

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* ini bertujuan sebagai sarana hiburan dan untuk melatih ketangkasan pemain. 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 *Windows* dengan versi minimal *Windows Vista*.
2. Objek-objek yang digunakan dalam *game* ini menggunakan *sprite* berbasis 2D.
3. AI (*Artificial Intelligence*) pada *game* ini menggunakan algoritma pencarian dijkstra yang digunakan untuk menggerakkan musuh menuju lokasi pemain.

III.1.2. Analisis Masalah

Dalam merancang sebuah aplikasi tidak luput dari beberapa masalah. Adapun analisa masalah di dalam *game tank* 2D 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 mengalahkan semua musuh yang ada untuk mencapai *level* selanjutnya. Jenis *game* ini adalah *sidescrolling shooter*.

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

Game ini menggunakan objek 2D berbasis *sprite*, dimana dalam pembuatannya menggunakan *software Corel Draw X6* dan *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*.

3. 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.1.3. 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 *software CorelDraw X6* dan *Adobe Photoshop CS4*.

Adapun untuk menjalankan *game* ini menggunakan PC(*Personal Computer*) dengan spesifikasi minimum sebagai berikut :

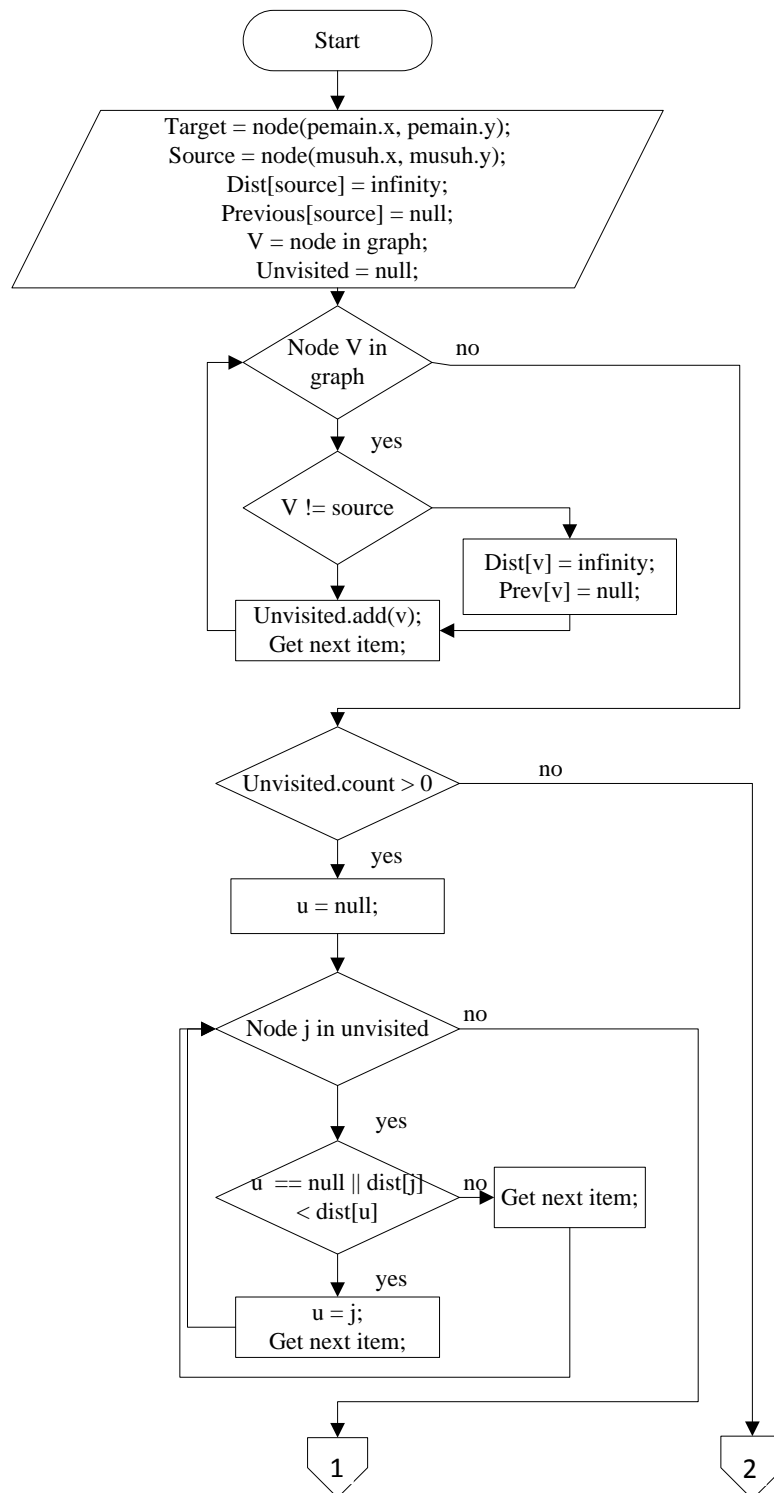
1. *Processor* 1.2 GHz
2. *Harddisk* 80 GB
3. RAM 512MB
4. Sistem operasi: *Windows Vista* x32

