



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - IF184802

PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCULUS RIFT DAN LEAP MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

DIAS ADHI PRATAMA
NRP 05111540000173

Dosen Pembimbing
Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.
Ridho Rahman Hariadi, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - IF184802

PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCULUS RIFT DAN LEAP MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

DIAS ADHI PRATAMA
NRP 05111540000173

Dosen Pembimbing
Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.
Ridho Rahman Hariadi, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - IF184802

VIRTUAL REALITY GAME GRAND WIZARD USING OCULUS RIFT AND LEAP MOTION WITH POINT CLOUD RECOGNITION CONTROL

**DIAS ADHI PRATAMA
NRP 05111540000173**

**Advisor
Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.
Ridho Rahman Hariadi, S.Kom., M.Sc.**

**DEPARTMENT of INFORMATICS ENGINEERING
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCULUS RIFT DAN LEAP MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Interaksi, Grafika dan Seni
Program Studi S-1 Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh:

DIAS ADHI PRATAMA

NRP: 05111540000173

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Eng. Darlis Herumurti, S.Kom., M.Kom

NIP: 19771217 200312 1 001

(pembimbing 1)

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

NIP: 19870213 201404 1 001

(pembimbing 2)

**SURABAYA
JANUARI 2020**

[Halaman ini sengaja dikosongkan]

PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCULUS RIFT DAN LEAP MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

Nama Mahasiswa : Dias Adhi Pratama
NRP : 05111540000173
Departemen : Teknik Informatika FTEIC-ITS
Dosen Pembimbing 1 : Dr. Eng. Darlis Herumurti, S.Kom.,
M.Kom.
Dosen Pembimbing 2 : Ridho Rahman Hariadi, S.Kom., M.Sc.

ABSTRAK

Perkembangan teknologi dengan menggunakan realitas virtual sudah semakin maju. Tidak hanya untuk membantu manusia dalam mensimulasikan aktifitas di dunia nyata, juga menjadi sarana untuk bermain permainan yang terasa nyata. Untuk merasakan teknologi realitas virtual ada banyak macam alat yang dapat digunakan, salah satunya adalah Oculus Rift. Alat ini berfungsi untuk menampilkan dunia maya kepada pengguna.

Ide yang digunakan dalam tugas akhir ini adalah membangun sebuah permainan First Person Shooter (FPS) yang memiliki control Leap Motion dan Point Cloud Recognition. Permainan ini menjadikan tangan dan Gesture/gerakan sebagai sumber masukan dalam melakukan interaksi. Permainan ini dibangun menggunakan Unity dan Leap Motion SDK. Tujuan dari permainan ini adalah untuk meningkatkan adrenalin pemain untuk mengalahkan musuh yang ada sebelum musuh mengalahkannya lebih dulu.

Hasil dari pengujian ini akan berupa sebuah permainan yang dapat berjalan di perangkat personal computer high-end. Permainan ini dibangun dengan Unity Versi 2018.2.18f1 dengan bahasa pemrograman C#, Library Point Cloud Recognition, dan

Leap Motion SDK. Proses pembuatan asset permainan sebagian besar mengambil dari asset store dan sebagian dibuat menggunakan tools Blender, dan Corel Draw X7. Dengan pengujian beta dan juga kuisioner dapat disimpulkan permainan telah mengimplementasikan perancangan dengan baik.

Kata kunci: Oculus Rift, Realitas Virtual, Point-Cloud Recognition.

VIRTUAL REALITY GAME GRAND WIZARD USING OCULUS RIFT AND LEAP MOTION WITH POINT CLOUD RECOGNITION CONTROL

Name : Dias Adhi Pratama
NRP : 0511154000173
Department : Informatics Engineering ELECTICS-ITS
Supervisor I : Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.
Supervisor II : Ridho Rahman Hariadi, S.Kom., M.Sc.

ABSTRACT

The development of technology by using virtual reality is increasingly advanced. Not only to help humans in simulating activities in the real world, it also becomes a means to play a game that feels real. To experience virtual reality technology there are many different tools that can be used, one of them is Oculus Rift. This tool serves to show virtual world to users.

The idea used in this final project is to build a FirstPerson Shooter (FPS) game that has Leap Motion and Point Cloud Recognition control. This game makes hands and Gesture as a source of input in the interaction. The game is built using Unity and Leap Motion SDK. The purpose of the game is to increase the adrenaline of players to defeat the enemies before the enemy defeats them first.

The result of this testing will be a game that can run on highend personal computer devices. The game is built with Unity Version 2018.2.18f1 with c# programming language, Point Cloud Recognition Library, and Leap Motion SDK. The process of making asset games mostly take from the asset store and partly made using tools Blender, and Corel Draw X7. With beta testing and also the questionnaire can be concluded the game has implemented the design properly.

Keywords: Oculus Rift, Realitas Virtual, Point-Cloud Recognition.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah Subhanahu Wata'ala karena atas karunia, nikmat dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCULUS RIFT DAN LEAP MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

Sholawat serta salam semoga tak lupa kita sampaikan kepada Rasulullah Shallallahu 'Alaihi Wassalam, yang telah membawa risalah Islam dan Qur'an kepada seluruh umat manusia.

Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Orang tua dan keluarga yang senantiasa mendoakan, memotivasi dan mendukung lahir maupun batin penulis dalam menyelesaikan tanggung jawab ini.
2. Bapak Dr. Eng. Darlis Herumurti, S.Kom., M.Kom. dan Bapak Ridho Rahman Hariadi, S.Kom., M.Sc. selaku dosen pembimbing I dan II penulis, yang senantiasa membimbing, memberikan saran, arahan, serta bantuan-bantuan lainnya dalam menyelesaikan tugas akhir ini.
3. Administrator Lab IGS yang setia berada di laboratorium IGS, membantu penulis selama perkuliahan dan pengerjaan tugas akhir ini.
4. User TA IGS lainnya yang senantiasa membantu, mendukung, menghibur, dan menemani keseharian penulis di dalam laboratorium IGS.
5. Firman Maulana selaku adik tingkat saya yang sudah berkenan meminjamkan dan membantu dalam memenuhi aset-aset untuk pembuatan tugas akhir ini.
6. Teman-teman TC 2015 yang banyak membantu penulis dalam segala hal, yang berjuang bersama-sama penulis

dari awal masuk perkuliahan hingga akhir masa perkuliahan ini.

7. Teman-teman penghuni Roemah Perdjoengan yang senantiasa menemani ketika berada di kosan dan selalu kompak.
8. Teman-teman Pecinta Damai yang selalu menemani dan menghibur penulis serta bersama-sama berjuang.
9. Serta pihak-pihak lain yang mohon maaf tidak dapat disebutkan satu per satu, yang telah turut andil dalam membantu penulis selama perkuliahan.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyelesaikan tugas akhir ini. Namun, penulis mohon maaf apabila terdapat kekurangan ataupun kesalahan yang penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan untuk ke depannya.

Surabaya, Januari 2020

Dias Adhi Pratama

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
1 BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Permasalahan.....	1
1.3. Batasan Permasalahan	2
1.4. Tujuan.....	2
1.5. Manfaat.....	2
1.6. Metodologi	3
1.7. Sistematika Penulisan.....	5
2 BAB II TINJAUAN PUSTAKA	7
2.1. Realitas Virtual.....	7
2.2. Unity 3D	8
2.3. Bahasa Pemrograman C#	9
2.4. First Person Shooter	9
2.5. \$P Point Cloud Recognizer	10
2.6. Oculus Rift	10
2.7. Leap Motion Controller.....	10
2.8. Library Leap Motion	11
3 BAB III ANALISIS DAN PERANCANGAN SISTEM.....	13
3.1. Analisis Sistem	13
3.1.1. Spesifikasi Kebutuhan Sistem	13
3.1.2. Identifikasi Pengguna	15
3.2. Perancangan Sistem.....	15
3.2.1. Deskripsi Umum Sistem.....	15
3.2.2. Arsitektur Sistem	16

3.2.3.	Perancangan Tampilan Antarmuka	17
3.2.4.	User Interface	17
3.2.5.	Aset Permainan.....	19
3.3.	Perancangan Skenario Simulasi	21
3.3.1.	Alur Simulasi.....	21
3.3.2.	Aturan Main	23
3.4.	Perancangan Tampilan Antarmuka	23
3.4.1.	Tampilan Menu Permainan	23
3.4.2.	Tampilan Halaman How to Play	24
3.4.3.	Tampilan Permainan.....	25
3.4.4.	Tampilan Pause Menu	28
3.4.5.	Tampilan Game Over	29
4 BAB IV	IMPLEMENTASI SISTEM.....	31
4.1.	Lingkungan Implementasi	31
4.2.	Implementasi Permainan	31
4.2.1.	Implementasi Main Menu.....	32
4.2.2.	Implementasi Halaman How to Play	34
4.2.3.	Implementasi Tampilan Permainan	35
4.2.4.	Implementasi Cara Bermain	37
4.2.5.	Implementasi Spell Damage.....	51
4.2.6.	Implementasi Enemy Attack	53
4.2.7.	Implementasi Dragon (Boss) Attack	56
4.2.8.	Implementasi Tampilan Pause Menu	61
4.2.9.	Implementasi Tampilan Game Over	63
5 BAB V	PENGUJIAN DAN EVALUASI	67
5.1	Lingkungan Pengujian.....	67
5.2	Pengujian Sistem	67
5.2.1	Uji Coba Menu Permainan	68
5.2.2	Uji Coba Kontrol Permainan	69
5.2.3	Uji Coba Tampilan Pasue Menu.....	71
5.2.4	Uji Coba Akhir Permainan	73
5.2.5	Hasil Uji Coba.....	74
5.3	Pengujian Subjektivitas Sistem	75
5.3.1	Skenario Pengujian Pengguna	75
5.3.2	Daftar Penguji Permainan.....	77

5.3.3	Hasil Pengujian Pengguna.....	78
5.3.4	Kritik dan Saran Pengguna.....	81
5.3.5	Evaluasi Pengujian	82
6	BAB VI KESIMPULAN DAN SARAN.....	85
6.1.	Kesimpulan.....	85
6.2.	Saran.....	85
	DAFTAR PUSTAKA.....	87
	LAMPIRAN	89
	BIODATA PENULIS.....	101

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 3.1 Arsitektur Grand Wizard	17
Gambar 3.2 Panel yang Diginakan pada Permainan	18
Gambar 3.3 Button dan Icon yang Digunakan pada Permainan..	19
Gambar 3.4 Aset Gaia Terrain Maker	20
Gambar 3.5 Aset Character (Goblin & Dragon)	20
Gambar 3.6 Aset Particle Effect.....	21
Gambar 3.7 Rancangan Antarmuka Halaman Awal Permainan .	24
Gambar 3.8 Rancangan Antarmuka Halaman How to Play	25
Gambar 3.9 Rancangan Antarmuka Tampilan HUD	26
Gambar 3.10 Rancangan Antarmuka Cara Bermain (1)	27
Gambar 3.11 Rancangan Antarmuka Cara Bermain (2)	27
Gambar 3.12 Rancangan Antarmuka Pause Menu	28
Gambar 3.13 Rancangan Antarmuka Game Over	29
Gambar 4.1 Implementasi Main Menu.....	32
Gambar 4.2 Implementasi Halaman How to Play	34
Gambar 4.3 Implementasi Tampilan Permainan	36
Gambar 4.4 Implementasi Cara Bermain (1)	38
Gambar 4.5 Implementasi Cara Bermain (2)	38
Gambar 4.6 Implementasi Rancangan Pause Menu	61
Gambar 4.7 Implementasi Tampilan Game Over.....	64
Lampiran 1 Penguji Coba (1)	89
Lampiran 2 Penguji Coba (2)	89
Lampiran 3 Penguji Coba (3)	90
Lampiran 4 Penguji Coba (4)	90

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Kebutuhan Fungsional Sistem.....	13
Tabel 3.2 Kebutuhan Non-Fungsional Sistem.....	15
Tabel 4.1 Spesifikasi Lingkungan Implementasi	31
Tabel 5.1 Tabel Lingkungan Pengujian Sistem.....	67
Tabel 5.2 Pengujian Halaman Menu Permainan	68
Tabel 5.3 Pengujian Kontrol Permainan	70
Tabel 5.4 Pengujian Tampilan Pause Menu	71
Tabel 5.5 Pengujian Tampilan Akhir Permainan	73
Tabel 5.6 Hasil Evaluasi Uji Coba	75
Tabel 5.7 Rentang Nilai.....	76
Tabel 5.8 Pertanyaan Kuesioner Mengenai Karakteristik Pengguna	76
Tabel 5.9 Pernyataan Kuesioner Penilaian Pengguna	77
Tabel 5.10 Daftar Penguji	78
Tabel 5.11 Hasil Pengujian Pengguna.....	79
Tabel 5.12 Hasil Akhir Pengujian Pengguna	80
Tabel 5.13 Kritik dan Saran Pengguna.....	81
Tabel 5.14 Tabel Hasil Evaluasi Pengujian.....	83

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Main Menu	33
Kode Sumber 4.2 Implementasi Halaman How to Play	35
Kode Sumber 4.3 Implementasi Tombol Pause	37
Kode Sumber 4.4 Implementasi Cara Bermain (1)	42
Kode Sumber 4.5 Implementasi Cara Bermain (2)	45
Kode Sumber 4.6 Implementasi Cara Bermain (3)	51
Kode Sumber 4.7 Implementasi Spell Damage.....	52
Kode Sumber 4.8 Implementasi Enemy Attack	56
Kode Sumber 4.9 Implementasi Dragon (Boss) Attack	60
Kode Sumber 4.10 Implementasi Tampilan Pause Menu	63
Kode Sumber 4.11 Implementasi Tampilan Game Over	66

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Permainan digital merupakan salah satu media hiburan yang sedang berkembang pesat, dan bermain permainan digital merupakan salah satu cara untuk mengisi waktu luang. Hingga saat ini ada banyak jenis permainan digital yang sudah ada, dan juga teknologi yang digunakan dalam pembuatan permainan telah berkembang.

Perkembangan teknologi dari waktu ke waktu mengalami kemajuan yang sangat pesat. Seiring dengan perkembangan itu pula, teknologi yang dipakai untuk membuat permainan terus berkembang dan semakin banyak. Teknologi permainan yang sedang berkembang dan menarik perhatian dari para *gamers* adalah permainan berbasis realitas virtual.

Realitas virtual merupakan teknologi yang dapat membuat pengguna berinteraksi dengan suatu lingkungan yang disimulasikan oleh komputer, suatu lingkungan sebenarnya yang ditiru atau benar-benar suatu lingkungan yang hanya ada dalam imajinasi.

Dalam hal ini penulis ingin membuat sebuah permainan yang berjudul Grand Wizard. Grand Wizard merupakan sebuah permainan realitas virtual dengan *genre action* yang menggunakan kendali dari Touch Controller Oculus dan deteksi *hand gesture* menggunakan leap motion dan point cloud untuk menggunakan mantra yang dimiliki pemain dalam permainan.

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana merancang *gameplay* Grand Wizard dengan fitur 3D *First Person Shooter* (FPS) menggunakan konsep Realitas Virtual?

2. Bagaimana merancang sistem mekanik dari pemain menggunakan Leap Motion Controller untuk mendeteksi hand gesture pada tangan pemain?
3. Bagaimana merancang sistem mekanik untuk menggunakan mantra di dalam game dengan memanfaatkan sistem Point Cloud Recognition sebagai input gesture pada mantra yang akan digunakan?
4. Bagaimana implementasi dari rancangan game yang akan dibuat dibangun menggunakan *Game Engine Unity*?

1.3. Batasan Permasalahan

Batasan masalah pada tugas akhir ini antara lain:

1. Lingkungan pengembangan yang digunakan menggunakan aplikasi Unity 3D lisensi gratis dan aset gratis dari Unity Store.
2. Bahasa pemrograman yang digunakan meliputi bahasa pemrograman C# untuk script yang digunakan pada permainan.

1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah untuk membuat permainan realitas virtual menggunakan perangkat Oculus Rift dan Leap Motion Controller dengan implementasi dari Point Cloud Recognition sebagai kendali dari mantra yang akan digunakan dengan cara menggambar bentuk yang sesuai dengan mantra yang akan dipanggil.

