



**TUGAS AKHIR - KI141502**

**RANCANG BANGUN PEMBANGKIT *WORLD*  
DINAMIS PADA *TURN-BASED STRATEGY GAME*  
*ANCIENT EMPIRES 2* PADA *ANDROID***

Wiby Mahan Faqih  
NRP 5111100 025

Dosen Pembimbing  
Imam Kuswardayan, S.Kom., M.T.  
Ridho Rahman Hariadi, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2015

*[Halaman ini sengaja dikosongkan]*



**FINAL PROJECT - KI141502**

# **DESIGN AND IMPLEMENTATION OF DYNAMIC WORLD GENERATOR IN TURN-BASED STRATEGY GAME ANCIENT EMPIRES 2 ON ANDROID**

Wiby Mahan Faqih  
NRP 5111100 025

Advisor  
Imam Kuswardayan, S.Kom., M.T.  
Ridho Rahman Hariadi, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA  
2015

*[Halaman ini sengaja dikosongkan]*

**LEMBAR PENGESAHAN**

**RANCANG BANGUN PEMBANGKIT *WORLD DYNAMIS* PADA *TURN-BASED STRATEGY GAME ANCIENT EMPIRES 2* PADA *ANDROID***

**Tugas Akhir**

**Dijadikan Untuk Memenuhi Salah Satu Syarat Memperoleh  
Gelara Sarjana Komputer  
pada  
Rampun Mata Kuliah Interaksi, Grafika, dan Seni  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember**

Oleh:

**WIBY MAHAN FAQIH**

**NRP 5111 100 025**

Disetujui oleh Dosen Pembimbing Tugas Akhir

Iman Karwardayan, S.Kom., M.T.

NRP 29761215 200312 1 001

(pembimbing 1)

Wahid Rahman Hariadi, S.Kom., M.Sc.

NRP 2970213 201404 1 001

(Pembimbing 2)

**SURABAYA**

**JUNI, 2015**

*[Halaman ini sengaja dikosongkan]*

## **RANCANG BANGUN PEMBANGKIT WORLD DINAMIS PADA *TURN-BASED STRATEGY GAME ANCIENT EMPIRES* 2 PADA ANDROID**

Nama Mahasiswa : Wiby Mahan Faqih  
NRP : 5111 100 025  
Jurusan : Teknik Informatika FTIf-ITS  
Dosen Pembimbing I : Imam Kuswardayan, S.Kom., M.T.  
Dosen Pembimbing II : Ridho Rahman Hariadi, S.Kom., M.Sc.

### **ABSTRAK**

*Saat ini game memiliki perkembangan yang sangat pesat. Game saat ini juga tidak hanya pada perangkat komputer namun juga telah merambah ke perangkat mobile khususnya dalam sistem operasi Android.*

*Pada perkembangannya selain bisa dimainkan secara bersama-sama, game juga dituntut untuk bisa memberikan tantangan bagi pengguna berupa level permainan. Untuk mendapatkan tantangan dari level permainan, digunakan pembangkit world dinamis.*

*Dalam tugas akhir ini akan dibangun suatu permainan bergenre *turn-based strategy* yang menggunakan pembangkit world menggunakan algoritma *Diamond-Square* yang merupakan pengembangan dari algoritma *Midpoint-Displacement* yang menggabungkan antara kecepatan dari algoritma *Midpoint-Displacement* dan kualitas hasil dari algoritma *Value Noise*. Algoritma ini dimulai dengan mengacak angka dari empat sudut persegi, kemudian mendapatkan nilai titik tengah dari persegi tersebut dengan menghitung rata-rata dari empat angka sudut tersebut kemudian ditambah angka acak. Kemudian mendapatkan empat titik tengah dari keempat sisi persegi. Kemudian diulangi langkah tersebut hingga semua titik terisi.*

*Dari hasil uji performa diketahui bahwa permainan yang telah dibangun berhasil membangkitkan world dengan waktu minimal 0,057 detik dan maksimal sampai 0,267 detik. Dan memiliki total*

*rata-rata 0,106 detik dari 20 kali percobaan. Dan juga fungsionalitas dari permainan ini sudah sesuai dengan apa yang diharapkan.*

***Kata kunci: Pembangkit World Dinamis, Turn-Based Strategy, Diamond-Square Algorithm, Perangkat Mobile, Sistem Operasi Android.***

# **DESIGN AND IMPLEMENTATION OF DYNAMIC WORLD GENERATOR IN TURN-BASED STRATEGY GAME ANCIENT EMPIRES 2 ON ANDROID**

Student Name : Wiby Mahan Faqih  
NRP : 5111 100 025  
Major : Teknik Informatika FTIf-ITS  
Advisor I : Imam Kuswardayan, S.Kom., M.T.  
Advisor II : Ridho Rahman Hariadi, S.Kom., M.Sc.

## **ABSTRACT**

*Currently the game has a very rapid development . Game today not only on computers but also have reached to a mobile device, especially in the Android operating system.*

*On its development, not only be able to play together, the game also required to be able to provide a challenge for the user in the form of a game level . To get a challenge from the level of the game, used dynamic world generators.*

*In this final project built an turn-based strategy game that uses dynamic world generators using diamond-square algorithm which is the development of Midpoint-Displacement algorithm that combines the speed of the Midpoint-Displacement algorithm and quality of the results of the Value Noise algorithm. This algorithm begins to randomize the numbers from the four corners of the square , then get the value of the midpoint of the square by calculating the average of the four corner numbers are then added random numbers . Then get four midpoints of the four sides of a square . Then repeat these steps until all the points are filled.*

*From the results of the performance test is known that the game has been built successfully generated world with a time of 0.057 sec minimum and a maximum of up to 0.267 seconds. And have an average total 0.106 seconds from 20 attempts. And also the functionality of this game is in conformity with what was expected.*

***Keywords: Dynamuc World Generator, Turn-Based Strategy, Diamond-Square Algorithm, Mobile Device, Android OS.***

## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Allah SWT atas karunia dan kesempatan yang diberikanNya. Atas ijinNya, penulis akhirnya dapat menyelesaikan tugas akhir yang berjudul “Rancang Bangun Pembangkit World Dinamis pada Turn-Based Strategy Game Ancient Empires 2 pada Android”.

Proses pengerjaan tugas akhir ini juga tidak lepas dari bantuan berbagai pihak, baik secara langsung maupun tidak. Maka dari itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang tulus dan sebesar-besarnya kepada semua pihak, khususnya:

1. Ayah penulis, Abdur Rosyid, dan Ibu penulis, Nu Afifah, atas doa dan kasih sayangnya yang tulus serta motivasi, semangat dan dukungan yang tiada hentinya sehingga dapat mengantarkan penulis untuk menyelesaikan masa studinya.
2. Adik penulis, Nabial Ihlasi, yang selalu memberikan doa, semangat, perhatian dan bantuan kepada penulis.
3. Bapak Muchammad Husni, selaku dosen wali penulis, yang selalu memberikan bimbingan, dukungan serta motivasi selama masa perkuliahan di Teknik Informatika ITS.
4. Bapak Imam Kuswardayan selaku dosen pembimbing Tugas Akhir pertama dan yang telah memberikan arahan dalam pengerjaan Tugas Akhir ini.
5. Bapak Ridho Rahman Hariadi selaku dosen pembimbing Tugas Akhir kedua yang dengan sabar membimbing penulis dalam pengerjaan Tugas Akhir ini.
6. Segenap dosen dan karyawan Teknik Informatika ITS atas ilmu dan pengalaman yang telah diberikan selama penulis menjalani masa studi di Teknik Informatika ITS.
7. Akbar, Ampuh, Habibi, Ibnu, Ilmi, Wahyu, Yudi dan teman-teman seperjuangan TC 2011 yang selalu memberikan bantuan, semangat, canda tawa serta kebersamaan selama masa perkuliahan.

8. Arsega, Jaeger, Janken, kimkimkim, SOE, SSS, ZackFire, Zarkbarz, dan teman-teman yang setia menemani penulis bermain Dota dikala penulis sedang bosan. GGWP.
9. Awim, Dila, Faid, Irwan, Khoir, Roshif, Shofi, dan teman-teman dari PB Klinteng yang selalu memberikan semangat kepada penulis dalam pengerjaan Tugas Akhir ini.

Akhirnya penulis menyadari masih banyak kekurangan dalam tugas akhir ini. Untuk itu penulis mengharapkan adanya kritik dan saran yang dapat menyempurnakan tugas akhir ini. Harapan penulis, semoga tugas akhir ini dapat bermanfaat bagi pembaca.

Surabaya, Juni 2015  
Penulis

Wiby Mahan Faqih

## DAFTAR ISI

LEMBAR PENGESAHAN .....	vii
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
DAFTAR KODE SUMBER.....	xxi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Permasalahan .....	2
1.3. Batasan Permasalahan.....	2
1.4. Tujuan .....	2
1.5. Manfaat .....	2
1.6. Metodologi.....	3
1.7. Sistematika Penulisan .....	4
BAB II TINJAUAN PUSTAKA .....	7
2.1. Unity (Game Engine).....	7
2.2. Turn-Based Strategy .....	8
2.3. C#.....	9
2.4. Ancient Empires 2.....	10
2.5. Diamond-Square Algorithm.....	10
2.6. Android .....	10
BAB III ANALISIS DAN PERANCANGAN .....	11
3.1. Analisis Sistem.....	11
3.2. Perancangan Permainan .....	11
3.2.1. Deskripsi Umum Perangkat Lunak .....	12
3.2.2. Spesifikasi Kebutuhan Fungsional.....	12
3.2.3. Spesifikasi Kebutuhan Non-Fungsional.....	12
3.2.4. Karakteristik Pemain.....	13
3.3. Perancangan Sistem .....	13
3.3.1. Perancangan Pembangkit World.....	13

3.3.2.	Perancangan Finite State Machine.....	22
3.3.3.	Perancangan Diagram Kasus Penggunaan.....	23
3.3.4.	Perancangan Skenario Kasus Penggunaan .....	23
3.3.5.	Perancangan Antarmuka Pengguna .....	32
3.3.6.	Perancangan Kontrol Permainan .....	36
3.3.7.	Perancangan Aturan Permainan.....	36
3.3.8.	Perancangan Skenario Permainan .....	37
BAB IV	IMPLEMENTASI .....	39
4.1.	Lingkungan Implementasi .....	39
4.2.	Implementasi Permainan .....	39
4.2.1.	Implementasi Pembangkit World .....	39
4.2.2.	Implementasi Menjalankan Pasukan .....	45
4.2.3.	Implementasi Membeli Pasukan.....	50
BAB V	PENGUJIAN DAN EVALUASI .....	53
5.1.	Lingkungan Uji Coba .....	53
5.2.	Pengujian Performa Kecepatan Pembangkit World ....	53
5.2.1.	Skenario Data Uji Coba Kecepatan .....	54
5.2.2.	Hasil Pengujian Performa .....	54
5.3.	Pengujian Kesesuaian World Yang Dibangkitkan .....	56
5.3.1.	Skenario Data Uji Coba Kesesuaian <i>World</i> .....	57
5.3.2.	Hasil Pengujian Kesesuaian <i>World</i> .....	57
5.4.	Pengujian Fungsionalitas dengan Metode Black-Box .	61
5.4.1.	Skenario Pengujian Fungsionalitas.....	61
5.5.	Pengujian Pengguna .....	72
5.5.1.	Skenario Uji Coba Pengguna.....	72
5.5.2.	Daftar Penguji Perangkat Lunak.....	73
5.5.3.	Hasil Uji Coba Pengguna .....	73
5.5.4.	Hasil Pengujian Pengguna .....	75
BAB VI	KESIMPULAN DAN SARAN .....	77
5.6.	Kesimpulan.....	77
5.7.	Saran.....	78
DAFTAR PUSTAKA	.....	79
LAMPIRAN	.....	81
BIODATA PENULIS	.....	91

## DAFTAR TABEL

Tabel 3.1 Karakteristik Pengguna.....	13
Tabel 3.2 Skenario Kasus Penggunaan.....	24
Tabel 3.3 Skenario Kasus Penggunaan Menjalankan Pasukan.....	25
Tabel 3.4 Skenario Kasus Penggunaan Membeli Pasukan.....	26
Tabel 3.5 Skenario Kasus Penggunaan Menyerang Pasukan Lawan.....	26
Tabel 3.6 Skenario Kasus Penggunaan Menguasai Kastil dan Rumah.....	27
Tabel 3.7 Skenario Kasus Penggunaan Mengakhiri Giliran.....	27
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak.....	39
Tabel 5.1 Lingkungan Uji Coba Perangkat Lunak (bagian 1).....	53
Tabel 5.2 Lingkungan Uji Coba Perangkat Lunak (bagian 2).....	53
Tabel 5.3 Skenario Pengujian Performa.....	54
Tabel 5.4 Hasil Uji Coba Performa Pembangkit World.....	56
Tabel 5.5 Skenario Pengujian <i>World</i> .....	57
Tabel 5.6 Hasil Uji Coba Kesesuaian <i>World</i> Yang Dibangkitkan...	60
Tabel 5.7 Pengujian Permainan.....	61
Tabel 5.8 Hasil Pengujian Fungsionalitas.....	72
Tabel 5.9 Daftar Nama Penguji Coba Aplikasi.....	73
Tabel 5.10 Keterangan Nilai Pada Kuesioner.....	74
Tabel 5.11 Penilaian Antarmuka.....	74
Tabel 5.12 Penilaian Performa Sistem.....	75

*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

Gambar 3.1 Penjelasan Diamond-Square (bagian 1).....	16
Gambar 3.2 Penjelasan Diamond-Square (bagian 2).....	16
Gambar 3.3 Penjelasan Diamond-Square (bagian 3).....	17
Gambar 3.4 Penjelasan Diamond-Square (bagian 4).....	18
Gambar 3.5 Penjelasan Diamond-Square (bagian 5).....	19
Gambar 3.6 Penjelasan Diamond-Square (bagian 6).....	19
Gambar 3.7 Perbaikan Hasil (bagian 1).....	20
Gambar 3.8 Perbaikan Hasil (bagian 2).....	21
Gambar 3.9 Keterangan Nilai Pada <i>Tile</i> .....	21
Gambar 3.10 Implementasi Tiles.....	22
Gambar 3.11 Finite State Machine Main Menu.....	23
Gambar 3.12 Diagram Kasus Aplikasi.....	24
Gambar 3.13 Diagram Aktifitas Menjalankan Pasukan.....	28
Gambar 3.14 Diagram Aktifitas Membeli Pasukan.....	29
Gambar 3.15 Diagram Aktifitas Menyerang Pasukan Lawan.....	30
Gambar 3.16 Diagram Aktifitas Menguasai Rumah dan Kastil.....	31
Gambar 3.17 Diagram Aktifitas Mengakhiri Giliran.....	32
Gambar 3.18 Tampilan Main Menu.....	33
Gambar 3.19 Tampilan Saat Bermain.....	34
Gambar 3.20 Tampilan Toko Pasukan.....	35
Gambar 4.1 Isi Variabel Setelah Fungsi DiamondSquare Dijalankan.....	41
Gambar 4.2 Tampilan Hasil Pembangkitan Map 9x9.....	42
Gambar 4.3 Tampilan Hasil Pembangkitan Map 17x17.....	43
Gambar 4.4 Tampilan Hasil Pembangkitan Map 9x9 Menggunakan Nilai Random.....	44
Gambar 4.5 Tampilan Hasil Pembangkitan Map 17x17 Menggunakan Nilai Random.....	45
Gambar 4.6 Tampilan Implementasi Menjalankan Unit.....	48
Gambar 4.7 Tampilan Menyerang Unit Lawan.....	50
Gambar 4.8 Tampilan Membeli Unit.....	52
Gambar 5.1 Tampilan Hasil Kode Perhitungan Waktu.....	55

