

BAB III

ANALISA DAN PERANCANGAN

III.1. Analisis Sistem

Analisis sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka.

Analisis sistem juga banyak orang yang mendefinisikan sebagai bagaimana memahami dan menspesifikasikan dengan detail apa yang harus dilakukan oleh sistem. Analisis sistem merupakan tahapan paling awal dari pengembangan sistem yang menjadi fondasi untuk menentukan keberhasilan sistem yang dihasilkan nantinya.

Adapun perancangan *game roguelike* ini bertujuan sebagai sarana hiburan dan ditujukan untuk pemain yang tidak terlalu paham dengan permainan *genre roguelike* ini. Oleh sebab itu penulis merancang permainan yang sederhana namun tetap memberikan tantangan kepada pemain.

Kendala-kendala yang dihadapi pada sistem yang sedang berjalan yaitu:

1. *Game* yang dibuat dapat berjalan pada sistem operasi *Android* dengan versi minimal *Android* versi 4.1 (*Jellybean*).
2. Objek-objek yang digunakan dalam *game* ini menggunakan *sprite* berbasis 2D.
3. *Stage (scene)* dalam *game* ini dibuat tak terhingga (*infinite stage*) dengan tujuan agar pemain dapat mencapai *stage* setinggi mungkin.

III.2. Analisis Masalah

Dalam merancang sebuah aplikasi tidak luput dari beberapa masalah. Adapun analisa masalah di dalam *game roguelike* ini adalah sebagai berikut:

1. Penentuan Tujuan dan Jenis *Game*

Penentuan tujuan *game* sangat berpengaruh terhadap perancangan *game* itu sendiri. Adapun tujuan dari *game* ini yaitu pemain berusaha mencapai *stage* setinggi mungkin serta mengalahkan musuh untuk mendapatkan *experience* yang diperlukan untuk menaikkan *level* dari karakter pemain. Jenis *game* ini adalah *roguelike* dengan menggunakan mekanisme permainan giliran (*turn based*).

2. Penentuan *Storyline* (Alur Cerita)

Setelah penentuan tujuan *game*, langkah selanjutnya adalah pembuatan *storyline*. *Storyline game* adalah penceritaan *enviroment* (lingkungan), target aktor *game* (pemain), dan kenaikan *level game*.

Dalam *game roguelike* ini *enviroment* berupa *tiled based map* dengan setiap *tile* diisi oleh satu model objek. Model objek yang digunakan adalah model yang menggambarkan suasana hutan seperti pohon, semak, dan lain-lain sehingga *map* yang dihasilkan akan terlihat seperti hutan.

Target dari pemain adalah untuk mengendalikan karakter utama untuk mencapai *stage* selanjutnya dan mengalahkan musuh yang ada. Dalam *game* ini ada dua hal yang disebut sebagai *level*. Yang pertama yaitu *level* dari *game* yang disebut *stage*, dan yang kedua level pemain yang disebut *level*. Untuk mencapai *stage* selanjutnya pemain harus mengalahkan semua musuh di setiap *stage* untuk memunculkan pintu keluar kemudian pemain mengarahkan karakter utama

menuju pintu keluar. Semakin tinggi *stage* maka semakin kuat karakter musuh sehingga akan meningkatkan kesulitan dalam permainan.

3. Pembuatan Model *Game* dan Implementasi Model ke *Game*

Game ini menggunakan objek 2D berbasis *sprite*, dimana dalam pembuatannya menggunakan *Adobe Photoshop CS4*. Hasil dari pembuatan model tersebut akan di-*convert* ke dalam bentuk *.jpg* dan *.png*. Setelah pemodelan selesai dilakukan, *file-file* tersebut di-*export* ke *Unity Engine* agar model yang dibuat dapat dibaca oleh *Unity Engine*.

