



**TUGAS AKHIR - KI141502**  
**ASYNCHRONOUS MULTIPLAYER CARDGAME**  
**PADA GAME MONSTER KING**

**IKROM AULIA FAHDI**  
**NRP 5112100157**

**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**  
**2016**



**FINAL PROJECT- KI141502**  
**ASYNCHRONOUS MULTIPLAYER CARDGAME**  
**ON GAME OF MONSTER KING**

**IKROM AULIA FAHDI**  
**NRP 5112100157**

**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**  
**2016**

## LEMBAR PENGESAHAN

### ASYNCHRONOUS MULTIPLAYER CARDGAME PADA GAME MONSTER KING

#### Tugas Akhir

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Rumpun Mata Kuliah Interaksi, Grafika, dan Seni  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**IKROM AULIA FAHDI**

NRP. 5121 100 157

Disetujui oleh Dosen Pembimbing Tugas Akhir

Imam Kuswardayan, S.Kom., M.Ts.

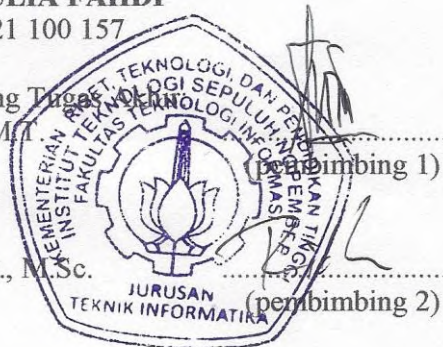
NIP: 19761215 200312 1 001

(pembimbing 1)

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

NIP: 19870213 201404 1 001

(pembimbing 2)



**SURABAYA**

**JUNI, 2016**

## **ASYNCHRONOUS MULTIPLAYER CARDGAME PADA GAME MONSTER KING**

Nama Mahasiswa : Ikrom Aulia Fahdi  
NRP : 5112 100 157  
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**

*Game merupakan salah satu media hiburan yang sangat populer untuk saat ini, dan bermain game merupakan salah satu cara untuk mengisi waktu luang. Hingga saat ini banyak sekali jenis game yang sudah ada, salah satunya adalah card game. Card game merupakan game yang menggunakan kartu sebagai media utama permainan, selain itu juga ada aturan permainan yang harus dipatuhi. Card game ini menggunakan konsep turn based dalam permainannya, sehingga pemain harus bermain secara bergantian dengan player lain.*

*Selain itu juga terdapat jenis game yang lain, yaitu game online. Game online merupakan game yang dapat dimainkan oleh multipemain dengan memanfaatkan jaringan internet. Selain itu game online bisa membuat setiap pemain bermain bersama dan saling berinteraksi baik secara realtime (synchronous) maupun turn based (asynchronous).*

*Monster King merupakan card game yang terhubung dengan internet, sehingga pemain bisa bermain bersama pemain lawan melalui internet. Monster King menggunakan konsep turnbased (asynchronous) dalam aturan mainnya. Selain itu dalam pengembangannya, Monster King menggunakan Photon Realtime agar pemain dapat bermain dengan koneksi yang lancar.*

*Hasil pengujian dari Tugas Akhir ini menunjukkan bahwa pengimplementasian asynchronous dan card game berhasil dilakukan. Framework yang digunakan untuk mendukung*

*implementasi asynchronous pada card game adalah Photon Realtime.*

***Kata kunci: Multiplayer, Card Game, Unity, Photon Cloud, Photon Realtime, Asynchronous***

## **ASYNCHRONOUS MULTIPLAYER CARDGAME ON GAME OF MONSTER KING**

Student Name : Ikrom Aulia Fahdi  
NRP : 51121 100 157  
Major : Teknik Informatika FTIf-ITS  
Advisor I : Imam Kuswardayan, S.Kom., M.T.  
Advisor II : Ridho Rahman Hariadi, S.Kom., M.Sc.

### **ABSTRACT**

*Game is one of popular media entertainment at this time, and playing game is one of way to spend the time. Until now, there are many kind of games, one of which is a card game. Card game is game that use cards as the game's main media, but it also has game rules that must be obeyed by players. Card game use turn-based concept, so players must play alternately with other player.*

*There are also other types of games, the online game. Online game is a game that can be played by multiplayer using internet. And online game can also make every player play together and interact each other both in realtime (synchronous) or turn-based (asynchronous).*

*Monster King is a card game that is connected to the internet, so that player can play with other players using internet. Monster King uses turnbased concept (asynchronous) within the rules. Also Monster King using Photon Realtime so that players can play with a smooth connection.*

*The result of this final project shows that the implementation of asynchronous and card game successfully. Framework that be used to support the implementation of asynchronous in the card game is Photon Realtime.*

***Keywords: Multiplayer, Card Game, Unity, Photon Cloud, Photon Realtime, Asynchronous***

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK .....	vii
ABSTRACT .....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvii
DAFTAR KODE SUMBER .....	xix
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Metodologi .....	3
1.7 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Unity3D .....	5
2.2 Android SDK.....	5
2.3 <i>Photon Realtime</i> .....	5
2.4 <i>Asynchronous Multiplayer</i> .....	6
2.5 <i>Card Game</i> .....	7
2.6 <i>Photon Cloud</i> .....	7
BAB III ANALISIS DAN PERANCANGAN.....	9
3.1 Analisis Perancangan Permainan.....	9
3.1.1 Deskripsi Umum Aplikasi Permainan .....	9
3.1.2 Spesifikasi Kebutuhan Fungsional.....	10
3.1.3 Spesifikasi Kebutuhan Non-Fungsional .....	10
3.1.4 Identifikasi Pengguna .....	10
3.2 Perancangan Permainan.....	11
3.2.1 Perancangan Alur Permainan.....	11
3.2.2 Perancangan Kontrol Permainan .....	17

3.2.3	Perancangan Masukkan Data Sinkronisasi .....	17
3.2.4	Perancangan Antarmuka Permainan .....	17
<b>BAB IV</b>	<b>IMPLEMENTASI</b> .....	<b>23</b>
4.1	Lingkungan Implementasi .....	23
4.2	Implementasi Antarmuka.....	24
4.2.1	Implementasi Antarmuka <i>Home</i> .....	24
4.2.2	Implementasi Antarmuka <i>Input Username</i> .....	24
4.2.3	Implementasi Antarmuka <i>Lobby</i> .....	25
4.2.4	Implementasi Antarmuka <i>Gameplay</i> .....	26
4.3	Implementasi Jaringan Permainan .....	37
4.3.1	Implementasi <i>Matchmaking</i> .....	37
4.3.2	Implementasi <i>Sinkronisasi Data</i> .....	39
<b>BAB V</b>	<b>PENGUJIAN DAN EVALUASI</b> .....	<b>41</b>
5.1	Lingkungan Uji Coba .....	41
5.1.1	Lingkungan Perangkat Keras .....	41
5.1.2	Lingkungan Perangkat Lunak .....	41
5.2	Skenario Pengujian .....	42
5.2.1	Pengujian Fungsionalitas .....	42
5.2.2	Pengujian Terhadap Pengguna.....	50
5.3	Evaluasi Pengujian.....	51
5.3.1	Evaluasi Pengujian Fungsionalitas.....	51
5.3.2	Evaluasi Pengujian Terhadap Pengguna .....	52
<b>BAB VI</b>	<b>KESIMPULAN DAN SARAN</b> .....	<b>57</b>
6.1.	Kesimpulan .....	57
6.2.	Saran .....	57
<b>DAFTAR PUSTAKA</b> .....		<b>59</b>
<b>LAMPIRAN</b> .....		<b>61</b>
<b>BIODATA PENULIS</b> .....		<b>71</b>



## DAFTAR GAMBAR

Gambar 2.1 Konsep <i>Time Gap</i> .....	6
Gambar 2.2 Arsitektur <i>Photon Cloud</i> .....	7
Gambar 3.1 Perancangan Alur Koneksi ke Photon Cloud .....	11
Gambar 3.2 Perancangan Alur <i>Main Menu</i> .....	12
Gambar 3.3 Perancangan Alur <i>Waiting Game</i> .....	13
Gambar 3.4 Perancangan Alur Menang Kalah.....	14
Gambar 3.5 Perancangan Alur Aksi Monster.....	15
Gambar 3.6 Perancangan Alur Memanggil Monster.....	16
Gambar 3.7 Perancangan Alur Menyerang Bangunan Utama ....	16
Gambar 3.8 Perancangan Antarmuka <i>Home</i> .....	18
Gambar 3.9 Perancangan Antarmuka <i>Input Username</i> .....	18
Gambar 3.10 Perancangan Antarmuka <i>Lobby</i> .....	19
Gambar 3.11 Perancangan Antarmuka <i>Waiting Player</i> .....	19
Gambar 3.12 Perancangan Antarmuka <i>Gameplay</i> .....	20
Gambar 3.13 Perancangan Antarmuka <i>Popup Gameplay</i> .....	21
Gambar 3.14 Perancangan Antarmuka Hasil Permainan .....	21
Gambar 4.1 Implementasi Antarmuka <i>Home</i> .....	24
Gambar 4.2 Implementasi Antarmuka <i>Input Username</i> .....	25
Gambar 4.3 Implementasi Antarmuka <i>Lobby</i> .....	26
Gambar 4.4 Implementasi Antarmuka <i>Waiting Player</i> .....	26
Gambar 4.5 Implementasi Antarmuka <i>Gameplay</i> .....	27
Gambar 4.6 Implementasi Antarmuka <i>Gameplay</i> Saat Pemain Memanggil Monster .....	27
Gambar 4.7 Implementasi Antarmuka <i>Gameplay</i> Saat Muncul Tile Yang Berbeda.....	30
Gambar 4.8 Implementasi Antarmuka <i>Gameplay</i> Saat Monster Bergerak .....	30
Gambar 4.9 Implementasi Antarmuka <i>Gameplay</i> Saat Monster Menyerang.....	33
Gambar 4.10 Implementasi Antarmuka <i>Gameplay</i> Saat Monster Menyerang <i>Base</i> Pemain lawan .....	33
Gambar 4.11 Implementasi Antarmuka <i>Popup Menu Gameplay</i>	36

Gambar 4.12 Implementasi Antarmuka Hasil Permainan .....	37
Gambar 5.1 Kondisi Awal Uji Coba Permainan Menang .....	43
Gambar 5.2 Kondisi Akhir Uji Coba Permainan Menang .....	43
Gambar 5.3 Kondisi Awal Uji Coba Permainan Kalah .....	44
Gambar 5.4 Kondisi Akhir Uji Coba Permainan Kalah .....	45
Gambar 5.5 Kondisi Awal Uji Coba <i>Create Room</i> .....	45
Gambar 5.6 Kondisi Akhir Uji Coba <i>Create Room</i> .....	46
Gambar 5.7 Kondisi Awal Pengujian <i>Join Random Room</i> .....	47
Gambar 5.8 Kondisi Akhir Pengujian <i>Join Random Room</i> .....	47
Gambar 5.9 Tampilan Saat Pemain ke-1 .....	48
Gambar 5.10 Tampilan Saat Pemain ke-20 .....	48
Gambar 5.11 Tampilan Pemain Ke-21 .....	49
Gambar 5.12 Tampilan <i>Log Error</i> Pemain Ke-21 .....	49
Gambar A.1 Kuesioner Responden Pertama .....	61
Gambar A.2 Kuesioner Responden Kedua.....	62
Gambar A.3 Kuesioner Responden Ketiga.....	63
Gambar A.4 Kuesioner Responden Keempat .....	64
Gambar A.5 Kuesioner Responden Kelima .....	65
Gambar A.6 Kuesioner Responden Keenam .....	66
Gambar A.7 Kuesioner Responden Ketujuh .....	67
Gambar A.8 Kuesioner Responden Kedelapan .....	68
Gambar A.9 Kuesioner Responden Kesembilan .....	69
Gambar A.10 Kuesioner Responden Kesepuluh .....	70

## DAFTAR TABEL

Tabel 3.1 Identifikasi Pengguna .....	10
Tabel 3.2 Detil Data Sinkronisasi.....	17
Tabel 4.1 Lingkungan Implementasi Pengembangan .....	23
Tabel 4.2 Lingkungan Implementasi <i>Debugging</i> .....	23
Tabel 5.1 Lingkungan Perangkat Keras .....	41
Tabel 5.2 Lingkungan Perangkat Lunak .....	41
Tabel 5.3 Hasil Uji Coba Permainan Menang.....	42
Tabel 5.4 Hasil Uji Coba Permainan Kalah .....	44
Tabel 5.5 Hasil Uji Coba <i>Create Room</i> .....	46
Tabel 5.6 Hasil Uji Coba <i>Join Random Room</i> .....	47
Tabel 5.7 Hasil Uji Coba Performa Photon.....	49
Tabel 5.8 Kuesioner Pengguna.....	50
Tabel 5.9 Hasil Pengujian Fungsionalitas .....	51
Tabel 5.10 Hasil Kuesioner .....	52
Tabel 5.11 Hasil Kuesioner untuk Pengguna Mahasiswa Teknik Informatika .....	54
Tabel 5.12 Hasil Kuesioner untuk Pengguna non-Mahasiswa Teknik Informatika.....	54

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Memanggil Monster .....	28
Kode Sumber 4.2 Munculnya <i>tile</i> .....	29
Kode Sumber 4.3 Menggerakkan Monster .....	31
Kode Sumber 4.4 Monster Menyerang .....	32
Kode Sumber 4.5 Monster Menyerang <i>Base</i> .....	34
Kode Sumber 4.6 Proses <i>end turn</i> pemain .....	35
Kode Sumber 4.7 Hasil Permainan .....	36
Kode Sumber 4.8 <i>State</i> Pemain Saat Berada di <i>Lobby</i> .....	38
Kode Sumber 4.9 Implementasi <i>Create Room</i> .....	38
Kode Sumber 4.10 Implementasi <i>Join Random Room</i> .....	39
Kode Sumber 4.11 Implementasi Sinkronisasi Data.....	39

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

*Game* merupakan salah satu media hiburan yang sangat populer untuk saat ini, dan bermain game merupakan salah satu cara untuk mengisi waktu luang. Hingga saat ini banyak sekali jenis *game* yang sudah ada, salah satunya adalah *Card game*. *Card game* merupakan *game* yang menggunakan kartu sebagai media utama permainan, selain itu juga ada aturan permainan yang harus dipatuhi [1]. *Card game* ini menggunakan konsep *turn based* dalam permainannya, sehingga pemain harus bermain secara bergantian dengan pemain lawan dan bisa dimainkan lebih dari 1 orang pemain.

