# BAB 3

# ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini menjelaskan mengenai analisis dan gambaran dari sistem yang ada dan bagaimana perancangan program aplikasi yang akan dibuat. Metodologi yang digunakan adalah metodologi terstruktur.

## 3.1 Analisis Sistem

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

### 3.1.1 Analisis Masalah

Berdasakan latar belakang yang sudah diuraikan di atas, maka masalahnya yaitu game tower defense yang ada saat ini belum ada yang bertema polisi hutan, selain itu pula medan pertempuran masih bersifat statis atau tidak bisa dirubah oleh pemain dan musuh hanya berjalan berdasarkan rute yang telah ditentukan. Selanjutnya perumusan masalah dapat dirumuskan dalam pertanyaan sebagai berikut :

1. Bagaimana membuat *game tower defense* dengan medan pertempuran yang dinamis.
2. Bagaimana cara membuat *game tower defense* dengan tema polisi hutan dan menyisipkan pesan edukasi.

### 3.1.1.1 Analisis *Game* Sejenis

#### 3.1.1.1.1 *Asgard Defense*




Gambar 3.1 *Asgard Defense*

Asgard defense adalah game tower defense untuk PC, game ini memiliki tujuan untuk mempertahankan rumah kita (asgard) dari serangan para mahluk berbahaya. Bila mahluk tersebut berhasil masuk ke dalam asgard maka nyawa kita akan berkurang sebanyak jumlah monster yang berhasil masuk.

Alasan kami memilih game ini karena game ini memiliki mekanisme Upgrade dan Morph Building yang kami masukan ke dalam game Tower Defense Polisi Hutan.




Gambar 3.2 Gameplay Asgard Defense

Pada awalnya di dalam game ini kita dapat membuat empat jenis tower yang dapat kita upgrade, baik secara kekuatan ataupun merubah jenis. Total ada 15 jenis tower yang dapat ditemukan pada game ini.

Kita dapat mendapatkan uang pada game ini setelah membunuh musuh-musuh yang ada dan setelah menyelesaikan satu buah aliran musuh. Dalam game ini kita mempunyai fitur bunga, jadi bila kita memiliki uang setelah satu aliran musuh maka uang yang kita punya akan mendapatkan bunga sebesar 5% dari uang yang kita miliki. Total aliran musuh yang ada dalam satu stage adalah 100 jadi bila kita berhasil menahannya maka kita akan memenangkan stage tersebut.

Game ini memiliki fitur level editor, jadi game ini memiliki jumlah stage yang tak terhingga tergantung dari setiap pemain untuk mengembangkan game ini.

#### 3.1.1.1.2 Kelebihan *Asgard Defense*

Kelebihan game *Asgard Defense* ini antara lain :

1. Permainan sangat mudah untuk dimainkan oleh anak-anak.
2. *Layout* tampilan cukup sederhana tetapi tetap menarik.
3. Kelengkapan *item* dan *gameplay* cukup bagus.
4. Memiliki mekanisme Upgrade dan Morph Building.

#### 3.1.1.1.3 Kekurangan *Asgard Defense*

Adapun kekurangan game *Asgard Defense* ini adalah :

1. NPC tidak cerdas.
2. Jalur musuh monoton (hanya lurus ke depan).

### 3.1.1.2 Gambaran Sistem

*Game* Tower Defense Polisi Hutan adalah aplikasi permainan berbasis desktop yang bergenre *Real Time Strategy Tower Defense*. Pemain diharapkan mendapatkan hiburan dari bermain *game* ini sekaligus juga membantu untuk berpikir secara strategis dalam pengambilan keputusan.

Tipe permainan yang digunakan adalah tipe *Tower Defense*, dimana pemain harus bertahan dari serangan musuh menggunakan pasukan yang ada.

Pada awal permainan hanya akan diberikan *Money* untuk modal menyewa pasukan. Pasukan tambahan dapat direkrut di kota, yang tentunya menggunakan sejumlah *Money* untuk membelinya. Faktor penentu kemenangan adalah dengan bertahan hingga jumlah musuh mencapai 0. Pemain akan kalah jika jumlah pohon yang dijaga mencapai nilai 0. Setelah memenangkan permainan,pemain akan mendapatkan pesan bahwa hutan yang telah berhasil dijaga sangat baik untuk kehidupan manusia.

Untuk dapat memainkan permainan ini, hanya dibutuhkan Sistem Operasi Windows, dan *installer game Tower Defense Polisi Hutan*. Pemain dapat menggunakan perangkat komputer bersistem operasi Windows. Permainan ini tidak membutuhkan registrasi ataupun koneksi internet.

Tujuan / Misi utama dari *aplikasi game Tower Defense Polisi Hutan* adalah :

1. Mengalahkan karakter Penebang Hutan

2. Melindungi hutan dari serangan musuh.

**3.1.1.3 Storyline**

*Game* ini bercerita tentang polisi hutan. Dimana polisi hutan tersebut harus menjaga hutan dari berbagai musuh yang datang untuk merusak hutan. Musuh yang datang untuk merusak hutan tersebut antara lain adalah penebang hutang liar. Penebang hutang liar yang akan dihadapi diantaranya adalah penebang dengan kapak, penebang dengan gergaji mesin, dan penebang dengan menggunakan alat berat.

**3.1.1.4 Gameplay**

Pada permainan polisi hutan tersebut pemain harus dapat menempatkan lokasi atau posisi pos dari polisi hutan agar dapat menghalang dan mencegan para penjahat untuk merusak hutan. Pada medan pertempuran dibuat dimamis sehingga

pemain dapat membuat jalur sendiri yang yang dianggap paling baik untuk pertahanan. Apabila pemain berhasil melindungi hutan maka akan muncul pesan edukasi manfaat dari hutan yang telah dilindung. Dan apabila pemain gagal dalam melindungi hutan makan akan muncul pesan akan bahaya nyang mengancam apabila hutan musnah.

Cara bermain *game* ini yaitu pemain harus menempatkan tower pada *map* untuk menggagalkan atau menghadang musuh yang akan merusak hutan. Apabila musuh berhasil dimusnahkan maka pemain akan mendapatkan uang yang dapat digunakan untuk membeli tower, *upgrade* tower, dan menambah tanaman. Sedangkan apabila musuh berhasil menuju hutan dan merusaknya maka jumlah tanaman atau lives dari pemain akan berkurang sesuai dengan jumlah musuh yang merusak hutan.

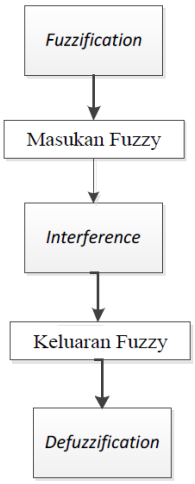
#### 3.1.1.5 Kontrol Game

Kontrol *game* pada aplikasi ini yaitu menggunakan *mouse* atau pointer untuk mengaktifkan tombol.

### 3.1.2 Analisis *Fuzzy Logic*

#### 3.1.2.1 Klasifikasi dengan Logika *fuzzy*

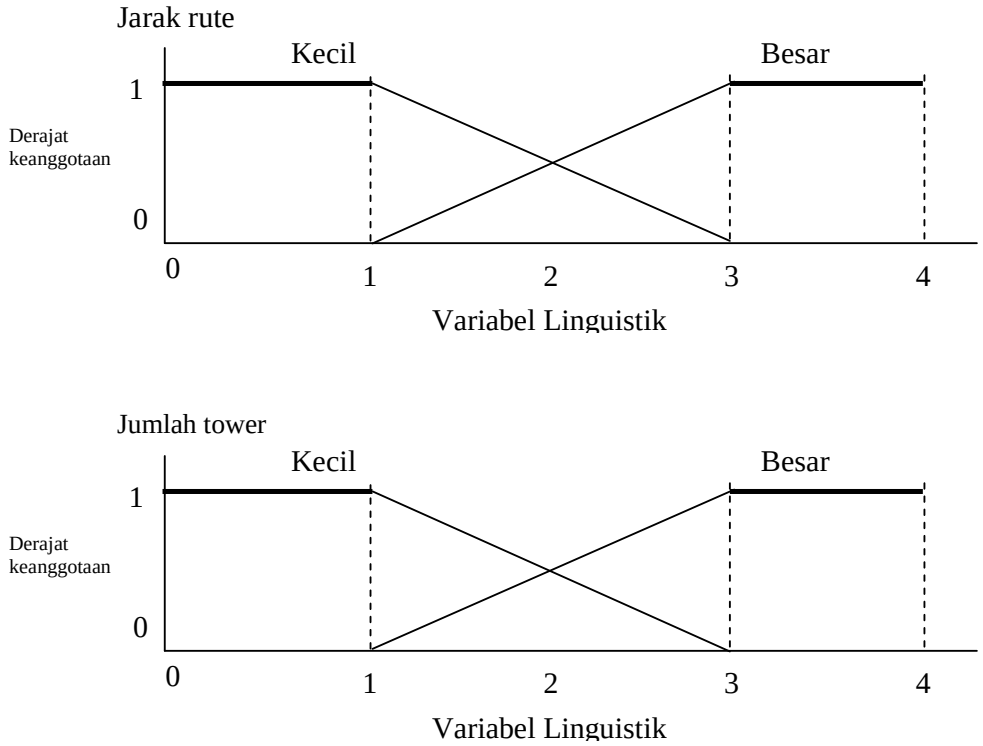
Tahap klasifikasi bertujuan untuk mengklasifikasikan kondisi rute 1, rute 2, dan rute 3. Klasifikasi yang digunakan yaitu *logika fuzzy* yang bertujuan untuk mengambil keputusan dalam pendeteksian rute teraman. Ada beberapa langkah yang digunakan yaitu *Fuzzification, Inference,* dan *Defuzzification.*




Gambar 3.3 Langkah Klasifikasi Menggunakan *Logika fuzzy*

#### 3.1.2.1.1 *Fuzzification*

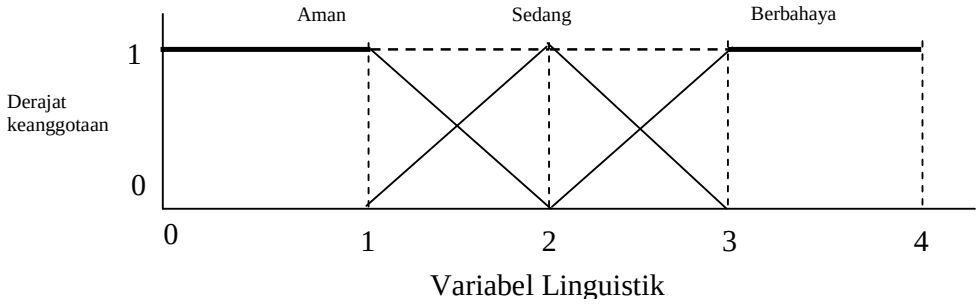
Nilai dua parameter yang diperoleh dari hasil ekstraksi ciri dikonversi ke bentuk masukan fuzzy, nilai ini dinamakan variabel linguistik, variabel linguistik digunakan untuk merepresentasikan dua masukan yaitu rute terpendek dan jumlah tower. Untuk setiap variabel linguistik memiliki dua nilai linguistik yaitu kecil dan besar. Fungsi keanggotaan yang digunakan yaitu fungsi trapesium dengan setiap parameter memiliki satu fungsi keanggotaan, sehingga dibutuhkan dua fungsi keanggotaan untuk sebuah keputusan. Berikut ini bentuk fungsi keanggotaan trapesium untuk dua variabel linguistik, dimana sumbu horizontal menunjukkan variabel linguistik dan sumbu vertikal menunjukkan derajat keanggotaan :




Gambar 3.4 Fungsi Keanggotaan Masukan Fuzzy

**3.1.2.1.2 *Inference***

*Inference* merupakan proses untuk menentukan aturan fuzzy yang akan digunakan dalam sistem. Hasil *inference* menggunakan fungsi keanggotaan trapesium dengan tiga nilai linguistik yaitu aman, sedang, dan berbahaya seperti yang terlihat pada gambar berikut :




Gambar 3.5 Fungsi Keanggotaan Keluaran Fuzzy

Model aturan fuzzy yang digunakan yaitu model mamdani, dengan menggunakan cara inferensi *clipping (alpha-cut).* Jumlah aturan yg diperlukan pada sistem ini yaitu $2^2$=4. Berikut adalah empat aturan yang digunakan :

1. **IF** jarak kecil **AND** jumlah kecil **THEN** Aman.