Gambar 5.2 <i>World Uji Coba 1</i> .....	57
Gambar 5.3 Nilai <i>World Uji Coba 1</i> .....	58
Gambar 5.4 <i>World Uji Coba 2</i> .....	58
Gambar 5.5 Nilai <i>World Uji Coba 2</i> .....	59
Gambar 5.6 <i>World Uji Coba 3</i> .....	59
Gambar 5.7 Nilai <i>World Uji Coba 3</i> .....	60
Gambar 5.8 Tampilan Skenario Permainan .....	62
Gambar 5.9 Tampilan Awal Permainan .....	63
Gambar 5.10 Tampilan Menu Setelan .....	64
Gambar 5.11 Tampilan Menu Instruksi dan Tentang .....	65
Gambar 5.12 Tampilan Menguasai Rumah .....	66
Gambar 5.13 Tampilan Membeli Pasukan .....	67
Gambar 5.14 Tampilan Menjalankan Pasukan .....	67
Gambar 5.16 Tampilan Menyerang Pasukan Lawan (bagian 1) .....	68
Gambar 5.17 Tampilan Menyerang Pasukan Lawan (bagian 2) .....	69
Gambar 5.18 Mengakhiri Permainan .....	70
Gambar 5.19 Tampilan Menambah <i>Health Point</i> .....	71

## DAFTAR KODE SUMBER

Kode Sumber 3.1 Algoritma <i>Diamond-Square</i> .....	15
Kode Sumber 4.1 Fungsi DiamondSquae.....	40
Kode Sumber 4.2 Fungsi FillDiamondSquare.....	40
Kode Sumber 4.3 Fungsi GeneratedMap.....	42
Kode Sumber 4.4 Pemanggilan Fungsi Cek .....	45
Kode Sumber 4.5 Fungsi Cek .....	46
Kode Sumber 4.6 Fungsi Moving.....	47
Kode Sumber 4.7 Fungsi IEnumerator MoveToPosition.....	47
Kode Sumber 4.8 Fungsi Serang .....	49
Kode Sumber 4.9 Fungsi Beli.....	51
Kode Sumber 5.1 Menghitung Waktu Fungsi GenerateMap.....	54

*[Halaman ini sengaja dikosongkan]*

# BAB I PENDAHULUAN

Pada bagian ini akan dijelaskan beberapa hal dasar mengenai tugas akhir ini yang meliputi: latar belakang, tujuan, manfaat permasalahan, batasan permasalahan, metodologi serta sistematika penulisan tugas akhir. Penjelasan tentang hal-hal tersebut diharapkan dapat memberikan gambaran umum mengenai permasalahan sehingga penyelesaian masalah dapat dipahami dengan baik.

## 1.1. Latar Belakang

Perkembangan teknologi komputer pada saat ini sangat pesat, dimana yang telah kita ketahui dalam instansi pemerintahan maupun swasta, lebih mengutamakan menggunakan teknologi komputer dalam menyelesaikan pekerjaannya, dan juga bagi para pelajar atau mahasiswa dan atau mahasiswi serta dosen dalam mempermudah proses belajar mengajar.

Salah satu hasil dari perkembangan teknologi ini adalah *game*. *Game* merupakan sebuah alat rekreasi yang sangat populer dimasyarakat. Hampir seluruh lapisan masyarakat saat ini pernah memainkan sebuah *game*.

Pada perkembangannya selain bisa dimainkan secara bersama-sama, *game* juga dituntut untuk bisa memberikan tantangan bagi pengguna berupa level permainan. Untuk mendapatkan tantangan dari level permainan, digunakan pembangkit *world* dinamis.

Dengan digunakannya pembangkit *world* dinamis, diharapkan memberikan tantangan baru bagi pemain agar tidak menyebabkan mudah bosan. Tugas akhir ini lebih difokuskan pada pembuatan pembangkit *world* dinamis yang diterapkan dalam *turn-based strategy game* yang bisa memberikan tantangan yang baru setiap levelnya.

## 1.2. Rumusan Permasalahan

Rumusan masalah yang akan diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana aturan main, level dan skenario permainan pada *Ancient Empires 2*?
2. Bagaimana menganalisa pola tingkat kesulitan dan skenario permainan pada *Ancient Empires 2*?
3. Bagaimana mengimplementasi metode pembangkit world dinamis pada *Ancient Empires 2*?

## 1.3. Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Aplikasi yang dibangun menggunakan *game engine Unity*.
2. Aplikasi yang dibangun berbasis platform mobile *android* versi 4.2 keatas.
3. Aplikasi yang dibangun melalui komponen objek dua dimensi.
4. Aplikasi yang dibangun mengadopsi aturan main dari *Ancient Empires 2*.

## 1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah membangun *game Ancient Empires 2* pada *android* yang mempunyai fitur pembangkit *world* dinamis.

## 1.5. Manfaat

Manfaat dari hasil pembuatan tugas akhir ini antara lain:

1. Pengguna mendapat kepuasan bermain dari *world* baru.
2. Dapat meningkatkan pemikiran strategis pada pengguna.
3. Memberikan media hiburan bagi pengguna.
4. Dapat digunakan sebagai eksplorasi untuk *development game* kedepan.

## 1.6. Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

### A. Studi literatur

Tahap Studi Literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan Tugas Akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi dan pustaka, serta pemahaman dasar teori yang digunakan pada topik tugas akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. *Diamond Square Algorithm*;
2. *Turn-Based Strategy*;
3. Instalasi pada *Platform Perangkat Mobile Android*;
4. *Artificial Intelligence*;
5. *Tile-based Game*.

### B. Perancangan perangkat lunak

Langkah yang akan digunakan pada tahap ini adalah sebagai berikut:

1. Pencarian dan pendataan materi yang akan digunakan dalam *game Ancient Empires 2*.
2. Perancangan sistem dan mekanisme *game Ancient Empires 2*.
3. Analisis kebutuhan non fungsional.
4. Analisis algoritma dan formula dalam membangun *game Ancient Empires 2*.
5. Perancangan pembangkit *world* dinamis pada *game Ancient Empires 2*.

### C. Implementasi dan pembuatan sistem

Aplikasi ini akan dibangun dengan bahasa pemrograman C# menggunakan *game engine Unity*. Aplikasi yang akan dibangun berbasis perangkat *mobile Android*.

### D. Pengujian dan evaluasi

#### 1. Pengujian blackbox

Pengujian blackbox adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat

mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program. Pengujian ini dilakukan untuk menguji apakah proses kinerja aplikasi *game* ini sudah sesuai dengan kebutuhan pengguna atau tidak [1].

## 2. Pengujian usabilitas

Pengujian usabilitas dilakukan dengan cara melakukan survei ke pengguna yaitu beberapa pengguna yang suka bermain *game* di sekitar lingkungan ITS. Survei dilakukan untuk mengukur tingkat kegunaan dari aplikasi yang dibuat dalam membantu pengguna.

## E. Penyusunan laporan tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan tugas akhir.

## 1.7. Sistematika Penulisan

Buku tugas akhir ini terdiri dari beberapa bab, yang dijelaskan sebagai berikut.

### **BAB I PENDAHULUAN**

Bab ini berisi latar belakang masalah, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan tugas akhir, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

### **BAB II TINJAUAN PUSTAKA**

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir ini.

### **BAB III ANALISIS DAN PERANCANGAN**

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas

rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, arsitektur, data, dan antarmuka.

#### **BAB IV IMPLEMENTASI**

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembangkitan area permainan, dan antarmuka permainan.

#### **BAB V PENGUJIAN DAN EVALUASI**

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

#### **BAB VI PENUTUP**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

*[Halaman ini sengaja dikosongkan]*

## BAB II TINJAUAN PUSTAKA

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan tugas akhir. Teori-teori tersebut adalah *Unity (Game Engine)*, *Turn-Based Strategy*, *C#*, *Ancient Empires 2*, *Diamond-Square Algorithm*, dan *Android*.

### 2.1. *Unity (Game Engine)*

*Unity 3D* adalah sebuah *game engine* yang berbasis *cross-platform*. *Unity* dapat digunakan untuk membuat sebuah *game* yang bisa digunakan pada perangkat komputer, ponsel pintar android, iPhone, PS3, dan bahkan X-BOX.

*Unity* adalah sebuah *tool* yang terintegrasi untuk membuat game, arsitektur bangunan dan simulasi. *Unity* bisa untuk PC *game* dan *Online game*. Untuk *Online game* diperlukan sebuah plugin, yaitu *Unity Web Player*, sama halnya dengan *Flash Player* pada Browser.

*Unity* tidak dirancang untuk proses desain atau modelling, dikarenakan *unity* bukan tool untuk mendesain. Jika ingin mendesain, penggunaan 3D editor lain seperti *3dsmax* atau *Blender*. Banyak hal yang bisa dilakukan dengan *unity*, ada fitur *audio reverb zone*, *particle effect*, dan *Sky Box* untuk menambahkan langit.

Fitur scripting yang disediakan, mendukung 3 bahasa pemrograman, JavaScript, C#, dan Boo. Fleksibel dan *EasyMoving*, rotating, dan scaling objects hanya perlu sebaris kode. Begitu juga dengan *Duplicating*, *removing*, dan *changing properties*. *Visual Properties Variables* yang di definisikan dengan *scripts* ditampilkan pada Editor. Bisa digeser, *drag and drop*, bisa memilih warna dengan *color picker*. Berbasis .NET, artinya perjalanan program dilakukan dengan *Open Source .NET platform*, Mono.

Pada setiap *project Unity* terdapat sebuah *Assets folder*. Isi dari *Assets folder* ditampilkan dalam bentuk panel *project* dalam *editor unity*. *Assets folder* adalah tempat untuk menyimpan semua

komponen dari game seperti tingkatan game (*level scenes*), *scripts*, *3D models*, *teksture*, dan file audio.

Untuk menambahkan *assets* ke dalam *project*, cukup dengan menarik (*drag*) file yang ingin ditambahkan ke dalam panel *project*. Atau dengan memilih menu *Assets->Import New Asset*. Untuk membuat *scene* baru, gunakan tombol *Control-N* (pada keyboard). Untuk menyimpan *scene* yang sedang aktif, gunakan *Control-S* (pada keyboard).

Panel *Hierarchy* menampung semua *GameObject* yang terdapat di *Scene* yang sedang aktif. Beberapa dari *GameObject* tersebut berhubungan langsung ke *asset* seperti objek 3D. Objek yang terdapat pada *hierarchy* dapat di seleksi dan dihapus. Jika objek dihapus atau ditambahkan pada *scene*, maka objek tersebut juga akan hilang atau muncul pada *hierarchy*.

*Unity* menggunakan sebuah konsep yang disebut *Parenting*. Ini digunakan untuk membuat sebuah *GameObject* menjadi anak dari *GameObject* yang lain. Tarik sebuah *GameObject* dan pindahkan tepat di atas tulisan *GameObject* yang akan dijadikan *parent* dalam *hierarchy*. *GameObject* yang terdapat dalam sebuah *GameObject* lainnya akan mengikuti perpindahan dan perputaran ketika *GameObject parent* mengalami perubahan posisi [2].

## 2.2. Turn-Based Strategy

*Turn-Based Strategy* (TBS) *game* adalah *game* strategi dimana pemain mempunyai giliran untuk melakukan suatu aksi, kemudian menyelesaikan gilirannya. Setelah giliran pemain tersebut selesai, maka pemain tersebut harus menunggu giliran selanjutnya untuk melakukan aksi selanjutnya [3].

Hal inilah yang menjadi daya tarik *game turn-based strategy*, sebab pemain harus mengatur strategi yang tepat sebelum bertindak untuk dapat memenangkan permainan. Salah satu *game turn based strategy* yang cukup dikenal adalah *King's Bounty*. *King's Bounty* adalah *game turn based strategy* yang didesain oleh Jon Van Canegham dari *New World Computing* pada tahun 1990. *Game* ini menggunakan tema fantasi, dimana *player* bermain sebagai ksatria

dari Raja Maximus yang bertugas untuk mengambil kembali *Sceptre of Order* dari tangan *Forces of Chaos*.

### 2.3. C#

Microsoft C# (disebut C sharp) adalah sebuah bahasa pemrograman yang didesain untuk membangun jangkauan aplikasi enterprise yang berjalan di atas framework .NET. Sebuah evolusi Microsoft C dan Microsoft C++, C# sederhana, modern, aman dan Object Oriented. C# dikenal sebagai visual C# dalam Visual Studio .Net. Dukungan untuk Visual C# termasuk proyek template, desainer, halaman properti, kode, model objek dan fitur lain dari lingkungan pengembangan. Library untuk pemrograman visual c# adalah .NET Framework [4]

Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti *Java*, *Delphi*, *Visual Basic*, dan lain-lain) dengan beberapa penyederhanaan. Menurut standar ECMA-334 *C# Language Specification*, nama C# terdiri atas sebuah huruf Latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam keyboard standar.

Bahasa pemrograman C# ditujukan untuk digunakan dalam mengembangkan komponen perangkat lunak yang mampu mengambil keuntungan dari lingkungan terdistribusi. C# ditujukan agar cocok digunakan untuk menulis program aplikasi baik dalam sistem klien-server (*hosted system*) maupun sistem embedded (*embedded system*), mulai dari perangkat lunak yang sangat besar yang menggunakan sistem operasi yang canggih hingga kepada perangkat lunak yang sangat kecil yang memiliki fungsi-fungsi terdedikasi. Meskipun aplikasi C# ditujukan agar bersifat 'ekonomis' dalam hal kebutuhan pemrosesan dan memori komputer, bahasa C# tidak ditujukan untuk bersaing secara langsung dengan kinerja dan

ukuran perangkat lunak yang dibuat dengan menggunakan bahasa pemrograman C dan bahasa rakitan.

#### **2.4. *Ancient Empires 2***

*Ancient Empire 2* adalah *Turn-Based Strategy game* yang dikembangkan oleh Glu Mobile, Inc. atau Glu Games pada pertengahan tahun 2005. Permainan ini dibuat pada perangkat bergerak yang berhasil memenangkan Mobie Award untuk game strategi terbaik pada tahun 2005.

#### **2.5. *Diamond-Square Algorithm***

Algoritma *Diamond-Square* adalah pengembangan dari algoritma *Midpoint-Displacement* yang menggabungkan antara kecepatan dari algoritma *Midpoint-Displacement* dan kualitas hasil dari algoritma *Value Noise*. Algoritma ini dimulai dengan mengacak angka dari empat sudut persegi, kemudian mendapatkan nilai titik tengah dari persegi tersebut dengan menghitung rata-rata dari empat angka sudut tersebut kemudian ditambah angka acak. Kemudian mendapatkan empat titik tengah dari keempat sisi persegi. Kemudian diulangi langkah tersebut hingga semua titik terisi [5].

#### **2.6. *Android***

*Android* adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. *Android* awalnya dikembangkan oleh *Android, Inc.*, dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel *Android* pertama mulai dijual pada bulan Oktober 2008 [6].

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

Pada bab ini akan dibahas mengenai analisis dan perancangan yang akan digunakan untuk menyelesaikan tugas akhir.

#### **3.1. Analisis Sistem**

Permainan video berkembang sangat pesat akibat tuntutan perkembangan zaman. Masyarakat berminat terhadap permainan video yang tentunya menyenangkan. Salah satu faktor yang mempengaruhi tingkat kesenangan pemain dari permainan video yaitu *world* atau *battlefield* dari permainan tersebut. *World* permainan yang beragam tentunya memberikan variasi tingkat kesulitan pada permainan. Pembuatan *world* permainan secara manual dengan *modeling* memakan waktu cukup lama. Dibutuhkan suatu sistem yang dapat membangkitkan *world* permainan setiap kali suatu *level* ingin diciptakan.

Aplikasi ini dibangun dengan tujuan membantu para pemain dapat menikmati permainan yang variatif dan otomatis sekaligus dapat menambah pemikiran strategis. *World* pada sistem permainan dibangun secara dinamis. Aplikasi ini diharapkan dapat memberikan pembelajaran yang menyenangkan kepada para pengguna tanpa meninggalkan unsur Fun pada suatu game.