Selain itu juga terdapat jenis *game online* yang bisa membuat setiap pemain bermain bersama dan saling berinteraksi baik secara *realtime (synchronous)* maupun *turn based (asynchronous)* yang bisa diakses melalui internet menggunakan perangkat *android, iphone, windows phone*, dll. *Game online* tersebut menggunakan *server* untuk menangani para pengguna dalam berinteraksi.

Saat ini lumayan banyak *game online* yang menggunakan konsep *turn based (Asynchronous)* dimana interaksi antar pengguna tidak dipertemukan secara *realtime*. Tetapi masih sedikit *Card game* yang menggunakan konsep *turn based* secara *online*. Pada *game* ini setiap pemain akan saling menyerang dan bertahan menggunakan monster yang telah di-*summon* oleh setiap pemain. Selain itu pemain bisa menyerang bangunan utama dari pemain lawan. Permainan akan berakhir jika *Life Points* bangunan utama dari salah satu tim sudah mencapai nilai 0.

*Game* ini utamanya akan berjalan pada perangkat *Android* dan untuk menghandle fitur *multiplayer* menggunakan *framework Photon Realtime* [2] sehingga pengguna dapat berinteraksi dengan sesama pengguna perangkat *Android*.

## 1.2 Rumusan Masalah

Berikut rumusan masalah yang diangkat dalam tugas akhir ini adalah :

1. Bagaimana merancang *gameplay card game* menggunakan konsep *turn based*?
2. Bagaimana mengimplementasikan mekanisme *Asynchronous* dalam *card game* berbasis perangkat bergerak?
3. Bagaimana implementasi dari rumusan di atas diselesaikan dengan *Game Engine Unity*?

## 1.3 Batasan Masalah

Berikut batasan masalah pada tugas akhir ini adalah:

1. Pembuatan *game* ini akan menggunakan IDE Unity dan bahasa pemrograman C#.
2. Dalam 1 pertandingan hanya terdapat 2 pemain.
3. Total maksimum CCU (*Concurrent Users*) yang dapat terhubung ke Photon Cloud yaitu 20 CCU.
4. Hanya bisa dimainkan pada *platform android*

## 1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah untuk membuat aplikasi yang mengimplementasikan konsep *turn based* (*Asynchronous*) dalam *gameplay card game*.

## 1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini antara lain:

1. Sebagai bentuk implementasi konsep *turn based* (*Asynchronous*) pada *card game*.
2. Mengasah kemampuan dan kecepatan berpikir pemain dalam menyusun strategi untuk memenangkan permainan.
3. Sebagai sarana hiburan untuk para pengguna.

## 1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

### A. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu mengenai *Game Engine Unity3D*, *Android SDK*, *Photon Realtime*, konsep mengenai *Asynchronous Multiplayer* dan *Card game* itu sendiri.

### B. Analisis dan desain perangkat lunak

Analisis dan perancangan dalam pembuatan *Game Monster King* antara lain sebagai berikut :

1. Mempelajari konsep mekanisme *Asynchronous Multiplayer*.
2. Mempelajari dokumentasi *Photon Realtime*.
3. Perancangan *gameplay*.

### C. Implementasi perangkat lunak

*Game* ini dibangun dengan menggunakan bahasa C# pada *game engine Unity3D*. Selain itu, menggunakan *Photon Realtime* untuk menangani masalah jaringan / *Asynchronous multiplayer* pada *game* ini.

### D. Pengujian dan evaluasi

Pengujian akan dilakukan kepada 5 mahasiswa Informatika dan 5 mahasiswa non Informatika, mereka akan menjadi tester dan memainkan *Monster King*. Pengujian dilakukan untuk mengukur tingkat kegunaan dari *game* yang dibuat dalam membantu pengguna. Selain itu akan dilakukan pengujian terhadap jaringan.

### E. Penyusunan buku 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 alur permainan dan antarmuka.

### **BAB IV IMPLEMENTASI**

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembuatan *Game Monster King*.

### **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.



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Unity3D**

Unity merupakan sebuah *game engine* yang berbasis *cross platform* dan berfungsi untuk membantu orang mengembangkan *game environment* 2D maupun 3D [3]. Selain itu, Unity merupakan sebuah *tool* yang terintegrasi untuk membuat arsitektur bangunan dan simulasi. Unity dapat digunakan untuk membuat sebuah game yang bisa digunakan pada Windows, Mac, Xbox 360, PlayStation 3, Wii, iPad, Iphone, dan Android. Unity bisa untuk *games PC* dan *games Online*. Untuk *games Online* diperlukan sebuah *plugin*, yaitu Unity Web Player, sama halnya dengan Flash Player pada *Browser*.

#### **2.2 Android SDK**

Android-SDK merupakan *tools* bagi para programmer yang ingin mengembangkan aplikasi berbasis *android* [4]. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan *tutorial*. Persyaratan mencakup JDK, Apache Ant dan Python 2.2 atau yang lebih baru. IDE yang didukung secara resmi adalah Android Studio. Dengan menggunakan android studio ini pengembang dapat menggunakan teks editor untuk mengedit *file* Java dan XML serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* aplikasi Android, dll.

#### **2.3 Photon Realtime**

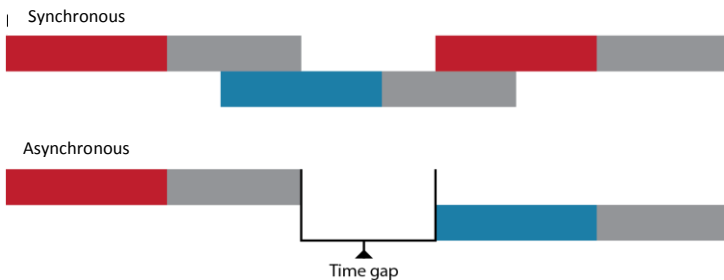
Photon Realtime adalah sebuah *framework Software Development Kit* untuk IDE Unity yang dapat digunakan untuk mengembangkan *multiplayer game* baik yang bertipe *synchronous* maupun *asynchronous*. Model arsitektur dalam pembuatan *game multiplayer* salah satunya adalah SaaS (Software as a Service).

Kemampuan *web service* pada *server*nya memiliki stabilitas yang sangat direkomendasikan untuk para pengembang yang menggunakan Unity dalam pengembangan permainan.

Berbeda dengan Photon Unity Networking, Photon Realtime ini memiliki *class* dan fitur yang bisa membantu *developer* untuk membuat *multiplayer game* yang bertipe *asynchronous* [5]. Sedangkan PUN itu sendiri susah jika digunakan untuk membuat *asynchronous multiplayer game*. PUN lebih cocok jika digunakan untuk membuat *synchronous multiplayer game*.

## 2.4 Asynchronous Multiplayer

*Asynchronous Multiplayer* merupakan metode permainan *multiplayer* dimana pemain tidak harus bermain dalam satu waktu / *state* yang sama dengan pemain lawan dan tidak ada giliran / *turn based game* seperti catur, dll. Setelah pemain A melakukan interaksi terhadap pemain B, pemain A tersebut tidak memperhatikan respon apa yang akan diberikan oleh pemain B. Sedangkan pemain B bisa memberi respon kapan saja terhadap interaksi yang diberikan oleh pemain A. Selisih antara waktu awal interaksi dan waktu respon disebut Time Gap. Time Gap pada *Asynchronous* ini tidak dibatasi, jadi pemain B bisa melakukan respon kapan saja. Berikut penjelasan perbedaan konsep antara *synchronous* dan *asynchronous* pada gambar 2.1



**Gambar 2.1 Konsep Time Gap**

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan tentang analisis dan perancangan aplikasi *card game* Monster King dengan menggunakan Unity dan Photon Realtime. Analisis terdiri dari deskripsi umum perangkat lunak, spesifikasi kebutuhan fungsional dan non-fungsional serta identifikasi pengguna. Sedangkan perancangan terdiri dari perancangan alur permainan, perancangan kontrol permainan, perancangan masukan data sinkronisasi, dan perancangan antarmuka permainan.

#### **3.1 Analisis Perancangan Permainan**

Pada Subbab ini akan dibahas mengenai analisis system yang meliputi deskripsi umum perangkat lunak, spesifikasi kebutuhan fungsional, spesifikasi kebutuhan non-fungsional, dan karakteristik pengguna.

##### **3.1.1 Deskripsi Umum Aplikasi Permainan**

Tugas akhir yang akan dikembangkan adalah sebuah permainan 2D bergenre *cardgame* dengan konsep *asynchronous*. Permainan dengan judul “*Monster King*” ini dikembangkan untuk perangkat *android*. *Game* ini bisa dimainkan secara *multiplayer* melalui koneksi internet dan menggunakan *Framework* Photon Realtime. Untuk mensinkronisasikan antar pemain, aplikasi permainan harus dapat mengatur pertukaran data antar klien untuk setiap *state*. Setiap klien akan mengirimkan datanya dan menerima data dari *cloud server*. *Framework* Photon Realtime mengatur proses ini agar performa permainan tetap stabil.

*Game* Monster King ini bercerita tentang pertempuran antara ras monster dan ras manusia. Mereka masing-masing juga memiliki bangunan utama yang harus dilindungi agar tidak dihancurkan oleh pemain lawan. Bangunan utama tersebut memiliki *Life Points* sebanyak 100 *points*. Masing-masing *player*

memiliki 4 buah kartu yang bisa digunakan. Masing – masing pemain membutuhkan *mana* untuk memanggil monster / karakter. Energi tersebut didapatkan dari bangunan utama, bangunan utama ini yang menghasilkan 1 buah *mana* setiap turn.

### 3.1.2 Spesifikasi Kebutuhan Fungsional

Berdasarkan deskripsi umum aplikasi permainan, terdapat kebutuhan fungsional dari aplikasi permainan ini yaitu dapat bermain *multiplayer* secara *asynchronous* dengan pemain lawan.

### 3.1.3 Spesifikasi Kebutuhan Non-Fungsional

Terdapat beberapa kebutuh non-fungsional yang jika dipenuhi, maka dapat meningkatkan kualitas dari aplikasi permainan. Berikut daftar kebutuhan non-fungsional:

1. Aplikasi permainan akan optimal dengan menggunakan layar dengan aspek rasio 16:9.
2. Pemain dapat bermain secara lancar, nyaman dan mampu menangani *bug*.

### 3.1.4 Identifikasi Pengguna

Berdasarkan deskripsi umum diatas, maka dapat diketahui bahwa pengguna yang akan menggunakan aplikasi ini hanya satu yaitu pemain yang menjalankan permainan dan berinteraksi antar pemain. Penjelasan tersebut terdapat pada Tabel 3.1.

**Tabel 3.1 Identifikasi Pengguna**

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

## 3.2 Perancangan Permainan

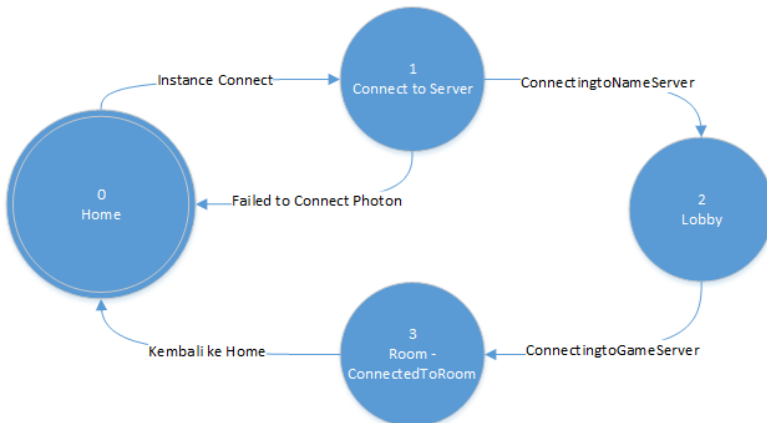
Tahap perancangan dalam subbab ini dibagi menjadi beberapa bagian yaitu perancangan alur permainan, perancangan kontrol permainan, perancangan masukkan data sinkronisasi dan perancangan antarmuka permainan.

### 3.2.1 Perancangan Alur Permainan

Pada rancangan proses permainan ini akan dijelaskan mengenai proses yang terjadi dalam permainan. Proses terdiri dari alur *state* pada *main menu* dan *gameplay* pada *game*.

#### 3.2.1.1 Perancangan Alur Koneksi ke Photon Cloud

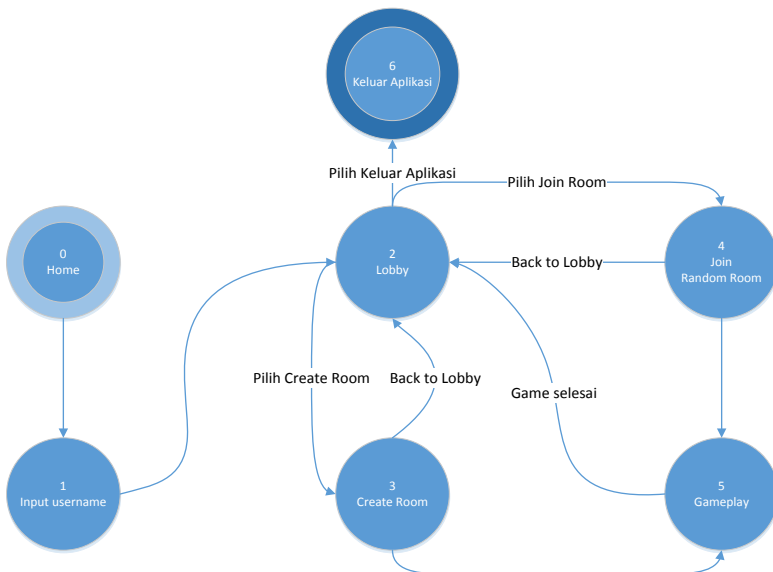
Alur ini merupakan alur terhubungnya aplikasi ke *cloud server*. Proses ini sendiri terdiri dari *state-state* yang terdapat pada *cloud server* hingga pemain bisa bertemu dengan *player* lain. Alur dapat digambarkan dengan FSM diagram yang ditunjukkan pada Gambar 3.1.