2. **IF** jarak kecil **AND** jumlah besar **THEN** Sedang.

3. **IF** jarak besar **AND** jumlah kecil **THEN** Sedang.

4. **IF** jarak besar **AND** jumlah besar **THEN** Berbahaya.

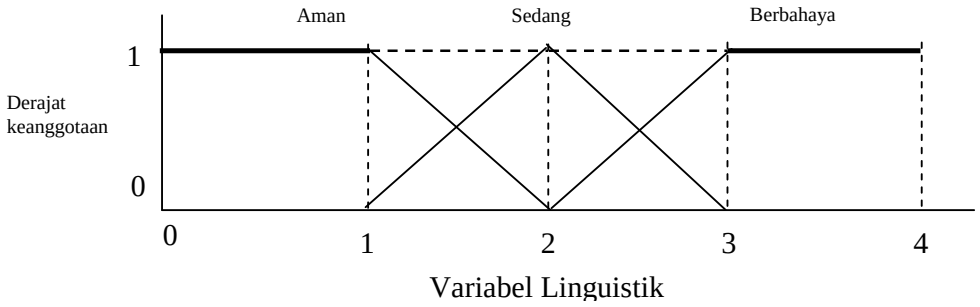
Dari aturan-aturan tersebut, maka proses *inference* yang terjadi dengan menggunakan inferensi *Clipping* dengan langkah sebagai berikut :

1. Menggunakan aturan *Conjunction* (∧) dengan memilih derajat keanggotaan minimum dari nilai-nilai linguistik yang dihubungkan oleh ∧ (AND), dan lakukan *Clipping* pada fungsi keanggotaan Trapesium.

2. Menggunakan aturan *Disjunction* (∨) dengan memilih derajat keanggotaan maksimum dari nilai-nilai linguistik yang telah diperoleh dari aturan *Conjunction* (∧).

#### 3.1.2.1.3 *Defuzzification*

Merupakan suatu proses untuk memperoleh nilai *crisp* sebagai solusi dari sistem fuzzy. Pada tugas akhir ini, menggunakan *Centroid method.*




Gambar 3.6 Grafik Defuzzifikasi

Keluaran untuk hasil deteksi kondisi rute dibuat dalam sebuah *range*. *Range* ini didapatkan dari hasil analisis terhadap dua parameter, sehingga diperoleh *range* keluaran sebagai berikut :

1. Aman (0-1)

2. Sedang (2-3)

3. Berbahaya (4)

### 3.1.3 Analisis Kebutuhan Sistem

Dari analisis kebutuhan sistem yang dilakukan maka kebutuhan sistem dapat didefinisiskan sebagai berikut:

1. Game Tower Defense Polisi Hutan ini berbasis desktop dan dapat berjalan pada system operasi Windows.

2. Aplikasi ini dapat dijadikan sebagai alternatif media pembelajaran dan pengetahuan tentang pentingnya peran Polisi Hutan dan Hutan.

### 3.1.4 Analisis dan Perancangan Terstuktur

Yang dimaksud dengan analisis adalah proses mendekomposisi atau menginvestigasi domain *problem* atau *requirements* menjadi bagian-bagian yang kita mengerti. Proses analisis lebih menekankan kepada pertanyaan seperti apakah sistem yang akan kita kerjakan, bagaimana sistem tersebut digunakan sehari-hari oleh *user*. Sedangkan perancangan adalah melakukan sintesa model *software* dari *problem* yang kita analisis, menterjemahkan domain model kepada *software* model sehingga bisa dimengerti dan dikomunikasikan kepada anggota tim lain dan juga para profesional *software* lainya.

Perancangan terstuktur berfungsi mendifinisikan dan mengilustrasikan organisasi dari sistem secara berjenjang dalam bentuk modul dan submodul. Badan terstuktur ini menggambarkan hubungan elemen data dan elemen kontrol serta hubungan antar modulnya

### 3.1.5 Analisis Kebutuhan Non-Fungsional

Analisis dan kebutuhan non-fungsional meliputi analisis dan kebutuhan perangkat lunak, analisis dan kebutuhan perangkat keras, serta analisis dan kebutuhan *user.*

#### 3.1.5.1 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk membuat *game tower defense* polisi hutan ini adalah sebagai berikut :

1. Sistem operasi Windows 7.
2. Game Maker Studio 1.0.
3. Adobe Photoshop CS4.

Berdasarkan analisis perangkat lunak di atas, rekomendasi minimum perangkat lunak untuk mengakses *game tower defense* polisi hutan ini adalah Windows XP SP 1.

### 3.1.5.2 Analisis Kebutuhan Perangkat Keras

Adapun kebutuhan perangkat keras yang diperlukan untuk membangun sistem ini adalah sebagai berikut :

1. Processor 2.8 GHz.
2. RAM 4 GB.
3. Harddisk 250 GB.
4. Monitor LED 16’.
5. VGA.

Rekomendasi spesifikasi minimum perangkat keras untuk *game tower defense* polisi hutan ini adalah sebagai berikut :

1. Processor minimal 1.8 GHz.
2. RAM 1 GB atau lebih.
3. Harddisk 100 MB.
4. Monitor CRT 14’.
5. VGA.

### 3.1.5.3 Analisis Kebutuhan *User*

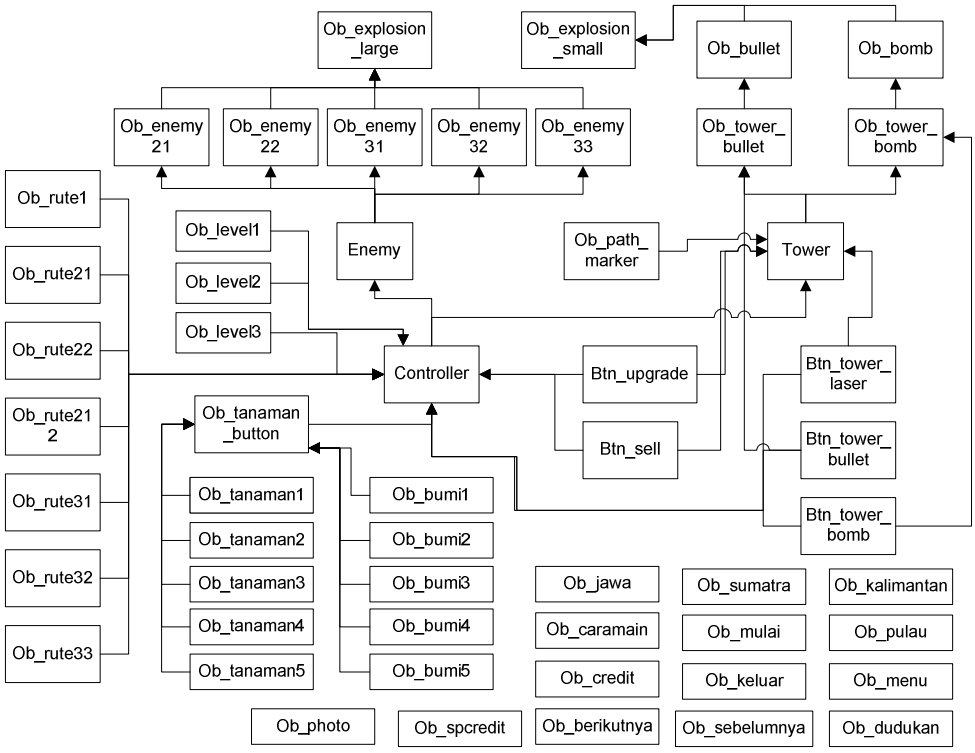
Selain dibutuhkannya perangkat lunak dan perangkat keras, *user* juga sangat dibutuhkan dalam penggunakaan aplikasi psikotes ini. Adapun spesifikasi *user* yang dibutuhkan:

1. Mengerti mengoperasionalkan komputer.
2. Dapat mengoperasikan aplikasi.
3. *User* tidak memiliki buta warna.
4. Mengerti menggunakan *mouse*

### 3.1.6 Analisis Kebetuhan Fungsional

### 3.1.7 Diagram Object

Object Diagram menggambarkan objek apa saja yang terdapat pada *Game Tower Defense* Polisi Hutan ini.




