

## BAB IV

### HASIL DAN UJI COBA

#### IV.1. Tampilan Hasil

Berikut adalah tampilan hasil dan uji coba dari aplikasi *game roguelike* berbasis *Android*:

##### IV.1.1. Tampilan *Stage*

Tampilan utama ini merupakan tampilan awal pada saat aplikasi dijalankan. Pada tampilan ini berupa layar berwarna hitam dengan teks yang menunjukkan *level stage* yang akan dimainkan oleh pemain, ketika pertama kali dijalankan maka menampilkan *level stage* 1 seperti terlihat pada Gambar IV.1.



**Gambar IV.1. Tampilan *Stage* Level 1**

Tampilan ini diperlukan untuk proses *loading* atau memuat *game* per *level stage*-nya dikarenakan pada setiap pergantian *stage*, *game* akan otomatis memuat *stage* yang akan dimainkan dan membutuhkan waktu untuk prosesnya. Waktu

untuk proses *loading* tergantung dari performa perangkat yang digunakan. Setelah proses *loading* selesai maka tampilan *game* akan muncul dan pemain dapat menggerakkan karakter utama.

#### IV.1.2. Tampilan Awal *Game*

Dalam *game* ini digunakan *map* berbasis *tile* dengan setiap *tile* menyimpan satu objek dengan luas 20 x 20 *tile*. Untuk objek pada *game* yaitu penghalang pergerakan karakter berupa *sprite* tumbuhan dan batu, serta musuh dan *item* dimana jumlah maupun letaknya dibuat *random* yang diatur oleh *script* (*procedural generate*) sehingga bentuk dari setiap level dan setiap kali permainan diulang bentuk *map* akan berbeda.



**Gambar IV.2. Tampilan Map *Game***

Letak karakter utama pada *game* ini berada pada sudut kiri bawah setiap kali akan memulai *stage* dengan kamera yang berfokus pada karakter utama. Pada *User Interface game* terdapat dua buah *button* yaitu *help* yang dilambangkan dengan *button* bulat kecil dengan tanda tanya (?) dan *exit* yang dilambangkan dengan *button* bulat kecil dengan tanda X, serta teks yang menunjukkan info tentang *food* dan *hp* yang dimiliki oleh karakter utama



**Gambar IV.3. Tampilan Awal Game**

Untuk menggerakkan karakter utama pemain harus mengusap layar sesuai arah yang diinginkan pemain dan setiap kali pemain melakukan percobaan pergerakan maka *point* dari *food* akan berkurang satu. Dan untuk memulihkan *point food* pemain harus mengambil makanan yang tersedia.



**Gambar IV.4. Tampilan Karakter Utama dan Makanan**

Untuk mengambil makan pemain harus mengarahkan karakter utama menuju letak dari *sprite* makanan dan setelah pemain sampai pada makanan tersebut objek makanan tersebut hilang dan *point food* akan bertambah dengan nilai acak (nilai antara 10 – 40) dan muncul teks yang memberi info tentang *point food* yang didapat.



**Gambar IV.5. Tampilan Penambahan *Point Food***

### IV.1.3. Tampilan Pertarungan

*Game* ini menggunakan prinsip permainan bergiliran (*turn based*) dimana dalam *game* ini musuh akan melakukan percobaan pergerakan setelah pemain melakukan pergerakan sebanyak dua kali. Untuk menyerang musuh pemain harus menggerakkan karakter utama mendekati musuh setelah itu pemain mengusap layar sesuai arah dimana musuh berada. Karakter utama akan memukul musuh dan memberikan *damage* dengan nilai acak sesuai dengan level yang dimilikinya (nilai antara  $(1 + \text{level pemain}) - (5 + (\text{level pemain} \times 2))$ ). Dan muncul teks yang memberi info tentang *damage* yang dihasilkan.



**Gambar IV.6. Tampilan Karakter Utama Memukul Musuh**

Setiap kali pemain mengalahkan musuh maka pemain akan mendapatkan *exp point* (*experience point*) yang digunakan untuk menaikkan level karakter utama.



**Gambar IV.7. Tampilan Kenaikan Level Karakter**

Setelah melakukan percobaan gerakan selama dua kali selanjutnya giliran musuh yang akan bergerak, musuh akan mencari dan menyerang pemain dengan pergerakan sesuai dengan AI (*artificial intelligence*) yang telah diterapkan sebelumnya. *Damage* yang diberikan oleh musuh ke karakter utama memiliki nilai acak dengan parameter sesuai dengan level dari musuh tersebut, jadi semakin besar level musuh maka semakin besar kemungkinan *damage* yang dihasilkan oleh musuh. *Point damage* yang diberikan kepada musuh akan mengurangi *point hp* karakter utama.



**Gambar IV.8. Tampilan Musuh Menyerang Karakter Utama**

Untuk memulihkan *point hp* dari karakter utama, pemain harus mengambil *potion*, setelah *potion* berhasil diambil maka objek *potion* akan hilang dan muncul teks yang memberikan info mengenai *point hp* yang didapat oleh karakter utama.