Penulis menggunakan teknologi *Unity (Game Engine)* dengan Bahasa pemrograman *C#* untuk memfasilitasi pengembangan permainan. *World* pada permainan ini dibangun dengan menggunakan *diamond-square algorithm. gameplay* dari permainan ini mengikuti *gameplay* dari game *Ancient Empires 2*.

#### **3.2. Perancangan Permainan**

Tahap perancangan dalam subbab ini dibagi menjadi beberapa bagian yaitu deskripsi umum perangkat lunak, spesifikasi kebutuhan fungsional, spesifikasi kebutuhan nonfungsional, dan karakteristik pemain.

### **3.2.1. Deskripsi Umum Perangkat Lunak**

Tugas akhir yang akan dikembangkan adalah sebuah permainan 2D bergenre *turn-base strategy* dan *tile-based game*. *World* dari permainan ini dibangkitkan menggunakan *diamond-square algorithm*. *World* yang dibangkitkan mempunyai dua macam ukuran, yaitu 9x9 atau 17x17. Permainan ini akan dijalankan pada perangkat *smartphone android* agar dapat dimainkan secara fleksibel.

Pengguna utama dari permainan ini adalah semua orang yang ingin bermain. Pemain akan menjalankan pasukan dari sebuah kerajaan yang bertujuan untuk menguasai kerajaan lawan. Kerajaan lawan akan dijalankan oleh Kecerdasan Buatan (AI).

Untuk memenangkan permainan ini, raja dari pasukan harus menguasai kastil kerajaan lawan. Untuk membantu raja menguasai kastil, pemain dapat membeli berbagai macam pasukan dengan uang yang didapatkan ketika giliran pemain yang jumlah yang didapatkan berdasarkan pada kastil dan rumah yang pemain miliki.

### **3.2.2. Spesifikasi Kebutuhan Fungsional**

Berdasarkan deskripsi umum sistem, maka disimpulkan bahwa kebutuhan fungsional dari aplikasi ini hanya ada satu yaitu bermain *game*.

### **3.2.3. Spesifikasi Kebutuhan Non-Fungsional**

Terdapat beberapa kebutuhan non-fungsional yang apabila dipenuhi, dapat meningkatkan kualitas dari permainan ini. Berikut daftar kebutuhan non-fungsional:

#### **1. *FrameRate***

Permainan ini harus mampu dimainkan secara lancar, tidak ada *lag* dan nyaman di mata. Sebagian permainan 2D biasanya optimal pada *Frame Rate* 24-30 fps (*Frame per Second*). Namun untuk pembangkitan level secara dinamis dapat mempengaruhi kelancaran *Frame Rate* di awal pembentukan. Serta kelancaran *Frame Rate* dipengaruhi dan tergantung oleh spesifikasi *smartphone* yang akan digunakan.

## 2. Kebutuhan Grafis

Kenyamanan bermain berbanding lurus dengan kualitas grafis yang disajikan dalam permainan. Efek seperti animasi merupakan salah satu daya tarik dalam suatu permainan. Efek-efek ini bisa membuat *drop rate fps* dan permainan melambat (*lag*), karena membutuhkan tambahan komputasi.

### 3.2.4. Karakteristik Pemain

Berdasarkan deskripsi umum diatas, maka dapat diketahui bahwa pengguna yang akan menggunakan aplikasi ini ada dua orang, yaitu pemain yang memainkan permainan, dan pengembang. Karakteristik pengguna tercantum dalam Tabel 3.1.

**Tabel 3.1 Karakteristik Pengguna**

<b>Nama Aktor</b>	<b>Tugas</b>	<b>Hak Akses Aplikasi</b>	<b>Kemampuan yang harus dimiliki</b>
Pemain	Pihak luar yang memainkan permainan.	Memainkan permainan	Tidak ada

## 3.3. Perancangan Sistem

Tahap perancangan dalam subbab ini dibagi menjadi beberapa bagian yaitu perancangan pembangkit world, perancangan diagram kasus penggunaan, perancangan skenario kasus penggunaan, perancangan antarmuka pengguna, perancangan kontrol permainan, dan perancangan alur permainan.

### 3.3.1. Perancangan Pembangkit World

Dalam aplikasi tugas akhir ini, digunakan algoritma diamond square untuk membangkitkan world secara dinamis. Diamond-square adalah salah satu algoritma yang digunakan untuk membangkitkan map. Algoritma ini membutuhkan map persegi yang berukuran  $2^n+1$ .

```

// Algoritma Diamond-Square//
Fungsi DiamondSquare()
{
    n ← input          //ukuran map 2n+1
    min ← input        //nilai minimum pada map
    max ← input        //nilai maksimum pada map
    r ← input          //range angka random
    u ← 2n+1          //ukuran map
    a ← (1,1)          //a → vector pojok kiri bawah
    b ← (u,1)          //b → vector pojok kanan bawah
    c ← (1,u)          //c → vector pojok kiri atas
    d ← (u,u)          //d → vector pojok kanan atas
    map[a] ← random(-r,r)
    map[b] ← random(-r,r)
    map[c] ← random(-r,r)
    map[d] ← random(-r,r)

    FillDiamondSquare(a,b,c,d)
}

Fungsi FillDiamondSquare(a,b,c,d)
{
    mid ← (a.x+(b.x-a.x)/2, a.y+(c.y-a.y)/2)
    atas ← (mid.x, c.y)
    kiri ← (a.x, mid.y)
    kanan ← (b.x, mid.y)
    bawah ← (mid.x, a.y)

    map[mid] ← Rata(a,b,c,d)
    if(c.y==u) //tepi map
        map[atas] ← Rata(c,mid,d)
    else
        map[atas] ← Rata(c,mid,d,(mid.x, 2*c.y-mid.y))
    if(a.x==1) //tepi map
        map[kiri] ← Rata(a,mid,c)
    else
        map[kiri] ← Rata(a,mid,c,(2*a.x-mid.x, mid.y))
    if(b.x==u) //tepi map
        map[kanan] ← Rata(b,mid,d)
    else
        map[kanan] ← Rata(b,mid,d,(2*b.x-mid.x, mid.y))
}

```

```

if(a.y==0) //tepi map
    map[bawah] ← (a,mid,b)
else
    map[bawah] ← (a,mid,b,(mid.x, 2*a.y-mid.y))

if(b.x-a.x>2)
    FillDiamondSquare(a,bawah,kiri,mid)
    FillDiamondSquare(bawah,b,mid,kanan)
    FillDiamondSquare(kiri,mid,c,atas)
    FillDiamondSquare(mid,kanan,atas,d)
}

int Rata(v1,v2,v3,...)
{
    return (map[v1]+map[v2]+map[v3]+...)/v+random(-r,r)
}

```

### Kode Sumber 3.1 Algoritma *Diamond-Square*

Algoritma *Diamond-Square* seperti pada Kode Sumber 3.1 membutuhkan *input* berupa ukuran, nilai minimum dan maksimum, serta *range* nilai acak. Kemudian mempunyai *output* berupa variabel *integer* dua dimensi yang sudah terisi.

Pada permainan ini akan digunakan map yang berukuran 9x9 dan 17x17. Berikut penjelasan dari algoritma diamond-square. Di aplikasi ini digunakan *range* angka dari 0 sampai 4 yang digunakan untuk merepresentasikan *tile* yang akan membentuk *world* pada aplikasi.

3							1
1							1

**Gambar 3.1 Penjelasan Diamond-Square (bagian 1)**

Langkah pertama dari algoritma diamond-square adalah mengisi nilai pojok dengan angka acak seperti yang terlihat pada Gambar 3.1.

3							1
			0				
1							1

**Gambar 3.2 Penjelasan Diamond-Square (bagian 2)**

Kemudian langkah kedua, menghitung titik tengah yang didapatkan dari nilai rata-rata dari keempat titik pojok yang kemudian ditambah dengan nilai random antara -1 sampai 1. Seperti yang

terlihat pada Gambar 3.2, yaitu nilai pada warna biru didapatkan dari hasil rata-rata dari nilai pada warna hijau ditambah nilai random.

3			0			1
2			0			1
3			0			1
2			0			1
1			1			1

**Gambar 3.3 Penjelasan Diamond-Square (bagian 3)**

Selanjutnya langkah ketiga, menghitung keempat titik tengah dari tepi persegi yang didapatkan dari rata-rata nilai atas, kanan, kiri, dan bawah kemudian ditambahkan dengan nilai random antara -1

sampai 1. Seperti yang terlihat pada Gambar 3.3, dimana titik tengah dari tepi atas yang berwarna biru diperoleh dari rata-rata nilai atas, kanan, kiri, dan bawah yang berwarna hijau. Bagian tepi atas-bawah dan kiri-kanan sebenarnya saling terhubung, sehingga nilai atas dari warna biru berada dibagian bawah. Pada Gambar 3.2 keempat warna hijau membentuk pola persegi (*square*), sedangkan pada Gambar 3.3 keempat warna hijau membentuk pola *diamond*, inilah yang menyebabkan algoritma ini dinamai dengan algoritma *Diamond-Square*.

3				0				1
		0				0		
2				0				1
		2				0		
1				1				1

**Gambar 3.4 Penjelasan Diamond-Square (bagian 4)**

Kemudian langkah kelima, memisah world tersebut menjadi empat bagian, lalu mengulangi langkah kedua sampai langkah kelima hingga nilai pada world tersebut terisi seluruhnya. Seperti yang terlihat pada Gambar 3.4, nilai titik tengah warna hijau diperoleh dari hasil rata-rata keempat nilai pojok warna biru.

3			0				1
	0	1	0				
2			0				1
	2			0			
1			1				1

**Gambar 3.5 Penjelasan Diamond-Square (bagian 5)**

Pada Gambar 3.5 merupakan langkah keempat dari algoritma *diamond-square*, yaitu menghitung titik tengah (warna hijau) dari keempat tepi persegi yang diperoleh dari keempat titik atas, kanan, kiri, dan bawah (warna biru).

3	3	1	2	0	2	1	2	1
3	1	2	2	1	1	2	1	2
2	2	0	0	1	1	0	0	1
2	2	0	0	1	2	0	0	2
2	1	1	1	0	1	1	1	1
1	3	2	1	0	0	1	0	0
2	2	2	1	0	0	0	0	0
1	1	1	1	0	0	0	0	1
1	1	1	1	1	0	0	0	1

**Gambar 3.6 Penjelasan Diamond-Square (bagian 6)**

Pada Gambar 3.6 merupakan pengisian terakhir dari algoritma *Diamond-Square*. Dimana titik tengah dari tepi bawah (warna hijau) diperoleh dari keempat nilai atas, kanan, kiri, dan bawah (warna biru).

3	3	1	2	0	2	1	2	1
3	1	2	2	1	1	2	1	2
2	2	0	0	1	1	0	0	1
2	2	0	0	1	2	0	0	2
2	1	1	1	0	1	1	1	1
1	3	2	1	0	0	1	0	0
2	2	2	1	0	0	0	0	0
1	1	1	1	0	0	0	0	1
1	1	1	1	1	0	0	0	1

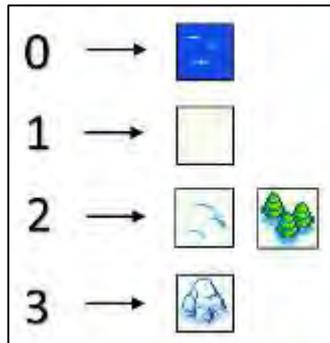
**Gambar 3.7 Perbaikan Hasil (bagian 1)**

Selanjutnya untuk membangkitkan *tiles* yang direpresentasikan oleh nilai hasil dari perhitungan algoritma *Diamond-Square*, dilakukan pengecekan dari angka 0 yang mempunyai sisi angka 0 kurang dari 2, selanjutnya angka 0 tersebut akan dirubah menjadi angka 1 sampai 4 secara acak. Seperti yang terlihat pada gambar Gambar 3.7.

Pada Gambar 3.8 merupakan hasil dari perbaikan dari nilai pada Gambar 3.7, dimana hasil nilai perbaikan adalah acak dari angka 1 sampai 4.

3	3	1	2	2	2	1	2	1
3	1	2	2	1	1	2	1	2
2	2	0	0	1	1	0	0	1
2	2	0	0	1	2	0	0	2
2	1	1	1	2	1	1	1	1
1	3	2	1	0	0	1	0	0
2	2	2	1	0	0	0	0	0
1	1	1	1	0	0	0	0	1
1	1	1	1	1	0	0	0	1

**Gambar 3.8 Perbaikan Hasil (bagian 2)**



**Gambar 3.9 Keterangan Nilai Pada Tile**

Pada Gambar 3.9 merupakan keterangan *tile* pada angka hasil pembangkitan *world* menggunakan algoritma *Diamond-Square*.

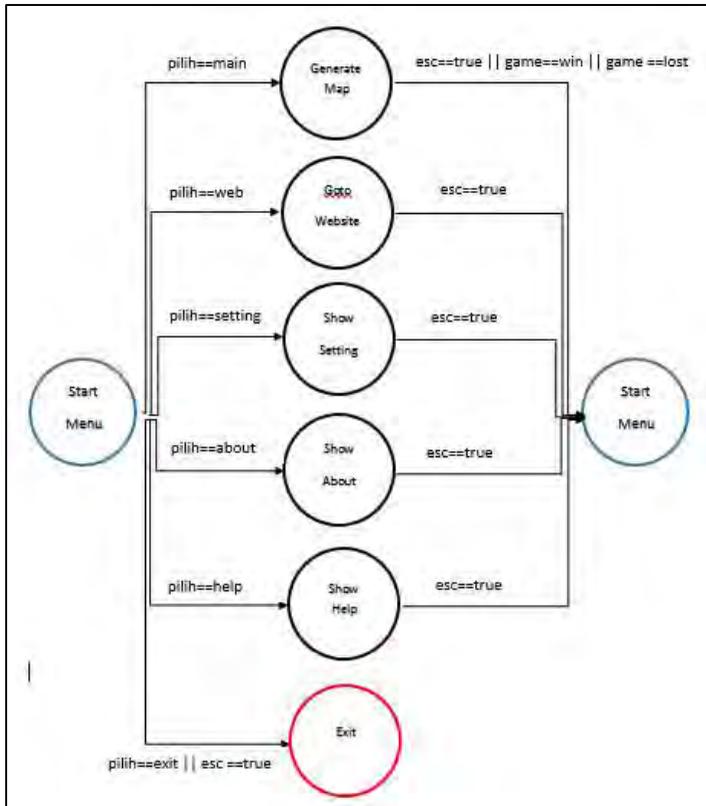
3	3	2	2	2	1	2	1	
3	1	2	2	1	1	2	2	
2	2	0	0	1	2	0	0	1
2	2	0	0	1	2	0	0	2
2	2	1	1	2	1	1	1	1
1	3	2	2	0	0	1	0	0
2	2	2	1	0	0	0	0	0
1	1	1	1	0	0	0	0	1
1	1	1	1	1	0	0	0	1

**Gambar 3.10 Implementasi Tiles**

Pada Gambar 3.10 merupakan contoh hasil world implementasi tiles yang direpresentasikan oleh nilai yang diperoleh dari perhitungan menggunakan algoritma *Diamond-Square*. Dimana tiap nilai dalam kotak merepresentasikan *tile* tersendiri.

### 3.3.2. Perancangan Finite State Machine

Dalam aplikasi tugas akhir ini, terdapat FSM pada saat sedang berada dalam main menu, untuk memudahkan memahami state pada main menu. Seperti pada Gambar 3.11 dibawah.



