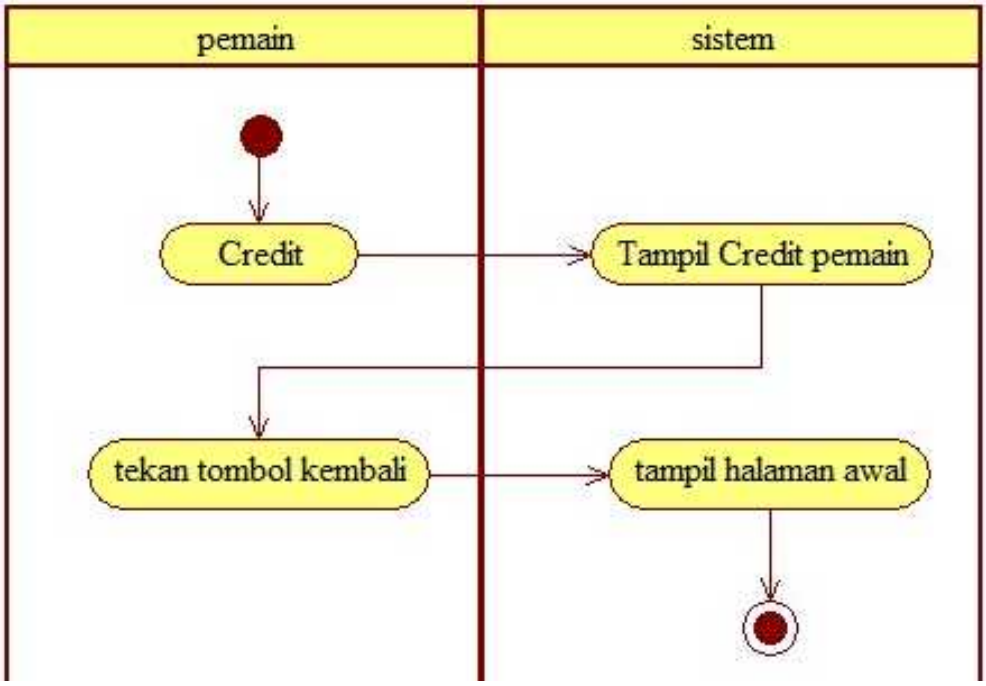
Activity diagram ini menjelaskan alur dari aktivitas mulai antar aktor dan sistem yang dijelaskan pada gambar Gambar 3.16.



Gambar 3. 16 Diagram *Activity* Mulai

2. *Activity* Diagram *Credit*

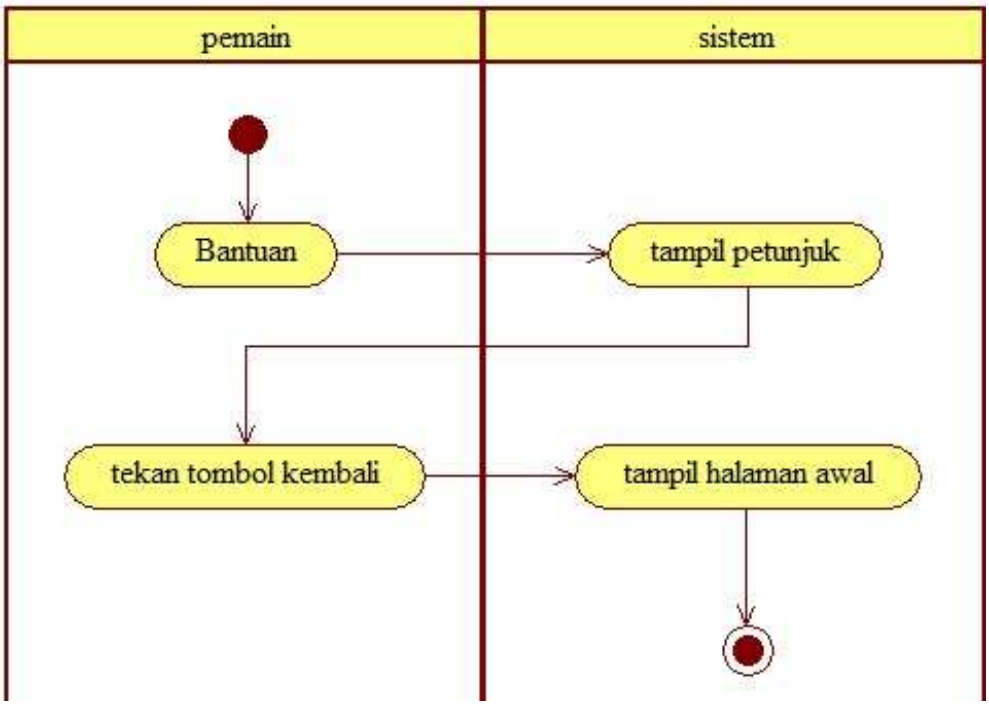
Activity diagram ini menjelaskan alur dari aktivitas *credit* antar aktor dan sistem yang dijelaskan pada gambar Gambar 3.17.



Gambar 3. 17 Diagram *Activity* Credit

3. *Activity* Diagram Bantuan

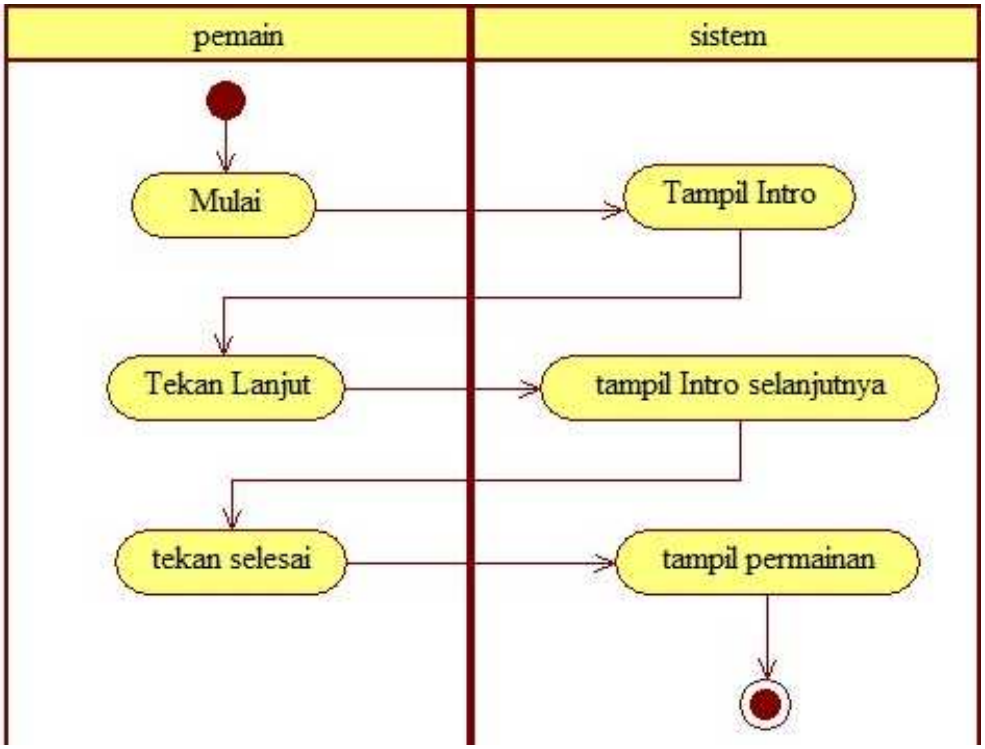
Activity diagram ini menjelaskan alur dari aktivitas Bantuan antar aktor dan sistem yang dijelaskan pada Gambar 3.18.



Gambar 3. 18 Diagram *Activity* Bantuan

4. *Activity* Diagram Intro

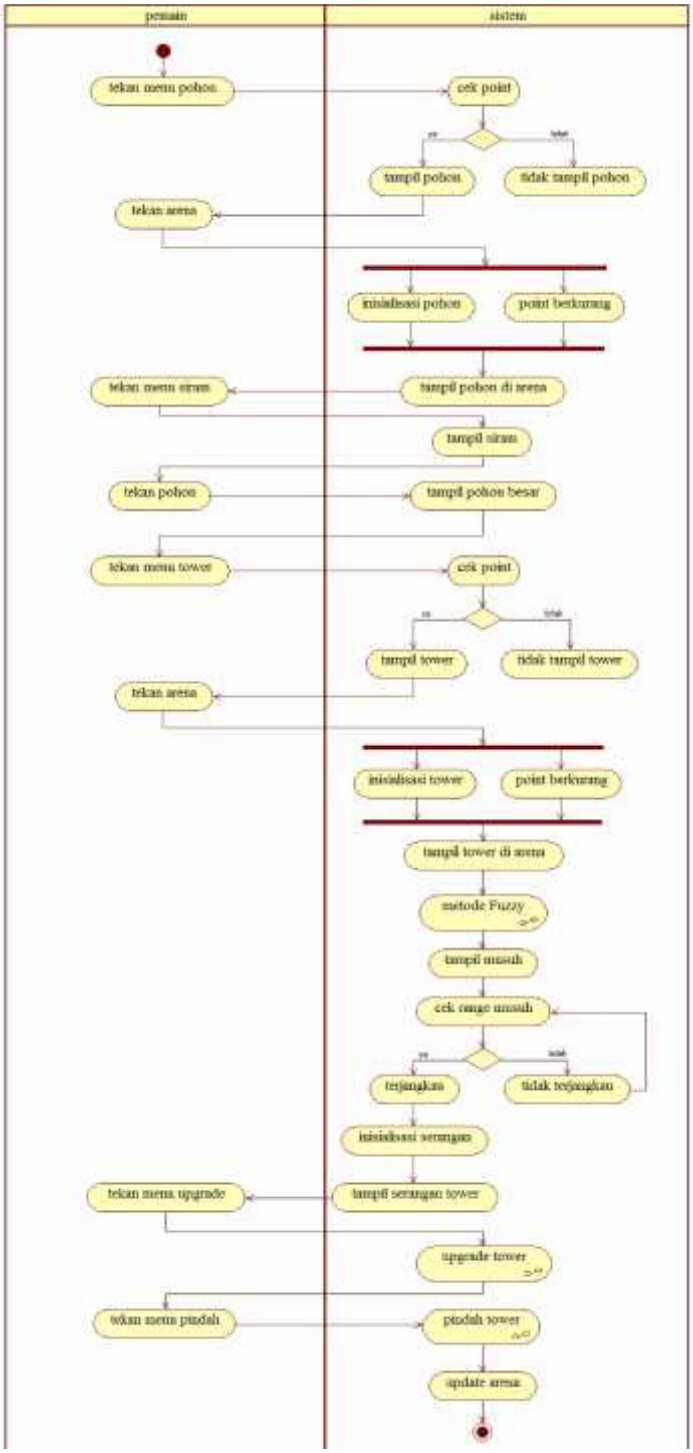
Activity diagram ini menjelaskan alur dari aktivitas Intro antar aktor dan sistem yang dijelaskan pada Gambar 3.19.