Gambar 3.7 Object Diagram

### 3.1.7.1 Detail Object

Tabel 3.1 Detail Object

| Object | Events | Actions |
|---|---|---|
| Tower | Create | Execute a piece of code:<br>a. Inisialisasi semua variabel |
| | Step | Test variable (active=0) |
| | | Start of a block |
| | | Jump to position (mouse_x,mouse_y) |
| | | Align to a grid of 16 by 16 |
| | | End of a block |
| | | Else |
| | | Execute a piece of code:<br>a. Cek enemy dalam range<br>b. Active firing jika ada enemy |
| | | Test variable (firing=1) |
| | | Start of a block |
| | | Change sprite into |

| | | |
|---|---|---|
| | | sp_tower_laser2 |
| | | End of a block |
| | | Else |
| | | Start of a block |
| | | Change sprite into sp_tower_laser |
| | | End of a block |
| | Begin step | Test expression (collision ob_path_marker) |
| | | Test expression (collision ob_tower) |
| | | Set variable coll_path to 0 |
| | | Execute a piece of code:<br>a. Check for collisions |
| | Collision with object tower | Set variable coll_path to 1 |
| | Collision with object ob_path_marker | Set variable coll_path to 1 |
| | Mouse (left pressed) | Execute a piece of code:<br>a. Pilih tower<br>b. Tower activate |

| | | |
|---|---|---|
| | Draw | Test variable (active=0) |
| | | Execute a piece of code:<br>a. Draw a ghost of the tower and its range |
| | | Else |
| | | Execute a piece of code:<br>a. Draw info tower |
| Enemy | Create | Execute a piece of code:<br>a. Inisialisasi variable<br>b. Inisialisasi path |
| | Step | Execute a piece of code:<br>a. Hapus emeny jika HP habis<br>b. Create explosion<br>c. Tambah money |
| | End of path | Execute a piece of code:<br>a. Hapus enemy |
| | Draw | Execute a piece of code:<br>a. Draw sprite<br>b. Draw healthbar |
| Ob_rute1, | Create | Execute a piece of code: |

| | | |
|---|---|---|
| ob_rute21,<br>ob_rute22,<br>ob_rute212,<br>ob_rute31,<br>ob_rute32,<br>ob_rute33, | | a. Inisialisasi variable |
| | Step | Execute a piece of code:<br>a. Kill instance<br>b. Tambah jumlah tower |
| Btn_upgrade | Step | Execute a piece of code:<br>a. Cek visible upgrade |
| | Mouse (left pressed) | Execute a piece of code:<br>a. Upgrade tower |
| Btn_sell | Step | Execute a piece of code:<br>a. Cek visibe sell |
| | Mouse (left pressed) | Execute a piece of code:<br>a. Sell tower |
| | Draw | Execute a piece of code:<br>a. Draw sprite |
| Ob_tanaman_button | Create | Execute a piece of code:<br>a. Inisialisai price variable |
| | Mouse (left pressed) | Execute a piece of code:<br>a. Cek money<br>b. Tambah lives |

| | | |
|---|---|---|
| Btn_tower_laser | Create | Execute a piece of code:<br>a. Inisialisai price variable |
| | Begin step | Execute a piece of code:<br>a. Image index = 0 |
| | Mouse (no button) | Execute a piece of code:<br>a. Image index =1 |
| | Mouse (left pressed) | Execute a piece of code:<br>a. Cek money<br>b. Other tower<br>c. Create tower |
| | Draw | Execute a piece of code:<br>a. Draw sprite<br>b. Draw price |
| Btn_tower_bullet,<br>Btn_tower_bomb<br>(Btn_tower_laser as parent) | Create | Execute a piece of code:<br>a. Inisialisai price variable |
| | Mouse (left pressed) | Execute a piece of code:<br>a. Cek money<br>b. Other tower<br>c. Create tower |
| Ob_path_marker | - | - |
| Ob_explosion_small, | Other (animation end) | Execute a piece of code: |

| | | |
|---|---|---|
| ob_explosion_large | | a. Instace destroy |
| | Draw | Execute a piece of code:<br>a. Draw sprite dengan right alpha |
| Ob_bullet, ob_bomb | Create | Execute a piece of code:<br>a. Inisialisasi damage<br>b. Sound play |
| | Collision with enemy | Execute a piece of code:<br>a. Damage enemy<br>b. Instance create<br>c. Istance destroy<br>d. Sound play |
| | Collision with ob_enemy21 | Execute a piece of code:<br>a. Damage enemy<br>b. Instance create<br>c. Istance destroy<br>d. Sound play |
| | Collision with ob_enemy22 | Execute a piece of code:<br>a. Damage enemy<br>b. Instance create<br>c. Istance destroy<br>d. Sound play |
| | Collision with | Execute a piece of code: |

| | | |
|---|---|---|
| | ob_enemy31 | a. Damage enemy<br>b. Instance create<br>c. Istance destroy<br>d. Sound play |
| | Collision with ob_enemy32 | Execute a piece of code:<br>a. Damage enemy<br>b. Instance create<br>c. Istance destroy<br>d. Sound play |
| | Collision with ob_enemy33 | Execute a piece of code:<br>a. Damage enemy<br>b. Instance create<br>c. Istance destroy<br>d. Sound play |
| | Collision with ob_rute1 | Execute a piece of code:<br>a. Damage enemy<br>b. Istance destroy |
| | Collision with ob_rute21 | Execute a piece of code:<br>a. Damage enemy<br>b. Istance destroy |
| | Collision with ob_rute22 | Execute a piece of code:<br>a. Damage enemy<br>b. Istance destroy |

| | | |
|---|---|---|
| | Collision with ob_rute212 | Execute a piece of code:<br>a. Damage enemy<br>b. Istance destroy |
| | Collision with ob_rute31 | Execute a piece of code:<br>a. Damage enemy<br>b. Istance destroy |
| | Collision with ob_rute32 | Execute a piece of code:<br>a. Damage enemy<br>b. Istance destroy |
| | Collision with ob_rute33 | Execute a piece of code:<br>a. Damage enemy<br>b. Istance destroy |
| | Other (outside room) | Execute a piece of code:<br>a. Istance destroy |
| Ob_jawa | Mouse (left pressed) | Different room (jawa) |
| | Mouse (mouse enter) | Change sprite into sp_jawa2 |
| | Mouse (mouse leave) | Change sprite into sp_jawa |
| Ob_sumatra | Mouse (left pressed) | Different room (sumatra) |
| | Mouse (mouse enter) | Change sprite into sp_sumatra2 |
| | Mouse (mouse leave) | Change sprite into sp_sumatra |

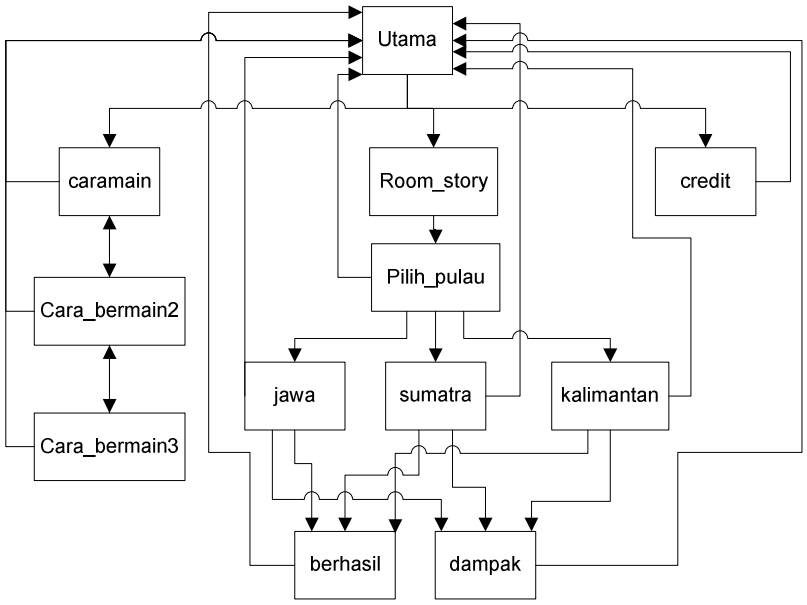
| | | |
|---|---|---|
| Ob_kalimantan | Mouse (left pressed) | Different room (Kalimantan) |
| | Mouse (mouse enter) | Change sprite into sp_kalimantan2 |
| | Mouse (mouse leave) | Change sprite into sp_kalimantan |
| Controller | Create | Execute a piece of code:<br>a. Inisialisai variable |
| | Step | Execute a piece of code:<br>a. Check whether the tower should be deselected<br>b. Execute wave system<br>c. Cek jumlah tower<br>d. Pilih rute teraman<br>e. Instance create |
| | | Execute a piece of code:<br>a. Cek lives<br>b. Instance create |
| | Other (no more lives) | Different room (dampak) |
| | Draw | Execute a piece of code:<br>a. Tampil info pulau<br>b. Tampil live<br>c. Tampil money |

| | | |
|---|---|---|
| | | d. Tampil wave |
| | Key press (space) | Execute a piece of code:<br>a. Menuju wave berikutnya |
| Ob_caramain | Mouse (left pressed) | Different room (caramain) |
| | Mouse (mouse enter) | Change sprite into sp_caramain2 |
| | Mouse (mouse leave) | Change sprite into sp_caramain |
| Ob_mulai | Mouse (left pressed) | Different room (room_story) |
| | Mouse (mouse enter) | Change sprite into sp_mulai2 |
| | Mouse (mouse leave) | Change sprite into sp_mulai |
| Ob_pulau | Mouse (left pressed) | Different room (pilih_pulau) |
| Ob_credit | Mouse (left pressed) | Different room (credit) |
| | Mouse (mouse enter) | Change sprite into sp_credit2 |
| | Mouse (mouse leave) | Change sprite into sp_credit |
| Ob_keluar | Mouse (left pressed) | End the game |
| | Mouse (mouse enter) | Change sprite into sp_keluar2 |
| | Mouse (mouse leave) | Change sprite into sp_keluar |
| Ob_menu | Mouse (left pressed) | Different room (utama) |

| | | |
|---|---|---|
| | Mouse (mouse enter) | Change sprite into sp_menu2 |
| | Mouse (mouse leave) | Change sprite into sp_menu |
| Ob_berikutnya | Mouse (left pressed) | Next room |
| Ob_sebelumnya | Mouse (left pressed) | Previous room |
| Ob_tanaman1-5 | - | - |
| Ob_level1-3 | End step | Execute a piece of code:<br>a. Input nilai level ke controller |
| Ob_dudukan | - | - |
| Ob_bumi1-5 | - | - |
| Ob_photo | - | - |
| Ob_spcredit | - | - |

### 3.1.8 Diagram Room

Room Diagram menggambarkan room apa saja yang terdapat pada *Game Tower Defense* Polisi Hutan ini.