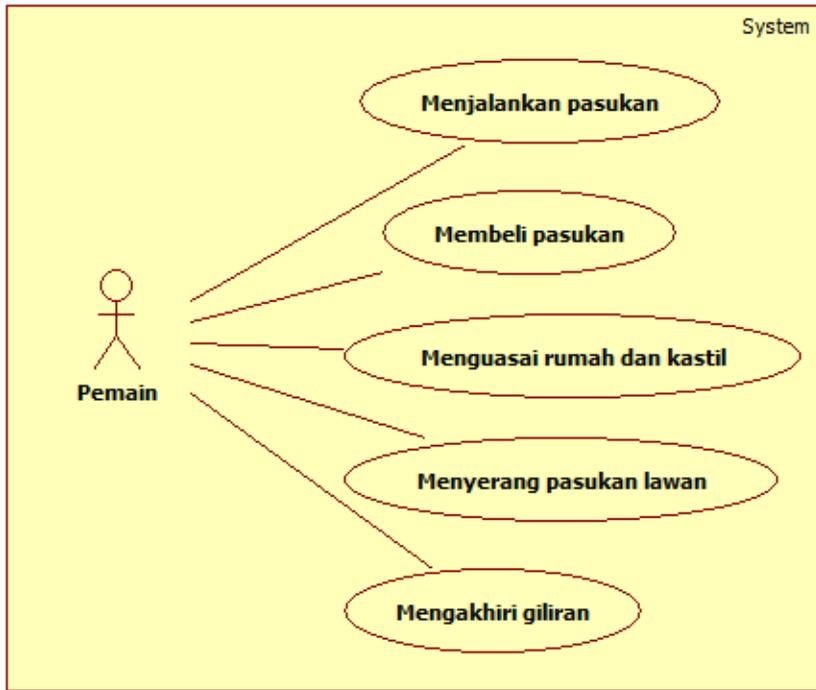
**Gambar 3.11 Finite State Machine Main Menu**

### 3.3.3. Perancangan Diagram Kasus Penggunaan

Dalam aplikasi tugas akhir ini, pada saat sedang bermain terdapat lima kasus penggunaan. Lima kasus penggunaan ini diantaranya adalah menjalankan pasukan, membeli pasukan, menguasai rumah dan kastil, menyerang pasukan lawan, dan mengakhiri giliran.

### 3.3.4. Perancangan Skenario Kasus Penggunaan

Kasus penggunaan yang terdapat didalam sistem dicantumkan pada Gambar 3.12.



**Gambar 3.12 Diagram Kasus Aplikasi**

Penjelasan dari masing-masing kasus penggunaan dicantumkan pada Tabel 3.2. Tabel tersebut berisi penjelasan skenario yang akan dilakukan ketika pengujian.

**Tabel 3.2 Skenario Kasus Penggunaan**

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1	UC-001	Menjalankan Pasukan	Untuk menguji coba pemain dalam menjalankan pasukan.
2	UC-002	Membeli Pasukan	Untuk menguji coba pemain dalam membeli pasukan.

3	UC-003	Menguasai Rumah dan Kastil	Untuk menguji coba pemain dalam menguasai rumah dan kastil.
4	UC-004	Menyerang Pasukan Lawan	Untuk menguji coba pemain dalam menyerang pasukan lawan.
5	UC-005	Mengakhiri giliran	Untuk menguji coba pemain dalam mengakhiri giliran.

### 3.3.4.1. Kasus Penggunaan Permainan

Penjelasan kasus penggunaan permainan untuk skenario UC-001 yakni Menjalankan Pasukan dijelaskan pada Tabel 3.3.

**Tabel 3.3 Skenario Kasus Penggunaan Menjalankan Pasukan**

<b>Nama Kasus Penggunaan</b>	Menjalankan Pasukan
<b>Kode</b>	UC-001
<b>Deskripsi</b>	Kasus penggunaan dimana aktor menjalankan pasukan yang dipilih.
<b>Aktor</b>	Pemain.
<b>Kondisi Awal</b>	Pemain sudah masuk ke aplikasi dan memilih menu main.
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pemain memilih pasukan dengan menyentuh pasukan tersebut.</li> <li>2. Sistem menampilkan menu <b>aksi</b> dari pasukan tersebut.</li> <li>3. Pemain memilih menu <b>jalan</b>.</li> <li>4. Sistem menampilkan tile yang dapat didatangi oleh pasukan tersebut.</li> <li>5. Pemain memilih salah satu dari tile tersebut.</li> <li>6. Sistem menampilkan menu aksi dari pasukan tersebut.</li> <li>7. Pemain memilih menu selesai.</li> </ol>

Selanjutnya penjelasan kasus penggunaan permainan untuk skenario UC-002 yakni Membeli Pasukan dijelaskan pada Tabel 3.4.

**Tabel 3.4 Skenario Kasus Penggunaan Membeli Pasukan**

<b>Nama Kasus Penggunaan</b>	Membeli Pasukan
<b>Kode</b>	UC-002
<b>Deskripsi</b>	Kasus penggunaan dimana aktor membeli pasukan.
<b>Aktor</b>	Pemain.
<b>Kondisi Awal</b>	Pemain sudah masuk ke aplikasi dan memilih menu main
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pemain memilih kastil dengan menyentuh kastil tersebut.</li> <li>2. Sistem menampilkan menu aksi dari kastil tersebut.</li> <li>3. Pemain memilih menu beli.</li> <li>4. Sistem menampilkan toko pasukan.</li> <li>5. Pemain memilih pasukan di toko tersebut dan menekan tombol centang.</li> <li>6. Sistem membuat pasukan yang telah dibeli.</li> </ol>

Kemudian penjelasan kasus penggunaan permainan untuk skenario UC-002 yakni Menyerang Pasukan Lawan dijelaskan pada Tabel 3.5.

**Tabel 3.5 Skenario Kasus Penggunaan Menyerang Pasukan Lawan**

<b>Nama Kasus Penggunaan</b>	Menyerang Pasukan Lawan
<b>Kode</b>	UC-003
<b>Deskripsi</b>	Kasus penggunaan dimana aktor menyerang pasukan lawan.
<b>Aktor</b>	Pemain.
<b>Kondisi Awal</b>	Pemain sudah masuk ke aplikasi dan memilih menu main.
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pemain memilih pasukan yang berada di jarak serang pasukan lawan dengan menyentuh pasukan tersebut.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Sistem menampilkan menu aksi dari pasukan tersebut.</li> <li>3. Pemain memilih menu serang.</li> <li>4. Sistem menampilkan tile yang dapat diserang oleh pasukan tersebut.</li> <li>5. Pemain memilih salah satu pasukan lawan yang berada di tile tersebut.</li> </ol>
--	---

Selanjutnya penjelasan kasus penggunaan permainan untuk skenario UC-004 yakni Menguasai Kastil dan Rumah dijelaskan pada Tabel 3.6.

**Tabel 3.6 Skenario Kasus Penggunaan Menguasai Kastil dan Rumah**

<b>Nama Kasus Penggunaan</b>	Menguasai Kastil dan Rumah
<b>Kode</b>	UC-004
<b>Deskripsi</b>	Kasus penggunaan dimana aktor menguasai kastil dan rumah.
<b>Aktor</b>	Pemain.
<b>Kondisi Awal</b>	Pemain sudah masuk ke aplikasi dan memilih menu main.
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pemain memilih pasukan soldier atau raja yang berada di kastil atau rumah yang belum dikuasai dengan menyentuh pasukan tersebut.</li> <li>2. Sistem menampilkan menu aksi dari pasukan tersebut.</li> <li>3. Pemain memilih menu kuasai.</li> </ol>

Kemudian penjelasan kasus penggunaan permainan untuk skenario UC-005 yakni Mengakhiri Giliran dijelaskan pada Tabel 3.7.

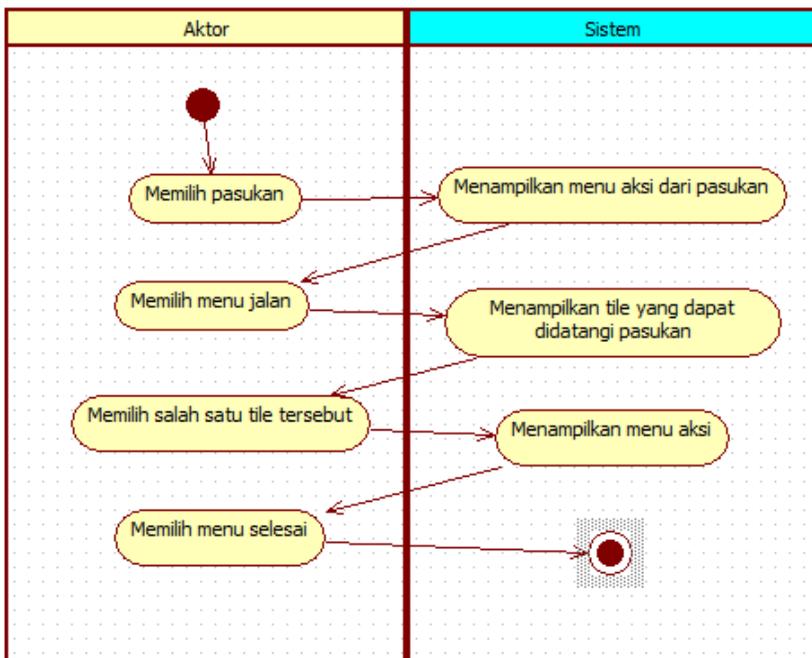
**Tabel 3.7 Skenario Kasus Penggunaan Mengakhiri Giliran**

<b>Nama Kasus Penggunaan</b>	Mengakhiri Giliran
<b>Kode</b>	UC-005

<b>Deskripsi</b>	Kasus penggunaan dimana aktor mengakhiri giliran.
<b>Aktor</b>	Pemain.
<b>Kondisi Awal</b>	Pemain sudah masuk ke aplikasi dan memilih menu main.
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pemain menyentuh tombol giliran selesai yang berada di pojok kanan bawah.</li> <li>2. Sistem merubah giliran musuh.</li> </ol>

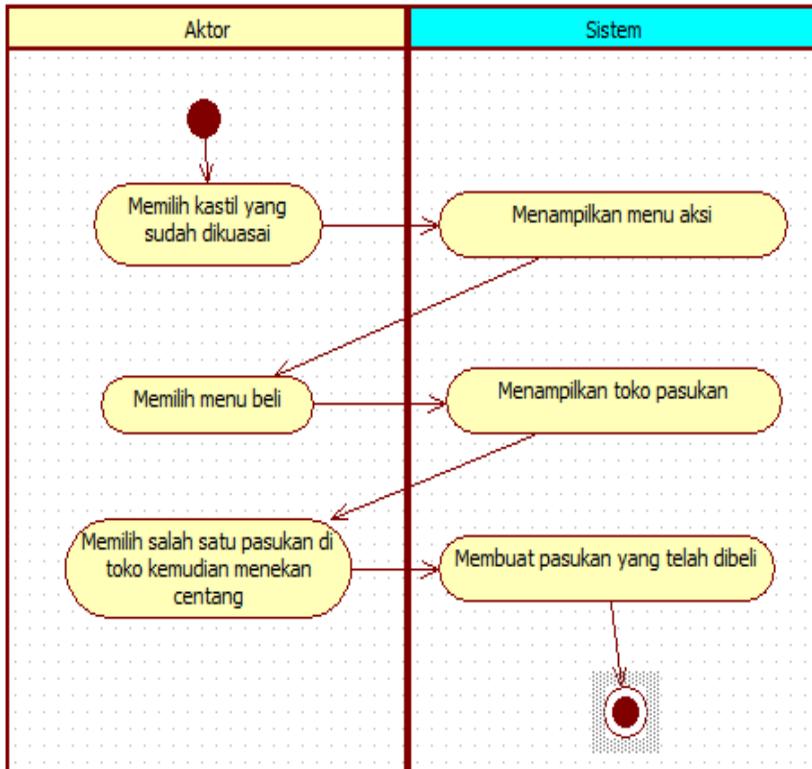
### 3.3.4.2. Diagram Aktifitas

Diagram aktivitas menampilkan langkah-langkah normal yang harus dilakukan pemain untuk menjalankan studi kasus permainan dimulai dari awal permainan hingga kondisi akhir. Berikut Gambar 3.13 diagram aktivitas untuk dari kasus penggunaan UC-001.



**Gambar 3.13 Diagram Aktifitas Menjalankan Pasukan**

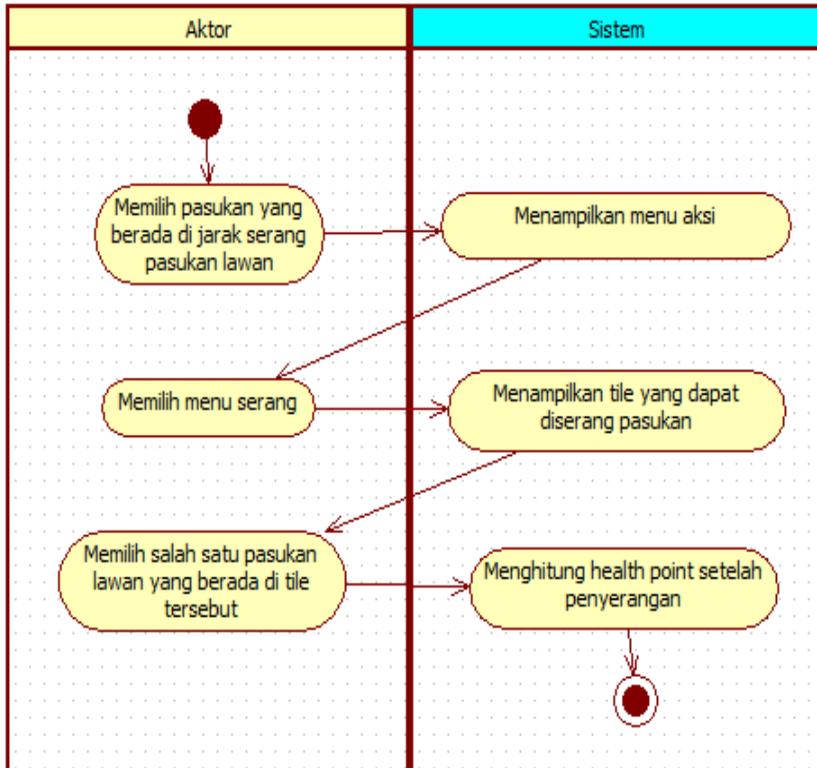
Kemudian pada Gambar 3.14 merupakan diagram aktivitas untuk dari kasus penggunaan UC-002 yakni Membeli Pasukan. Diagram aktivitas menyatakan bahwa aktivitas dimulai dari memilih kastil yang sudah dikuasai yang dilakukan oleh pengguna dan diakhiri dengan membuat pasukan yang telah dibeli yang dilakukan oleh sistem.



**Gambar 3.14 Diagram Aktifitas Membeli Pasukan**

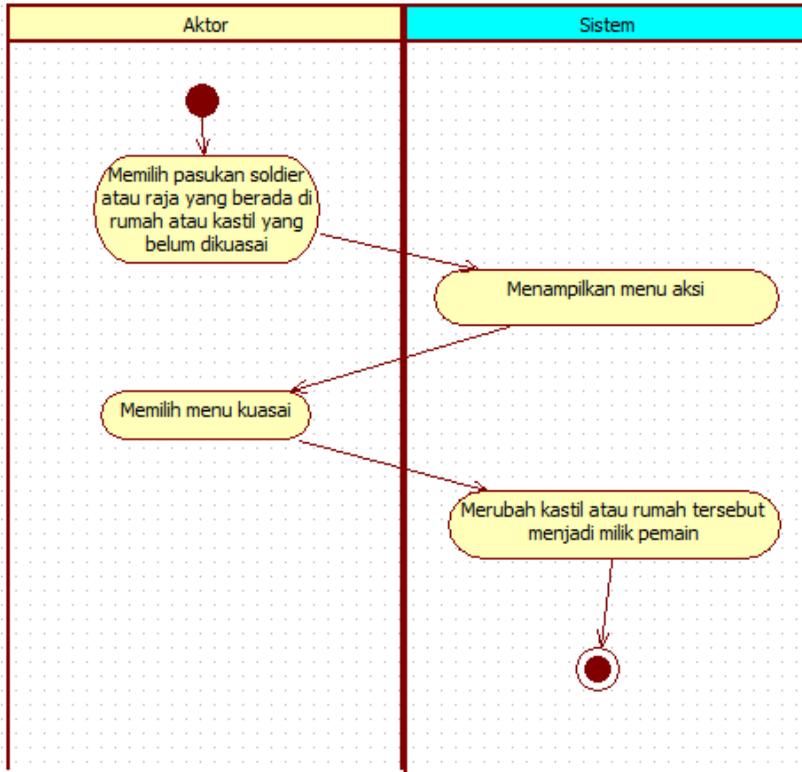
Kemudian pada Gambar 3.15 merupakan diagram aktivitas untuk dari kasus penggunaan UC-003 yakni Menyerang Pasukan Lawan. Diagram aktivitas menyatakan bahwa aktivitas dimulai dari

pasukan yang berada di jarak serang pasukan lawan yang dilakukan oleh pengguna dan diakhiri dengan perhitungan *health point* yang dilakukan oleh sistem.