**Gambar 3.1 Perancangan Alur Koneksi ke Photon Cloud**

### 3.2.1.2 Perancangan Alur *Main Menu*

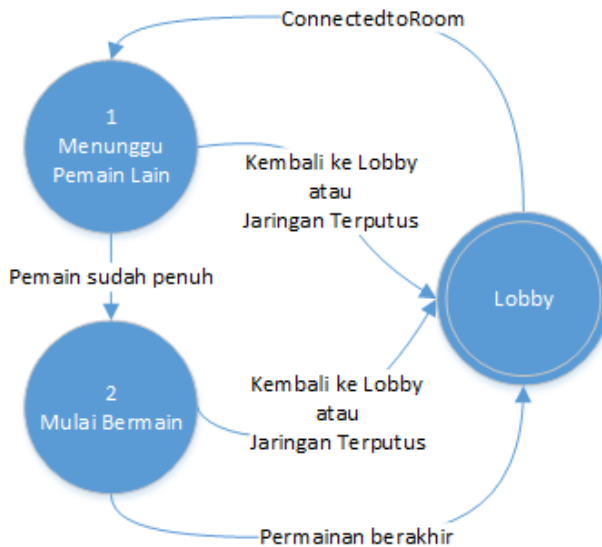
Alur ini merupakan alur perpindahan sistem menu aplikasi hingga memulai permainan. Saat membuka aplikasi, pemain akan berada pada halaman awal aplikasi, lalu saat pemain menekan tombol *play* pada layar, maka pemain akan berpindah halaman dan disuruh untuk memasukkan *username*. Saat pemain menekan tombol *input*, secara otomatis pemain akan terkoneksi dengan *Photon Cloud* dengan menggunakan *username* tersebut. Setelah terhubung, pemain akan berpindah ke halaman *Lobby*. Terdapat pilihan menu untuk *create room* dan *join random room*. Jika memilih tombol *create room*, maka pemain akan membuat *room*, lalu masuk *loading panel* untuk menunggu pemain lawan. Jika memilih menu *join random room*, maka pemain akan langsung masuk *room* secara *random*. Alur dapat digambarkan dengan FSM (*Finite State Machine*) diagram yang dapat ditunjukkan pada Gambar 3.2.



**Gambar 3.2 Perancangan Alur *Main Menu***

### 3.2.1.3 Perancangan Alur *Waiting Game*

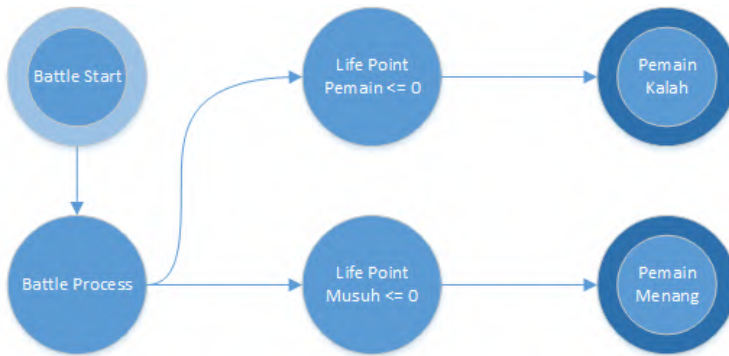
Alur ini merupakan alur untuk menunggu pemain lawan yang akan terhubung dengan *room* yang sama. Jumlah maksimal pemain dalam 1 *room* adalah 2 pemain. Permainan akan dimulai jika jumlah pemain dalam 1 *room* sudah mencapai 2 pemain. Alur dapat digambarkan dengan FSM diagram yang ditunjukkan pada Gambar 3.3.



**Gambar 3.3** Perancangan Alur *Waiting Game*

### 3.2.1.4 Perancangan Alur Menang Kalah

Alur ini merupakan alur untuk menentukan siapakah yang menang atau kalah dalam suatu pertandingan. Indikator yang digunakan untuk menentukan siapa yang menang atau kalah dalam suatu pertandingan adalah *Life Point* pemain. Alur dapat digambarkan dengan FSM diagram yang ditunjukkan pada Gambar 3.4.



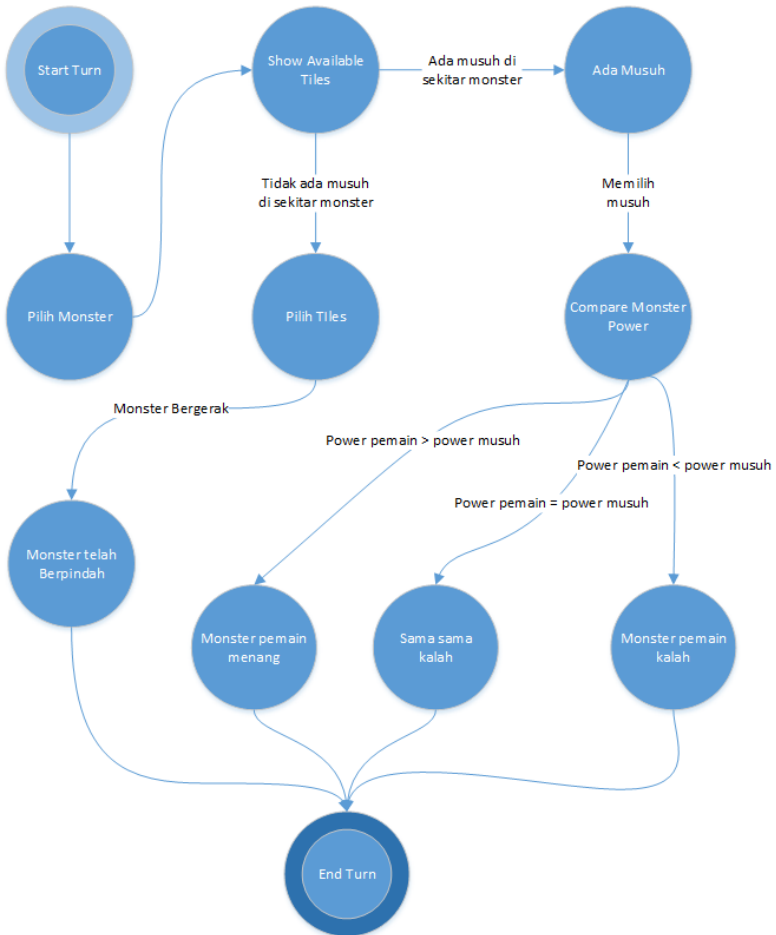
**Gambar 3.4 Perancangan Alur Menang Kalah**

### 3.2.1.5 Perancangan Alur Aksi Monster

Alur ini merupakan alur yang menjelaskan aksi apa yang bisa dilakukan oleh monster, antara lain bergerak dan menyerang. Saat tiba gilirannya, pemain bisa memilih monster yang telah dipanggil, setelah itu akan muncul *tile* yang bisa dituju oleh monster tersebut. *Tile* yang tersedia untuk setiap monster itu berbeda, tergantung dari kemampuan setiap monster. Jika tidak ada monster pemain lawan di sekitar monster pemain, maka pemain bisa menggerakkan monster tersebut ke *tile* yang tersedia. Setelah itu monster dapat berpindah menuju *tile* yang dituju. Setelah berpindah, monster tidak bisa melakukan apapun hingga giliran berikutnya.

Sebaliknya, jika terdapat monster pemain lawan di sekitar pemain lawan pemain, maka pemain bisa memilih aksi untuk menyerang monster pemain lawan. Pada setiap monster memiliki *power* masing-masing, *power* ini digunakan untuk menyerang monster pemain lawan maupun *base* pemain lawan. Saat *power* salah satu monster lebih besar daripada monster lain, maka monster yang memiliki *power* lebih kecil akan kalah / hancur, dan sebaliknya. Sedangkan jika *power* yang dimiliki kedua monster sama besar, maka kedua monster tersebut akan hancur. Alur dapat digambarkan dengan FSM diagram yang ditunjukkan pada Gambar 3.5.



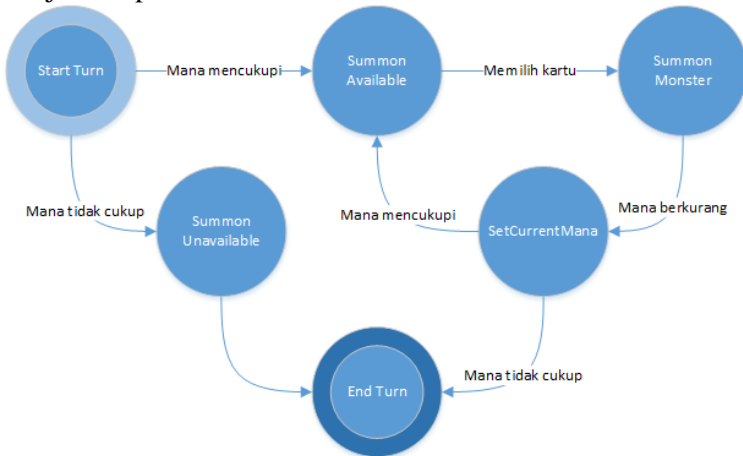


**Gambar 3.5 Perancangan Alur Aksi Monster**

### 3.2.1.6 Perancangan Alur Memanggil Monster

Alur ini merupakan alur untuk memanggil monster dari pilihan kartu yang dimiliki tiap pemain menuju *field*. Sebelum memanggil monster, dicek terlebih dahulu apakah *mana* yang dimiliki oleh pemain cukup untuk memanggil monster, jika tidak

maka pemain tidak bisa memanggil monster tersebut, dan sebaliknya. Alur dapat digambarkan dengan FSM diagram yang ditunjukkan pada Gambar 3.6.



**Gambar 3.6 Perancangan Alur Memanggil Monster**

### 3.2.1.7 Perancangan Alur Menyerang Bangunan Utama Pemain lawan

Alur ini merupakan alur untuk menyerang bangunan utama dari pemain lawan. Saat monster berada di *tile* paling dekat dengan bangunan utama pemain lawan, maka monster tersebut bisa menyerang bangunan utama pemain lawan. Lalu *life point* bangunan utama akan berkurang sebanyak *power* yang dimiliki monster tersebut. Alur dapat digambarkan dengan FSM diagram yang ditunjukkan pada Gambar 3.7



**Gambar 3.7 Perancangan Alur Menyerang Bangunan Utama**

### 3.2.2 Perancangan Kontrol Permainan

Terdapat 1 jenis kontrol dalam permainan yaitu *tap*. Kontrol *tap* digunakan untuk memilih semua menu, menggerakkan, memilih, menyerang monster, dan menyerang bangunan utama.

### 3.2.3 Perancangan Masukkan Data Sinkronisasi

Perancangan ini adalah mekanisme sinkronisasi dari masukkan data ketika sebuah klien melakukan *streaming* data ke Photon Cloud. Sinkronisasi data terjadi ketika pemain mengakhiri gilirannya. Saat salah satu pemain belum mengakhiri gilirannya, maka pemain lawan menunggu hingga gilirannya tiba dan tampilan aplikasi pada kedua pemain tersebut bisa saja beda. Pada aplikasi permainan ini akan dijabarkan tipe data apa saja yang digunakan pada saat sinkronisasi permainan. Berikut detail penjabaran pada Tabel 3.2.

**Tabel 3.2 Detil Data Sinkronisasi**

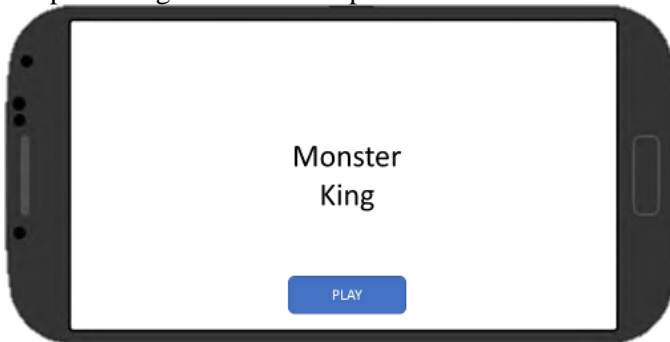
<b>Nama Sinkronisasi</b>	<b>Detil</b>	<b>Data</b>
<i>Turn Player</i>	Giliran pemain	<i>Int</i>
<i>Monsters Position</i>	Posisi untuk setiap monster	<i>Array of Int</i>
<i>Life Point</i>	Kapasitas <i>Health</i>	<i>Float</i>

### 3.2.4 Perancangan Antarmuka Permainan

Subbab ini membahas bagaimana rancangan antarmuka permainan yang akan digunakan untuk tugas akhir. Rancangan antarmuka yang dibahas meliputi ketentuan masukan dan rancangan jendela tampilan. Dalam aplikasi ini terdapat beberapa tampilan yang dijelaskan pada subbab berikut.

### 3.2.4.1 Perancangan Antarmuka *Home*

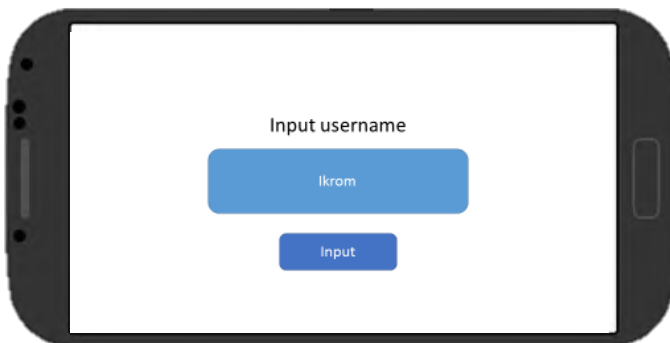
Pada antarmuka ini, aplikasi permainan akan menampilkan halaman awal ketika pemain baru saja membuka aplikasi. Terdapat tombol *Play* agar pemain bisa memulai permainan. Berikut ilustrasi perancangan antarmuka pada Gambar 3.8.



Gambar 3.8 Perancangan Antarmuka *Home*

### 3.2.4.2 Perancangan Antarmuka *Input Username*

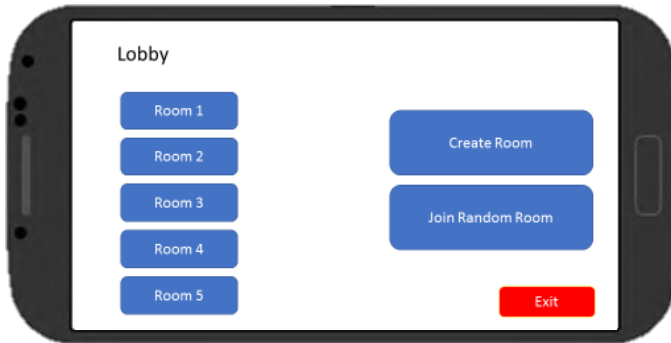
Pada antarmuka ini, akan menampilkan *form* yang digunakan untuk memasukkan *username* yang diinginkan oleh *user*. *Username* tersebut akan digunakan sebagai identitas pemain selama permainan berlangsung. Berikut ilustrasi perancangan antarmuka pada Gambar 3.9.