1.5. Manfaat

Manfaat dari tugas akhir ini adalah sebagai berikut:

1. Memberikan pengalaman bermain yang baru menggunakan *Input Gesture* dari *Point Cloud Recognition* pada permainan realitas virtual.
2. Mengasah kemampuan pengambilan keputusan dan strategi pemain untuk menyelesaikan permainan.
3. Sebagai sarana hiburan untuk para pemain.

1.6. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu mengenai Game Engine Unity3D, Bahasa Pemrograman C#, Survival Game, First Person Shooter (FPS), Realitas Virtual, Leap Motion Controller, dan Point Cloud Recognition.

2. Analisis dan desain sistem

Analisis dan perancangan dalam pembuatan permainan Grand Wizard antara lain sebagai berikut :

1. Mempelajari konsep *Point Cloud Recognition*.
2. Mempelajari dokumentasi dan tutorial Oculus Rift.
3. Mempelajari dokumentasi dan tutorial Unity3D.
4. Mempelajari dokumentasi dan tutorial Leap Motion.
5. Perancangan *gameplay* realitas virtual Grand Wizard dengan menambahkan fitur *Point Cloud Recognition*.

3. Implementasi sistem

Permainan ini dibangun dengan menggunakan bahasa C# pada *game engine* Unity3D. Selain itu, menggunakan perangkat keras Oculus Rift dengan OS Windows 10 dan menggunakan kendali Leap Motion dan Touch Controller serta *Input Gesture* dari *Point Cloud* pada permainan realitas virtual Grand Wizard.

4. Pengujian dan evaluasi

Tahap pengujian dan evaluasi berisi pengujian aplikasi dan evaluasi berdasarkan hasil pengujian. Pada tahap ini dilakukan pengujian dari fungsionalitas perangkat lunak, apakah sesuai dengan yang diharapkan serta diharapkan tidak terdapat bug. Pengujian akan dilakukan kepada 10 mahasiswa Departemen Informatika, mereka akan menjadi penguji dan memainkan permainan Grand Wizard. Pengujian dilakukan untuk mengukur tingkat ketepatan pengenalan Gambar dari fitur Point Cloud yang diterapkan

5. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, tinjauan pustaka, metode, implementasi, proses yang telah dilakukan, pengujian, evaluasi dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

1.7. Sistematika Penulisan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

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

Bab II Tinjauan Pustaka

Bab ini menjelaskan beberapa pustaka-pustaka yang dijadikan penunjang dan berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai analisis dan perancangan sistem yang akan dibangun.

Bab IV Implementasi Sistem

Bab ini membahas mengenai bagaimana implementasi sistem dari analisis dan desain yang sudah dirancang.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari metode yang ditawarkan dalam tugas akhir untuk mengetahui kesesuaian metode dengan data yang ada.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang telah dilakukan. Bab ini juga membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

Merupakan bab tambahan yang berisi data atau daftar istilah yang penting pada tugas akhir ini.

BAB II

TINJAUAN PUSTAKA

Bab ini membahas pustaka/teori-teori yang menjadi dasar dalam pembuatan tugas akhir.

2.1. Realitas Virtual

Realitas virtual adalah sebuah teknologi yang membuat pengguna dapat berinteraksi dengan lingkungan yang ada dalam dunia maya yang disimulasikan oleh komputer, sehingga pengguna merasa berada di dalam lingkungan tersebut. Teknologi realitas virtual sejatinya telah banyak diterapkan di beberapa sektor industri seperti kedokteran, penerbangan, pendidikan, arsitek, militer, hiburan, dan lain sebagainya. Realitas virtual sangat membantu dalam menyimulasikan sesuatu yang sulit untuk dihadirkan secara langsung dalam dunia nyata. Tentunya ini dapat membuat lebih praktis dan lebih ekonomis.

Kelebihan utama dari realitas virtual adalah pengalaman yang membuat pengguna merasakan sensasi dunia nyata dalam dunia virtual/maya. Bahkan perkembangan teknologi realitas virtual saat ini memungkinkan tidak hanya indra penglihatan dan pendengaran saja yang bisa merasakan sensasi nyata dari realitas virtual, namun juga indra yang lainnya.

Untuk memunculkan sensasi nyata dari realitas virtual diperlukan perangkat pendukung. Paling tidak dibutuhkan sebuah alat seperti Oculus Rift, HTC Vive, dan lain lain yang mendukung realitas virtual untuk bisa merasakan sensasi realitas virtual. Terdapat 4 elemen penting dalam realitas virtual. Adapun 4 elemen itu adalah sebagai berikut:

- a. *Virtual world*, sebuah konten yang menciptakan dunia virtual dalam bentuk *screenplay* maupun *script*.
- b. *Immersion*, sebuah sensasi yang membawa pengguna teknologi realitas virtual merasakan ada di sebuah

lingkungan nyata yang padahal fiktif. *Immersion* dibagi dalam 3 jenis, yakni:

- *Mental immersion*, membuat mental penggunanya merasa seperti berada di dalam lingkungan nyata.
- *Physical immersion*, membuat fisik penggunanya merasakan suasana di sekitar lingkungan yang diciptakan oleh realitas virtual tersebut.
- *Mentally immersed*, memberikan sensasi kepada penggunanya untuk larut dalam lingkungan yang dihasilkan realitas virtual.

c. *Sensory feedback*, berfungsi untuk menyampaikan informasi dari virtual world ke indera penggunanya. Elemen ini mencakup visual (penglihatan), audio (pendengaran) dan sentuhan.

d. *Interactivity*, bertugas untuk merespons aksi dari pengguna, sehingga pengguna dapat berinteraksi langsung dalam medan fiktif atau virtual world

2.2. Unity 3D

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

Unity merupakan sebuah *tool* yang terintegrasi, untuk membuat *game*, arsitektur bangunan dan simulasi. Unity juga bisa digunakan untuk permainan PC dan permainan *online*. Untuk permainan *online* diperlukan sebuah *plugin*, yaitu Unity Web *Player*, sama halnya dengan Flash *Player* pada browser.

Unity tidak dirancang untuk proses desain atau *modelling*, dikarenakan Unity bukanlah sebuah *tools* untuk mendesain, Unity hanyalah sebuah *game engine* 2D atau 3D. Banyak hal yang bisa dilakukan dengan Unity dengan berbagai fitur yang dimilikinya,

seperti adanya fitur *audio reverb zone*, *particle effect*, dan *sky box* untuk menambahkan langit. Fitur *scripting* yang disediakan mendukung 3 bahasa pemrograman, JavaScript, C#, dan Boo.

2.3. Bahasa Pemrograman C#

C# (dibaca: c sharp) merupakan sebuah bahasa pemrograman berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangkaan .NET *framework*. Bahasa pemrograman ini dibuat berdasarkan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic dan lain-lain dengan beberapa penyederhanaan.

2.4. First Person Shooter

Pengembangan *First Person Shooter* (FPS) dimulai pada tahun 1973, dan 1974, Spasim. Setelah itu, lebih banyak judul bermunculan seperti MIDI Maze tahun 1987, kemudian bersatu dengan genre kekerasan membentuk Wolfenstein 3D tahun 1992, akan tetapi lebih kalah populer dengan permainan Doom. Half Life (1998) dan Half Life 2 (2004) meningkatkan narasi dan elemen teka-teki.

2.5. $\$P$ Point Cloud Recognizer

$\$P$ Point-Cloud Recognizer adalah pengenalan stroke-gesture 2-D yang dirancang untuk prototipe cepat sebuah antarmuka pengguna berbasis gesture. $\$P$ adalah pencocokan point-cloud pertama dalam $\$$ -family recognizer, yang mencakup $\$I$ untuk unistrokes dan $\$N$ untuk multistrok. Meskipun sekitar setengah dari kode $\$P$ berasal dari $\$I$, tidak seperti $\$I$ dan $\$N$, $\$P$ tidak mewakili gerakan sebagai poin yang berurutan, tetapi sebagai awan titik yang tidak berurutan. Dengan menyatakan gerakan sebagai *point-cloud*, $\$P$ dapat menangani unistrokes dan multistrokes secara setara, dan tanpa overhead kombinatorik sebesar $\$N$, karena jumlah stroke, urutan, dan arah semuanya diabaikan. Ketika membandingkan dua *point-cloud*, $\$P$ menemukan solusi untuk masalah penugasan klasik antara dua grafik bipartit menggunakan perkiraan algoritma Hungaria. Menggantikan $\$P$ adalah $\$P+$, yang lebih akurat dan dioptimalkan untuk orang-orang dengan penglihatan rendah; dan $\$Q$, yang merupakan pengenalan super cepat yang dioptimalkan untuk ponsel berdaya rendah dan perangkat yang dapat dikenakan, mencapai kecepatan $142\times$ dengan akurasi yang sedikit ditingkatkan.

2.6. Oculus Rift

Oculus Rift adalah sistem realitas virtual yang benar-benar menenggelamkan anda ke dalam dunia maya. Rift adalah peranti layar ikat kepala untuk menampilkan realitas virtual yang saat ini dikembangkan oleh Oculus VR

2.7. Leap Motion Controller

Leap Motion Controller Merupakan suatu perangkat yang dikembangkan oleh Leap motion, Inc pada tahun 2008 oleh David Holz. Leap Motion Controller sendiri digunakan sebagai *Input* dari komputer tanpa sentuh. Bisa dikatakan bahwa Leap Motion ini merupakan pengganti *Mouse* pada komputer karena memiliki fungsi yang sama.

Leap Motion Merupakan Sebuah Perangkat Sensor hardware yang mendukung gerakan tangan dan jari sebagai *Input*, tetapi tidak memerlukan kontak tangan atau sentuhan langsung terhadap komputer.

Leap Motion dari sisi *hardware* sebenarnya cukup sederhana. Inti dari Leap Motion ini terletak pada pemanfaatan dua kamera *stereo* dan terdapat tiga lampu pemancar inframerah yang menyebar secara konvergen sehingga mampu untuk menjangkau area yang lebih luas. Jadi pada tahap ini inframerah akan menyebar untuk membentuk sebuah area seperti setengah lingkaran dengan jarak jangkauan maksimal 50 cm.

Dalam tugas akhir ini, Leap Motion dimanfaatkan untuk mendapatkan gerakan kedua tangan sebagai acuan kendali pergerakan dan menggambar pola garis untuk memunculkan mantra. Gerakan tangan ini diamati oleh sensor pada Leap Motion Controller sebagai input yang dapat memicu kondisi tertentu di dalam Permainan ini.

2.8. Library Leap Motion

SDK Leap Motion berisi dua *library* dasar yang mendefinisikan API ke data pelacakan Leap Motion. Satu *library* ditulis dengan bahasa C++, yang kedua ditulis dalam bahasa. *Class* pembungkus untuk *library* ini mendefinisikan bahasa pengikat untuk C# dan Objective-C. Dalam Unity 5, baik versi Pro atau Personal mendukung *plugins*. *Plugin* Unity menggunakan definisi kelas C# dalam folder LeapC dari *asset* inti. Library ini memuat library leapC.dll (Windows).

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tentang analisis dan perancangan Sistem Realitas Virtual untuk *Grand Wizard*. Pembahasan yang dilakukan meliputi analisis sistem, perancangan sistem, skenario simulasi, dan perancangan antar muka sistem.

3.1. Analisis Sistem

Sub bab ini akan membahas tentang analisis kebutuhan sistem, meliputi spesifikasi kebutuhan sistem, baik itu kebutuhan fungsional sistem maupun kebutuhan non-fungsional sistem, dan identifikasi pengguna sistem.

3.1.1. Spesifikasi Kebutuhan Sistem

Pada sistem ini terdapat beberapa kebutuhan fungsional dan kebutuhan non-fungsional yang mendukung berjalannya sistem. Kebutuhan fungsional sistem dapat dilihat pada Tabel 3.1, sedangkan kebutuhan non-fungsional sistem dapat dilihat pada Tabel 3.2.

Tabel 3.1 Kebutuhan Fungsional Sistem

Kode	Deskripsi
F1	Pemain dapat melihat <i>Main Menu</i>
F2	Pemain dapat melihat <i>How to Play Menu</i>
F3	Pemain dapat keluar dari permainan
F4	Pemain dapat menggambar di udara
F5	Pemain dapat bergerak menggunakan <i>Touch Controller</i>
F6	Point-Cloud Recognizer dapat mendeteksi bentuk garis yang dibuat oleh pemain
F7	Pemain dapat mengeluarkan <i>Spell</i> sesuai gambar yang dibuat
F8	Musuh dapat menyerang pemain

F9	Pemain dapat memulai kembali permainan
F10	Pemain memiliki <i>Health Point</i> yang dapat berkurang
F11	Musuh memiliki <i>Health Point</i> yang dapat berkurang
F12	Pemain dapat memenangkan permainan dengan mengalahkan <i>Dragon (Boss)</i>

Tabel 3.2 Kebutuhan Non-Fungsional Sistem

Kode	Deskripsi
NF1	Sistem dapat dijalankan pada perangkat komputer
NF2	Sistem dapat dijalankan pada Sistem Operasi Windows 10
NF3	Sistem memiliki tampilan antar muka yang mudah dipahami.
NF4	Sistem dapat digunakan dengan mudah.

3.1.2. Identifikasi Pengguna

Pengguna yang dapat menggunakan permainan Grand Wizard ini adalah siapa saja (umum). Sehingga, pengguna berhak menggunakan seluruh fungsionalitas yang terdapat pada sistem.

3.2. Perancangan Sistem

Sub bab ini membahas tentang bagaimana sistem ini dirancang, meliputi deskripsi umum sistem dan arsitektur sistem.

3.2.1. Deskripsi Umum Sistem

Grand Wizard merupakan permainan realitas virtual berbasis *personal computer* bergenre FPS dan *action game* yang menggunakan perangkat Oculus Rift dan Leap Motion dengan *Point Cloud Recognition* sebagai kendalinya. Sistem ini merupakan permainan realitas virtual yang menghadirkan atmosfer dan suasana ketika berada di dunia fantasi.

Pembangunan sistem ini dimulai dari membuat *terrain* yang bertema fantasi yang akan dijadikan sebagai lingkungan nyata di dalam permainan VR.

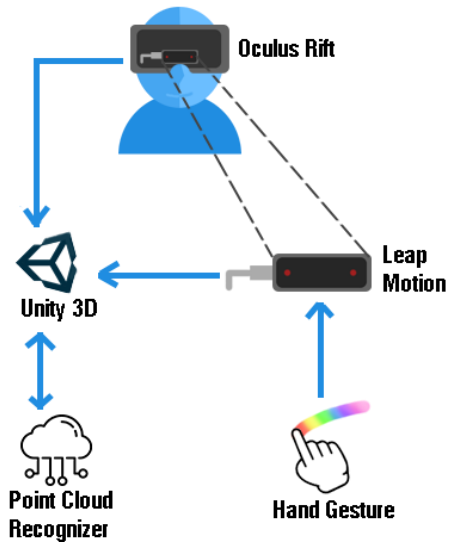
Proses pembangunan sistem selanjutnya yaitu dengan merancang gameplay yang sesuai dengan tema dari permainan ini. Mencari aset aset yang sesuai, dan melengkapi aset dengan mendesainnya secara manual.

3.2.2. Arsitektur Sistem

Grand Wizard mengintegrasikan antara *Personal Computer*, Oculus Rift, dan Leap Motion. Diproses di PC dan ditampilkan di Oculus Rift, dengan menggunakan Leap Motion, permainan Grand Wizard dapat mendeteksi gerakan tangan pemain yang berada depan sensor Leap Motion. Lalu dengan menggunakan pergerakan tangan dan jari, permainan Grand Wizard dapat mengetahui apa yang digambar oleh pemain menggunakan deteksi dari *Point Cloud Recognition*. Dengan begini, permainan Grand Wizard lebih terasa nyata, karena menggunakan tangan dan gerakan sendiri untuk berinteraksi dalam permainan.

Dari *player*, sistem menerima input berupa *gesture* dan titik-titik dari gambar garis dengan isyarat gerakan tangan melalui Leap Motion, lalu diproses oleh Point Cloud Recognition untuk dijadikan data, hasil dari pemrosesan data tersebut dibandingkan dengan *Dataset* berupa kumpulan data titik-titik gambar yang sudah disediakan, lalu dikeluarkan kembali dalam output *string* dari nama gambar yang sesuai. Kemudian, hasil deteksi gambar tersebut disesuaikan dengan jenis *spell* yang sesuai dengan nama keluaran dari Point Cloud Recognition.