4. Penulisan *Script* Program dan Penggabungan *Script* ke Objek

Tahap ini ditujukan untuk mengatur setiap objek tertentu, antara lain pembentukan map secara prosedural, mengatur pergerakan karakter utama dan musuh, deteksi antar objek, dan pengaturan suara (*background music* dan *sound effect*).

III.3. Analisis Kebutuhan

Game ini dirancang menggunakan *software Unity Engine* versi 5.0.1, penulisan *listing* program (*scripting*) menggunakan *software MonoDevelop* versi 4.0.1, dan untuk pembuatan model objek menggunakan *Adobe Photoshop CS4*.

Adapun untuk menjalankan *game roguelike* ini menggunakan *smartphone Android* dengan spesifikasi minimum sebagai berikut:

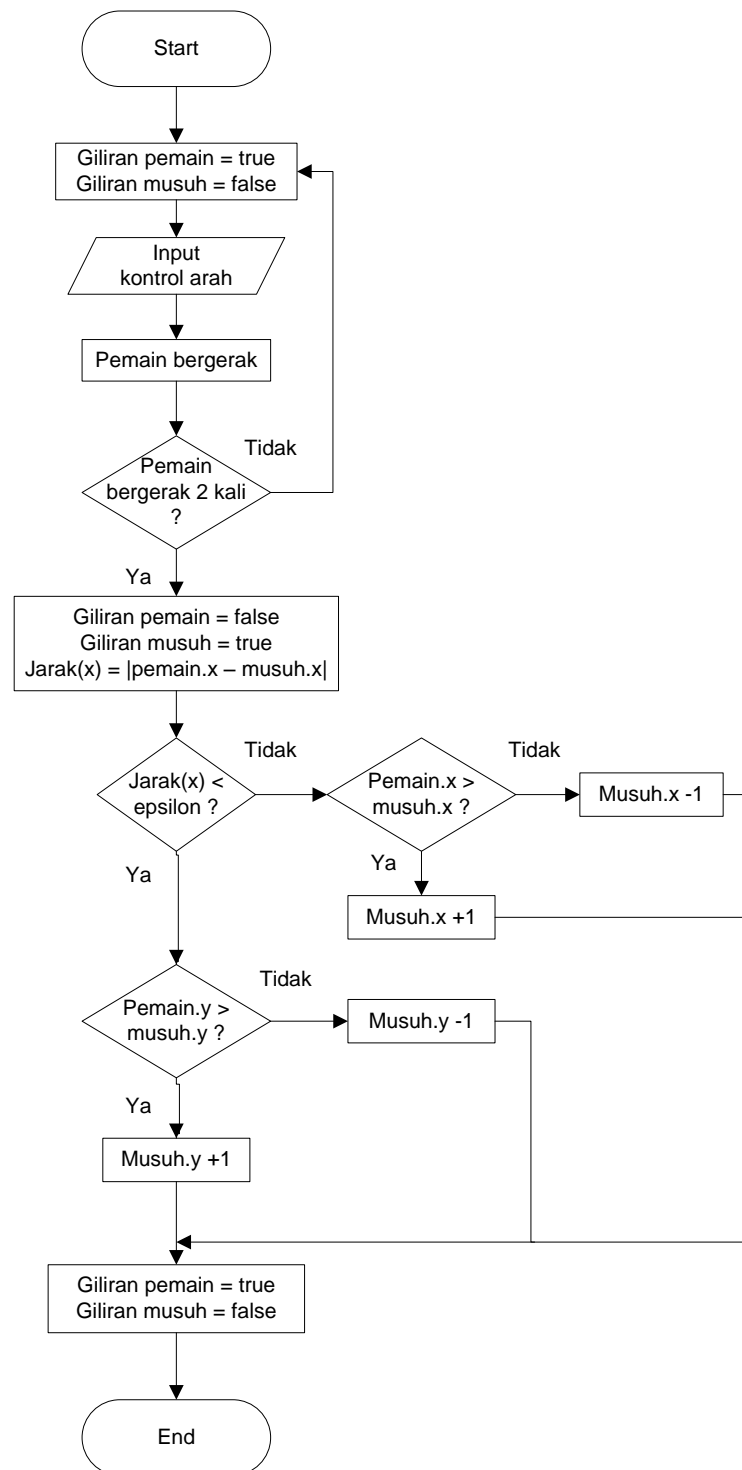
1. *Processor* 1.2 Ghz
2. *Harddisk* 128 MB
3. RAM 512MB