Gambar 3.9 Perancangan Antarmuka *Input Username*

### 3.2.4.3 Perancangan Antarmuka *Lobby*

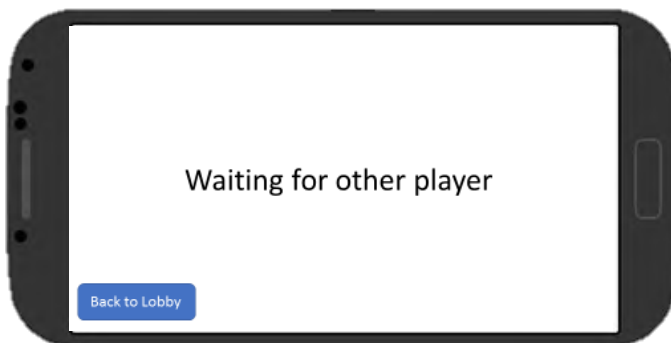
Pada antarmuka ini, menampilkan informasi mengenai *lobby*. Pada antarmuka ini terdapat *list room* yang tersedia, tombol *create room*, *join random room* dan *exit* untuk keluar aplikasi. Berikut ilustrasi perancangan antarmuka pada Gambar 3.10.



Gambar 3.10 Perancangan Antarmuka *Lobby*

### 3.2.4.4 Perancangan Antarmuka *Waiting Player*

Antarmuka ini muncul ketika jumlah pemain dalam 1 *room* kurang dari 2, sehingga harus menunggu pemain lawan. Terdapat tombol *Back to Lobby* jika pemain ingin kembali ke halaman *lobby*. Berikut ilustrasi perancangan antarmuka pada Gambar 3.11.



Gambar 3.11 Perancangan Antarmuka *Waiting Player*

### 3.2.4.5 Perancangan Antarmuka *Gameplay*

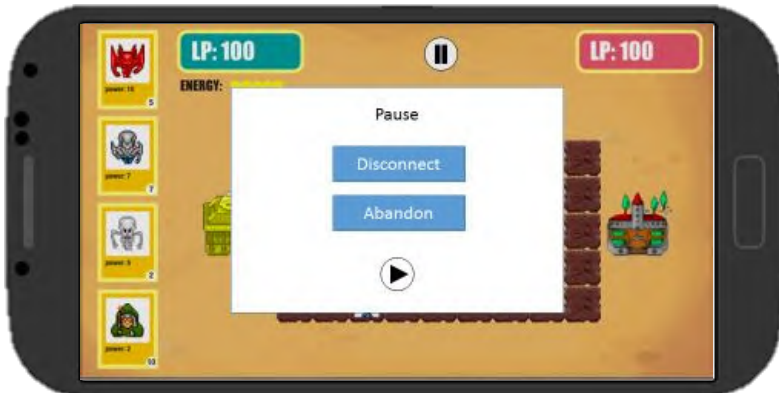
Pada antarmuka ini, menampilkan informasi mengenai tampilan *gameplay* dari aplikasi. Pada tampilan ini terdapat status mengenai pemain yang sedang bermain, seperti *username*, *life point* dan *mana* yang dimiliki pemain. Selain itu terdapat *field* yang digunakan sebagai tempat untuk memanggil monster yang dimiliki pemain. Terdapat juga 4 buah kartu yang dimiliki setiap pemain. Pada sebelah kanan dan kiri *field*, terdapat bangunan utama/*base* yang harus dilindungi dan *base* pemain lawan yang harus diserang agar bisa memenangkan permainan. Berikut ilustrasi perancangan antarmuka pada Gambar 3.12.



Gambar 3.12 Perancangan Antarmuka *Gameplay*

### 3.2.4.6 Perancangan Antarmuka *Popup Gameplay*

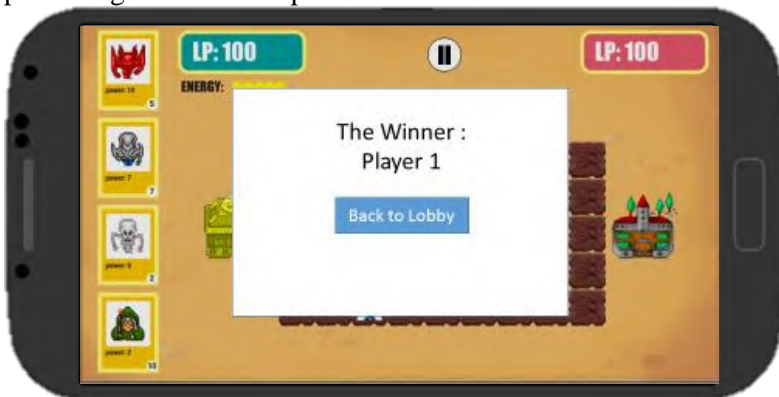
Pada antarmuka ini, menampilkan informasi mengenai *Popup* yang berada di dalam antarmuka *gameplay*. *Popup* ini muncul ketika pemain menekan tombol *pause*. Di dalam tampilan *popup* ini terdapat tombol *disconnect* untuk keluar sementara dari permainan, tombol *resume* untuk kembali ke dalam permainan dan tombol *abaddon* untuk keluar dan meninggalkan permainan, sehingga otomatis pemain tersebut akan dianggap kalah. Berikut ilustrasi perancangan antarmuka pada Gambar 3.13.



Gambar 3.13 Perancangan Antarmuka *Popup Gameplay*

### 3.2.4.7 Perancangan Antarmuka Hasil Permainan

Pada antarmuka ini, menampilkan informasi mengenai hasil akhir permainan ketika terdapat salah pemain yang *life pointnya*  $\leq 0$ . Setiap pemain akan mengetahui siapa yang memenangkan permainan. Selain itu terdapat tombol *Back to Lobby*, agar pemain bisa kembali ke *lobby* setelah permainan berakhir. Berikut ilustrasi perancangan antarmuka pada Gambar 3.14.



Gambar 3.14 Perancangan Antarmuka Hasil Permainan

*(Halaman ini sengaja dikosongkan)*



## **BAB IV IMPLEMENTASI**

Pada bab ini akan dibahas mengenai implementasi dari perancangan aplikasi permainan. Di dalamnya mencakup proses penerapan dan pengimplementasian algoritma dan antarmuka yang mengacu pada rancangan yang telah dibahas sebelumnya.

### **4.1 Lingkungan Implementasi**

Lingkungan implementasi merupakan lingkungan dimana aplikasi akan dibangun. Lingkungan implementasi dibagi menjadi dua, yaitu lingkungan implementasi untuk pengembangan aplikasi yang ditunjukkan pada Tabel 4.1 dan lingkungan implementasi untuk *debugging* yang ditunjukkan pada Tabel 4.2.

**Tabel 4.1 Lingkungan Implementasi Pengembangan**

Perangkat Keras	Prosesor : - Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz Memori : - 4 GB
Perangkat Lunak	Sistem Operasi : - Microsoft Windows Embedded 8.1 Industry Pro 64-bit Perangkat Pengembang : - Unity Engine - Mono Develop

**Tabel 4.2 Lingkungan Implementasi *Debugging***

Perangkat Keras	Tipe: Asus Padfone S T00N Prosesor: Quad-core 2.3 GHz Krait 400 Memori: 2048MB RAM
Perangkat Lunak	Sistem Operasi: Android OS, v4.4.2 (KitKat)

## 4.2 Implementasi Antarmuka

Pada subbab ini, akan dijelaskan implementasi dari perancangan antarmuka permainan yang terdapat pada bab 3. Terdapat 4 antarmuka utama dalam aplikasi permainan ini, antara lain antarmuka *home*, *input username lobby*, dan *gameplay*. Desain antarmuka tersebut dirancang oleh mahasiswa Desain Komunikasi Visual ITS yang bernama Yoghi Cahyo Nugroho.

### 4.2.1 Implementasi Antarmuka *Home*

Pada saat pemain masuk ke dalam aplikasi permainan, maka tampilan yang pertama kali muncul adalah antarmuka *home*. Pada antarmuka ini terdapat logo dari *game Monster King*. Selain itu terdapat 1 buah tombol *Play* pada antarmuka ini. Ketika pemain menekan tombol tersebut, maka pemain akan masuk ke dalam antarmuka selanjutnya, yaitu *lobby*. Antarmuka *home* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Implementasi Antarmuka *Home*

### 4.2.2 Implementasi Antarmuka *Input Username*

Antarmuka *Input Username* muncul ketika pemain pertama kali memainkan aplikasi permainan ini. Jadi ketika pemain menekan tombol *play* pada antarmuka *home*, aplikasi pemain ini

akan mengecek pada *playerprefab*, jika *username* pada *playerprefab* tidak ditemukan, maka *form* akan kosong, dan sebaliknya. Pada antarmuka ini pemain disuruh untuk memasukkan *username* yang akan digunakan sepanjang permainan. Selanjutnya *username* tersebut akan disimpan pada *playerprefab*. Antarmuka *input username* dapat dilihat pada Gambar 4.2.



**Gambar 4.2 Implementasi Antarmuka *Input Username***

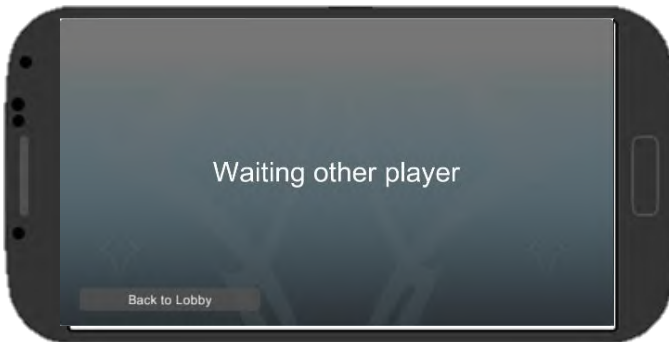
### 4.2.3 Implementasi Antarmuka *Lobby*

Pada antarmuka *Lobby* ini terdapat 3 tombol, yaitu *create room*, *join random room*, dan *exit application*. Tombol *create room* berfungsi untuk membuat *room* yang nantinya akan digunakan oleh para pemain untuk bermain. Tombol *join random room* berfungsi agar pemain bisa bergabung / *join* pada *room* yang tersedia. Pemain hanya bisa bergabung dengan *room* yang jumlah pemainnya kurang dari 2.

Setelah pemain memilih *create room* / *join random room*, maka pemain akan berpindah menuju antarmuka *waiting player*. Pemain akan berada pada antarmuka ini selama jumlah pemain pada *room* tersebut kurang dari 2. Jika jumlah pemain pada *room* tersebut sama dengan 2, maka kedua pemain tersebut akan berpindah menuju antarmuka *gameplay*. Antarmuka *lobby* dapat dilihat pada Gambar 4.3, sedangkan antarmuka *waiting player* dapat dilihat pada Gambar 4.4.



**Gambar 4.3 Implementasi Antarmuka *Lobby***



**Gambar 4.4 Implementasi Antarmuka *Waiting Player***

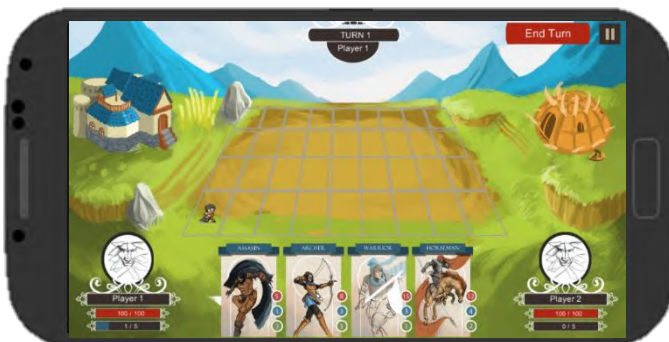
#### **4.2.4 Implementasi Antarmuka *Gameplay***

Antarmuka *gameplay* merupakan tampilan saat pemain sedang bermain dalam 1 *room*. Dalam antarmuka terdapat berbagai informasi, seperti *turn player*, sekarang *turn* berapa, nama pemain 1 dan pemain 2, *life point* dan *mana* pemain 1 dan pemain 2, *list* kartu yang dimiliki pemain 1 dan pemain 2. Selain itu juga terdapat tombol *end turn* dan *pause* pada pojok kanan atas. Antarmuka *gameplay* dapat dilihat pada Gambar 4.5.



**Gambar 4.5 Implementasi Antarmuka *Gameplay***

Pada saat pemain sedang berada di dalam gilirannya, maka pemain bisa memilih kartu dan memanggil monster sesuai dengan mana yang dimilikinya. Setelah itu pemain bisa memanggil monster tersebut ke dalam arena permainan (*field*). Setelah pemain memanggil monster, pemain tidak bisa langsung menggerakkan monster ataupun memilih aksi yang lainnya, pemain harus menunggu giliran berikutnya. Antarmuka *gameplay* pada saat pemain memanggil monster bisa dilihat pada Gambar 4.6.



**Gambar 4.6 Implementasi Antarmuka *Gameplay* Saat Pemain Memanggil Monster**

## Proses memanggil monster bisa dilihat pada Kode Sumber 4.1 Memanggil Monster.

```

public void DoAction(){
    //summon card
    if (CardsController.selectedCard >= 0) {
        if (!tiles.IsHasChild (TilesController.selected
TileId) && (tiles.IsFirstColumn (TilesController.select
edTileId) || tiles.IsLastColumn (TilesController.select
edTileId))) {
            cards.SummonCard ();
        }
        tiles.DisableTiles ();
        CardsController.selectedCard = -1;
    }
}

public void SummonCard(){
    countCard = 0;
    monsterObj = Instantiate (monsterList [selectedCar
d]) as GameObject;
    monsterObj.GetComponent<Image> ().sprite = artex [
selectedCard];
    monsterObj.transform.SetParent (tiles.tilesPanel.t
ransform.GetChild (TilesController.selectedTileId));
    monsterObj.GetComponent<RectTransform> ().anchored
Position = new Vector3 (0, 0, 0);
    if (GameplayController.turnPlayer == 1) tiles.Disa
bleFirstColumnTiles ();
    else tiles.DisableLastColumnTiles ();

    if(GameplayController.turnPlayer == 1) gameplay.ac
tiveTilePlayer1.Add(TilesController.selectedTileId);
    else if(GameplayController.turnPlayer == 2) gamepl
ay.activeTilePlayer2.Add(TilesController.selectedTileId
);
}
}

```

**Kode Sumber 4.1 Memanggil Monster**

Setelah pemain memanggil monster ke dalam *field*, pemain bisa memilih monster tersebut dan memilih aksi selanjutnya, apakah bergerak atau menyerang pemain lawan / *base* pemain lawan. Saat pemain memilih monster, maka akan muncul *tile* yang berbeda di sekitar pemain. *Tile* tersebut merupakan *tile* yang bisa dipilih oleh pemain untuk monster tersebut, jadi monster tersebut hanya bisa bergerak / menyerang pada *tile* yang muncul. Jika pada saat terdapat monster teman maupun pemain lawan pada sekitar monster pemain, maka *tile* yang muncul akan berhenti hingga lokasi *tile* monster teman / pemain lawan, *tile* yang muncul tidak akan melewati *tile* dimana monster lain berada. Antarmuka *gameplay* ketika muncul *tile* yang berbeda bisa dilihat pada Gambar 4.7. Proses munculnya *tile* dapat dilihat pada Kode Sumber 4.2

```

public void DoAction(){
    int idMonster;

    if (tiles.IsHasChild (TilesController.selectedTile
    Id) && tiles.tilesPanel.transform.GetChild (TilesContro
    ller.selectedTileId).GetComponent<Button> ().image.color
    != Color.red) {
        getMonster = true;
        currentMonster = GameObject.Find ("Tiles").trans
        form.GetChild (TilesController.selectedTileId).GetCompo
        nentInChildren<SpriteRenderer> ();
        //show active tile
        if (!show) {
            idMonster = dataMonster.GetId(TilesController
            .selectedTileId);
            tiles.show (dataMonster.GetMovevement(idMonst
            er));
            currentTile = TilesController.selectedTileId;
            show = true;
        }
    }
}

```

**Kode Sumber 4.2 Munculnya *tile***



**Gambar 4.7 Implementasi Antarmuka Gameplay Saat Muncul Tile Yang Berbeda**

Pemain bisa memilih untuk menggerakkan / menyuruh monster menyerang monster lain. Ketika pemain ingin menggerakkan monster ke *tile* yang tersedia, maka pemain hanya tinggal memilih *tile* yang dituju, lalu monster tersebut akan langsung pindah menuju *tile* yang dituju. Setelah monster tersebut pindah, maka monster tersebut akan menjadi *disable* hingga giliran berikutnya. Jangkauan *tile* tiap monster berbeda-beda, tergantung dari kemampuan monster tersebut. Antarmuka *gameplay* saat monster bergerak bisa dilihat pada Gambar 4.8. Proses saat monster bergerak dapat dilihat pada Kode Sumber 4.3.