Seperti yang dijelaskan diatas, Gambar 3.1 berikut merupakan tampilan diagram dari alur sistem permainan Grand Wizard.



Gambar 3.1 Arsitektur Grand Wizard

3.2.3. Perancangan Tampilan Antarmuka

Pada sub bab ini menjelaskan tentang user interface dan aset yang terdapat dalam permainan. Selain itu akan dibahas pula penggunaannya pada permainan.

3.2.4. User Interface

User Interface pada permainan ini antara lain:

1. *Panel*

Panel digunakan pada halaman *Main Menu*, *Pause Menu*, dan *Game Over*. Gambar 3.2 merupakan tampilan *Panel* yang digunakan.



Gambar 3.2 Panel yang Diginakan pada Permainan

2. Button dan Icon

Tombol digunakan pada tiap halaman yang ada dalam permainan. Icon digunakan untuk memperbagus tampilan game dan sebagai petunjuk dalam permainan. Gambar 3.3 merupakan tampilan dari tombol dan ikon yang digunakan.



Gambar 3.3 Button dan Icon yang Digunakan pada Permainan

3.2.5. Aset Permainan

Aset yang digunakan pada permainan ini antara lain:

1. *Environment*

Environment digunakan pada setiap halaman permainan yang ada dalam permainan. Gambar 3.4 merupakan tampilan dari *environment* yang digunakan. *Environment* yang digunakan saat ini berasal dari aset berbayar di Unity Store yang bernama “Gaea Terrain Maker”.



Gambar 3.4 Aset Gaia Terrain Maker

2. *Character*

Aset *Character* yang digunakan pada pembuatan game ini menggunakan aset berbayar dari Unity bernama “HEROIC FANTASY CREATURE FULL PACK”. Karakter yang digunakan pada permainan ini dapat dilihat pada Gambar 3.5.



Gambar 3.5 Aset Character (Goblin & Dragon)

3. *Particle Effect*

Particle Effect digunakan sebagai efek tambahan pada permainan agar permainan lebih terasa menarik. Beberapa spell yang digunakan pada permainan juga menggunakan aset partikel ini. Pada pembuatan permainan ini digunakan aset berbayar dari Unity yang bernama “KRIPTO FX”. Beberapa efek partikel yang digunakan pada permainan ini dapat dilihat pada Gambar 3.6.



Gambar 3.6 Aset Particle Effect

3.3. Perancangan Skenario Simulasi

Pada sub bab ini menjelaskan tentang skenario permainan untuk menentukan kondisi menang atau kalah. Selain itu akan dibahas pula aturan permainan dari permainan.

3.3.1. Alur Simulasi

Alur permainan dari permainan Grand Wizard antara lain:

1. Saat permainan pertama kali dijalankan, pemain akan melihat tampilan *Main Menu* yang memiliki 3 tombol interaksi yaitu, tombol *Play*, *How to Play*, dan *Quit*.
2. Jika pemain menekan tombol *How to Play*, maka akan muncul tampilan cara bermain dan beberapa informasi mengenai

- permainan beserta daftar *spell* yang dapat digunakan dan bagaimana cara memanggilnya.
3. Jika pemain menekan tombol *Quit*, maka pemain akan keluar dari permainan dan permainan akan tertutup.
 4. Untuk mulai bermain, pemain harus menekan tombol *Play*.
 5. Pemain akan ditempatkan pada titik *spawn* awal pada ujung *map* dalam keadaan *Health Point* pada kondisi penuh dan semua *spell* tidak *cooldown*.
 6. Saat permainan telah dimulai, pemain dapat menekan tombol *Pause* untuk menjeda permainan dan menampilkan *Pause Menu*.
 7. Pada *Pause Menu*, pemain dapat menekan beberapa pilihan tombol.
 8. Jika pemain menekan tombol *Restart*, maka permainan akan diulang kembali.
 9. Jika pemain menekan tombol *How to Play*, maka akan menampilkan tampilan cara bermain dan daftar *spell* yang dapat digunakan.
 10. Jika pemain menekan tombol *Quit*, maka permainan akan terhenti an pemain keluar dari permainan.
 11. Pada keadaan awal musuh akan dalam keadaan *idle* hingga pemain memasuki jarak serang mereka atau ada interaksi serangan dari pemain.
 12. Musuh *Goblin* tersebar di sepanjang area dengan jumlah 3 *Goblin* di setiap satu *Horde* (kelompok).
 13. Musuh *Goblin* akan menyerang pemain ketika jarak pemain memasuki jarak serang *Goblin*.
 14. *Goblin* akan mengejar pemain dan menyerang ketika berada jarak terdekat dengan pemain dan akan berhenti mengejar saat pemain keluar dari jarak serangan *Goblin*.
 15. Posisi *Dragon* (*Boss*) ada di ujung area permainan.
 16. *Dragon* akan menembakkan bola api ke arah pemain setiap 10 detik (HP > 50%), 6 detik (HP < 50%), 3 detik (HP < 10%).

3.3.2. Aturan Main

Dalam memainkan permainan ini ada beberapa aturan main sebagai berikut:

1. Pemain dinyatakan kalah jika *Health Point* pemain habis.
2. Pemain harus mengalahkan semua musuh yang tersebar di sepanjang area.
3. Pemain harus mengalahkan *Dragon (Boss)* untuk memenangkan permainan.
4. Untuk dapat bergerak, Pemain harus menggunakan *analog controller* pada tangan kirinya.
5. Untuk mengeluarkan *spell*, pemain harus menggunakan tangan kanannya dan membuka jari telunjuknya untuk menggambar di udara. Lalu membuka seluruh jari untuk melancarkan serangan.
6. Tipe *spell* yang dikeluarkan bergantung pada gambar yang digambar oleh pemain.
7. Setiap *spell* memiliki *cooldown* masing-masing. Jika *spell* dalam keadaan *cooldown*, pemain tidak bisa mengeluarkan *spell* tersebut hingga waktu *cooldown* selesai.

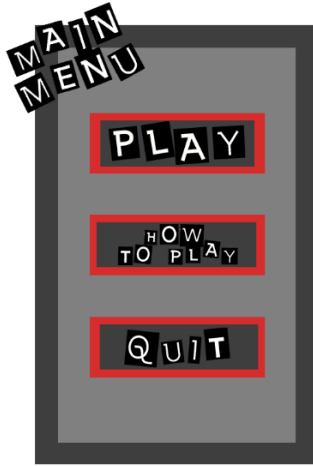
3.4. Perancangan Tampilan Antarmuka

Sub bab ini membahas bagaimana rancangan antarmuka pengguna yang akan digunakan untuk tugas akhir. Rancangan antarmuka yang dibahas meliputi ketentuan masukan dan rancangan halaman tampilan. Di dalam aplikasi ini terdapat beberapa tampilan, yaitu tampilan menu permainan, tampilan cara bermain, tampilan permainan (inti permainan), dan tampilan akhir permainan.

3.4.1. Tampilan Menu Permainan

Tampilan menu permainan merupakan tampilan yang pertama kali muncul ketika aplikasi dijalankan. Pada tampilan awal terdapat tiga tombol, yaitu tombol *Play*, *How to Play*, dan *Quit*.

Tampilan rancangan antarmuka dapat dilihat pada Gambar 3.7 berikut:



Gambar 3.7 Rancangan Antarmuka Halaman Awal Permainan

1. Tombol **Play**, berfungsi untuk memulai permainan pertama kali.
2. Tombol **How to Play**, berfungsi untuk menampilkan panel berupa tampilan cara bermain dan beberapa informasi mengenai permainan beserta daftar *spell* yang dapat digunakan dan bagaimana cara memanggilnya..
3. Tombol **Quit**, berfungsi untuk keluar dari permainan dan permainan akan tertutup.

3.4.2. Tampilan Halaman How to Play

Tampilan halaman *How to Play* menampilkan berbagai macam instruksi dalam permainan dan juga informasi tentang cara bermain dan daftar *spell* yang dapat digunakan beserta cara

memanggilnya. Rancangan haman pengaturan dapat dilihat pada Gambar 3.8.

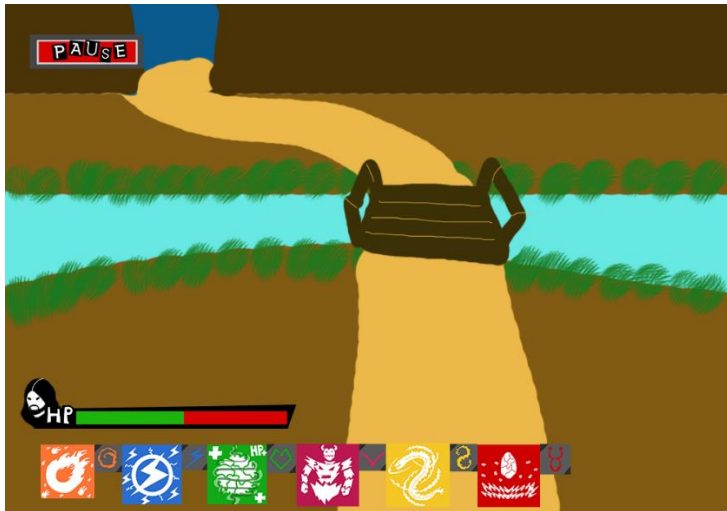


Gambar 3.8 Rancangan Antarmuka Halaman How to Play

1. Panel bagian **Kiri**, menunjukkan aturan main secara singkat.
2. Panel bagian **Tengah**, menunjukkan cara bermain permainan.
3. Panel bagian **Kanan**, menunjukkan daftar *spell* yang dapat digunakan dan gambar untuk memanggilnya.

3.4.3. Tampilan Permainan

Tampilan permainan merupakan tampilan awal dari permainan ketika pemain menekan tombol *Play* pada *Main Menu*. Tampilan ini menampilkan kondisi awal permainan dan tampilan *HUD (Heads Up Display)* yang menampilkan informasi *Health Point*, kondisi *spell* saat ini, dan tombol *Pause*. Rancangan tampilan permainan dapat dilihat pada Gambar 3.9, 3.10, dan 3.11.

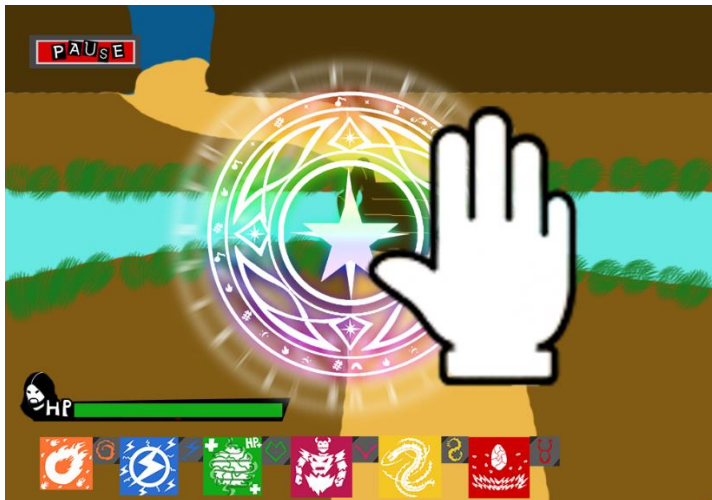


Gambar 3.9 Rancangan Antarmuka Tampilan HUD

Untuk petunjuk cara bermain dapat dilihat pada Gambar dimana pemain dapat menggambar di udara dengan cara membuka jari telunjuknya dan mulai menggambar di udara. Untuk melancarkan serangan *spell*, pemain harus membuka seluruh jari mereka. Kemudian sistem akan mengeluarkan *spell* yang sesuai dengan gambar yang pemain buat.



Gambar 3.10 Rancangan Antarmuka Cara Bermain (1)



Gambar 3.11 Rancangan Antarmuka Cara Bermain (2)



Gambar 3.12 Rancangan Antarmuka Pause Menu

3.4.4. Tampilan Pause Menu

Tampilan Pause Menu akan muncul jika pemain menekan tombol Pause pada HUD. Rancangan tampilan *Pause Menu* dapat dilihat pada Gambar 3.12. Berikut penjelasan dari konten Pause Menu:

1. Tombol **Restart**, berfungsi untuk memulai permainan kembali dari awal.
2. Tombol **How to Play**, berfungsi untuk menampilkan panel berupa tampilan cara bermain dan beberapa informasi mengenai permainan beserta daftar *spell* yang dapat digunakan dan bagaimana cara memanggilnya..
3. Tombol **Quit**, berfungsi untuk keluar dari permainan dan permainan akan tertutup.
4. Tombol **Close [X]**, berfungsi untuk menutup Menu Pause dan melanjutkan permainan.

3.4.5. Tampilan Game Over

Tampilan *Game Over* adalah tampilan akhir permainan yang muncul ketika *Health Point* pemain habis atau pemain telah menyelesaikan permainan dengan cara membunuh *Dragon (Boss)*. Rancangan tampilan *Game Over* dapat dilihat pada Gambar 3.13.



Gambar 3.13 Rancangan Antarmuka Game Over

Berikut penjelasan dari konten tampilan *Game Over* ketika permainan berakhir atau pemain kehabisan *Health Point*:

1. Tombol **Restart**, berfungsi untuk memulai permainan kembali dari awal.
2. Tombol **Quit**, berfungsi untuk keluar dari permainan dan permainan akan tertutup.

Saat permainan berakhir, pemain dapat memilih untuk memulai kembali permainan atau mengakhiri bermain dengan menutup permainan.

BAB IV IMPLEMENTASI SISTEM

Bab ini membahas implementasi dari perancangan sistem sesuai dengan perancangan yang telah dibuat. Bahasa pemrograman yang digunakan untuk implementasi sistem adalah Bahasa pemrograman C#.

4.1. Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan tugas akhir ini memiliki spesifikasi perangkat keras dan perangkat lunak yang ditunjukkan oleh Tabel 4.1.

Tabel 4.1 Spesifikasi Lingkungan Implementasi

Perangkat	Spesifikasi
Perangkat Keras	<ul style="list-style-type: none">• Perangkat Pengembangan Sistem: Intel Core i7-7700 3.6GHz, NVIDIA GeForce GTX 1060 3GB, RAM 8GB• Oculus Rift• Leap Motion Controller
Perangkat Lunak	<ul style="list-style-type: none">• Sistem Operasi: Microsoft Windows 10 64-bit• Perangkat Pengembang: Unity 2018.2.18f1 (64-bit) dan Visual Studio Community 2017• Perangkat Pembantu: Microsoft Word 2017, Corel Draw X7, Adobe Photoshop CS6, Audacity, dan Blender

4.2. Implementasi Permainan

Implementasi dari masing-masing fungsi utama dituliskan menggunakan kode berbahasa C#. implementasi fungsi diurut berdasarkan antarmuka-antarmuka yang ada pada permainan.

4.2.1. Implementasi Main Menu

Gambar 4.1 merupakan hasil implementasi dari rancangan Main Menu.



Gambar 4.1 Implementasi Main Menu

Berikut dijelaskan penjelasan dari konten MainMenu sebagai berikut:

1. Tombol **Play**, berfungsi untuk memulai permainan pertama kali.
2. Tombol **How to Play**, berfungsi untuk menampilkan panel berupa tampilan cara bermain dan beberapa informasi mengenai permainan beserta daftar *spell* yang dapat digunakan dan bagaimana cara memanggilnya..
3. Tombol **Quit**, berfungsi untuk keluar dari permainan dan permainan akan tertutup.

Pada Kode Sumber 4.1 dijelaskan beberapa fungsi yang dibuat untuk menjalankan sistem Main Menu berupa fungsi untuk tombol Play untuk dapat memulai permainan pertama kali, tombol How to Play untuk menampilkan cara bermain, dan tombol Quit untuk keluar dan menutup permainan.


```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.SceneManagement;
5.
6. public class GameManagement : MonoBehaviour {
7.
8.     public GameObject howToPlayUI;
9.     public GameObject mainMenuUI;
10.
11.    public void Play()
12.    {
13.        SceneManager.LoadScene("MainGame");
14.    }
15.
16.    public void Quit()
17.    {
18.        Debug.Log("Quit Game");
19.        Application.Quit();
20.    }
21.
22.    public void HowToPlay()
23.    {
24.        Debug.Log("How To Play");
25.
26.        mainMenuUI.SetActive(false);
27.        howToPlayUI.SetActive(true);
28.
29.    }
30.
31.    public void Back()
32.    {
33.        Debug.Log("Back");
34.
35.        mainMenuUI.SetActive(true);
36.        howToPlayUI.SetActive(false);
37.    }
38. }
```

Kode Sumber 4.1 Implementasi Main Menu

4.2.2. Implementasi Halaman How to Play

Pada Gambar 4.2 dapat dilihat hasil Implementasi tampilan halaman *How to Play*.