Gambar 3.8 *Room Diagram*

#### 3.1.8.1 *Detail Room*

Tabel 3.2 *Detail Room*

| Room | Objects |
|---|---|
| Utama | Ob_mulai |
| | Ob_caramain |
| | Ob_credit |
| | Ob_keluar |

| | |
|---|---|
| Caramain | Ob_berikutnya |
| | Ob_menu |
| Cara_bermain2 | Ob_berikutnya |
| | Ob_menu |
| | Ob_sebelumnya |
| Cara_bermain3 | Ob_menu |
| | Ob_sebelumnya |
| Room_story | Ob_pulau |
| Jawa | Controller |
| | Ob_rute1, ob_rute21, ob_rute22, ob_rute212, ob_rute31, ob_rute32, ob_rute33, |
| | Btn_upgrade |
| | Btn_sell |
| | Ob_tanaman_button |
| | Btn_tower_laser |
| | Btn_tower_bullet, |
| | Btn_tower_bomb |
| | Ob_path_marker |
| | Ob_menu |

| | |
|---|---|
| | Ob_tanaman1-5 |
| | Ob_level1-3 |
| | Ob_dudukan |
| | Ob_bumi1-5 |
| Sumatra | Controller |
| | Ob_rute1, ob_rute21, ob_rute22, ob_rute212, ob_rute31, ob_rute32, ob_rute33, |
| | Btn_upgrade |
| | Btn_sell |
| | Ob_tanaman_button |
| | Btn_tower_laser |
| | Btn_tower_bullet, |
| | Btn_tower_bomb |
| | Ob_path_marker |
| | Ob_menu |
| | Ob_tanaman1-5 |
| | Ob_level1-3 |
| | Ob_dudukan |
| | Ob_bumi1-5 |

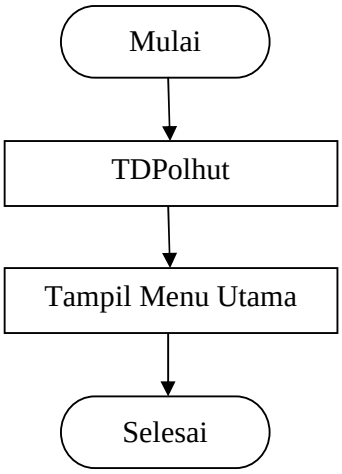
| | |
|---|---|
| Kalimantan | Controller |
| | Ob_rute1, ob_rute21, ob_rute22, ob_rute212, ob_rute31, ob_rute32, ob_rute33, |
| | Btn_upgrade |
| | Btn_sell |
| | Ob_tanaman_button |
| | Btn_tower_laser |
| | Btn_tower_bullet, |
| | Btn_tower_bomb |
| | Ob_path_marker |
| | Ob_menu |
| | Ob_tanaman1-5 |
| | Ob_level1-3 |
| | Ob_dudukan |
| | Ob_bumi1-5 |
| Dampak | Ob_menu |
| Pilh_pulau | Ob_jawa |
| | Ob_sumatra |
| | Ob_kalimantan |

| | |
|---|---|
| | Ob_menu |
| Credit | Ob_menu |
| berhasil | Ob_menu |

### 3.1.9 *Flowchart Game Tower Defense* Polisi Hutan

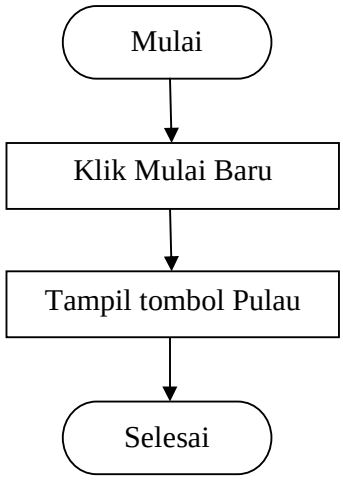
*Flowchart* yang dirancang berdasarkan kebutuhan yang ada pada *Game Tower Defense* Polisi Hutan ini adalah sebagai berikut.