**Gambar 4.8 Implementasi Antarmuka *Gameplay* Saat Monster Bergerak**



```

public void MoveMonster(int currentTile){
    int index;
    if (GameplayController.turnPlayer == 1) {
        index = gameplay.activeTilePlayer1.IndexOf (curr
entTile);
        gameplay.activeTilePlayer1.RemoveAt (index);
        gameplay.activeTilePlayer1.Add (TilesController
.selectedTileId);
        tiles.tilesPanel.transform.GetChild (TilesControll
er.selectedTileId).GetComponent<Button> ().interactable
= false;
    }
    else if (GameplayController.turnPlayer == 2) {
        index = gameplay.activeTilePlayer2.IndexOf (cur
rentTile);
        gameplay.activeTilePlayer2.RemoveAt (index);
        gameplay.activeTilePlayer2.Add (TilesController
.selectedTileId);
        tiles.tilesPanel.transform.GetChild (TilesContr
oller.selectedTileId).GetComponent<Button> ().interacta
ble = false;
    }

    SpriteRenderer monster = gameplay.currentMonster;
    if (!tiles.IsHasCHILD((TilesController.selectedTile
Id))) {
        monster.transform.SetParent (tiles.tilesPanel.t
ransform.GetChild (TilesController.selectedTileId));
        monster.GetComponent<RectTransform> ().anchored
Position = new Vector3 (0, 0, 0);
        gameObject.GetComponent<TilesController> ().Dis
ableTiles ();
    }
}

```

**Kode Sumber 4.3 Menggerakkan Monster**

Saat terdapat monster pemain lawan di sekitar monster pemain, maka *tile* yang muncul akan berubah warna menjadi merah dan pemain bisa menyerang monster pemain lawan tersebut. Jika monster pemain lawan tidak berada di sekitar monster pemain, maka *tile* akan tetap berwarna putih. Monster yang memiliki *power*

paling rendah akan kalah, tetapi jika kedua monster memiliki *power* yang sama, maka kedua monster akan hancur. Antarmuka *gameplay* saat monster menyerang dapat dilihat pada Gambar 4.9. Proses saat monster menyerang dapat dilihat pada Kode Sumber 4.4.

```

public void Attack(int monster1, int monster2){
    int index;
    int id1 = dataMonster.GetId (monster1);
    int id2 = dataMonster.GetId (monster2);
    int power1 = dataMonster.GetPower(id1);
    int power2 = dataMonster.GetPower (id2);
    if (power1 > power2) {
        index = gameplay.activeTilePlayer2.IndexOf (Til
esController.selectedTileId);
        gameplay.activeTilePlayer2.RemoveAt (index);
        Destroy (tiles.tilesPanel.transform.GetChild (m
onster2).transform.GetChild(0).gameObject);
        tiles.tilesPanel.transform.GetChild (monster1).
GetComponent<Button> ().interactable = false;
        tiles.tilesPanel.transform.GetChild (monster2).
GetComponent<Button> ().interactable = false;
    } else if (power1 < power2) {
        Destroy (tiles.tilesPanel.transform.GetChild (m
onster1).transform.GetChild(0).gameObject);
        tiles.tilesPanel.transform.GetChild (monster1).
GetComponent<Button> ().interactable = false;
    } else if (power1 == power2) {
        Destroy (tiles.tilesPanel.transform.GetChild (m
onster1).transform.GetChild(0).gameObject);
        Destroy (tiles.tilesPanel.transform.GetChild (m
onster2).transform.GetChild(0).gameObject);
        tiles.tilesPanel.transform.GetChild (monster1).
GetComponent<Button> ().interactable = false;
        tiles.tilesPanel.transform.GetChild (monster2).
GetComponent<Button> ().interactable = false;
    }
    tiles.DisableTiles ();
}

```

**Kode Sumber 4.4 Monster Menyerang**



**Gambar 4.9 Implementasi Antarmuka *Gameplay* Saat Monster Menyerang**

Saat monster berada di *tile* yang paling dekat dengan *base* pemain lawan, *base* pemain lawan akan berubah warna menjadi merah. Setelah itu monster tersebut akan secara otomatis menyerang *base* tersebut dan mengurangi *life point* pemain sebesar *power* yang dimiliki oleh monster tersebut. Antarmuka *gameplay* saat monster menyerang *base* pemain lawan dapat dilihat pada Gambar 4.10. Proses saat monster menyerang *base* pemain lawan dapat dilihat Kode Sumber 4.5



**Gambar 4.10 Implementasi Antarmuka *Gameplay* Saat Monster Menyerang *Base* Pemain lawan**

```

//attack base
public void AttackBase(int destinationPlayer){
    int idMonster, powerMonster;
    if (destinationPlayer == 2) {
        idMonster = dataMonster.GetId (TilesController.s
electedTileId);
        powerMonster = dataMonster.GetPower (idMonster);
        gameplay.lifePoint2 -= powerMonster;
        if (gameplay.lifePoint2 < 0) gameplay.lifePoint2
= 0;
    }
    else if (destinationPlayer == 1) {
        idMonster = dataMonster.GetId (TilesController.s
electedTileId);
        powerMonster = dataMonster.GetPower (idMonster);
        gameplay.lifePoint1 -= powerMonster;
        if (gameplay.lifePoint1 < 0) gameplay.lifePoint1
= 0;
    }
}

```

#### Kode Sumber 4.5 Monster Menyerang Base

Tombol *end turn* berfungsi untuk mengakhiri giliran dari suatu pemain. Pada setiap *turn* pemain, pemain bisa menggerakkan monster dan memanggil monster. Selain itu mana untuk setiap pemain akan bertambah setelah berakhirnya giliran pemain tersebut. Pada giliran selanjutnya, setiap monster yang dimiliki oleh pemain pada giliran tersebut akan aktif. Saat proses *end turn*, *list* pemain monster akan muncul sesuai dengan giliran pemain yang sekarang. Saat *end turn* juga terjadi proses pengecekan apakah ada monster di *tile* yang paling dekat dengan monster. Jika ada, maka *life point* pemain akan berkurang dengan sendirinya sesuai dengan *power* tersebut. Proses *end turn* dapat dilihat pada kode Kode Sumber 4.6

```

public void EndTurn(){
    turnCount++;
    //turn for player 1
    if (turnCount % 2 != 0) {
        turnPlayer = 1;
        if (manaPlayer1 < 5) manaPlayer1++;
        panelPlayer1.SetActive (true);
        panelPlayer2.SetActive (false);
        tiles.CheckTilesNearBase(2);
    }
    //turn for player 2
    else {
        turnPlayer = 2;
        if (manaPlayer2 < 5) manaPlayer2++;
        panelPlayer2.SetActive (true);
        panelPlayer1.SetActive (false);
        tiles.CheckTilesNearBase(1);
    }
    tiles.DisableMonsterOtherPlayer();
    tiles.DisableLastColumnTiles ();
    tiles.ActiveMonsterTile();
    cards.countCard = 0;
    cards.checkCard.Clear ();
}

```

#### Kode Sumber 4.6 Proses *end turn* pemain

Pada antarmuka ini juga terdapat tombol *pause* yang berada di sebelah kanan atas. Saat pemain menekan tombol tersebut, maka akan muncul *popup menu gameplay*. Pada antarmuka *popup menu gameplay* terdapat beberapa tombol, antar lain tombol, *abaddon*, dan *resume game*. Jika pemain menekan tombol *abaddon*, maka pemain meninggalkan *room* dan langsung kalah. Sedangkan tombol *resume game* berfungsi jika pemain ingin kembali ke antarmuka *gameplay*. Saat pemain menekan tombol ini, maka pemain tersebut tidak bisa berinteraksi dengan *monster / tile* yang berada di antarmuka *gameplay*. Antarmuka *popup menu gameplay* dapat dilihat pada Gambar 4.11.



**Gambar 4.11 Implementasi Antarmuka *Popup Menu Gameplay***

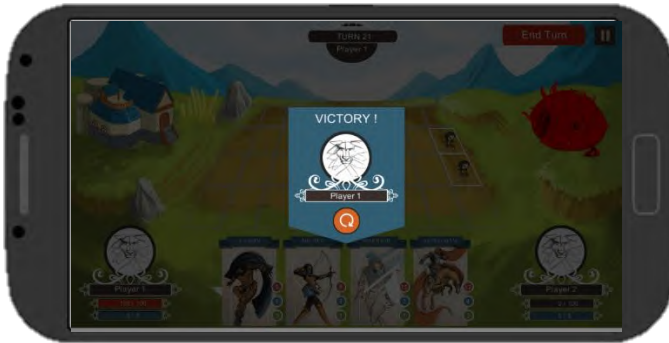
Ketika salah satu pemain memiliki *life point*  $\leq 0$ , maka permainan tersebut dinyatakan kalah dan permainan akan berakhir. Setelah itu akan muncul antarmuka hasil permainan. Pada antarmuka hasil permainan terdapat tombol back to lobby. Jika pemain memilih tombol ini, maka pemain akan kembali menuju antarmuka *lobby*. Antarmuka hasil permainan dapat dilihat pada Gambar 4.12. Proses hasil permainan dapat dilihat pada Kode Sumber 4.7.

```

void Update(){
    if (lifePoint1 <= 0) {
        winningPanel.SetActive (true);
        winningPanel.gameObject.transform.GetChild(2).
GetComponentInChildren<Text>().text = "Player 2";
    } else if (lifePoint2 <= 0) {
        winningPanel.SetActive (true);
        winningPanel.gameObject.transform.GetChild(2).
GetComponentInChildren<Text>().text = "Player 1";
    }
}

```

**Kode Sumber 4.7 Hasil Permainan**



**Gambar 4.12 Implementasi Antarmuka Hasil Permainan**

### 4.3 Implementasi Jaringan Permainan

Untuk merealisasikan komponen jaringan pada aplikasi permainan, maka dibutuhkan pengatur jaringan untuk mengatur *state-state* yang ada pada Photon dan merealisasikan algoritma *matchmaking* untuk mempertemukan antar pemain. Selain itu implementasi jaringan berfungsi untuk membuat *room*, *join room*, dan sinkronisasi data untuk setiap giliran.

#### 4.3.1 Implementasi *Matchmaking*

*Matchmaking* merupakan proses untuk mempertemukan satu pemain dengan pemain lawan yang akan bermain bersama dalam satu *room*. Sebelum setiap pemain bertemu dan selanjutnya bertanding dengan pemain lawan, mereka akan berkumpul terlebih dahulu di dalam *lobby*. Pada saat pemain sedang berada di dalam *lobby*, maka *state* dari pemain tersebut adalah *JoinedLobby*. Setelah pemain memiliki *state JoinedLobby*, selanjutnya panel *lobby* akan aktif sehingga pemain bisa melihat antarmuka *lobby*. Untuk antarmuka *lobby* itu sendiri bisa dilihat pada Gambar 4.3. Pada saat pemain berada di dalam *lobby*, pemain bisa memilih untuk *create room*, *join random room* atau bahkan keluar aplikasi. Proses saat pemain berada di dalam *lobby* dapat dilihat pada Kode Sumber 4.8.

```
private void OnStateChanged(ClientState state) {
    if (state == ClientState.JoinedLobby)
    {
        ChangeActivePanel (lobbyPanel);
    }
}
```

**Kode Sumber 4.8 State Pemain Saat Berada di Lobby**

#### 4.3.1.1 Implementasi Create Room

Setelah berada di dalam *lobby*, pemain bisa memilih untuk *create room*. Diperlukan beberapa parameter untuk membuat *room*, salah satunya nama *room*. Pada aplikasi ini nama *room* akan mengambil dari nama *user* itu sendiri. Setelah itu pemain akan membuat *room* tersebut dan menunggu pemain lawan untuk bermain. Pemain yang dibutuhkan dalam 1 *room* adalah 2 orang. Proses pembuatan *room* dapat dilihat pada

```
public void CreateRoom() {
    roomName = this.username + "'s - Room";
    RoomOptions roomProp = new RoomOptions () {
        MaxPlayers = 2,
        EmptyRoomTtl = 5000,
        PlayerTtl = int.MaxValue
    };
    this.GameInstance.OpCreateRoom (roomName, roomProp,
    TypedLobby.Default);
    GM.SetActive(true);
}
```

**Kode Sumber 4.9 Implementasi Create Room**

#### 4.3.1.2 Implementasi Join Random Room

*Join* yang digunakan pada aplikasi ini adalah *Join Random Room*. Dimana pemain akan *join* terhadap salah satu *room* yang tersedia. Jadi pemain tidak perlu memilih ingin masuk *room* yang mana. Pemain hanya bisa masuk ke dalam *room* yang jumlah



pemainnya kurang dari 2. Proses *join random room* bisa dilihat pada Kode Sumber 4.10.

```
public void Join(){
    this.GameInstance.OpJoinRandomRoom (null, 0);
    ChangeActivePanel (gameplayPanel);
}
```

#### Kode Sumber 4.10 Implementasi *Join Random Room*

### 4.3.2 Implementasi *Sinkronisasi Data*

Untuk setiap *turn*, diperlukan sinkronisasi data untuk setiap pemain agar pemain dapat mengetahui *state* / keadaan dari pemain lawan. Sinkronisasi data ini menggunakan *hashtable* yang menyimpan beberapa variabel penting dalam permainan. Setelah giliran pemain selesai, maka pemain tersebut akan mengirim *hashtable* tersebut kemudian akan dibaca oleh pemain lawan dan mengambil nilai variabel dari *hashtable* tersebut. Proses sinkronisasi data dapat dilihat pada Kode Sumber 4.11.

```
public void SaveTileAsProperty(){
    Tiles tile = GameObject.FindObjectOfType(typeof
(Tiles)) as tile;
    this.turnNumber = this.turnNumber + 1;
    this.lastTurnPlayer = this.LocalPlayer.ID;

    Hashtable tileProps = tile.GetTileAsCustomProperties();
    tileProps .Add("lt", this.lastTurnPlayer);
    tileProps .Add("t#", this.turnNumber);

    this.OpSetCustomPropertiesOfRoom(tileProps);

    Debug.Log("saved tile to props " + SupportClass
.DictionaryToString(tileProps)); }
```

#### Kode Sumber 4.11 Implementasi Sinkronisasi Data

*(Halaman ini sengaja dikosongkan)*

## **BAB V**

### **PENGUJIAN DAN EVALUASI**

Bab ini berisi bahasan mengenai uji coba dan evaluasi aplikasi permainan Monster King menggunakan Unity Engine dengan Photon Realtime. Uji coba dilakukan menggunakan metode kotak hitam berdasarkan skenario yang telah ditentukan. Uji coba dilakukan terhadap hasil implementasi perangkat lunak yang telah dipaparkan.