4. Sistem operasi: *Android Jellybean* (4.1)

III.4. Penerapan Algoritma

Algoritma merupakan kumpulan instruksi atau langkah-langkah yang jelas untuk menyelesaikan suatu masalah. Algoritma memegang peranan penting dalam bidang pemrograman sehingga dapat dipahami konsep dasar algoritma program yang lebih efektif dan efisien. Dalam *game* ini terdapat algoritma yang diterapkan adalah Algoritma *Pathfinding*.

Algoritma *pathfinding* diterapkan pada karakter musuh dimana musuh akan mencari kemudian bergerak ke arah karakter utama. *Pathfinding* merupakan cara untuk mendapatkan *route* (jalur) antara dua buah *point* atau titik, dalam hal ini *route* antara karakter musuh menuju karakter utama. Ada banyak jenis algoritma *pathfinding* yang dapat digunakan dalam *game*, akan tetapi penulis memilih algoritma *Brute Force* dikarenakan algoritma yang sederhana dan mudah diterapkan. Cara kerja algoritma *Brute Force* adalah membandingkan posisi sekarang dengan posisi tujuan dan menentukan langkah berikutnya (Ahmad Fauzan, 2012). Untuk penerapan algoritma ini dapat disesuaikan menurut kondisi yang diinginkan. Adapun untuk penjelasan dari algoritma ini dapat dilihat pada flowchart pada gambar III.1:



Gambar III.1. Flowchart Pathfinding pada Karakter Musuh

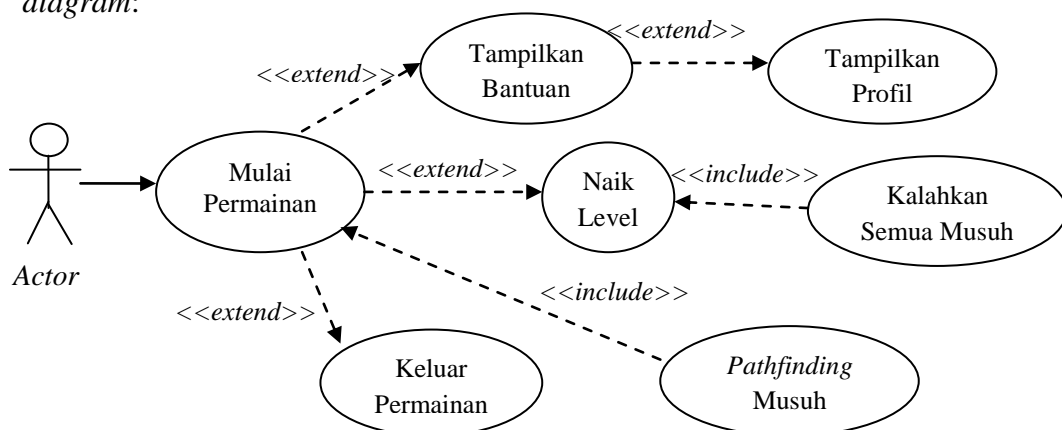
III.5. Desain Sistem

Metode perancangan yang digunakan untuk merancang *game roguelike* berbasis *Android* ini adalah metode perancangan terstruktur menggunakan UML. UML pada dasarnya merupakan konsep perancangan untuk mengetahui proses dan jalannya aliran data serta penyimpanan data yang merupakan konsep perancangan termudah dengan pendekatan *top down*.