Gambar 3. 19 Diagram *Activity* Intro

5. *Activity* Diagram *level* 1

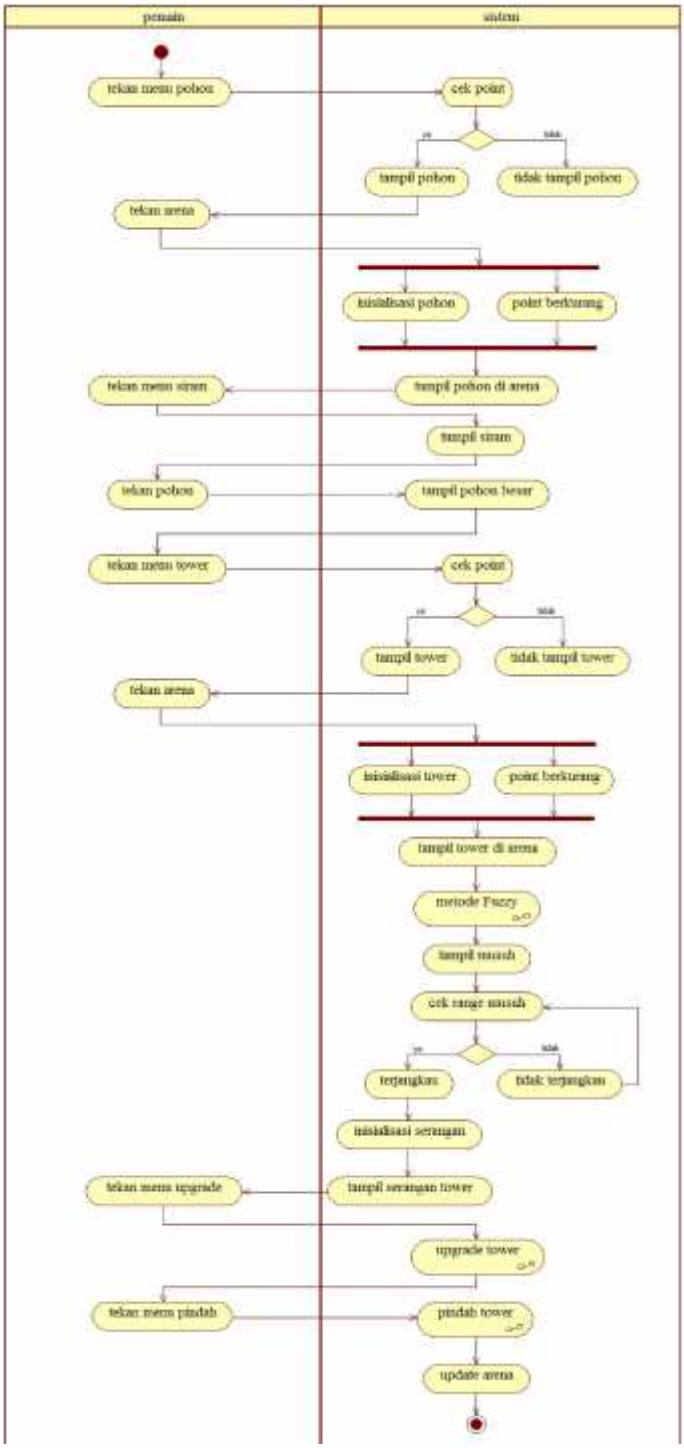
Activity diagram ini menjelaskan alur dari aktivitas *Level* 1 antar aktor dan sistem yang dijelaskan pada Gambar 3.20.



Gambar 3. 20 Diagram *Activity level* 1

6. *Activity* Diagram *level* 2

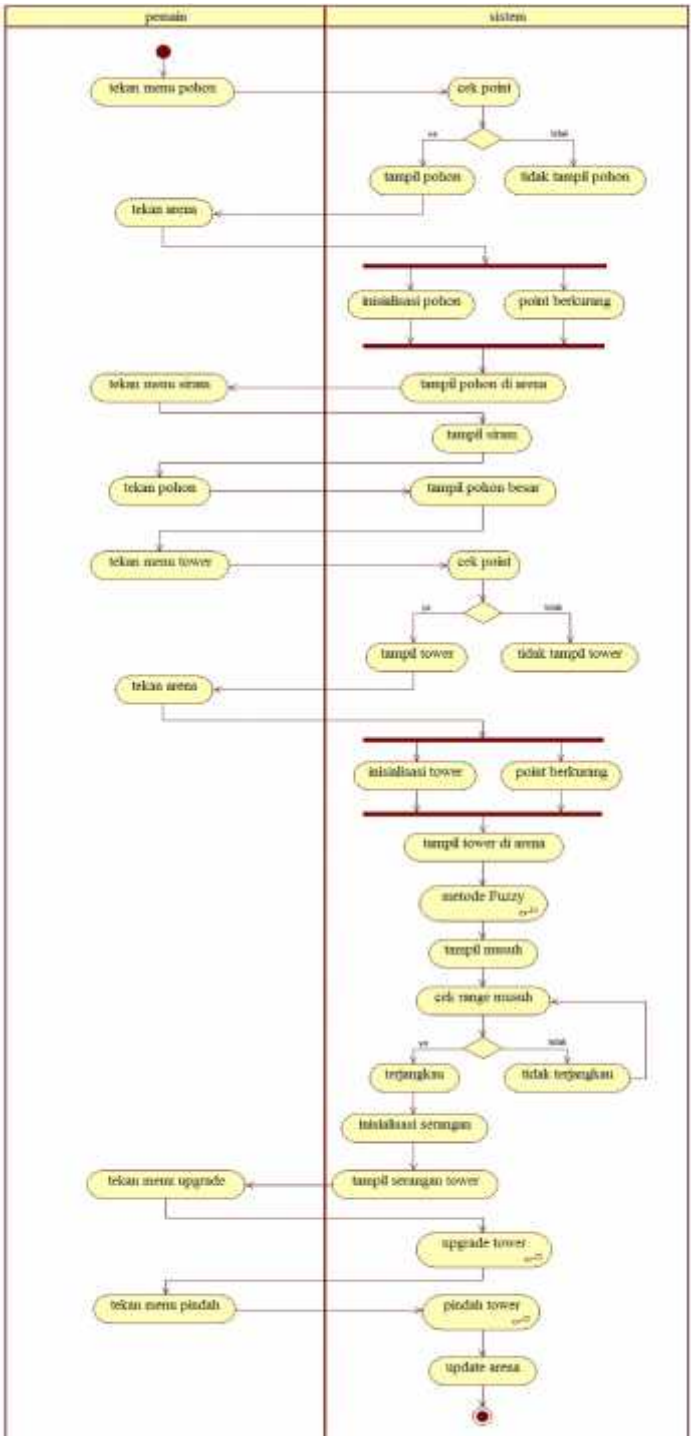
Activity diagram ini menjelaskan alur dari aktivitas *Level* 2 antar aktor dan sistem yang dijelaskan pada Gambar 3.21.



Gambar 3. 21 Diagram *Activity level* 2

7. *Activity* Diagram *level* 3

Activity diagram ini menjelaskan alur dari aktivitas *Level* 3 antar aktor dan sistem yang dijelaskan pada Gambar 3.22.



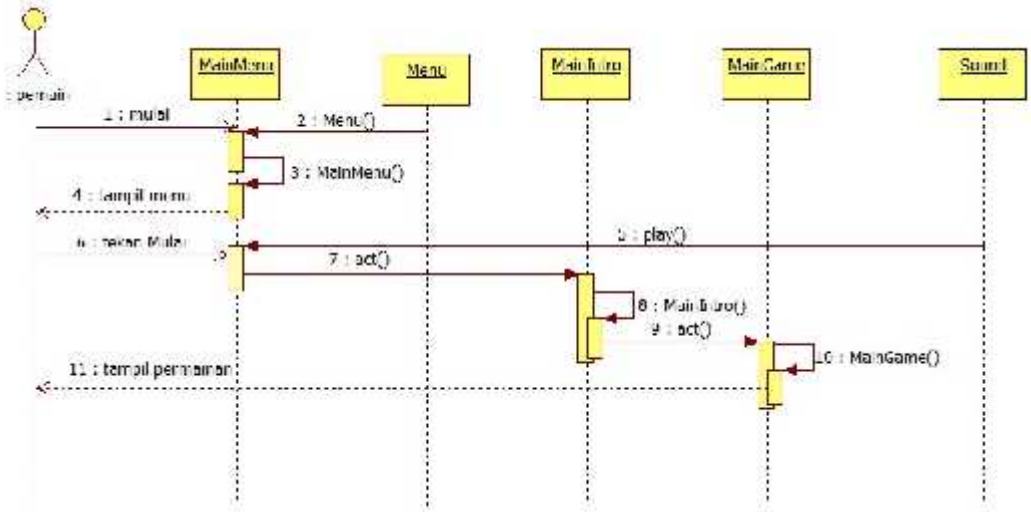
Gambar 3. 22 Diagram *Activity level* 3

3.1.5.5 *Sequence* Diagram

Sequence diagram menjelaskan kelakuan objek pada *use case* dengan mendeskripsikan berdasarkan waktu hidup dari pesan yang dikirim dan diterima antar objek.

1. ***Sequence* Diagram Mulai**

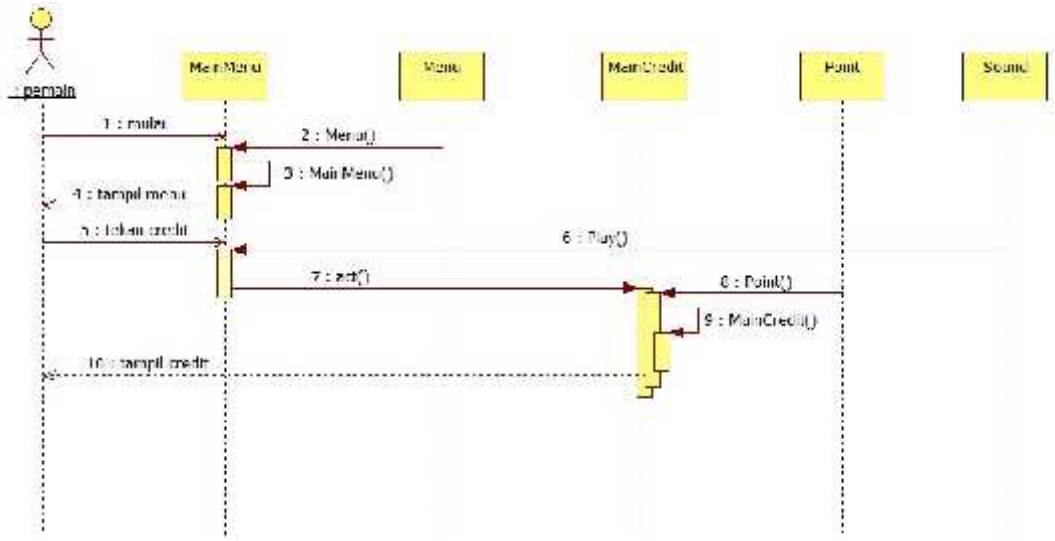
   Sequence diagram ini menjelaskan urutan interaksi dari *use case* mulai, dimana antar objek dan pesan yang dijelaskan pada gambar 3.23.



Gambar 3. 23 *Sequence* Diagram Mulai

2. ***Sequence* Diagram *Credit***

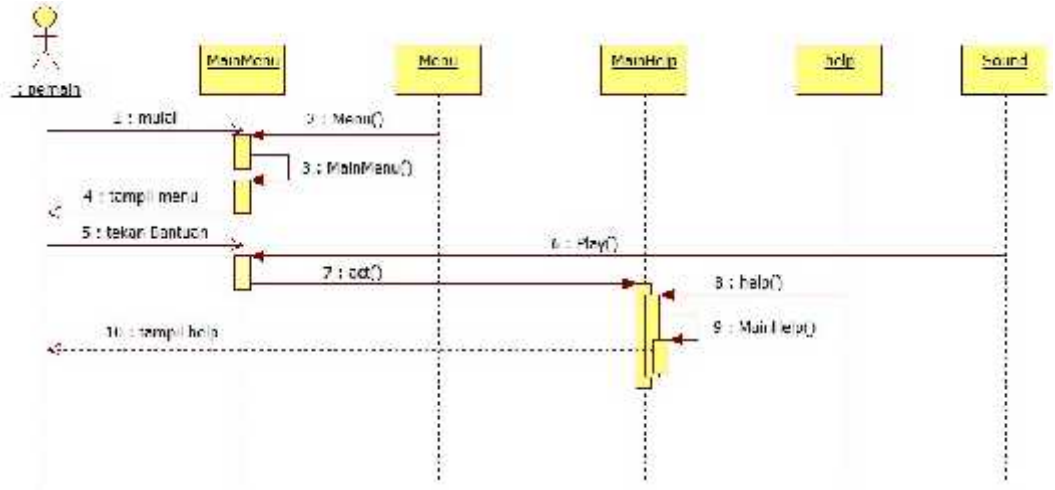
   Sequence diagram ini menjelaskan urutan interaksi dari *use case credit*, dimana antar objek dan pesan yang dijelaskan pada gambar 3.24.