1. Flowchart Tampil Menu Awal



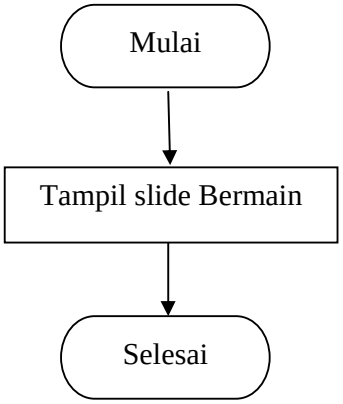

Gambar 3.9 *Flowchart* Tampil Menu Awal

2. *Flowchart* Tampil Pilih Pulau



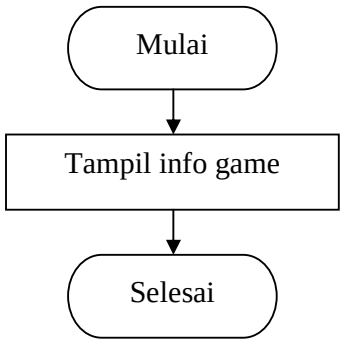

Gambar 3.10 *Flowchart* Tampil Pilih Pulau

3. *Flowchart* Cata Bermain



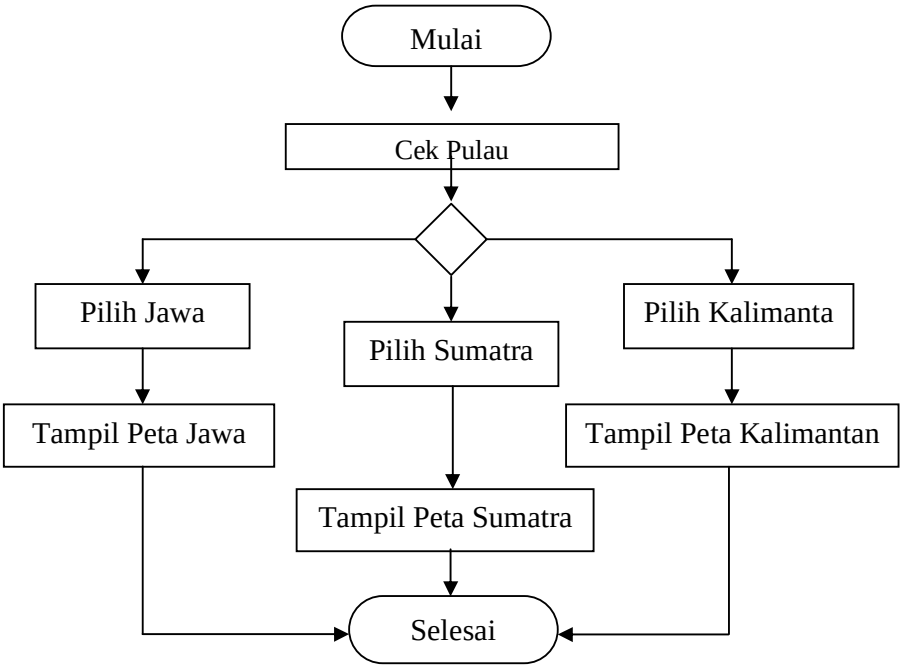

Gambar 3.11 *Flowchart* Tampil Cata Bermain

4. *Flowchart* Tampil Credits



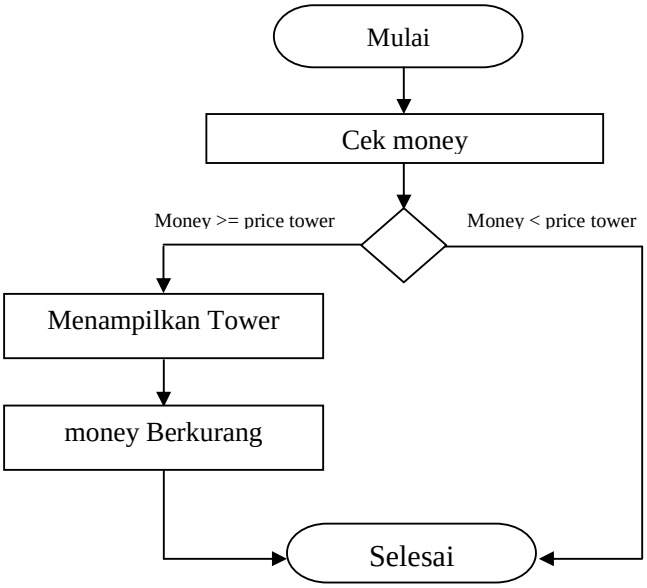

Gambar 3.12 *Flowchart* Tampil Credit

5. *Flowchart* Pilih Pulau



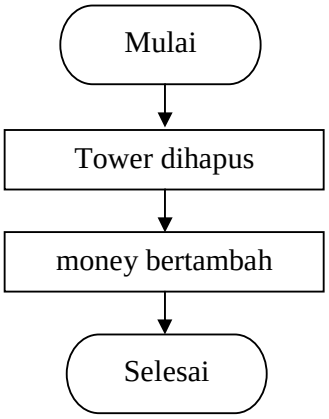

Gambar 3.13 *Flowchart* Pilih Pulau

6. *Flowchart* Bangun Tower



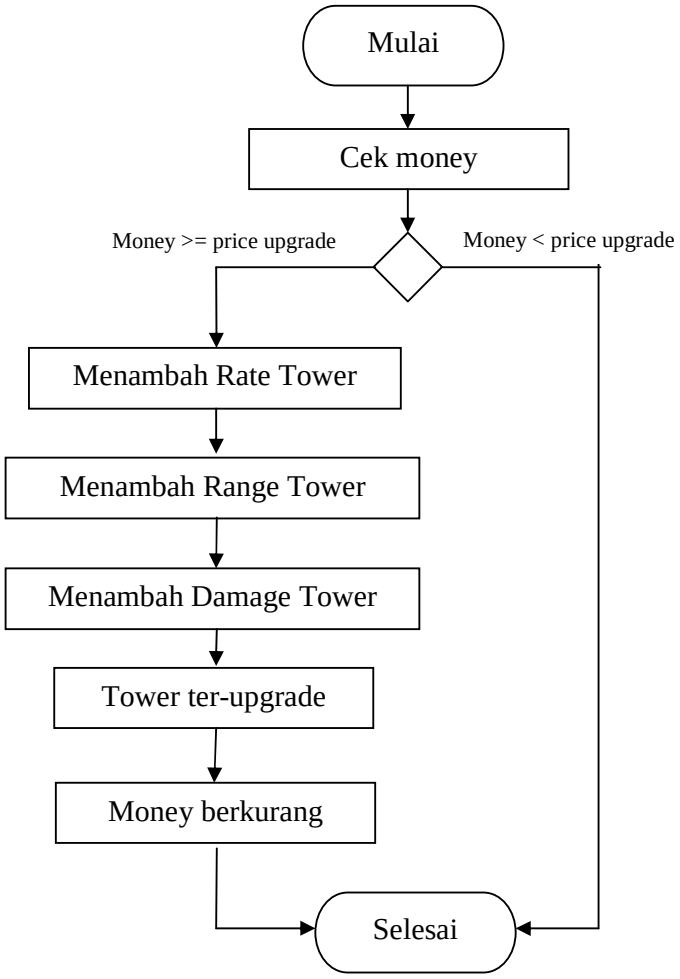

Gambar 3.14 *Flowchart* Bangun Tower

7. *Flowchart Sell* Tower



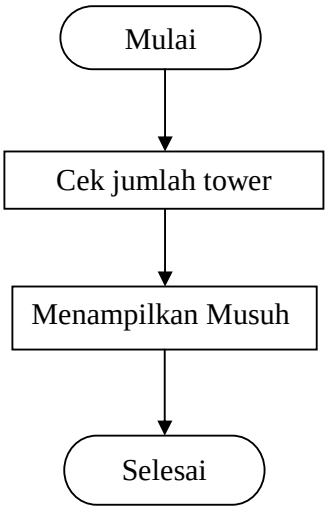

Gambar 3.15 *Flowchart Sell* Tower

8. *Flowchart Upgrade* Tower



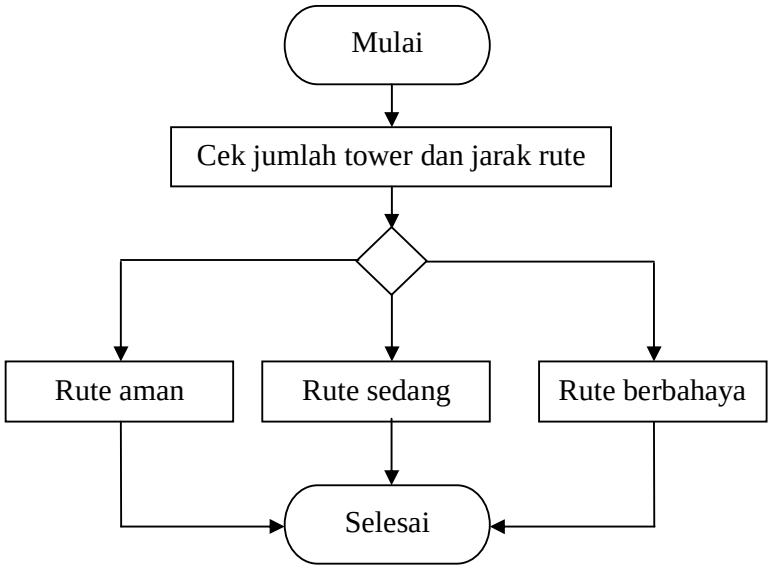

Gambar 3.16 *Flowchart Upgrade* Tower

9. *Flowchart Create* Musuh



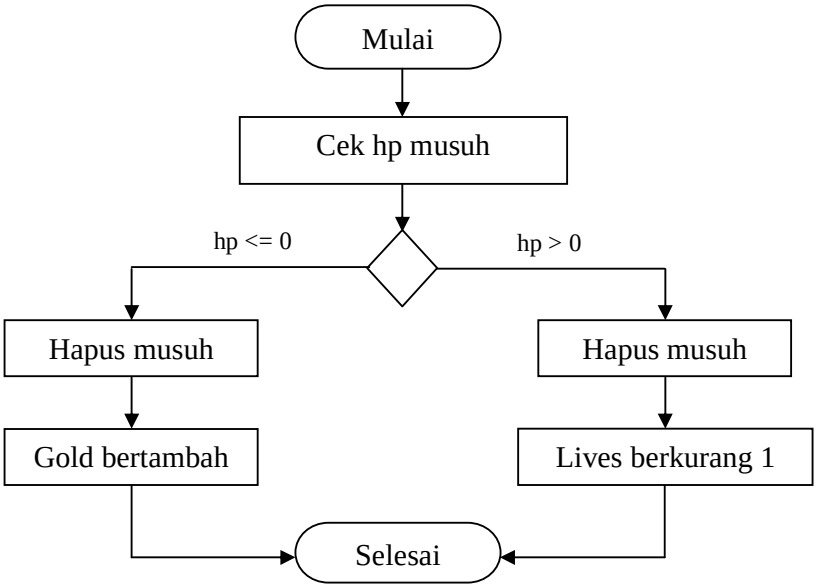

Gambar 3.17 *Flowchart Create* Musuh

10. *Flowchart* Status rute (*fuzzy*)



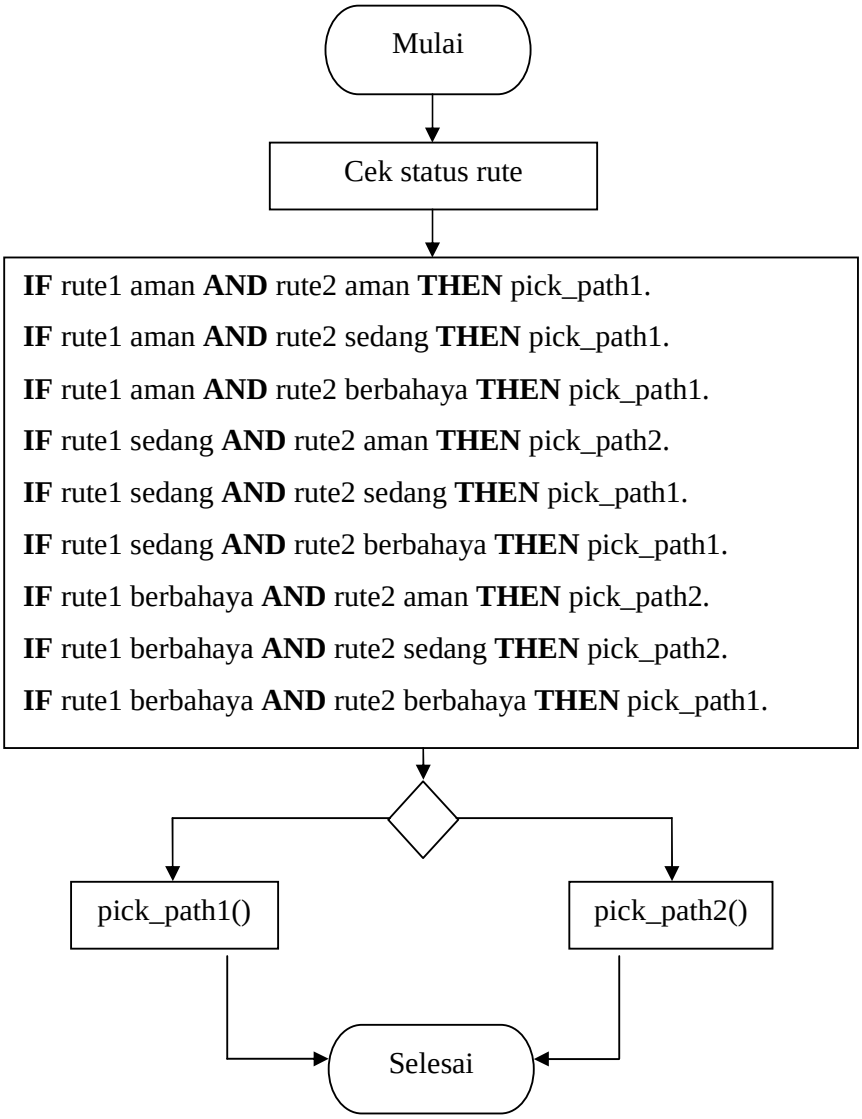

Gambar 3.18 *Flowchart* Status Rute

11. *Flowchart* Cek HP Musuh

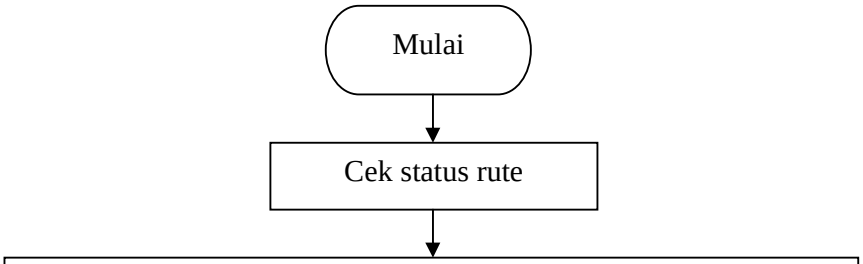



Gambar 3.19 *Flowchart* Cek HP Musuh

12. *Flowchart* Jalur Teraman




Gambar 3.20 *Flowchart* Jalur Teraman Sumatra




**IF** rute1 aman **AND** rute2 aman **AND** rute3 aman **THEN** pick_path1.

**IF** rute1 aman **AND** rute2 aman **AND** rute3 sedang **THEN** pick_path1.

**IF** rute1 aman **AND** rute2 aman **AND** rute3 berbahaya **THEN** pick_path1.

**IF** rute1 aman **AND** rute2 sedang **AND** rute3 aman **THEN** pick_path1.

**IF** rute1 aman **AND** rute2 sedang **AND** rute3 sedang **THEN** pick_path1.

**IF** rute1 aman **AND** rute2 sedang **AND** rute3 berbahaya **THEN** pick_path1.

**IF** rute1 aman **AND** rute2 berbahaya **AND** rute3 aman **THEN** pick_path1.

**IF** rute1 aman **AND** rute2 berbahaya **AND** rute3 sedang **THEN** pick_path1.

**IF** rute1 aman **AND** rute2 berbahaya **AND** rute3 berbahaya **THEN** pick_path1.

**IF** rute1 sedang **AND** rute2 aman **AND** rute3 aman **THEN** pick_path2.

**IF** rute1 sedang **AND** rute2 aman **AND** rute3 sedang **THEN** pick_path2.

**IF** rute1 sedang **AND** rute2 aman **AND** rute3 berbahaya **THEN** pick_path2.

**IF** rute1 sedang **AND** rute2 sedang **AND** rute3 aman **THEN** pick_path3.

**IF** rute1 sedang **AND** rute2 sedang **AND** rute3 sedang **THEN** pick_path1.

**IF** rute1 sedang **AND** rute2 sedang **AND** rute3 berbahaya **THEN** pick_path1.

**IF** rute1 sedang **AND** rute2 berbahaya **AND** rute3 aman **THEN** pick_path3.

**IF** rute1 sedang **AND** rute2 berbahaya **AND** rute3 sedang **THEN** pick_path1.

**IF** rute1 sedang **AND** rute2 berbahaya **AND** rute3 berbahaya **THEN** pick_path1.

**IF** rute1 berbahaya **AND** rute2 aman **AND** rute3 aman **THEN** pick_path2.

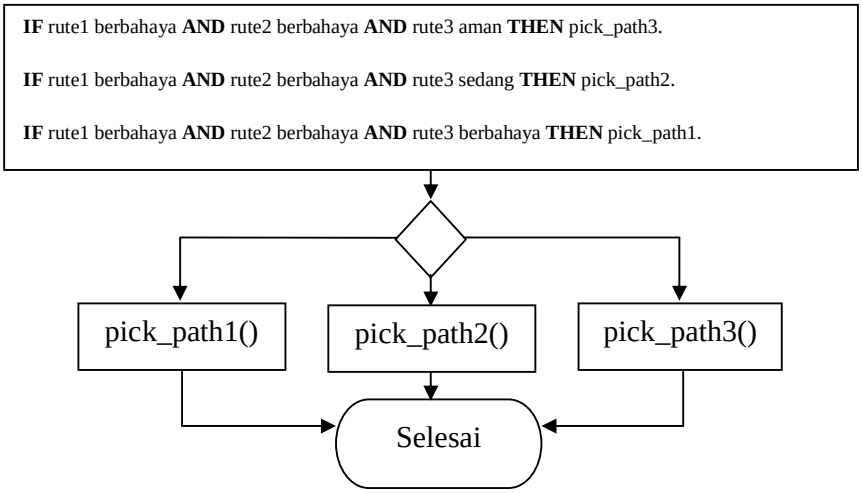
**IF** rute1 berbahaya **AND** rute2 aman **AND** rute3 sedang **THEN** pick_path2.

**IF** rute1 berbahaya **AND** rute2 aman **AND** rute3 berbahaya **THEN** pick_path2.

**IF** rute1 berbahaya **AND** rute2 sedang **AND** rute3 aman **THEN** pick_path3.

**IF** rute1 berbahaya **AND** rute2 sedang **AND** rute3 sedang **THEN** pick_path2.

**IF** rute1 berbahaya **AND** rute2 sedang **AND** rute3 berbahaya **THEN** pick_path2.




Gambar 3.21 *Flowchart* Jalur Teraman Kalimantan

## 3.2 Perancangan Sistem

Perancangan sistem adalah suatu proses yang menggambarkan bagaimana suatu sistem dibangun untuk memenuhi kebutuhan pada fase analisis. Adapun tahapan yang dilakukan dalam perancangan sistem ini membahas mengenai tujuan perancangan sistem, dan perancangan antar muka.