**Gambar IV.9. Tampilan Karakter Utama dan *Potion***



**Gambar IV.10. Tampilan Penambahan *Point Hp***

#### **IV.1.4. Tampilan Kenaikan *Level Stage***

Untuk dapat mencapai *stage* selanjutnya pada *game* ini, pemain harus mengarahkan karakter utama menuju pintu keluar yang berupa *sprite* bunga berwarna *pink*. Pada awal permainan objek bunga tersebut belum muncul, untuk memunculkan objek bunga tersebut pemain harus terlebih dahulu mengalahkan semua musuh yang ada. Setelah objek bunga tersebut muncul maka pemain dapat mengarahkan karakter utama menuju bunga tersebut untuk mencapai *stage* selanjutnya.



**Gambar IV.11. Tampilan Karakter Utama dan Pintu Keluar**

Setelah karakter utama berhasil menuju pintu keluar selanjutnya *scene* akan di-*reset* dan muncul tampilan *stage* yang menunjukkan info *level stage* selanjutnya yang akan dimainkan oleh pemain.



**Gambar IV.12. Tampilan *Stage* Level 2**

#### IV.1.5. Tampilan Menu *Help*

Di menu ini pemain disajikan cara memainkan *game* ini dan terdapat dua buah tombol yaitu tombol *Close* dan tombol Profil. Tombol *Close* berfungsi untuk menutup menu *help* dan kembali ke permainan dan tombol Profil berfungsi untuk menampilkan menu Profil yang berisi informasi dari pembuat *game* ini.



Gambar IV.13. Tampilan Menu Help

#### IV.1.6. Tampilan Menu *Exit*

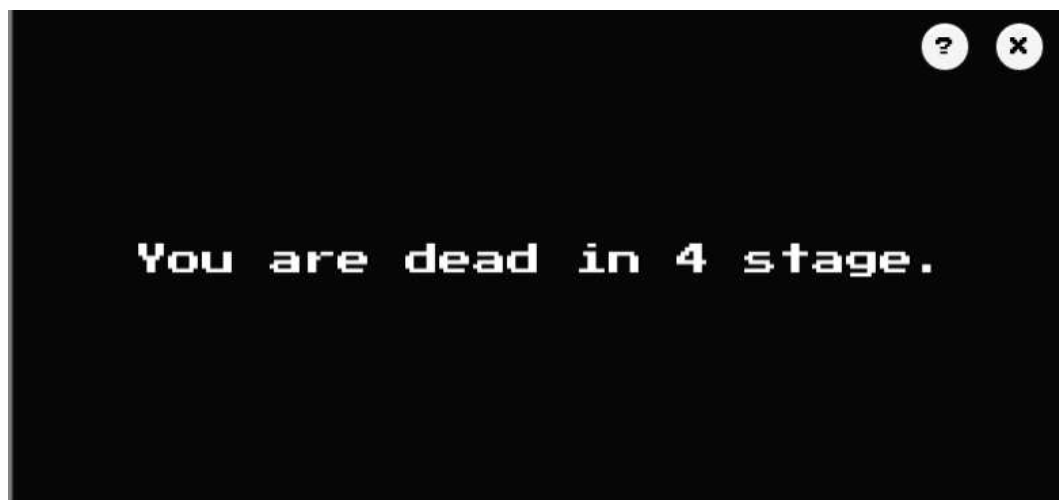
Menu ini akan muncul jika pemain menekan tombol *exit*. Menu ini berisi teks yang menanyakan konfirmasi untuk keluar dari *game*, terdapat 2 buah tombol yaitu tombol *yes* untuk keluar dari *game* dan tombol *no* untuk menutup menu ini dan melanjutkan permainan.



**Gambar IV.14. Tampilan Menu *Exit***

#### **IV.1.7. Tampilan *Game Over***

Tampilan ini akan muncul apabila *point food* atau *point hp* dari karakter utama mencapai nilai 0. Tampilan ini menandakan permainan berakhir dan pemain harus menekan tombol *exit* untuk keluar dan memainkan *game* ini dari awal.



**Gambar IV.15. Tampilan *Game Over***

## IV.2. Pembahasan

Aplikasi *game roguelike* ini dibuat menggunakan *software game engine* Unity 5 dan bahasa pemrograman C#. Dalam pembuatan *game roguelike* ini hanya terdapat satu buah *scene*, yaitu *main scene*. Pada dasarnya, *infinite level* pada *game* ini berupa satu buah *scene* yang diulang dengan penambahan kesulitan berupa penambahan level dari karakter musuh, dimana level tersebut berpengaruh terhadap jumlah *hp* dan *damage* dari karakter musuh.

