



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - IF184802

# IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL MENGGUNAKAN TEKNOLOGI HTC VIVE

ANDREAN JANUAR PRIATMOJO  
NRP 0511154000029

Dosen Pembimbing I  
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.

Dosen Pembimbing II  
Anny Yuniarti, S.Kom., M.Comp.Sc.

DEPARTEMEN INFORMATIKA  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019

*[Halaman ini sengaja dikosongkan]*



**TUGAS AKHIR - IF184802**

**IMPLEMENTASI PERMAINAN MARBLE MAZE  
GAME BERBASIS REALITAS VIRTUAL  
MENGUNAKAN TEKNOLOGI HTC VIVE**

**ANDREAN JANUAR PRIATMOJO  
NRP 0511154000029**

**Dosen Pembimbing I  
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.**

**Dosen Pembimbing II  
Anny Yuniarti, S.Kom., M.Comp.Sc.**

**DEPARTEMEN INFORMATIKA  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019**

*[Halaman Ini Sengaja Dikosongkan]*



**FINAL PROJECT- IF184802**

**MARBLE MAZE BOARD GAME  
IMPLEMENTATION IN VIRTUAL REALITY GAME  
USING HTC VIVE TECHNOLOGY**

**ANDREAN JANUAR PRIATMOJO  
NRP 0511154000029**

**First Advisor  
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.**

**Second Advisor  
Anny Yuniarti, S.Kom., M.Comp.Sc.**

**DEPARTMENT OF INFORMATICS  
Faculty of Information and Communication Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019**

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

### IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL MENGUNAKAN TEKNOLOGI HTC VIVE

### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Rumpun Mata Kuliah Interaksi, Grafika, dan Seni  
Program Studi S-1 Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**ANDREAN JANUAR PRIATMOJO**

NRP. 0511154000029

Disetujui oleh Dosen Pembimbing Tugas Akhir

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

NIP: 197712172003121001

(pembimbing 1)

Anny Yuniarti, S.Kom., M.Comp.Sc.

NIP: 198106222005012002

(pembimbing 2)



**SURABAYA  
JULI, 2019**

*[Halaman ini sengaja dikosongkan]*



# IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL MENGGUNAKAN TEKNOLOGI HTC VIVE

**Nama Mahasiswa** : Andrian Januar Priatmojo  
**NRP** : 0511154000029  
**Departemen** : Informatika FTIK-ITS  
**Dosen Pembimbing I** : Dr.Eng. Darlis Herumurti, S.Kom.,  
M.Kom.  
**Dosen Pembimbing II** : Anny Yuniarti, S.Kom.,  
M.Comp.Sc.

## ABSTRAK

*Permainan merupakan salah satu media hiburan yang sudah ada sejak lama dan juga pembelajaran yang efektif bagi anak-anak. Salah satu permainan yang sederhana namun dapat melatih anak berpikir secara kritis yaitu jenis permainan maze. Marble Maze Game adalah salah satu permainan dari jenis maze. Permainan ini berupa board permainan dan dimainkan dengan cara mengatur bidang permainan sehingga kelereng dapat menuju finish dengan melewati rintangan yang ada.*

*Dalam perkembangannya, banyak permainan yang dibuat dalam bentuk permainan digital. Teknologi yang digunakan dalam pembuatan permainan digital telah berkembang. Teknologi permainan yang sedang menarik perhatian banyak orang adalah permainan berbasis realitas virtual. Salah satu contoh alat realitas virtual adalah HTC Vive. HTC Vive merupakan sebuah alat realitas virtual yang diciptakan oleh HTC dan Valve Corporation. Ide yang digunakan dalam tugas akhir ini adalah dengan membangun sebuah permainan berbasis realitas virtual dengan menggunakan teknologi HTC Vive dan mengimplementasikan permainan Marble Maze Game.*

*Hasil dari pengujian beta dan juga kuesioner tugas akhir ini telah terpenuhi dan dapat disimpulkan permainan ini telah mengimplementasikan perancangan dengan baik. Tugas akhir ini berupa sebuah permainan yang dibangun dengan Unity Versi 2019.1.1f1 dengan Bahasa Pemrograman C#, Steam VR Plugging dan Virtual Reality Toolkit. Proses pembuatan asset permainan sebagian besar diambil dari asset store unity dan unduhan dari sumber di internet.*

***Kata kunci: HTC Vive, Marble Maze Game, Realitas Virtual, Virtual Reality Toolkit, SteamVR.***

# MARBLE MAZE BOARD GAME IMPLEMENTATION IN VIRTUAL REALITY GAME USING HTC VIVE TECHNOLOGY

**Student Name** : Andrean Januar Priatmojo  
**NRP** : 0511154000029  
**Major** : Informatika FTIK-ITS  
**Advisor I** : Dr.Eng. Darlis Herumurti, S.Kom.,  
M.Kom.  
**Advisor II** : Anny Yuniarti, S.Kom., M.Comp.Sc.