III.5.1. Use Case Diagram

Use case diagram adalah sebuah gambaran dari fungsi sistem yang dipandang dari sudut pandang pemakai. *Actor* adalah segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi (Whitten, 2004: 258).

Berikut perancangan aplikasi yang digambarkan menggunakan *use case diagram*:











Gambar III.2. Use Case Diagram

III.5.2. Storyboard

Storyboard adalah area berseri dari sebuah gambar sketsa yang digunakan sebagai alat perencanaan untuk menunjukkan secara visual bagaimana aksi dari sebuah cerita berlangsung. *Storyboard* sangat diperlukan dalam hal menjelaskan

proses-proses yang terjadi pada *game* yang dirancang. Berikut adalah *storyboard* dari *game roguelike* berbasis *Android*:

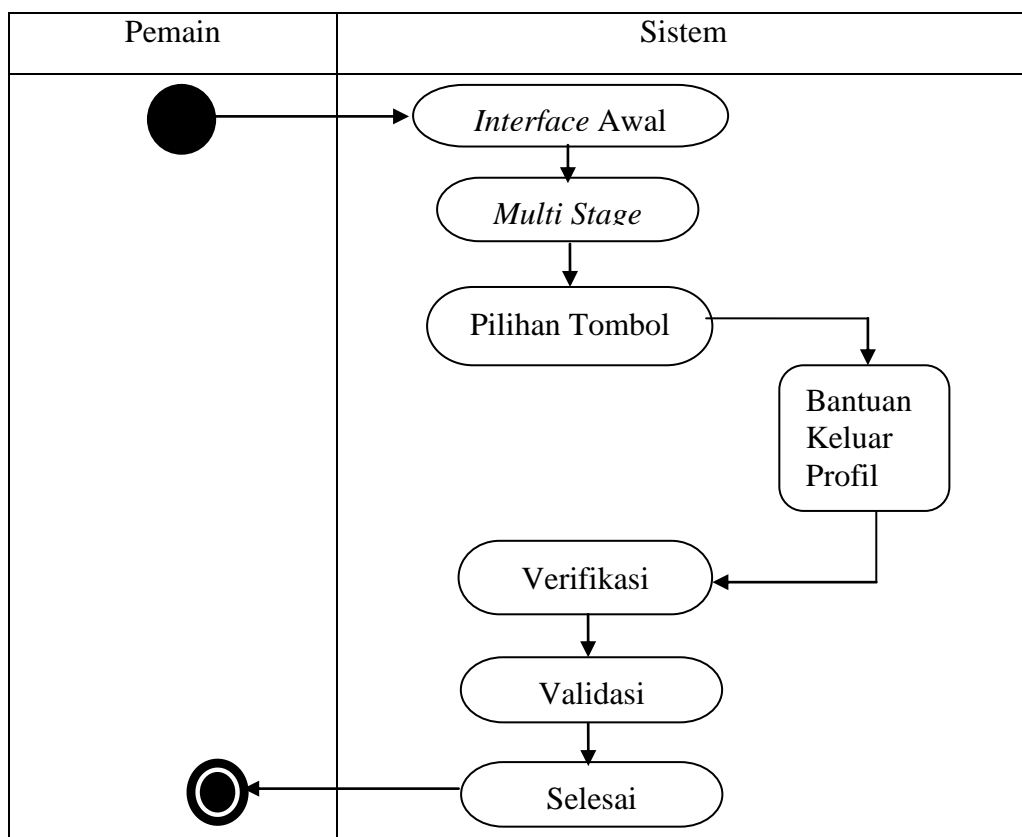
Tabel III.1. Storyboard Game Roguelike Berbasis Android