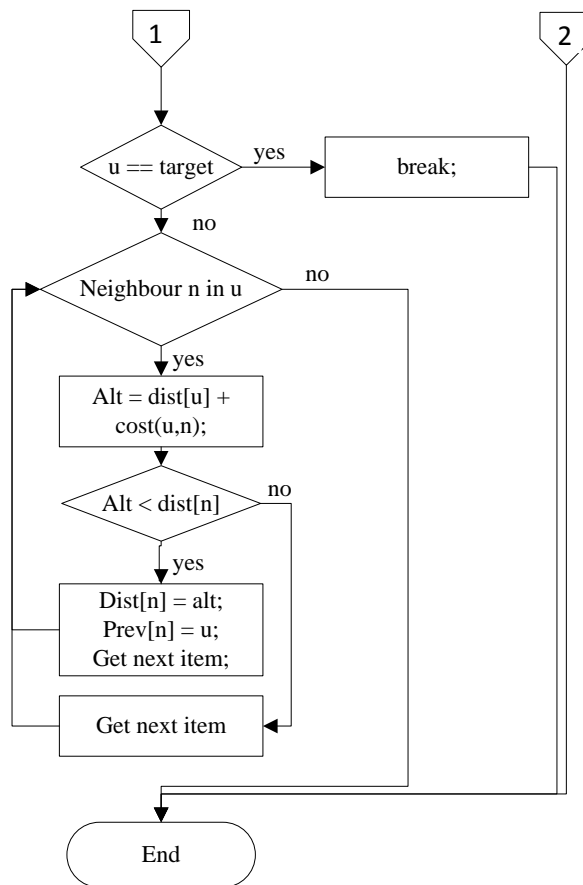
III.2. Penerapan Algoritma

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 yaitu Algoritma *Dijkstra*.

Algoritma *Dijkstra* merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari verteks a ke z dalam graph berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif. Namun jika terjadi demikian, maka penyelesaian yang

diberikan adalah infiniti (Tak Hingga). Pada algoritma Dijkstra, node digunakan karena algoritma Dijkstra menggunakan graph berarah untuk penentuan rute listasan terpendek. Pada *game* ini, algoritma Dijkstra diterapkan pada karakter musuh untuk pencarian jalur antara musuh dan pemain dimana setiap node merupakan *tile* yang dapat dilalui pada map bernilai 1 dan halangan atau *tile* yang tidak dapat dilalui bernilai tak hingga(*infinite*) sehingga didapatkan jalur yang optimal. Adapun untuk penjelasan dari algoritma ini dapat dilihat pada *flowchart* pada gambar III.1 berikut :





Gambar III.1. Flowchart Dijkstra pada Karakter Musuh

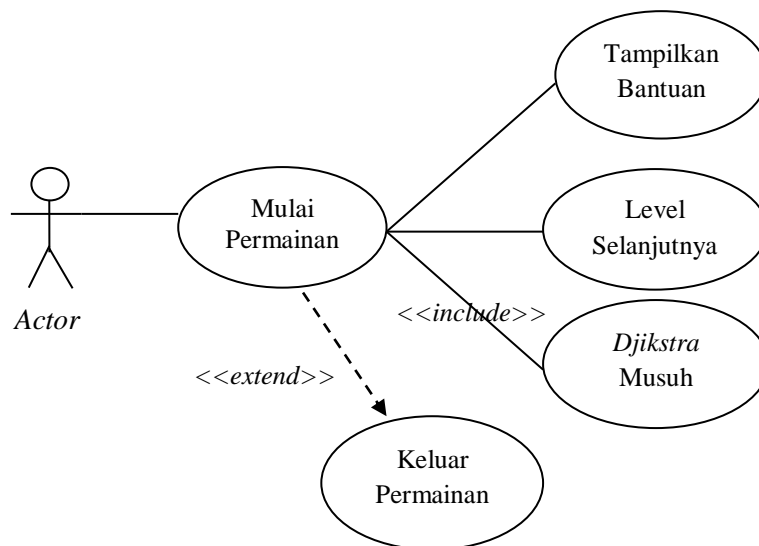
III.3. Desain Sistem

Metode perancangan yang digunakan untuk merancang *game tank* 2D 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.3.1. 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* :