Gambar 3. 24 *Sequence* Diagram *Credit*

3. *Sequence* Diagram Bantuan

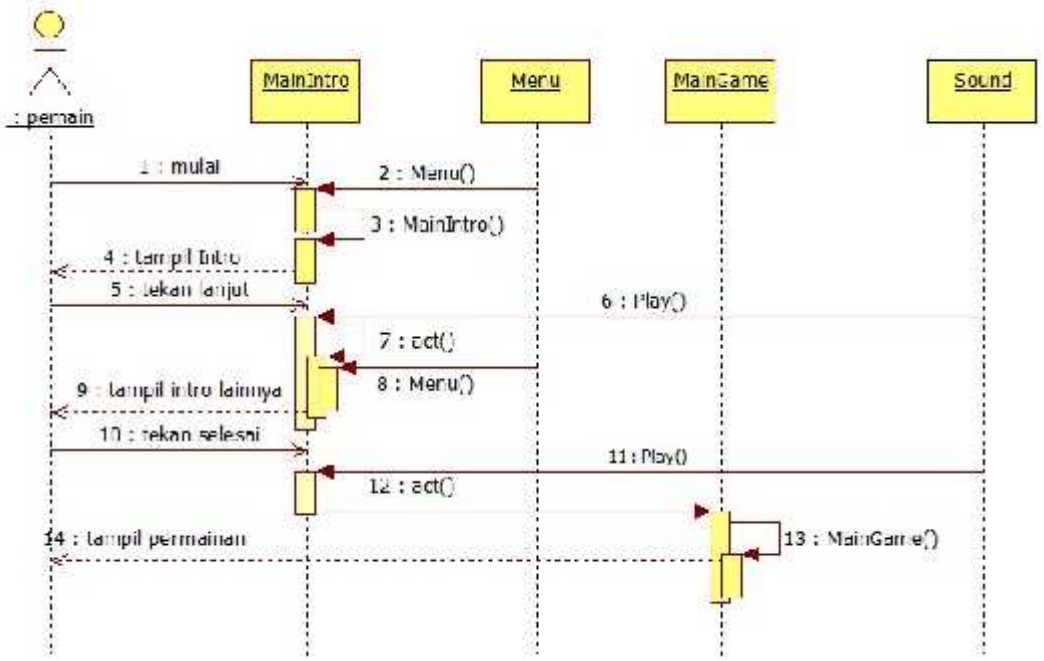
Sequence diagram ini menjelaskan urutan interaksi dari *use case* bantuan, dimana antar objek dan pesan yang dijelaskan pada gambar 3.25.



Gambar 3. 25 *Sequence* Diagram Bantuan

4. *Sequence* Diagram *Intro*

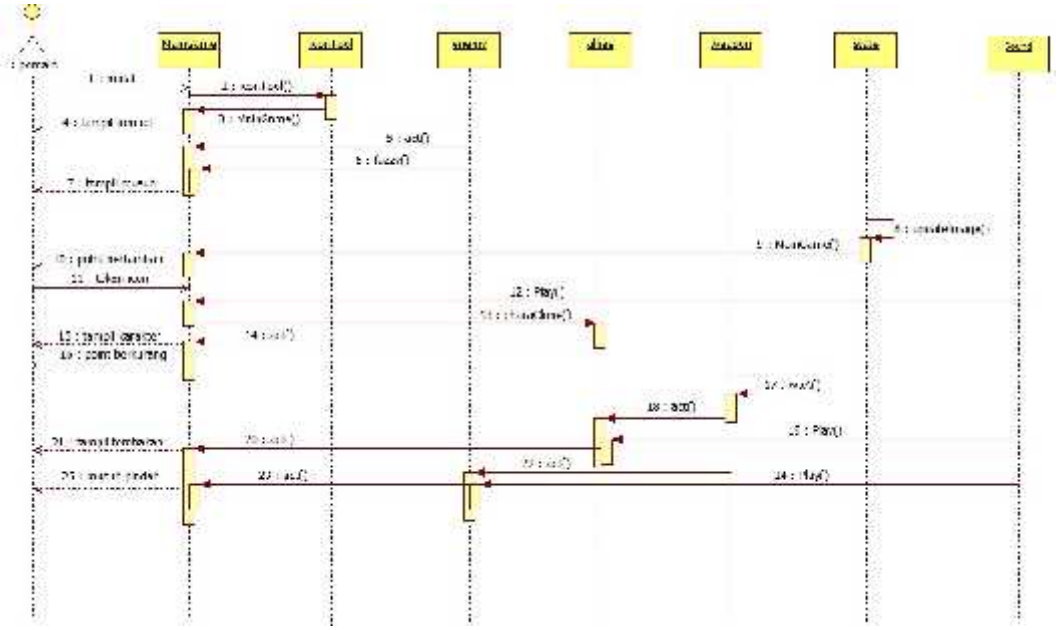
Sequence diagram ini menjelaskan urutan interaksi dari *use case intro*, dimana antar objek dan pesan yang dijelaskan pada gambar 3.26.



Gambar 3. 26 *Sequence* Diagram *Intro*

5. *Sequence* Diagram *level* 1

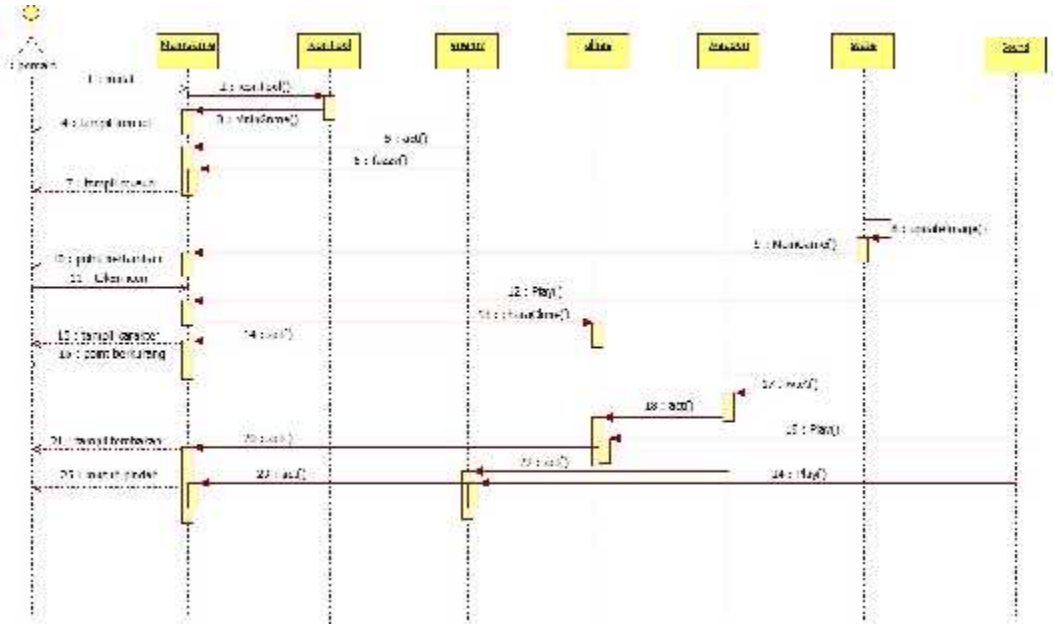
Sequence diagram ini menjelaskan urutan interaksi dari *use case level* 1, dimana antar objek dan pesan yang dijelaskan pada gambar 3.27.



Gambar 3. 27 *Sequence* Diagram *level* 1

6. *Sequence* Diagram *level* 2

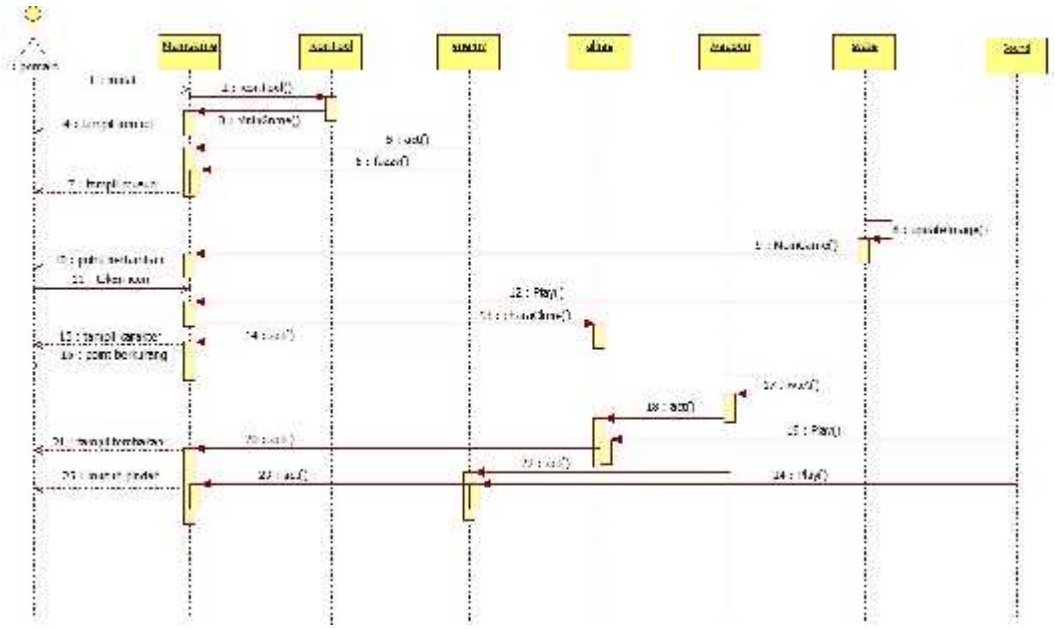
Sequence diagram ini menjelaskan urutan interaksi dari *use case level* 2, dimana antar objek dan pesan yang dijelaskan pada gambar 3.28.