Objek gambar yang digunakan pada *game* ini berupa *sprite* 2D yang dibuat menggunakan *software Adobe Photoshop* CS4 dan hasil dari pembuatan gambar tersebut dikonversi ke dalam bentuk *.png* dan di-*import* ke dalam Unity 5 untuk disesuaikan agar dapat ditampilkan secara baik di dalam perangkat *Android*. File *audio* yang digunakan berasal dari file *audio* yang diunduh dari internet yang kemudian dikonversikan ke bentuk *.ogg* menggunakan *software Format Factory* untuk memperkecil ukuran dari file *audio* tersebut sehingga ukuran dari *game* ini tidak terlalu besar mengingat *game* ini akan dijalankan pada perangkat *Android* yang memiliki keterbatasan penyimpanan dan memori dibanding komputer.

Aplikasi *game* ini berekstensi *.apk* dan hanya dapat dijalankan pada perangkat *Android* dengan versi minimal 4.1 (*Jellybean*). Namun *game* ini dapat dibuat versi untuk *platform Windows, MAC, Linux, serta IOS*.

### IV.3. Hasil Uji Coba

#### IV.3.1. Analisa Hasil

Adapun analisa hasil yang didapatkan berdasarkan pengujian dengan metode *black box* adalah sebagai berikut:

**Tabel IV.1. Uji Coba Program**

No	Nama Pengujian	Keterangan	Hasil yang Diharapkan	Hasil yang Didapat
1	Run Aplikasi	Cek hasil aplikasi	Aplikasi berjalan dengan lancar	OK
2	Tes Fungsi GUI <i>Button</i>	Cek GUI <i>Button</i>	Aplikasi bisa menampilkan layer sesuai dengan <i>Button</i> yang dipilih	OK
3	Tes Fungsi <i>Swipe</i>	Mengontrol pergerakan karakter utama	Aplikasi dapat menggerakkan karakter utama sesuai arah <i>swipe</i> yang dilakukan oleh pemain	OK
4	Tes Kontrol Permainan	Mengontrol permainan pada <i>game</i> sesuai dengan peraturan yang ada pada <i>game</i>	Aplikasi dapat menjalankan kontrol permainan dalam <i>game</i>	OK
5	Tes Algoritma	Mengontrol pergerakan musuh pada <i>game</i> sesuai dengan algoritma yang diterapkan	Musuh pada <i>game</i> dapat bergerak sesuai dengan algoritma yang telah diterapkan	OK
6	Test <i>Audio</i>	Cek <i>Audio</i>	Aplikasi dapat menjalankan <i>audio</i> sesuai dengan kondisi yang ditentukan	OK
7	Keluar Permainan	Keluar dari Permainan	Aplikasi dapat menjalankan perintah keluar dari permainan	OK

### IV.3.2. Spesifikasi

Dalam menjalankan aplikasi *game roguelike* ini, terdapat spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam menjalankan aplikasi tersebut adalah sebagai berikut:

#### 1. Perangkat Keras (*Hardware*)

Perangkat keras minimum yang digunakan untuk menjalankan aplikasi ini adalah sebagai berikut:

- a. Perangkat *Android* dengan prosesor 1,2 GHz
- b. RAM 512 MB
- c. Kapasitas Penyimpanan 512 MB

Perangkat keras minimum yang digunakan untuk merancang aplikasi ini adalah sebagai berikut:

- a. *Personal Computer (PC)* dengan *processor Core i3 2,2 Ghz*
- b. RAM 2 GB
- c. *Harddisk 500 GB*

#### 2. Perangkat Lunak (*Software*)

Perangkat lunak yang dibutuhkan untuk menjalankan aplikasi ini adalah sebagai berikut:

- a. Sistem Operasi *Android: Jellybean (4.1)*

Perangkat lunak yang digunakan untuk merancang aplikasi ini adalah sebagai berikut:

- a. *Unity Engine 5*
- b. *MonoDevelop 4.0.1*
- c. *Android SDK (Software Development Kit)*
- d. *Adobe Photoshop CS4*

#### **IV.4. Kelebihan dan Kekurangan**

Aplikasi *game roguelike* ini tidak luput dari beberapa kelebihan dan kekurangan. Kedua aspek itu sangat berpengaruh terhadap kinerja dan tujuan yang dihasilkan, sehingga pengguna mengetahuinya lebih lanjut.

##### **IV.4.1. Kelebihan dari sistem yang dirancang**

- a. Aplikasi *game* ini menggunakan *single scene* sehingga ukuran dari *game* ini tidak terlalu besar.
- b. Aplikasi *game* ini mempunyai fitur *swipe* untuk menggerakkan karakter utama sehingga memaksimalkan penggunaan perangkat *Android* dalam menjalankan *game* ini.

##### **IV.4.2. Kekurangan dari sistem yang dirancang**

- a. Aplikasi *game* yang dibuat hanya memiliki fitur pertarungan sederhana dimana faktor penting di dalam pertarungan tersebut hanya berdasarkan level karakter.

- b. AI yang digunakan pada karakter musuh menggunakan konsep algoritma *pathfinding brute force* dimana pada algoritma ini hanya mendeteksi jarak musuh dengan karakter utama, sehingga musuh tidak dapat bergerak melewati halangan yang terdapat pada *map*.