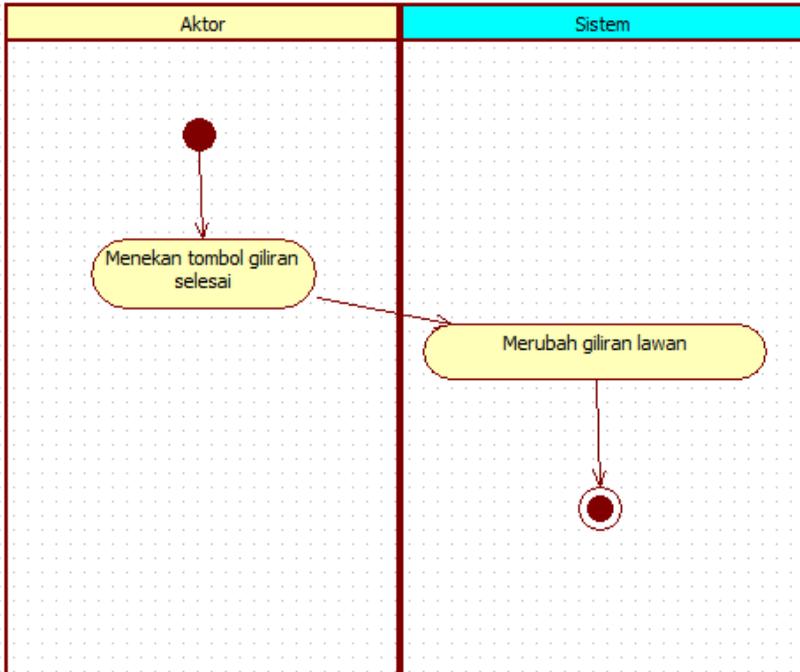
**Gambar 3.15 Diagram Aktifitas Menyerang Pasukan Lawan**

Kemudian Gambar 3.16 diagram aktivitas untuk dari kasus penggunaan UC-004 yakni Menguasai Rumah dan Kastil. Diagram aktivitas menyatakan bahwa aktivitas dimulai dari pasukan soldier atau raja yang berada di rumah atau kastil yang belum dikuasai yang dilakukan oleh pengguna dan diakhiri dengan merubah rumah atau kastil menjadi milik pemain yang dilakukan oleh sistem.



**Gambar 3.16 Diagram Aktifitas Menguasai Rumah dan Kastil**

Kemudian pada Gambar 3.17 merupakan diagram aktivitas untuk dari kasus penggunaan UC-005 yakni Mengakhiri Giliran. Diagram aktivitas menyatakan bahwa aktivitas dimulai dari menekan tombol giliran selesai yang berada di bagian kanan bawah yang dilakukan oleh pengguna dan diakhiri dengan merubah giliran lawan yang dilakukan oleh sistem.



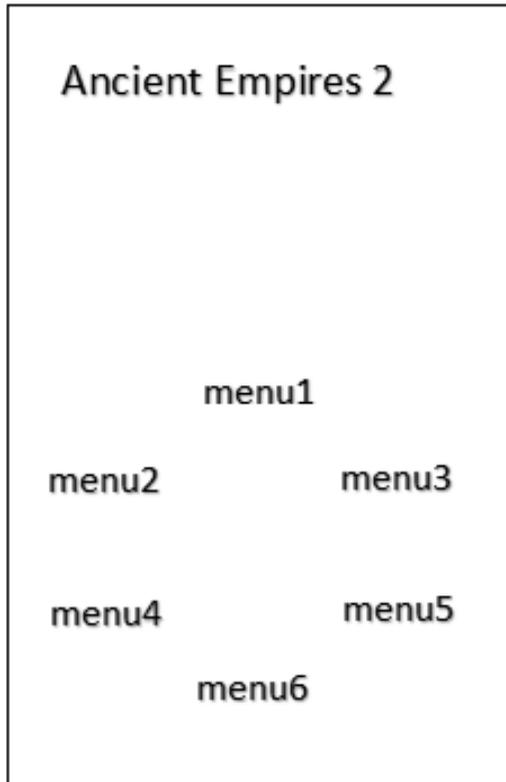
**Gambar 3.17 Diagram Aktifitas Mengakhiri Giliran**

Pada Gambar 3.17 Diagram Aktifitas Mengakhiri Giliran, berawal dari pemain menekan tombol giliran selesai yang berada di pojok kanan bawah, kemudian sistem akan merubah giliran dari pemain.

### **3.3.5. Perancangan Antarmuka Pengguna**

Subbab ini membahas bagaimana rancangan antarmuka pengguna yang akan digunakan untuk tugas akhir. Rancangan antarmuka yang dibahas meliputi ketentuan masukan dan rancangan jendela tampilan.

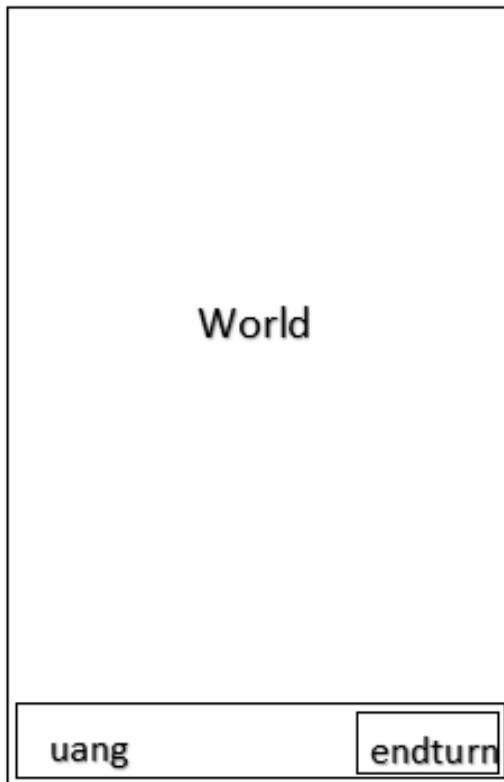
### 3.3.5.1. Tampilan Main Menu



**Gambar 3.18 Tampilan Main Menu**

Tampilan main menu merupakan tampilan yang pertama kali muncul ketika aplikasi dijalankan dan pemain menyentuh layar, seperti pada Gambar 3.18 di atas.

### 3.3.5.2. Tampilan Saat Bermain



**Gambar 3.19 Tampilan Saat Bermain**

Tampilan saat bermain, Gambar 3.19, merupakan tampilan yang muncul ketika pemain memilih menu main pada main menu. World pada tampilan saat bermain ini selalau berubah setiap permainan baru.

### 3.3.5.3. Tampilan Toko Pasukan

namaunit	harga		
atk	def		
move			
Keterangan unit			
unit1	unit2	uni3	unit4
uang	endturn		

**Gambar 3.20 Tampilan Toko Pasukan**

Tampilan toko pasukan, Gambar 3.20, merupakan tampilan yang muncul ketika pemain memilih menu beli pada menu aksi. Di dalam toko tersebut terdapat 9 macam pasukan yang mempunyai karakteristik dan harga yang berbeda-beda.

### **3.3.6. Perancangan Kontrol Permainan**

Untuk kontrol permainan ini menggunakan tap dan swipe. Semua interaksi dari user hanya menggunakan kontrol tersebut. Tap digunakan untuk menekan tombol dan memilih pasukan. Sedangkan swipe digunakan untuk menggeser *world* sesuai yang kita inginkan.

### **3.3.7. Perancangan Aturan Permainan**

Aturan permainan pada permainan ini mengikuti aturan main dari game Ancient Empires 2. Tap unit untuk menggerakkan pasukan tersebut. Kotak merah merupakan daya jelajah dari unit tersebut. Daya jelajah dari unit tersebut dipengaruhi oleh kemampuan jelajah unit tersebut dan medan yang dilewati. Berbagai macam medan mempengaruhi unit. Misalnya gunung memberi pertahanan ekstra, tapi mengurangi daya jelajah unit. Setelah unit dimainkan maka akan berubah menjadi abu-abu, yang menandakan bahwa unit tersebut sudah dijalankan pada giliran sekarang, dan tidak bias dipakai lagi sampai giliran yang berikutnya.

Setelah selesai menggerakkan semua unit, tekan tombol yang berada di pojok kanan bawah untuk mengakhiri giliran. Untuk menyerang pasukan musuh yang berada di jarak serang, gerakkan unit dan pilih serang. Hanya raja yang dapat menguasai kastil. Kastil yang sudah dikuasai bias dipakai untuk membeli unit baru. Untuk dapat memenangkan permainan ini, unit raja lawan harus dihancurkan dan juga unit raja pemain harus menguasai kastil dari kerajaan lawan. Gunakan unit soldier dan raja untuk menguasai rumah. Rumah yang dikuasai akan berubah warna. Rumah yang sudah dikuasai akan menambah uang pemasukan. Unit bias menambah health point jika berada di rumah.

Unit elemental adalah amfibi. Serangan, pertahanan, dan pergerakan unit ini bertambah jika berada di air. Sedangkan unit wolf mempunyai serangan beracun yang dapat mengurangi pertahanan unit yang terkena serangannya untuk satu giliran. Sedangkan unit penyihir punya kekuatan untuk membangkitkan unit tengkorak dari batu nisan yang ada. Sedangkan unit wisp dapat menambah serangan dari unit teman yang berada di sekitarnya.

### **3.3.8. Perancangan Skenario Permainan**

Scenario dari permainan ini hamper menyerupai skenario dari game ancient empires 2. Pemain menjalankan unit biru yang dipimpin oleh raja Galamar yang tujuan utamanya adalah mengalahkan unit merah yang dipimpin oleh Demon Lord. Pada saat awal permainan pemain dan lawan masing-masing diberikan sebuah kastil yang berada di posisi acak. Dan juga terdapat beberapa rumah yang belum dikuasai yang ditempatkan secara acak dalam map yang berukuran acak antara 9x9 dan 17x17.

*[Halaman ini sengaja dikosongkan]*

## BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Di dalamnya mencakup proses penerapan dan pengimplementasian algoritma, dan antar muka yang mengacu pada rancangan yang telah dibahas sebelumnya.

### 4.1. Lingkungan Implementasi

Lingkungan implementasi dari tugas akhir dijelaskan pada Tabel 4.1.

**Tabel 4.1 Lingkungan Implementasi Perangkat Lunak**

Perangkat Keras	Prosesor: Pentium(R) Core(TM) i3-32170 CPU @ 1.80GHz Memori: 4 GB
Perangkat Lunak	Sistem Operasi: Microsoft Windows 8.1 64-bit Perangkat Pengembang: Unity 4.3

### 4.2. Implementasi Permainan

Implementasi dari masing-masing fungsi utama dituliskan menggunakan *pseudocode* berdasarkan antarmuka utama yang ada pada permainan. Penjelasan implementasi hanya berupa antar muka yang berhubungan dengan pembangkit *world* menggunakan algoritma *diamond-square*, menjalankan pasukan, dan menyerang lawan.

#### 4.2.1. Implementasi Pembangkit World

Implementasi pembangkit world terdiri dari beberapa fungsi yang dituliskan pada Kode Sumber 4.1 dan Kode Sumber 4.2.

```
// fungsi DiamondSquare //  
generatedMap[1,1]<-random(0,4)  
generatedMap[9,1]<-random(0,4)  
generatedMap[1,9]<-random(0,4)  
generatedMap[9,9]<-random(0,4)
```

```

for(i=0 to 3)
  for(j=0 to 2^i)
    for(k=0 to 2^i)
      a=(k*y+1, j*y+1) //pojok kiri bawah
      b=(k*y+y+1, j*y+1) //pojok kanan bawah
      c=(k*y+1, j*y+y+1) //pojok kiri atas
      d=(k*y+y+1, j*y+y+1) //pojok kanan atas
      FillDiamondSquare(a,b,c,d)

```

#### Kode Sumber 4.1 Fungsi DiamondSquare

Pada fungsi `DiamondSquare()` langkah pertama yang dilakukan adalah mengisi nilai dari keempat titik pojok dari map yang akan dibuat, yang direpresentasikan oleh variable `generatedMap[,]`. Kemudian terdapat iterasi untuk membagi persegi tersebut menjadi 4 bagian dan seterusnya hingga semua nilai pada variable `generatedMap[,]` terisi. Didalam iterasi tersebut memanggil fungsi `FillDiamondSquare()` untuk mengisi nilai-nilai pada variable `generatedMap[,]`.

```

// fungsi FillDiamondSquare //
Mid<-Average(generatedMap[a], generatedMap[b],
generatedMap[c], generatedMap[d]) + random(-1,1)

Bawah<-Average(Mid, generatedMap[a], generatedMap[b],
generatedMap[Bawah.x,Bawah.y-(Mid.y-Bawah.y)]) +
random(-1,1)

Kiri<-Average(Mid, generatedmap[a], generatedMap[c],
generatedmap[Kiri.x-(Mid.x-Kiri.x),Kiri.y]) +
random(-1,1)

Kanan<-Average(Mid, generatedMap[b], generatedMap[d],
generatedMap[Kanan.x+(Kanan.x-Mid.x),Kanan.y]) +
random(-1,1)

Atas<-Average(Mid, generatedMap[c], generatedMap[d],
generatedMap[Atas.x,Atas.y+(Atas.y-Mid.y)]) +
random(-1,1)

```

#### Kode Sumber 4.2 Fungsi FillDiamondSquare

Pada fungsi `FillDiamondSquare()` langkah pertama yang dilakukan adalah mengisi nilai titik tengah yang didapatkan dari nilai rata-rata dari keempat nilai di keempat sudut. Kemudian mengisi nilai titik tengah dari masing-masing sisi tepi persegi dari nilai rata-rata dari keempat titik kiri, kanan, atas, dan bawah. Yang jika keempat titik tersebut dihubungkan akan membentuk pola *diamond*.

3	3	1	2	2	2	1	2	1
3	1	2	2	1	1	2	1	2
2	2	0	0	1	1	0	0	1
2	2	0	0	1	2	0	0	2
2	1	1	1	2	1	1	1	1
1	3	2	1	0	0	1	0	0
2	2	2	1	0	0	0	0	0
1	1	1	1	0	0	0	0	1
1	1	1	1	1	0	0	0	1

**Gambar 4.1** Isi Variabel Setelah Fungsi `DiamondSquare` Dijalankan

Pada Gambar 4.1 merupakan variabel `generatedMap[,]` yang sudah terisi setelah pemanggilan fungsi `DiamondSquare`. Selanjutnya akan dilakukan pemanggilan fungsi `GenerateMap`, yang berfungsi untuk merubah nilai-nilai pada variabel `generatedMap[,]` menjadi *tile* sesuai dengan nilai yang dimiliki.

```
// fungsi GenerateMap //  
for(i=1 to 9)  
  for(j=1 to 9)  
    position<- (i,j)  
    gameObject<- tileType(generatedMap[i,j])  
    Instantiate(gameObject, position)
```

### Kode Sumber 4.3 Fungsi GeneratedMap

Fungsi GenerateMap pada Kode Sumber 4.3 berfungsi untuk membangkitkan *tile* berdasarkan masing-masing nilai pada variabel generatedMap[.,].