Gambar 3. 28 *Sequence* Diagram *level* 2

7. *Sequence* Diagram *level* 3

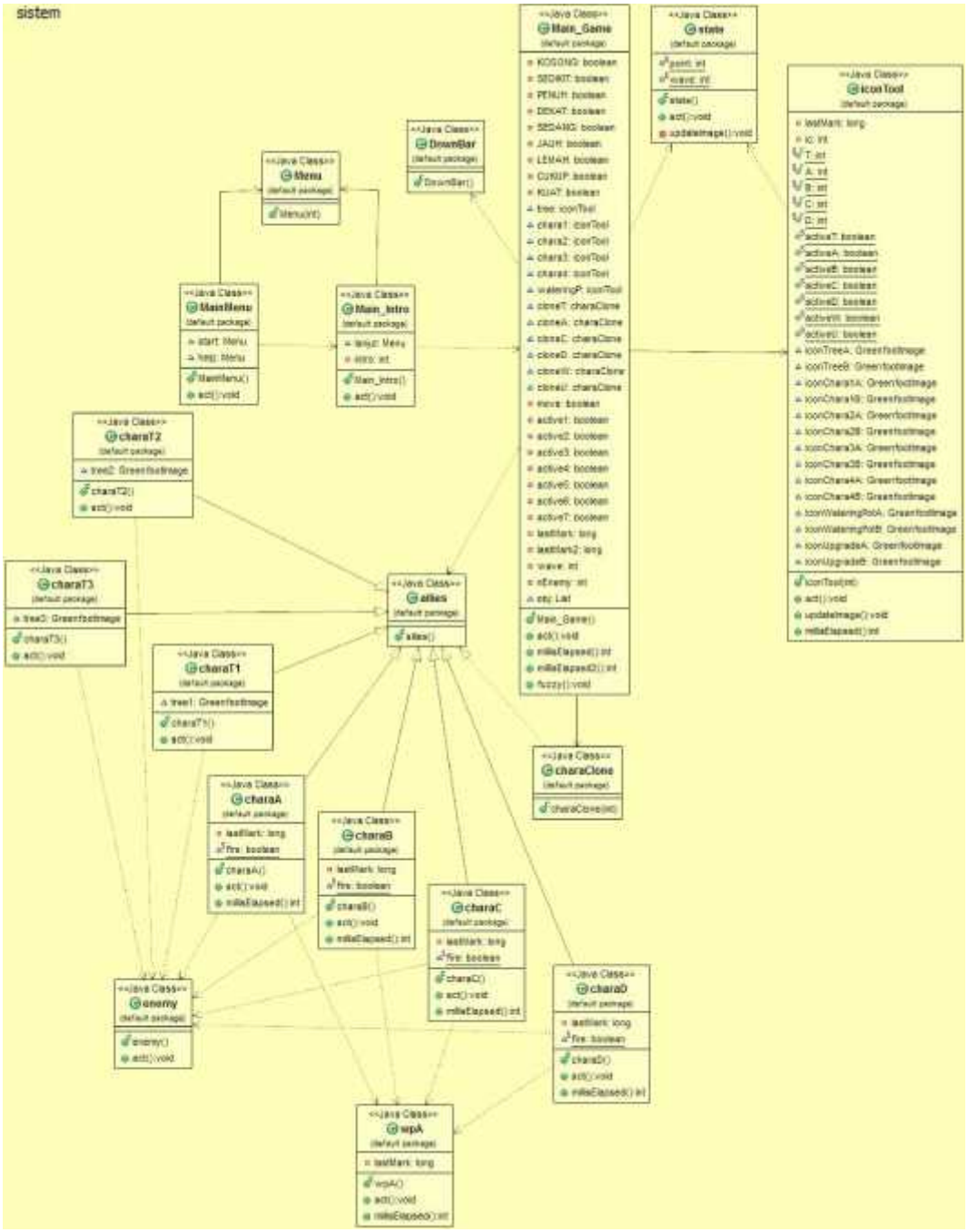
Sequence diagram ini menjelaskan urutan interaksi dari *use case level* 3, dimana antar objek dan pesan yang dijelaskan pada gambar 3.29.



Gambar 3. 29 *Sequence* Diagram *level* 3

3.1.5.6 *Class* Diagram

Class diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem. *Class* diagram memberikan gambaran sistem berupa relasi antar kelas-kelas ataupun paket-paketnya. *Class* diagram *Game Tower defense* Selamatkan Hutan Indonesia dijelaskan pada Gambar 3.30.



Gambar 3. 30 *Class* Diagram

3.2 Perancangan Sistem

Perancangan sistem merupakan tahap pengembangan suatu perangkat lunak yang dilakukan setelah melalui tahapan analisis. Dalam tahap ini digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean ke dalam suatu bahasa pemrograman.

3.2.1 Perancangan *Game*

3.2.1.1 Perancangan Karakter

Karakter dalam sebuah *game* merupakan konponen terpenting sebagai objek yang bermain di dalam sebuah *game*. Karakter yang terdapat dalam *game Tower defense* Selamatkan Hutan Indonesia ini adalah sebagai berikut :

Tabel 3. 26 Keterangan karakter

No	Nama	Gambar	Keterangan
1	Suku jawa		Ini adalah karakter yang berasal dari pulau Jawa. Karakter ini mempunyai *price point* 25, *damage* 3 ,memakai senjata kujang, dan menyerang dengan *radius* yang dekat.
2	Suku batak		Ini adalah yang berasaal dari pulau Sumatera. Karakter ini mempunyai *price point* 35, *damage 4*, memakai senjata golok, dan menyerang dengan *radius* yang dekat.
3	Suku dayak		Ini adalah yang berasaal dari pulau Kalimantan. Karakter ini mempunyai *price point* 45, *damage* 5, memakai senjata panah, dan menyerang dengan *radius* yang jauh.
4	Suku asmat		Ini adalah yang berasaal dari pulau Papua. Karakter ini mempunyai *price point* 55, *damage* 5, memakai senjata tombak, dan menyerang dengan *radius* yang jauh.

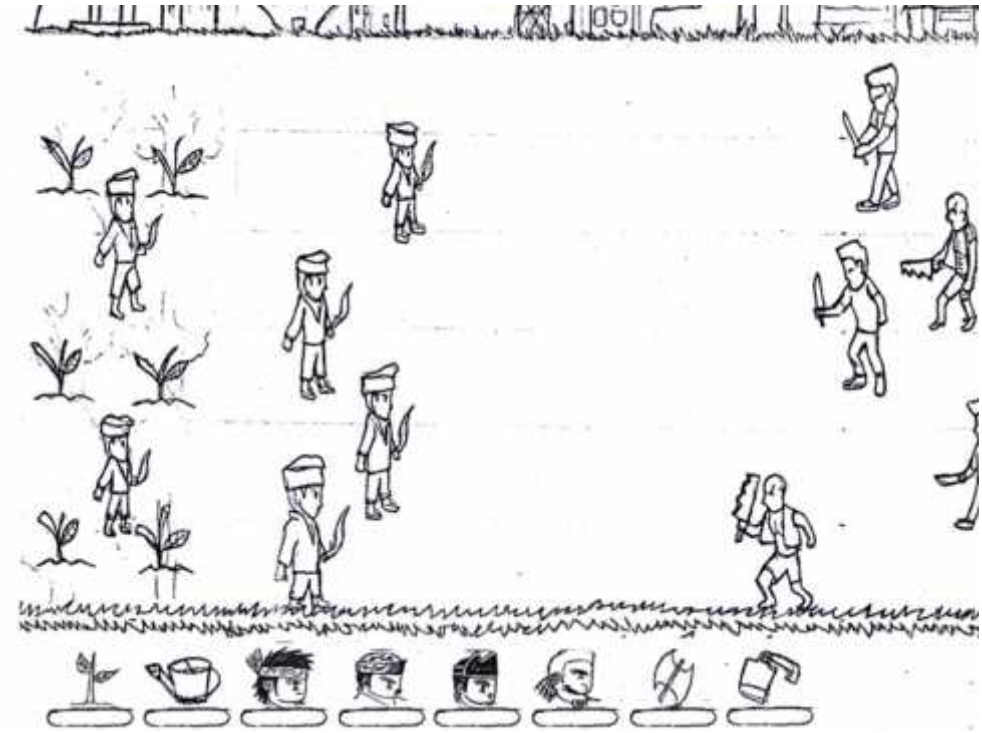
5	*Illegal Logging* Merah		Ini adalah karakter musuh dalam kelompok *Illegal Logging*. Karakter ini mempunyai *health point* 50 belum memiliki Fuzzy logic.
6	*Illegal Logging* Biru		Ini adalah karakter musuh dalam kelompok *Illegal Logging*. Karakter ini mempunyai *health point* 50, memiliki Fuzzy logic dan dapat berbelok ketika berjalan.
7	*Illegal Logging* Hijau		Ini adalah karakter musuh dalam kelompok *Illegal Logging*. Karakter ini mempunyai *health point* 35 memiliki Fuzzy logic.
6	Pohon		Ini adalah karakter Tanaman yang berfungsi untuk menambah *point* karakter dalam permainan *increase point* 5.
7	Penyiram tanaman		Ini adalah karakter yang berfungsi untuk mengubah karakter pohon dalam permainan.

3.2.1.2 Perancangan *Storyboard*

Storyboard merupakan sketsa gambar yang akan disusun berurutan sesuai naskah dan tingkat kesulitan, dengan *storyboard* kita dapat menggambarkan ide cerita kita kepada orang lain dengan mudah. Karena kita dapat mengiringi

pemahaman seseorang dengan mengikuti gambar-gambar yang ada, sehingga menghasilkan pemahaman yang sama pada ide cerita kita. Berikut merupakan *storyboard* dari *game Tower defense* Selamatkan Hutan Indonesia.

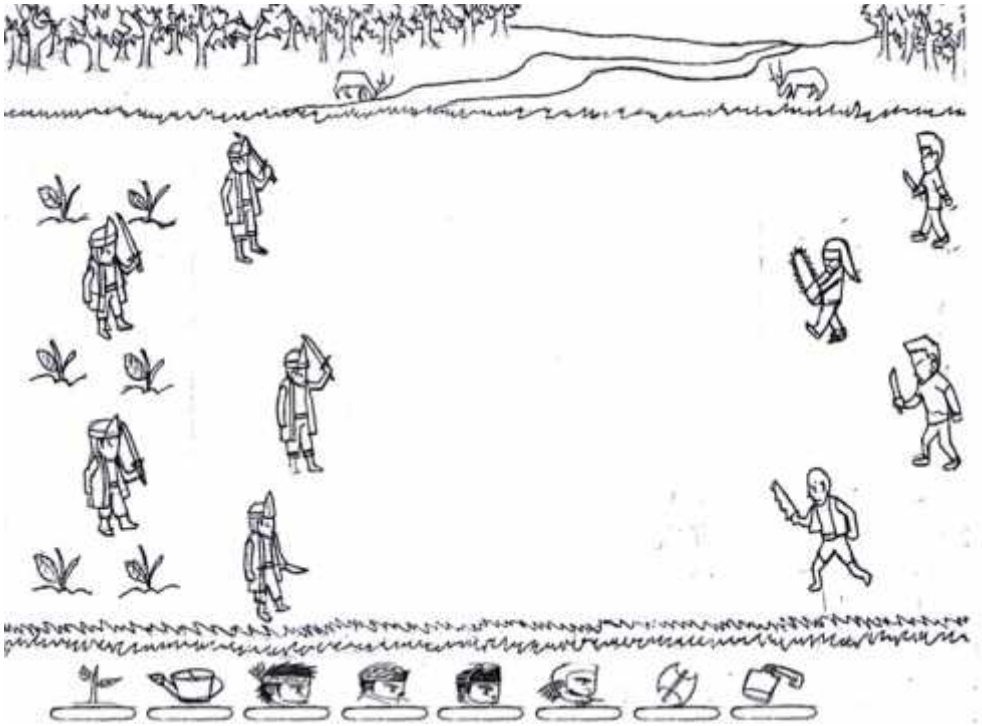
1. ***Storyboard level* 1**



Gambar 3. 31 *Storyboard level* 1

Sudah diterapkan algoritma Fuzzy. pemain akan tampil di kiri kemudian musuh akan tampil pada sebelah kanan layar. Tingkat kesulitan pada *level* 1 ini dipengaruhi oleh jumlah *wave* musuh yang harus dibendung oleh pemain yaitu 25.

