



TUGAS AKHIR - KI141502

RANCANG BANGUN PEMBANGKIT WORLD DINAMIS PADA 3D VIRTUAL GAME 'THE LABYRINTH' MENGGUNAKAN ALGORITMA RECURSIVE DIVISION

PURBO PANAMBANG
NRP 5113100030

Dosen Pembimbing
Darlis Herumurti, S.Kom., M.Kom.
Imam Kuswardayan, S.Kom., M.T.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

**RANCANG BANGUN PEMBANGKIT WORLD
DINAMIS PADA 3D VIRTUAL GAME 'THE
LABYRINTH' MENGGUNAKAN ALGORITMA
RECURSIVE DIVISION**

**PURBO PANAMBANG
NRP 5113100030**

**Dosen Pembimbing
Darlis Herumurti, S.Kom., M.Kom.
Imam Kuswardayan, S.Kom., M.T.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman Ini Sengaja Dikosongkan]



FINAL PROJECT- KI141502

**DYNAMIC WORLD DEVELOPMENT IN 3D
VIRTUAL GAME 'THE LABYRINTH' USING
RECURSIVE DIVISION ALGORITHM**

**PURBO PANAMBANG
NRP 5113100030**

**Advisor
Darlis Herumurti, S.Kom., M.Kom.
Imam Kuswardayan, S.Kom., M.T.**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

**RANCANG BANGUN PEMBANGKIT WORLD DINAMIS
PADA 3D VIRTUAL GAME 'THE LABYRINTH'
MENGUNAKAN ALGORITMA RECURSIVE DIVISION**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

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

Oleh:

PURBO PANAMBANG
NRP. 5113 100 630

Disetujui oleh Dosen Pembimbing Tugas Akhir
Darlis Herumurti, S.Kom., M.Kom.
NIP: 197712172003121001



(pembimbing 1)

Imam Kuswardayan, S.Kom., M.T.
NIP: 197612152003121001

(pembimbing 2)

**SURABAYA
JANUARI, 2017**

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN PEMBANGKIT WORLD DINAMIS PADA 3D VIRTUAL GAME ‘THE LABYRINTH’ MENGGUNAKAN ALGORITMA RECURSIVE DIVISION

Nama Mahasiswa : Purbo Panambang
NRP : 5113100030
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Darlis Herumurti, S.Kom.,
M.Kom.
Dosen Pembimbing II : Imam Kuswardayan, S.Kom., M.T.

Abstrak

Perkembangan aplikasi permainan saat ini mengalami kemajuan yang sangat pesat. Aplikasi permainan dengan berbagai genre terus muncul, salah satunya labirin. Aplikasi permainan dengan lingkungan permainan berbentuk labirin sudah sangat banyak, misalnya pacman, layer maze dan scary maze. Hal ini menjadi pertimbangan bagi para developer untuk mengembangkan aplikasi permainan labirin atau maze menjadi lebih menarik. Untuk itu, dalam pengembangan aplikasi permainan labirin diperlukan suatu generator untuk menciptakan maze secara dinamis. Selain itu, agar aplikasi permainan bisa memberikan pengalaman bermain yang lebih nyata maka digunakan teknologi realitas virtual untuk membantu menangani permasalahan tersebut.

Realitas virtual adalah teknologi yang membuat pengguna dapat berinteraksi dengan suatu lingkungan yang disimulasikan oleh komputer. Realitas virtual pada tugas akhir ini dikembangkan pada perangkat bergerak dengan SDK Google Cardboard supaya pengguna dalam hal ini penggemar permainan labirin dapat memainkannya berulang kali sesuai keinginan karena labirin yang terbentuk selalu berubah-ubah. Realitas virtual juga memberikan

rasa seperti benar-benar berada di dalam labirin yang sebenarnya. Pada aplikasi ini terdapat model tiga dimensi di sekitar lingkungan permainan yang dibuat sebaik mungkin agar terlihat nyata. .

Pengujian aplikasi dilakukan dengan mengoperasikannya pada perangkat berbeda dan juga melalui kuesioner responden yang pernah memainkan permainan yang sejenis labirin atau maze sebelumnya. Berdasarkan pengujian, dapat diketahui tidak ada perbedaan waktu yang signifikan untuk menjalankan aplikasi pada perangkat mobile dengan processor quad-core atau processor octa-core maupun dengan memori 2 GB atau memori 3 GB. Sedangkan berdasarkan hasil kuesioner responden dapat diketahui tingkat kenyamanan antarmuka mencapai 75%, kemiripan model dengan objek asil mencapai 71,30%, kejelasan dan kegunaan informasi mencapai 77,78%, performa aplikasi mencapai 69,50%, dan perasaan immersive mencapai 72,73%. Dengan dikembangkannya aplikasi ini, diharapkan dapat menjadi dasar untuk mengembangkan aplikasi permainan berjenis maze atau labirin yang lebih menarik.

Kata kunci: Perangkat bergerak, labirin, maze generator, cardboard, realitas virtual.

DYNAMIC WORLD DEVELOPMENT IN 3D VIRTUAL GAME 'THE LABYRINTH' USING RECURSIVE DIVISION ALGORITHM

Student Name : Purbo Panambang
NRP : 5113100030
Major : Teknik Informatika FTIf-ITS
Advisor I : Darlis Herumurti, S.Kom., M.Kom.
Advisor II : Imam Kuswardayan, S.Kom., M.T.

Abstract

The development of game applications is currently progressing very rapidly. Applications of the game with various genres continue to appear, one of them labyrinth. Application game with game environment in the form of maze is very much, for example pacman, layer maze and scary maze. This is a consideration for developers to develop a maze game or maze game to be more interesting. For that, in the development of game maze application required a maze generator to create maze dynamically. In addition, in order for game apps to provide a more real-life experience, virtual reality technology is used to help deal with the problem. Virtual reality is a technology that enables users to interact with an environment simulated by a computer. The virtual reality in this final project is developed on mobile devices with the Google Cardboard SDK so that users in this case the labyrinth game fans can use it repeatedly as they see fit. Virtual reality also gives a sense of being truly inside the real maze. In this application there is a three-dimensional model around the game environment that made the best possible to make it look real. . Testing the application is done by operating on different devices and also through questionnaires of respondents who have visited

the place of pilgrimage directly. Based on testing, it can be seen there is no significant time difference to run applications on mobile devices with quad-core processor or octa-core processor or with 2 GB memory or 3 GB memory. While based on the questionnaire of respondents can know the level of interface comfort reaches 75%, similarity model with the object of acil reaches 71.30%, clarity and usefulness of information reaches 77.78%, application performance reaches 69.50%, and immersive feeling reaches 72 , 73%. With the development of this application, is expected to be the basis for developing applications maze game more interesting.

Key words: Mobile devices, labyrinth, maze generator, cardboard, virtual reality.

KATA PENGANTAR

Segala puji dan syukur senantiasa penulis panjatkan kepada Tuhan Yang Maha Esa yang telah melimpahkan rahmat-Nya sehingga penulis masih dikaruniai kesehatan dan keselamatan

Pada kesempatan kali ini penulis dapat menyelesaikan tugas akhir yang berjudul “Rancang Bangun Pembangkit World Dinamis Pada 3d Virtual Game ‘The Labyrinth’ Menggunakan Algoritma Recursive Division”. Pengerjaan tugas akhir ini adalah saat bagi penulis untuk menerapkan ilmu-ilmu yang penulis dapatkan semenjak awal perkuliahan sampai hari ini di lingkungan jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya

Dalam pelaksanaan dan pembuatan tugas akhir ini, penulis menerima banyak bantuan dari berbagai pihak, oleh karena itu melalui lembaran kata pengantar ini, penulis menyampaikan ucapan terima kasih kepada:

1. Kedua orang tua penulis, bapak Suprpto dan ibu Suharti yang selalu mencurahkan doa, dukungan semangat, perhatian, serta kasih sayang kepada penulis.
2. Bapak Darlis Herumurti, S.Kom., M.Kom. dan Bapak Imam Kuswardayan, S.Kom., M.T. selaku dosen pembimbing yang senantiasa memberikan arahan dan semangat.
3. Bapak Imam Kuswardayan, S.Kom., M.T. selaku dosen wali sejak di mulai semester 8 hingga sekarang.
4. Kakak-kakak dan saudara penulis yang selalu memberi dukungan selama menempuh masa kuliah.
5. Guruh, Fajar, Cayza dan teman-teman TC’13 lainnya yang menjadi tempat diskusi dan selalu siap sedia ketika penulis mengalami kesulitan.
6. Sahabat-sahabati Sepuluh Nopember yang menemani penulis dalam hal semangat spiritual.

7. Para admin laboratorium IGS (Interaksi Grafika dan Seni) beserta IGSahabat sebagai tempat tukar ilmu dan pengalaman selama pengerjaan tugas akhir.
8. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya tugas akhir ini, penulis mengucapkan terima kasih.

Penulis telah mengerjakan sebaik mungkin dalam penyusunan aplikasi tugas akhir ini. Penulis juga memohon maaf apabila terdapat kekurangan, kesalahan, maupun kelalaian yang telah penulis lakukan.

Surabaya, Juni 2017

Purbo Panambang

DAFTAR ISI

| | |
|--|------|
| LEMBAR PENGESAHAN..... | v |
| Abstrak | vii |
| Abstract | ix |
| KATA PENGANTAR..... | xi |
| DAFTAR ISI..... | xiii |
| DAFTAR GAMBAR | xvii |
| DAFTAR TABEL | xix |
| DAFTAR KODE SUMBER | xxi |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Batasan Masalah..... | 2 |
| 1.4 Tujuan..... | 2 |
| 1.5 Manfaat..... | 3 |
| 1.6 Metodologi | 3 |
| 1.7 Sistematika Penulisan..... | 4 |
| BAB II TINJAUAN PUSTAKA..... | 7 |
| 2.1 Pemodelan 3D pada <i>Mobile</i> | 7 |
| 2.2 Realitas Virtual..... | 8 |
| 2.3 Unity3D | 9 |
| 2.4 Recursive Division | 10 |
| 2.5 Google Cardboard | 11 |
| BAB III ANALISIS DAN PERANCANGAN..... | 13 |
| 3.1 Analisis Sistem | 13 |
| 3.2 Perancangan Pembangkit <i>Maze</i> | 14 |
| 3.3 Perancangan Data | 16 |
| 3.4 Perancangan Model 3D | 16 |
| 3.5 Perancangan Animasi | 17 |
| 3.6 Perancangan Proses | 17 |
| 3.6.1 Proses Tampilkan Informasi Score | 18 |
| 3.6.2 Proses Tampilkan Informasi Batas Waktu (Timer) dan Permainan Selesai..... | 18 |

| | | |
|--------------------------|---|----|
| 3.6.3 | Proses Tampilkan informasi <i>Game Complete</i> | 18 |
| 3.7 | Perancangan Perangkat Lunak..... | 19 |
| 3.7.1 | Deskripsi Umum Perangkat Lunak..... | 19 |
| 3.7.2 | Spesifikasi Kebutuhan Fungsional..... | 20 |
| 3.7.3 | Spesifikasi Kebutuhan Non-Fungsional..... | 20 |
| 3.7.4 | Karakteristik Pengguna..... | 21 |
| 3.8 | Perancangan Sistem..... | 21 |
| 3.8.1 | Perancangan Diagram Kasus Penggunaan..... | 21 |
| 3.8.2 | Spesifikasi Skenario Kasus Penggunaan..... | 22 |
| 3.8.3 | Perancangan Antarmuka Pengguna..... | 25 |
| 3.8.4 | Perancangan Skenario Aturan Main..... | 27 |
| 3.8.5 | Perancangan Kontrol Aplikasi..... | 28 |
| BAB IV IMPLEMENTASI..... | | 31 |
| 4.1 | Lingkungan Implementasi..... | 31 |
| 4.2 | Implementasi Pembuatan Objek 3D..... | 31 |
| 4.2.1 | Implementasi Model Labirin..... | 32 |
| 4.2.2 | Implementasi Model Koin..... | 34 |
| 4.2.3 | Implementasi Objek Gerbang..... | 35 |
| 4.3 | Implementasi Pembuatan Aplikasi..... | 36 |
| 4.3.1 | Implementasi Antarmuka Menu..... | 36 |
| 4.3.2 | Implementasi Realitas Virtual dengan Cardboard..... | 38 |
| 4.3.3 | Implementasi Pembuatan <i>Scene</i> | 38 |
| 4.3.4 | Implementasi Import Assets..... | 38 |
| 4.3.5 | Implementasi <i>Load</i> Objek ke <i>Scene</i> | 39 |
| 4.3.6 | Implementasi Penambahan Audio..... | 40 |
| 4.3.7 | Implementasi Pembuatan <i>Script</i> | 41 |
| 4.3.8 | Implementasi Tampilan Langit (<i>Skybox</i>)..... | 42 |
| 4.3.9 | Implementasi Kontrol Aplikasi..... | 42 |
| 4.3.10 | Implementasi Menampilkan Informasi..... | 45 |
| 4.4 | Implementasi Aplikasi Permainan..... | 48 |
| 4.4.1 | Implementasi <i>Maze Generator</i> | 48 |
| 4.4.2 | Implementasi Menelusuri Labirin dan Mengambil Koin..... | 51 |
| 4.4.3 | Implementasi Memperoleh Informasi..... | 52 |
| 4.4.4 | Implementasi Menemukan Finish..... | 54 |

| | |
|---|----|
| BAB V PENGUJIAN DAN EVALUASI | 57 |
| 5.1 Lingkungan Uji Coba | 57 |
| 5.2 Skenario dan Hasil Uji Coba | 58 |
| 5.2.1 Pengujian Fungsionalitas | 58 |
| 5.2.2 Pengujian Non-Fungsionalitas | 63 |
| 5.3 Evaluasi Pengujian | 68 |
| 5.3.1 Evaluasi Pengujian Fungsionalitas | 68 |
| 5.3.2 Evaluasi Pengujian Non-Fungsionalitas | 69 |
| BAB VI KESIMPULAN DAN SARAN..... | 71 |
| 6.1. Kesimpulan..... | 71 |
| 6.2. Saran..... | 72 |
| 7 DAFTAR PUSTAKA | 73 |
| A. LAMPIRAN A PEMBUATAN MODEL 3D | 75 |
| B. LAMPIRAN B SCREENSHOT APLIKASI | 77 |
| C. LAMPIRAN C KUESIONER RESPONDEN | 85 |
| D. BIODATA PENULIS | 91 |

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Perbedaan model 3D <i>lowpoly</i> dengan <i>highpoly</i> | 8 |
| Gambar 2.2 Penggunaan tekstur yang tepat | 8 |
| Gambar 2.3 Realitas virtual pada sebuah game <i>action</i> | 9 |
| Gambar 2.4 Kacamata <i>cardboard</i> dengan <i>trigger magnet</i> | 11 |
| Gambar 2.5 Kacamata VR dan bluetooth controller | 12 |
| Gambar 3.1 Simulasi Pembangkit Maze | 15 |
| Gambar 3.2 Diagram kasus penggunaan | 22 |
| Gambar 3.3 Rancangan antarmuka halaman awal | 26 |
| Gambar 3.4 Rancangan antarmuka <i>scene</i> penelusuran | 27 |
| Gambar 4.1 Pembuatan Objek Dinding..... | 33 |
| Gambar 4.2 Pembuatan Objek Pilar | 33 |
| Gambar 4.3 Pembuatan Objek Lantai | 34 |
| Gambar 4.4 Pembuatan objek koin sebelum <i>texturing</i> | 34 |
| Gambar 4.5 Pembuatan objek koin setelah <i>texturing</i> | 35 |
| Gambar 4.6 Pembuatan Objek Gerbang | 36 |
| Gambar 4.7 Antarmuka Halaman Menu | 37 |
| Gambar 4.8 Import Unity Package | 38 |
| Gambar 4.9 Import Asset ke Projek Unity | 39 |
| Gambar 4.10 <i>Generate Collider</i> pada Model 3D | 40 |
| Gambar 4.11 Inspector Komponen Audio Source | 41 |
| Gambar 4.12 Pembuatan <i>Script</i> Baru | 41 |
| Gambar 4.13 Menambahkan <i>Skybox</i> pada <i>Scene</i> | 42 |
| Gambar 4.14 Pengaturan Komponen pada Objek | 44 |
| Gambar 4.15 Membuat Kotak <i>Trigger</i> Informasi | 46 |
| Gambar 4.16 Membuat Area Informasi..... | 46 |
| Gambar 4.17 Mengatur posisi teks informasi terhadap kamera .. | 47 |
| Gambar 4.18 Tampilan Informasi Menu | 53 |
| Gambar 4.19 Tampilan menu saat game <i>finish</i> | 54 |
| Gambar A.1 Rancangan Gerbang Finish..... | 75 |
| Gambar A.2 Rancangan Koin | 75 |
| Gambar A.3 Pemberian Tektur pada Lantai..... | 76 |
| Gambar B.1 Tampilan <i>Game</i> saat menhadap ke depan..... | 77 |
| Gambar B.1 Tampilan <i>Game</i> saat menhadap ke kiri..... | 77 |

| | |
|--|----|
| Gambar B.1 Tampilan <i>Game</i> saat menhadap ke kanan..... | 78 |
| Gambar B.1 Tampilan <i>Game</i> saat menhadap ke belakang | 78 |
| Gambar B.1 Tampilan <i>Game</i> saat menhadap ke atas | 79 |
| Gambar B.2 Tampilan Menu Utama | 79 |
| Gambar B.1 Tampilan <i>stage 1</i> awal | 80 |
| Gambar B.1 Tampilan <i>stage 1</i> saat <i>maze</i> berubah..... | 80 |
| Gambar B.1 Tampilan <i>stage 2</i> awal | 81 |
| Gambar B.1 Tampilan <i>stage 2</i> saat <i>maze</i> berubah..... | 81 |
| Gambar B.1 Tampilan <i>stage 3</i> awal | 82 |
| Gambar B.1 Tampilan <i>stage 3</i> saat <i>maze</i> berubah..... | 82 |
| Gambar B.1 Tampilan saat <i>Game Over</i> | 83 |
| Gambar B.1 Tampilan saat <i>Game Complete</i> | 83 |
| Gambar C.1 Kuesioner Responden Guruh Arya Senna..... | 85 |
| Gambar C.2 Kuesioner Responden Adi Wicaksana | 86 |
| Gambar C.3 Kuesioner Responden Naufal B F..... | 87 |
| Gambar C.4 Kuesioner Responden Fahmy | 88 |
| Gambar C.5 Kuesioner Responden Setiyo | 89 |
| Gambar C.6 Kuesioner Responden Mooy..... | 90 |

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

| | |
|--|----|
| Tabel 3.1 Karakteristik Pengguna | 21 |
| Tabel 3.2 Spesifikasi Kasus Penggunaan UC-001 | 22 |
| Tabel 3.3 Spesifikasi Kasus Penggunaan UC-002 | 23 |
| Tabel 3.4 Spesifikasi Kasus Penggunaan UC-003 | 24 |
| Tabel 3.5 Spesifikasi Kasus Penggunaan UC-004 | 25 |
| Tabel 3.6 Skenario Permainan..... | 27 |
| Tabel 4.1 Lingkungan Implementasi Perangkat Lunak..... | 31 |
| Tabel 5.1 Lingkungan Perangkat Keras | 57 |
| Tabel 5.2 Skenario Uji Coba Fungsionalitas | 58 |
| Tabel 5.3 Hasil Uji Coba Memilih Lokasi | 59 |
| Tabel 5.4 Hasil Uji Coba Menelusuri Tempat..... | 59 |
| Tabel 5.5 Hasil Uji Coba Melihat Informasi | 60 |
| Tabel 5.6 Hasil Uji Coba Menemukan Finish..... | 61 |
| Tabel 5.7 Hasil Uji Coba Pembangkit Maze | 62 |
| Tabel 5.8 Skenario Non-Fungsionalitas Aspek Operasional..... | 63 |
| Tabel 5.9 Daftar Penguji Coba Aplikasi..... | 63 |
| Tabel 5.10 Pemberian Skor Jawaban | 64 |
| Tabel 5.11 <i>Rating Scale</i> | 65 |
| Tabel 5.12 Indikator Penilaian Non-Fungsionalitas | 65 |
| Tabel 5.13 Hasil Pengujian Operasional | 66 |
| Tabel 5.14 Hasil Penilaian Antarmuka..... | 66 |
| Tabel 5.15 Hasil Penilaian Kemiripan Objek..... | 66 |
| Tabel 5.16 Hasil Penilaian Kegunaan Informasi | 67 |
| Tabel 5.17 Hasil Penilaian Performa Sistem | 67 |
| Tabel 5.18 Hasil Penilaian Realitas Virtual | 68 |
| Tabel 5.19 Rekapitulasi Hasil Uji Fungsionalitas | 68 |
| Tabel 5.20 Rekapitulasi Hasil Uji Coba Non-Fungsionalitas..... | 69 |

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

| | |
|--|----|
| Kode Sumber 4.1 Kontrol Aksi Tombol | 37 |
| Kode Sumber 4.2 Jalan Otomatis | 43 |
| Kode Sumber 4.3 Implementasi <i>IGvrGazeResponder</i> | 45 |
| Kode Sumber 4.4 Posisi Canvas Terhadap Kamera..... | 48 |
| Kode Sumber 4.5 Penentuan dinding yang akan dihilangkan | 50 |
| Kode Sumber 4.6 Pembentuk <i>Maze</i> secara acak | 51 |
| Kode Sumber 4.7 Implementasi Proses Mengambil Koin | 52 |
| Kode Sumber 4.8 Objek mengikuti kamera | 53 |
| Kode Sumber 4.9 Implementasi menu mencapai <i>finish</i> | 55 |

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan dunia game saat ini semakin pesat, salah satu permainan yang cukup populer adalah permainan maze. Maze adalah sebuah permainan dengan jalan sempit yang berkeluk dimana pemain atau player harus menjelajahi maze tersebut untuk mendapat jalan yang benar [1].