Gambar 4.2 Tampilan Hasil Pembangkitan Map 9x9

Gambar 4.2 merupakan hasil pembangkitan map yang diperoleh dari fungsi yang terdapat dalam Kode Sumber 4.1, Kode Sumber 4.2,

dan Kode Sumber 4.3 Map pada Gambar 4.2 merupakan hasil dari variabel `generatedMap[,]` yang terdiri dari beberapa *tile*. Misalnya *tile* hutan merupakan representasi dari nilai 2 dan *tile* air merupakan representasi dari nilai 0 yang terdapat dalam variabel `generatedMap[,]` tersebut. Sedangkan pada Gambar 4.3 merupakan hasil pembangkitan map dengan ukuran 17x17.



**Gambar 4.3 Tampilan Hasil Pembangkitan Map 17x17**



**Gambar 4.4 Tampilan Hasil Pembangkitan Map 9x9 Menggunakan Nilai Random**

Pada *map* ukuran 9x9 untuk pembangkitan *map* menggunakan nilai acak masih terlihat baik seperti pada Gambar 4.4. sedangkan untuk hasil *map* ukuran 17x17 menggunakan nilai acak seperti pada Gambar 4.5 terlihat kurang bervariasi jika dibandingkan dengan hasil pembangkitan map 17x17 menggunakan algoritma *Diamond-Square* seperti pada Gambar 4.3.



**Gambar 4.5** Tampilan Hasil Pembangkitan Map 17x17 Menggunakan Nilai Random

#### 4.2.2. Implementasi Menjalankan Pasukan

Implementasi Menjalankan Pasukan diawali dengan pemanggilan fungsi `Cek()` yang berada di *class* `CekAlpha`.

```
// pemanggilan fungsi Cek //
_cekalpha <- gameObject.AddComponent<CekAlpha>
_cekalpha.Cek(transform.position)
```

**Kode Sumber 4.4** Pemanggilan Fungsi Cek

Fungsi Cek() bertujuan untuk menghitung jarak pergerakan dari suatu unit, yang nantinya akan direpresentasikan oleh kotak-kotak berwarna merah.

```
// fungsi_Cek //
maxmoves<-5
if(unit==wolf)
  maxmoves<-6
if(unit==dragon)
  maxmoves<-7

for(i=1 to 9)
  for(j=1 to 9)
    if(tileMap[i,j]==mountain || water)
      groundMap<-3
    if(tileMap[i,j]==snow || jungle)
      groundMap<-2
    if(tileMap[i,j]==land)
      groundMap<-1
    if(unit==dragon)
      groundMap<-1
    if(unit==elemental && tileMap[i,j]==water)
      groundMap<-1

for(i=1 to 9)
  for(j=1 to 9)
    p<- unit.position
    jarak<- ShortestPath(groundMap[p],groundMap[i,j])
    if(jarak<=maxmoves)
      Instantiate(redTile, (i,j))
```

#### Kode Sumber 4.5 Fungsi Cek

Pada Kode Sumber 4.5 di atas, variabel maxmoves digunakan untuk mengetahui pergerakan maksimal dari sebuah unit. Umumnya unit mempunyai pergerakan nilai 5 kecuali unit *wolf* yang mempunyai pergerakan 6 dan unit *dragon* yang mempunyai pergerakan 7.

Juga terdapat variabel groundMap[,] yang digunakan untuk menyimpan bobot dari tiles yang akan dilewati oleh unit. Misalnya

*tile* gunung dan air mempunyai bobot 3 yang maksudnya unit yang akan melewati *tile* tersebut akan menghabiskan 3 poin pergerakan. Sedangkan unit *dragon* tidak terpengaruh oleh bobot *tile* tersebut karena unit *dragon* bergerak di udara.

Pada Kode Sumber 4.5 iterasi terakhir digunakan untuk perhitungan dalam mencari jalan terpendek dari sebuah unit menuju *tile* yang akan dituju. Jika *tile* tersebut mempunyai total jarak kurang dari jarak pergerakan suatu unit maka akan diberi tanda kotak merah pada *tile* tersebut.

```
// fungsi Moving //
If(isBlack==false)
    StartCoroutine(MoveToPosition(target))
```

#### Kode Sumber 4.6 Fungsi Moving

Pada Kode Sumber 4.6 di atas berfungsi untuk memanggil fungsi `MoveToPosition` terus-menerus hingga unit yang dijalankan sampai di posisi target.

```
// fungsi IEnumerator MoveToPosition //
while(position!=target)
    unit.move(target)
    yield return 0;
if(position==target)
    isBlack=true
```

#### Kode Sumber 4.7 Fungsi IEnumerator MoveToPosition

Pada Kode Sumber 4.7 di atas, unit akan berhenti bergerak jika sudah sampai target. Setelah itu sistem akan menampilkan menu aksi dari unit tersebut.



**Gambar 4.6 Tampilan Implementasi Menjalankan Unit**

Gambar 4.6 di atas bagian kiri merupakan tampilan hasil dari fungsi Cek pada Kode Sumber 4.5. Untuk menjalankan unit, pemain harus memilih salah satu dari kotak merah tersebut.

Gambar 4.6 bagian kanan merupakan tampilan menu aksi dari unit setelah berjalan. Jika unit tersebut bisa menguasai rumah dan berjalan menuju rumah maka akan keluar menu kuasai. Dan jika unit tersebut berada di jarak serang dari unit musuh maka akan keluar menu serang.

```
// fungsi Serang //
dmg<- Attacker.dmg * Attacker.hp/100
def<- Defender.armor + armorMap[Defender.position]
Defender.hp-=dmg-def
if(Defender.hp<=0)
  Destroy(Defender)

dmg<- Defender.dmg * Defender.hp/100
def<- Attacker.armor + armorMap[Attacker.position]
Attacker.hp-=dmg-def
if(Attacker.hp<=0)
  Destroy(Attacker)
```

#### Kode Sumber 4.8 Fungsi Serang

Pada Kode Sumber 4.8 dijalankan ketika pemain memilih aksi serang. Dimana akan menampilkan kotak kuning yang menunjukkan pasukan lawan yang berada dalam jarak serang unit pemain. Kode Sumber tersebut juga berisi perhitungan dari penyerangan. Dimana damage dari sebuah unit sama dengan damage unit tersebut dikalikan dengan prosentase *health point* dari unit tersebut. Sedangkan *armor* dari sebuah unit sama dengan *armor* dari unit tersebut ditambah pertahanan dari *tile* dimana unit tersebut berada.



**Gambar 4.7 Tampilan Menyerang Unit Lawan**

Gambar 4.7 di atas bagian kanan merupakan implementasi dari perhitungan damage pada *pseudocode* yang berada pada Kode Sumber 4.8.

### 4.2.3. Implementasi Membeli Pasukan

Implementasi Membeli Pasukan diawali dengan menampilkan menu aksi beli.

```
// fungsi Beli //
If(tokopilih==1 && gold>=150)
    Instantiate(soldier, castle.position)
If(tokopilih==2 && gold>=250)
    Instantiate(archer, castle.position)
```

```
If(tokopilih==3 && gold>=300)
    Instantiate(elemental, castle.position)
If(tokopilih==4 && gold>=400)
    Instantiate(sorcerer, castle.position)
If(tokopilih==5 && gold>=500)
    Instantiate(wisp, castle.position)
If(tokopilih==6 && gold>=600)
    Instantiate(wolf, castle.position)
If(tokopilih==7 && gold>=700)
    Instantiate(golem, castle.position)
If(tokopilih==8 && gold>=800)
    Instantiate(catapult, castle.position)
If(tokopilih==9 && gold>=1000)
    Instantiate(dragon, castle.position)
```

#### Kode Sumber 4.9 Fungsi Beli

Menu aksi beli akan muncul apabila pemain menekan kastil pemain. Pada Kode Sumber 4.9 di atas merupakan *pseudocode* untuk realisasi pembelian unit. Jika pemain membeli unit soldier misalnya maka kondisi pertama pada *pseudocode* di atas yang akan di eksekusi. Pertama sistem membangkitkan unit yang telah dibeli di posisi kastil pemain. Kemudian sistem mengurangi uang dari pemain sebanyak harga dari unit yang dibeli tersebut.

Sistem memunculkan menu aksi beli ketika pemain menekan kastil pemain pada Gambar 4.8 bagian kiri. Gambar 4.8 di bawah merupakan implementasi dari *pseudocode* yang terdapat pada Kode Sumber 4.9. pada gambar di bawah, pemain membeli unit wisp yang berharga 500 gold. Kemudian pada Gambar 4.8 bagian kanan, uang dari pemain tersisa 0 gold setelah melakukan pembelian unit wisp yang berharga 500 gold.



## **BAB V**

### **PENGUJIAN DAN EVALUASI**

Pada bab ini akan dijelaskan mengenai rangkaian uji coba dan evaluasi yang dilakukan. Proses pengujian dilakukan menggunakan metode kotak hitam berdasarkan skenario yang telah ditentukan dan pengujian dilakukan dengan survei langsung kepada pengguna.

#### **5.1. Lingkungan Uji Coba**

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam rangka uji coba perangkat lunak ini dicantumkan pada Tabel 5.1 dan Tabel 5.2.

**Tabel 5.1 Lingkungan Uji Coba Perangkat Lunak (bagian 1)**

Perangkat Keras	Prosesor: Pentium(R) Core(TM) i3-32170 CPU @ 1.80GHz Memori: 4 GB
Perangkat Lunak	Sistem Operasi: Microsoft Windows 8.1 64-bit Perangkat Pengembang: Unity 4.3

**Tabel 5.2 Lingkungan Uji Coba Perangkat Lunak (bagian 2)**

Perangkat Keras	Alat : <i>Smartphone, 4.0 inch, 480×800 pixels</i> <i>Dual-core 1.2 GHz</i> Memori: 1 GB
Perangkat Lunak	Sistem Operasi: Android 4.4

#### **5.2. Pengujian Performa Kecepatan Pembangkit *World***

Pengujian dilakukan untuk menguji seberapa cepat sistem dalam membangkitkan *world*. Pengujian dilakukan dengan

menambahkan *code* pada *script* untuk memudahkan perhitungan waktu pembangkitan world.

### 5.2.1. Skenario Data Uji Coba Kecepatan

Skenario Pengujian kecepatan performa digunakan untuk memberikan tahap-tahap dalam pengujian sistem dalam hal kecepatan. Skenario ini tertera pada Tabel 5.3.

**Tabel 5.3 Skenario Pengujian Performa**

Deskripsi	Bertujuan untuk mengetahui performa sistem dalam membangkitkan world area permainan
<b>Langkah pengujian</b>	<ol style="list-style-type: none"> <li>1. Tambahkan code untuk menghitung waktu pembangkitan pada script.</li> <li>2. Jalankan scene game pada project di unity.</li> <li>3. Pada <i>console</i> Unity akan menampilkan waktu pembangkitan world.</li> <li>4. Ulangi langkah ke-1 sampai ke-4.</li> <li>5. Cari rata-rata waktu kecepatan pembangkitan world.</li> </ol>

```
// menghitung waktu GenerateMap//
waktuawal<- Time.realtimeSinceStartup;
DiamondSquare()
GenerateMap()
waktuakhir<- Time.realtimeSinceStartup;
Print(waktuakhir-waktuawal)
```

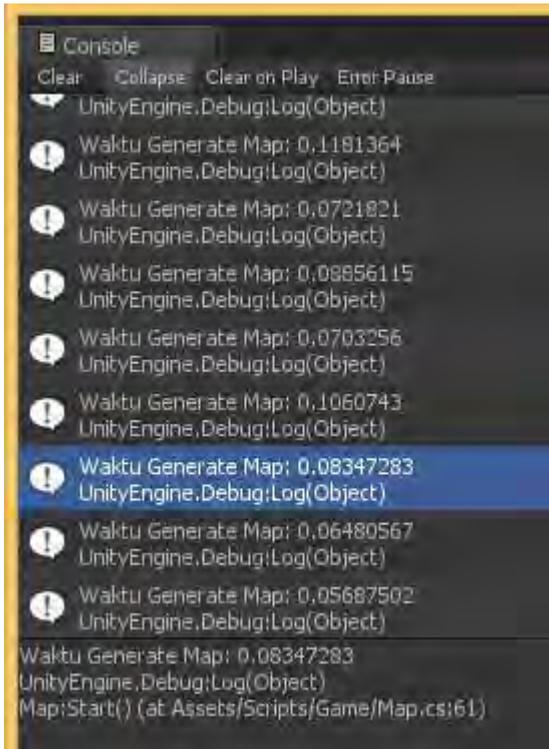
### Kode Sumber 5.1 Menghitung Waktu Fungsi GenerateMap

Pada Kode Sumber 5.1 ditambahkan pada script yang bertujuan untuk menghitung waktu pembangkitan world yang ada pada fungsi GenerateMap().

### 5.2.2. Hasil Pengujian Performa

Hasil pengujian performa dikhususkan untuk menghitung waktu yang dibutuhkan dalam membangkitkan *world* dimana

dijelaskan pada Tabel 5.3. semua nilai pada tabel dihitung dalam satuan detik.



**Gambar 5.1 Tampilan Hasil Kode Perhitungan Waktu**

Pada Gambar 5.1 di atas, merupakan tampilan log dari Unity Game Engine yang merupakan hasil dari Kode Sumber 5.1.

**Tabel 5.4 Hasil Uji Coba Performa Pembangkit World**

<b>Uji Coba ke-</b>	<b>Waktu yang dibutuhkan (detik)</b>
1	0.267
2	0.192
3	0.118
4	0.072
5	0.089
6	0.070
7	0.106
8	0.083
9	0.065
10	0.057
11	0.117
12	0.139
13	0.071
14	0.105
15	0.098
16	0.113
17	0.080
18	0.097
19	0.107
20	0.084
<b>Rata-rata</b>	<b>0.106</b>

Berdasarkan Tabel 5.4, dapat disimpulkan kecepatan pembangkitan world membutuhkan waktu rata-rata 0.106 detik. Waktu eksekusi yang cukup singkat untuk membangun sebuah world.

### **5.3. Pengujian Kesesuaian *World* Yang Dibangkitkan**

Pengujian dilakukan untuk menguji apakah hasil world yang telah dibangkitkan menggunakan algoritma Diamond-Square berjalan seperti yang diharapkan.

### 5.3.1. Skenario Data Uji Coba Kesesuaian *World* Yang Dibangkitkan

Skenario Pengujian *world* yang dibangkitkan digunakan untuk memberikan tahap-tahap dalam pengujian sistem dalam hal *world* yang dibangkitkan. Skenario ini tertera pada.

**Tabel 5.5 Skenario Pengujian *World***

Deskripsi	Bertujuan untuk mengetahui ketepatan dari <i>world</i> yang dibangkitkan
Langkah pengujian	<ol style="list-style-type: none"> <li>1. Jalankan aplikasi sehingga membangkitkan <i>world</i>.</li> <li>2. Simpan hasil <i>world</i> yang berukuran 9x9.</li> <li>3. Rubah <i>tile</i> pada <i>world</i> tersebut menjadi nilai-nilai yang merepresentasikan <i>tile</i> tersebut.</li> <li>4. Pilih beberapa nilai yang merepresentasikan <i>tile</i> tersebut secara acak.</li> <li>5. Hitung nilai tersebut, apakah sudah sesuai dengan algoritma <i>Diamond-Square</i>,</li> <li>6. Ulangi langkah ke-1 sampai ke-4.</li> <li>7. Lihat hasil dari keseluruhan data dari langkah percobaan di atas.</li> </ol>

### 5.3.2. Hasil Pengujian Kesesuaian *World* yang dibangkitkan



**Gambar 5.2 *World* Uji Coba 1**

Pada Gambar 5.2 di atas merupakan hasil *world* pertama untuk uji coba.

2	2	2	2	1	2	1	0	0
1	2	1	0	0	2	2	0	0
2	1	2	0	0	1	2	0	0
1	1	1	0	0	0	1	2	1
1	1	1	0	0	0	1	2	1
1	2	1	1	1	0	0	1	1
1	1	2	1	1	0	0	1	2
1	2	2	2	2	2	2	2	3
1	2	1	2	2	2	2	3	3

**Gambar 5.3 Nilai World Uji Coba 1**

Pada Gambar 5.3, diambil 4 angka random yang diberi warna berbeda.