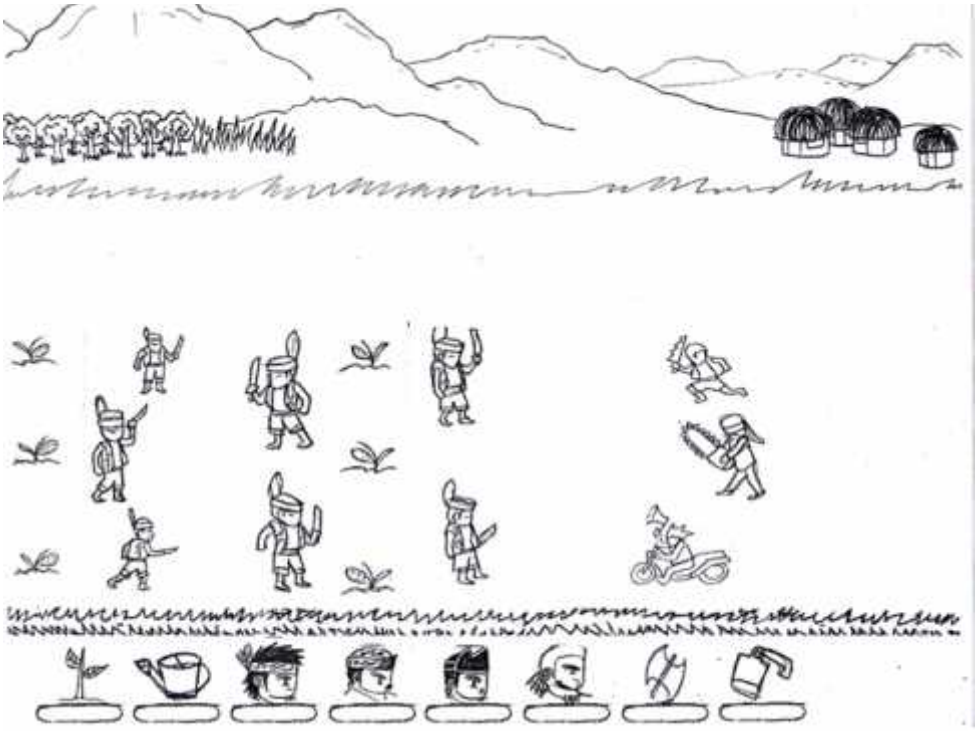
2. ***Storyboard level* 2**



Gambar 3. 32 Storyboard *level* 2

Pada *story board level* 2 juga permainan sudah diterapkan algoritma Fuzzy. pemain akan tampil di kiri kemudian musuh akan tampil pada sebelah kanan layar. Tingkat kesulitan pada *level* 2 ini dipengaruhi oleh jumlah *wave* musuh yang harus dibendung oleh pemain yaitu 25 dan musuh yang dapat menghindari blok pemain.

3. *Storyboard level* 3

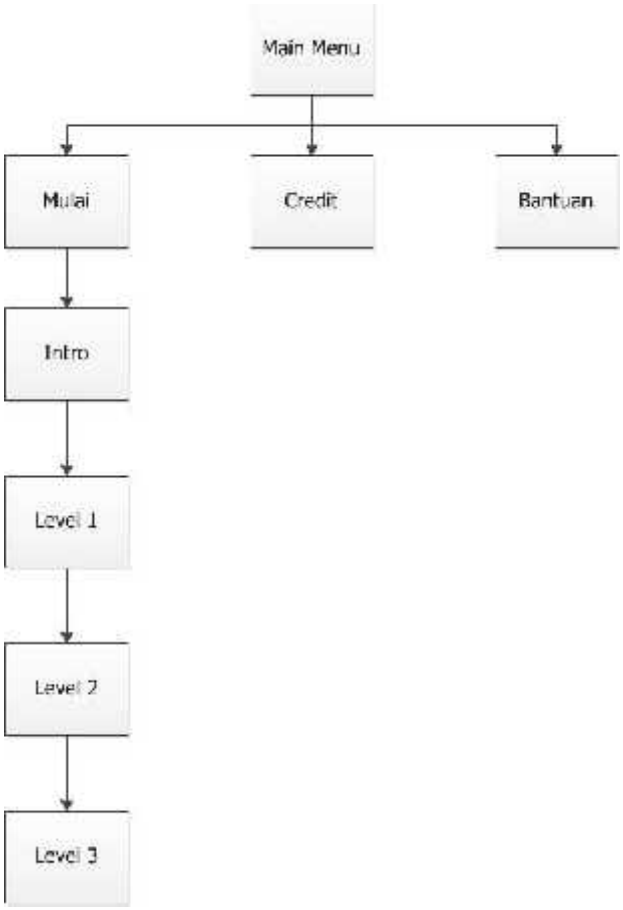


Gambar 3. 33 Storyboard *level* 3

Pada *story board level* 3 juga permainan sudah diterapkan algoritma Fuzzy. pemain akan tampil di kiri kemudian musuh akan tampil pada sebelah kanan layar. Tingkat kesulitan pada *level* 3 ini dipengaruhi oleh jumlah *wave* musuh yang harus dibendung oleh pemain yaitu 25 dan kecepatan wave yang lebih cepat.

3.2.3 Perancangan Arsitektur Menu

Untuk mempermudah dalam pembuatan sistem, diperlukan perancangan struktur menu program yang akan dibangun. Perancangan struktur menu program ini membantu dalam merancang bagian-bagian dari sistem yang sebenarnya dan untuk mengetahui bagian mana yang nantinya akan diakses terlebih dahulu setelah program selesai dibuat. Perancangan arsitektur menu pada *game Tower defense* Selamatkan Hutan Indonesia dapat dilihat pada gambar 3.34.

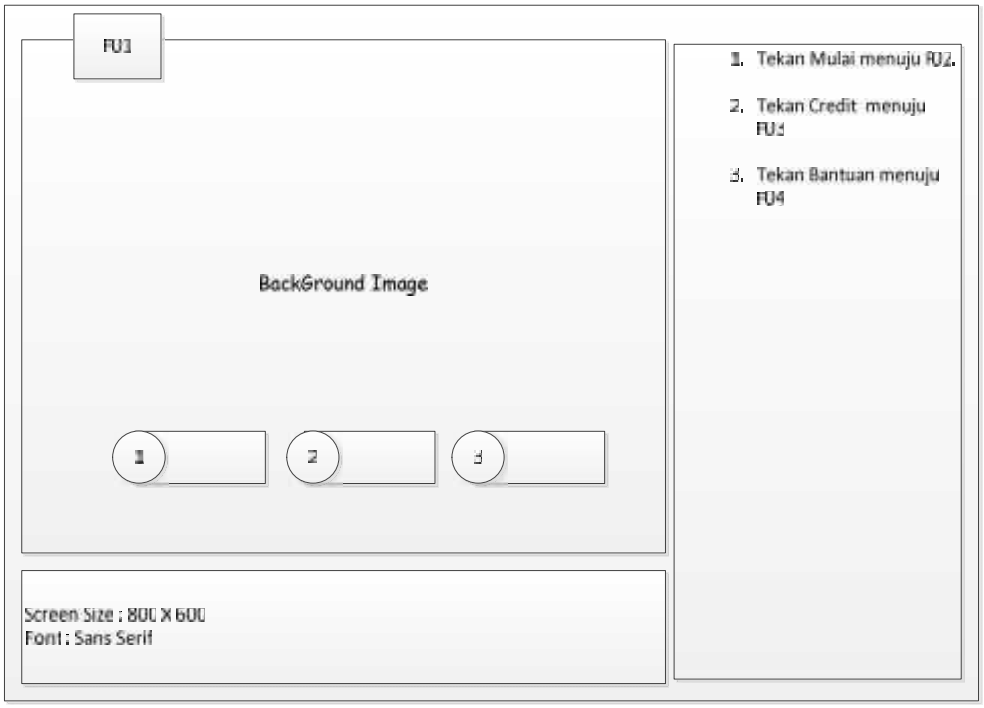


Gambar 3. 34 Struktur Menu

3.2.4 Perancangan Antar Muka

Perancangan antar muka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan mempermudah pembuatan aplikasi. Berikut ini merupakan perancangan antar muka pada *game Tower defense* Selamatkan Hutan Indonesia:

1. Perancangan Antar Muka Menu Utama



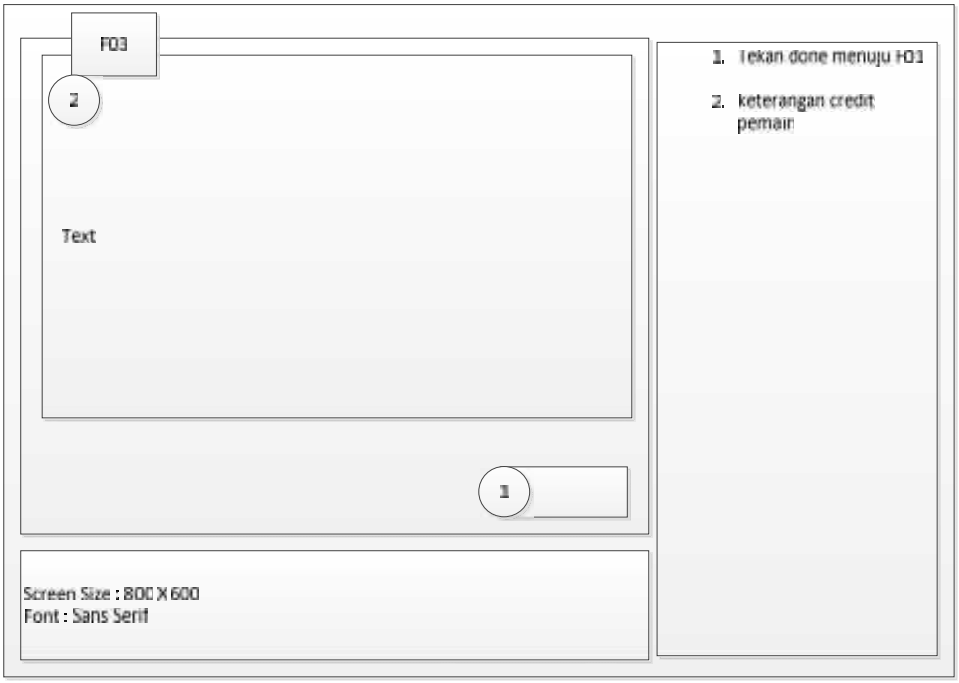
Gambar 3. 35 Perancangan Antar Muka Menu Utama

2. Perancangan Antar Muka Menu Intro



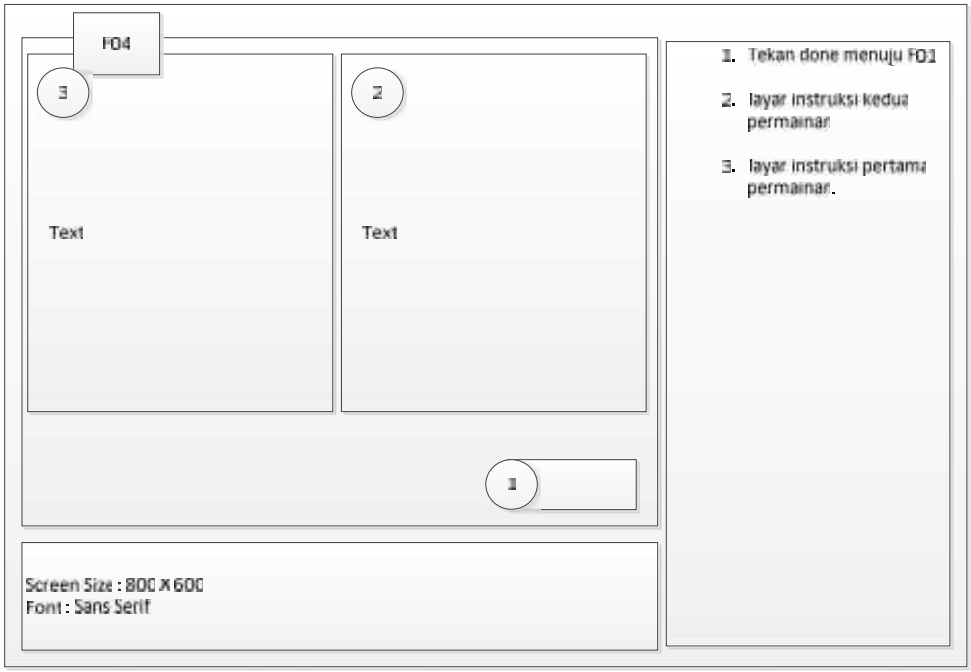
Gambar 3. 36 Perancangan Antar Muka Menu Intro