Namun untuk sekarang ini kebanyakan game maze masih menggunakan lingkungan bermain atau map yang statis sehingga menyebabkan pemain merasa cepat bosan memainkan game tersebut. Oleh karena itu, diperlukan pembangkit lingkungan bermain/map dinamis untuk game maze atau bisa disebut maze generator. Maze generator ini bertujuan untuk membentuk maze yang berbeda setiap kali player memainkan game tersebut. Sehingga player tidak cepat merasa bosan saat bermain game maze tersebut. Dalam rancang bangun 3D Virtual Game “The Labyrinth” ini akan menggunakan algoritma Recursive Division untuk diimplementasikan dalam maze generator.

Perkembangan game bukan hanya dalam jenis permainan saja melainkan juga perkembangan teknologi yang digunakan. Salah satu teknologi yang sedang populer saat ini adalah game menggunakan virtual reality. Virtual reality adalah teknologi yang memungkinkan pengguna bisa berinteraksi terhadap objek nyata yang harus distimulasikan menggunakan komputer [7]. Oleh karena teknologi virtual reality yang sedang populer tersebut maka akan diterapkan pada ‘The Labyrinth’ agar game tersebut menjadi lebih seru dan membuat player tidak cepat bosan.

1.2 Rumusan Masalah

Perumusan masalah pada Tugas Akhir ini antara lain sebagai berikut :

1. Bagaimana rancangan aturan main game The Labyrinth?
2. Bagaimana rancangan konsep algoritma Recursive Division dalam penerapan di *maze generator* pada *game* tersebut?
3. Bagaimana mengimplementasikan teknologi *virtual reality* ke dalam *game* 'The Labyrinth'?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan masalah, sebagai berikut:

1. Aplikasi yang dibuat merupakan aplikasi android berbasis *virtual reality* dengan menggunakan *google cardboard*.
2. Lingkungan pengembangan yang digunakan menggunakan aplikasi Unity 3D 5.5.0f3 Free License dan bahasa pemrograman C#.
3. Menggunakan *bluetooth controller* untuk memainkan permainan sebagai input dalam aplikasi.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah, sebagai berikut:

1. Menerapkan konsep dari algoritma Recursive Division untuk membangkitkan world dinamis berupa maze.
2. Pemanfaatan teknologi virtual reality menggunakan google cardboard untuk menciptakan lingkungan permainan yang terasa lebih nyata.

1.5 Manfaat

Manfaat dari penelitian tugas akhir ini adalah terciptanya suasana permainan yang tidak begitu membosankan karena mempunyai lingkungan permainan yang nyata dengan menerapkan teknologi *virtual reality* dan mempunyai lingkungan bermain yang dinamis dengan menerapkan algoritma Recursive Division untuk membuat maze generator. Hasil penelitian ini diharapkan mampu memberikan solusi terhadap pembangkit world dinamis pada aplikasi permainan yang lebih kompleks.

1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

- A. Penyusunan proposal tugas akhir
Proposal tugas akhir ini berisi deskripsi pendahuluan tugas akhir yang akan dikerjakan. Bagian pendahuluan terdiri dari latar belakang tugas akhir, rumusan masalah, batasan masalah, tujuan pengerjaan tugas akhir, dan manfaat dari hasil pengerjaan tugas akhir. Selain itu dalam proposal tugas akhir juga dijabarkan tinjauan pustaka yang digunakan sebagai referensi pendukung pengerjaan tugas akhir. Dalam proposal ini juga terdapat penjelasan mengenai metodologi yang dipakai, mulai dari tahap penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat juga sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.
- B. Studi literatur
Pada tahap ini, akan dilakukan pencarian informasi dan studi literatur sejumlah referensi tentang bagaimana cara mengimplementasikan konsep recursive Division pada game berjenis maze serta bagaimana cara menerapkan

teknologi virtual reality pada game yang nantinya akan dibuat.

- C. Implementasi perangkat lunak
Aplikasi ini akan dibangun menggunakan Game Engine Unity 3D Free, dengan bahasa pemrograman C#. Sedangkan untuk membuat *environment* dari game ini menggunakan Blender.
- D. Uji coba dan evaluasi
Pada tahap pengujian dan evaluasi, akan di lakukan pengujian dimana akan di pertanyakan apakah fungsionalitas permainan sudah sesuai. Seperti kesesuaian antara permainan dengan dunia nyata serta menguji komponen permainan seperti tombol – tombol apakah sudah berjalan sesuai fungsinya.
- E. Penyusunan laporan tugas akhir
Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan tugas akhir.

1.7 Sistematika Penulisan

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

BAB I PENDAHULUAN

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

BAB II TINJAUAN PUSTAKA

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

BAB III ANALISIS DAN PERANCANGAN

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, data, dan antarmuka.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembuatan objek, implementasi pembuatan aplikasi, dan implementasi pembuatan simulasi.

BAB V PENGUJIAN DAN EVALUASI

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

BAB VI PENUTUP

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

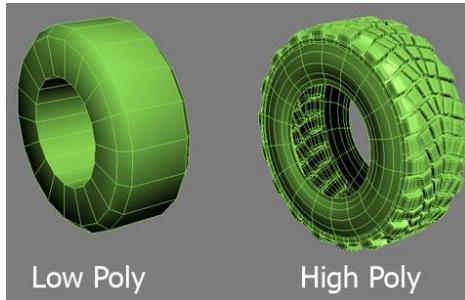
[Halaman ini sengaja dikosongkan]

BAB II TINJAUAN PUSTAKA

2.1 Pemodelan 3D pada *Mobile*

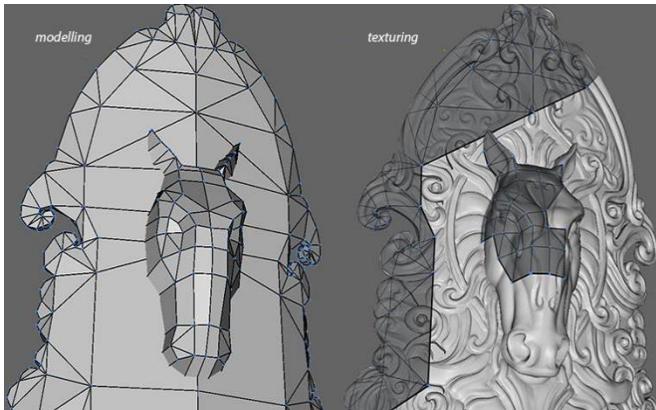
Kebanyakan model 3D dan animasi 3D banyak digunakan pada perangkat desktop karena mempunyai spesifikasi *hardware* yang lebih tinggi dibandingkan dengan perangkat bergerak (*mobile*). Pemodelan 3D pada desktop menggunakan metode pemodelan *highpoly*. Namun penggunaan *highpoly* pada perangkat *mobile* mempunyai resiko tidak berjalannya fungsionalitas sistem sebagaimana mestinya dengan baik. Perangkat *mobile* mempunyai konfigurasi *hardware* yang rendah sehingga tergolong dalam *Low Cost Access Devices (LCADs)* sehingga memerlukan pemodelan 3D dengan metode *lowpoly*. Hal tersebut dikarenakan tidak adanya *Graphic Accelerator* pada perangkat LCADs. Pemodelan 3D dengan metode *lowpoly* dapat mengurangi ukuran file, *mesh-density*, dan waktu *loading* objek 3D pada perangkat *mobile* [6].

Dalam sebuah model 3D, ketika tiga atau lebih titik yang dikenal dengan istilah *vertex* dihubungkan oleh sebuah *edge* maka akan membentuk poligon. Semakin banyak poligon pada sebuah model akan memberi kesan halus pada model 3D [7]. Seperti pada Gambar 2.1, terlihat perbedaan permukaan hasil metode *highpoly* dan *lowpoly* pada model 3D. Model 3D dengan banyak poligon mempunyai waktu *rendering* yang lebih lama jika dibandingkan dengan model 3D dengan sedikit poligon.



Gambar 2.1 Perbedaan model 3D *lowpoly* dengan *highpoly*

Model 3D dengan metode *highpoly* akan memberikan nilai estetika yang baik karena detail dari objek dapat divisualisasikan menyerupai aslinya. Hal tersebut yang menjadi kelemahan model 3D pada perangkat *mobile*, namun model 3D dengan metode *lowpoly* juga dapat memiliki nilai estetika dari pemilihan tekstur yang tepat. Pemilihan tekstur ini dapat memberi kesan lebih realistis. Sebagai contoh dapat dilihat pada Gambar 2.2 berikut.



Gambar 2.2 Penggunaan tekstur yang tepat

2.2 Realitas Virtual

Realitas virtual atau yang lebih disebut *Virtual Reality (VR)* adalah suatu teknologi yang memberi interaksi grafis secara

langsung (*real-time*) dengan model tiga dimensi yang dikombinasikan dengan teknologi layar yang mampu memberi kesan *immersive* (seolah-olah berada pada dunia nyata) kepada pengguna. Realitas Virtual pada dasarnya mengacu pada kesan *immersive*, interaktif, multi-sensor, pengolahan komputer yang menghasilkan lingkungan tiga dimensi dan kombinasi teknologi yang dibutuhkan untuk membangun lingkungan tersebut.

Realitas virtual memungkinkan untuk menyimulasikan dunia nyata ke dalam sistem visualisasi tiga dimensi dan memungkinkan interaksi pengguna dengan model karena kemampuan pelacakan gerak dan kemampuan untuk bertindak dengan produk secara *real time* [6]. Contoh dari pemanfaatan realitas virtual dapat dilihat pada Gambar 2.3.



Gambar 2.3 Realitas virtual pada sebuah game *action*

2.3 Unity3D

Unity adalah ekosistem pengembangan game yang terintegrasi kuat dengan satu set lengkap alat intuitif dan alur kerja yang cepat untuk membuat 3D interaktif dan konten 2D. Unity merupakan *easy multiplatform publishing*. Unity memiliki *tool Asset* yang menyediakan *Asset* untuk di unduk secara gratis maupun berbayar. Terdapat pula *Unity Community* yang menyediakan tutorial secara gratis untuk semua pengguna unity.

Fitur scripting yang disediakan, mendukung 3 bahasa pemrograman, yaitu JavaScript, C#, dan Boo. Flexible and EasyMoving, rotating, dan scaling objects hanya perlu sebaris kode. Begitu juga dengan Duplicating, removing, dan changing properties. Visual Properties Variables yang di definisikan dengan scripts ditampilkan pada Editor. Bisa digeser, di drag and drop, bisa memilih warna dengan color picker. Berbasis .NET. Artinya penjalanan program dilakukan dengan Open Source .NET platform, Mono. Serta mendukung pengembangan aplikasi Microsoft, SONY, Qualcomm, BlackBerry, Samsung, Ninetendo, Oculus VR dan Intel.

2.4 Recursive Division

Metode *Recursive Division* merupakan salah satu metode untuk membuat sebuah maze. Menurut Jamis Black “Metode *Recursive Division* adalah satu metode yang mengimplementasikan penambahan dinding”. Metode ini berfokus pada pembuatan dinding. Hal ini sangat berbeda dengan kebanyakan metode yang berfokus pada pembuatan jalur[5].

Recursive Division dibuat berdasarkan algoritma *divide-and-conquer*. Dalam penyelesaian algoritma *divide-and-conquer* dilakukan dengan cara membagi *problem* menjadi dua sub *problem*, kemudian setiap *subproblem* tersebut diselesaikan secara *recursive*. Langkah *recursive* tersebut dilakukan sampai pada *subproblem* terkecil yang tidak bisa dibagi lagi. Setelah mendapatkan hasil dari setiap *subproblem*, maka dikombinasikan kembali hasil tersebut sampai mendapat hasil dari *problem* utama. Dibawah ini adalah formula *reccurrence relation* dari algoritma *divide-and-conquer*

$$T_n = a T_{n/b} + f(n)$$

Dari *reccurrence relation* diketahui bahwa T_n adalah jumlah step dari *worst case* untuk menyelesaikan problem sejumlah n . Dan a adalah jumlah subproblem, lalu $f(n)$ adalah jumlah yang dibutuhkan untuk mengkombinasikan solusi dari setiap subproblem menjadi satu solusi[5].

2.5 Google Cardboard

Google Cardboard adalah teknologi penampilan realitas virtual yang berupa *Head Mounted Displays* (HMD). Semula, Google Cardboard terbuat dari karton dan dirancang oleh Google untuk pengguna *smartphone* (Gambar 2.4). Fungsi dari perangkat keras ini adalah sebagai alat bantu untuk memainkan aplikasi atau game VR. Google Cardboard bekerja baik dengan berbagai resolusi *smartphone* [8].



Gambar 2.4 Kacamata *cardboard* dengan *trigger magnet*

Dalam perkembangannya, Google Carboard membuat generasi keduanya dengan bahan dasar material plastik yang berkualitas untuk menopang *smartphone* dan dilengkapi dengan tali yang elastis agar dapat dikaitkan di kepala. Cardboard ini memiliki lensa berbentuk petak yang memberi ruang pengelihatn lebih luas mencapai 68 derajat atau setara dengan layar 150 inch pada jarak 3 meter. Bentuk dari VR generasi kedua ini juga mengalami peningkatan fungsionalitas. Bagi pengguna yang berkaca mata dapat menggunakan VR tanpa perlu melepas kaca mata. Cardboard ini dapat digunakan untuk semua *smartphone* dengan ukuran 4 inch sampai 5.7 inch.

Seiring perkembangan model Google Carboard, *controller* pada Google Cardboard juga mengalami perkembangan yang semula hanya menggunakan sebuah magnet pada sisi Cardboard

untuk melakukan *trigger*. Kini controller pada Cardboard sudah dapat menggunakan *bluetooth remote control*. *Controller* realitas virtual dapat dilihat pada Gambar 2.5.



Gambar 2.5 Kacamata VR dan bluetooth controller

Pihak pengembang Google Cardboard juga telah menyertakan *Software Development Kit* (SDK) yang berisi modul-modul pembantu untuk mempermudah pembuatan aplikasi yang berbasis Google Cardboard. SDK ini berisi berbagai modul sumber kode yang dapat langsung digunakan oleh pengembang. SDK Google Carboard dapat digunakan dengan berbagai aplikasi lain seperti Unity, Android Studio, dll.

BAB III

ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang analisis dan perancangan pembangkit world dinamis pada 3D virtual game ‘the labyrinth’ menggunakan algoritma recursive division. Pembahasan yang akan dilakukan meliputi analisis fitur yang dibutuhkan dan perancangan perangkat lunak.

3.1 Analisis Sistem

Sudah banyak kita jumpai game berbasis labirin atau *maze*, misalnya seperti Pacman. Namun kebanyakan dari game tersebut mempunyai map yang statis atau tidak berubah-ubah pada setiap *stagenya*. Sehingga pengguna akan cepat merasa bosan setelah menyelesaikan suatu *stage* dari game tersebut. Memang ada beberapa game yang telah menerapkan generator *maze* untuk menciptakan *maze* yang dinamis, namun masih belum memberikan pengalaman yang berbeda dalam memainkan game tersebut. Oleh karena itu diperlukan sebuah generator *maze* untuk menciptakan *maze* yang dinamis yang dipadukan dengan teknologi *virtual reality* dengan Google Cardboard untuk memberikan pengalaman bermain yang tidak membosankan dan terasa lebih nyata karena menerapkan *virtual reality*.