### 3.2.1 Tujuan Perancangan Sistem

Adapun tujuan perancangan *game* ini adalah untuk menghasilkan perangkat lunak yang mampu :

1. Memberi pemahaman tentang pentingnya hutan.
2. Membantu dan melatih berfikir strategi.
3. Merangsang daya pikir.

### 3.2.2 Gambaran Umum Sistem

Pembuatan *game* ini akan dibuat mampu menampilkan bagaimana cara menghadang musuh yang menyarang. Sedangkan musuhnya sendiri dapat menentukan rute teraman menuju sasarannya.

### 3.2.3 Perancangan Program

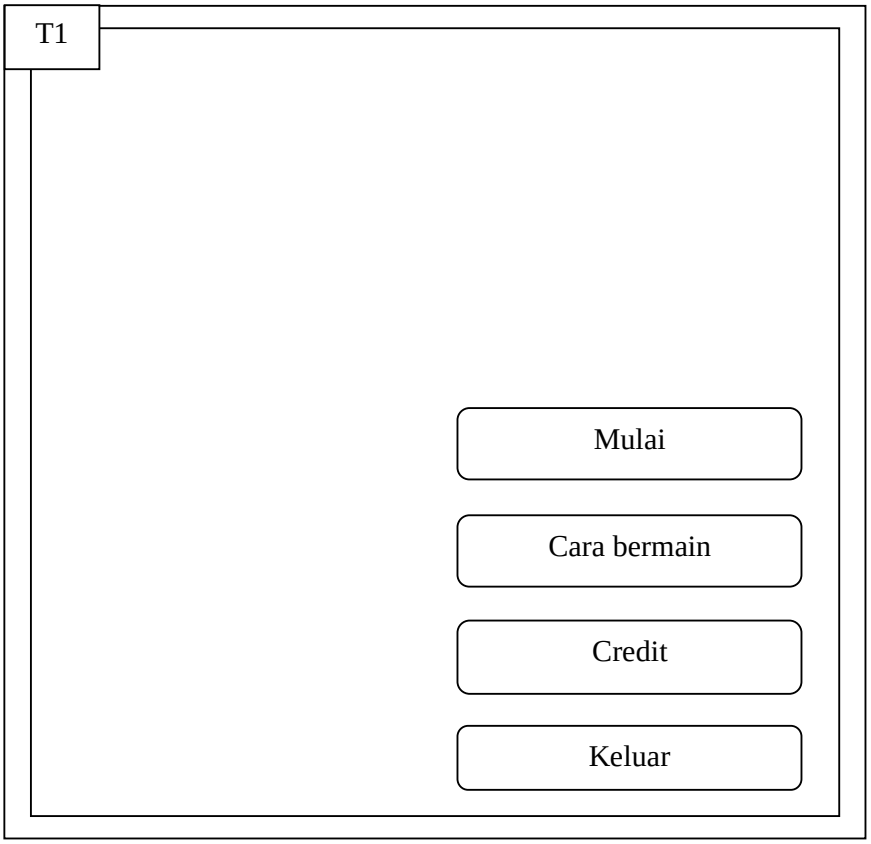
Program dapat didefinisikan sebagai kumpulan dari intruksi-intruksi atau perintah-perintah terperinci yang sudah disiapkan oleh komputer sehingga dapat melakukan fungsi sesuai dengan yang telah ditentukan. Tujuan dari perancangan *game* ini adalah untuk membantu kemampuan daya pikir strategi dalam menghadang musuh.

### 3.2.4 Perancangan Antar Muka

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

### 3.2.4.1 Antar Muka Menu Utama

Antar Muka Menu utama dari aplikasi *Game*. Gambar 3.22 menampilkan rancangan menu utama *Game Tower Defense* Polisi Hutan.




Gambar 3.22 Menu utama

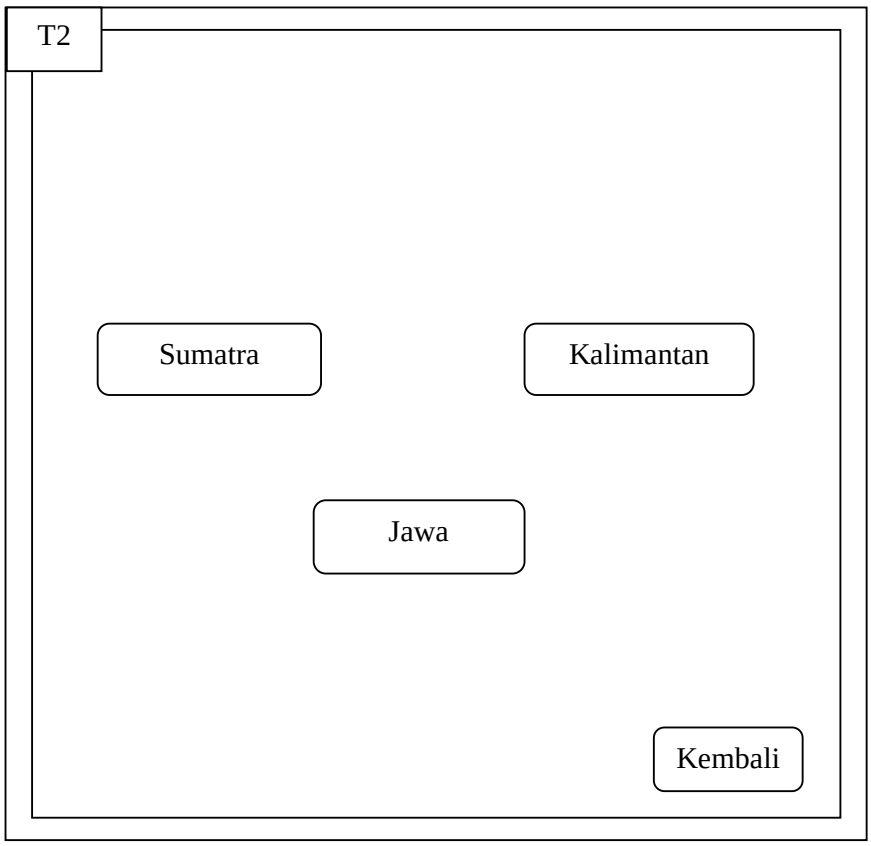
Tabel 3.3 Penggunaan Tombol Menu Utama

| No | Jenis Object | Nama Object | Keterangan |
|---|---|---|---|
| 1 | Button | Mulai | Untuk masuk pilih pulau |
| 2 | | Cara bermain | Melihat *slide* cara bermain |

| | | | |
|---|---|---|---|
| 3 | | Credit | Melihat informasi *game* |
| 4 | | Keluar | Keluar dari game psikotes |

### 3.2.4.2 Antar Muka Pilih Pulau

Antar Muka Pilih Pulau dari aplikasi *game*. Gambar 3.23 menampilkan rancangan menu Pilih Pulau *Game Tower Defense* Polisi Hutan.



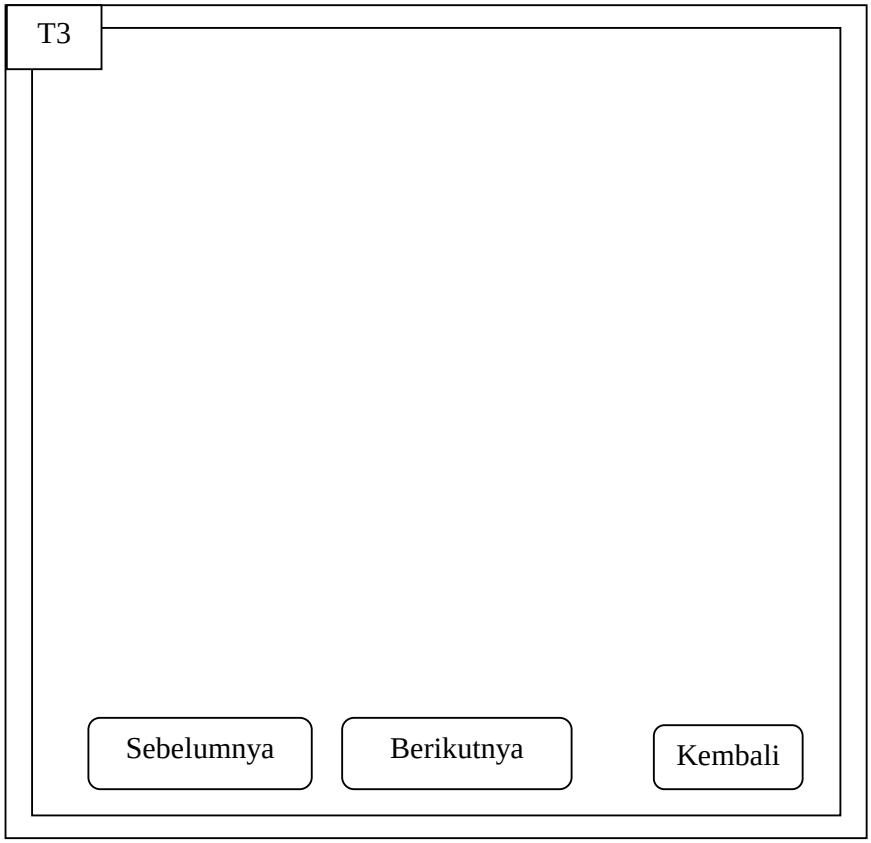

Gambar 3.23 Pilih Pulau

Tabel 3.4 Penggunaan Tombol Menu Pilih Pulau

| No | Jenis Object | Nama Object | Keterangan |
|---|---|---|---|
| 1 | Button | Sumatra | Menuju permainan di pulau Sumatra |
| 2 | | Kalimantan | Menuju permainan di pulau Kalimantan |
| 3 | | Jawa | Menuju permainan di pulau Jawa |
| 4 | | Kembali | Kembali ke menu utama |

### 3.2.4.3 Antar Muka Cara Bermain

Antar Muka Cara Bermain dari aplikasi *game*. Gambar 3.24 menampilkan rancangan menu Cara Bermain *Game Tower Defense* Polisi Hutan.