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

Proses uji coba dilakukan pada lingkungan yang telah ditentukan. Pada uji coba ini, lingkungan dibedakan menjadi lingkungan perangkat keras dan lingkungan perangkat lunak. Berikut ini dijelaskan mengenai tiap-tiap lingkungan uji coba aplikasi.

##### **5.1.1 Lingkungan Perangkat Keras**

Deskripsi perangkat keras untuk proses uji coba perangkat lunak ini dapat dilihat pada Tabel 5.1.

**Tabel 5.1 Lingkungan Perangkat Keras**

Perangkat Keras	Tipe: Asus Padfone S T00N Prosesor: Quad-core 2.3 GHz Krait 400 Memori: 2048MB RAM
-----------------	--

##### **5.1.2 Lingkungan Perangkat Lunak**

Deskripsi perangkat lunak untuk proses uji coba dapat dilihat pada Tabel 5.2

**Tabel 5.2 Lingkungan Perangkat Lunak**

Perangkat Lunak	Sistem Operasi: Android OS, v4.4.2 (KitKat)
-----------------	---

## 5.2 Skenario Pengujian

Pada skenario pengujian, proses pengujian dilakukan menggunakan metode *black-box* berdasarkan skenario yang telah ditentukan terhadap fungsionalitas permainan dan pengujian dilakukan dengan survei langsung kepada pengguna.

### 5.2.1 Pengujian Fungsionalitas

Pada subbab ini akan dijelaskan mengenai detail dari skenario pengujian dan hasil yang didapatkan dari pengujian aplikasi permainan yang dibangun. Penjelasan disajikan dengan menampilkan tujuan uji coba, kondisi awal, penjelasan skenario, keluaran yang diharapkan, hasil pengujian, dan kondisi akhir.

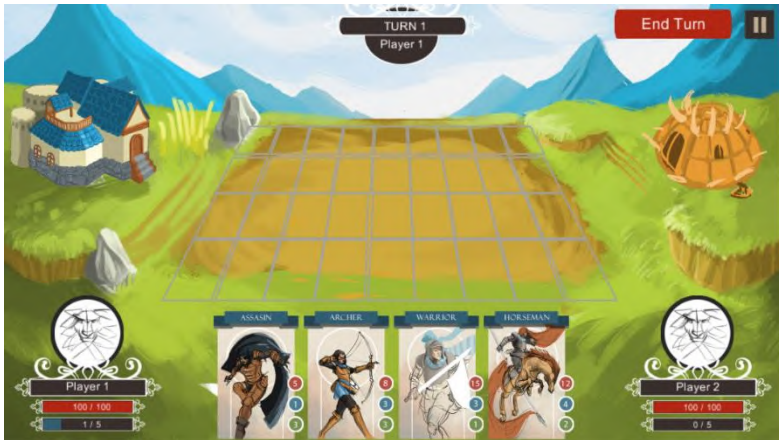
#### 5.2.1.1 Uji Coba Permainan Menang

Uji coba permainan menang berfungsi untuk mengetahui apakah aplikasi permainan dapat mencapai kondisi di mana pemain dinyatakan menang. Pemain dinyatakan menang apabila telah berhasil mengalahkan pemain lawan dengan cara menyerang *base* pemain lawan hingga *life point* pemain lawan  $\leq 0$ . *Life point* berkurang jika monster berada di *tile* yang dekat dengan *base* pemain lawan dan menyerang *base* tersebut. Hasil dari pengujian dapat dilihat pada Tabel 5.3. Hasil dari skenario uji coba dapat dilihat pada Gambar 5.1 dan Gambar 5.2.

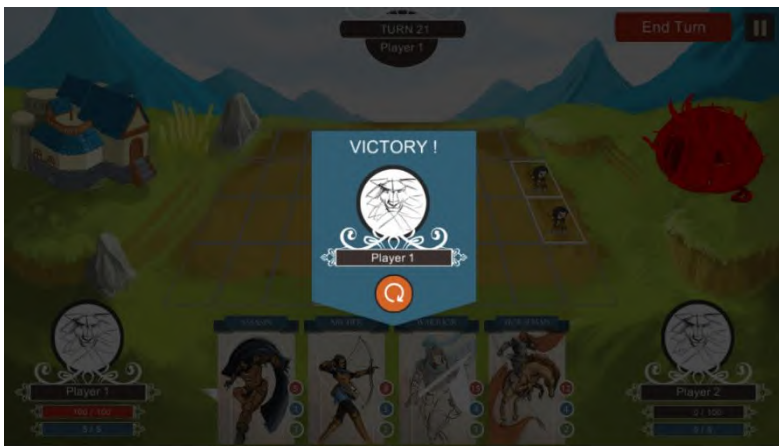
**Tabel 5.3 Hasil Uji Coba Permainan Menang**

ID	UJ-P-01
Nama	Uji Coba Permainan Menang
Tujuan Uji Coba	Pemain dapat memenangkan permainan
Kondisi awal	Pemain berada pada <i>stage</i> awal
Skenario 1	Pemain mengalahkan pemain lawan hingga <i>life point</i> pemain lawan $\leq 0$
Keluaran yang diharapkan	Aplikasi memunculkan panel hasil permainan yang berisi pesan kemenangan

Hasil uji coba	Berhasil
Kondisi akhir	Muncul panel hasil permainan yang berisi pesan kemenangan pada layar



**Gambar 5.1 Kondisi Awal Uji Coba Permainan Menang**



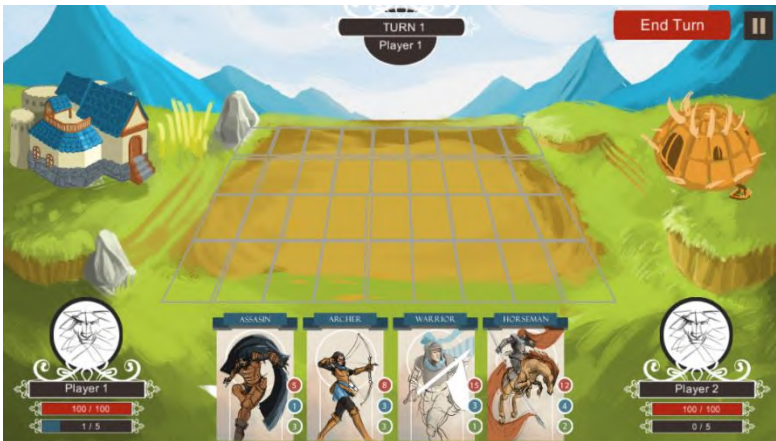
**Gambar 5.2 Kondisi Akhir Uji Coba Permainan Menang**

### 5.2.1.2 Uji Coba Permainan Kalah

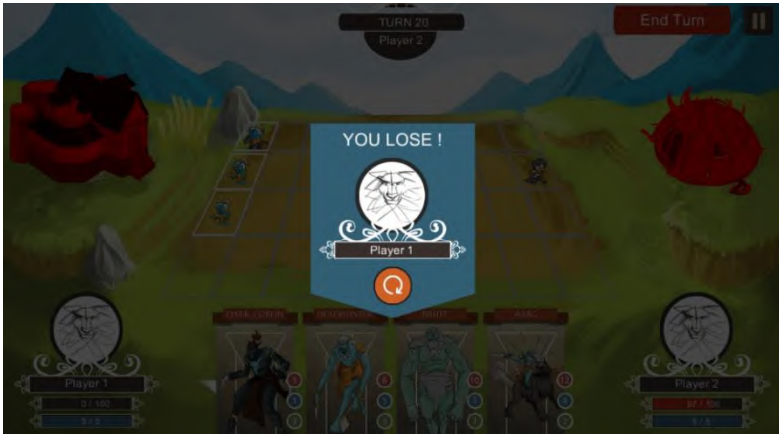
Uji coba permainan kalah berfungsi untuk mengetahui apakah aplikasi permainan dapat mencapai kondisi di mana pemain dinyatakan kalah. Pemain dinyatakan kalah apabila telah berhasil dikalahkan oleh pemain lawan *life point* pemain  $\leq 0$ . Hasil dari pengujian dapat dilihat pada Tabel 5.4. Hasil dari skenario uji coba dapat dilihat pada Gambar 5.3 dan Gambar 5.4.

**Tabel 5.4 Hasil Uji Coba Permainan Kalah**

ID	UJ-P-02
Nama	Uji Coba Permainan Kalah
Tujuan Uji Coba	Pemain dapat kalah saat bermain
Kondisi awal	Pemain berada pada <i>stage</i> awal
Skenario 1	Pemain membiarkan pemain lawan menyerang <i>base</i> pemain hingga <i>life point</i> pemain $\leq 0$
Keluaran yang diharapkan	Aplikasi memunculkan panel hasil permainan yang berisi pesan kekalahan
Hasil uji coba	Berhasil
Kondisi akhir	Muncul panel hasil permainan yang berisi pesan kekalahan pada layar



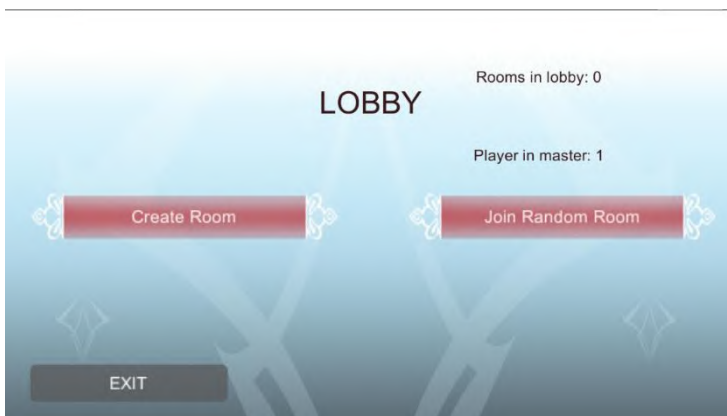
**Gambar 5.3 Kondisi Awal Uji Coba Permainan Kalah**



**Gambar 5.4 Kondisi Akhir Uji Coba Permainan Kalah**

### 5.2.1.3 Uji Coba *Create Room*

Uji coba *create room* berfungsi untuk mengetahui apakah pemain dapat berhasil untuk membuat *room*. Di *room* inilah nanti para pemain akan bertanding. Hasil dari pengujian dapat dilihat pada Tabel 5.5 Tabel 5.3. Hasil dari skenario uji coba dapat dilihat pada Gambar 5.5 dan Gambar 5.6.



**Gambar 5.5 Kondisi Awal Uji Coba *Create Room***



**Gambar 5.6 Kondisi Akhir Uji Coba *Create Room***

**Tabel 5.5 Hasil Uji Coba *Create Room***

ID	UJ-P-03
Nama	Uji Coba Permainan <i>Create Room</i>
Tujuan Uji Coba	Pemain dapat membuat <i>room</i>
Kondisi awal	Pemain berada pada <i>lobby</i>
Skenario 1	Pemain memilih tombol <i>create room</i>
Keluaran yang diharapkan	Pemain berhasil membuat <i>room</i> .
Hasil uji coba	Berhasil
Kondisi akhir	Pemain masuk ke dalam <i>waiting panel</i>

#### 5.2.1.4 Uji Coba *Join Random Room*

Uji coba *join random room* berfungsi untuk mengetahui apakah pemain dapat berhasil untuk *join room* secara *random*. Jadi pemain tidak perlu memilih ingin masuk ke dalam *room* mana. Pemain bisa memasuki suatu *room* jika pemain pada *room* tersebut kurang dari 2. Hasil dari pengujian dapat dilihat pada Tabel 5.6 Tabel 5.5 Tabel 5.3. Hasil dari skenario uji coba dapat dilihat pada Gambar 5.7 Gambar 5.3 dan Gambar 5.8.