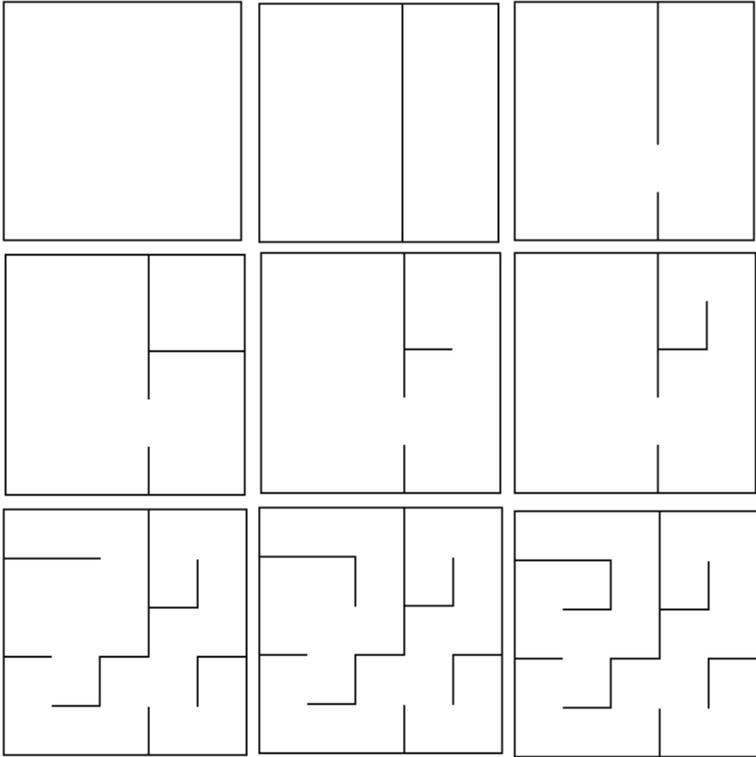
Aplikasi ini dibangun dengan teknologi Google Cardboard sehingga pengguna dapat menjalakkannya pada perangkat android yang mendukung sensor *gyroscope*. Aplikasi ini akan dibuat dengan menggunakan aplikasi Bleder untuk menyediakan objek 3D dan Unity karena kedua aplikasi ini telah menyediakan fitur yang mendukung Google Cardboard untuk *virtual reality*. Aplikasi ini menerapkan teknologi realitas virtual, sehingga pengguna dapat melihat dan merasakan pengalaman bermain yang nyata dalam menelusuri labirin tersebut.

3.2 Perancangan Pembangkit *Maze*

Perancangan Pada aplikasi permainan dengan *environment* berbentuk labirin atau maze ini akan menggunakan maze generator. Maze generator digunakan untuk menciptakan labirin atau maze secara otomatis dengan pola yang berbeda-beda. Untuk membuat maze generator tersebut memerlukan sebuah algoritma. Algoritma untuk membuat maze generator ini adalah *Recursive Division*. Proses pembentukan maze menggunakan algoritma Recursive Division adalah, sebagai berikut :

1. Membuat sebuah area dengan persegi panjang yang disebut ruang.
2. Tambahkan sebuah dinding secara vertikal atau secara horisontal secara acak untuk membagi ruang tersebut.
3. Beri jalan keluar di suatu titik secara acak pada dinding yang dibuat tadi.
4. Lakukan proses seperti langkah nomor 2 dan 3 sampai terbentuk *maze* yang memiliki jalan keluar dari *start* sampai *finish*. Proses ini berhenti saat *maze* yang terbentuk telah sampai pada *cell* terkecil dengan ukuran 1 x 1.

Dengan algoritma Recursive Division maka dapat menciptakan labirin atau maze yang selalu memiliki jalan keluar minimal satu. Untuk lebih jelas dapat dilihat pada Gambar 3.1.



Gambar 3.1 Simulasi Pembangkit Maze

Dalam recursive division terdapat *reccurence relation*. *Recursive division* mempunyai dua *variable* input yang digunakan. *Reccurence relation* dari *recursive division* adalah sebagai berikut :

$$divide(0, m, 0, n) = \begin{cases} divide(0, m/2, 0, n) & \text{untuk } m = n \text{ rand} > 50 \\ divide(0, m, 0, n/2) & \text{untuk } m = n \text{ rand} < 50 \\ divide(0, m/2, 0, n) & \text{untuk } m > n \\ divide(0, m, 0, n/2) & \text{untuk } m < n \\ 0 & \text{untuk } m < 1 \text{ dan } n < 1 \end{cases}$$

Dari *recurrence relation* tersebut diketahui bahwa m adalah lebar labirin dan n adalah tinggi labirin. Terdapat beberapa kondisi untuk membangkitkan labirin pada algoritma tersebut. Pertama, saat lebar maze dan tinggi maze bernilai sama dan nilai random lebih dari 50 maka maze akan di *divide vertical*, sehingga nilai m akan dibagi dua. Kedua, saat lebar maze dan tinggi maze bernilai sama dan nilai random kurang dari 50 maka maze akan di *divide horisontal*, sehingga nilai n akan dibagi dua. Ketiga, saat lebar maze lebih dari tinggi maze maka akan dilakukan *divide vertical*, nilai m akan dibagi dua. Keempat, saat lebar maze lebih kecil dari tinggi maze akan dilakukan *divide horisontal*, nilai n akan dibagi dua. *Recursive division* akan berjalan saat nilai m dan n lebih dari satu, maka saat nilai m dan n kurang dari satu pembangkit maze akan berhenti. Dari *recurrence relation* tersebut dapat dijadikan acuan untuk membuat program pembangkit labirin dengan algoritma *recursive division*.

3.3 Perancangan Data

Perancangan data merupakan hal yang diperlukan dalam memilih dan menyusun sistem agar tepat dan dapat beroperasi secara benar. Data yang diperlukan dalam pengoperasian perangkat lunak, yaitu meliputi objek-objek yang digunakan dalam membangun labirin, seperti dinding, lantai, pillar dan item-item yang berhubungan dengan aplikasi game ini. Audio juga diperlukan untuk menambah suasana yang lebih nyata pada aplikasi ini. Data mengenai lingkungan tersebut diambil dari berbagai sumber meliputi foto, video, model 3D serta aplikasi serupa. Sedangkan data audio doa didapatkan dari halaman web yang menyediakan asset audio gratis untuk game.

3.4 Perancangan Model 3D

Perancangan model dibuat menggunakan Blender versi 2.78b. Model yang dibuat untuk membentuk sebuah labirin diantaranya adalah tembok, lantai pillar dan gerbang untuk pintu

keluar labirin. Setelah model jadi, kemudian model diberikan tekstur untuk bisa diolah di Unity.

Perancangan model juga dilakukan dengan cara perancangan ulang model 3D yang sudah ada. Perancangan ulang model dilakukan supaya mendapatkan model *lowpoly*. Perancangan model *lowpoly* tentunya tidak melupakan aspek kemiripan dengan aslinya. Tujuan dirancangnya model *lowpoly* ini adalah untuk mengurangi beban kerja *processor* perangkat bergerak.

Model yang sudah dirancang dilakukan pengecekan ulang untuk pengoptimalan jumlah poligon (*mesh*). Caranya yaitu menghilangkan bagian yang tidak utama dari objek. Bagian yang dihilangkan kemudian diganti dengan pemberian tekstur yang berbeda.

3.5 Perancangan Animasi

Perancangan animasi ini diterapkan pada *Non Playable Character* (NPC) yang nantinya dapat dilihat oleh pengguna. NPC dalam aplikasi ini berfungsi untuk memberikan kesan yang lebih nyata dan menghibur pada aplikasi game ini. Pada aplikasi game pengguna diminta untuk mengumpulkan item dalam bentuk koin yang letaknya diatur secara acak. Pada item koin tersebut diberikan animasi agar koin tersebut berotasi. Berikut akan dijelaskan urutan perancangan animasi berjalan:

1. Tentukan objek yang akan diberi animasi
2. tambahkan *rigging* pada objek
3. tentukan posisi awal objek
4. atur panjang *frame*
5. ubah posisi dan arah objek pada *frame* tertentu

3.6 Perancangan Proses

Pada subbab ini akan dijelaskan rancangan proses yang akan dilakukan untuk memenuhi kebutuhan fungsionalitas perangkat lunak yang dibangun. Berikut ini adalah penjelasan dari rancangan proses yang dilakukan.

3.6.1 Proses Tampilkan Informasi Score

Pembuatan informasi score yang selalu ditampilkan di depan kamera untuk menunjukkan jumlah score yang didapatkan berdasarkan item koin yang telah diambil. Score akan bertambah saat *Trigger* pada item koin aktif. *Trigger* akan aktif jika kamera menyentuhnya, item koin akan dihilangkan dan score akan bertambah, berikut proses pembuatan informasi score:

1. Buat objek untuk item koin
2. tambahkan *collider* dan *trigger* pada objek tersebut
3. cek apakah *trigger* aktif atau tidak
4. jika aktif, *destroy* objek item koin tersebut dan tambah jumlah *score*.

3.6.2 Proses Tampilkan Informasi Batas Waktu (Timer) dan Permainan Selesai

Pada aplikasi permainan ini terdapat batasan waktu untuk menyelesaikan suatu level. Batasan waktu tersebut berbeda-beda untuk setiap levelnya. Saat batas waktu yang telah ditentukan telah habis maka sistem akan menampilkan informasi bahwa permainan selesai dan pengguna diberikan pilihan untuk mengulang level tersebut atau menyerah. Berikut proses tampilkan informasi batas waktu dan permainan selesai:

1. Buat penghitung penghitung batas waktu mundur (timer)
2. buat objek informasi *Game Over*
3. sembunyikan tekstur informasi *Game Over*
4. cek apakah batas waktu sudah habis atau masih tersisa
5. jika waktu habis, tampilkan tekstur informasi *Game Over*

3.6.3 Proses Tampilkan informasi *Game Complete*

Ketika pengguna telah menemukan jalan keluar atau garis finish maka sistem akan menampilkan informasi *Game Complete* yang menunjukkan bahwa suatu level dari aplikasi game tersebut telah berhasil diselesaikan. Pembuatan informasi pada garis *finish* membutuhkan area untuk menampilkan informasi dan *Trigger* untuk menampilkan informasi. *Trigger* akan aktif jika

kamera menabraknya, sehingga informasi *Game Complete* akan tampil. Proses pembuatan informasi *Game Complete* adalah sebagai berikut:

1. Buat *mesh* didekat objek garis *finish*
2. Tambahkan *collider* dan *trigger* pada *mesh* tersebut
3. cek apakah *trigger* aktif atau tidak
4. jika aktif, tampilkan informasi *game complete*.

3.7 Perancangan Perangkat Lunak

Pada subbab ini akan dibahas mengenai deskripsi umum perangkat lunak, spesifikasi kebutuhan fungsional dan kebutuhan non-fungsional serta bagaimana karakteristik pengguna aplikasi.

3.7.1 Deskripsi Umum Perangkat Lunak

Tugas akhir ini berbentuk realitas virtual dan diperuntukkan pada sistem operasi android, di mana pengguna harus menggunakan pada perangkat bergerak dengan sistem android versi 4.1 atau lebih tinggi yang dilengkapi dengan sensor *Accelerometer* dan sensor *Gyroscope*[5]. Perangkat lain yang dibutuhkan adalah kacamata VR (*Virtual Reality*) dan *controller* agar dapat memberi rasa *immersive* yang lebih baik. Pengguna utama dari aplikasi ini adalah orang yang ingin merasakan pengalaman bermain game berjenis labirin dengan suasana yang lebih nyata dan dengan lingkungan permainan yang dinamis. Dalam aplikasi ini pengguna dapat menelusuri labirin dengan bebas sambil mengumpulkan item yang tersedia. Objek-objek dalam aplikasi ini dibuat sedemikian rupa hingga menyerupai objek aslinya, termasuk sifat-sifat fisiknya seperti tidak tembus jika pengguna menabraknya. Terdapat juga objek tambahan yang tidak ada pada kondisi aslinya, objek ini ada yang bersifat kasat mata tetapi dapat ditembus dan ada juga yang tak kasat mata dan dapat ditembus. Objek tersebut berfungsi untuk menampilkan informasi atau memutar audio.

3.7.2 Spesifikasi Kebutuhan Fungsional

Pada aplikasi ini ada beberapa kebutuhan fungsionalitas yang mendukung untuk berjalannya aplikasi. Kebutuhan fungsionalitas pada aplikasi ini antara lain sebagai berikut :

- a) Mendeteksi input pengguna
Input pengguna yang diberikan kepada aplikasi dapat dibedakan menjadi dua jenis. Pertama input langsung dari pengguna, yaitu seperti menekan *trigger* pada Google Cardboard atau pada *bluetooth remote controller* dan gerakan terhadap posisi *smartphone*. Kedua, input pengguna yang merupakan tindak lanjut dari input langsung, seperti posisi kamera atau kamera yang melewati dan mengaktifkan *trigger* dalam aplikasi.
- b) Sistem dapat menampilkan informasi
Aplikasi menampilkan informasi sesuai masukkan dari pengguna saat menggunakan aplikasi.
- c) Sistem dapat memainkan audio musik
Aplikasi memutar audio musik sesuai kondisi dan *trigger* yang diberikan pengguna.

3.7.3 Spesifikasi Kebutuhan Non-Fungsional

Terdapat beberapa kebutuhan non-fungsional. Apabila kebutuhan non-fungsionalitas dipenuhi, dapat meningkatkan kualitas dari aplikasi ini. Berikut daftar kebutuhan non-fungsional:

- a) Aspek Performa
Aspek performa ini meliputi waktu jeda aksi input dari pengguna dengan reaksi yang diberikan oleh aplikasi. Semakin cepat waktu jeda maka kenyamanan penggunaan aplikasi semakin meningkat. Dengan kata lain, aplikasi dapat berjalan secara *real time*. Aplikasi maksimal harus merespon arah tampilan ketika perangkat digerakkan maksimal satu detik.
- b) Aspek Operasional

Aspek operasional yang mempengaruhi aplikasi ini berkaitan dengan lingkungan pengoperasional, yaitu meliputi jumlah *core* CPU perangkat, RAM, kecepatan prosesor dll.

c) Aspek Usabilitas

Pada aspek ini indikator yang digunakan adalah kemudahan penggunaan menu, kemudahan dalam mendapatkan informasi, dan kesan *immersive*.

3.7.4 Karakteristik Pengguna

Pengguna yang akan menggunakan aplikasi ini berjumlah satu orang, yaitu pengguna yang akan menjalankan aplikasi ini. Karakteristik pengguna tercantum dalam Tabel 3.1 berikut.

Tabel 3.1 Karakteristik Pengguna

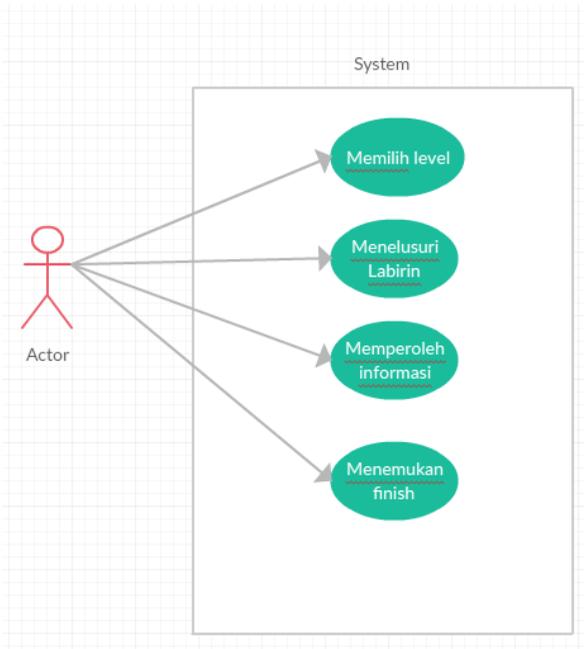
| Nama Aktor | Tugas | Hak Akses Aplikasi | Kemampuan yang harus dimiliki |
|------------|----------------------------------|----------------------|-------------------------------|
| Pengguna | Pihak luar yang mencoba aplikasi | Menjalankan aplikasi | Tidak ada |

3.8 Perancangan Sistem

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

3.8.1 Perancangan Diagram Kasus Penggunaan

Dalam aplikasi tugas akhir ini, terdapat empat kasus penggunaan yang ada yaitu memilih level, menelusuri labirin, memperoleh informasi, mengambil item, dan menemukan *finish*. Rancangan kasus penggunaan dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram kasus penggunaan

3.8.2 Spesifikasi Skenario Kasus Penggunaan

Setelah diketahui bagaimana perancangan kasus penggunaan, maka spesifikasi kasus penggunaan akan dijelaskan pada subbab berikut.

3.8.2.1 Memilih Stage

Spesifikasi kasus penggunaan Memilih Level dapat dijabarkan pada Tabel 3.2.

Tabel 3.2 Spesifikasi Kasus Penggunaan UC-001

| Spesifikasi | Keterangan |
|-----------------------|------------|
| Kode Kasus Penggunaan | UC-001 |

| | | |
|-----------------------|---|-------------------------------------|
| Nama Kasus Penggunaan | Memilih level | |
| Deskripsi | Kasus pengguna ini berfungsi supaya aktor dapat memilih level permainan yang diinginkan | |
| Kondisi Awal | Aplikasi sudah berjalan dan menampilkan pilihan level | |
| Kondisi Akhir | <i>Scene</i> level yang dipilih terbuka | |
| Aktor | Pengguna | |
| Alur Utama | Aktor | Sistem |
| | Mengarahkan <i>reticle</i> ke menu level yang dipilih | Menampilkan nama level |
| | Menekan <i>trigger</i> pada <i>remote</i> | <i>Load</i> dan <i>render</i> objek |
| Alur Alternatif | Aktor | Sistem |
| | Menekan tombol <i>back</i> di <i>smartphone</i> | Membuka halaman utama |

3.8.2.2 Menelusuri Tempat

Spesifikasi kasus penggunaan Menelusuri Tempat dapat dijabarkan pada Tabel 3.3.

Tabel 3.3 Spesifikasi Kasus Penggunaan UC-002

| Spesifikasi | Keterangan | |
|-----------------------|--|---|
| Kode Kasus Penggunaan | UC-002 | |
| Nama Kasus Penggunaan | Menelusuri Labirin | |
| Deskripsi | Kasus pengguna ini berfungsi supaya aktor dapat berjalan dan melihat-lihat di lingkungan permainan | |
| Kondisi Awal | Objek sudah tampil pada aplikasi | |
| Kondisi Akhir | - | |
| Aktor | Pengguna | |
| Alur Utama | Aktor | Sistem |
| | Mengarahkan perangkat pada objek tertentu | Menampilkan objek sesuai arah sudut <i>smartphone</i> |

| | | |
|-----------------|--|---|
| | Mengarahkan perangkat pada sudut tertentu (kondisi berhenti) | Menggerakkan kamera ke depan |
| | Mengarahkan perangkat pada sudut tertentu (kondisi berjalan) | Menghentikan gerak kamera ke depan |
| Alur Alternatif | Aktor | Sistem |
| | Berhenti di titik tertentu sampai batas waktu habis | Memulai ulang <i>scene stage</i> sekarang |

3.8.2.3 Memperoleh Informasi *Timer* dan Penambahan *Score*

Spesifikasi kasus penggunaan Memperoleh Informasi dapat dijabarkan pada Tabel 3.4.

Tabel 3.4 Spesifikasi Kasus Penggunaan UC-003

| Spesifikasi | Keterangan | |
|-----------------------|--|--|
| Kode Kasus Penggunaan | UC-003 | |
| Nama Kasus Penggunaan | Memperoleh informasi | |
| Deskripsi | Kasus pengguna ini berfungsi supaya aktor dapat mendapatkan informasi batas waktu dan score yang diperoleh | |
| Kondisi Awal | Teks informasi telah muncul | |
| Kondisi Akhir | Teks informasi berubah nilai | |
| Aktor | Pengguna | |
| Alur Utama | Aktor | Sistem |
| | Mengarahkan perangkat pada sudut tertentu | Menampilkan teks informasi score dan batas waktu |
| | Menabrak objek tertentu | Mengubah nilai dari informasi score dan informasi batas waktu berubah secara konstan |

| | | |
|-----------------|-------------------------|-----------------------------|
| Alur Alternatif | Aktor | Sistem |
| | Mentrigger menu kembali | Membuka halaman pilih level |

3.8.2.4 Menemukan *Finish*

Spesifikasi kasus penggunaan Menemukan *Finish* dapat dijabarkan pada Tabel 3.5.