3. Perancangan Antar Muka Menu *Credit*



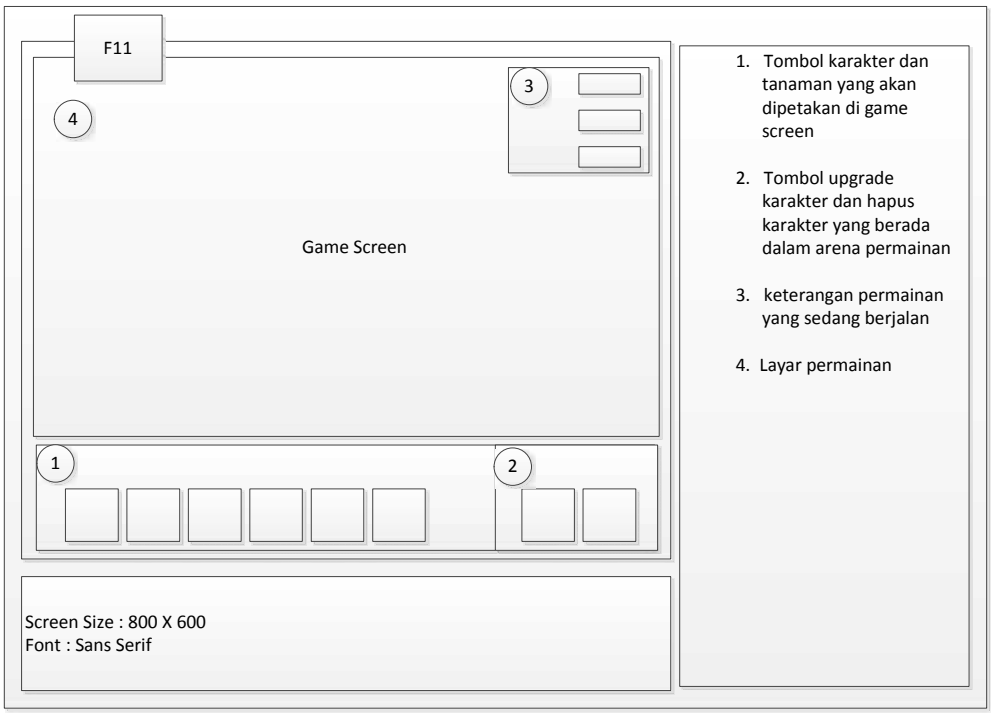
Gambar 3. 37 Perancangan Antar Muka Menu *Credit*

4. Perancangan Antar Muka Menu Bantuan



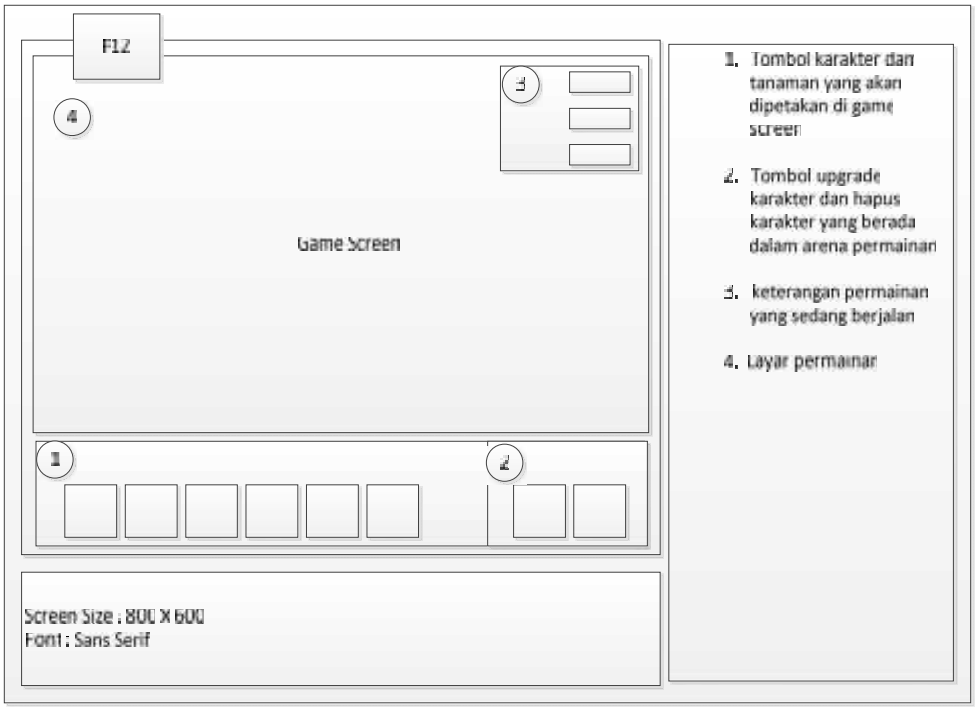
Gambar 3. 38 Perancangan Antar Muka Menu Bantuan

5. Perancangan Antar Muka Menu *Level* 1



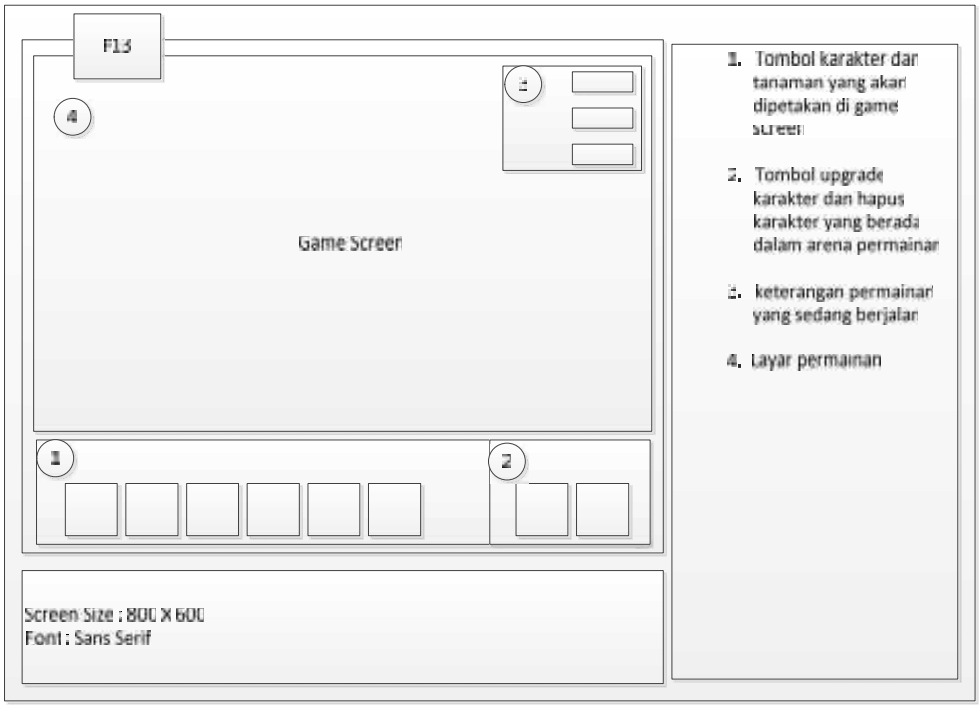
Gambar 3. 39 Perancangan Antar Muka Menu *Level* 1

6. Perancangan Antar Muka Menu *Level* 2



Gambar 3. 40 Perancangan Antar Muka Menu *Level* 2

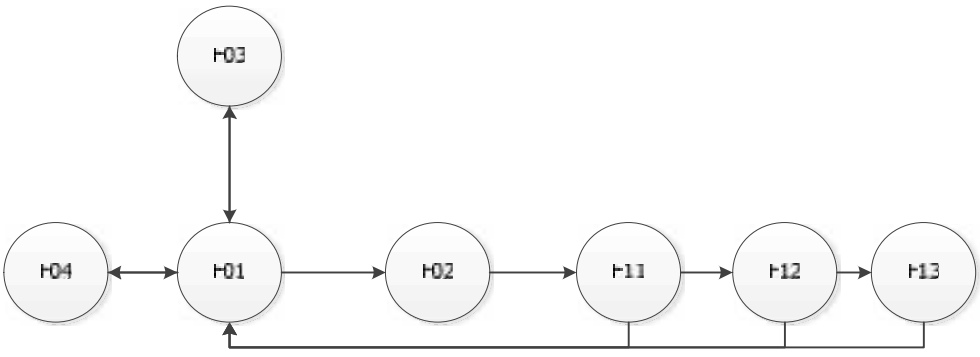
7. Perancangan Antar Muka Menu *Level* 3



Gambar 3. 41 Perancangan Antar Muka Menu *Level* 3

3.2.5 Jaringan Semantik

Jaringan semantik adalah gambaran tentang relasi dari masing- masing halaman yang bisa diakses oleh seorang pengguna. Jaringan semantik ini, digunakan sebagai panduan penggunaan menu-menu dalam sebuah aplikasi.



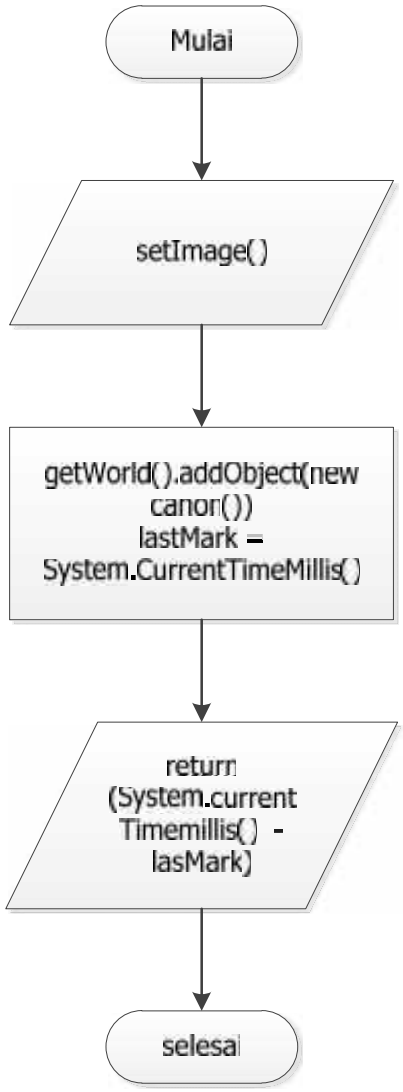
Gambar 3. 42 Jaringan Semantik

3.2.6 Perancangan *Method*

Perancangan *method* merupakan perncangan yang berfungsi untuk mendeskripsikan *method-method* yang berada di dalam aplikasi. *Method* dapat