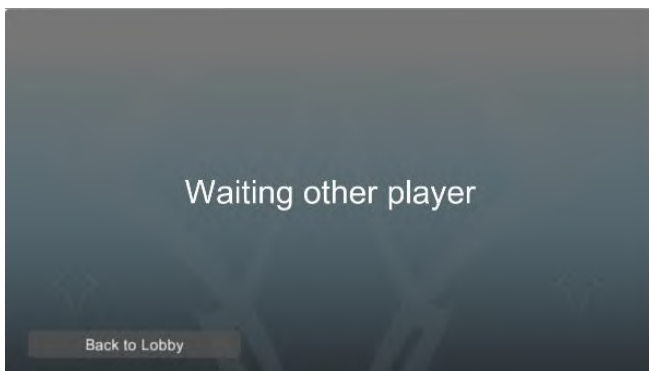




**Gambar 5.7 Kondisi Awal Pengujian *Join Random Room***

**Tabel 5.6 Hasil Uji Coba *Join Random Room***

ID	UJ-P-04
Nama	Uji Coba Permainan <i>Join Random Room</i>
Tujuan Uji Coba	Pemain dapat <i>join room</i> secara <i>random</i>
Kondisi awal	Pemain berada pada <i>lobby</i>
Skenario 1	Pemain memilih tombol <i>join random room</i>
Keluaran yang diharapkan	Pemain berhasil masuk ke dalam <i>room</i> .
Hasil uji coba	Berhasil
Kondisi akhir	Pemain masuk ke dalam <i>waiting panel</i>



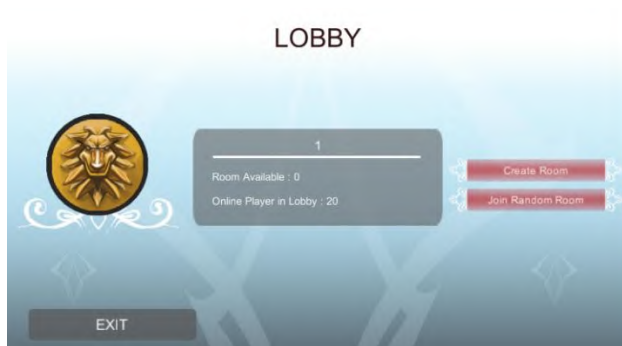
**Gambar 5.8 Kondisi Akhir Pengujian *Join Random Room***

### 5.2.1.5 Uji Coba Performa Photon

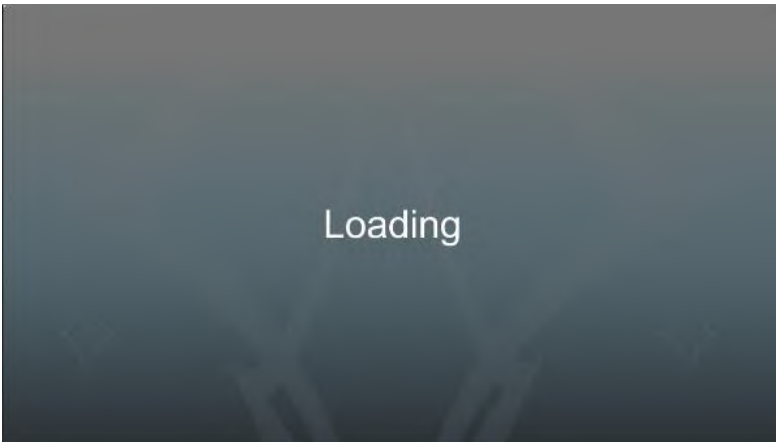
Uji coba performa Photon berfungsi untuk mengetahui berapa jumlah pemain yang bisa terkoneksi pada 1 aplikasi. Pada aplikasi ini menggunakan tipe Photon yang gratis yang hanya menampung 20 pemain. Hasil dari pengujian dapat dilihat pada Tabel 5.7. Uji coba skenario dilakukan dengan cara sejumlah 20 pemain masuk ke dalam aplikasi. Pemain ke-1 hingga ke-20 berhasil masuk, hal ini bisa dilihat pada Gambar 5.9 dan Gambar 5.10. Setelah itu, jika terdapat pemain yang ingin masuk ke dalam aplikasi, maka akan muncul tampilan seperti pada Gambar 5.11 dan *log error* seperti pada Gambar 5.12



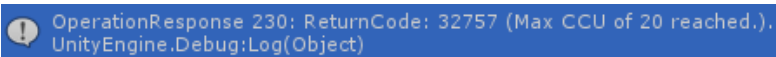
**Gambar 5.9 Tampilan Saat Pemain ke-1**



**Gambar 5.10 Tampilan Saat Pemain ke-20**



**Gambar 5.11 Tampilan Pemain Ke-21**



**Gambar 5.12 Tampilan Log Error Pemain Ke-21**

**Tabel 5.7 Hasil Uji Coba Performa Photon**

ID	UJ-P-05
Nama	Uji Coba Performa Photon
Tujuan Uji Coba	Mengetahui berapa jumlah maksimal pemain yang dapat ditampung oleh Photon pada 1 aplikasi.
Kondisi awal	Pemain masuk aplikasi.
Skenario 1	Pemain memilih tombol <i>play</i> pada halaman antarmuka awal
Keluaran yang diharapkan	Terdapat 20 pemain yang berhasil masuk ke dalam aplikasi.
Hasil uji coba	Berhasil
Kondisi akhir	Jumlah pemain yang berhasil masuk sejumlah 20 pemain.

### 5.2.2 Pengujian Terhadap Pengguna

Permainan ini perlu diuji oleh pemain secara langsung. Hal ini bertujuan untuk mengetahui tingkat kenyamanan, kemudahan, dan bagaimana tanggapan pemain mengenai *gameplay*. Pengujian ini mengacu pada metode *blackbox* yang telah dijabarkan dalam menguji fungsionalitas aplikasi.

Pengujian dilakukan oleh 5 mahasiswa Teknik Informatika dan 5 orang umum / non - Teknik Informatika. Kemudian pemain akan diminta untuk mencoba menjalankan permainan terlebih dahulu dan kemudian pemain diminta untuk mengisi kuesioner yang telah disediakan. Kuesioner dapat dilihat pada Tabel 5.8. Setelah mengisi kuesioner, pengguna diminta untuk memberikan kritik dan saran terhadap permainan ini yang diharapkan dapat bermanfaat untuk pengembangan berikutnya. Bobot penilaian pada kuesioner memiliki ketentuan sebagai berikut:

- Sangat Setuju (SS) = 5
- Setuju (S) = 4
- Cukup (C) = 3
- Tidak Setuju (TS) = 2
- Sangat Tidak Setuju (STS) = 1

**Tabel 5.8 Kuesioner Pengguna**

No.	Parameter	Penilaian				
		STS	TS	C	S	SS
	<b>Parameter Antarmuka</b>					
1	Aplikasi memiliki tampilan dan desain yang menarik					
2	Aplikasi memiliki menu yang mudah digunakan					
3	Aplikasi memiliki tata letak tombol yang sesuai					

	<b>Parameter Performa dan Kenyamanan</b>					
4	Aplikasi nyaman untuk dimainkan					
5	Aplikasi menampilkan perpindahan menu dengan lancar					
6	Aplikasi berjalan dengan lancar					
	<b>Parameter Gameplay</b>					
7	<i>Gameplay</i> aplikasi jarang ditemui di aplikasi lain					
8	<i>Gameplay</i> aplikasi membuat pemain betah bermain					

### 5.3 Evaluasi Pengujian

Subbab ini membahas mengenai evaluasi terhadap pengujian fungsionalitas yang telah dilakukan dan evaluasi terhadap pengguna.

#### 5.3.1 Evaluasi Pengujian Fungsionalitas

Evaluasi pengujian fungsionalitas dilakukan dengan menampilkan data rekapitulasi aplikasi permainan yang telah dipaparkan pada subbab 5.2. Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Tabel 5.9. Dari data yang terdapat pada tabel tersebut, diketahui bahwa aplikasi permainan telah berhasil melalui pengujian fungsionalitas.

**Tabel 5.9 Hasil Pengujian Fungsionalitas**

No.	ID	Nama	Hasil
1	UJ-P-01	Uji Coba Permainan Menang	Berhasil

2	UJ-P-02	Uji Coba Permainan Kalah	Berhasil
3	UJ-P-03	Uji Coba <i>Create Room</i>	Berhasil
4	UJ-P-04	Uji Coba <i>Join Random Room</i>	Berhasil
5	UJ-P-05	Uji Coba Performa Photon	Berhasil

### 5.3.1.1 Evaluasi Pengujian Terhadap Pengguna

Pengisian kuesioner dilakukan oleh 5 orang Mahasiswa Teknik Informatika ITS dan 5 orang umum / non-Teknik Informatika. Rentang umur pengguna berkisar antara 20 sampai 22 tahun. Pengguna terdiri dari wanita dan pria. Hasil kuesioner dari setiap kriteria penilaian akan diambil modulusnya. Jawaban kuesioner dapat dilihat pada lembar Lampiran. Sementara hasil kuesioner dapat dilihat pada Tabel 5.10.

**Tabel 5.10 Hasil Kuesioner**

No.	Parameter	Penilaian				
		STS	TS	C	S	SS
	<b>Parameter Antarmuka</b>					
1	Aplikasi memiliki tampilan dan desain yang menarik	-	-	-	3	<b>7</b>
2	Aplikasi memiliki menu yang mudah digunakan	-	-	1	<b>6</b>	3
3	Aplikasi memiliki tata letak tombol yang sesuai	-	-	1	4	<b>5</b>
	<b>Parameter Performa dan Kenyamanan</b>					
4	Aplikasi nyaman untuk dimainkan	-	-	1	<b>5</b>	4

5	Aplikasi menampilkan perpindahan menu dengan lancar	-	-	2	7	1
6	Aplikasi berjalan dengan lancar	-	-	2	6	2
<b>Parameter Gameplay</b>						
7	<i>Gameplay</i> aplikasi jarang ditemui di aplikasi lain	-	-	1	8	1
8	<i>Gameplay</i> aplikasi membuat pemain betah bermain	-	-	3	4	3

### 5.3.1.2 Evaluasi Antarmuka

Untuk parameter antarmuka, terdapat 3 aspek penilaian. Aspek pertama adalah tampilan dan desain. Dari aspek ini, mayoritas pengguna sangat setuju bahwa permainan ini memiliki tampilan dan desain yang menarik dengan nilai 7 dari 10 pengguna. Aspek selanjutnya adalah kemudahan menu, dimana 6 dari 10 pengguna setuju bahwa permainan memiliki menu yang mudah digunakan. Untuk aspek terakhir yaitu tata letak tombol. Sebanyak 5 dari 10 pengguna sangat setuju bahwa permainan memiliki tata letak tombol yang sesuai. Rincian hasil kuesioner bisa dilihat pada Tabel 5.10.

### 5.3.1.3 Evaluasi Pengujian Terhadap Pengguna

Untuk parameter performa dan kenyamanan terdapat 3 aspek penilaian. Aspek pertama adalah kenyamanan. Dari aspek ini, mayoritas pengguna setuju bahwa permainan ini nyaman dimainkan dengan nilai 5 dari 10 pengguna. Aspek selanjutnya adalah perpindahan menu, dimana 7 dari 10 pengguna setuju bahwa permainan menampilkan perpindahan menu dengan lancar. Pengguna tersebut merasa terganggu dengan waktu yang cukup Untuk aspek terakhir yaitu kelancaran. Sebanyak 6 dari 10 pengguna setuju bahwa permainan mampu berjalan dengan lancar. Rincian hasil kuesioner bisa dilihat pada Tabel 5.7

### 5.3.1.4 Evaluasi Gameplay

Untuk parameter *gameplay*, terdapat 2 aspek penilaian yang dipengaruhi oleh apakah pengguna tersebut mahasiswa Teknik Informatika atau bukan. Jika pengguna tersebut merupakan mahasiswa Teknik Informatika, diperkirakan telah bermain beberapa jenis *game* yang memiliki *gameplay* yang berbeda-beda. Sehingga bisa menilai apakah permainan ini memiliki *gameplay* yang unik / jarang ditemui atau tidak. Hasil kuesioner untuk evaluasi *gameplay* bagi pengguna mahasiswa Teknik Informatika dapat dilihat pada Tabel 5.11, sedangkan hasil kuesioner untuk evaluasi *gameplay* bagi pengguna non – mahasiswa Teknik Informatika dapat dilihat pada Tabel 5.12

**Tabel 5.11 Hasil Kuesioner untuk Pengguna Mahasiswa Teknik Informatika**

No.	Parameter	Penilaian				
		STS	TS	C	S	SS
1	<i>Gameplay</i> aplikasi jarang ditemui di aplikasi lain	-	-	-	5	-
2	<i>Gameplay</i> aplikasi membuat pemain betah bermain	-	-	1	2	2

**Tabel 5.12 Hasil Kuesioner untuk Pengguna non-Mahasiswa Teknik Informatika**

No.	Parameter	Penilaian				
		STS	TS	C	S	SS
1	<i>Gameplay</i> aplikasi jarang ditemui di aplikasi lain	-	-	1	3	1
2	<i>Gameplay</i> aplikasi membuat pemain betah bermain	-	-	2	2	1

Dari hasil yang ditunjukkan pada Tabel 5.11 dan Tabel 5.12 terdapat persamaan dan perbedaan. Untuk aspek keunikan *gameplay* dari aplikasi, mayoritas kedua jenis pengguna setuju




bahwa aplikasi memiliki *gameplay* yang jarang ditemui di aplikasi lain.


Untuk aspek berikutnya, apakah *gameplay* aplikasi membuat pemain betah bermain, terdapat perbedaan antara kedua jenis pengguna tersebut. Untuk pengguna yang merupakan mahasiswa Teknik Informatika, mayoritas dari mereka setuju bahwa *gameplay* tersebut membuat pemain betah bermain. Sedangkan untuk jenis pengguna non mahasiswa Teknik Informatika, mereka cukup setuju bahwa *gameplay* tersebut membuat pemain betah bermain.

*(Halaman ini sengaja dikosongkan)*

## LAMPIRAN



**ITS**  
Institut Teknologi  
Sepuluh Nopember



**KUESIONER TUGAS AKHIR – 5112100157 IKROM AULIA FAHDI**

---

**ASYNCHRONOUS MULTIPLAYER CARDGAME PADA GAME MONSTER KING**

---

**Identitas Responden**

Nama Lengkap : Angga Saputra Dwi      Jenis Kelamin DP  
 Usia : 22 Tahun

**A. KARAKTERISTIK RESPONDEN**

- Apakah Anda mahasiswa teknik informatika?  
 a. Ya      b. Tidak
- Apakah anda pernah bermain *card game*?  
 a. Ya      b. Tidak

**B. PENILAIAN TERHADAP APLIKASI**  
*Isilah tabel dibawah ini dengan menggunakan tanda (v)*  
 SS = Sangat Setuju      S = Setuju      C = Cukup  
 TS = Tidak Setuju      STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik					<input checked="" type="checkbox"/>
2	Aplikasi memiliki menu yang mudah digunakan					<input checked="" type="checkbox"/>
3	Aplikasi memiliki tata letak tombol yang sesuai					<input checked="" type="checkbox"/>
Parameter Performa dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan				<input checked="" type="checkbox"/>	
5	Aplikasi menampilkan perpindahan menu dengan lancar					<input checked="" type="checkbox"/>
6	Aplikasi berjalan dengan lancar				<input checked="" type="checkbox"/>	
Parameter Gameplay						
7	Gameplay aplikasi jarang ditemui di aplikasi lain				<input checked="" type="checkbox"/>	
8	Gameplay aplikasi membuat pemain betah bermain				<input checked="" type="checkbox"/>	

**C. KRITIK DAN SARAN**

Bidang permainan Terlalu kecil

.....

.....