Gambar 4.2 Implementasi Halaman How to Play

Berikut dijelaskan penjelasan dari tiap konten pada halaman *How to Play* sebagai berikut:

1. Panel bagian **Kiri**, menunjukkan aturan main secara singkat.
2. Panel bagian **Tengah**, menunjukkan cara bermain permainan.
3. Panel bagian **Kanan**, menunjukkan daftar *spell* yang dapat digunakan dan gambar untuk memanggilnya.

Pada Kode Sumber 4.2 dijelaskan beberapa fungsi yang dipakai pada implementasi halaman *How to Play*. Fungsi yang dibuat hanya digunakan sebagai *trigger* untuk tombol *Back* yang digunakan untuk menutup halaman *How to Play* dan kembali ke halaman *Main Menu*.

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5.
6. public class GameManagement : MonoBehaviour {
7.
8.     public GameObject mainMenuUI;
9.     public GameObject howToPlayUI;
10.
11.
12.     public void Back()
13.     {
14.         Debug.Log("Back");
15.
16.         mainMenuUI.SetActive(true);
17.         howToPlayUI.SetActive(false);
18.     }
19. }
```

Kode Sumber 4.2 Implementasi Halaman How to Play

4.2.3. Implementasi Tampilan Permainan

Pada Gambar 4.3 ditampilkan hasil implementasi dari tampilan permainan berupa *HUD (Heads Up Display)* yang menampilkan informasi tentang *Health Point* pemain, kondisi *spell* saat ini, dan adanya tombol Pause pada pojok kiri atas layar.



Gambar 4.3 Implementasi Tampilan Permainan

Untuk penjelasan kode sumbernya dapat dilihat pada Kode Sumber 4.3 dimana dibuatnya fungsi untuk tombol *Pause Game* yang berguna untuk menjeda permainan dan membuka tampilan *Pause Menu*.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.SceneManagement;
5.
6. public class GameManagement : MonoBehaviour {
7.
8.
9.     public GameObject pauseMenuUI;
10.
11.    public GameObject pauseButton;
12.
13.    public Demo demo;
14.    public OVRPlayerController move;
15.    public void PauseGame()
16.    {

```

```
17.     Debug.Log("Pause");
18.     demo.enabled = false;
19.     move.enabled = false;
20.     Time.timeScale = 0.4f;
21.     pauseButton.SetActive(false);
22.     pauseMenuUI.SetActive(true);
23. }
24.
25. public void ClosePause()
26. {
27.     Time.timeScale = 1f;
28.     demo.enabled = true;
29.     move.enabled = true;
30.     pauseButton.SetActive(true);
31.
32.     pauseMenuUI.SetActive(false);
33. }
34.
35. }
```

Kode Sumber 4.3 Implementasi Tombol Pause

4.2.4. Implementasi Cara Bermain

Pada Gambar 4.4 dan 4.5 dapat dilihat hasil implementasi dari cara bermain dimana pemain menggunakan jari telunjuk pemain untuk menggambar di udara, lalu pemain membuka seluruh jari pemain untuk meluncurkan spell berdasarkan kecocokan gambar dengan dataset dan tipe spell yang berhubungan.



Gambar 4.4 Implementasi Cara Bermain (1)



Gambar 4.5 Implementasi Cara Bermain (2)

Pada Kode Sumber 4.4 dibuat *script* untuk menangani sistem menggambar di udara dan menyambungkannya dengan sistem *Point Cloud Recognizer*.

```
1. using UnityEngine;  
2. using System;
```

```
3. using System.Collections;
4. using System.Collections.Generic;
5. using System.IO;
6. using Leap.Unity;
7. using PDollarGestureRecognizer;
8.
9.
10. public class Demo : MonoBehaviour
11. {
12.
13.     public ExtendedFingerDetector hand;
14.
15.     public Transform gestureOnScreenPrefab;
16.     public Transform indexTip;
17.     private bool triggered = true;
18.     public String resultSpell;
19.
20.     private List<Gesture> trainingSet = new List<
    Gesture>();
21.
22.     private List<Point> points = new List<Point>(
    );
23.     private int strokeId = -1;
24.
25.     private Vector3 virtualKeyPosition = Vector2.
    zero;
26.     private int vertexCount = 0;
27.
28.     private List<LineRenderer> gestureLinesRender
    er = new List<LineRenderer>();
29.     private LineRenderer currentGestureLineRender
    er;
30.
31.     private string message;
32.     private bool recognized = false;
33.     private bool flag = false;
34.
35.     void Start()
36.     {
37.         //Load pre-made gestures
38.         TextAsset[] gesturesXml = Resources.LoadA
    ll<TextAsset>("GestureSet/10-stylus-MEDIUM/");
```

```
39.         foreach (TextAsset gestureXml in gestures
        Xml)
40.             trainingSet.Add(GestureIO.ReadGesture
        FromXML(gestureXml.text));
41.
42.         //Load user custom gestures
43.         string[] filePaths = Directory.GetFiles(A
        pplication.persistentDataPath, "*.xml");
44.         foreach (string filePath in filePaths)
45.             trainingSet.Add(GestureIO.ReadGesture
        FromFile(filePath));
46.         Debug.Log(Application.persistentDataPath.
        ToString());
47.         Transform tmpGesture = Instantiate(gestur
        eOnScreenPrefab, transform.position, transform.ro
        tation) as Transform;
48.
49.         currentGestureLineRenderer = tmpGesture.G
        etComponent<LineRenderer>();
50.
51.         gestureLinesRenderer.Add(currentGestureLi
        neRenderer);
52.     }
53.
54.     void Update()
55.     {
56.         if (hand.isHandActive)
57.         {
58.             virtualKeyPosition = new Vector3(inde
        xTip.transform.position.x, indexTip.transform.pos
        ition.y, indexTip.transform.position.z);
59.         }
60.
61.         if (!hand.isHandActive && triggered)
62.         {
63.             triggered = false;
64.             Recognized();
65.
66.             if (recognized)
67.             {
68.                 Debug.Log("recognized");
69.
```



```
70.             recognized = false;
71.             strokeId = -1;
72.
73.             points.Clear();
74.
75.             foreach (LineRenderer lineRenderere
76. r in gestureLinesRenderer)
77.             {
78.                 lineRenderer.SetVertexCount(0
79. );
80.                 Destroy(lineRenderer.gameObject);
81.             }
82.             gestureLinesRenderer.Clear();
83.         }
84.
85.         ++strokeId;
86.
87.         Transform tmpGesture = Instantiate(gestureOnScreenPrefab, transform.position, transform.rotation) as Transform;
88.
89.         currentGestureLineRenderer = tmpGesture.GetComponent<LineRenderer>();
90.
91.         gestureLinesRenderer.Add(currentGestureLineRenderer);
92.
93.         vertexCount = 0;
94.     }
95.
96.     if (hand.isHandActive)
97.     {
98.         triggered = true;
99.         flag = true;
100.         points.Add(new Point(virtualKeyPosition.x, virtualKeyPosition.y, strokeId));
101.         currentGestureLineRenderer.SetVertexCount(++vertexCount);
```

```

102.         currentGestureLineRenderer.Set
           Position(vertexCount - 1, virtualKeyPosition);
103.     }
104. }
105.
106.     void Recognized()
107.     {
108.         if (!hand.isHandActive && flag)
109.         {
110.             flag = false;
111.             recognized = true;
112.
113.             Gesture candidate = new Gestur
           e(points.ToArray());
114.             int n = points.ToArray().Lengt
           h;
115.
116.             if (n == 1)
117.             {
118.                 return;
119.             }
120.             else
121.             {
122.                 Result gestureResult = Poi
           ntCloudRecognizer.Classify(candidate, trainingSet
           .ToArray());
123.                 message = gestureResult.Ge
           stureClass + " " + gestureResult.Score;
124.                 resultSpell = gestureResul
           t.GestureClass;
125.                 Debug.Log(message);
126.             }
127.
128.
129.         }
130.     }
131. }

```

Kode Sumber 4.4 Implementasi Cara Bermain (1)

Cara kerja kode sumber diatas adalah menghubungkan antara hasil gambar dari *Line Renderer* yang dibuat dengan Leap Motion lalu menyimpan titik-titik koordinat dari *Line Renderer* tersebut dan menyimpannya pada sebuah *Array List*. Kemudian hasil *Array List* tersebut dikirimkan ke Kode Sumber 4.5 dan dicocokkan dengan *Dataset* yang sudah tersedia. Kode Sumber 4.4 akan memanggil fungsi **Classify()** pada Kode Sumber 4.5 yang berfungsi untuk membandingkan antara kedua *Array List*.

```

1. using System;
2. using System.Collections.Generic;
3.
4. using UnityEngine;
5.
6. namespace PDollarGestureRecognizer
7. {
8.
9.     public class PointCloudRecognizer
10.    {
11.        public static Result Classify(Gesture can
12.        didate, Gesture[] trainingSet)
13.        {
14.            float minDistance = float.MaxValue;
15.            string gestureClass = "";
16.            foreach (Gesture template in training
17.            Set)
18.            {
19.                float dist = GreedyCloudMatch(can
20.                didate.Points, template.Points);
21.                if (dist < minDistance)
22.                {
23.                    minDistance = dist;
24.                    gestureClass = template.Name;
25.                }
26.            }
27.
28.            return gestureClass == "" ? new Result
29.            t() { GestureClass = "No match", Score = 0.0f } :
30.            new Result() { GestureClass = gestureClass, Scor

```

```

    e = Mathf.Max((minDistance - 2.0f) / -
2.0f, 0.0f) };
26.     }
27.
28.     private static float GreedyCloudMatch(Point[] points1, Point[] points2)
29.     {
30.         int n = points1.Length;
31.         float eps = 0.5f;
32.         int step = (int)Math.Floor(Math.Pow(n
, 1.0f - eps));
33.         float minDistance = float.MaxValue;
34.         for (int i = 0; i < n; i += step)
35.         {
36.             float dist1 = CloudDistance(point
s1, points2, i);
37.             float dist2 = CloudDistance(point
s2, points1, i);
38.             minDistance = Math.Min(minDistanc
e, Math.Min(dist1, dist2));
39.         }
40.         return minDistance;
41.     }
42.
43.     private static float CloudDistance(Point[
] points1, Point[] points2, int startIndex)
44.     {
45.         int n = points1.Length;
46.         bool[] matched = new bool[n];
47.         Array.Clear(matched, 0, n);
48.
49.         float sum = 0;
50.         int i = startIndex;
51.         do
52.         {
53.             int index = -1;
54.             float minDistance = float.MaxValu
e;
55.             for (int j = 0; j < n; j++)
56.                 if (!matched[j])
57.                     {

```

```

58.         float dist = Geometry.Sqr
           EuclideanDistance(points1[i], points2[j]);
59.         if (dist < minDistance)
60.         {
61.             minDistance = dist;
62.             index = j;
63.         }
64.     }
65.     matched[index] = true;
66.     float weight = 1.0f - ((i - start
           Index + n) % n) / (1.0f * n);
67.     sum += weight * minDistance;
68.     i = (i + 1) % n;
69.     } while (i != startIndex);
70.     return sum;
71. }
72. }
73. }

```

Kode Sumber 4.5 Implementasi Cara Bermain (2)

Kode Sumber 4.5 akan mengembalikan keluaran berupa tipe data *string* yang akan digunakan pada Kode Sumber 4.4 sebagai penentu jenis *spell* yang akan digunakan atau diluncurkan. Selanjutnya hasil keluaran tersebut akan disimpan pada suatu variabel dan dikirimkan ke Kode Sumber 4.6 dimana nantinya akan dicocokkan dengan menggunakan statement *Switch Case* pada Kode Sumber 4.6.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using PDollarGestureRecognizer;
5.
6. public class SpellCasting : MonoBehaviour
7. {
8.
9.     protected string spellWord = "";
10.    public Demo resultDemo;
11.    public GameObject castPoint;

```

```

12.     public List<GameObject> fp = new List<GameObj
      ect>();
13.     public List<GameObject> vfx = new List<GameOb
      ject>();
14.     public List<GameObject> block = new List<Game
      Object>();
15.
16.     public List<float> cooldown = new List<float>
      ();
17.     public List<bool> cd = new List<bool>();
18.     public AudioManager soundFX;
19.     bool unknown = false;
20.     public GameObject magicCircle;
21.
22.     private GameObject spellToSpawn;
23.
24.     void Update()
25.     {
26.
27.         spellWord = resultDemo.resultSpell;
28.         if (spellWord != null)
29.         {
30.             switch (spellWord)
31.             {
32.                 case "fireball":
33.                     Debug.Log(spellWord);
34.                     spellToSpawn = vfx[0];
35.                     castPoint = fp[1];
36.
37.                     if (!cd[0])
38.                     {
39.                         SpawnSpell(castPoint);
40.                         cd[0] = true;
41.                         block[0].SetActive(true);
42.
43.                         cooldown[0] = 3f;
44.                     }
45.                     else
46.                     {
47.                         Debug.LogWarning("Cooldown
      n");
      }

```

```
48.         break;
49.         case "lightning bolt":
50.             Debug.Log(spellWord);
51.             spellToSpawn = vfx[1];
52.             castPoint = fp[1];
53.
54.             if (!cd[1])
55.             {
56.                 SpawnSpell(castPoint);
57.                 cd[1] = true;
58.                 block[1].SetActive(true);
59.
60.                 cooldown[1] = 3f;
61.             }
62.             else
63.             {
64.                 Debug.LogWarning("Cooldown
65. n");
66.             }
67.             break;
68.         case "heal":
69.             Debug.Log(spellWord);
70.             spellToSpawn = vfx[2];
71.             castPoint = fp[1];
72.
73.             if (!cd[2])
74.             {
75.                 SpawnSpell(castPoint);
76.                 cd[2] = true;
77.                 block[2].SetActive(true);
78.
79.                 cooldown[2] = 25f;
80.             }
81.             else
82.             {
83.                 Debug.LogWarning("Cooldown
84. n");
85.             }
86.             break;
87.         case "golem":
88.             Debug.Log(spellWord);
89.             spellToSpawn = vfx[3];
```

```
86.         castPoint = fp[0];
87.
88.         if (!cd[3])
89.         {
90.             SpawnSpell(castPoint);
91.             cd[3] = true;
92.             block[3].SetActive(true);
93.
94.             cooldown[3] = 8f;
95.         }
96.         else
97.         {
98.             Debug.LogWarning("Cooldown
99. n");
100.        }
101.        break;
102.        case "dragon":
103.            Debug.Log(spellWord);
104.
105.            spellToSpawn = vfx[4];
106.            castPoint = fp[0];
107.            if (!cd[4])
108.            {
109.                SpawnSpell(castPoint);
110.                cd[4] = true;
111.                block[4].SetActive
112. (true);
113.                cooldown[4] = 10f;
114.            }
115.            else
116.            {
117.                Debug.LogWarning("
118. Cooldown");
119.            }
120.            break;
121.            case "meteor":
122.                Debug.Log(spellWord);
```



```
119.                spellToSpawn = vfx[5];
120.                castPoint = fp[0];
121.
122.                if (!cd[5])
123.                {
124.                    SpawnSpell(castPoint);
125.                    cd[5] = true;
126.                    block[5].SetActive(true);
127.                    cooldown[5] = 10f;
128.                }
129.                else
130.                {
131.                    Debug.LogWarning("Cooldown");
132.                }
133.                break;
134.                default:
135.                    //unknown = true;
136.                    break;
137.            }
138.        }
139.        cooldown[0] -= Time.deltaTime;
140.        cooldown[1] -= Time.deltaTime;
141.        cooldown[2] -= Time.deltaTime;
142.        cooldown[3] -= Time.deltaTime;
143.        cooldown[4] -= Time.deltaTime;
144.        cooldown[5] -= Time.deltaTime;
145.
146.        if (cooldown[0] <= 0)
147.        {
148.            cd[0] = false;
149.            block[0].SetActive(false);
150.        }
151.        if (cooldown[1] <= 0)
152.        {
153.            cd[1] = false;
154.            block[1].SetActive(false);
155.        }
```