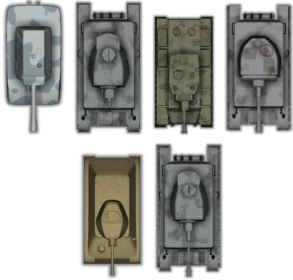
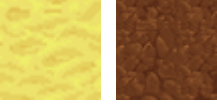

Gambar III.2. Use Case Diagram

III.3.2. 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 tank 2D* ini :

Tabel III.1. Storyboard Game Tank 2D

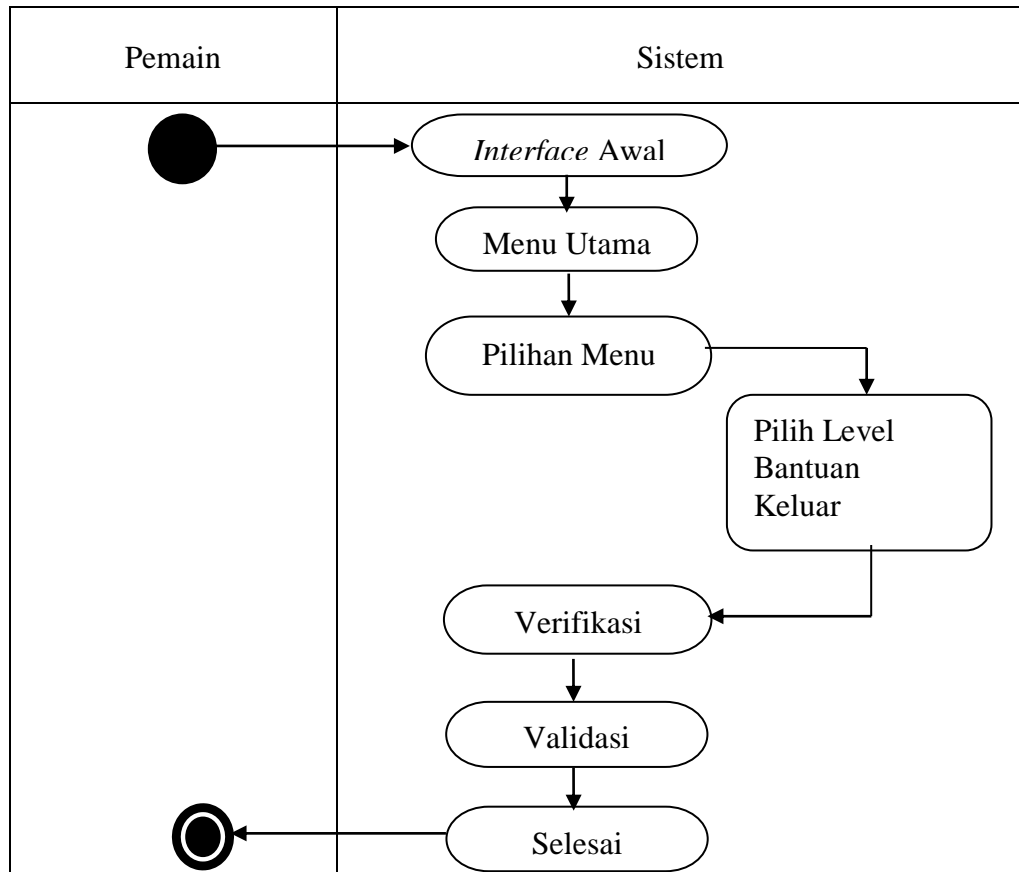
Nama Objek	Gambar	Deskripsi
Karakter utama		Karakter yang digerakkan oleh pemain untuk mengalahkan musuh yang ada.
Karakter musuh		Karakter yang akan bergerak menuju karakter utama dan menyerangnya sesuai dengan algoritma yang diterapkan.
<i>TileMap</i>		Gambar yang digunakan untuk mereferensikan bidang tanah pada <i>game</i> .
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.