**Gambar 5.4 World Uji Coba 2**

Pada Gambar 5.4 di atas, merupakan hasil *world* kedua untuk uji coba.

3	3	3	2	1	2	1	1	1
3	1	1	2	2	2	2	1	2
2	1	2	2	2	1	2	1	2
1	2	1	2	1	0	0	0	2
2	1	1	2	1	0	0	0	1
1	1	2	2	0	0	1	2	1
1	1	0	0	1	1	2	1	1
2	1	0	0	2	1	2	2	1
1	2	1	1	1	2	2	1	2

Gambar 5.5 Nilai *World Uji Coba 2*

Pada Gambar 5.5, diambil 4 angka random yang diberi warna berbeda.



3	3	2	2	1	1	2	1	1	1
3	1	2	2	1	1	2	2	2	2
2	2	0	0	1	1	0	0	0	1
2	2	0	0	1	2	0	0	0	2
2	2	1	1	2	1	1	1	1	1
1	3	2	1	0	0	1	0	0	0
2	2	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	1
1	1	1	1	1	0	0	0	0	1

Gambar 5.6 *World Uji Coba 3*

Pada Gambar 5.6 di atas, merupakan hasil *world* ketiga untuk uji coba.

3	3	1	2	2	2	1	2	1
3	1	2	2	1	1	2	1	2
2	2	0	0	1	1	0	0	1
2	2	0	0	1	2	0	0	2
2	1	1	1	2	1	1	1	1
1	3	2	1	0	0	1	0	0
2	2	2	1	0	0	0	0	0
1	1	1	1	0	0	0	0	1
1	1	1	1	1	0	0	0	1

Gambar 5.7 Nilai *World* Uji Coba 3

Pada Gambar 5.7, diambil 4 angka random yang diberi warna berbeda.

Tabel 5.6 Hasil Uji Coba Kesesuaian *World* Yang Dibangkitkan

Uji Coba ke-	Angka Sumber	Rata-rata	Penambahan Angka Random (-1 sampai 1)	Hasil	Keterangan
1	2, 1, 2, 0	1,25	$1,25-1=0,25$	0	Sesuai
2	2, 0, 1, 0	0,75	$0,75+0=0,75$	1	Sesuai
3	1, 2, 2, 2	1,75	$1,75+0=1,75$	2	Sesuai
4	2, 1, 1	1,3	$1,3+1=2,3$	2	Sesuai
5	3, 3, 2, 2	2,5	$2,5-1=1,5$	1	Tidak Sesuai
6	1, 2, 2, 2	1,75	$1,75+0=1,75$	2	Sesuai
7	2, 1, 1, 2	2,5	$2,5-1=1,5$	0	Tidak Sesuai

8	1, 0, 1, 1	0,75	$0,75+0=0,75$	1	Sesuai
9	2, 3, 1, 2	2	$2+0=2$	2	Sesuai
10	0, 2, 0, 1	0,75	$0,75-1=-0,25$	0	Sesuai
11	1, 0, 0	0,6	$0,6-1=-0,4$	0	Sesuai
12	2, 0, 1, 1	1	$1+0=1$	1	Sesuai

Berdasarkan Tabel 5.6 di atas, dapat disimpulkan bahwa hasil *world* yang dibangkitkan sudah sesuai dengan perhitungan algoritma *Diamond-Square*. Dari 12 data uji coba, terdapat 2 yang tidak sesuai yang mungkin disebabkan oleh perubahan angka 0 untuk menyesuaikan bentuk *tile* air. Di sekitar *tile* air harus terdapat minimal 2 *tile* air, kalau tidak *tile* air tersebut akan dirubah menjadi *tile* yang lain untuk menyesuaikan bentuk *tile* air.

## 5.4. Pengujian Fungsionalitas dengan Metode Black-Box

Pengujian ini dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi pada tahap kebutuhan benar-benar diimplementasikan dan bekerja semestinya. Selain itu juga untuk mengetahui kesesuaian keluaran dari setiap tahapan atau langkah penggunaan fitur terhadap skenario yang dipersiapkan. Pengujian dilakukan dengan menggunakan metode *black-box*.

### 5.4.1. Skenario Pengujian Fungsionalitas

Skenario Pengujian fungsionalitas digunakan untuk memberikan tahap-tahap dalam pengujian sistem. Skenario ini tertera pada Tabel 5.7.

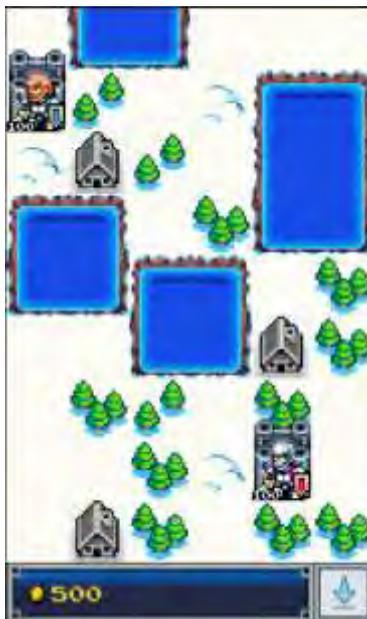
**Tabel 5.7 Pengujian Permainan**

<b>UFG01</b>	
Kondisi Awal	Pengguna berada pada halaman awal permainan.
Prosedur Pengujian	Pengguna memainkan permainan hingga selesai dari beberapa fungsionalitas yang ingin diuji

Hasil yang diharapkan	Pengguna berhasil menyelesaikan permainan dan fungsionalitas permainan berjalan dengan lancar.
Hasil yang diperoleh	Pengguna berhasil menyelesaikan permainan dan fungsionalitas berjalan lancar. Akan tetapi masih terdapat error pada aplikasi.
Kesimpulan.	Pengujian berhasil.

#### 5.4.1.1. Pengujian Skenario Permainan

Pada pengujian ini akan digunakan untuk melihat apakah Skenario dari permainan sudah berjalan dengan baik.



**Gambar 5.8 Tampilan Skenario Permainan**

Skenario permainan pada Gambar 5.8 berjalan dengan baik seperti yang diinginkan, dimana pada awal permainan terdapat dua kerajaan, yaitu kerajaan biru (pemain) dan kerajaan merah (lawan).

Dan juga tiap kerajaan diberikan kastil, raja dan 500 uang. Dan penempatan kastil dan rumah sudah baik seperti yang diinginkan.

#### 5.4.1.2. Pengujian Layar Main Menu

Pengujian ini difokuskan pada saat pemain berada di layar main menu. Dan bertujuan untuk mencoba fungsionalitas dari layar main menu pada permainan.



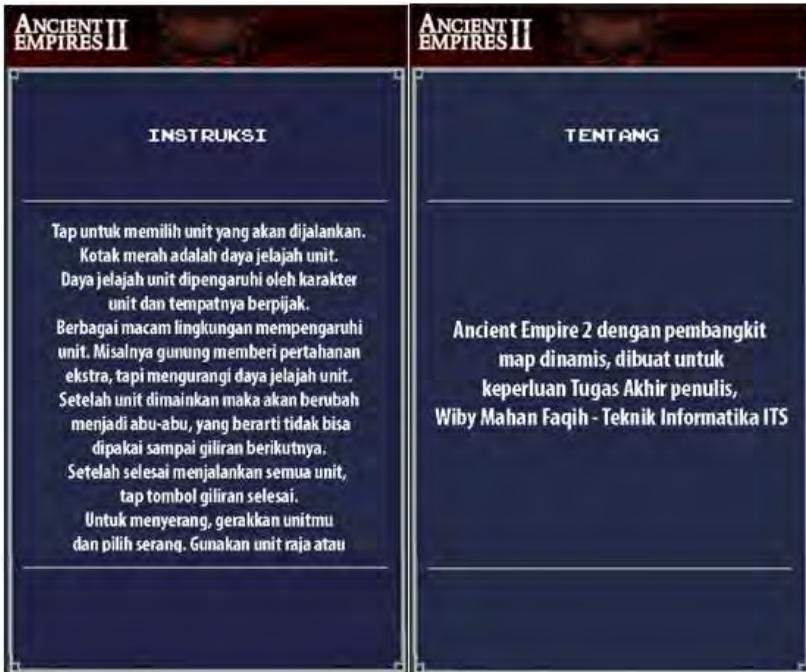
**Gambar 5.9 Tampilan Awal Permainan**

Pada Gambar 5.9 sebelah kiri, fungsi tap to start sudah berjalan sesuai yang diinginkan dengan menampilkan main menu dari permainan seperti pada Gambar 5.9 sebelah kanan.



**Gambar 5.10 Tampilan Menu Setelan**

Pada Gambar 5.10 merupakan tampilan dari setelan menu permainan. Terdapat dua setelan yaitu musik dan map. Untuk merubah setelan, tap pada status on/off. Pada setelan map, kita dapat memilih ukuran map dari 9x9 atau 17x17 atau acak antara kedua ukuran tersebut.



**Gambar 5.11 Tampilan Menu Instruksi dan Tentang**

Pada Gambar 5.11 bagian kiri merupakan tampilan menu instruksi cara memainkan permainan tersebut. Sedangkan pada Gambar 5.11 bagian kanan merupakan tampilan menu tentang dari permainan tersebut.

Dari keseluruhan hasil uji coba pada layar menu permainan, dapat disimpulkan bahwa semua menu telah berjalan seperti yang diharapkan.

### 5.4.1.3. Pengujian Layar Permainan

Pengujian ini difokuskan pada saat pemain berada di layar permainan. Dan bertujuan untuk mencoba fungsionalitas saat bermain. Pertama akan dilakukan pengujian saat menguasai bangunan.



**Gambar 5.12 Tampilan Menguasai Rumah**

Pengujian dalam menguasai rumah, Gambar 5.12, berjalan dengan baik dengan menambah pendapatan pada pemain dan merubah total uang yang dimiliki pemain setelah giliran baru seperti yang ada pada Gambar 5.12 sebelah kanan. Selanjutnya pengujian saat membeli unit pasukan baru.



**Gambar 5.13 Tampilan Membeli Pasukan**

Pengujian dalam membeli unit pasukan berjalan dengan baik seperti yang ada pada Gambar 5.13. uang pemain juga sudah berkurang seharga dengan unit yang dibeli. Selanjutnya pengujian menjalankan pasukan.



**Gambar 5.14 Tampilan Menjalankan Pasukan**

Pada Gambar 5.14 di atas, menu aksi dari unit dragon muncul ketika pemain memilih unit naga dengan menekan unit tersebut. Setelah memilih menu jalan pada menu unit maka akan keluar kotak merah yang menandakan jarak pergerakan dari unit tersebut. Pada saat pemain memilih salah satu dari kotak merah maka unit tersebut akan bergerak menuju kotak merah yang dipilih tersebut.

Pada Gambar 5.14 bagian kanan, setelah unit tersebut sampai pada titik yang dituju, maka unit tersebut akan menampilkan menu aksi. Setelah mengakhiri giliran dari unit tersebut maka unit tersebut akan berubah warna menjadi abu-abu yang menandakan giliran unit tersebut sudah selesai untuk saat ini. Selanjutnya pengujian saat menyerang pasukan lawan.



**Gambar 5.15 Tampilan Menyerang Pasukan Lawan (bagian 1)**

Pada Gambar 5.15, jika unit kita berada di jarak serang unit musuh maka akan keluar menu aksi serang. Pada Gambar 5.15 bagian kanan, kotak kuning menandakan jarak serang unit pemain. Pemain harus memilih pasukan musuh yang berada dalam kotak kuning.



**Gambar 5.16 Tampilan Menyerang Pasukan Lawan (bagian 2)**

Pada Gambar 5.16 bagian kiri menampilkan animasi saat unit pemain menyerang unit lawan. Gambar 5.16 bagian kanan menampilkan hasil dari pertarungan berupa health point yang berkurang. Jika health point dari unit habis maka unit tersebut akan mati. Hasil dari pengujian menyerang pasukan lawan berjalan dengan baik seperti yang diinginkan. Tetapi kemampuan masing-masing unit masih belum terimplementasikan secara keseluruhan. Misalnya unit

archer dapat menyerang dari jarak jauh. Selanjutnya pengujian dalam mengakhiri permainan.



**Gambar 5.17 Mengakhiri Permainan**

Pada Gambar 5.17, untuk mengakhiri permainan, unit raja dari pasukan pemain harus berada pada kastil dari pasukan lawan. Kemudian akan muncul menu kuasai seperti pada Gambar 5.17 bagian kiri. Setelah berhasil menguasai kastil pasukan lawan, akan muncul keterangan tim biru menang, seperti pada Gambar 5.17 bagian kanan. Kemudian akan kembali ke main menu. Selanjutnya pengujian dalam menambah health point dari unit yang berada pada rumah atau kastil kerajaannya.



**Gambar 5.18 Tampilan Menambah *Health Point***

Pada Gambar 5.18 di atas, penambahan *health point* sudah berjalan seperti yang diinginkan. Seperti yang terdapat pada Gambar 5.18, *health point* dari unit dragon bertambah 20 *health point* setelah giliran berikutnya. Jika *health point* kurang dari sama dengan 80 maka akan bertambah 20 *health point*, namun jika *health point* lebih dari 80 maka *health point* akan menjadi 100.

#### **5.4.1.4. Hasil Pengujian Fungsionalitas**

Hasil uji fungsionalitas yang sudah dilakukan berdasarkan pada empat pengujian pokok diatas. Empat uji coba yang telah dilakukan menunjukkan bahwa semua fungsionalitas permainan berjalan dengan baik dan sesuai dengan sebagaimana mestinya skenario yang telah

dibuat pada perancangan. Rekap hasil pengujian fungsionalitas dicantumkan pada Tabel 5.8.

**Tabel 5.8 Hasil Pengujian Fungsionalitas**

No	Nama Pengujian	Hasil Pengujian
1	Pengujian Hasil Pembangkitan World	Berhasil
2	Pengujian Skenario Permainan	Berhasil
3	Pengujian Layar Main Menu	Berhasil
4	Pengujian Layar Permainan	Berhasil

## **5.5. Pengujian Pengguna**

Pengujian pada perangkat lunak yang dibangun tidak hanya dilakukan pada fungsionalitas yang dimiliki, tetapi juga pada pengguna untuk mencoba secara langsung. Pengujian ini berfungsi sebagai pengujian subjektif yang bertujuan untuk mengetahui tingkat keberhasilan aplikasi yang dibangun dari sisi pengguna. Hal ini dapat dicapai dengan meminta penilaian dan tanggapan dari pengguna terhadap sejumlah aspek perangkat lunak yang ada.

### **5.5.1. Skenario Uji Coba Pengguna**

Dalam melakukan pengujian perangkat lunak, penguji diminta mencoba menggunakan perangkat lunak untuk mencoba semua fungsionalitas dan fitur yang ada. Pengujian aplikasi oleh pengguna dilakukan dengan sebelumnya memberikan informasi seputar aplikasi, kegunaan, dan fitur-fitur yang dimiliki. Setelah informasi tersampaikan, pengguna kemudian diarahkan untuk langsung mencoba aplikasi dengan spesifikasi lingkungan yang sama dengan yang telah diuraikan pada uji coba fungsionalitas.

Jumlah pengguna yang terlibat dalam pengujian perangkat lunak sebanyak sepuluh orang. Dalam melakukan pengujian,

pengguna melakukan percobaan lebih dari satu kali penggunaan untuk masing-masing pengguna.

Dalam memberikan penilaian, penulis memberi lembar kuesioner kepada penguji setelah selesai pengujian perangkat lunak. Pertanyaan yang ditanyakan dalam pengujian perangkat lunak ini memiliki beberapa aspek penilaian.

### 5.5.2. Daftar Penguji Perangkat Lunak

Pada subbab ini ditunjukkan daftar pengguna yang bertindak sebagai penguji coba aplikasi yang dibangun. Daftar nama penguji aplikasi ini dapat dilihat pada Tabel 5.9.