```
156.         if (cooldown[2] <= 0)
157.         {
158.             cd[2] = false;
159.             block[2].SetActive(false);
160.         }
161.         if (cooldown[3] <= 0)
162.         {
163.             cd[3] = false;
164.             block[3].SetActive(false);
165.         }
166.         if (cooldown[4] <= 0)
167.         {
168.             cd[4] = false;
169.             block[4].SetActive(false);
170.         }
171.         if (cooldown[5] <= 0)
172.         {
173.             cd[5] = false;
174.             block[5].SetActive(false);
175.         }
176.         resultDemo.resultSpell = null;
177.     }
178.
179.     void SpawnSpell(GameObject firePoint)
180.     {
181.         GameObject vfx;
182.         soundFX.Casting();
183.         StartCoroutine(MagicCircle(0.6f));
184.
185.         if (spellWord != null)
186.         {
187.             Debug.Log("Cast Spell " + spel
188.             lWord);
189.             vfx = Instantiate(spellToSpawn
190.             , firePoint.transform.position, firePoint.transfo
191.             rm.rotation);
192.             resultDemo.resultSpell = null;
193.         }
194.     }
195.     else
```

```

192.         {
193.             Debug.Log("On Cooldown");
194.         }
195.     }
196.
197.     public IEnumerator MagicCircle(float duration)
198.     {
199.         float elapsed = 0.0f;
200.
201.         while (elapsed < duration)
202.         {
203.             magicCircle.SetActive(true);
204.
205.             yield return null;
206.         }
207.         magicCircle.SetActive(false);
208.     }
209. }

```

Kode Sumber 4.6 Implementasi Cara Bermain (3)

Pada Kode Sumber 4.6 akan diproses pada sebuah variabel yang nantinya akan dicocokkan dengan menggunakan statement *Switch Case* lalu *string* yang cocok akan dipanggil statement di dalamnya yang memiliki variabel *spell name*, *spell cooldown*, dan *spell firepoint*. Selanjutnya ketiga variabel tersebut akan digunakan pada fungsi **SpawnSpell()** yang kemudian akan memproses *instantiate object* menggunakan variabel sebelumnya. *Spell* akan dipanggil berdasarkan *spell name* dan diluncurkan pada *spell firepoint* lalu *spell* tersebut akan dalam keadaan *cooldown*.

4.2.5. Implementasi Spell Damage

Setiap *spell* yang dikeluarkan dapat memberikan *damage* pada musuh (kecuali *heal*). Pada Kode Sumber 4.7 dijelaskan bagaimana setiap *spell* dapat memberikan *damage* pada musuh.

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class FireballDamage : MonoBehaviour
6. {
7.
8.     GameObject spell;
9.     GameObject par;
10.
11.     void Start()
12.     {
13.         par = transform.parent.gameObject;
14.         spell = par.transform.parent.gameObject;
15.     }
16.
17.     void Update()
18.     {
19.         Destroy(spell, 2f);
20.     }
21.
22.     private void OnTriggerEnter(Collider other)
23.     {
24.         Debug.Log(other.gameObject.ToString());
25.         if (other.gameObject.layer == 9)
26.         {
27.             other.gameObject.GetComponent<EnemyHealth>().InstantDamage2();
28.             Destroy(spell, 2f);
29.         }
30.
31.
32.     }
33. }
```

Kode Sumber 4.7 Implementasi Spell Damage

Pada kode diatas digunakan fungsi **OnTriggerEnter()** yang bekerja apabila *collider* dari *spell* bersentuhan dengan objek musuh. Ketika *collider* dari *spell* bersentuhan dengan objek yang

memiliki tag “Enemy”, fungsi pada *script* akan dipanggil dan menjalankan statement yang ditentukan. Fungsi ini akan memanggil fungsi lain yang ada pada *script* lainnya yang dipasang pada object musuh. Fungsi ini akan memanggil fungsi lain pada object musuh yang dapat memberikan *damage* pada *Health Point* musuh. Setelah *spell* bersentuhan dengan objek atau telah melewati batas waktu tertentu setelah peluncuran, objek *spell* akan hancur untuk menghilangkan objek tersebut pada hiarki sistem.

4.2.6. Implementasi Enemy Attack

Pada bagian ini akan dijelaskan mengenai Kode Sumber 4.8 yang dibuat agar musuh dapat melakukan serangan pada pemain ketika pemain memasuki jarak tertentu dari musuh.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class GoblinAnimation : MonoBehaviour
6. {
7.
8.     public Animator goblinAnimator;
9.     public EnemyHealth goblinHealth;
10.    public bool dead = false;
11.    public List<AudioClip> sfx = new List<AudioCl
    ip>();
12.    public AudioSource sounds;
13.
14.    public Transform Player;
15.    public Transform head;
16.    public bool chasing;
17.    public bool attack;
18.    private bool y = true;
19.    Rigidbody rb;
20.    bool takedamage = false;
21.
22.    void Start()
23.    {

```

```

24.     goblinAnimator = GetComponent<Animator>()
    ;
25.     rb = GetComponent<Rigidbody>();
26.     rb.freezeRotation = true;
27.     attack = false;
28.     chasing = false;
29.     if (Player == null)
30.     {
31.         Player = GameObject.FindGameObjectWith
    hTag("Player").GetComponent<Transform>() as Transform;
32.     }
33. }
34.
35. void Update()
36. {
37.
38.     Vector3 direction = Player.position - this
    s.transform.position;
39.     direction.y = 0;
40.
41.     float temps = Vector3.Distance(Player.position,
    this.transform.position);
42.     if (temps > 25)
43.         attack = false;
44.
45.     else if (temps < 20)
46.     {
47.         chasing = true;
48.         goblinAnimator.SetBool("idle", false)
    ;
49.         this.transform.rotation = Quaternion.
    Slerp(this.transform.rotation, Quaternion.LookRotation
    (direction), 0.1f);
50.
51.         if (direction.magnitude > 2)
52.         {
53.             rb.isKinematic = false;
54.             this.transform.position += transform
    .forward * 1 * Time.deltaTime;
55.             goblinAnimator.SetBool("walk", true);

```

```

56.         goblinAnimator.SetBool("attack",
57.         false);
58.     }
59.     else
60.     {
61.         rb.isKinematic = true;
62.         goblinAnimator.SetBool("walk", fa
63.         lse);
64.         goblinAnimator.SetBool("attack",
65.         true);
66.     }
67.     }
68.     else if (attack == true)
69.     {
70.         chasing = true;
71.         goblinAnimator.SetBool("idle", false)
72.         ;
73.         this.transform.rotation = Quaternion.
74.         Slerp(this.transform.rotation, Quaternion.LookROT
75.         ation(direction), 0.1f);
76.     }
77.     if (direction.magnitude > 2)
78.     {
79.         rb.isKinematic = false;
80.         this.transform.position += transf
81.         orm.forward * 1 * Time.deltaTime;
82.         goblinAnimator.SetBool("walk", tr
83.         ue);
84.         goblinAnimator.SetBool("attack",
85.         false);
86.     }
87.     else
88.     {
89.         rb.isKinematic = true;
90.         goblinAnimator.SetBool("walk", fa
91.         lse);
92.         goblinAnimator.SetBool("attack",
93.         true);
94.     }
95.     }
96.     else

```

```

87.     {
88.         goblinAnimator.SetBool("idle", true);
89.         goblinAnimator.SetBool("walk", false)
90.     ;
91.         goblinAnimator.SetBool("attack", false);
92.         chasing = false;
93.     }
94. }

```

Kode Sumber 4.8 Implementasi Enemy Attack

Pada kode diatas dijelaskan tentang bagaimana musuh dapat menyerang pemain ketika pemain memasuki jarak serang dari musuh tersebut. Pertama-tama kode mencari objek pemain dengan *tag* "Player" lalu menghitung jarak antara objek pemain dan musuh tersebut. Pada posisi awal, musuh dalam keadaan diam dan tidak menargetkan pemain. Ketika pemain memasuki jarak kurang dari 20 m dengan musuh, fungsi pada musuh akan dipanggil dan membuat musuh berbelok menghadap pemain lalu musuh akan mulai berjalan mendekati pemain diiringi dengan memainkan animasi berjalan milik musuh. Jika pemain menjauh dari jarak serang musuh, maka musuh akan kembali pada posisi diam dan tidak menarget pemain. Apabila pemain melakukan serangan pada jarak kurang dari 25 m dan melakukan *damage* pada musuh, maka musuh akan berubah pada kondisi menyerang dan mulai menargetkan pemain hingga pemain keluar dari jarak 25m lebih.

4.2.7. Implementasi Dragon (Boss) Attack

Pada bagian ini dijelaskan mengenai sistem menyerang dari *Dragon (Boss)* dari permainan ini. Sebelum *Dragon* memasuki mode *battle*, *Dragon* akan berada pada keadaan *idle*.

Sebelum sampai di area *Dragon*, dibuat sebuah *trigger* yang akan mengaktifkan *battle mode* pada *Dragon* ketika objek pemain melewati atau menyentuh *trigger* tersebut.

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class WyvernAnimation : MonoBehaviour
6. {
7.     public GameObject fireball;
8.     public GameObject firePoint;
9.     public Animator wyvernAnimator;
10.    public EnemyHealth wyvernHealth;
11.    public bool dead = false;
12.    public List<AudioClip> sfx = new List<AudioCl
    ip>();
13.    public AudioSource sounds;
14.
15.    public bool battled = false;
16.    GameObject target;
17.    public float dist;
18.    public Transform locate;
19.    float count = 10f;
20.    float handicap = 8f;
21.    public EnemyHealth Life;
22.
23.    void Start()
24.    {
25.        wyvernAnimator = GetComponent<Animator>()
;
26.        sounds.clip = sfx[0];
27.        sounds.Play();
28.        sounds.loop = true;
29.    }
30.
31.    void Update()
32.    {
33.
34.        if (wyvernHealth.damaged == true)
35.        {
```

```
36.         InvokeRepeating("damaged", 0.5f, 1f);
37.     }
38.
39.     if (wyvernHealth.damaged == false)
40.     {
41.         wyvernAnimator.SetBool("damaged", false);
42.         CancelInvoke();
43.     }
44.
45.     if (wyvernHealth.die == true)
46.     {
47.         sounds.clip = sfx[2];
48.         battled = false;
49.         die();
50.     }
51.
52.     if (battled == true)
53.     {
54.         findTarget();
55.         Debug.Log(count);
56.
57.         if (Life.cur_health <= 50)
58.         {
59.             handicap = 3.5f;
60.         }
61.         else if (Life.cur_health <= 150)
62.         {
63.             handicap = 6f;
64.         }
65.         if (count <= handicap)
66.         {
67.             wyvernAnimator.SetBool("fireball"
68. , false);
69.         }
70.         if (count <= 2)
71.         {
72.             wyvernAnimator.SetBool("fireball"
73. , true);
74.         }
75.         if (count <= 0)
```

```
74.         {
75.             attackPhase();
76.         }
77.
78.         count -= Time.deltaTime;
79.     }
80. }
81.
82. public void findTarget()
83. {
84.     GameObject Player = GameObject.FindGameOb
85.     jectWithTag("Player");
86.     if (Player == null)
87.     {
88.         Debug.Log("Player Died");
89.         return;
90.     }
91.     Vector3 direction = Player.transform.posi
92.     tion - this.transform.position;
93.     direction.y = 0;
94.     this.transform.rotation = Quaternion.Sler
95.     p(this.transform.rotation, Quaternion.LookRotatio
96.     n(direction), 0.1f);
97. }
98.
99. public void attackPhase()
100. {
101.     GameObject shooting;
102.     shooting = Instantiate(fireball, firePoint.transform.position, firePoint.transform.
103.     rotation);
104.     count = handicap;
105. }
106.
107. public void die()
108. {
109.     dead = true;
110.     var rb = GetComponent<Rigidbody>()
111.     ;
112.     rb.isKinematic = true;
```

```

109.         var col = GetComponent<Collider>()
        ;
110.         wyvernAnimator.SetBool("dead", true);
111.     }
112.
113.     public void BattleStart()
114.     {
115.         wyvernAnimator.SetBool("battle", true);
116.         wyvernAnimator.SetBool("idle", true);
117.         battled = true;
118.         var rb = GetComponent<Rigidbody>()
        ;
119.         rb.useGravity = true;
120.         rb.isKinematic = false;
121.     }
122. }

```

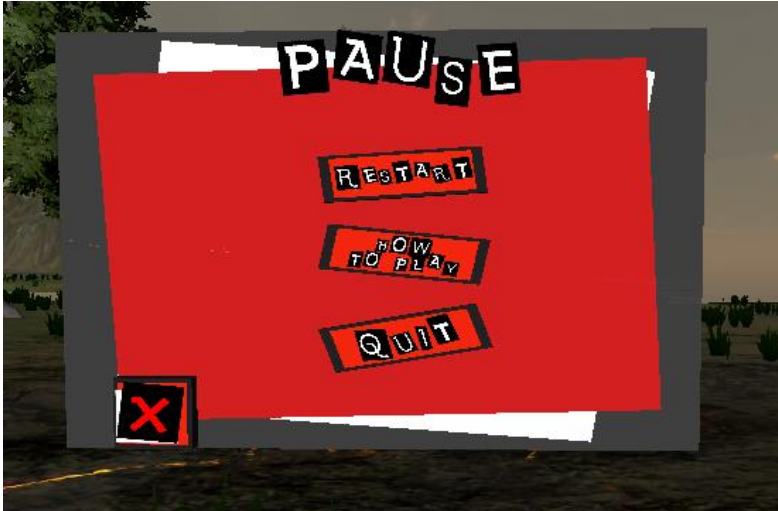
Kode Sumber 4.9 Implementasi Dragon (Boss) Attack

Pertama-tama kode akan memeriksa apakah *Dragon* dalam keadaan *battle mode*. Jika *battle mode* telah diaktifkan, kode akan mencari objek pemain dan membuat *Dragon* menarget pemain sepanjang pertarungan.

Dragon dapat mengeluarkan bola api setiap 10 detik. Bola api yang dikeluarkan akan mengurangi *Health Point* pemain apabila itu mengenai karakter pemain. Selama pertarungan akan ada peningkatan *handicap* atau kesulitan seiring dengan berkurangnya *Health Point* dari *Dragon*. Pada keadaan *Health Point* diatas 50% *Dragon* akan mengeluarkan bola api setiap 10 detik, mengeluarkan bola api setiap 6 detik saat *Health Point* mencapai 50%, dan Mengeluarkan bola api setiap 3 detik saat *Health Point* mencapai 10%.

4.2.8. Implementasi Tampilan Pause Menu

Pada Gambar 4.6 ditampilkan hasil implementasi dari rancangan *Pause Menu*.



Gambar 4.6 Implementasi Rancangan Pause Menu

Tampilan Pause Menu akan muncul jika pemain menekan tombol Pause pada HUD. Berikut penjelasan dari konten Pause Menu:

1. Tombol **Restart**, berfungsi untuk memulai permainan kembali dari awal.
2. Tombol **How to Play**, berfungsi untuk menampilkan panel berupa tampilan cara bermain dan beberapa informasi mengenai permainan beserta daftar *spell* yang dapat digunakan dan bagaimana cara memanggilya..
3. Tombol **Quit**, berfungsi untuk keluar dari permainan dan permainan akan tertutup.

4. Tombol **Close [X]**, berfungsi untuk menutup Menu Pause dan melanjutkan permainan.

Pada Kode Sumber 4.10 dijelaskan mengenai fungsi yang digunakan pada tombol-tombol di *Pause Menu*.