## ABSTRACT

*Game is one of the entertainment media that has been around a long time and effective learning for children. One of the game that is simple but can train children to think critically is the maze genre. Marble Maze Game is one of the game in the maze type toys. This game is in the form of a board game and is played by rotate the playing board so that the marbles can go to the finish point by passing the obstacles.*

*In the process, a lot of game made in the form of digital games. The technology used in the manufacture of digital games has grown. Game technology that is attracting many people's attention is a virtual reality based game. One example of a virtual reality tool is HTC Vive. HTC Vive is a virtual reality tool created by HTC and Valve Corporation. The idea used in this final project is to build a virtual reality-based game using HTC Vive technology and implement the Marble Maze Game.*

*Results of the beta testing and also questionnaire of this final project have been fulfilled and it can be concluded the game has implemented the design properly. This final project is a game built with Unity Version 2019.1.1f1 with C# Programming Language, Steam VR Plugin and Virtual Reality Toolkit. The process of making game assets is mostly taken from asset store unity and downloads from another sources on the internet.*

**Key words:** *HTC Vive, Marble Maze Game, Virtual Reality, Virtual Reality Toolkit, SteamVR.*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur bagi Allah Tuhan Yang Maha Esa, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **Implementasi Permainan *Marble Maze Game* Berbasis Realitas Virtual Menggunakan Teknologi HTC Vive**. Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari. Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah Tuhan Yang Maha Esa atas anugerahnya yang tidak terkira kepada penulis.
2. Kedua orang tua penulis, Bapak Anang Mudjiono dan Ibu Titik Sri Haryanti, dan keluarga besar penulis Ibu Sri Hartini selaku nenek penulis dan Sekar Eda Kartika selaku adik penulis yang selalu memberi semangat dan doa sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Bapak Dr.Eng. Darlis Herumurti, S.Kom, M.Kom. Selaku pembimbing Tugas Akhir pertama yang dengan sabar telah memberikan arahan dalam mengerjakan Tugas Akhir ini.
4. Ibu Anny Yuniarti, S.Kom., M.Comp.Sc. Selaku pembimbing Tugas Akhir kedua yang dengan sabar membimbing penulis dalam mengerjakan Tugas Akhir ini.
5. Juwita Septiani yang selalu mendukung, menemani dan memberikan doa kepada penulis dalam mengerjakan Tugas Akhir ini.

6. Teman – teman penulis Renaldi Wahyudiono, Ronald Andrian, Fuad Dary Rosyadi, Rizka Anisa dan Hidayatul Munawaroh yang selalu membantu penulis ketika mengalami kesulitan selama pengerjaan Tugas Akhir ini.
7. Seluruh rekan – rekan Lab IGS, Adi Darmawan, GD Wahyu Nugraha, Dias Ardi, Subhan Maulana, Narendra H B, Yuga Mitra Hakiki, yang selalu menemani dan berbagi keluh kesah ketika penulis mengerjakan Tugas Akhir ini.
8. Seluruh admin Lab IGS yang telah bersedia membantu dan berbagi tempat selama penulis mengerjakan Tugas Akhir ini.
9. Teman – teman saudara seperjuangan, TC'15 yang selalu menjadi cerita tersendiri suatu hari nanti.
10. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juli 2019

Andrian Januar Priatmojo

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK .....	vii
ABSTRACT .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xv
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.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi.....	3
1.7 Sistematika Penulisan .....	5
BAB II TINJAUAN PUSTAKA .....	7
2.1 Unity 3D .....	7
2.2 Bahasa Pemrograman C#.....	7
2.3 Marbled Maze Game .....	8
2.4 Maze .....	9
2.5 Realitas Virtual.....	9
2.6 HTC Vive.....	10
2.6.1 Spesifikasi HTC Vive .....	12
2.6.2 Spesifikasi Komputer yang dibutuhkan .....	13
2.6.3 Penerapan HTC Vive dalam Pemrograman Unity ....	14
2.6.4 Kelebihan dan Kekurangan HTC Vive .....	14
2.7 Virtual Reality Toolkit (VRTK).....	15
BAB III ANALISIS DAN PERANCANGAN .....	17
3.1 Analisis Perangkat Lunak .....	17
3.1.1 Deskripsi Umum Perangkat Lunak .....	17
3.1.2 Spesifikasi Kebutuhan Sistem .....	18
3.1.3 Karakteristik Pengguna .....	20