**Tabel 5.9 Daftar Nama Penguji Coba Aplikasi**

No	Nama	Pekerjaan
1	Yuan Akbarshar P.	Mahasiswa
2	Irwan Hidayatulloh	Mahasiswa
3	Roshif Syamsudin	Mahasiswa
4	Miftahul Khoir.	Mahasiswa
5	Saadilah Alfaruq	Mahasiswa
6	M. Awwalu B. S.	Mahasiswa
7	Masdar Fawaid	Mahasiswa
8	Lutfi Thobibi	Pelajar
9	M. Ahsani Taqwim	Pelajar
10	Yogi Maulana H.	Pelajar

### 5.5.3. Hasil Uji Coba Pengguna

Uji coba yang dilakukan terhadap beberapa pengguna memiliki beberapa aspek yang dipisahkan berdasarkan antarmuka dan fungsionalitas yang dimiliki. Sistem penilaian didasarkan pada skala penghitungan satu sampai empat di mana skala satu menunjukkan nilai terendah dan skala empat menunjukkan skala tertinggi. Penilaian akhir kemudian dilakukan dengan menghitung berapa banyak penguji yang memilih suatu skala tertentu dan kemudian dicari nilai rata-ratanya. Hasil uji coba dipaparkan secara lengkap dengan disertai

tabel yang dapat dilihat pada subbab. Keterangan nilai pada kuesioner dijelaskan pada Tabel 5.10.

**Tabel 5.10 Keterangan Nilai Pada Kuesioner**

Nilai	Keterangan
1	Kurang
2	Cukup
3	Baik
4	Sangat Baik

### 5.5.3.1. Hasil Penilaian Antarmuka

Penilaian antarmuka difokuskan pada penilaian pengguna terhadap kemudahan penggunaan antarmuka dan sifat-sifat lain yang perlu dimiliki. Hasil penilaian pengguna terhadap antarmuka aplikasi dapat dilihat pada Tabel 5.11.

**Tabel 5.11 Penilaian Antarmuka**

No.	Antarmuka	Penilaian				Rata-Rata
		1	2	3	4	
1	Kemudahan Penggunaan	0	0	8	2	3,2
2	Kelengkapan Menu	0	0	1	9	3,9
3	Keindahan Tampilan	0	0	5	5	3,5
4	Kecepatan Pemilihan Menu/Fitur	0	3	6	1	2,8
5	Kesesuaian tema	0	2	6	2	3
6	Ketertarikan Bermain	0	0	7	3	3,3
<b>Nilai Akhir</b>						<b>3,2</b>

### 5.5.3.2. Hasil Penilaian Performa Sistem

Penilaian performa sistem difokuskan pada penilaian pengguna terhadap kemampuan aplikasi dalam menghasilkan performa dari interaksi pengguna. Penilaian ini juga ditujukan untuk mendapatkan tingkat kecepatan dan kelancaran sistem atas interaksi yang dibuat oleh pengguna. Hasil penilaian performa sistem dapat dilihat pada Tabel 5.12.

**Tabel 5.12 Penilaian Performa Sistem**

No.	Performa Sistem	Penilaian				Rata-Rata
		1	2	3	4	
1	Performa atau kinerja pada permainan	0	1	4	5	3,4
2	Nilai world yang dibangkitkan	0	0	5	5	3,5
3	Ketepatan dalam menampilkan data	0	0	3	7	3,7
4	Kelancaran animasi	0	0	7	3	3,3
5	Kesesuaian animasi dengan sistem permainan	0	0	7	3	3,3
6	Nilai penempatan kastil dan rumah	0	0	2	8	3,8
<b>Nilai Akhir</b>						<b>3,5</b>

### 5.5.4. Hasil Pengujian Pengguna

Evaluasi pengujian pengguna dilakukan dengan menampilkan data rekapitulasi perangkat lunak yang telah dipaparkan. Dari data diketahui bahwa aplikasi telah memenuhi unsur yang seharusnya seperti perancangan dimana mendapatkan nilai akhir 3,2 dan 3,5. Arti dari nilai 3,2 dan 3,5 ini merupakan penilaian yang baik. Permainan yang dibuat telah mendapatkan apresiasi oleh pengguna sebenarnya. Sehingga pula dapat disimpulkan bahwa pengguna secara tidak langsung cukup menyetujui dan memberikan komentar baik kepada aplikasi permainan yang dibuat. Juga mendapat nilai poin sedikit pada

kecepatan pemilihan menu/fitur. Pada kecepatan pemilihan menu/fitur mungkin pada saat membangkitkan world dibutuhkan waktu dua sampai tiga detik jika menggunakan smartphone dengan spesifikasi seperti pada Tabel 5.2, sedangkan dengan menggunakan PC seperti pada Tabel 5.1 diperoleh rata-rata 0,106 detik.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat pula saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

#### **5.6. Kesimpulan**

Dalam proses pengerjaan tugas akhir mulai dari tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Aplikasi berhasil membangun permainan menggunakan pembangkit *world*. Keandalan permainan diuji menggunakan kebutuhan fungsional dan kebutuhan pengguna yang meliputi menu, aturan main, level dan skenario berjalan seperti yang diharapkan. Berdasarkan uraian yang telah disampaikan dapat disimpulkan semua fungsionalitas permainan (menjalankan unit, menyerang unit lawan, membeli unit, menguasai rumah dan kastil, serta mengakhiri giliran) baik itu aturan main ataupun skenario berfungsi seperti yang diharapkan.
2. Berdasarkan uji coba kesesuaian *world* yang dibangkitkan menggunakan algoritma *Diamond-Square*, aplikasi berhasil membangkitkan *world* seperti yang diharapkan.
3. Untuk tingkat kesulitan dan skenario dipengaruhi oleh posisi kastil dan rumah yang di bangkitkan secara acak.
4. Berdasarkan uji performa, aplikasi berhasil membangkitkan *world* dengan waktu minimal 0,057 detik dan maksimal sampai 0,267 detik. Dan memiliki total rata-rata 0,106 detik dari 20 kali percobaan dengan menggunakan PC Pentium Core i3 1.80 GHz 4GB.

### **5.7. Saran**

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi dan uji coba yang telah dilakukan.

1. Penambahan Kecerdasan Buatan yang lebih bagus. Sehingga dapat memberikan tantangan kepada pemain.
2. Membuat permainan mode cerita yang mempunyai alur dan map yang sama dengan game Ancient Empires 2.

## LAMPIRAN

**Nama :** *Soradilum Alfarida*      **Pelajaran :** *Manajemen*

No.	Kelas/teor	Penilaian			
		1	2	3	4
1	Keambahan penggunaan			✓	
2	Kelengkapan menu			✓	✓
3	Selidihan tur/pilan				✓
4	Kecepatan pemilihan menu/fitur			✓	
5	Keseruan tema			✓	
6	Keseruanan bermain			✓	
7	Performa atau kinerja pada permainan			✓	
8	Nilai world yang dibangkitkan			✓	
9	Kelengkapan data dan memaparkan data			✓	
10	Kelancaran animasi			✓	
11	Kesesuaian animasi dengan sistem permainan			✓	
12	Nilai presentasi materi dan rumah			✓	

*( Soradilum Alfarida )*

Nama : Yon Ardiyaningrum P

Pekerjaan : Mahasiswa

No.	Kriteria	Penilaian			
		1	2	3	4
1	Kemudahan penggunaan				✓
2	Kelengkapan menu				✓
3	Kehidupan tampilan			✓	✓
4	Kesepuan pembuatan menu/desa				✓
5	Kesesuaian tema				✓
6	Keterampilan bernilai				✓
7	Performa atau kinerja pada pemantapan				✓
8	Nilai world yang dibagikan				✓
9	Kemampuan dalam menyampaikan data				✓
10	Kelancaran animasi			✓	✓
11	Kesesuaian animasi dengan sistem pemantapan				✓
12	Nilai penampilan hasil dan rumah				✓

Yon Ardiyaningrum P

Nama : Fathif Syamsuddin

Pekerjaan : Mahasiswa

No.	Kategori	Penilaian			
		1	2	3	4
1	Kemudahan penggunaan				✓
2	Kelengkapan fitur				✓
3	Kemudahan tampilan				✓
4	Kecepatan perubahan tersetu fitur				✓
5	Ketersuaian urus				✓
6	Keteraturan, kemain				✓
7	Portofolio atau koneksi pada permasalahan				✓
8	Nilai word yang ditunjukkan				✓
9	Keterampilan dalam menampilkan data				✓
10	Kelancaran animasi				✓
11	Kesesuaian animasi dengan desain permasalahan				✓
12	Nilai keseluruhan ke di dan rumah				✓

Fathif Syamsuddin

Mama : Irwan Hidayatulloh Pekerjaan : Mahasiswa

No.	Kuesioner	Penilaian			
		1	2	3	4
1	Kemudahan menggunakan				✓
2	Kelengkapan fitur				✓
3	Kendaraan tampilan		✓		
4	Kecapahan perubahan tampilan		✓		
5	Kesesuaian tema		✓		
6	Keteraturan bermain		✓		
7	Performa dan kinerja pada permainan		✓		
8	Nilai Waktu yang dibutuhkan		✓		
9	Komponen dalam permainan dan		✓		
10	Kebijakan aplikasi		✓		
11	Kesesuaian aplikasi dengan sistem permainan		✓		
12	Nilai persentase hasil dan grafik		✓		

( Irwan H )

Nama : *Mhammad Akash & S Pakerjan* : *Mhammad*

No.	Kriteria	Penilaian			
		1	2	3	4
1	Kemudahan penggunaan				✓
2	Kelengkapan isi				✓
3	Kelengkapan tampilan				✓
4	Kecepatan pemilahan informasi				✓
5	Kesudahan versi				✓
6	Keterbacaan format				✓
7	Performa atau fungsi pada peramban				✓
8	Nilai world yang ditayangkan				✓
9	Ketepatan dalam menampilkan data				✓
10	Kelancaran animasi				✓
11	Kemudahan akses dengan sistem peramban				✓
12	Nilai pemenuhan kuali dan rumah				✓

*Mhammad Akash & S Pakerjan*

Nama : *Yogi Mahendra W* Pekerjaan : *Pejabat*

No	Kerjanya	Perbaikan			
		1	2	3	4
1	Kemudahan protektifitas			✓	
2	Kelengkapan mesin				✓
3	Kesalahan tarifasi				✓
4	Kecapaian pontifika menaruhur		✓		
5	Kecapaian zona		✓		
6	Kecapaian bertuan		✓		
7	Perforasi atau kinerja pada permukaan		✓		
8	Nilai world yang dimungkinkan		✓		
9	Kecapaian dalam menampilkan data		✓		
10	Kelengkapan asuransi		✓		
11	Kecapaian asuransi dengan standar permukaan		✓		
12	Nilai pelepasan hasil dan rumah				✓

( *Yogi Mahendra W* )

Nama : LUTFI THORISI

Pelajaran : Filsafat

No.	Kuestioni	Penilaian			
		1	2	3	4
1	Kemudahan penggunaan			✓	
2	Kelengkapan menu				✓
3	Kemudahan tampilan				✓
4	Kecepatan pemilhan menu/fitur			✓	
5	Kecerasan tema			✓	
6	Kelengkapan bernilai			✓	
7	Performa atau kinerja pada pemerintahan				✓
8	Nilai world yang ditayangkan				✓
9	Ketepatan dalam memaparkan data			✓	
10	Kelengkapan a.tknial			✓	
11	Kesesuaian animasi dengan sistem pemerintahan				✓
12	Nilai penyempurnaan hasil dan rumah				✓

*Handwritten signature*

Nama: *Syifa Nur Hafid*

Pekerjaan: *Marketing*

No.	Kualifikasi	Penilaian			
		1	2	3	4
1	Kemampuan pengorganisasian			✓	
2	Kelengkapan materi			✓	
3	Ketepatan tanggapan			✓	
4	Kesegaran penampilan narasumber			✓	
5	Kesesuaian tema			✓	
6	Keterampilan bermain				✓
7	Performa atau kinerja pada permainan				✓
8	Nilai world yang dibagikan				✓
9	Ketepatan dalam menyampaikan data				✓
10	Kelancaran animasi			✓	
11	Kesesuaian arsitektur dengan sistem permainan				✓
12	Nilai presentasi keseluruhan				✓



Nama : Misdad Fauzid

Pekerjaan : Mahasiswa

No.	Kategori	Penilaian			
		1	2	3	4
1	Kemudahan pengumpulan			✓	
2	Kelengkapan materi				✓
3	Keindahan tampilan				✓
4	Kecepatan penulisan materi/foto		✓		
5	Kesesuaian tema			✓	
6	Keterarikan bermula			✓	
7	Perbedaan atau kinerjanya pada permainan			✓	
8	Nilai/warta yang disampaikan				✓
9	Ketepatan dalam menyampaikan data				✓
10	Kelancaran animasi			✓	
11	Kesesuaian animasi dengan sistem permainan				✓
12	Nilai pengantar hasil dan rumah				✓

(Fauzid)

Nama : M. Alvin Teguh Petyan

Pelajar : Petyan

No.	Kuesioner	Pembelian			
		1	2	3	4
1	Kemudahan pengigninan			✓	
2	Kelengkapan menu			✓	✓
3	Kelengkapan tampilan			✓	
4	Kecaptesan perubahan menu/hur			✓	
5	Kesesuaian tema			✓	
6	Kesesuaian bernilai			✓	
7	Perforansi atau kinerja pada pembelian			✓	
8	Nilai wujud yang dibangkitkan			✓	
9	Ketepatan dalam menampilkan foto			✓	
10	Kelengkapan atribusi			✓	
11	Kesesuaian animasi dengan sistem pembelian			✓	
12	Nilai pemenuhan hasil dan rumah			✓	✓

(Signature)

## DAFTAR PUSTAKA

- [1] A. Nicolescu, "Black box techniques," 28 4 2014. [Online]. Available: <https://www.qualitance.com/blog/black-box-techniques>. [Accessed 19 12 2014].
- [2] Unity Technologies, "Unity Overview," 19 December 2014. [Online]. Available: <http://docs.unity3d.com/Manual/UnityOverview.html>.
- [3] J. Eldridge, "Best turn-based strategy games, definition and meaning," 13 3 2013. [Online]. Available: <http://www.examiner.com/article/the-definition-of-turn-based-strategy-games-with-game-examples>. [Accessed 19 12 2014].
- [4] "C#.NET Programming," [Online]. Available: <http://brainmatics.com/c-net-programming/>. [Accessed 21 12 2014].
- [5] "Explanation of the Diamond-Square algorithm," 27 9 2010. [Online]. Available: <https://code.google.com/p/fractalterraingeneration/wiki/Diamond-Square>. [Accessed 23 12 2014].
- [6] C. Janssen, "Android Operating System," [Online]. Available: <http://www.techopedia.com/definition/25106/android-operating-system>. [Accessed 19 12 2014].

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Penulis, Wiby Mahan Faqih lahir pada tanggal 30 Mei 1993 di Gresik, merupakan anak pertama dari dua bersaudara. Penulis menempuh pendidikan formal pertamanya di MI Ihyaul Islam Bolo (1999-2005). Kemudian penulis melanjutkan pendidikannya di SMP Negeri 1 Sidayu (2005-2008). Saat SMA penulis melanjutkan pendidikannya di luar kota. Tepatnya di SMA Darul Ulum 2 Jombang (2008-2011). Tidak berhenti disitu masa pendidikan formal penulis.

Saat kuliah penulis memilih untuk berkuliah di Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Di Teknik Informatika ITS, penulis tidak hanya aktif di perkuliahan saja. Penulis juga sempat beberapa kali bergabung dengan beberapa organisasi mahasiswa, seperti HMTC ITS (Himpunan Mahasiswa Teknik Computer Informatika) dan KMI ITS (Keluarga Muslim Informatika). Untuk menghubungi penulis, dapat melewati surel [wiby15@gmail.com](mailto:wiby15@gmail.com) atau melewati akun *facebook* penulis <https://www.facebook.com/wibymf>.