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.SceneManagement;
5.
6. public class GameManagement : MonoBehaviour
7. {
8.     public GameObject pauseMenuUI;
9.     public GameObject howToPlayUI;
10.    public GameObject pauseButton;
11.
12.    public Demo demo;
13.    public OVRPlayerController move;
14.    private void Start()
15.    {
16.        Time.timeScale = 1f;
17.    }
18.
19.    public void Restart()
20.    {
21.        demo.enabled = true;
22.        move.enabled = true;
23.        Debug.Log("Restart");
24.        Time.timeScale = 1f;
25.        SceneManager.LoadScene("MainGame");
26.    }
27.
28.    public void Quit()
29.    {
30.        Debug.Log("Quit Game");
31.        Application.Quit();
32.    }
33.
34.    public void HowToPlay()
35.    {
```

```
36.         Debug.Log("How To Play");
37.
38.         pauseMenuUI.SetActive(false);
39.         howToPlayUI.SetActive(true);
40.
41.     }
42.
43.     public void ClosePause()
44.     {
45.         Time.timeScale = 1f;
46.         demo.enabled = true;
47.         move.enabled = true;
48.         pauseButton.SetActive(true);
49.
50.         pauseMenuUI.SetActive(false);
51.     }
52. }
```

Kode Sumber 4.10 Implementasi Tampilan Pause Menu

4.2.9. Implementasi Tampilan Game Over

Pada bagian ini dijelaskan mengenai implementasi dari *Game Over* saat pemain menyelesaikan permainan atau mati karena *Health Point* pemain telah habis. Pada Gambar 4.7 ditunjukkan hasil implementasi rancangan tampilan *Game Over*.



Gambar 4.7 Implementasi Tampilan Game Over

Berikut penjelasan dari konten tampilan *Game Over* ketika permainan berakhir atau pemain kehabisan *Health Point*:

1. Tombol **Restart**, berfungsi untuk memulai permainan kembali dari awal.
2. Tombol **Quit**, berfungsi untuk keluar dari permainan dan permainan akan tertutup.

Saat permainan berakhir, pemain dapat memilih untuk memulai kembali permainan atau mengakhiri bermain dengan menutup permainan.

Pada Kode Sumber 4.11 dijelaskan mengenai fungsi yang digunakan untuk menampilkan tampilan *Game Over* dan fungsi setiap tombol pada tampilan tersebut.

```

1. using System.Collections;
2. using System.Collections.Generic;
  
```



```
3. using UnityEngine;
4. using UnityEngine.SceneManagement;
5.
6. public class GameManagement : MonoBehaviour
7. {
8.     public GameObject gameOverUI;
9.     public GameObject deadUI;
10.
11.     public PlayerHealth playerHealth;
12.     public EnemyHealth bossHealth;
13.     public Demo demo;
14.     public OVRPlayerController move;
15.     private void Start()
16.     {
17.         Time.timeScale = 1f;
18.     }
19.
20.     private void Update()
21.     {
22.         if (playerHealth.cur_health <= 0)
23.         {
24.             Dead();
25.
26.         }
27.
28.         if (bossHealth.cur_health <= 0)
29.         {
30.             GameOver();
31.         }
32.     }
33.     public void GameOver()
34.     {
35.         demo.enabled = false;
36.         move.enabled = false;
37.         Debug.Log("Game Over");
38.         pauseButton.SetActive(false);
39.         Time.timeScale = 0.3f;
40.
41.         gameOverUI.SetActive(true);
42.     }
43.
44.     public void Dead()
```