Nama Objek	Gambar	Deskripsi
Karakter utama		Karakter yang digerakkan oleh pemain menuju pintu keluar, setiap kali karakter bergerak maka poin makanan akan berkurang.
Karakter musuh		Karakter yang akan bergerak menuju karakter utama dan menyerangnya sesuai dengan algoritma yang diterapkan
Halangan		Objek <i>game</i> yang terdapat pada <i>map</i> yang berfungsi sebagai halangan dimana letaknya acak sesuai dengan <i>script</i> yang telah dibuat
Makanan		<i>Item</i> yang digunakan untuk menambah poin makanan dari karakter utama
Ramuan Obat		<i>Item</i> yang digunakan untuk menambah darah dari karakter utama.
Pintu Keluar		Objek <i>game</i> yang berfungsi untuk mengarahkan pemain menuju <i>stage</i> selanjutnya
Teks <i>Hp</i> dan <i>Food</i>		Teks yang menampilkan info mengenai jumlah poin dari makanan dan darah yang dimiliki karakter utama
Teks Aksi		Teks yang menampilkan info mengenai aksi yang terjadi seperti jumlah damage dan jumlah poin yang didapat dari <i>item</i>

III.5.3. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* merupakan *state diagram* khusus, yang sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu, *activity diagram* tidak menggambarkan perilaku internal sebuah sistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Adapun *activity diagram* pada *game roguelike* berbasis *Android* ini adalah:

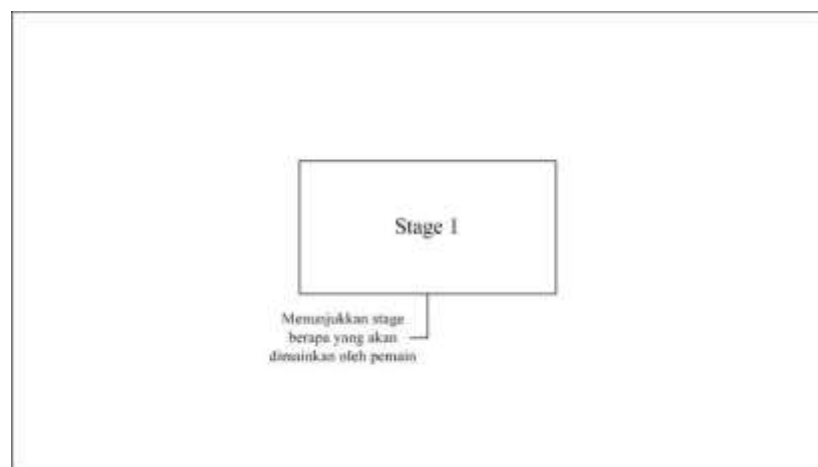


Gambar III.3. Activity Diagram

III.6. Desain *User Interface*

1. Tampilan *Layer Stage*

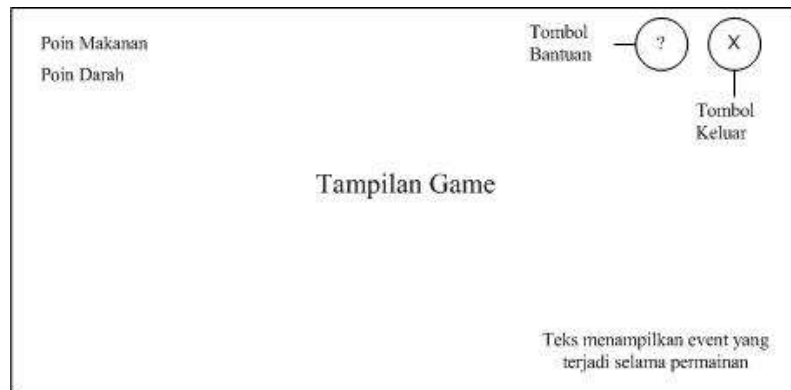
Tampilan awal ketika pemain memulai *game* ini adalah tampilan layar yang menunjukkan *stage (stage layer)* berapa yang dimainkan oleh pemain, kemudian tampilan *stage layer* akan tertutup yang kemudian akan menampilkan tampilan *game*.