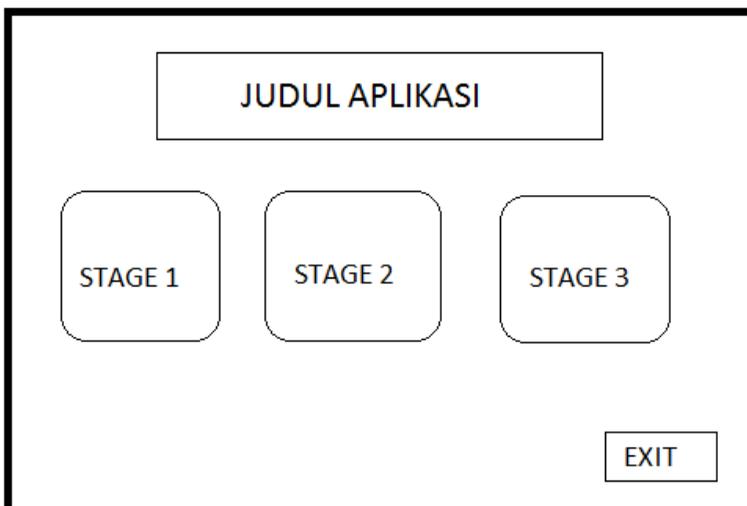
Tabel 3.5 Spesifikasi Kasus Penggunaan UC-004

| Spesifikasi | Keterangan | |
|-----------------------|---|---|
| Kode Kasus Penggunaan | UC-004 | |
| Nama Kasus Penggunaan | Mengambil Item | |
| Deskripsi | Kasus pengguna ini berfungsi supaya aktor dapat mendapatkan informasi dan pilihan untuk melanjutkan ke level berikutnya | |
| Kondisi Awal | Teks informasi tidak muncul | |
| Kondisi Akhir | Pindah ke <i>scene level</i> selanjutnya | |
| Aktor | Pengguna | |
| Alur Utama | Aktor | Sistem |
| | Menabrak objek finish | Menampilkan teks informasi dan pilihan untuk ke level selanjutnya |
| | Mentrigger kotak <i>next level</i> | Berpindah ke <i>scene level</i> berikutnya |
| Alur Alternatif | Aktor | Sistem |
| | Mentrigger kotak <i>Restart</i> | Memulai ulang <i>scene stage</i> sekarang |

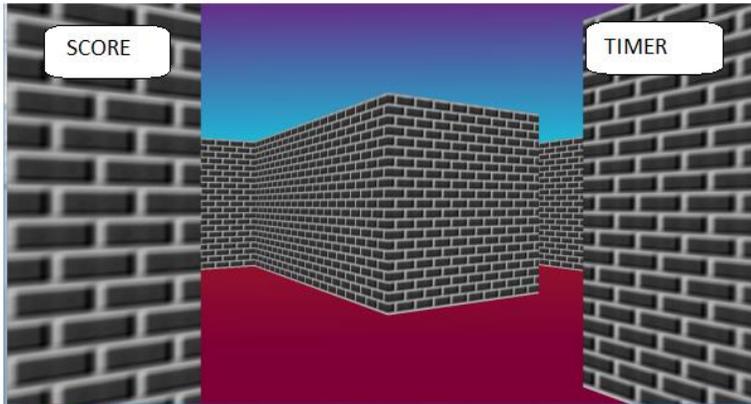
3.8.3 Perancangan Antarmuka Pengguna

Subbab ini membahas bagaimana rancangan antarmuka pengguna yang akan digunakan untuk Tugas Akhir. Aplikasi ini akan dibangun dengan mode normal dan mode *First Person Camera* (FPS), maka rancangan antarmuka yang dari aplikasi ini

hanya berjumlah satu buah layar kosong yang akan diisi oleh objek-objek. Objek yang ditampilkan akan diisi sesuai *scene* masing-masing. Selain itu, tampilan juga ditambah dengan penampilan stereoskopik agar memperdalam visual. Rancangan dari antarmuka aplikasi ini dapat dilihat pada Gambar 3.3 dan Gambar 3.4.



Gambar 3.3 Rancangan antarmuka halaman awal



Gambar 3.4 Rancangan antarmuka *scene* penelusuran

3.8.4 Perancangan Skenario Aturan Main

Pada aplikasi ini menerapkan aturan main dengan cara menemukan jalan keluar dari labirin. Langkah pertama yang harus dilakukan untuk memulai permainan ini adalah memilih stage antara *stage 1*, *stage 2*, dan *stage 3*. Setelah memilih stage, sistem akan langsung masuk ke permainan. Dalam permainan, kontrol yang digunakan adalah dengan mengarahkan kamera kearah lantai untuk bergerak maju atau berjalan, lalu mengarahkan kamera ke objek selain lantai untuk berhenti. Pengguna harus menemukan garis finish untuk menyelesaikan permainan ini dengan menelusuri labirin. Saat menelusuri labirin pengguna diminta mengambil koin sebanyak banyaknya. Kondisi kalah atau menang dalam permainan ini yaitu, pengguna akan menang dan menyelesaikan suatu stage jika mencapai finish sebelum waktu yang disediakan habis. Kemudian pengguna akan kalah dan mengulang pada stage tersebut jika tidak bisa mencapai garis finish sampai waktu yang ditentukan habis. Skenario permainan dapat dilihat pada Tabel 3.6

Tabel 3.6 Skenario Permainan

| Spesifikasi | Keterangan |
|---------------|------------------|
| Nama Skenario | Aturan Permainan |

| | | |
|-----------------|---|---|
| Deskripsi | Skeneraio ini berfungsi agar aktor dapat menyelesaikan aplikasi permainan | |
| Kondisi Awal | Objek sudah tampil pada aplikasi | |
| Kondisi Akhir | Pindah ke <i>scene level</i> selanjutnya | |
| Aktor | Pengguna | |
| Alur Utama | Aktor | Sistem |
| | Mengarahkan perangkat pada objek tertentu | Menampilkan objek sesuai arah sudut <i>smartphone</i> |
| | Mengarahkan perangkat pada sudut tertentu (kondisi berhenti) | Menggerakkan kamera ke depan |
| | Mengarahkan perangkat pada sudut tertentu (kondisi berjalan) | Menghentikan gerak kamera ke depan |
| | Mengearahkan kamera ke koin | Menggerakkan kamera ke koin (menggambil koin) |
| | Mengarahkan kamera ke objek finish | Menggerakkan kamera menuju garis finish |
| | Menarahkan kamera kedalam objek kastil | Menampilkan informasi permainan selesai |
| Alur Alternatif | Aktor | Sistem |
| | Berhenti di titik tertentu sampai batas waktu habis | Memulai ulang <i>scene stage</i> sekarang |

3.8.5 Perancangan Kontrol Aplikasi

Pada aplikasi ini terdapat dua jenis perangkat input, yaitu *Smartphone* android dan *Controller VR box / Cardboard*. *Smartphone* akan membantu fungsi pergerakan arah tampilan aplikasi yang dilihat oleh pengguna. Apabila *smartphone* dimiringkan ke atas atau ke bawah, maka pergerakan kamera juga akan bergerak ke atas dan ke bawah. Tidak hanya arah atas dan

dan bawah, arah kamera juga akan mengikuti posisi *smartphone* sehingga dapat melihat ke arah 360 derajat.

Controller VR box / Cardboard berfungsi seperti klik pada mouse maupun seperti tap pada perangkat *smartphone*. *Controller* ini dapat yang terhubung langsung dengan cardboard yaitu berupa magnet pada sisi cardboard maupun yang terpisah dengan perangkat cardboard yang dihubungkan dengan *bluetooth*. Untuk membantu klik pada tombol atau objek, pengguna dapat mengarahkan pointer atau yang dapat disebut *reticle* pada tengah layar ke arah tombol atau objek yang dituju.

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

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

4.1 Lingkungan Implementasi

Lingkungan implementasi tugas akhir dijelaskan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

| | |
|-----------------|---|
| Perangkat Keras | Pengembangan Aplikasi Prosesor : Intel(R) Core(TM) i5-4200U CPU @ 2.50GHz 2.50GHz Memori : 6.00 GB Implementasi Aplikasi Prosesor: Qualcomm Snapdragon 650 Hexa-core @ 1.80GHz Memori : 3 GB |
| Perangkat Lunak | Sistem Operasi : - Windows 8 64 bit (Pengembangan) - Android 5.1.1 (Implementasi) Perangkat Pengembang : - Blender 2.78b - Unity 5.5.0f3 - MonoDevelop v5.9.6 |

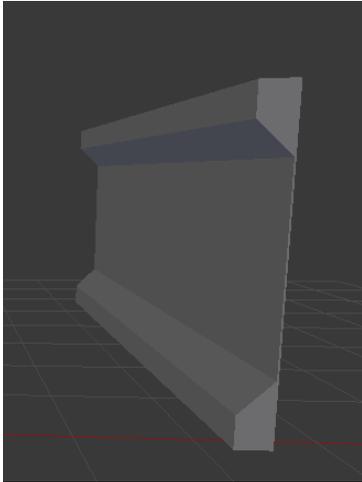
4.2 Implementasi Pembuatan Objek 3D

Implementasi pembuatan objek 3D dilakukan dengan memodelkan objek ke bentuk tiga dimensi terlebih dahulu. Setelah

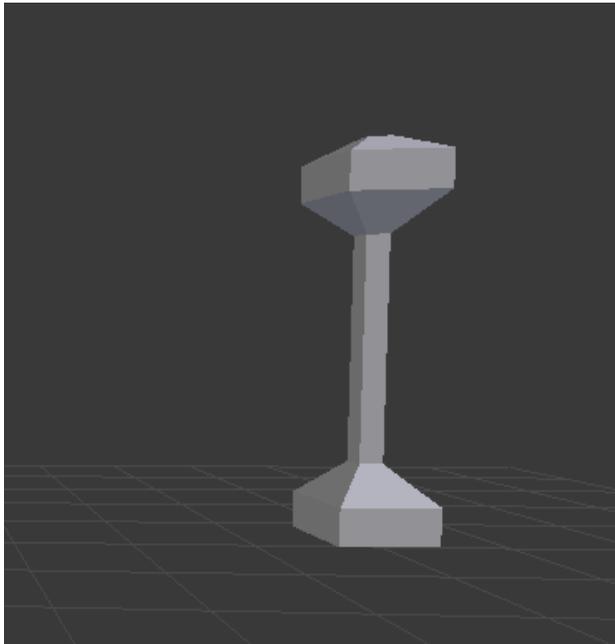
objek 3D terbentuk, dilakukan pengecekan kembali untuk mengetahui apakah objek 3D tersebut dapat dikurangi poligonnya. Setelah model 3D terbentuk, selanjutnya adalah pemberian tekstur yang disesuaikan dengan objek aslinya. Setelah diberi tekstur, objek diekspor dalam ekstensi .fbx karena mempunyai ukuran file yang lebih kecil dibandingkan .blend. Bab ini dibagi ke dalam tiga subbab meliputi Implementasi Model Labirin, Implementasi Model koin, dan Implementasi Objek Objek Gerbang.

4.2.1 Implementasi Model Labirin

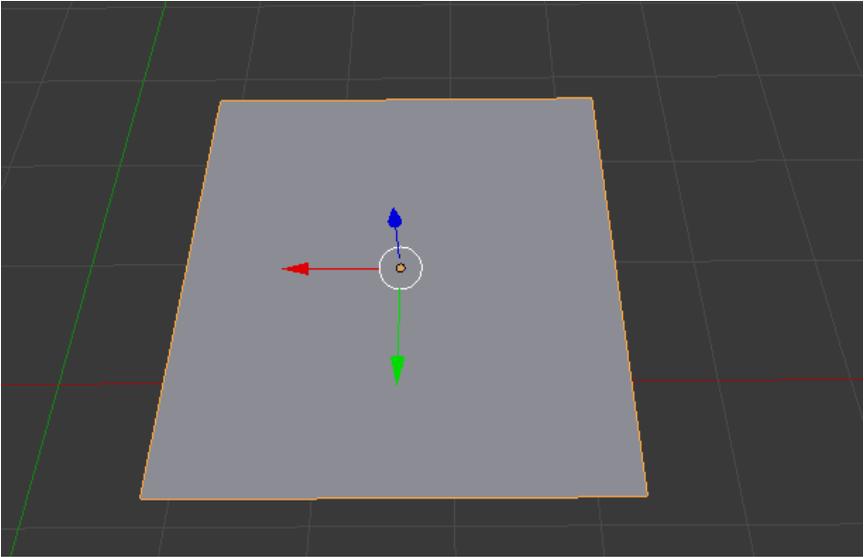
Pada subbab ini dijelaskan hasil pembuatan objek labirin. Dalam pembuatan model labirin ada 3 objek yang harus dibuat yaitu objek dinding, pilar dan lantai. Objek dinding dapat dilihat pada Gambar 4.1, objek pilar pada Gambar 4.2. Sedangkan pada Gambar 4.3 adalah objek lantai. Dari ketiga objek tersebut akan disusun menjadi sebuah labirin dengan *maze generator*. Selain membuat model dari awal, penulis juga melakukan modifikasi poligon model dari sumber yang tersedia di unity assetstore. Modifikasi model yaitu dengan menganalisis *edge* yang dapat dihilangkan sehingga model menjadi *lowpoly*. Objek-objek tersebut yaitu koin dan pintu gerbang.



Gambar 4.1 Pembuatan Objek Dinding



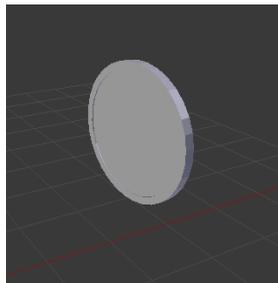
Gambar 4.2 Pembuatan Objek Pilar



Gambar 4.3 Pembuatan Objek Lantai

4.2.2 Implementasi Model Koin

Pada subbab ini dijelaskan hasil pembuatan objek *Koin*. Objek koin ini dibuat pada aplikasi blender tanpa memberikan texture. Proses pemberian texture akan dilakukan langsung pada aplikasi unity. Pembuatan objek koin sebelum diberi texture dapat dilihat pada Gambar 4.4. Setelah objek diberi texture objek koin dapat dilihat pada Gambar 4.5.



Gambar 4.4 Pembuatan objek koin sebelum *texturing*



Gambar 4.5 Pembuatan objek koin setelah *texturing*

4.2.3 Implementasi Objek Gerbang

Pembuatan objek gerbang ini akan digunakan untuk garis *finish* yang akan diterapkan dalam aturan permainan. Objek gerbang ini merupakan hasil modifikasi dari objek yang sudah tersedia di unity assetstore. Pembuatan objek gerbang dapat dilihat pada Gambar 4.6.



Gambar 4.6 Pembuatan Objek Gerbang

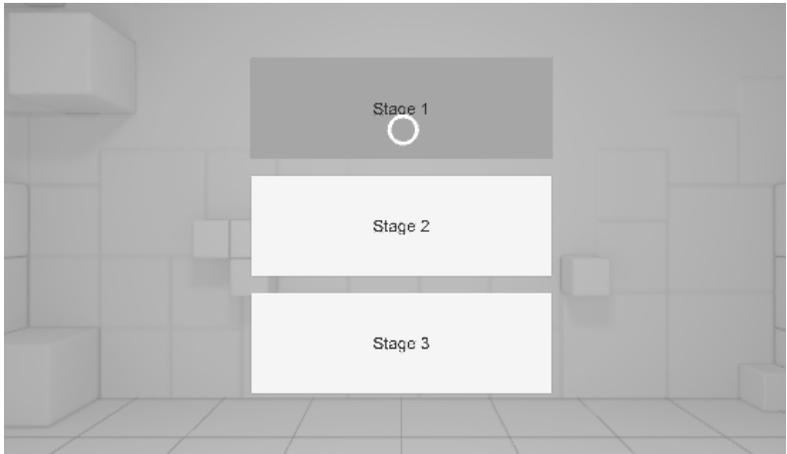
4.3 Implementasi Pembuatan Aplikasi

Pada tahap implementasi pembuatan aplikasi ini, akan dibagi ke dalam sepuluh subbab. Kesepuluh subbab tersebut yaitu meliputi Implementasi Antarmuka Menu, Implementasi Realitas Virtual dengan *Cardboard*, Implementasi Pembuatan *Scene*, Implementasi Import *Assets*, Implementasi *Load* Objek ke *Scene*, Implementasi Penambahan Audio, Implementasi Pembuatan *Script*, Implementasi Tampilan Langit (*Skybox*), Implementasi Kontrol Aplikasi, dan Implementasi Mendapatkan Informasi.

4.3.1 Implementasi Antarmuka Menu

Antarmuka menu yang diimplementasikan oleh penulis antara lain antarmuka menu awal. Antarmuka menu awal

merupakan halaman yang ditampilkan ketika pengguna membuka aplikasi. Pada halaman ini terdapat tiga tombol untuk memilih *stage* permainan yaitu *stage 1*, *stage 2*, dan *stage 3*. Untuk lebih jelasnya perhatikan Gambar 4.7. Untuk mengatur pergantian halaman dapat dilihat pada Kode Sumber 4.1.



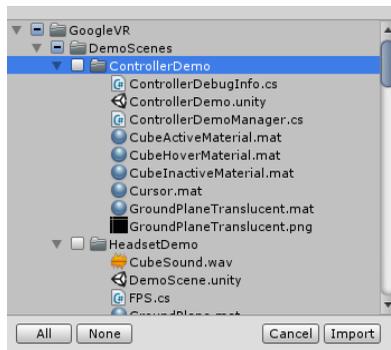
Gambar 4.7 Antarmuka Halaman Menu

```
1. public void Level1(){  
2.     SceneManager.LoadScene("Level1");  
3. }  
4. public void Level2(){  
5.     SceneManager.LoadScene("Level2");  
6. }  
7. public void Level3(){  
8.     SceneManager.LoadScene("Level3");  
9. }
```

Kode Sumber 4.1 Kontrol Aksi Tombol

4.3.2 Implementasi Realitas Virtual dengan Cardboard

Untuk pembuatan projek virtual reality pada perangkat bergerak, diperlukan SDK Google Carboard yang dapat diunduh pada <https://developers.google.com/vr>. Karena pengembangan aplikasi ini dilakukan pada Unity maka pilih Google VR SDK for Unity. Setelah SDK sudah diunduh maka import package kedalam projek Unity. Caranya dengan memilih menu “Assets” lalu pilih “Import Package” dan “Custom Package”. Pilih SDK yang diunduh dan tunggu hingga proses load assets selesai. Jika sudah pilih tombol import. Perhatikan Gambar 4.8.