```

45.     {
46.         demo.enabled = false;
47.         move.enabled = false;
48.         Debug.Log("Die");
49.         pauseButton.SetActive(false);
50.         Time.timeScale = 0.3f;
51.
52.         deadUI.SetActive(true);
53.     }
54.
55.     public void Restart()
56.     {
57.         demo.enabled = true;
58.         move.enabled = true;
59.         Debug.Log("Restart");
60.         Time.timeScale = 1f;
61.         SceneManager.LoadScene("MainGame");
62.     }
63.
64.     public void Quit()
65.     {
66.         Debug.Log("Quit Game");
67.         Application.Quit();
68.     }
69. }

```

Kode Sumber 4.11 Implementasi Tampilan Game Over

Kode sumber diatas menjelaskan beberapa fungsi yang dipanggil untuk menampilkan tampilan *Game Over*. Tampilan *Game Over* akan ditampilkan apabila:

1. Pemain berhasil mengalahkan *Dragon (Boss)*.
2. Pemain kehabisan *Health Point*.

Pada tampilan *Game Over* ditampilkan dua buah tombol yang dapat dipilih. Pemain dapat memulai kembali permainan dengan menekan tombol “Restart” atau keluar dari permainan dan menutup permainan dengan menekan tombol “Quit”.

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas mengenai rangkaian uji coba dan evaluasi yang dilakukan terhadap Grand Wizard. Proses pengujian dilakukan dengan menggunakan metode *blackbox* berdasarkan skenario yang telah ditentukan.

5.1 Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan tugas akhir ini dilakukan pada lingkungan dan alat kaku dapat dilihat di Tabel 5.1. di bawah ini.

Tabel 5.1 Tabel Lingkungan Pengujian Sistem

Perangkat	Spesifikasi
Perangkat Keras	<ul style="list-style-type: none"> • Perangkat Pengembangan Sistem: Intel Core i7-7700 3.6GHz, NVIDIA GeForce GTX 1060 3GB, RAM 8GB • Oculus Rift • Leap Motion Controller
Perangkat Lunak	<ul style="list-style-type: none"> • Sistem Operasi: Microsoft Windows 10 64-bit • Perangkat Pengembang: Unity 2018.2.18f1 (64-bit) dan Visual Studio Community 2017 • Perangkat Pembantu: Microsoft Word 2017, Corel Draw X7, Adobe Photoshop CS6, Audacity, dan Blender

5.2 Pengujian Sistem

Pengujian fungsionalitas sistem dilakukan dengan menyiapkan sejumlah skenario sebagai tolok ukur keberhasilan

pengujian. Pengujian fungsionalitas dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi benar-benar diimplementasikan dan bekerja sebagaimana seharusnya. Pengujian fungsionalitas yang terdapat pada permainan dijabarkan sebagai berikut.

5.2.1 Uji Coba Menu Permainan

Pada sub bab ini dijelaskan secara detil mengenai skenario yang dilakukan dan hasil yang didapatkan dari pengujian fungsionalitas perangkat lunak yang dibangun pada halaman awal. Penjelasan disajikan dengan menampilkan kondisi awal, masukan, keluaran, hasil yang dicapai, dan kondisi akhir.

Pada menu permainan yang akan diuji adalah fungsionalitas tombol yang terdapat di menu utama, yaitu tombol *Play*, *How to Play*, dan *Quit*. Tampilan menu permainan dapat dilihat pada Gambar 4.1.

Pengujian dimulai ketika pengguna membuka permainan yang kemudian dihadapkan dengan halaman menu awal permainan. Skenario pengujian yang dilakukan dapat dilihat pada Tabel 5.2.

Tabel 5.2 Pengujian Halaman Menu Permainan

Kode Uji	UF-001
Deskripsi	Menguji antarmuka yang terdapat pada halaman <i>Main Menu</i>
Kondisi Awal	Pengguna berada pada posisi awal permainan
Prosedur Pengujian	Pengguna mengarahkan tangan ke salah satu tombol yang terdapat di <i>Main Menu</i> , kemudian menekan salah satu tombol tersebut
Skenario 1 – Memilih tombol <i>Play</i>	
Masukan	Menekan tombol Play pada <i>Main Menu</i> menggunakan tangan leap motion

Hasil yang Diharapkan	Sistem memulai permainan pada posisi awal bermain
Hasil yang Diperoleh	Sistem memulai permainan pada posisi awal bermain
Kesimpulan	Berhasil
<i>Skenario 2 – Memilih tombol How to Play</i>	
Masukan	Menekan tombol How to Play pada <i>Main Menu</i> menggunakan tangan leap motion
Hasil yang Diharapkan	Sistem menampilkan halaman <i>How to Play</i>
Hasil yang Diperoleh	Sistem menampilkan halaman <i>How to Play</i>
Kesimpulan	Berhasil
<i>Skenario 3 – Memilih tombol Quit</i>	
Masukan	Menekan tombol Quit pada <i>Main Menu</i> menggunakan tangan leap motion
Hasil yang Diharapkan	Sistem menutup permainan dan pemain keluar dari permainan
Hasil yang Diperoleh	Sistem menutup permainan dan pemain keluar dari permainan
Kesimpulan	Berhasil

5.2.2 Uji Coba Kontrol Permainan

Pada sub bab ini akan dijelaskan mengenai skenario pengujian kontrol pada permainan yang meliputi kontrol bergerak pemain dan kontrol serangan *spell* menggunakan Leap Motion.

Pengujian dimulai ketika permainan dimulai setelah pemain menekan tombol *Play* pada *Main Menu*. Pemain dapat bergerak menggunakan Touch Controller Oculus pada tangan kiri pemain

dan pemain dapat melancarkan serangan menggunakan tangan kanan pemain. Skenario pengujian yang dilakukan dapat dilihat pada Tabel 5.3.

Tabel 5.3 Pengujian Kontrol Permainan

Kode Uji	UF-002
Deskripsi	Menguji kontrol pada permainan
Kondisi Awal	Pengguna berada pada posisi awal bermain setelah pemain menekan tombol <i>Play</i>
Prosedur Pengujian	<ul style="list-style-type: none"> • Pemain bergerak menggunakan Touch Controller Oculus • Pemain menggambar di udara menggunakan jari telunjuk melalui Leap Motion Controller • Pemain membuka seluruh jari tangan untuk melancarkan serangan <i>spell</i>
<i>Skenario 1 – Bergerak menggunakan Touch Controller Oculus</i>	
Masukan	Menggerakkan analog pada Touch Controller Oculus
Hasil yang Diharapkan	<ul style="list-style-type: none"> • Pemain dapat bergerak maju dan mundur • Pemain dapat berputar ke kanan dan kiri
Hasil yang Diperoleh	<ul style="list-style-type: none"> • Pemain dapat bergerak maju dan mundur • Pemain dapat berputar ke kanan dan kiri
Kesimpulan	Berhasil
<i>Skenario 2 – Menggambar di udara menggunakan Jari Telunjuk</i>	
Masukan	Pemain menggunakan jari telunjuknya untuk menggambar di udara
Hasil yang Diharapkan	Muncul <i>Line Renderer</i> sebagai bentuk yang digambar oleh pemain

Hasil yang Diperoleh	Muncul <i>Line Renderer</i> sebagai bentuk yang digambar oleh pemain
Kesimpulan	Berhasil
<i>Skenario 3 – Membuka seluruh jari untuk melancarkan serangan spell</i>	
Masukan	Pemain membuka seluruh jari tangannya untuk mengkonfirmasi bentuk gambar yang dibuat dan untuk melancarkan serangan <i>spell</i>
Hasil yang Diharapkan	Muncul <i>spell</i> tertentu sesuai dengan gambar yang dibuat oleh pemain
Hasil yang Diperoleh	Muncul <i>spell</i> tertentu sesuai dengan gambar yang dibuat oleh pemain
Kesimpulan	Berhasil

5.2.3 Uji Coba Tampilan Pasue Menu

Pada suba bab ini akan dijelaskan mengenai skenario pengujian pada tampilan *Pause Menu*. Pemain dapat menjeda permainan dengan menekan tombol *Pause* pada pojok kiri atas layar tampilan.

Pengujian dilakukan saat pemain berada di tengah permainan dan dapat menekan tombol *Pause*. Skenario pengujian dapat dilihat pada Tabel 5.4.

Tabel 5.4 Pengujian Tampilan *Pause Menu*

Kode Uji	UF-003
Deskripsi	Menguji tampilan <i>Pause Menu</i>
Kondisi Awal	Pengguna berada pada posisi tengah bermain
Prosedur Pengujian	<ul style="list-style-type: none"> • Pemain menekan tombol Pause • Pemain menekan tombol Restart

	<ul style="list-style-type: none"> • Pemain menekan tombol How to Play • Pemain menekan tombol Back • Pemain menekan tombol Quit
Skenario 1 – Menekan tombol <i>Pause</i>	
Masukan	Menekan tombol Pause pada <i>HUD</i> menggunakan tangan leap motion
Hasil yang Diharapkan	<ul style="list-style-type: none"> • Permainan dijeda • Pemain tidak bisa bergerak • Pemain tidak bisa menggambar
Hasil yang Diperoleh	<ul style="list-style-type: none"> • Permainan dijeda • Pemain tidak bisa bergerak • Pemain tidak bisa menggambar
Kesimpulan	Berhasil
Skenario 2 – Menekan tombol <i>Restart</i>	
Masukan	Menekan tombol Restart pada <i>Pause Menu</i> menggunakan tangan leap motion
Hasil yang Diharapkan	Permainan akan dimulai kembali pada posisi awal bermain
Hasil yang Diperoleh	Permainan akan dimulai kembali pada posisi awal bermain
Kesimpulan	Berhasil
Skenario 3 – Menekan tombol <i>How to Play</i>	
Masukan	Menekan tombol How to Play pada <i>Pause Menu</i> menggunakan tangan leap motion
Hasil yang Diharapkan	Sistem menampilkan tampilan <i>How to Play</i>
Hasil yang Diperoleh	Sistem menampilkan tampilan <i>How to Play</i>
Kesimpulan	Berhasil
Skenario 4 – Menekan tombol <i>Back</i>	

Masukan	Menekan tombol Back pada halaman <i>How to Play</i> menggunakan tangan leap motion
Hasil yang Diharapkan	Sistem menutup tampilan <i>How to Play</i> dan kembali ke tampilan <i>Pause Menu</i>
Hasil yang Diperoleh	Sistem menutup tampilan <i>How to Play</i> dan kembali ke tampilan <i>Pause Menu</i>
Kesimpulan	Berhasil
Skenario 5 – Memilih tombol <i>Quit</i>	
Masukan	Menekan tombol Quit pada <i>Pause Menu</i> menggunakan tangan leap motion
Hasil yang Diharapkan	Sistem menutup permainan dan pemain keluar dari permainan
Hasil yang Diperoleh	Sistem menutup permainan dan pemain keluar dari permainan
Kesimpulan	Berhasil

5.2.4 Uji Coba Akhir Permainan

Pada sub bab ini dijelaskan mengenai skenario penjelasan pada tampilan akhir permainan atau *Game Over*. Tampilan *Game Over* akan muncul ketika pemain berhasil mengalahkan *Dragon (Boss)* atau *Health Point* dari pemain telah habis. Pada tampilan ini tersedia dua buah tombol berupa tombol *Restart* dan tombol *Quit*.

Pemain dapat memulai kembali permainan dengan menekan tombol *Restart* atau pemain dapat keluar dan menutup permainan dengan menekan tombol *Quit*. Skenario pengujian dapat dilihat pada Tabel 5.5.

Tabel 5.5 Pengujian Tampilan Akhir Permainan

Kode Uji	UF-004
Deskripsi	Menguji antarmuka yang terdapat pada halaman akhir permainan atau <i>Game Over</i>

Kondisi Awal	Pemain berada pada akhir permainan dimana pemain telah mengalahkan <i>Dragon (Boss)</i> atau <i>Health Point</i> pemain telah habis.
Prosedur Pengujian	<ul style="list-style-type: none"> • Pemain menekan tombol Restart • Pemain menekan tombol Quit
Skenario 1 – Menekan tombol <i>Restart</i>	
Masukan	Menekan tombol Restart pada <i>Game Over Menu</i> menggunakan tangan leap motion
Hasil yang Diharapkan	Permainan akan dimulai kembali pada posisi awal bermain
Hasil yang Diperoleh	Permainan akan dimulai kembali pada posisi awal bermain
Kesimpulan	Berhasil
Skenario 2 – Memilih tombol <i>Quit</i>	
Masukan	Menekan tombol Quit pada <i>Game Over Menu</i> menggunakan tangan leap motion
Hasil yang Diharapkan	Sistem menutup permainan dan pemain keluar dari permainan
Hasil yang Diperoleh	Sistem menutup permainan dan pemain keluar dari permainan
Kesimpulan	Berhasil

5.2.5 Hasil Uji Coba

Pada bagian ini dijelaskan mengenai hasil evaluasi dari pengujian yang dilakukan pada permainan. Hasil evaluasi dapat dilihat pada Tabel 5.6.

Kode	Deskripsi	Kemungkinan / Skenario	Perilaku Terlaksana
UF-001	Uji Coba Menu Permainan	Skenario 1	Ya
		Skenario 2	Ya
		Skenario 3	Ya
UF-002	Uji Coba Kontrol Permainan	Skenario 1	Ya
		Skenario 2	Ya
		Skenario 3	Ya
UF-003	Uji Coba Tampilan Pause Menu	Skenario 1	Ya
		Skenario 2	Ya
		Skenario 3	Ya
		Skenario 4	Ya
		Skenario 5	Ya
UF-004	Uji Coba Tampilan Game Over	Skenario 1	Ya
		Skenario 2	Ya

Tabel 5.6 Hasil Evaluasi Uji Coba

5.3 Pengujian Subjektivitas Sistem

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

5.3.1 Skenario Pengujian Pengguna

Dalam melakukan pengujian permainan, pengguna diminta mencoba memainkan permainan untuk mencoba semua fungsionalitas dan fitur yang ada. Pengujian permainan oleh pengguna dilakukan dengan sebelumnya memberikan informasi seputar permainan, kegunaan, dan fitur-fitur yang dimiliki. Setelah

informasi tersampaikan, pengguna kemudian diarahkan untuk langsung mencoba permainan.

Jumlah pengguna yang terlibat dalam pengujian perangkat sebanyak 10 orang.

Dalam memberikan penilaian dan tanggapan, pengguna diberikan kuesioner pengujian permainan. Kuesioner pengujian ini dilakukan dengan mengisi form kuesioner yang telah diberikan. Kuesioner pengujian ini memiliki beberapa aspek penilaian seputar desain antarmuka, *immersivity*, dan tingkat kenyamanan permainan. Nilai yang diberikan rentang nilai 1 hingga 5 dengan rincian pada Tabel 5.7. pada bagian akhir terdapat saran untuk perbaikan fitur. Detil kuesioner pengguna dapat dilihat pada Tabel 5.8 dan 5.9.

Tabel 5.7 Rentang Nilai

No	Keterangan	Nilai
1	Sangat Tidak Setuju (STS)	1
2	Tidak Setuju (TS)	2
3	Netral (N)	3
4	Setuju (S)	4
5	Sangat Setuju (SS)	5

Tabel 5.8 Pertanyaan Kuesioner Mengenai Karakteristik Pengguna

No	Pertanyaan Karakteristik Pengguna
1	Pernahkah anda memainkan sebuah permainan digital atau <i>game</i> di Komputer ?
2	Pernahkah anda memainkan game dengan genre Action ?
3	Pernahkah anda memainkan permainan atau game berbasis realitas virtual ?
4	Pernahkah anda memainkan permainan atau game berbasis realitas virtual menggunakan Oculus Rift ?

Tabel 5.9 Pernyataan Kuesioner Penilaian Pengguna

Pertanyaan Karakteristik Pengguna	
Parameter Antarmuka	
1	Permainan Grand Wizard memiliki tampilan, warna, dan desain antarmuka yang menarik.
2	Permainan Grand Wizard memiliki tata letak tombol, instruksi, dan informasi lainnya yang mudah dilihat / dikenali.
Parameter Immersive	
3	Saya merasakan sensasi layaknya penyihir dalam permainan.
4	Permainan ini dapat melatih kemampuan berpikir saya dalam mengambil keputusan dalam permainan ini.
5	Saya merasa belum pernah memainkan permainan genre action dengan gameplay yang seperti ini.
Parameter Kenyamanan	
6	Aplikasi dapat berjalan dengan lancar tanpa adanya lag dan/atau crash.
7	Saya merasa terbantu dengan adanya petunjuk yang disediakan permainan.
8	Saya merasa nyaman selama memainkan permainan ini.
9	Saya merasa tertarik untuk memainkan permainan ini untuk selanjutnya.

5.3.2 Daftar Penguji Permainan

Sub bab ini menunjukkan daftar pengguna yang bertindak sebagai penguji coba Grand Wizard. Dalam pengujian ini tidak terdapat kriteria atau batas usia yang harus dimiliki pengguna, karena aplikasi ini ditujukan kepada berbagai kalangan/siapa pun tanpa batasan umur. Daftar nama penguji aplikasi ini dapat dilihat pada Tabel 5.10.

Tabel 5.10 Daftar Penguji

No	Nama	Pekerjaan	Usia
1	Nur Muhammad Husnul Habib Yahya	Mahasiswa	20
2	Rafi R. Ramadhan	Mahasiswa	22
3	Yuga Mitra Hakiki	Mahasiswa	22
4	Anargya Widyadhana	Mahasiswa	20
5	Kevin Ashil F.	Mahasiswa	21
6	Ahmad Yahya A. A.	Mahasiswa	19
7	Rezky Alamsyah	Mahasiswa	21
8	Naufal Pranasetyo	Mahasiswa	21
9	Unggul Widodo W.	Mahasiswa	21
10	Wildan Ghiffarie Budhi	Mahasiswa	19

5.3.3 Hasil Pengujian Pengguna

Berdasarkan hasil kuesioner yang sudah diisi oleh penguji, maka yang pertama didapatkan ialah karakteristik pengguna, yaitu sebagai berikut:

- Mayoritas penguji pernah menggunakan aplikasi berbasis *virtual reality* (9/10). Hal ini menjadikan penguji lebih mudah dalam penggunaan aplikasi.
- Semua Penguji pernah bermain permainan digital pada komputer (10/10).
- Mayoritas (7/10) dari penguji pernah memainkan permainan *Virtual Reality* menggunakan Oculus Rift.

Kemudian pengujian terhadap aspek antarmuka, *immersivity*, dan tingkat kenyamanan pengguna mendapatkan hasil seperti yang ditunjukkan pada Tabel 5.11. Sistem penilaian yang digunakan yaitu dengan menjumlahkan seluruh nilai dari seluruh responden dengan kemudian dirata-rata dan dibagi dengan nilai maksimum (5), sehingga didapatkan persentase nilai dari setiap parameter yang diujikan. Hasil akhir dari pengujian setiap kategori/aspek dapat dilihat pada Tabel 5.12.

Tabel 5.11 Hasil Pengujian Pengguna

No	Pernyataan	Penilaian					Rata-rata
		1	2	3	4	5	
Parameter Antarmuka							
1	Permainan Grand Wizard memiliki tampilan, warna, dan desain antarmuka yang menarik.	0	0	0	4	6	4.6
2	Permainan Grand Wizard memiliki tata letak tombol, instruksi, dan informasi lainnya yang mudah dilihat / dikenali.	0	0	1	5	4	4.3
Parameter Immersive							
3	Saya merasakan sensasi layaknya penyihir dalam permainan.	0	0	0	3	7	4.7
4	Permainan ini dapat melatih kemampuan berpikir saya dalam mengambil keputusan dalam permainan ini.	0	0	3	5	2	3.9
5	Saya merasa belum pernah memainkan permainan genre action dengan gameplay yang seperti ini.	0	0	1	4	5	4.4
Parameter Kenyamanan							
6	Aplikasi dapat berjalan dengan lancar tanpa adanya lag dan/atau crash.	0	0	2	4	4	4.2
7	Saya merasa terbantu dengan adanya petunjuk	0	0	0	6	4	4.4

	yang disediakan permainan.						
8	Saya merasa nyaman selama memainkan permainan ini.	1	2	3	4	0	3.0
9	Saya merasa tertarik untuk memainkan permainan ini untuk selanjutnya.	0	0	3	5	2	3.9

Tabel 5.12 Hasil Akhir Pengujian Pengguna

No	Pernyataan	Rata-rata	Total	Total (%)
Parameter Antarmuka				
1	Permainan Grand Wizard memiliki tampilan, warna, dan desain antarmuka yang menarik.	4.6	4.45	89%
2	Permainan Grand Wizard memiliki tata letak tombol, instruksi, dan informasi lainnya yang mudah dilihat / dikenali.	4.3		
Parameter Immersivity				
3	Saya merasakan sensasi layaknya penyihir dalam permainan.	4.7	4.3	86.6%
4	Permainan ini dapat melatih kemampuan berpikir saya dalam mengambil keputusan dalam permainan ini.	3.9		
5	Saya merasa belum pernah memainkan permainan	4.4		

	genre action dengan gameplay yang seperti ini.			
Parameter Kenyamanan				
6	Aplikasi dapat berjalan dengan lancar tanpa adanya lag dan/atau crash.	4.2	3.87	77.5%
7	Saya merasa terbantu dengan adanya petunjuk yang disediakan permainan.	4.4		
8	Saya merasa nyaman selama memainkan permainan ini.	3.0		
9	Saya merasa tertarik untuk memainkan permainan ini untuk selanjutnya.	3.9		

5.3.4 Kritik dan Saran Pengguna

Selain memberikan penilaian dan tanggapan terhadap fungsionalitas aplikasi, pengguna juga memberikan tanggapan berupa kritik dan saran untuk perbaikan fitur aplikasi dalam pengembangan selanjutnya. Kritik dan saran pengguna dapat dilihat pada Tabel 5.13.

Tabel 5.13 Kritik dan Saran Pengguna

No	Nama	Kritik dan Saran
1	Nur Muhammad Husnul Habib Yahya	Pergerakan di-improve lagi.
2	Rafi R. Ramadhan	Pergerakan kurang smooth. Animasi kurang natural membuat pusing + mual.

3	Yuga Mitra Hakiki	Partikel saat di main menu kadang-kadang mengganggu, motion kamera yang tidak natural bikin pusing.
4	Anargya Widyadhana	Pergerakan di dalam game membuat agak pusing, mungkin karakter pemain agak terlalu kebawah, sehingga bisa dibuat lebih tinggi. Beberapa skill pemain agak susah digambar. Untuk tampilan, efek, gameplay, bagus, pemain seperti berada dalam permainan.
5	Kevin Ashil F.	Lebih natural dan lanjutkan
6	Ahmad Yahya A. A.	Mungkin diberi stage tambahan dan monster lain agar game lebih menarik. Juga diberi cerita agar lebih baik.
7	Rezky Alamsyah	Bisa mengarahkan view ke kanan dan ke kiri menggunakan button A dan Button B.
8	Naufal Pranasetyo	Diberi tambahan monster dan dapat disisipkan cerita intro maupun di tengah permainan. View dapat diubah menggunakan tombol.
9	Unggul Widodo W.	Motionnya sangat kuat, membuat pusing.
10	Wildan Ghiffarie Budhi	Gamenya keren

5.3.5 Evaluasi Pengujian

Sub bab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Dalam hal ini, evaluasi menunjukkan data rekapitulasi dari hasil pengujian fungsionalitas. Rekapitulasi disusun dalam bentuk Tabel yang dapat dilihat pada Tabel 5.14. Dari data yang terdapat pada Tabel tersebut, diketahui