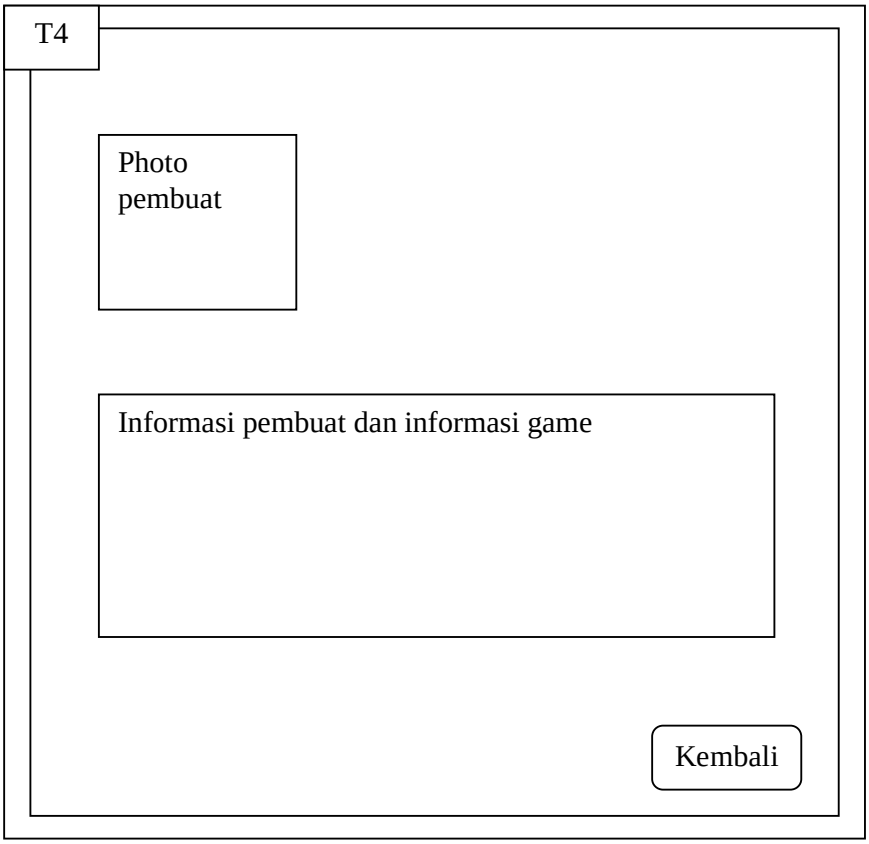

Gambar 3.24 Antar Muka Cara Bermain

Tabel 3.5 Penggunaan Tombol Menu Cara Bermain

| No | Jenis Object | Nama Object | Keterangan |
|---|---|---|---|
| 1 | Button | Sebelumnya | Menuju slide sebelumnya |
| 2 | | Berikutnya | Menuju slide berikutnya |
| 3 | | Kembali | Kembali ke menu utama |

### 3.2.4.4 Antar Muka *Credit*

Antar Muka *Credit* baru dari aplikasi *game*. Gambar 3.25 menampilkan rancangan menu *Credit Game Tower Defense* Polisi Hutan.



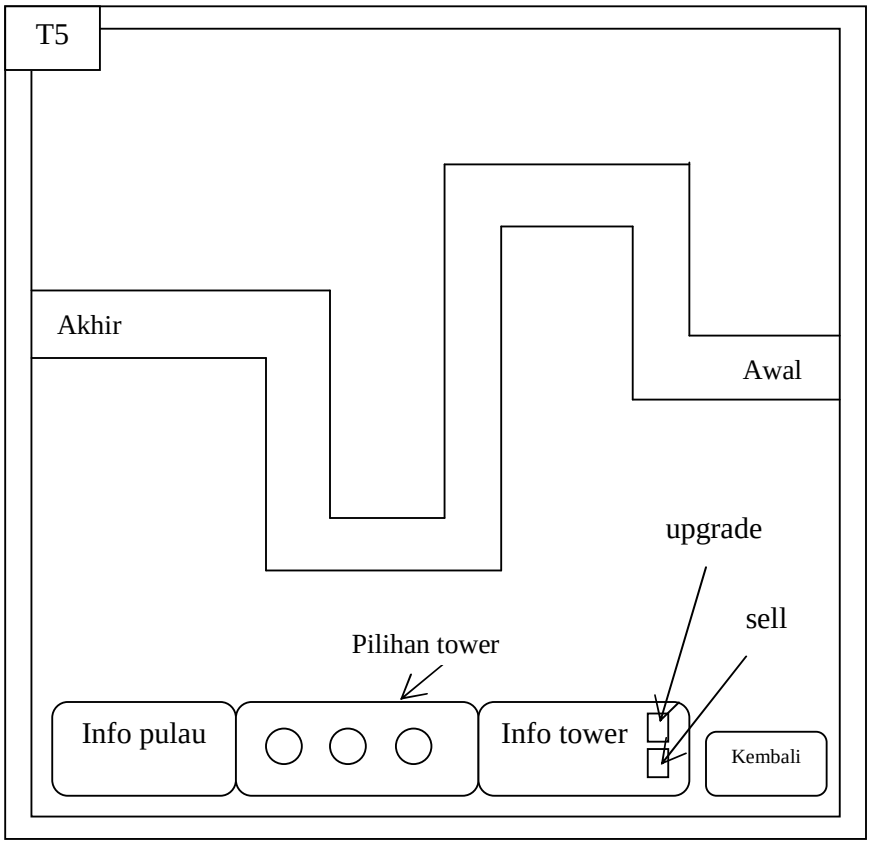

Gambar 3.25 *Credit*

Tabel 3.6 Penggunaan Tombol Menu Credit

| No | Jenis Object | Nama Object | Keterangan |
|---|---|---|---|
| 1 | Button | Kembali | Kembali ke menu utama |

### 3.2.4.5 Antar Muka Pulau Jawa

Antar Muka Pulau Jawa dari aplikasi *game*. Gambar 3.26 menampilkan rancangan Pulau Jawa *Game Tower Defense* Polisi Hutan.



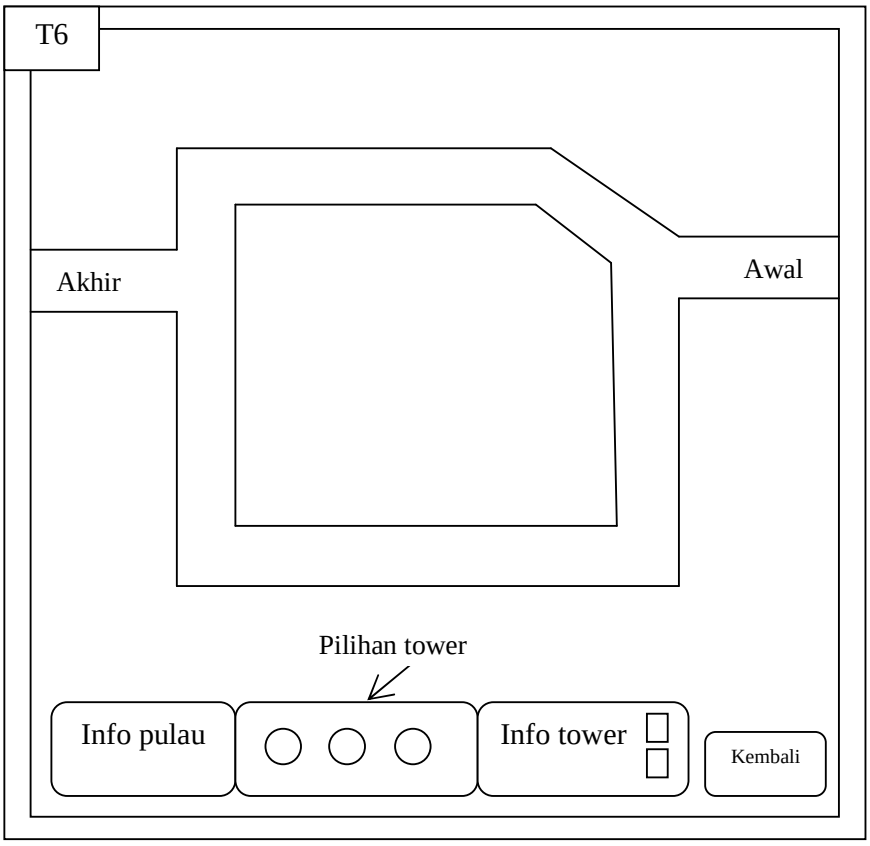

Gambar 3.26 Pulau Jawa

Tabel 3.7 Penggunaan Tombol Menu Pulau Jawa

| No | Jenis Object | Nama Object | Keterangan |
|---|---|---|---|
| 1 | Button | Pilihan tower | Memilih tower yang akan ditempatkan |
| 2 | | *Upgrade* | Meng-upgrade tower |
| 3 | | *Sell* | Menjual tower |
| 4 | | Kembali | Kembali ke menu utama |
| 5 | Label | Info pulau | Menampilkan informasi pulau, jumlah *wave*, dan jumlah *money* |
| 6 | | Info tower | Menampilkan informasi tower, besarnya *damage*, dan luas jangkauan tower |

### 3.2.4.6 Antar Muka Pulau Sumatra

Antar Muka Pulau Sumatra dari aplikasi *game*. Gambar 3.27 menampilkan rancangan Pulau Sumatra *Game Tower Defense* Polisi Hutan.



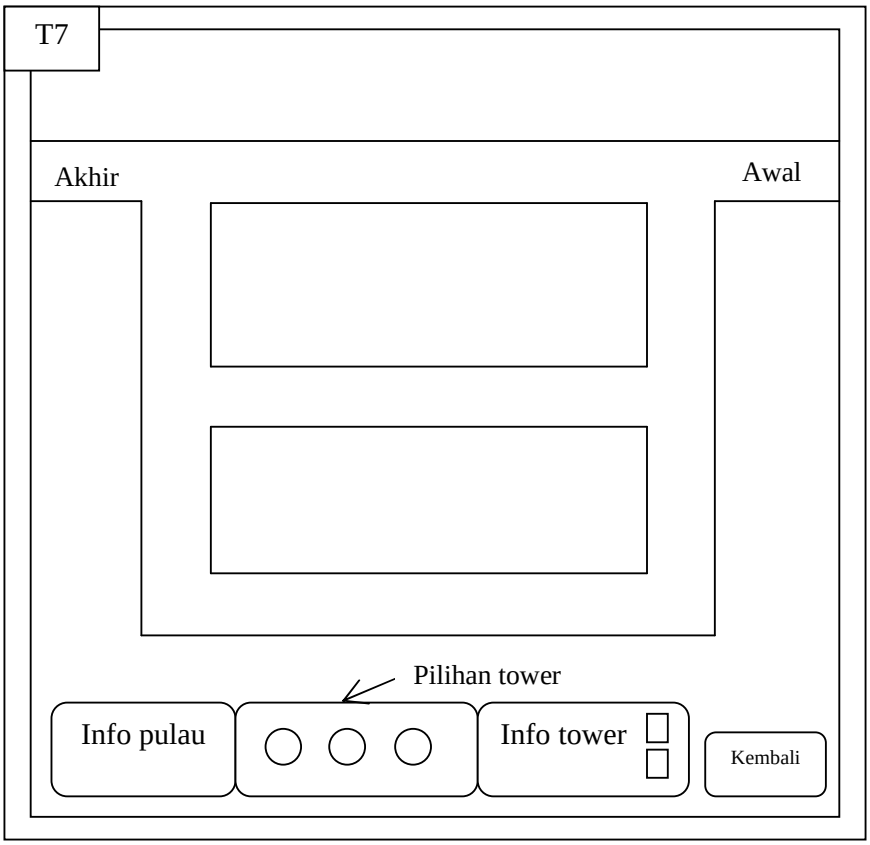

Gambar 3.27 Pulau Sumatra

Tabel 3.8 Penggunaan Tombol Menu Pulau Sumatra

| No | Jenis Object | Nama Object | Keterangan |
|---|---|---|---|
| 1 | Button | Pilihan tower | Memilih tower yang akan ditempatkan |
| 2 | | *Upgrade* | Meng-upgrade tower |
| 3 | | *Sell* | Menjual tower |
| 4 | | Kembali | Kembali ke menu utama |
| 5 | Label | Info pulau | Menampilkan informasi pulau, jumlah *wave*, dan jumlah *money* |
| 6 | | Info tower | Menampilkan informasi tower, besarnya *damage*, dan luas jangkauan tower |

### 3.2.4.7Antar Muka Pulau Kalimantan

Antar Muka Pulau Kalimantan dari aplikasi *game*. Gambar 3.28 menampilkan rancangan Pulau Kalimantan *Game Tower Defense* Polisi Hutan.