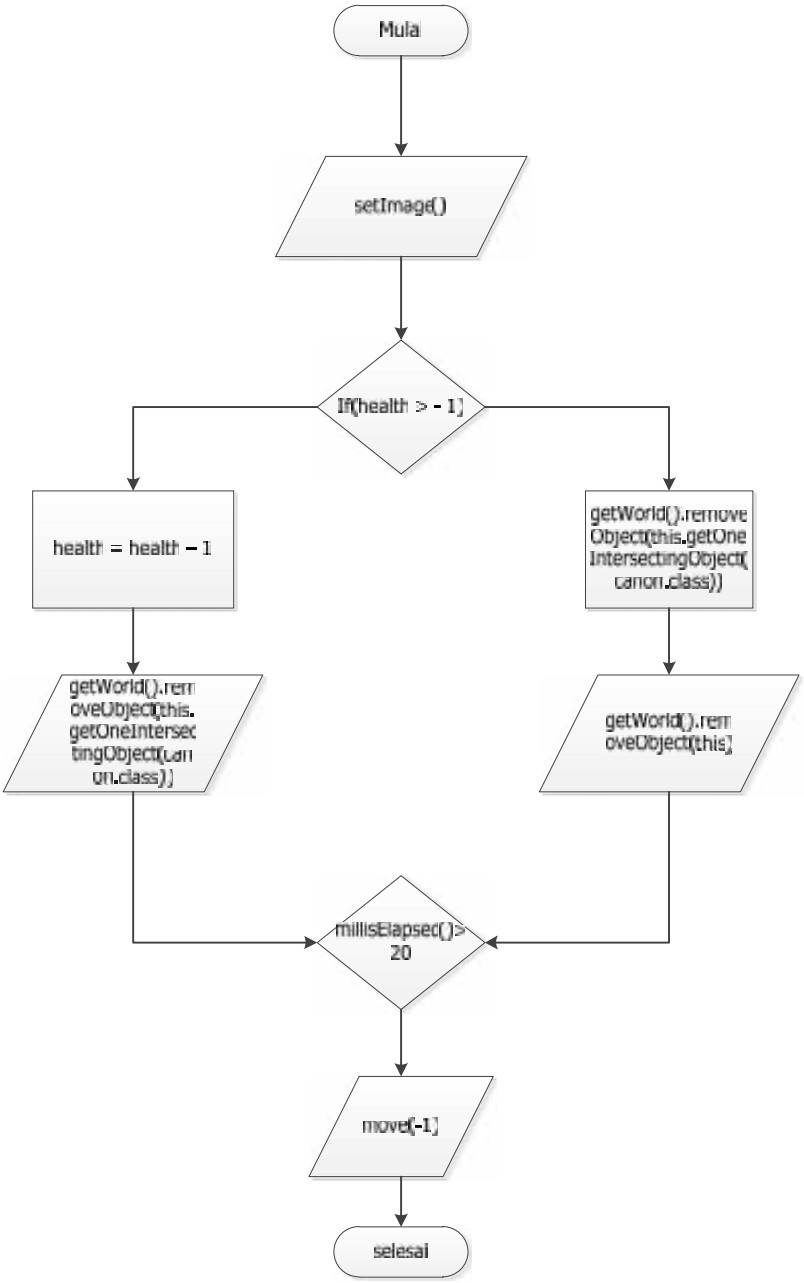
dipanggil dengan menyertakan variabel, baik hanya satu variabel, banyak variabel atau bahkan tidak ada sama sekali. Adapun *method-method* yang terdapat dalam *game Tower defense* Selamatkan Hutan Indonesia adalah sebagai berikut :