bahwa aplikasi yang dibuat telah berjalan sesuai dengan skenario yang diharapkan.

Tabel 5.14 Tabel Hasil Evaluasi Pengujian

Kode	Deskripsi	Kemungkinan / Skenario	Perilaku Terlaksana
UF-001	Uji Coba Menu Permainan	Skenario 1	Ya
		Skenario 2	Ya
		Skenario 3	Ya
UF-002	Uji Coba Kontrol Permainan	Skenario 1	Ya
		Skenario 2	Ya
		Skenario 3	Ya
UF-003	Uji Coba Tampilan Pause Menu	Skenario 1	Ya
		Skenario 2	Ya
		Skenario 3	Ya
		Skenario 4	Ya
		Skenario 5	Ya
UF-004	Uji Coba Tampilan Game Over	Skenario 1	Ya
		Skenario 2	Ya

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Bab ini akan membahas mengenai kesimpulan yang diperoleh selama pengerjaan tugas akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian yang dilakukan, dapat diambil kesimpulan sebagai berikut.

1. Permainan dapat menampilkan tampilan FPS dalam lingkungan realitas virtual.
2. Permainan berhasil mengimplementasikan antara dua *device* Oculus Rift dan Leap Motion secara bersamaan.
3. Permainan dapat mendeteksi gambar yang dibuat oleh pemain menggunakan pendeteksi Point Cloud Recognizer.
4. Permainan berhasil dibuat dengan *Game Engine* Unity 3D.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi, dan pengujian yang telah dilakukan. Di antaranya adalah sebagai berikut:

1. Ditambahkan variasi dari tipe musuh yang ada pada permainan.
2. Ditambahkan cerita pada permainan agar terlihat lebih menarik.
3. Untuk pergerakan pemain dibuat lebih *smooth* lagi agar terlihat dan terasa lebih natural dan tidak membuat pusing.
4. Kontrol pergerakan pemain dibuat lebih mudah lagi terutama pada bagian rotasi dan mengganti *view*.

5. Ditambahkan *level stage* agar permainan lebih menantang.

DAFTAR PUSTAKA

- [1] K. G. D. Herlangga, "Virtual Reality dan Perkembangannya," CodePolitan, 7 Maret 2016. [Online]. Available: <https://www.codepolitan.com/virtual-reality-dan-perkembangannya>. [Accessed Desember 2017].
- [2] Unity, "Unity," Unity, [Online]. Available: <https://unity3d.com/unity>. [Diakses Desember 2017].
- [3] Leap Motion, "Leap Motion," Motion Control, [Online]. Available: <https://www.leapmotion.com/>. [Diakses 9 December 2017].
- [4] D. V. C. "PDollar Point-Cloud Gesture Recognize," [Online]. Available: <https://assetstore.unity.com/packages/tools/input-management/pdollar-point-cloud-gesture-recognizer-21660>. [Accessed 3 January 2019].
- [5] W. c. "First-person shooter," [Online]. Available: https://en.wikipedia.org/wiki/First-person_shooter. [Accessed 3 January 2019].
- [6] W. c. "Unity (game engine)," [Online]. Available: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)). [Accessed 3 January 2019].
- [7] W. c. "Virtual Reality," [Online]. Available: https://en.wikipedia.org/wiki/Virtual_reality. [Accessed 3 January 2019].
- [8] W. c. "Oculus Rift," [Online]. Available: https://en.wikipedia.org/wiki/Oculus_Rift. [Accessed 3 January 2019].
- [9] W. L. S. Firdaus Syawal, "Permainan Realitas Virtual Magus Menggunakan Teknologi Oculus Rift dengan Kendali Voice Recognition," *Jurnal Teknik ITS*, vol. 7, 2018.
- [10] K. Yuan, "Pinch Drawing with LeapMotion (Unity Modules were Deprecated)," 14 July 2017. [Online]. Available:

https://github.com/JKirkYuan/Leap_Pinch_Desktop.
[Accessed 4 January 2019].

- [11] R. D. Vatavu, "\$P Point-Cloud Recognizer," 2012. [Online].
Available:
<http://depts.washington.edu/ilab/proj/dollar/pdollar.html>.
[Accessed 4 January 2019].

LAMPIRAN



Lampiran 1 Penguji Coba (1)



Lampiran 2 Penguji Coba (2)



Lampiran 3 Penguji Coba (3)



Lampiran 4 Penguji Coba (4)



KUISIONER TUGAS AKHIR
08111562000173 – JUSAS-AHNI PRATIKA

PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCULUS RIFT DAN LEAP MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

Identitas Responden Surabaya, 26 Juni 2019

Nama Lengkap: Vedya Mira Hovite
 Pekerjaan: Melaksanakan
 Usia: 22 Tahun

A. KARAKTERISTIK RESPONDEN

Jawablah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

- Pernakah anda memainkan sebuah permainan digital atau game di Komputer?
 Pernah
 Tidak Pernah
- Pernakah anda memainkan game dengan genre Action?
 Pernah
 Tidak Pernah
- Pernakah anda memainkan permainan atau game berbasis realitas virtual?
 Pernah
 Tidak Pernah
- Pernakah anda memainkan permainan atau game berbasis realitas virtual menggunakan Oculus Rift?
 Pernah
 Tidak Pernah

B. PENILAIAN TERHADAP PERMAINAN

Isilah tabel dibawah ini dengan menggunakan tanda centang (✓)
 SS = Sangat Setuju S = Setuju N = Netral
 TS = Tidak Setuju STS = Sangat Tidak Setuju

Parameter Antarmuka	STS		TS		N		S		SS	
1 Permainan Grand Wizard memiliki tampilan warna dan desain antarmuka yang menarik.										✓
2 Permainan Grand Wizard memiliki cara letak tombol, instruksi, dan informasi lainnya yang mudah dilihat / dikenali.										✓
Parameter Immersivity	STS	TS	N	S	SS					
3 Saya merasakan sensasi layaknya berada dalam permainan.										✓
4 Permainan ini dapat melatih kemampuan berpikir saya dalam mengambil keputusan dalam permainan ini.										✓
5 Saya merasa belum pernah memainkan permainan genre action dengan gameplay yang seperti ini.									✓	
Parameter Kenyamanan	STS	TS	N	S	SS					
6 Aplikasi dapat berjalan dengan lancar tanpa adanya lag dan/atau crash.										✓
7 Saya merasa nyaman dengan adanya perintah yang disediakan permainan.										✓
8 Saya merasa nyaman selama memainkan permainan ini.										✓
9 Saya merasa tertarik untuk memainkan permainan ini untuk selanjutnya.										✓

C. KRITIK DAN SARAN

Kritis: Saya merasa bahwa tampilan permainan ini kurang menarik.
 Saran: Untuk meningkatkan tampilan permainan ini, sebaiknya tampilan permainan ini dibuat lebih menarik.



KUISIONER TUGAS AKHIR
0211540000313 - 0485/2019/PBT/PAKA

PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCULUS RIFT DAN LEAP MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

Identitas Responden
Surabaya, Juni 2019
Nama Lengkap Unggul Wulandito W
Pekerjaan Mahasiswa
Usia 21

A. KARAKTERISTIK RESPONDEN

Jawablah pertanyaan di bawah ini dengan menggunakan tanda centang (v)

- Pernahkah anda memainkan sebuah permainan digital atau game di Komputer?
 Pernah
 Tidak Pernah
- Pernahkah anda memainkan game dengan Action?
 Pernah
 Tidak Pernah
- Pernahkah anda memainkan permainan atau game berbasis realitas virtual?
 Pernah
 Tidak Pernah
- Pernahkah anda memainkan permainan atau game berbasis realitas virtual menggunakan Oculus Rift?
 Pernah
 Tidak Pernah

B. PENILAIAN TERHADAP PERMAINAN

Isilah tabel dibawah ini dengan menggunakan tanda centang (v)

SS = Sangat Setuju S = Setuju N = Netral

TS = Tidak Setuju STS = Sangat Tidak Setuju

Parameter Antarmuka	STS		TS		N		S		SS	
1 Permainan Grand Wizard memiliki tampilan, warna, dan desain antarmuka yang menarik.										<input checked="" type="checkbox"/>
2 Permainan Grand Wizard memiliki tata letak tombol, intruksi, dan informasi lainnya yang mudah dilihat / diartikan.										<input checked="" type="checkbox"/>
Parameter Immersibility	STS	TS	N	S	SS					
3 Saya merasakan sensasi layaknya berada dalam permainan.										<input checked="" type="checkbox"/>
4 Permainan ini dapat membuat ketegangan berpikir: saya dalam mengambil keputusan dalam permainan ini.										<input checked="" type="checkbox"/>
5 Saya merasa belum pernah memainkan permainan genre action dengan gameplay yang seperti ini.										<input checked="" type="checkbox"/>
Parameter Kenyamanan	STS	TS	N	S	SS					
6 Apakah dapat berjalan dengan lancar tanpa adanya lag dan/atau crash.										<input checked="" type="checkbox"/>
7 Saya merasa terbantu dengan adanya petunjuk yang disediakan permainan.										<input checked="" type="checkbox"/>
8 Saya merasa nyaman selama memainkan permainan ini.										<input checked="" type="checkbox"/>
9 Saya merasa tertarik untuk memainkan permainan ini untuk selanjutnya.										<input checked="" type="checkbox"/>

C. KRITIK DAN SARAN

Pertanyaan: Sebagaimana kritikan terhadap desain?

.....



KUISIONER TUGAS AKHIR
021154000173 - 0162 KIRI PRATIKA

PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCULUS RIFT DAN LEAP MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

Identitas Responden
 Nama Lengkap : Naufal Paneskyo
 Pekerjaan : Mahasiswa
 Usia : 21
 Surabaya, 27 Juni 2019
Naufal PF

A. KARAKTERISTIK RESPONDEN

Jawablah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

1. Pernahkah anda memainkan sebuah permainan digital atau game di Komputer?
 Pernah
 Tidak Pernah
2. Pernahkah anda memainkan game dengan genre Action?
 Pernah
 Tidak Pernah
3. Pernahkah anda memainkan permainan atau game berbasis realitas virtual?
 Pernah
 Tidak Pernah
4. Pernahkah anda memainkan permainan atau game berbasis realitas virtual menggunakan Oculus Rift?
 Pernah
 Tidak Pernah

B. PENILAIAN TERHADAP PERMAINAN

Isilah tabel dibawah ini dengan menggunakan tanda centang (✓)

SS = Sangat Setuju S = Setuju N = Netral

TS = Tidak Setuju STS = Sangat Tidak Setuju

Parameter Antarmuka	STS	TS	N	S	SS
1 Permainan Grand Wizard memiliki tampilan, warna, dan desain antarmuka yang menarik.					✓
2 Permainan Grand Wizard memiliki tata letak tombol, menu, dan informasi lainnya yang mudah dimiter / dibacai.				✓	
Parameter Immersivty	STS	TS	N	S	SS
3 Saya merasakan sensasi layaknya berada dalam permainan.					✓
4 Permainan ini dapat membuat ketertarikan berlebih saya dalam mengambil keputusan dalam permainan ini.					✓
5 Saya merasa belum pernah memainkan permainan genre action dengan gameplay yang seperti ini.					✓
Parameter Kenyamanan	STS	TS	N	S	SS
6 Adanya dapat berjalan dengan lancar tanpa adanya lag dan/atau crash.				✓	
7 Saya merasa terbantu dengan adanya petunjuk yang disediakan permainan.				✓	
8 Saya merasa nyaman selama memainkan permainan ini.				✓	
9 Saya merasa tertarik untuk memainkan permainan ini untuk selanjutnya.					✓

C. KRITIK DAN SARAN

Diperlu untuk memberikan master lain dan setiap sesi tetap Carita jitu, supaya di bagian permainan. View dapat selalu menggunakan tombol



KUISIONER TUGAS AKHIR
051154003173 - DAK NPM P011A04

PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCULUS Rift DAN LEAP
MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

Identitas Responden

Nama Lengkap : Rizky Amby Surabaya, 27 Juni 2019
Pekerjaan : Mahasiswa
Usia : 21 1814111

A. KARAKTERISTIK RESPONDEN

Jawablah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

- Pernahkah anda memainkan sebuah permainan digital atau game di Komputer?
 Pernah
 Tidak Pernah
- Pernahkah anda memainkan game dengan game Action?
 Pernah
 Tidak Pernah
- Pernahkah anda memainkan permainan atau game berbasis realitas virtual?
 Pernah
 Tidak Pernah
- Pernahkah anda memainkan permainan atau game berbasis realitas virtual menggunakan Oculus Rift?
 Pernah
 Tidak Pernah

B. PENILAIAN TERHADAP PERMAINAN

Isilah tabel dibawah ini dengan menggunakan tanda centang (✓)
SS = Sangat Setuju S = Setuju N = Netral
TS = Tidak Setuju STS = Sangat Tidak Setuju

Parameter Antarmuka	STS			TS			N			S			SS		
1 Permainan Grand Wizard memiliki tampilan, warna, dan desain antarmuka yang menarik.															✓
2 Permainan Grand Wizard memiliki tata letak tombol, ikon, dan informasi lainnya yang mudah dilihat / dibaca.												✓			
3 Saya merasakan sensasi layaknya berada dalam permainan.															✓
4 Permainan ini dapat melatih kemampuan berpikir saya dalam mengambil keputusan dalam permainan ini.															✓
5 Saya merasa belum pernah memainkan permainan genre action dengan gameplay yang seperti ini.															✓
Parameter Kenyamanan															✓
6 Apakah dapat berjalan dengan lancar tanpa adanya lag dan/atau crash.															✓
7 Saya merasa terbantu dengan adanya petunjuk yang disediakan permainan.															✓
8 Saya merasa nyaman selama memainkan permainan ini.															✓
9 Saya merasa tertarik untuk memainkan permainan ini untuk selanjutnya.															✓

C. KRITIK DAN SARAN

Bisa menggunakan platform VR selain Oculus Rift
Kembangkan ke bagian A dan bagian B



KUISIONER TUGAS AKHIR
03111540000313 - DASARAH PRATIYASA

PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCULUS RIFT DAN LEAP MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

Identitas Responden
 Nama Lengkap : Ahmad Zahya A. A.
 Pekerjaan : Mahasiswa
 Usia : 19
 Surabaya, 27 Juni 2019

A. KARAKTERISTIK RESPONDEN

Jawablah pertanyaan di bawah ini dengan menggunakan tanda centang (v)

1. Perkenalkan anda memainkan sebuah permainan digital atau game di Komputer?
 Pernah
 Tidak Pernah
2. Perkenalkan anda memainkan game dengan game Action?
 Pernah
 Tidak Pernah
3. Perkenalkan anda memainkan permainan atau game berbasis realitas virtual?
 Pernah
 Tidak Pernah
4. Perkenalkan anda memainkan permainan atau game berbasis realitas virtual menggunakan Oculus Rift?
 Pernah
 Tidak Pernah

B. PENILAIAN TERHADAP PERMAINAN

Isilah tabel dibawah ini dengan menggunakan tanda centang (v)
 S = Sangat Setuju S = Setuju N = Netral
 TS = Tidak Setuju TS = Sangat Tidak Setuju

Parameter Antarmuka	STS			TS			N			S			SS		
1 Permainan Grand Wizard memiliki tampilan yang menarik, dan desain antarmuka yang menarik.															
2 Permainan Grand Wizard memiliki tase latar belakang yang menarik dan ceritanya yang mudah dilisah / dikenali.															
Parameter Immersibility	STS			TS			N			S			SS		
3 Saya merasakan sensasi layaknya berada dalam permainan.															
4 Permainan ini dapat melatih kemampuan berpikir saya dalam mengambil keputusan dalam permainan ini.															
5 Saya merasa belum pernah memainkan permainan game action dengan gamesky yang seperti ini.															
Parameter Kenyamanan	STS			TS			N			S			SS		
6 Aplikasi dapat berjalan dengan lancar tanpa adanya lag dan/atau crash.															
7 Saya merasa nyaman dengan adanya petunjuk yang disediakan permainan.															
8 Saya merasa nyaman selama memainkan permainan ini.															
9 Saya merasa tertarik untuk memainkan permainan ini untuk selanjutnya.															

C. KRITIK DAN SARAN

Mungkin bisa stage tambahan dan monster lain agar game lebih menarik juga di beri cerita agar lebih baik



PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCELLUS RIFT DAN LEAP MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

Identitas Responden

Nama Lengkap : Nur Muhammad Husni Hidayat Alkyo Surabaya, 20 Juni 2019
 Pekerjaan : Notaris
 Usia : 20

A. KARAKTERISTIK RESPONDEN

Jawablah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

1. Pernahkah anda memainkan sebuah permainan digital atau game di Komputer?
 Pernah
 Tidak Pernah
2. Pernahkah anda memainkan game dengan genre Action?
 Pernah
 Tidak Pernah
3. Pernahkah anda memainkan permainan atau game berbasis realitas virtual?
 Pernah
 Tidak Pernah
4. Pernahkah anda memainkan permainan atau game berbasis realitas virtual menggunakan Oculus Rift?
 Pernah
 Tidak Pernah

B. PENILAIAN TERHADAP PERMAINAN
Isilah tabel dibawah ini dengan menggunakan tanda centang (✓)
 SS = Sangat Setuju S = Setuju
 N = Netral
 TS = Tidak Setuju STS = Sangat Tidak Setuju

Parameter Antarmuka	STS		TS		N		S		SS	
1. Permainan Grand Wizard memiliki tampilan, warna, dan desain antarmuka yang menarik.										✓
2. Permainan Grand Wizard memiliki tata letak tombol, instruksi, dan informasi lainnya yang mudah diakses / dikenali.										✓
Parameter Immersivity	STS		TS		N		S		SS	
3. Saya merasakan sensasi layaknya berada dalam permainan.										✓
4. Permainan ini dapat membuat kemampuan berpikir saya dalam mengambil keputusan dalam permainan ini.						✓				
5. Saya merasa belum pernah merasakan permainan genre action dengan gameplay yang seperti ini.										✓
Parameter Kenyamanan	STS		TS		N		S		SS	
6. Aplikasi dapat berjalan dengan lancar tanpa adanya lag atau/atau crash.										✓
7. Saya merasa terbantu dengan adanya petunjuk yang disediakan permainan.										✓
8. Saya merasa nyaman selama memainkan permainan ini.									✓	
9. Saya merasa tertarik untuk memainkan permainan ini untuk selanjutnya.										✓

C. KRITIK DAN SARAN
Perbaiki dan improve lagi



KUISIONER TUGAS AKHIR
02115540003173 – DUNEK ACHM PRATIKA

PERMAINAN REALITAS VIRTUAL GRAND WIZARD MENGGUNAKAN OCULUS RIFT DAN LEAP MOTION DENGAN KENDALI POINT CLOUD RECOGNITION

Identitas Responden
Nama lengkap : Prof. Dr. Damadhar
Pekerjaan : Proh. C.I.T.W.G.
Usia : 77 tahun
Surabaya, Juni 2019
[Signature]
Oct P. Damadhar

A. KARAKTERISTIK RESPONDEN

Jawablah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

1. Pernahkah anda memainkan sebuah permainan digital atau game di komputer?
 Pernah
 Tidak Pernah
2. Pernahkah anda memainkan game dengan genre Action?
 Pernah
 Tidak Pernah
3. Pernahkah anda memainkan permainan atau game berbasis realitas virtual?
 Pernah
 Tidak Pernah
4. Pernahkah anda memainkan permainan atau game berbasis realitas virtual menggunakan Oculus Rift?
 Pernah
 Tidak Pernah

B. PENILAIAN TERHADAP PERMAINAN

Isilah tabel dibawah ini dengan menggunakan tanda centang (✓)
SS = Sangat Setuju S = Setuju N = Netral
TS = Tidak Setuju STS = Sangat Tidak Setuju

Parameter Antarmuka	STS		TS		N		S		SS	
1. Permainan Grand Wizard memiliki tampilan yang menarik, dan desain antarmuka yang menarik.										✓
2. Permainan Grand Wizard memiliki tata letak yang menarik, dan antarmuka yang lainnya yang mudah dilihat / difahami.										✓
Parameter Immersivity										
3. Saya merasakan sensasi layaknya berada dalam permainan.										SS
4. Permainan ini dapat melatih kemampuan berpikir saya dalam mengambil keputusan dalam permainan ini.									✓	
5. Saya merasa belum pernah memainkan permainan genre action dengan gameplay yang seperti ini.									✓	
Parameter Kenyamanan										
6. Aplikasi dapat berjalan dengan lancar tanpa adanya lag atau error-crash.									✓	
7. Saya merasa terbantu dengan adanya petunjuk yang disediakan permainan.										✓
8. Saya merasa nyaman selama memainkan permainan ini.									✓	
9. Saya merasa tertarik untuk memainkan permainan ini untuk selanjutnya.									✓	

C. KRITIK DAN SARAN

Permainan... Sangat... Menarik...
Pilihlah... yang... menarik...
Pembuat... perlu...
Pembuat... perlu...
Pembuat... perlu...

BIODATA PENULIS



Dias Adhi Pratama, lahir di Serang pada tanggal 22 Oktober 1996. Penulis menempuh pendidikan mulai dari TK Arta Kencana Serang, SDN 3 Serang, SMPN 1 Serang, SMAN 2 Serang, dan saat ini melanjutkan studinya di Departemen informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember di Surabaya.

Selama Menempuh di dunia perkuliahan, penulis aktif mengikuti organisasi dan kepanitiaan di kampus antara lain sebagai anggota staff Departemen Dalam Negeri HMTK Kabinet inspirasi, staff ahli Dalam Negeri HMTK kabinet Kreasi, staff 3D Schematics 2015, staff ahli 3D Schematics 2016, panitia Dekalasi INTERVAL 2015, Admin Lab IGS 2nd *generation*, dan diamanahi sebagai Koordinator Admin Lab IGS.

Dalam menyelesaikan pendidikan sarjananya, penulis mengambil bidang minat Interaksi, Grafika, dan Seni (IGS). Komunikasi pada penulis dapat dihubungi melalui alamat *e-mail*: yaskeee22@gmail.com