Gambar 3.28 Pulau Kalimantan

Tabel 3.9 Penggunaan Tombol Menu Pulau Kalimantan

| No | Jenis Object | Nama Object | Keterangan |
|---|---|---|---|
| 1 | Button | Pilihan tower | Memilih tower yang akan ditempatkan |
| 2 | | *Upgrade* | Meng-upgrade tower |
| 3 | | *Sell* | Menjual tower |
| 4 | | Kembali | Kembali ke menu utama |
| 5 | Label | Info pulau | Menampilkan informasi pulau, jumlah *wave*, dan jumlah *money* |
| 6 | | Info tower | Menampilkan informasi tower, besarnya *damage*, dan luas jangkauan tower |

### 3.2.5 *Storyboard*

Storybard menjelaskan mengenai komponen-komponen yang terkandung dalam aplikasi *game Tower Defense Polisi Hutan*.

### 3.2.5.1 Tower

Tower adalah pasukan Polisi Hutan yang dapat disewa untuk melindungi hutan dan menyelesaikan stage *Tower Defense.*

Tabel 3.10 Karakter Polisi Hutan

| No | Jenis | Gambar | Keterangan |
|---|---|---|---|
| 1 | Laser |  | Merupakan polisi dengan attack kecil, jangkauan pendek, namun waktu reload yang cepat. |
| 2 | Bullet |  | Merupakan polisi dengan attack besar, jangkauan luas, namun waktu reload yang lama. |
| 3 | Bomber |  | Merupakan polisi dengan jangkauan pendek, waktu reload yang lambat, namun memiliki attack poin yang sangat besar. |

**1. Laser**

Memiliki perkembangan kekuatan antara lain:

Tabel 3.11 Status Laser

| Level | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| **Attack** | 10 | 11 | 12 | 13 | 14 | 15 |
| **Range** | 100 | 110 | 121 | 133 | 146 | 161 |
| **Reload** | 3/s | 3,5/s | 4,32/s | 5,18/s | 6,22/s | 7,46/s |

2. ***Bullet***

Memiliki perkembangan yang stabil, dapat memperlambat pergerakan musuh.

Tabel 3.12 Status *Bullet*

| **Level** | **0** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|---|
| **Attack** | 25 | 28 | 31 | 34 | 37 | 41 |
| **Range** | 150 | 165 | 182 | 200 | 220 | 242 |
| **Reload** | 2/s | 2,40/s | 2,88/s | 3,46/s | 4,15/s | 4,98/s |

3. ***Bomber***

Merupakan *Polisi* yang tidak memiliki attack. Tetapi dapat memperlambat gerakan musuh.

Tabel 3.13 Status *Bomber*

| **Level** | **0** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|---|
| **Attack** | 35 | 38 | 42 | 46 | 51 | 56 |
| **Range** | 150 | 165 | 182 | 200 | 220 | 242 |
| **Reload** | 1/s | 1,20/s | 1,44/s | 1,73/s | 2,07/s | 2,49/s |

**3.2.5.2 *Enemy***

Pada bagian ini akan dijelaskan mengenai karakter musuh yang ada dalam aplikasi *game Tower Defense Polisi Hutan*. Data disajikan dalam bentuk tabel :

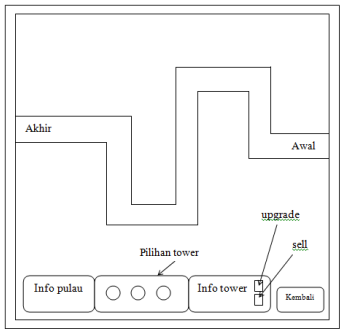
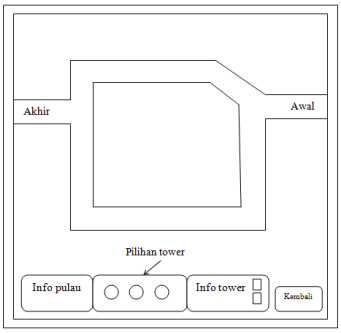
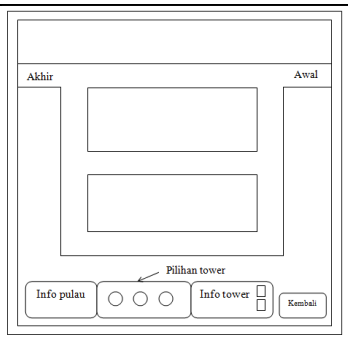
Tabel 3.14 Karakter Musuh

| No | Nama | Gambar | HP | Speed | Keterangan |
|---|---|---|---|---|---|
| 1 | Kapakwan |  | 10 | 15 | Merupakan Penebang dengan kapak yang memiliki Heal Poin (HP) terkecil. |
| 2 | Sawman |  | 20 | 10 | Merupakan Penebang dengan gergaji mesin yang memiliki HP menengah. |
| 3 | Buldozer |  | 30 | 10 | Merupakan Penebang dengan menggunakan Buldozer yang memiliki HP terbesar. |

### 3.2.5.3 *Map*

Pada sub bab ini, akan dijelaskan mengenai seluruh komponen tempat yang ada pada aplikasi *Game Tower Defense Polisi Hutan*. Keseluruhan pulau pada *Game Tower Defense Polisi Hutan* adalah 3 pulau. Berikut adalah tabel 3 pulau tersebut.

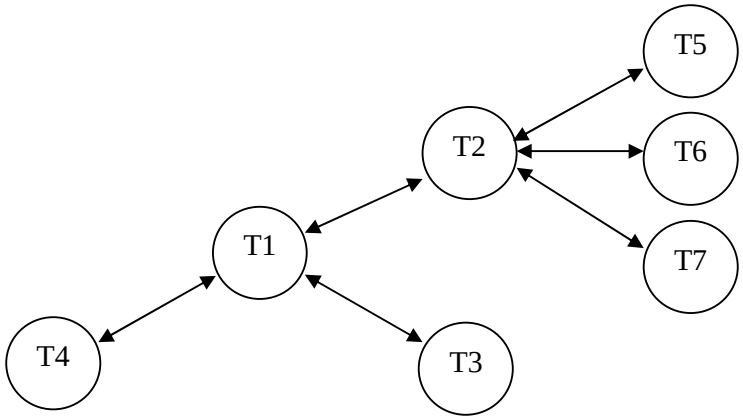
Tabel 3.15 Pulau

| No | Nama | Gambar | Keterangan |
|---|---|---|---|
| 1 | Pulau Jawa |  | Pulau untuk level 1 |
| 2 | Pulau Sumatra |  | Pulau untuk level 2 |
| 3 | Pulau Kalimantan |  | Pulau untuk level 3 |

### 3.2.6 Jaringan Semantik

Jaringan Semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar objek (lingkaran) dan garis yang

menggambarkan formasi antar objek tersebut. Adapun jaringan semantik untuk aplikasi *game* dapat dilihat pada Gambar 3.29.




Gambar 3.29 Jaringan Semantik

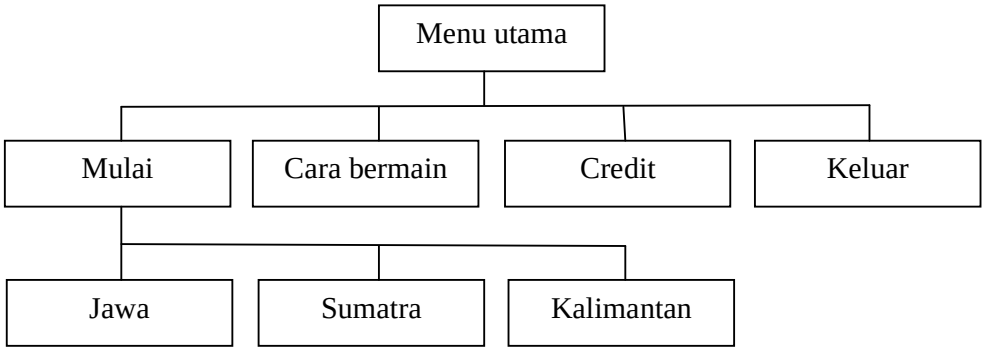
Tabel 3.16 Keterangan Jaringan Semantik

| NO | Nama Object | Keterangan |
|---|---|---|
| 1 | T1 | Antar muka menu utama |
| 2 | T2 | Antar muka pilihan pulau |
| 3 | T3 | Antar muka cara bermain |
| 4 | T4 | Antar muka credit |
| 5 | T5 | Antar muka Pulau Jawa |
| 6 | T6 | Antar muka Pulau Sumatra |
| 7 | T7 | Antar muka Pulau Kalimantan |

### 3.2.7 Struktur Menu Utama

Desain menu utama merupakan awal dimulainya proses *game* tersebut, desain ini dirancang untuk memenuhi pemakai agar proses dalam *game* tersebut dapat berjalan dengan sebagaimana mestinya.

Didalam menu utama ini terdapat beberapa menu yang berguna bagi *user* untuk menggunakan *Game Tower Defense Polisi Hutan* tersebut.




Gambar 3.30 Struktur menu

### 3.2.8 Perancangan Program

Proses perancangan ini merupakan tahap awal dari perancangan *Game Tower Defense Polisi Hutan*, yang dilakukan sebagai pemecahan masalah yang ada pada proses *game* ketika dimainkan dan yang dimaksud dengan analisis adalah proses mendekomposisi atau menginvestigasi domain *problem* atau *requirements* menjadi bagian-bagian yang kita mengerti. Proses analisis lebih menekankan kepada pertanyaan seperti apakah sistem yang akan kita kerjakan, bagaimana sistem tersebut digunakan sehari-hari oleh *user*.

Sedangkan perancangan adalah melakukan sintesa model *software* dari *problem* yang kita analisis, menterjemahkan domain model kepada *software* model sehingga bisa dimengerti dan dikomunikasikan kepada anggota tim lain dan juga para profesional *software* lainnya.