Gambar 4.8 Import Unity Package

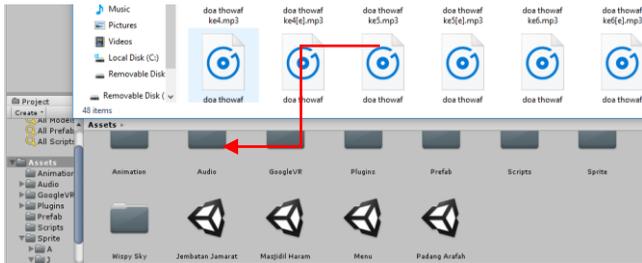
4.3.3 Implementasi Pembuatan Scene

Untuk menambahkan scene, hal yang perlu dilakukan adalah menuju menu “File” lalu pilih “New Scene”. Setelah scene baru terbuat, kita dapat menambah game objek kedalam scene dan mengatur sesuai rancangan yang dibuat.

4.3.4 Implementasi Import Assets

Implementasi import assets dapat dilakukan dengan cara *drag and drop* dari file explore ke tab “Project” didalam folder “Assets” pada unity. Untuk lebih jelasnya dapat dilihat pada Gambar 4.9. Kita juga dapat mengatur pengelompokan asset

dengan membuat folder-folder supaya lebih rapi dan memudahkan dalam pencarian asset.

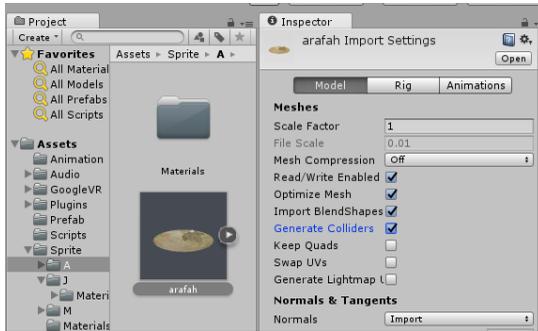


Gambar 4.9 Import Asset ke Proyek Unity

4.3.5 Implementasi *Load* Objek ke *Scene*

Untuk memasukan objek-objek ke dalam scene dapat dimasukkan dengan cara *drag and drop* berkas dari tab “Project” ke dalam *Scene*. Untuk melihat properti objek mengenai posisi atau yang lainnya, pilih objek yang akan dilihat propertinya. Informasi properti terdapat pada tab “Inspector”. Pada tab “Inspector” dapat ditambahkan beberapa komponen yang mempengaruhi objek pada *scene*.

Untuk objek 3D yang berasal dari file ekspor aplikasi blender dapat langsung ditambahkan *collider* agar dapat mendeteksi tumbukan antar objek. Hal ini diperlukan untuk mengatur pergerakan kamera agar tidak menembus objek. Langkah-langkahnya yaitu pada aset berekstensi .fbx aktifkan *Generate Collider*, kemudian klik tombol *Apply* (Gambar 4.10).



Gambar 4.10 *Generate Collider* pada Model 3D

Sedangkan untuk animasi yang berasal dari aplikasi blender dapat dilihat pada bagian “Animations”. Nama aplikasi dan pengaturan lainnya dapat juga diubah melalui bagian “Animations”.

4.3.6 Implementasi Penambahan Audio

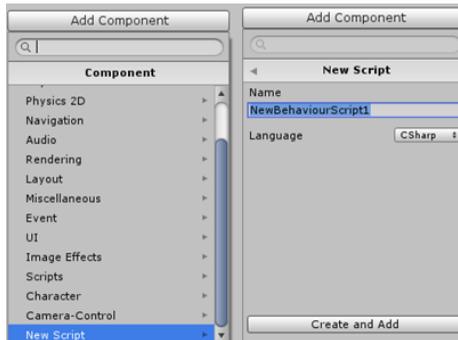
Untuk menambah audio langkah pertama yang harus dilakukan adalah pilih gameobject yang akan menyimpan audio tersebut. Kemudian pada tab “Inspector” tambahkan komponen *Audio Source*, jika berhasil maka akan muncul komponen *Audio Source* pada tab “Inspector” seperti pada Gambar 4.11. Pada komponen *Audio Source* terdapat *Audio Clip* yang berfungsi mengambil media music (.mp3 / .wav) dari tab “Project”. Hilangkan centang pada *Play on awake* jika tidak ingin memutar audio saat *Load Scene* dan centang *Loop* jika ingin memutar kembali audio setelah mencapai akhir.



Gambar 4.11 Inspector Komponen Audio Source

4.3.7 Implementasi Pembuatan Script

Untuk membuat *script* yang mempengaruhi objek adalah dengan memilih terlebih dahulu objek yang akan diberi *script*. Pada tab “Inspector” tambahkan komponen *script*. Jika tidak ada *script* yang sesuai, kita akan membuat file *script* baru dan mengubah isi file sesuai yang kita inginkan (Gambar 4.12).



Gambar 4.12 Pembuatan Script Baru

4.3.8 Implementasi Tampilan Langit (*Skybox*)

Untuk membuat *skybox*, penulis menggunakan tampilan yang tersedia di *assets store* unity. Caranya adalah dengan membuka *window* “Assets Store”. Pada kolom isian *search* masukkan kata kunci *skybox*. Jika sudah memilih *skybox* yang dituju, tekan tombol download. Assets *skybox* akan tersimpan pada proyek.

Setelah selesai mengunduh, selanjutnya ganti komponen *skybox* pada kamera utama dengan *skybox* yang telah diunduh. Untuk lebih jelasnya perhatikan Gambar 4.13.



Gambar 4.13 Menambahkan *Skybox* pada *Scene*

4.3.9 Implementasi Kontrol Aplikasi

Kontrol aplikasi adalah suatu cara untuk interaksi pengguna dengan sistem. Pengembangan aplikasi realitas virtual yang menggunakan Google Cardboard mempunyai keterbatasan dalam implementasi kontrol, salah satunya adalah keterbatasan menerima input. Untuk itu dalam implementasi kontrol aplikasi ini pengguna hanya bias memberi input berupa *trigger* dari Cardboard atau *bluetooth controller* ke perangkat *mobile* android.

4.3.9.1 Implementasi Jalan Otomatis

Pengguna dapat menelusuri tempat dengan cara berjalan otomatis, sehingga pengguna hanya perlu mengarahkan kamera ke arah objek yang memiliki nama “*Floor*” untuk berjalan dan untuk berhenti pengguna hanya perlu mengarahkan kamera ke arah objek selain “*Floor*”. Trigger tersebut akan mengubah nilai dari variable bertipe boolean “walking” (lihat Kode Sumber 4.2). Untuk itu diperlukan sebuah *script* yang mengatur gerakan kamera supaya dapat berjalan otomatis.

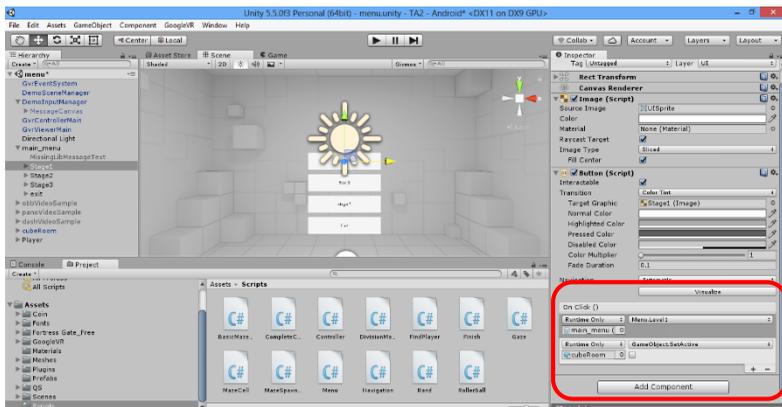
```
1.         if (walking) {
2.             transform.position = transform.position +
Camera.main.transform.forward * speed *
Time.deltaTime;
3.         }
4.         if (transform.position.y < -10f) {
5.             transform.position = SpawnPoint;
6.         }
7.
8.         Ray ray = Camera.main.ViewportPointToRay (new
Vector3 (0.5f, 0.5f, 0));
9.         RaycastHit hit;
10.        if (Physics.Raycast (ray, out hit)) {
11.            if (hit.collider.name.Contains ("Floor"))
|| hit.collider.name.Contains ("Finish") ) {
12.                walking = true;
13.            } else {
14.                walking = false;
15.            }
16.        }
17.
```

Kode Sumber 4.2 Jalan Otomatis

Pada Kode Sumber 4.2 terdapat variable “head” yang bertipe *GvrHead* (kelas dari package SDK Google VR). Variable digunakan untuk mengambil arah dari kamera. Terdapat juga variabel “speed” yang mengatur kecepatan jalan kamera.

4.3.9.2 Implementasi Pilih Stage

Pada pemilihan *stage*, pengguna perlu memberi *trigger* pada objek yang dilihat (dalam hal ini dibantu sebuah titik pada tengah kamera (*reticle*) untuk konfirmasi pilihan. Untuk membuat kode program berjalan maka diperlukan untuk menambah komponen *Collider* pada objek (Gambar 4.14).



Gambar 4.14 Pengaturan Komponen pada Objek

Tambahkan komponen *script* yang mengatur kode program. Langkah terakhir tambahkan *event trigger* yang merujuk *script* pada objek yang sama. Terdapat tiga jenis *event trigger* yang digunakan, yaitu:

1. *Pointer Enter* : merupakan kondisi saat *reticle* masuk pada area yang mempunyai *collider*
2. *Pointer Exit* : merupakan kondisi saat *reticle* keluar dari area yang mempunyai *collider*
3. *Pointer Click* : merupakan kondisi saat terjadi *trigger* pada area yang mempunyai *collider*

Kode program yang dibuat adalah *implement* dari kelas *IgvrResponder*. Sehingga dilakukan *implement* fungsi seperti pada Kode Sumber 4.3.

```
1. #region IGvrGazeResponder implementation
2. public void OnGazeEnter() {
3.     SetGazedAt(true);
4. }
5. public void OnGazeExit() {
6.     SetGazedAt(false);
7. }
8. public void OnGazeTrigger() {
9.     StartCoroutine(LoadNewScene());
10. }
11. #endregion
```

Kode Sumber 4.3 Implementasi *IGvrGazeResponder*

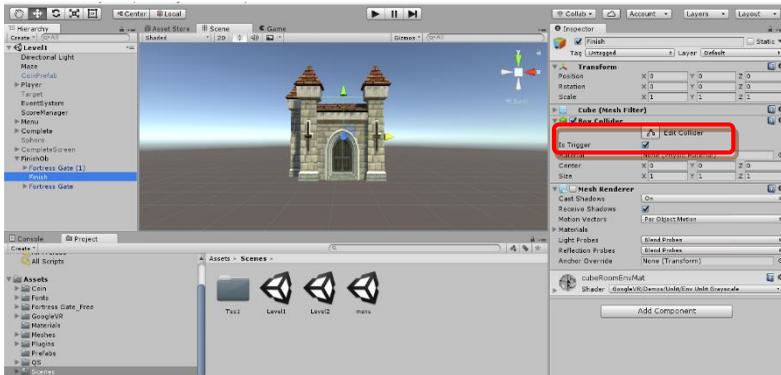
Pada implementasi fungsi diatas terdapat fungsi *StartCoroutine(LoadNewScene())* dengan parameter bertipe *IEnumerator*. Fungsi tersebut digunakan untuk memberi tanda ke pengguna bahwa proses *load* dan *render* objek masih berjalan.

4.3.10 Implementasi Menampilkan Informasi

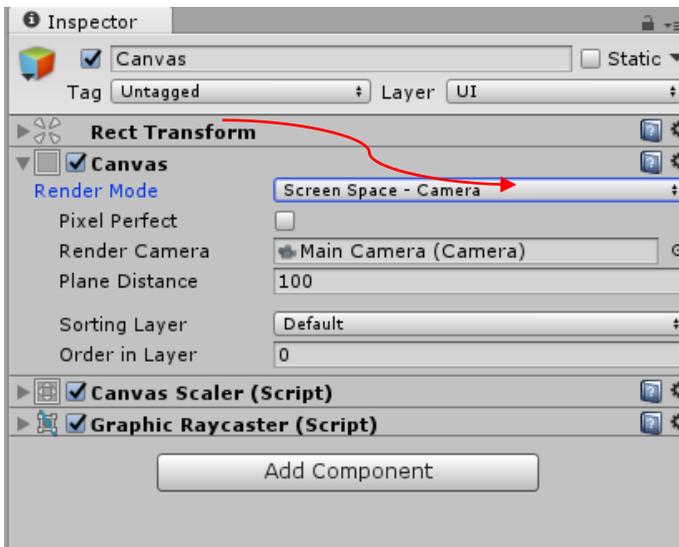
Untuk menampilkan sebuah informasi maka diperlukan sebuah *trigger* yang akan menampilkan informasi pada *screen* android jika *trigger* tersebut aktif. Hal yang diperlukan adalah sebuah canvas yang menampung isi informasi dan kotak *trigger* yang akan menampilkan informasi. Langkah-langkahnya adalah sebagai berikut:

1. Buat objek kotak yang menjadi *trigger*, tambahkan komponen *collider* dan aktifkan *trigger* (Gambar 4.15).
2. Tambahkan canvas pada hirarki proyek yang di dalamnya memuat latar informasi dan teks yang berisi informasi (Gambar 4.16). Atur mode “Render Canvas” sebagai *world space* dan *event camera* merujuk pada Main Camera di hirarki proyek.

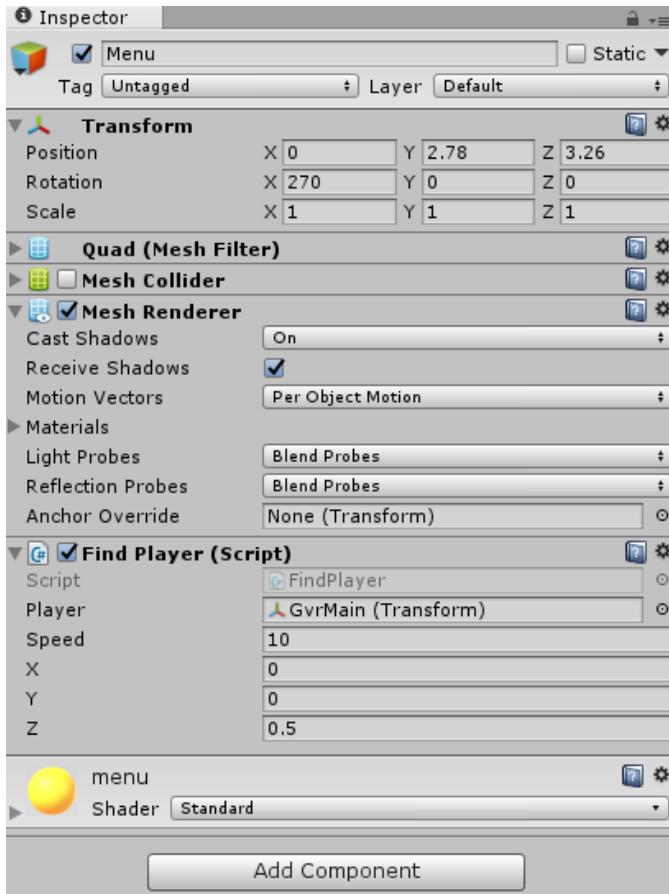
3. Pada latar informasi, tambahkan kode program yang mengatur informasi supaya selalu tampil mengikuti arah kamera. Perhatikan Gambar 4.17 dan Kode Sumber 4.4.



Gambar 4.15 Membuat Kotak *Trigger* Informasi



Gambar 4.16 Membuat Area Informasi



Gambar 4.17 Mengatur posisi teks informasi terhadap kamera

```

1.     void Start () {
2.         Player =
3.         GameObject.FindWithTag("Player").transform;
4.     }
5.     // Update is called once per frame
6.     void Update () {
7.         playerPos = new Vector3(Player.position.x,
8.         transform.position.y, Player.position.z + z );

```

```

8.         transform.position = Vector3.Lerp
           (transform.position, playerPos, Time.deltaTime *
           speed);
9.     }

```

Kode Sumber 4.4 Posisi Canvas Terhadap Kamera

Sebelum menambah *script*, terlebih dahulu menambahkan *collider* pada kamera supaya dapat mendeteksi tumbukan.

4.4 Implementasi Aplikasi Permainan

Dalam implementasi permainan akan dibagi menjadi empat subbab yang meliputi implementasi *maze generator*, memilih *stage*, implementasi mengambil koin, implementasi tampilan informasi menu dan implementasi mencapai *finish*.

4.4.1 Implementasi *Maze Generator*

Untuk membuat sebuah *maze generator* maka dapat dilihat pada Kode Sumber 4.5. Kode sumber tersebut menjelaskan tentang bagaimana menentukan dinding yang dihilangkan pada *maze* secara acak. Langkah yang dilakukan yaitu mulai dari menentukan dinding di sisi luar yang tidak boleh dihilangkan. Kemudian menentukan dinding mana yang akan dihilangkan secara horisontal maupun vertikal. Pada proses ini dinding yang akan dimunculkan diberi nilai *true* sedangkan dinding yang diberi lubang akan diberi nilai *false*.

```

1.     public override void GenerateMaze(){
2.         for (int row = 0; row < RowCount; row++) {
3.             GetMazeCell(row,0).WallLeft = true;
4.             GetMazeCell(row,ColumnCount-1).WallRight=
           true;
5.         }
6.         for (int column = 0; column < ColumnCount;
           column++) {
7.             GetMazeCell(0,column).WallBack = true;

```

```

8.             GetMazeCell (RowCount-1,column).WallFront
               = true;
9.             }
10.
11.            rectsToDivide.Enqueue (new IntRect (){left =
               0, right = ColumnCount, bottom = 0, top = RowCount});
12.            while (rectsToDivide.Count > 0) {
13.                IntRect currentRect =
               rectsToDivide.Dequeue ();
14.                int width = currentRect.right -
               currentRect.left;
15.                int height = currentRect.top -
               currentRect.bottom;
16.                if(width > 1 && height > 1){
17.                    if(width > height){
18.                        divideVertical(currentRect);
19.                    }else if(height > width){
20.                        divideHorizontal(currentRect);
21.                    }else if(height == width){
22.                        if(Random.Range(0,100) > 50){
23.                            divideVertical(currentRect);
24.                        }else{
25.                            divideHorizontal(currentRect);
26.                        }
27.                    }
28.                    }else if(width > 1 && height <=1){
29.                        divideVertical(currentRect);
30.                    }else if(width <=1 && height > 1){
31.                        divideHorizontal(currentRect);
32.                    }
33.                }
34.            }
35.            private void divideHorizontal(IntRect rect){
36.                int divRow = Random.Range(rect.bottom,
               rect.top-1);
37.                for (int col = rect.left; col < rect.right;
               col++) {
38.                    GetMazeCell (divRow,col).WallFront = true;
39.                    GetMazeCell (divRow+1,col).WallBack =
               true;
40.                }

```

```

41.         int space = Random.Range (rect.left,
rect.right);
42.         GetMazeCell (divRow, space).WallFront =
false;
43.         if (divRow + 1 < rect.top) {
44.             GetMazeCell (divRow + 1, space).WallBack
= false;
45.         }
46.         rectsToDivide.Enqueue (new IntRect () {left =
rect.left, right = rect.right, bottom = rect.bottom,
top = divRow+1});
47.         rectsToDivide.Enqueue (new IntRect () {left =
rect.left, right = rect.right, bottom = divRow+1, top
= rect.top});
48.     }
49. }

50.