Gambar III.4. Tampilan Awal *Game (Stage Layer)*

2. Tampilan Tombol *Input*

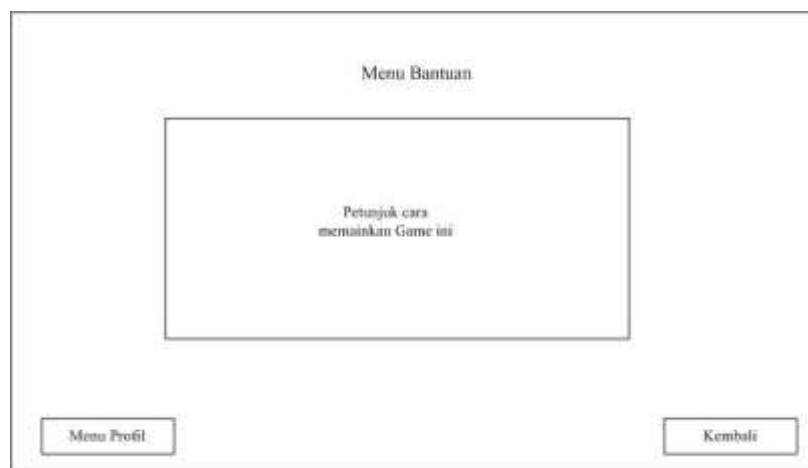
Desain tampilan *input* pada pembuatan *game roguelike* berbasis *Android* ini adalah pemain men-*touch* tombol yang terdapat pada layar *smartphone Android* tersebut, dimana tombol-tombol tersebut berupa tombol bantuan dan tombol keluar. Adapun untuk menggerakkan karakter pemain harus mengusap (*swipe*) layar sesuai arah yang ingin dituju pemain.



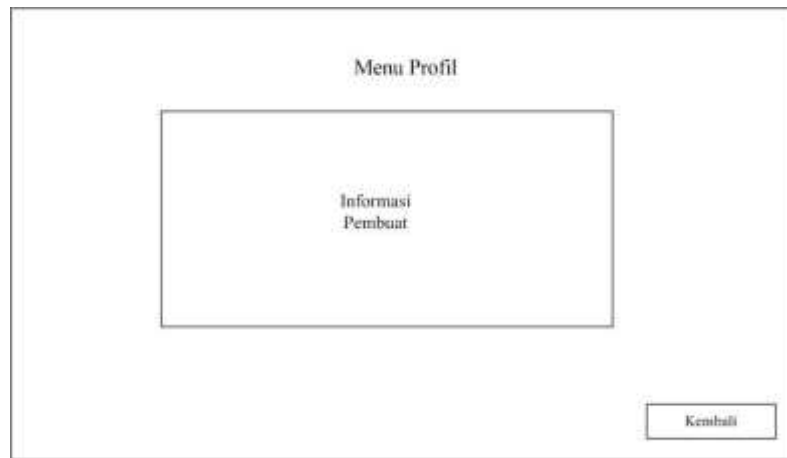
Gambar III.5. Tampilan *Input Game*

3. Tampilan Menu Bantuan

Pada saat pemain men-*touch* tombol bantuan maka tampilan bantuan akan terbuka. Menu bantuan berisi petunjuk cara memainkan *game* ini. Pada menu ini juga terdapat Menu Tentang yang berisi informasi tentang pembuat.



Gambar III.6. Tampilan Menu Bantuan



Gambar III.7. Tampilan Menu Profil

4. Tampilan *Game Over*

Pada saat poin makanan atau poin darah dari karakter utama mencapai angka nol, maka *game* berakhir dan layar tampilan *Game Over* akan muncul yang berisikan pesan *Game Over* dan *stage* yang sudah dicapai.



Gambar III.8. Tampilan *Game Over*