III.3.3. 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 tank 2D* ini adalah :



Gambar III.3. Activity Diagram

III.4. Desain User Interface

1. Tampilan Menu Utama

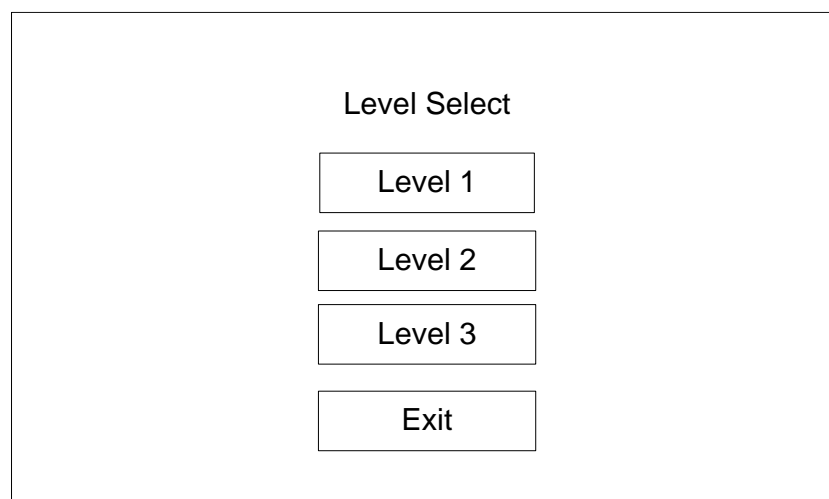
Tampilan awal ketika pemain memulai *game* ini adalah tampilan menu yang berisi beberapa tombol seperti tombol *Play*, tombol *Help*, dan tombol *Exit*.



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

2. Tampilan Menu Pilih Level

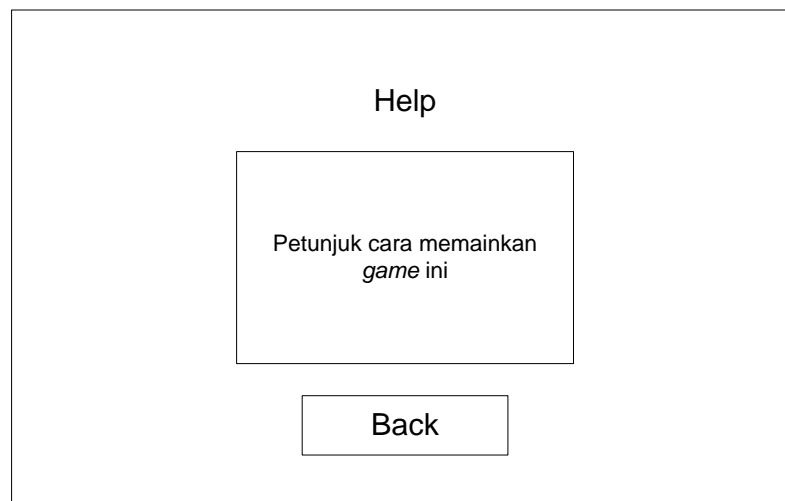
Saat pemain mengklik tombol *Play* maka menu untuk memilih level akan muncul. Pada menu ini terdapat 4 buah tombol yaitu 3 tombol untuk memilih level permainan dan 1 tombol untuk kembali ke menu utama.



Gambar III.5. Tampilan *Input Game*

3. Tampilan Menu Bantuan

Pada saat pemain mengklik tombol *Help* maka tampilan bantuan akan terbuka. Menu bantuan berisi petunjuk cara memainkan *game* ini.



Gambar III.6. Tampilan Menu Bantuan

4. Tampilan Permainan

Pada saat pemain mengklik salah satu tombol level dari menu pilih level maka permainan dimulai. Tampilan permainan terdapat teks yang berisi info mengenai jumlah nyawa yang dimiliki pemain dan tombol menu untuk kembali ke menu utama. Untuk menggerakkan karakter pemain dapat menggunakan tombol arah atau A, W, S, D dan menggunakan arah *mouse* serta klik untuk mengarahkan tembakan serta melakukan tembakan.



Gambar III.7. Tampilan Permainan

5. Tampilan *Game Over*

Pada saat poin pemain mencapai angka nol, maka *game* berakhir dan layar tampilan *Game Over* akan muncul yang berisikan pesan *Game Over* dan tombol menu untuk kembali ke menu utama.



Gambar III.8. Tampilan *Game Over*