```

Kode Sumber 4.5 Penentuan dinding yang akan dihilangkan

Setelah menentukan dinding mana yang akan diberi lubang secara vertikal ataupun horisontal maka langkah berikutnya adalah membuat labirin berdasarkan nilai *boolean* pada dinding. Nilai *boolean* pada dinding telah ditentukan melalui proses sebelumnya. Sehingga proses ini adalah pembentukan *maze* berdasarkan nilai tersebut, yaitu dinding yang mempunyai nilai *true* akan dibuat sedangkan dinding bernilai *false* akan dihilangkan. Proses pembentukan *maze* tersebut dapat dilihat pada Kode Sumber 4.6 .

```

1.         for (int row = 0; row < Rows; row++) {
2.             for(int column = 0; column < Columns;
column++){
3.                 float x = column*(CellWidth);
4.                 float z = row*(CellHeight);
5.                 int item = Random.Range(1,3);
6.                 MazeCell cell =
mMazeGenerator.GetMazeCell(row,column);
7.                 GameObject tmp;
8.                 tmp = Instantiate(Floor,new
Vector3(x,0,z), Quaternion.Euler(0,0,0)) as
GameObject;

```

```

9.         tmp.transform.parent = transform;
10.        if(cell.WallRight){
11.            tmp = Instantiate(Wall,new
Vector3(x+CellWidth/2,0,z)+Wall.transform.position,Qu
aternion.Euler(0,90,0)) as GameObject;// right
12.            tmp.transform.parent = transform;
13.        }
14.        if(cell.WallFront){
15.            tmp = Instantiate(Wall,new
Vector3(x,0,z+CellHeight/2)+Wall.transform.position,Q
uaternion.Euler(0,0,0)) as GameObject;// front
16.            tmp.transform.parent = transform;
17.        }
18.        if(cell.WallLeft){
19.            tmp = Instantiate(Wall,new
Vector3(x-
CellWidth/2,0,z)+Wall.transform.position,Quaternion.E
uler(0,270,0)) as GameObject;// left
20.            tmp.transform.parent = transform;
21.        }
22.        if(cell.WallBack){
23.            tmp = Instantiate(Wall,new
Vector3(x,0,z-
CellHeight/2)+Wall.transform.position,Quaternion.Eule
r(0,180,0)) as GameObject;// back
24.            tmp.transform.parent = transform;
25.        }
26.        if(item == 1){
27.            tmp = Instantiate(GoalPrefab,new
Vector3(x,1,z), Quaternion.Euler(0,0,0)) as
GameObject;
28.            tmp.transform.parent = transform;
29.        }
30.    }
31. }

```

Kode Sumber 4.6 Pembentuk *Maze* secara acak

4.4.2 Implementasi Menelusuri Labirin dan Mengambil Koin

Dalam implementasi aturan permainan pada aplikasi ini pengguna diminta untuk mengumpulkan koin sebanyak

mungkin selama menelusuri labirin. Untuk mengambil koin dalam permainan tersebut pengguna harus menabrak objek koin.

Untuk pembuatan proses mengambil koin objek koin yang ditempatkan secara acak dalam labirin diberi collider dan trigger terlebih dahulu. Kemudian dicek *trigger* aktif atau tidak. Jika *trigger* aktif maka koin akan *destroy* dan score pemain akan bertambah. Pembuatan implementasi mengambil koin dapat dilihat pada Kode Sumber 4.7.

```
1.
2.     void OnTriggerEnter(Collider other) {
3.         if (other.gameObject.name.Contains ("Coin"))
4.         {
5.             Destroy (other.gameObject);
6.             ScoreM.TambahScore ();
7.         }
8.     public void TambahScore() {
9.         score =score+1;
10.        scoreText.text = "S: " + score;
11.    }
```

Kode Sumber 4.7 Implementasi Proses Mengambil Koin

4.4.3 Implementasi Memperoleh Informasi

Tampilan informasi menu akan memberikan pilihan bagi pengguna agar dapat menghentikan permainan untuk sementara dan dapat melanjutkan permainan. Untuk membuat tampilan informasi menu dilakukan dengan cara membuat objek yang mengikuti kamera dari atas. Kemudian dalam objek tersebut diberi *trigger*. Objek yang *ditrigger* akan menjalankan fungsi yang ada didalamnya. Tampilan informasi menu dapat dilihat pada Gambar 4.18. Untuk membuat objek mengikuti kamera dapat dilihat pada Kode Sumber 4.8.



Gambar 4.18 Tampilan Informasi Menu

```

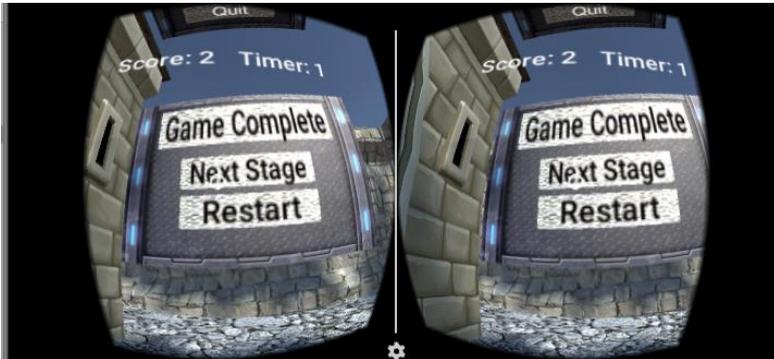
1.  using System.Collections;
2.  using System.Collections.Generic;
3.  using UnityEngine;
4.
5.  public class FindPlayer : MonoBehaviour {
6.      public Transform Player;
7.      private Vector3 playerPos;
8.      public float speed;
9.      public float x, y, z;
10.
11.     // Use this for initialization
12.     void Start () {
13.         Player =
14.         GameObject.FindWithTag("Player").transform;
15.     }
16.
17.     // Update is called once per frame
18.     void Update () {
19.         playerPos = new Vector3(Player.position.x,
20.         transform.position.y, Player.position.z + z );
21.         transform.position = Vector3.Lerp
22.         (transform.position, playerPos, Time.deltaTime *
23.         speed);
24.     }
25. }

```

Kode Sumber 4.8 Objek mengikuti kamera

4.4.4 Implementasi Menemukan Finish

Pada saat pengguna telah menemukan garis *finish* dari labirin maka permainan pada *stage* tersebut telah selesai. Pengguna akan mendapatkan pilihan untuk melanjutkan ke *stage* selanjutnya. Tampilan pilihan saat mencapai finish dapat dilihat pada Gambar 4.19. Pilihan menu tersebut hanya tampil saat pengguna mencapai *finish*. Untuk membuat menu tersebut langkah pertama adalah memberikan collider dan trigger pada objek *finish*. Kemudian cek trigger, jika aktif maka tampilkan menu. *Script* dapat dilihat pada Kode Sumber 4.9.



Gambar 4.19 Tampilan menu saat game *finish*

```
1. if (other.gameObject.name.Contains ("Finish")) {
2.     Destroy (other.gameObject);
3.     GameM.Complete ();
4. }
5.
6. public void GameOver() {
7.     GameOverScreen.SetActive (true);
8.     kalah=true;
9.     mulaiTimer = true;
10.    speedM.speed = 0.0f;
11. }
12.
13. public void Complete(){
14.     CompleteScreen.SetActive (true);
15.     menang = true;
16.     mulaiTimer = true;
17.     //speedM.speed = 0.0f;
```

```
18.     //Time.timeScale = 0.0f;
19. }
20.
21. public void NextLev () {
22.     SceneManager.LoadScene (NextLevel);
23. }
24.
25. public void Restart () {
26.     SceneManager.LoadScene (Level);
27. }
```

Kode Sumber 4.9 Implementasi menu mencapai *finish*

[Halaman ini sengaja dikosongkan]

BAB V PENGUJIAN DAN EVALUASI

Pada bab ini membahas mengenai uji coba dan evaluasi aplikasi “The Labyrinth” pada perangkat bergerak dengan teknologi Google Cardboard. Uji coba dilakukan menggunakan metode *black box* (kotak hitam), yaitu pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak berdasarkan skenario yang telah ditentukan.

5.1 Lingkungan Uji Coba

Pada proses uji coba ini, lingkungan dibedakan menjadi lingkungan perangkat keras dan perangkat lunak. Berikut ini akan dijelaskan mengenai tiap-tiap lingkungan uji coba aplikasi.

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam rangka uji coba perangkat lunak ini dicantumkan pada Tabel 5.1. Uji coba ini dilakukan menggunakan kacamata VR Box dengan *bluetooth remote controller* sebagai media input.

Tabel 5.1 Lingkungan Perangkat Keras

| No. | Deskripsi |
|-----|---|
| 1 | Model Perangkat Xiaomi Redmi Note 3 Pro Chipset : Qualcomm Snapdragon 650 CPU : Hexa-core @ 1.80GHz Memori : 3.00 GB Sistem Operasi : Android 5.1.1 API level: 19 |
| 2 | Model Perangkat Lenovo vibe k5 plus Chipset : Qualcomm MSM8939v2 Snapdragon 616 CPU : Octa-core (4x1.5 GHz Cortex-A53 & 4x1.2 GHz Cortex-A53) Memori : 2.00 GB Sistem Operasi : Android 5.1 |

5.2 Skenario dan Hasil Uji Coba

Pada subbab ini dijelaskan mengenai skenario yang dilakukan untuk mendapatkan hasil uji coba aplikasi. Skenario uji coba aplikasi dilakukan untuk mengetahui ketercapaian aplikasi dalam memenuhi kebutuhan fungsionalitas dan kebutuhan non-fungsionalitas.

5.2.1 Pengujian Fungsionalitas

Pengujian fungsionalitas aplikasi ini dapat dilakukan secara mandiri. Pengujian ini bertujuan untuk mengetahui kesesuaian keluaran dari tiap tahap dan langkah penggunaan fitur terhadap skenario yang dipersiapkan. Skenario yang dibuat mengacu pada kasus penggunaan yang telah dijelaskan pada subbab 3.6.2. Skenario uji coba fungsionalitas yang dilakukan terhadap aplikasi yang dibangun dijelaskan pada Tabel 5.2.

Tabel 5.2 Skenario Uji Coba Fungsionalitas

| Kode Uji Coba | Nama Uji Coba |
|---------------|----------------------------|
| UJ-UC-001 | Uji Memilih Stage |
| UJ-UC-002 | Uji Menelusuri Tempat |
| UJ-UC-003 | Uji Melihat Informasi |
| UJ-UC-004 | Uji Menemukan Finish |
| UJ-UC-005 | Uji Pembangkit <i>Maze</i> |

Setiap skenario akan dijelaskan mengenai kondisi awal, masukkan, dan keluaran yang diharapkan, kondisi akhir, dan hasil uji coba. Berikut ini merupakan penjabaran hasil setiap uji coba yang dilakukan.

5.2.1.1 Uji Memilih Stage

Uji coba ini bertujuan untuk membawa pengguna memasuki stage yang ingin dimainkan. Skenario dan hasil pengujian secara lengkap dapat dilihat pada Tabel 5.3.

Tabel 5.3 Hasil Uji Coba Memilih Lokasi

| Kode Uji Coba | UJ-UC-001 |
|--|---|
| Kondisi Awal | Aplikasi menampilkan tiga pilihan stage |
| Skenario 1 (Pengguna mengarahkan reticle ke pilihan stage) | |
| Masukan | Mengarahkan <i>reticle</i> ke kotak pilihan |
| Keluaran yang diharapkan | Skala kotak pilihan membesar |
| Kondisi Akhir | Skala kotak pilihan membesar |
| Hasil Uji Coba | Berhasil |
| Skenario 2 (Pengguna menekan trigger di controller) | |
| Masukan | Menekan <i>trigger</i> di <i>controller</i> |
| Keluaran yang diharapkan | Membuka <i>scene stage</i> yang dipilih |
| Kondisi Akhir | Membuka <i>scene stage</i> yang dipilih |
| Hasil Uji Coba | Berhasil |

5.2.1.2 Uji Menelusuri Tempat

Uji coba ini bertujuan untuk mengarahkan pengguna berkeliling dalam realitas virtual labirin untuk menemukan garis *finish* yang ditentukan. Skenario dan hasil pengujian secara lengkap dapat dilihat pada Tabel 5.4.

Tabel 5.4 Hasil Uji Coba Menelusuri Tempat

| Kode Uji Coba | UJ-UC-002 |
|---|--|
| Kondisi Awal | Proses <i>load</i> objek sudah selesai |
| Skenario 1 (Pengguna mengarahkan <i>smartphone</i> ke atas, bawah, kanan, kiri, dan belakang) | |
| Masukan | Mengarahkan <i>smartphone</i> ke sembarang arah |
| Keluaran yang diharapkan | Layar <i>smartphone</i> menampilkan objek 3D sesuai arah <i>smartphone</i> |

| | |
|--|--|
| Kondisi Akhir | Layar <i>smartphone</i> menampilkan objek 3D sesuai arah <i>smartphone</i> |
| Hasil Uji Coba | Berhasil |
| Skenario 2 (Pegguna mengarahkan <i>smartphone</i> kebawah dalam kondisi diam) | |
| Masukan | Mengarahkan kamera ke arah lantai |
| Keluaran yang diharapkan | Kamera bergerak ke depan |
| Kondisi Akhir | Kamera bergerak ke depan |
| Hasil Uji Coba | Berhasil |
| Skenario 3 (Pegguna mengarahkan <i>smartphone</i> keatas dalam kondisi bergerak) | |
| Masukan | Mengarahkan kamera ke arah selain lantai |
| Keluaran yang diharapkan | Kamera berhenti |
| Kondisi Akhir | Kamera berhenti |
| Hasil Uji Coba | Berhasil |

5.2.1.3 Uji Melihat Informasi

Uji coba ini bertujuan untuk memberi pengguna informasi mengenai objek-objek dalam peraturan permainan. Skenario dan hasil pengujian secara lengkap dapat dilihat pada Tabel 5.5.

Tabel 5.5 Hasil Uji Coba Melihat Informasi

| Kode Uji Coba | UJ-UC-003 |
|---|---|
| Kondisi Awal | Proses <i>load</i> objek sudah selesai |
| Skenario 1 (Pegguna menabrak objek berupa koin) | |
| Masukan | Pegguna menabrak objek koin |
| Keluaran yang diharapkan | Layar <i>smartphone</i> menampilkan perubahan informasi teks pada score |
| Kondisi Akhir | Layar <i>smartphone</i> menampilkan perubahan informasi teks pada score |
| Hasil Uji Coba | Berhasil |

| Skenario 2 (Pegguna menelusuri labirin, timer tetap berjalan) | |
|--|---|
| Masukan | Mengarahkan <i>smartphone</i> ke sembarang arah |
| Keluaran yang diharapkan | Layar <i>smartphone</i> menampilkan perubahan informasi teks pada timer |
| Kondisi Akhir | Layar <i>smartphone</i> menampilkan perubahan informasi teks pada timer |
| Hasil Uji Coba | Berhasil |
| Skenario 3 (Pegguna menelusuri labirin, batas waktu (timer) telah habis) | |
| Masukan | Mengarahkan <i>smartphone</i> ke sembarang arah |
| Keluaran yang diharapkan | Muncul keterangan <i>Game Over</i> kemudian membuka <i>scene stage</i> yang dipilih |
| Kondisi Akhir | Muncul keterangan <i>Game Over</i> kemudian membuka <i>scene stage</i> yang dipilih |
| Hasil Uji Coba | Berhasil |

5.2.1.4 Uji Menemukan *Finish*

Uji coba ini bertujuan untuk memberi pengguna informasi pilihan menu untuk menuju ke stage berikutnya. Skenario dan hasil dapat dilihat pada Tabel 5.6.

Tabel 5.6 Hasil Uji Coba Menemukan Finish

| Kode Uji Coba | UJ-UC-004 |
|--|---|
| Kondisi Awal | Proses <i>load</i> objek sudah selesai |
| Skenario 1 (Pegguna berjalan menelusuri labirin) | |
| Masukan | Pegguna menabrak objek <i>finish</i> |
| Keluaran yang diharapkan | Layar <i>smartphone</i> menampilkan informasi pilhan menu |
| Kondisi Akhir | Layar <i>smartphone</i> menampilkan informasi pilhan menu |

| | |
|--|---|
| Hasil Uji Coba | Berhasil |
| Skenario 2 (<i>Pengguna memilih salah satu pilihan menu</i>) | |
| Masukan | Menekan <i>trigger</i> di <i>controller</i> ke arah menu yang dipilih |
| Keluaran yang diharapkan | Masuk ke <i>scene stage</i> pada menu yang di pilih |
| Kondisi Akhir | Masuk ke <i>scene stage</i> pada menu yang di pilih |
| Hasil Uji Coba | Berhasil |

5.2.1.5 Uji Pembangkit *Maze* atau Labirin

Uji coba ini bertujuan untuk menunjukkan kepada pengguna setiap kali mengulang satu stage yang sama, bentuk labirin atau *maze* akan selalu berubah. Skenario dan hasil dapat dilihat pada Tabel 5.7.

Saat pengujian labirin juga dilihat labirin berubah dan mempunyai jalan menuju finish. Dapat dilihat pada lampiran B Screenshot Aplikasi Gambar B.7, Gambar B.8, Gambar B.9, Gambar B.10, Gambar B.11, Gambar B.12.

Tabel 5.7 Hasil Uji Coba Pembangkit *Maze*

| Kode Uji Coba | UJ-UC-005 |
|--|---|
| Kondisi Awal | Proses <i>load</i> objek sudah selesai |
| Skenario 1 (<i>Pengguna mencapai finish</i>) | |
| Masukan | Pengguna mentrigger objek menu <i>restart</i> |
| Keluaran yang diharapkan | Mengulang <i>scene stage</i> yang sama dengan bentuk labirin yang berbeda |
| Kondisi Akhir | Mengulang <i>scene stage</i> yang sama dengan bentuk labirin yang berbeda |
| Hasil Uji Coba | Berhasil |
| Skenario 2 (<i>Pengguna mengalami kehabisan waktu</i>) | |
| Masukan | Mengarahkan <i>smartphone</i> ke sembarang arah |

| | |
|--------------------------|---|
| Keluaran yang diharapkan | Menampilkan notifikasi <i>Game Over</i> dan Mengulang <i>scene stage</i> yang sama dengan bentuk labirin yang berbeda |
| Kondisi Akhir | Menampilkan notifikasi <i>Game Over</i> dan Mengulang <i>scene stage</i> yang sama dengan bentuk labirin yang berbeda |
| Hasil Uji Coba | Berhasil |

5.2.2 Pengujian Non-Fungsionalitas

Pengujian non-fungsionalitas terhadap aplikasi ini dilakukan dengan uji aspek operasional yang dilakukan sendiri oleh penulis. Pengujian ini bertujuan untuk mengetahui seberapa besar perbedaan waktu *load* objek pada perangkat yang berbeda. Skenario yang dilakukan adalah dengan menjalankan permainan labirin pada aplikasi dengan mencatat waktu tempuh permainan. Untuk lebih jelasnya, skenario uji coba dapat dilihat pada Tabel 5.8.