Surabaya, 26 Juli ..... 2016

[Signature]

Gambar A.1 Kuesioner Responden Pertama



KUESIONER TUGAS AKHIR – 5112100157 IKROM AULIA FAHDH

ASYNCHRONOUS MULTIPLAYER CARDGAME PADA GAME MONSTER KING



Identitas Responden

Nama Lengkap : Primo Sigit Rusandyana Jenis Kelamin : (L) P  
 Usia : 22 Tahun

A. KARAKTERISTIK RESPONDEN

1. Apakah Anda mahasiswa teknik informatika?  
 a. Ya                      b. Tidak
2. Apakah anda pernah bermain *card game*?  
 a. Ya                      b. Tidak

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (√)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik					<input checked="" type="checkbox"/>
2	Aplikasi memiliki menu yang mudah digunakan					<input checked="" type="checkbox"/>
3	Aplikasi memiliki tata letak tombol yang sesuai					<input checked="" type="checkbox"/>
<b>Parameter Performa dan Kenyamanan</b>						
4	Aplikasi nyaman untuk dimainkan					<input checked="" type="checkbox"/>
5	Aplikasi menampilkan perpindahan menu dengan lancar					<input checked="" type="checkbox"/>
6	Aplikasi berjalan dengan lancar					<input checked="" type="checkbox"/>
<b>Parameter Gameplay</b>						
7	Gameplay aplikasi jarang ditemui di aplikasi lain					<input checked="" type="checkbox"/>
8	Gameplay aplikasi membuat pemain betah bermain					<input checked="" type="checkbox"/>

C. KRITIK DAN SARAN

.....  
 Perhatikan simetris desain.....  
 .....

Surabaya, 26 Juli ..... 2016

Primo  
 \_\_\_\_\_

Gambar A.2 Kuesioner Responden Kedua



KUESIONER TUGAS AKHIR – 5112100157 IKROM AULIA FAHDU

ASYNCHRONOUS MULTIPLAYER CARDGAME PADA GAME MONSTER KING



Identitas Responden

Nama Lengkap : Luthi F Soehadak

Jenis Kelamin  L  P

Usia : 22 Tahun

A. KARAKTERISTIK RESPONDEN

- Apakah Anda mahasiswa teknik informatika?
  - Ya
  - Tidak
- Apakah anda pernah bermain *card game*?
  - Ya
  - Tidak

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju


No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik					✓
2	Aplikasi memiliki menu yang mudah digunakan					✓
3	Aplikasi memiliki tata letak tombol yang sesuai					✓
<b>Parameter Performa dan Kenyamanan</b>						
4	Aplikasi nyaman untuk dimainkan					✓
5	Aplikasi menampilkan perpindahan menu dengan lancar					✓
6	Aplikasi berjalan dengan lancar					✓
<b>Parameter Gameplay</b>						
7	Gameplay aplikasi jarang ditemui di aplikasi lain					✓
8	Gameplay aplikasi membuat pemain betah bermain					✓

C. KRITIK DAN SARAN


Pertambah animasi agar lebih menarik

Surabaya, 26 Juli 2016

Gambar A.3 Kuesioner Responden Ketiga



**ITS**  
Institut Teknologi Sepuluh Nopember  
Republik Indonesia



**KUESIONER TUGAS AKHIR – 5112100157 IKROM AULIA FAHDI**

---

**ASYNCHRONOUS MULTIPLAYER CARDGAME PADA GAME MONSTER KING**

---

**Identitas Responden**

Nama Lengkap : Pinasitika Cintya      Jenis Kelamin : L/B  
 Usia : 21 Tahun

**A. KARAKTERISTIK RESPONDEN**

- Apakah Anda mahasiswa teknik informatika?
  - Ya
  - Tidak
- Apakah anda pernah bermain *card game*?
  - Ya
  - Tidak

**B. PENILAIAN TERHADAP APLIKASI**

Isilah tabel dibawah ini dengan menggunakan tanda (√)

SS = Sangat Setuju      S = Setuju      C = Cukup  
 TS = Tidak Setuju      STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik					√
2	Aplikasi memiliki menu yang mudah digunakan				√	
3	Aplikasi memiliki tata letak tombol yang sesuai				√	
<b>Parameter Performa dan Kenyamanan</b>						
4	Aplikasi nyaman untuk dimainkan					√
5	Aplikasi menampilkan perpindahan menu dengan lancar				√	
6	Aplikasi berjalan dengan lancar				√	
<b>Parameter Gameplay</b>						
7	Gameplay aplikasi jarang ditemui di aplikasi lain				√	
8	Gameplay aplikasi membuat pemain betah bermain					√

**C. KRITIK DAN SARAN**

.....  
 baru nya kekecilan , apa hp nya yg kekecilan ?  
 .....  
 .....

Surabaya, 26 Juli ..... 2016

*[Signature]*  
 Pinasitika Cintya A.

**Gambar A.4 Kuesioner Responden Keempat**





KUESIONER TUGAS AKHIR – 5112100157 IKROM AULIA FAHDI

ASYNCHRONOUS MULTIPLAYER CARDGAME PADA GAME MONSTER KING

Identitas Responden

Nama Lengkap : Lita Amalia ..... Jenis Kelamin : L (P)  
 Usia : 21 Tahun

A. KARAKTERISTIK RESPONDEN

1. Apakah Anda mahasiswa teknik informatika?
  - a. Ya
  - b. Tidak
2. Apakah anda pernah bermain *card game*?
  - a. Ya
  - b. Tidak

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>
2	Aplikasi memiliki menu yang mudah digunakan	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>
3	Aplikasi memiliki tata letak tombol yang sesuai					<input checked="" type="checkbox"/>
<b>Parameter Performa dan Kenyamanan</b>						
4	Aplikasi nyaman untuk dimainkan					<input checked="" type="checkbox"/>
5	Aplikasi menampilkan perpindahan menu dengan lancar					<input checked="" type="checkbox"/>
6	Aplikasi berjalan dengan lancar					<input checked="" type="checkbox"/>
<b>Parameter Gameplay</b>						
7	Gameplay aplikasi jarang ditemui di aplikasi lain					<input checked="" type="checkbox"/>
8	Gameplay aplikasi membuat pemain betah bermain					<input checked="" type="checkbox"/>

C. KRITIK DAN SARAN

.....  
 .....  
 .....  
 .....

Surabaya, 27 Juli ..... 2016

Lita Amalia

Gambar A.6 Kuesioner Responden Keenam





KUESIONER TUGAS AKHIR – 5112100157 IKROM AULIA FAHDU

ASYNCHRONOUS MULTIPLAYER CARDGAME PADA GAME MONSTER KING

Identitas Responden

Nama Lengkap : Hamif Ramadhan Jenis Kelamin : 0 / P  
 Usia : 20 Tahun

A. KARAKTERISTIK RESPONDEN

1. Apakah Anda mahasiswa teknik informatika?
  - a. Ya
  - b. Tidak
2. Apakah anda pernah bermain *card game*?
  - a. Ya
  - b. Tidak

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik					✓
2	Aplikasi memiliki menu yang mudah digunakan					✓
3	Aplikasi memiliki tata letak tombol yang sesuai					✓
<b>Parameter Performa dan Kenyamanan</b>						
4	Aplikasi nyaman untuk dimainkan					✓
5	Aplikasi menampilkan perpindahan menu dengan lancar					✓
6	Aplikasi berjalan dengan lancar					✓
<b>Parameter Gameplay</b>						
7	Gameplay aplikasi jarang ditemui di aplikasi lain					✓
8	Gameplay aplikasi membuat pemain betah bermain					✓

C. KRITIK DAN SARAN

Kapasitas player ditambah.

.....

.....

.....

Surabaya, 27 Juli ..... 2016

Hamif R.

Gambar A.7 Kuesioner Responden Ketujuh



KUESIONER TUGAS AKHIR – 5112100157 IKROM AULIA FAHDI

ASYNCHRONOUS MULTIPLAYER CARDGAME PADA GAME MONSTER KING

Identitas Responden

Nama Lengkap : Hafidz Ramadhan Jenis Kelamin : L/P  
Usia : 20 Tahun

A. KARAKTERISTIK RESPONDEN

1. Apakah Anda mahasiswa teknik informatika?
  - a. Ya
  - b. Tidak
2. Apakah anda pernah bermain *card game*?
  - a. Ya
  - b. Tidak

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan			✓		
3	Aplikasi memiliki tata letak tombol yang sesuai			✓		
<b>Parameter Performa dan Kenyamanan</b>						
4	Aplikasi nyaman untuk dimainkan				✓	
5	Aplikasi menampilkan perpindahan menu dengan lancar			✓		
6	Aplikasi berjalan dengan lancar			✓		
<b>Parameter Gameplay</b>						
7	Gameplay aplikasi jarang ditemui di aplikasi lain			✓		
8	Gameplay aplikasi membuat pemain betah bermain			✓		

C. KRITIK DAN SARAN

.....

.....

.....

.....

Surabaya, 26 Juli 2016

Hafidz Ramadhan

Gambar A.8 Kuesioner Responden Kedelapan



KUESIONER TUGAS AKHIR – 5112100157 IKROM AULIA FAHDI

ASYNCHRONOUS MULTIPLAYER CARDGAME PADA GAME MONSTER KING

Identitas Responden

Nama Lengkap : Amalia Zapitri Jenis Kelamin : L (P)  
 Usia : 22 Tahun

A. KARAKTERISTIK RESPONDEN

1. Apakah Anda mahasiswa teknik informatika?
  - a. Ya
  - b. Tidak
2. Apakah anda pernah bermain *card game*?
  - a. Ya
  - b. Tidak

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik					✓
2	Aplikasi memiliki menu yang mudah digunakan					✓
3	Aplikasi memiliki tata letak tombol yang sesuai					✓
<b>Parameter Performa dan Kenyamanan</b>						
4	Aplikasi nyaman untuk dimainkan				✓	
5	Aplikasi menampilkan perpindahan menu dengan lancar				✓	
6	Aplikasi berjalan dengan lancar				✓	
<b>Parameter Gameplay</b>						
7	Gameplay aplikasi jarang ditemui di aplikasi lain					✓
8	Gameplay aplikasi membuat pemain betah bermain					✓

C. KRITIK DAN SARAN

.....

.....

.....

.....

Surabaya, 27 Juli ..... 2016

Amalia Zapitri

Gambar A.9 Kuesioner Responden Kesembilan



KUESIONER TUGAS AKHIR – 5112100157 IKROM AULIA FAHDI

ASYNCHRONOUS MULTIPLAYER CARDGAME PADA GAME MONSTER KING

Identitas Responden

Nama Lengkap : Puspita Hidayati  
Usia : 22 Tahun

Jenis Kelamin : L (P)

A. KARAKTERISTIK RESPONDEN

1. Apakah Anda mahasiswa teknik informatika?
  - a. Ya
  - b. Tidak
2. Apakah anda pernah bermain *card game*?
  - a. Ya
  - b. Tidak

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai				✓	
<b>Parameter Performa dan Kenyamanan</b>						
4	Aplikasi nyaman untuk dimainkan				✓	
5	Aplikasi menampilkan perpindahan menu dengan lancar			✓		
6	Aplikasi berjalan dengan lancar				✓	
<b>Parameter Gameplay</b>						
7	Gameplay aplikasi jarang ditemui di aplikasi lain				✓	
8	Gameplay aplikasi membuat pemain betah bermain			✓		

C. KRITIK DAN SARAN

.....

.....

.....

.....

Surabaya, 27 Juli 2016

*Puspita*

PUSPITA H.

Gambar A.10 Kuesioner Responden Kesepuluh

## **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.

#### **6.1. Kesimpulan**

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

1. Aplikasi permainan berhasil menjalankan kebutuhan fungsional skenario aturan main menggunakan konsep *turn based* dalam Unity.
2. Mekanisme *asynchronous* berhasil diimplementasikan menggunakan Photon Realtime.
3. Berdasarkan pengujian fitur *matchmaking*, aplikasi permainan berhasil dalam mencari dan mempertemukan antar pemain.

#### **6.2. 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. Pengembangan kedepannya sebaiknya dapat terintegrasi dengan fitur *Virtual Reality* atau *Augmented Reality*, sehingga permainan terasa nyata dan seru.
2. Pengembangan kedepannya aplikasi permainan dapat dimainkan banyak pengguna dan dijadikan produk komersial.
3. Pengembangan kedepannya juga sebaiknya diberi unsur edukasi, sehingga pemain juga mendapatkan ilmu / manfaat dari aplikasi tersebut.

## DAFTAR PUSTAKA

- [1] D. Parlett, "The Oxford guide to card games: a historical survey," Oxford University Press, 1990.
- [2] "Global Cross Platform Realtime Multiplayer Game Framework | Photon: Multiplayer Made Simple," Exit Games, [Online]. Available: <https://www.photonengine.com/en/Realtime>. [Accessed 13 Desember 2015].
- [3] "Unity," Unity, [Online]. Available: <https://unity3d.com/unity>. [Accessed 8 Desember 2015].
- [4] "Download Android Studio and SDK Tools | Android Developers," Google, [Online]. Available: <http://developer.android.com/sdk/index.html>. [Accessed 13 Desember 2015].
- [5] "Turnbased & Asynchronous Games | Exit Games," Exit Games, [Online]. Available: <http://doc.photonengine.com/en-us/realtime/current/tutorials/async-and-turnbased>. [Accessed 3 Januari 2016].
- [6] "Tour | Photon: Multiplayer Made Simple," Exit Games, [Online]. Available: <https://www.photonengine.com/en-US/Realtime/Tour>. [Accessed 10 Desember 2015].

## **BIODATA PENULIS**



Ikrom Aulia Fahdi merupakan anak ke-1 dari 3 bersaudara ini lahir di Probolinggo 18 April 1994. Penulis telah menempuh pendidikan formal yaitu TK Tunas Harapan Probolinggo (1999 – 2001), SDN Sukabumi 2 Probolinggo (2001 – 2007), SMPN 1 Probolinggo (2007 – 2009) SMAN 1 Probolinggo (2009 – 2012) dan mahasiswa S1 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya rumpun mata kuliah Interaksi, Grafika, dan Seni (2012 – 2016).

Selama menjadi mahasiswa, penulis pernah juga pernah mengikuti organisasi yang ada di jurusan maupun fakultas dan UKM yang ada di ITS antara lain Himpunan Mahasiswa Teknik Computer-Informatika (HMTC), Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi (BEM FTIf) dan Unit Kegiatan Mahasiswa (UKM) Musik ITS. Ditahun ke-2 penulis menjadi staff Kewirausahaan dan Bakat (KMB) HMTC dan staff eksternal UKM Musik. Sedangkan ditahun ke-3 penulis mendapat amanah sebagai Kepala Department Internal Affair (IA) BEM FTIf. Penulis juga pernah menjadi koordinator admin laboratorium IGS dan koordinator PIKTI Teknik Informatika ITS. Penulis juga pernah menjuarai lomba Devcomp (juara favorit) yang diadakan oleh BEM FTIf – ITS. Penulis dapat dihubungi via email [ichrome.fahdi@gmail.com](mailto:ichrome.fahdi@gmail.com).