1. ***Method* Karakter**



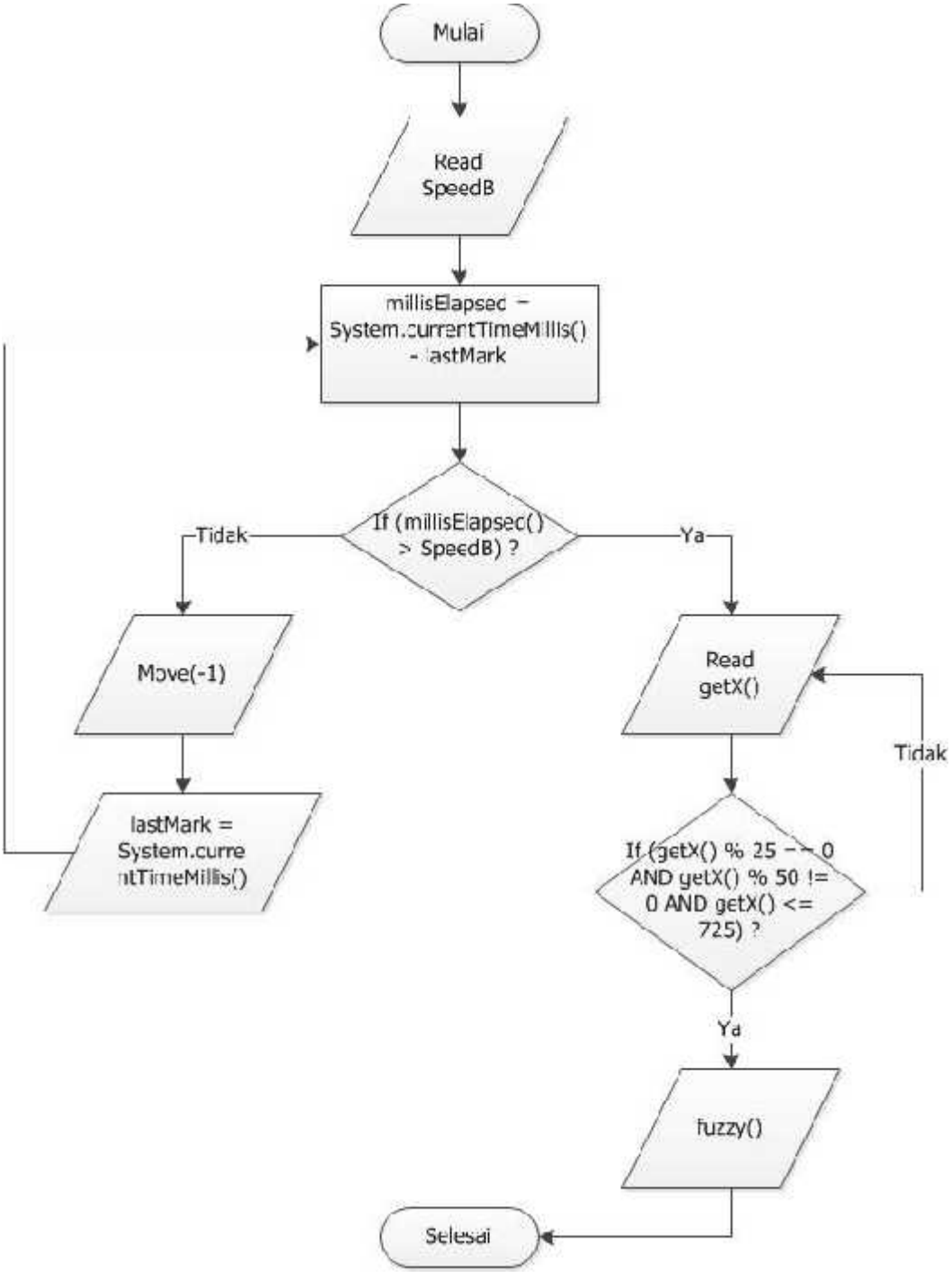
Gambar 3. 43 *Method* Karakter

2. *Method* Musuh



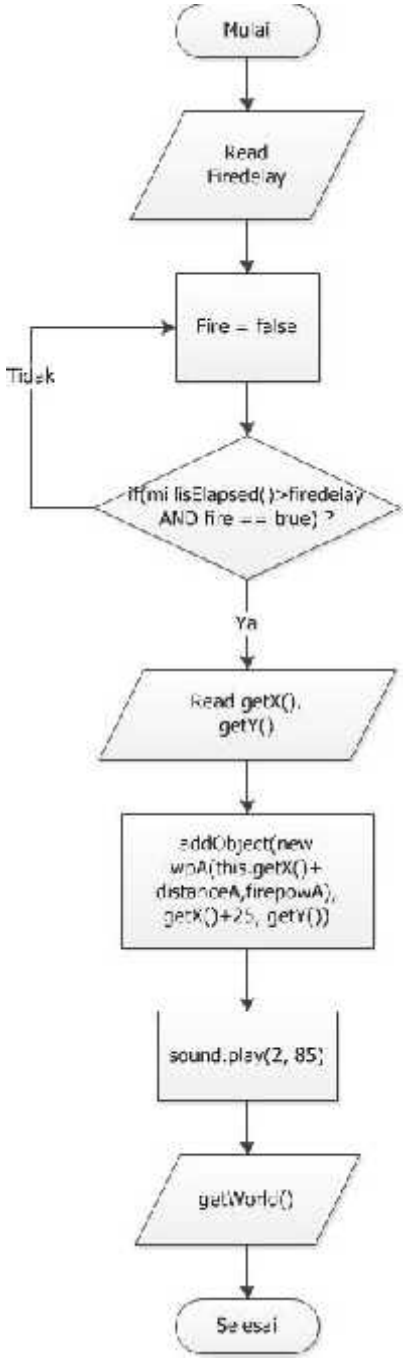
Gambar 3. 44 *Method* Musuh

3. *Method* Bergerak Jalan



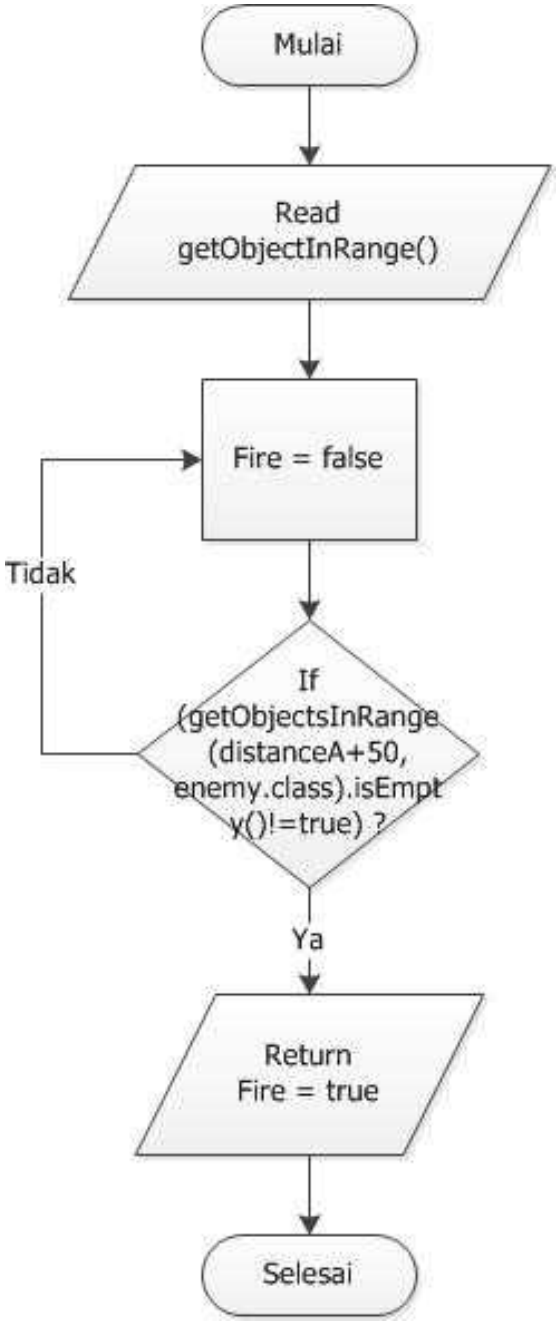
Gambar 3. 45 *Method* Bergerak Jalan

4. *Method* Tembakan



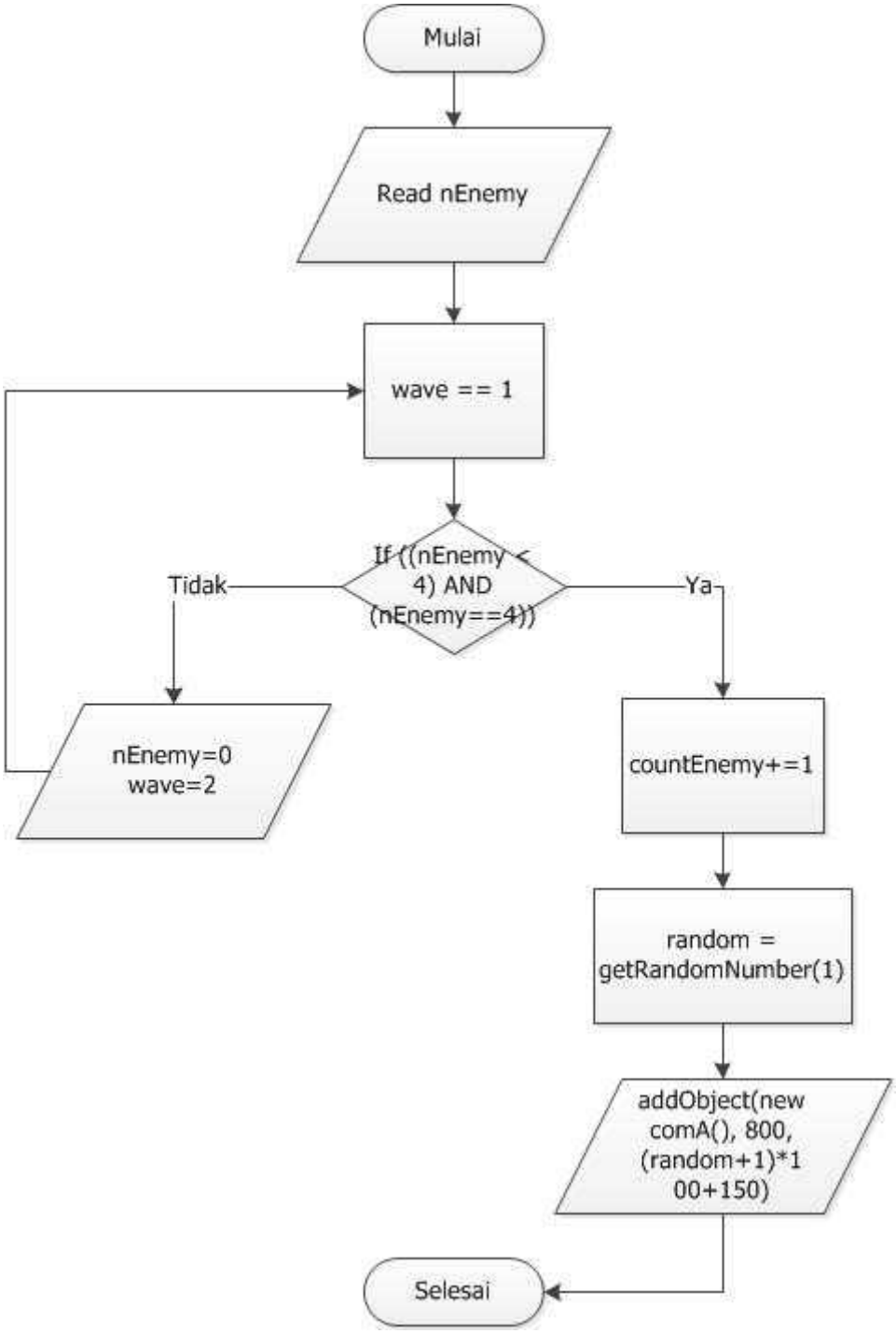
Gambar 3. 46 *Method* Tembakan

5. *Method* cek musuh



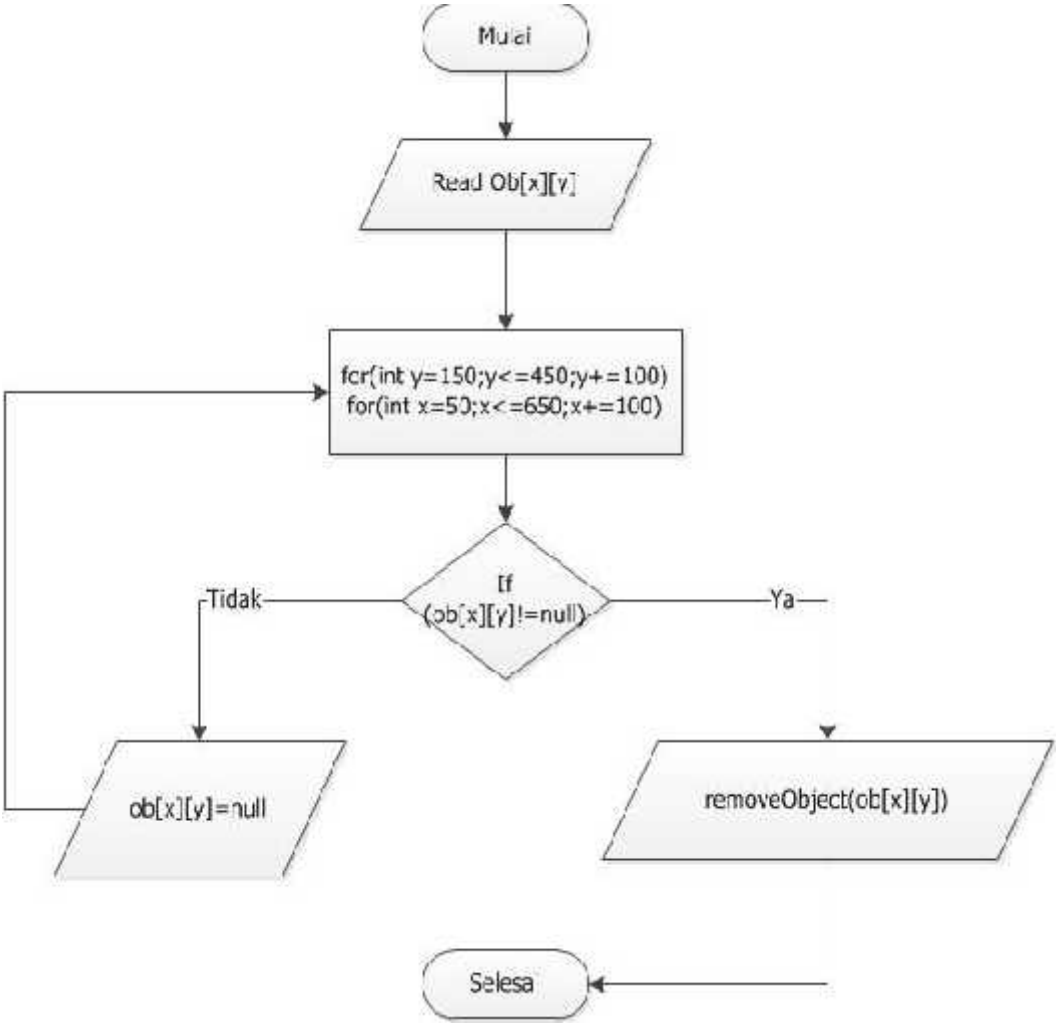
Gambar 3. 47 *Method* Cek Musuh

6. *Method Wave*



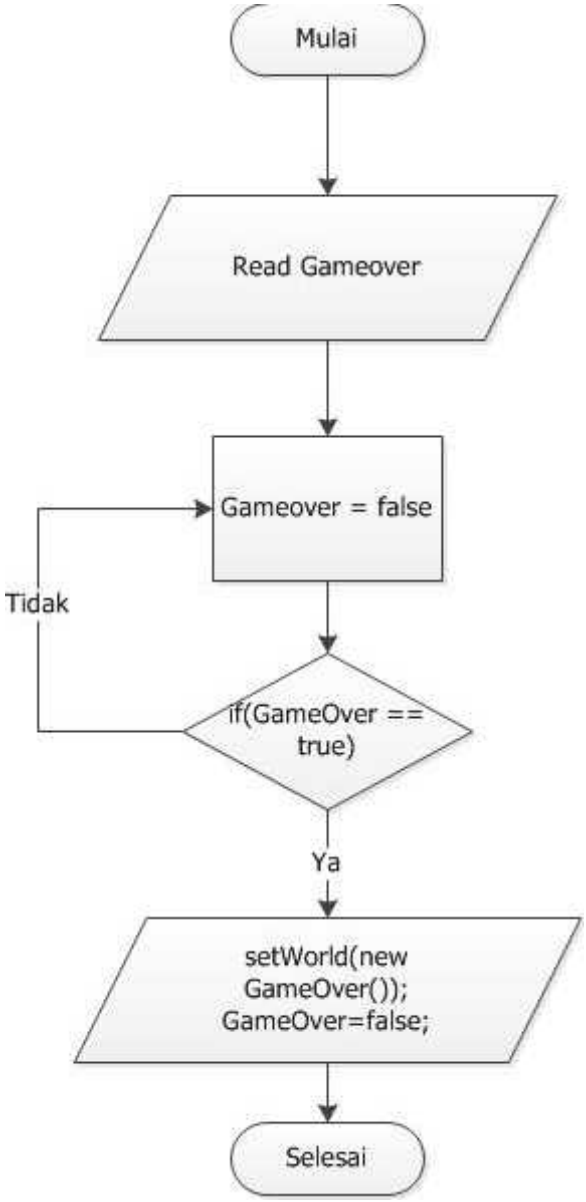
Gambar 3. 48 *Method Wave*

7. *Method* karakter mati



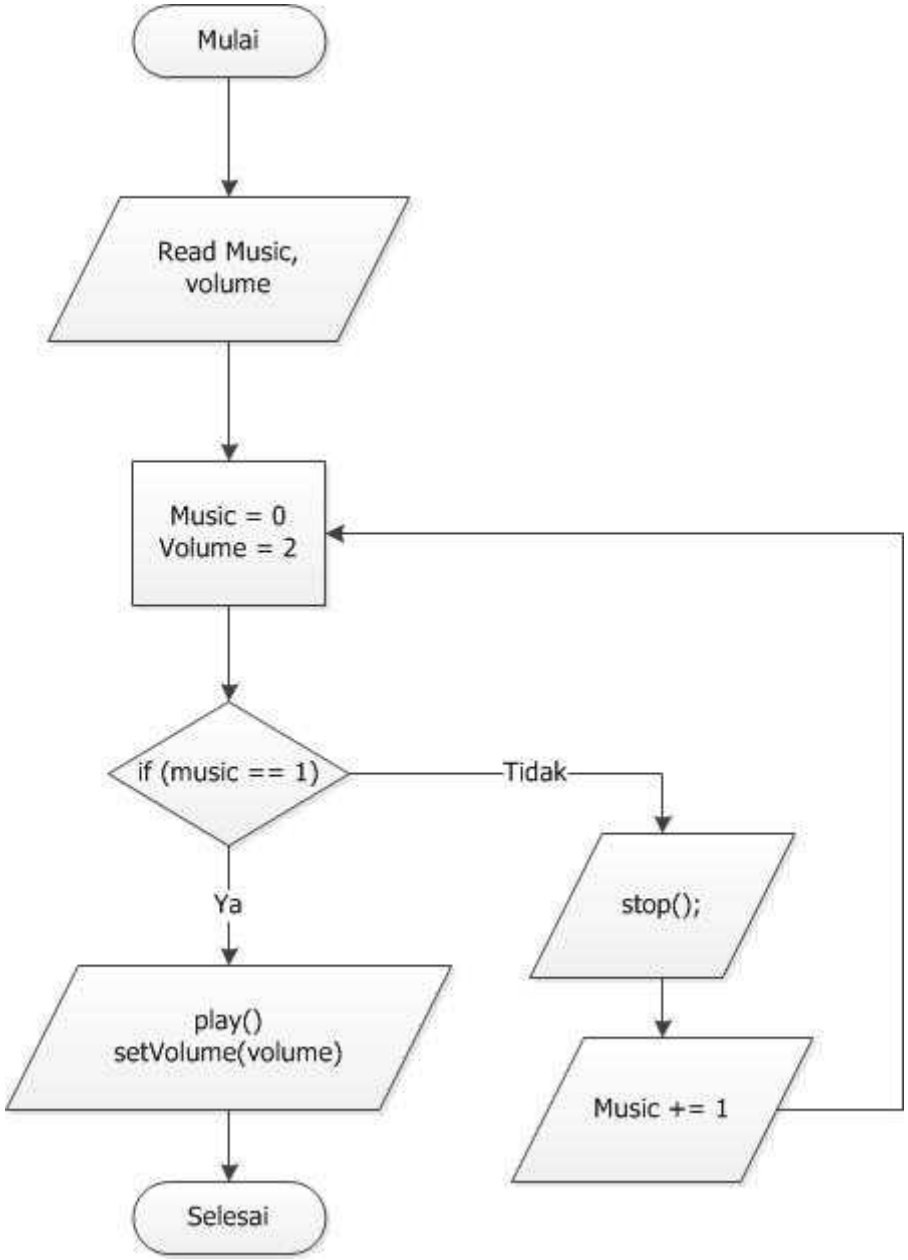
Gambar 3. 49 *Method* Karakter Mati

8. Method Game Over



Gambar 3. 50 *Method Game Over*

9. *Method* Suara



Gambar 3. 51 *Method* Suara