Tabel 5.8 Skenario Non-Fungsionalitas Aspek Operasional

| No | Kode Uji Coba | Deskripsi |
|----|---------------|-------------------|
| 1 | UJ-P-01 | Memainkan stage 1 |
| 2 | UJ-P-02 | Memainkan stage 2 |
| 3 | UJ-P-03 | Memainkan stage 3 |

Selain itu, juga dilakukan pengujian oleh satu responden yang sudah pernah mencoba permainan dengan jenis labirin dan sebelumnya sudah mencoba aplikasi. Kemudian setiap responden mengisi kuesioner yang sudah disediakan penulis. Pengujian ini bertujuan untuk mengetahui kenyamanan antarmuka menu, kemiripan objek, kegunaan informasi, performa sistem, dan perasaan *immersive* yang dirasakan pengguna. Daftar nama dan latar belakang penguji dapat dilihat pada Tabel 5.9.

Tabel 5.9 Daftar Penguji Coba Aplikasi

| Kode | Nama | Pekerjaan | Pengalaman Menggunakan |
|------|------|-----------|------------------------|
|------|------|-----------|------------------------|

| Google Cardboard | | | |
|---------------------|---------------------|-----------|--------|
| P1 | Guruh Arya Senna | Mahasiswa | Pernah |
| P2 | Fahmy T H | Mahasiswa | Pernah |
| P3 | Tikva I Mooy | Mahasiswa | Pernah |
| P4 | I.G.N Adi Wicaksana | Mahasiswa | Pernah |
| P5 | Setiyo A | Mahasiswa | Pernah |
| P6 | Naufal BF | Mahasiswa | Pernah |

Metode kuesioner yang digunakan adalah dengan metode skala lakert. Skala Lakert merupakan metode pengukuran yang digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau kelompok orang tentang fenomena sosial [12]. Jawaban dari kuesioner yang digunakan memiliki skala 6 dengan pemberian skor seperti pada Tabel 5.10.

Tabel 5.10 Pemberian Skor Jawaban

| Keterangan | Skor |
|---------------------|------|
| Sangat tidak setuju | 1 |
| Tidak setuju | 2 |
| Kurang Setuju | 3 |
| Cukup Setuju | 4 |
| Setuju | 5 |
| Sangat Setuju | 6 |

Untuk menganalisis hasil kuesioner yang diisi oleh responden digunakan rumus untuk mengetahui jumlah hasil nilai setiap pertanyaan. Rumus tersebut adalah sebagai berikut

$$\sum_{i=1}^n T \times P_i$$

n = skala maksimal
 T = jumlah responden
 P_i = Skor jawaban ke- i

Setelah jumlah hasil nilai diketahui, maka akan diketahui tingkat persetujuan responden dari *Rating Scale*. Untuk mengetahui *Rating Scale* setiap bobot skor maka diperlukan

penghitungan skor ideal dari semua bobot skor. Skor ideal dapat diketahui dengan rumus $skor \times Jumlah Responden$. Sehingga *Rating Scale* dapat dilihat pada Tabel 5.11 berikut.

Tabel 5.11 *Rating Scale*

| Nilai Jawaban | Skala |
|---------------|---------------------|
| 31 - 36 | Sangat Setuju |
| 25 - 30 | Setuju |
| 19 - 24 | Cukup Setuju |
| 13 - 18 | Kurang Setuju |
| 7 - 12 | Tidak Setuju |
| 0 - 6 | Sangat Tidak Setuju |

Pembuatan skenario uji coba non-fungsionalitas yang dituangkan dalam kuesioner akan dijabarkan pada Tabel 5.12.

Tabel 5.12 Indikator Penilaian Non-Fungsionalitas

| No | Penilaian | Indikator |
|----|--------------------|--|
| 1 | Antarmuka | Kenyamanan antarmuka menu |
| 2 | Kemiripan Objek | Kemiripan model labirin |
| | | Kemiripan objek koin |
| | | Kemiripan objek gerbang garish <i>finish</i> |
| 3 | Kegunaan Informasi | Kejelasan teks informasi |
| | | Kejelasan suara audio |
| | | Kejelasan simulasi animasi |
| 4 | Performa Sistem | <i>Real time rendering</i> |
| 5 | Realitas Virtual | Perasaan <i>immersive</i> |

Pada subbab berikutnya akan dijelaskan mengenai hasil uji coba non-fungsionalitas berdasarkan uji coba pada perangkat berbeda dan kuesioner responden.

5.2.2.1 Pengujian Operasional

Penilaian ini digunakan untuk mengetahui perbedaan lama waktu pengoperasian dan rekomendasi

spesifikasi perangkat untuk menjalankan aplikasi. Hasil uji coba dapat dilihat pada Tabel 5.13 berikut ini.

Tabel 5.13 Hasil Pengujian Operasional

| No | Uji Coba | Perangkat | Waktu (detik) |
|----|----------|-------------|---------------|
| 1 | UJ-P-01 | Perangkat 1 | 59 |
| 2 | | Perangkat 2 | 67 |
| 3 | UJ-P-02 | Perangkat 1 | 143 |
| 4 | | Perangkat 2 | 137 |
| 5 | UJ-P-03 | Perangkat 1 | 195 |
| 6 | | Perangkat 2 | 211 |

5.2.2.2 Penilaian Antarmuka

Penilaian antarmuka adalah penilaian pengguna terhadap aplikasi apakah antarmuka menu aplikasi mudah terlihat dan dipahami. Hasil penilaian antarmuka oleh responden selengkapnya dijabarkan pada Tabel 5.14.

Tabel 5.14 Hasil Penilaian Antarmuka

| No | Indikator | Penilaian | | | | | | Nilai |
|----|---------------------------|-----------|---|---|---|---|---|-------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | |
| 1 | Kenyamanan antarmuka menu | | | 1 | 2 | 2 | 1 | 27 |

5.2.2.3 Penilaian Kemiripan Objek

Penilaian kemiripan objek merupakan penilaian terhadap kemiripan objek dalam lingkungan virtual dengan objek aslinya. Hasil penilaian kemiripan objek oleh responden dijabarkan pada Tabel 5.15.

Tabel 5.15 Hasil Penilaian Kemiripan Objek

| No | Indikator | Penilaian | | | | | | Nilai |
|----|-------------------------|-----------|---|---|---|---|---|-------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | |
| 1 | Kemiripan model labirin | | | 2 | 1 | 3 | | 25 |
| 2 | Kemiripan objek koin | | | | 3 | 3 | | 27 |

| | | | | | | | | |
|------------------------|--|--|--|---|---|---|--|--------------|
| 3 | Kemiripan objek gerbang garish <i>finish</i> | | | 1 | 3 | 2 | | 25 |
| Nilai rata-rata | | | | | | | | 25,67 |

5.2.2.4 Penilaian Kegunaan Informasi

Penilaian kegunaan informasi adalah penilaian terhadap kemudahan dan kejelasan informasi pada aplikasi. Informasi tersebut meliputi informasi berupa tulisan yang dibaca dan audio yang didengar. Selain itu juga terdapat penilaian terhadap tambahan informasi berupa animasi dan simulasi. Hasil penilaian kegunaan informasi oleh responden selengkapnya dijabarkan pada Tabel 5.16.

Tabel 5.16 Hasil Penilaian Kegunaan Informasi

| No | Indikator | Penilaian | | | | | | Nilai |
|------------------------|--------------------------|-----------|---|---|---|---|---|-----------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | |
| 1 | Kejelasan teks informasi | | | 1 | 3 | 2 | | 25 |
| 2 | Kejelasan animasi | | | | 4 | 3 | | 31 |
| Nilai rata-rata | | | | | | | | 28 |

5.2.2.5 Penilaian Performa Sistem

Penilaian performa sistem merupakan penilaian terhadap kecepatan perangkat dalam memproses model yang dirasakan oleh pengguna. Pada penilaian ini perangkat yang digunakan adalah perangkat pertama. Hasil penilaian performa sistem oleh responden dijabarkan pada Tabel 5.17.

Tabel 5.17 Hasil Penilaian Performa Sistem

| No | Indikator | Penilaian | | | | | | Nilai |
|----|----------------------------|-----------|---|---|---|---|---|-------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | |
| 1 | <i>Real time rendering</i> | | | | 3 | 3 | | 27 |

5.2.2.6 Penilaian Realitas Virtual

Penilaian realitas virtual merupakan penilaian sensasi seolah-olah berada di dalam labirin yang dirasakan

pengguna. Hasil penilaian realitas virtual oleh responden dijabarkan pada Tabel 5.18.

Tabel 5.18 Hasil Penilaian Realitas Virtual

| No | Indikator | Penilaian | | | | | | Nilai |
|----|---------------------------|-----------|---|---|---|---|---|-----------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | |
| 1 | Perasaan <i>immersive</i> | | | 1 | 4 | 1 | | 24 |

5.3 Evaluasi Pengujian

Pada subbab evaluasi pengujian ini akan ditunjukkan data rekapitulasi dari hasil pengujian fungsionalitas maupun hasil pengujian non-fungsionalitas yang telah dilakukan sebelumnya pada subbab 5.2. Rekapitulasi masing-masing pengujian akan dijabarkan pada subbab berikut.

5.3.1 Evaluasi Pengujian Fungsionalitas

Evaluasi pengujian fungsionalitas dilakukan dengan menampilkan data rekapitulasi aplikasi yang telah dipaparkan pada subbab 5.2.1 Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Tabel 5.19. Dari data yang terdapat pada tabel tersebut, diketahui bahwa aplikasi yang dibuat telah memenuhi kasus penggunaan yang telah ditentukan.

Tabel 5.19 Rekapitulasi Hasil Uji Fungsionalitas

| Kode | Uji Coba | Hasil | |
|-----------|-----------------------------|------------|----------|
| UJ-UC-001 | Uji Memilih <i>Stage</i> | Skenario 1 | Berhasil |
| | | Skenario 2 | Berhasil |
| UJ-UC-002 | Uji Menelusuri Tempat | Skenario 1 | Berhasil |
| | | Skenario 2 | Berhasil |
| | | Skenario 3 | Berhasil |
| UJ-UC-003 | Uji Melihat Informasi | Skenario 1 | Berhasil |
| | | Skenario 2 | Berhasil |
| | | Skenario 3 | Berhasil |
| UJ-UC-004 | Uji Menemukan <i>Finish</i> | Skenario 1 | Berhasil |
| | | Skenario 2 | Berhasil |

| | | | |
|-----------|----------------|------------|----------|
| UJ-UC-005 | Uji Pembangkit | Skenario 1 | Berhasil |
| | <i>Maze</i> | Skenario 2 | Berhasil |

5.3.2 Evaluasi Pengujian Non-Fungsionalitas

Evaluasi pengujian fungsionalitas dilakukan dengan analisis hasil uji coba pengujian operasional dan kuesioner responden. Dengan melihat grafik perbandingan waktu pengoperasian pada ketiga perangkat dapat diketahui membentuk grafik yang hampir berimpitan. Sehingga hampir tidak ada perbedaan signifikan untuk mengoperasikan aplikasi pada ketiga perangkat.

Tabel 5.20 Rekapitulasi Hasil Uji Coba Non-Fungsionalitas

| Penilaian | Indikator | Hasil |
|------------------------------------|--|--------------------|
| Antarmuka | Kenyamanan antarmuka menu | 27 (setuju) |
| Kemiripan Objek | Kemiripan model labirin | 25 (setuju) |
| | Kemiripan objek koin | 27 (setuju) |
| | Kemiripan objek gerbang garish <i>finish</i> | 25 (setuju) |
| Hasil Penilaian Kemiripan Objek | | |
| Kegunaan Informasi | Kejelasan teks informasi | 31 (sangat setuju) |
| | Kejelasan animasi | 22 (cukup setuju) |
| Hasil Penilaian Kegunaan Informasi | | |
| Performa Sistem | <i>Real time rendering</i> | 27 (setuju) |
| Realitas Virtual | Perasaan <i>immersive</i> | 25 (setuju) |

Berdasarkan

Tabel 5.20 dapat diketahui penilaian kenyamanan antarmuka oleh penguji mencapai 75% (responden setuju, dengan nilai 27 dari 36), kemiripan objek mencapai 71,30% (responden setuju, dengan nilai rata-rata 25,67 dari 36),

kegunaan informasi mencapai 77,78% (responden setuju, dengan nilai rata-rata 28 dari 36), performa sistem mencapai nilai 75% (responden sangat setuju, dengan nilai 27 dari 36) dan perasaan *immersive* mencapai nilai 69,5% (responden cukup setuju, dengan nilai 24 dari 36).

BAB VI KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang didapatkan dari uji tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan. Selain kesimpulan, terdapat pula saran-saran untuk pengembangan perangkat lunak lebih lanjut.

6.1. Kesimpulan

Dari proses pengerjaan tugas akhir selama tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Berdasarkan pengujian fungsional, *maze generator* dapat membangkitkan labirin yang berbeda – beda setiap memainkan suatu level permainan.
2. Interaksi pengguna pada realitas virtual menggunakan Google Cardboard dapat melalui *trigger* perangkat eksternal yang berfungsi sebagai perintah “klik” dan interaksi tumbukan antara kamera dengan objek 3D.
3. Berdasarkan hasil uji operasional diketahui bahwa perbedaan waktu dalam menjalankan aplikasi pada perangkat bergerak yang memiliki CPU quad-core dan memori 2GB dengan CPU octa-core dan memori 3GB tidak memiliki perbedaan terlalu signifikan, sehingga aplikasi dapat beroperasi dengan baik pada perangkat dengan spesifikasi yang tidak begitu tinggi yaitu dengan CPU quad-core dan memori 2GB meskipun aplikasi menerapkan objek 3D dan *virtual reality*.
4. Untuk membuat objek 3D yang mempunyai ukuran file yang lebih kecil dapat dilakukan dengan membuat keseluruhan objek terlebih dahulu, selanjutnya dilakukan identifikasi ekstra *edge* yang dapat dihapus.

6.2. Saran

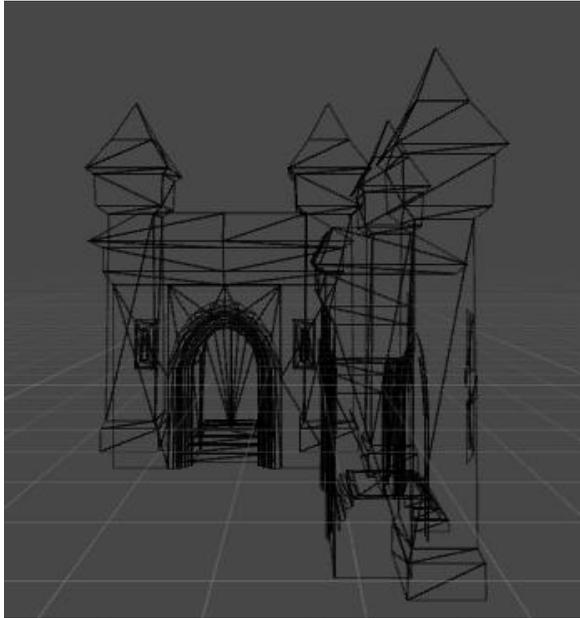
Berikut merupakan beberapa saran untuk perbaikan dan pengembangan aplikasi dimasa yang akan datang:

1. Aturan permainan dibuat lebih kompleks agar permainan menjadi lebih menarik.
2. Diberikan minimap pada saat permainan berlangsung.

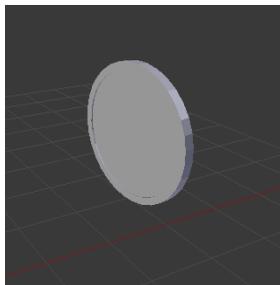
DAFTAR PUSTAKA

- [1] Haryanov, Aditya, "Prototipe Game Maze Chaser dengan Algoritma A*" Teknik Informatika STMIK AMIKOM, Yogyakarta.
- [2] S. Mandal, "Brief Introduction of Virtual Reality & its Challenges," *Int. J. Sci. Eng. Res.*, vol. 4, no. 4, 2013.
- [3] "Requirements for using Cardboard - Google Cardboard Help." [Daring]. Tersedia pada: <https://support.google.com/cardboard/answer/6295091?hl=en>. [Diakses: 14-Des-2016].
- [4] "Maze Generation: Recursive Division" 2011. [Online] Available: <http://webblog.jamisbuck.org/2011/1/12/maze-generation-recursive-division-algorithm>. [Accessed 1 Desember 2016].
- [5] Olga, "Polygonal 3D Modeling Techniques," *3D-Ace Studio*, 19-Jan-2012. [Daring]. Tersedia pada: <https://3d-ace.com/press-room/articles/polygonal-3d-modeling-techniques>. [Diakses: 20-Des-2016].
- [6] R. Pérez Fernández dan V. Alonso, "Virtual Reality in a shipbuilding environment," *Adv. Eng. Softw.*, vol. 81, hal. 30–40, Mar 2015.
- [7] Rancang Bangun Aplikasi Permainan Edukasi Berbasis Virtual Reality Menggunakan Google Cardboard," 2016. [Online]. Available: <http://is.its.ac.id/pubs/oajis/index.php/home/detail/1612/rancang-bangun-aplikasi-permainan-edukasi-berbasis-virtual-reality-menggunakan-google-cardboard>. [Accessed 12 December 2016].
- [8] "Google VR," *Google Developers*. [Daring]. Tersedia pada: <https://developers.google.com/vr/>. [Diakses: 14-Des-2016].
- [9] Habuc, Ciprian. 2009. "Shortest Path". Master student in Computer Science Departement, " Babeş-Bolyai" University, Cluj-Napoca, Romania.

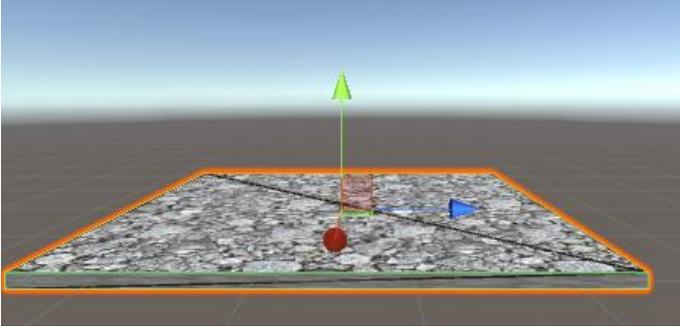
LAMPIRAN A PEMBUATAN MODEL 3D



Gambar A.1 Rancangan Gerbang Finish

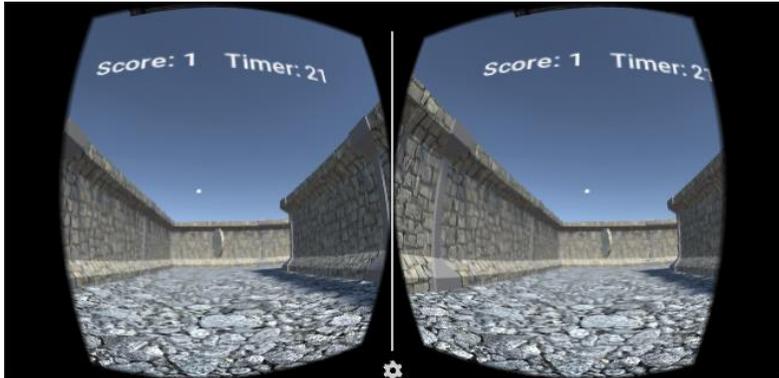


Gambar A.2 Rancangan Koin

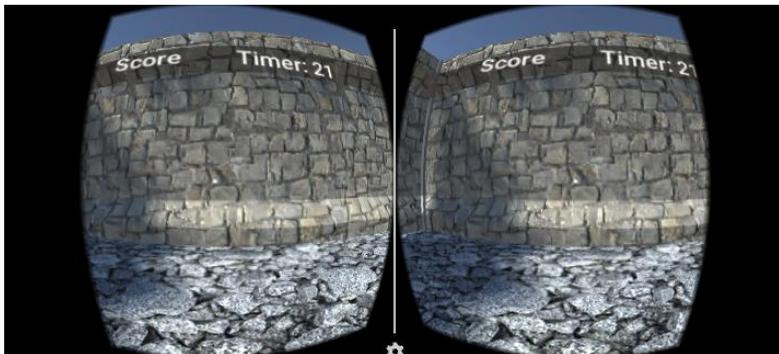


Gambar A.3 Pemberian Tektur pada Lantai

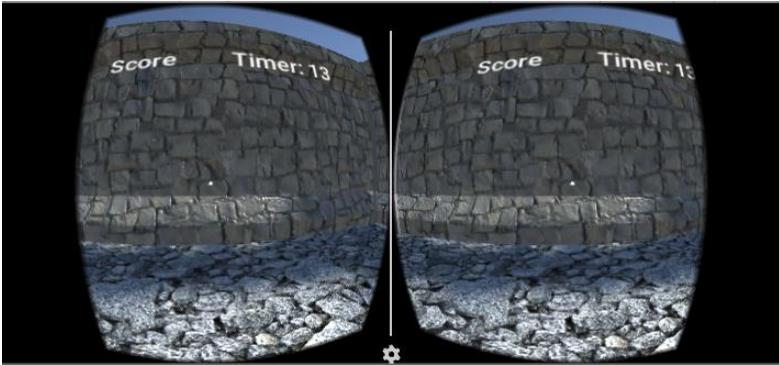
LAMPIRAN B SCREENSHOT APLIKASI



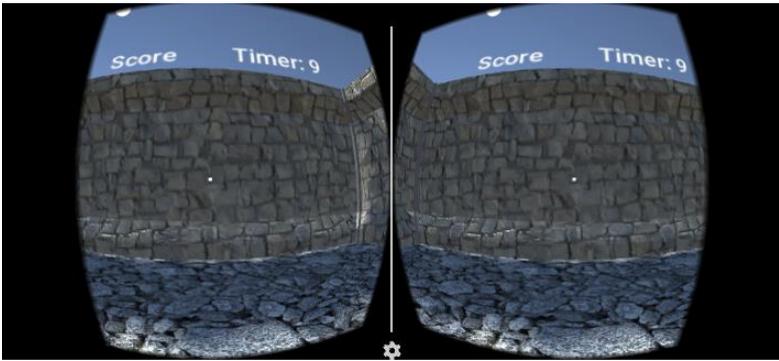
Gambar B.1 Tampilan *Game* saat menhadap ke depan



Gambar B.2 Tampilan *Game* saat menhadap ke kiri



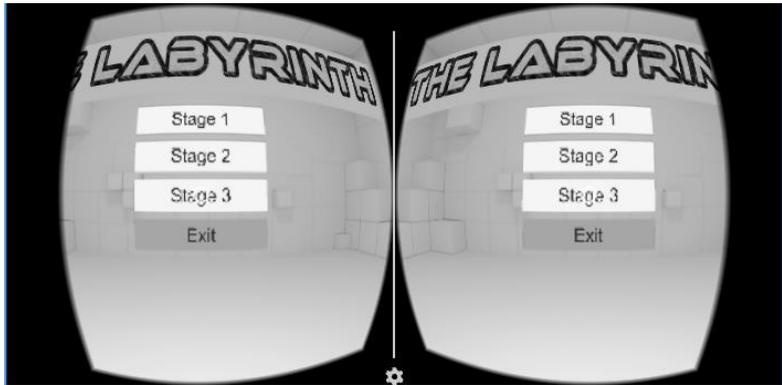
Gambar B.3 Tampilan *Game* saat menhadap ke kanan



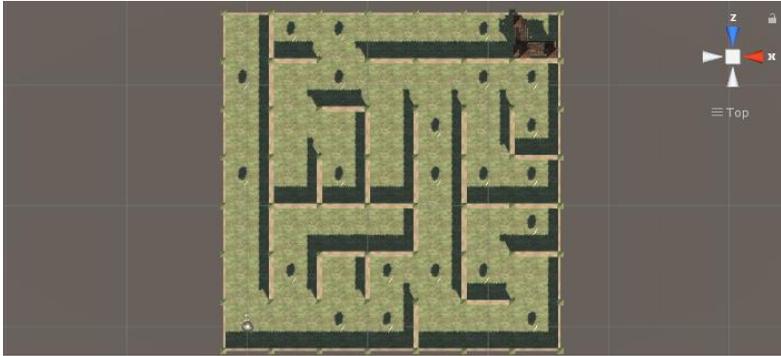
Gambar B.4 Tampilan *Game* saat menhadap ke belakang



Gambar B.5 Tampilan *Game* saat menhadap ke atas



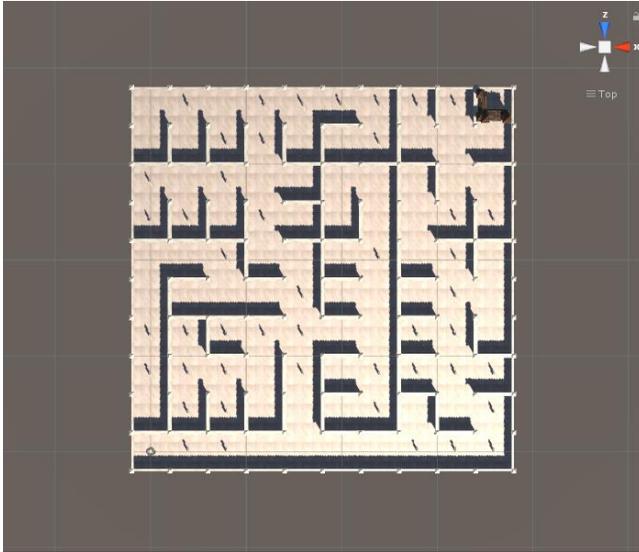
Gambar B.6 Tampilan Menu Utama



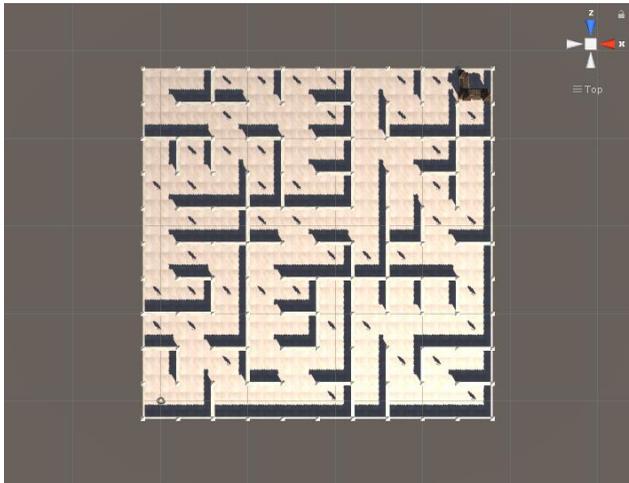
Gambar B.9 Tampilan *stage 2* awal



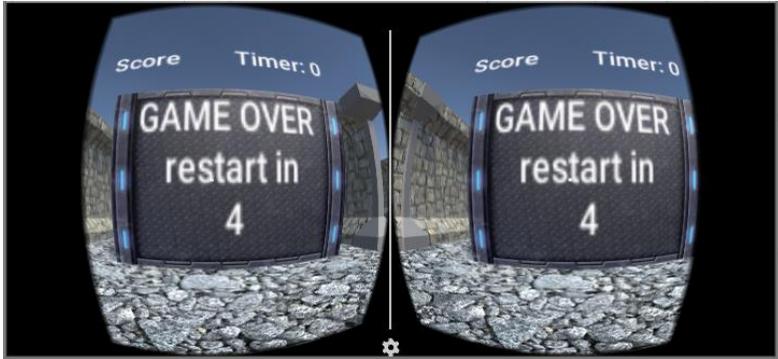
Gambar B.10 Tampilan *stage 2* saat *maze* berubah



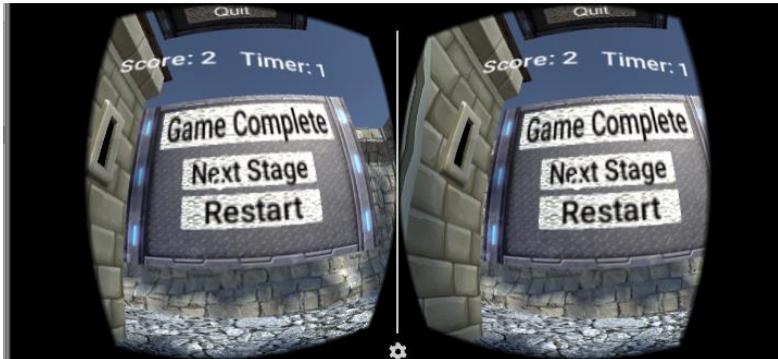
Gambar B.11 Tampilan *stage 3* awal



Gambar B.12 Tampilan *stage 3* saat *maze* berubah



Gambar B.13 Tampilan saat *Game Over*



Gambar B.14 Tampilan saat *Game Complete*

[Halaman ini sengaja dikosongkan]

LAMPIRAN C KUESIONER RESPONDEN




Kuesioner Tugas Akhir "The Labyrinth" Menggunakan Algoritma Recursive Division

Nama : Guruh Arya Senna
 NRP : 51502010
 Pekerjaan : Mahasiswa

Silahkan centang pada kolom yang sesuai berikut ini :

| No. | Pertanyaan | Jawab | | | | | |
|-----|---|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Apakah anda pernah menggunakan aplikasi virtual reality sebelumnya? | P | E | R | A | A | H |
| 2 | Apakah anda pernah memainkan aplikasi bergenre labirin sebelumnya? | P | E | R | A | A | H |

| No. | Pertanyaan | Jawab | | | | | |
|-----|--|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Objek objek dalam labirin sudah terlihat nyata seperti aslinya | | | | | ✓ | |
| 2 | Aplikasi memiliki tampilan dan desain yang menarik | | | | | ✓ | |
| 3 | Render model/objek tidak patah patah | | | | ✓ | | |
| 4 | Apakah aplikasi mudah dipahami? | | | | | ✓ | |
| 5 | Apakah tombol-tombol kontrol pada permainan tidak membingungkan? | | | | | | ✓ |
| 6 | Aplikasi sudah nyaman digunakan | | | | ✓ | | |

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
 tinggi karakter lebih baik ditambah

Surabaya, 4 Juni 2017



Gambar C.1 Kuesioner Responden Guruh Arya Senna

ITS
Institut Teknologi Sepuluh Nopember

Kuesioner Tugas Akhir "The Labyrinth" Menggunakan Algoritma Recursive Division

Nama : I G N Ad. Wicaksana
 NRP : 5113100110
 Pekerjaan : Mahasiswa

Silahkan centang pada kolom yang sesuai berikut ini :

| No. | Pertanyaan | Jawab | | | | | |
|-----|---|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Apakah anda pernah menggunakan aplikasi virtual reality sebelumnya? | P | E | R | M | A | H |
| 2 | Apakah anda pernah memainkan aplikasi bergenre labirin sebelumnya? | P | E | P | N | A | H |

| No. | Pertanyaan | Jawab | | | | | |
|-----|--|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Objek objek dalam labirin sudah terlihat nyata seperti aslinya | | | | | ✓ | |
| 2 | Aplikasi memiliki tampilan dan desain yang menarik | | | | | ✓ | |
| 3 | Render model/objek tidak patah patah | | | | ✓ | | |
| 4 | Apakah aplikasi mudah dipahami? | | | | ✓ | | |
| 5 | Apakah tombol-tombol kontrol pada permainan tidak membingungkan? | | | | | | ✓ |
| 6 | Aplikasi sudah nyaman digunakan | | | | | | ✓ |

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
 Tambahan user guide, sudah baik tambah monetar

Surabaya, 11 Juni 2019



Gambar C.2 Kuesioner Responden Adi Wicaksana

Kuesioner Tugas Akhir "The Labyrinth" Menggunakan Algoritma Recursive Division

Nama : Naufal B Farhan
 NRP : 515 200 175
 Pekerjaan : Mahasiswa

Silahkan centang pada kolom yang sesuai berikut ini :

| No. | Pertanyaan | Jawab | | | | | |
|-----|---|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Apakah anda pernah menggunakan aplikasi virtual reality sebelumnya? | P | E | R | M | A | H |
| 2 | Apakah anda pernah memainkan aplikasi bergenre labirin sebelumnya? | P | E | R | M | A | H |

| No. | Pertanyaan | Jawab | | | | | |
|-----|--|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Objek objek dalam labirin sudah terlihat nyata seperti aslinya | | | | | ✓ | |
| 2 | Aplikasi memiliki tampilan dan desain yang menarik | | | | | ✓ | |
| 3 | Render model/objek tidak patah patah | | | | | ✓ | |
| 4 | Apakah aplikasi mudah dipahami? | | | | | | ✓ |
| 5 | Apakah tombol-tombol kontrol pada permainan tidak membingungkan? | | | | | | ✓ |
| 6 | Aplikasi sudah nyaman digunakan | | | | ✓ | | |

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:

• Karena player sudah mencapai score, ketika selanjutnya pindah scene langsung akan ke 0 mengumpulkan coin lagi, dan dibuatnya game gratis (Free Edition) tidak bisa diinstall dan juga
 • Ada di pertengahan high score agar score tidak rendah Surabaya, 9 Juli 2017
 • Karena akan lebih mudah.

Naufal B Farhan

Gambar C.3 Kuesioner Responden Naufal B F

Kuisloner Tugas Akhir "The Labyrinth" Menggunakan Algoritma Recursive Division"

Nama : Fahmy Thoriqy Haq

NRP : 5112100037

Pekerjaan : Freelance

Silahkan centang pada kolom yang sesuai berikut ini :

| No. | Pertanyaan | Jawab | | | | | |
|-----|---|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Apakah anda pernah menggunakan aplikasi virtual reality sebelumnya? | 1 | 0 | 1 | 1 | 0 | 1 |
| 2 | Apakah anda pernah memainkan aplikasi bergenre labirin sebelumnya? | 0 | 0 | 1 | 1 | 1 | 1 |

| No. | Pertanyaan | Jawab | | | | | |
|-----|--|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Objek objek dalam labirin sudah terlihat nyata seperti aslinya | | | | ✓ | | |
| 2 | Aplikasi memiliki tampilan dan desain yang menarik | | | | ✓ | | |
| 3 | Render model/objek tidak patah patah | | | | ✓ | | |
| 4 | Apakah aplikasi mudah dipahami? | | | | | ✓ | |
| 5 | Apakah tombol-tombol kontrol pada permainan tidak membingungkan? | | | ✓ | | | |
| 6 | Aplikasi sudah nyaman digunakan | | | | ✓ | | |

Keterangan :

1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:

Menambah texture dan variasi terrain (mungkin obstacle)

Surabaya, 4 Juni 2017


(Fahmy T.H.)

Gambar C.4 Kuesioner Responden Fahmy




Kuisisioner Tugas Akhir "The Labyrinth" Menggunakan Algoritma Recursive Division

Nama : Setyo A.
 NRP : 51131006220
 Pekerjaan : Mahasiswa

Silahkan centang pada kolom yang sesuai berikut ini :

| No | Pertanyaan | Jawab | | | | | |
|----|---|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Apakah anda pernah menggunakan aplikasi virtual reality sebelumnya? | P | E | R | N | A | H |
| 2 | Apakah anda pernah memainkan aplikasi bergenre labirin sebelumnya? | P | E | R | N | A | H |

| No | Pertanyaan | Jawab | | | | | |
|----|--|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Objek objek dalam labirin sudah terlihat nyata seperti aslinya | | | | | | ✓ |
| 2 | Aplikasi memiliki tampilan dan desain yang menarik | | | ✓ | | | |
| 3 | Render model/objek tidak patah patah | | | | | ✓ | |
| 4 | Apakah aplikasi mudah dipahami? | | | ✓ | | ✓ | |
| 5 | Apakah tombol-tombol kontrol pada permainan tidak membingungkan? | | | ✓ | | | |
| 6 | Aplikasi sudah nyaman digunakan | | | | | ✓ | |

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
 Ketelitian menu dan tombol lebih diperjelas

Surabaya, 2017



Gambar C.5 Kuisisioner Responden Setiyo




Kuisisioner Tugas Akhir "The Labyrinth" Menggunakan Algoritma Recursive Division

Nama : Tiha I. Mooy
 NRP : 511320139
 Pekerjaan : Praktikan

Silahkan centang pada kolom yang sesuai berikut ini :

| No. | Pertanyaan | Jawab | | | | | |
|-----|---|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Apakah anda pernah menggunakan aplikasi virtual reality sebelumnya? | | | ✓ | | | |
| 2 | Apakah anda pernah memainkan aplikasi bergenre labirin sebelumnya? | | | | ✓ | | |

| No. | Pertanyaan | Jawab | | | | | |
|-----|--|-------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Objek objek dalam labirin sudah terlihat nyata seperti aslinya | | | | ✓ | | |
| 2 | Aplikasi memiliki tampilan dan desain yang menarik | | | | | ✓ | |
| 3 | Render model/objek tidak patah patah | | | | ✓ | | |
| 4 | Apakah aplikasi mudah dipahami? | | | | ✓ | | |
| 5 | Apakah tombol-tombol kontrol pada permainan tidak membingungkan? | | | | ✓ | | |
| 6 | Aplikasi sudah nyaman digunakan | | | ✓ | | | |

Keterangan :
1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
Game di tingkatkan

Surabaya, 4 Juli 2017

 Tiha I. Mooy

Gambar C.6 Kuesioner Responden Mooy

BIODATA PENULIS



Purbo Panambang, anak kedua dari 2 bersaudara. Lahir di Karanganyar, 2 Mei 1995. Alumni dari SD Kanisius Karangbangun Jumapolo, SMP N 1 Jumapolo dan SMA N 1 Karanganyar. Saat ini masih menjalani kuliah di Institut Teknologi Sepuluh Nopember Surabaya, Jurusan S1 Teknik Informatika dan sedang menempuh semester 8.

Selama menjadi mahasiswa, penulis pernah menjadi staf Web & 3D (Desain, Dokumentasi, dan Dekorasi) Schematics ITS 2014. Penulis juga pernah menjadi anggota Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS. Selain itu penulis juga telah menyelesaikan sertifikasi MTA (Microsoft Technology Associate) dari Microsoft.