3.2	Perancangan Perangkat Lunak .....	20
3.2.1	Perancangan Proses Permainan .....	20
3.2.2	Perancangan Aturan Permainan .....	21
3.2.3	Proses Perancangan Level Permainan.....	23
3.2.4	Perancangan Antarmuka Pengguna .....	25
3.2.5	Perancangan Objek dalam Game .....	30
<b>BAB IV IMPLEMENTASI.....</b>		<b>35</b>
4.1	Lingkungan Implementasi .....	35
4.2	Implementasi Pembuatan Aplikasi .....	35
4.2.1	Implementasi Antarmuka .....	35
4.2.2	Implementasi Permainan .....	51
<b>BAB V PENGUJIAN DAN EVALUASI .....</b>		<b>85</b>
5.1	Lingkungan Uji Coba .....	85
5.2	Pengujian Fungsionalitas .....	85
5.2.1	Uji Coba pada Menu Permainan .....	86
5.2.2	Uji Coba dalam Permainan.....	87
5.2.3	Uji Coba Akhir Permainan .....	91
5.3	Pengujian Pengguna .....	92
5.3.1	Skenario Uji Coba Pengguna.....	92
5.3.2	Tujuan Pertanyaan Kuesioner.....	93
5.3.3	Karakteristik Pengujian Pengguna .....	96
5.3.4	Hasil Pengujian Pengguna .....	96
5.3.5	Kritik dan saran pengguna.....	99
5.4	Evaluasi Pengujian .....	101
5.4.1	Evaluasi Pengujian Fungsionalitas.....	101
5.4.2	Evaluasi Pengujian Pengguna.....	102
<b>BAB VI KESIMPULAN DAN SARAN .....</b>		<b>105</b>
6.1.	Kesimpulan .....	105
6.2.	Saran.....	106
<b>DAFTAR PUSTAKA .....</b>		<b>107</b>
<b>LAMPIRAN .....</b>		<b>109</b>
<b>BIODATA PENULIS .....</b>		<b>135</b>



## DAFTAR GAMBAR

Gambar 2.1 Permainan Marbled Maze Game .....	8
Gambar 2.2 Vive Headset .....	10
Gambar 2.3 Vive Controller .....	11
Gambar 2.4 Vive Base Stations .....	12
Gambar 2.5 Virtual Reality Toolkit .....	15
Gambar 3.1 Diagram Alur Proses Permainan .....	22
Gambar 3.2 Rancangan Antarmuka Menu Utama .....	26
Gambar 3.3 Rancangan Antarmuka Cara Bermain.....	27
Gambar 3.4 Rancangan Antarmuka Permainan.....	27
Gambar 3.5 Rancangan Antarmuka Menu Pause .....	28
Gambar 3.6 Rancangan Antarmuka Win Condition .....	29
Gambar 3.7 Rancangan Antarmuka Win Condition .....	29
Gambar 3.8 Asset Model Pulau .....	30
Gambar 3.9 Contoh Model Pulau .....	31
Gambar 3.10 Model Pulau Game .....	31
Gambar 3.11 Model Lantai Maze .....	32
Gambar 3.12 Model Dinding Maze .....	32
Gambar 3.13 Model Bola .....	32
Gambar 3.14 Model Finish Maze .....	33
Gambar 3.15 Model Trampolin Maze.....	33
Gambar 3.16 Model Tuas Kontrol .....	34
Gambar 3.17 Model Controller HTC Vive .....	34
Gambar 4.1 Antarmuka Menu Utama .....	36
Gambar 4.2 Kontrol Memilih Menu .....	37
Gambar 4.3 Antarmuka Menu Cara Bermain.....	39
Gambar 4.4 Antarmuka Permainan.....	41
Gambar 4.5 Antarmuka Kontrol Permainan.....	41
Gambar 4.6 Antarmuka Kontrol Pada Sumbu Z.....	42
Gambar 4.7 Antarmuka Kontrol Pada Sumbu X .....	42
Gambar 4.8 Antarmuka Menu Pause .....	43
Gambar 4.9 Antarmuka Win Condition .....	46
Gambar 4.10 Antarmuka Game Over .....	49
Gambar 4.11 Pembuatan Dinding Maze .....	52

Gambar 4.12 Indikator SteamVR .....	70
Gambar 4.13 Setup Asset VRTK .....	71
Gambar 4.14 Objek Controller .....	72
Gambar 4.15 Implementasi Interaksi Grab Controller.....	72
Gambar 4.16 Implementasi Pada Tuas Kontrol.....	72
Gambar 4.17 Implementasi Pada Objek Canvas .....	74
Gambar 4.18 Implementasi Pointer Untuk User Interface .....	74
Gambar 4.19 Implementasi Level 1 Permainan .....	81
Gambar 4.20 Implementasi Level 2 Permainan .....	82
Gambar 4.21 Implementasi Level 3 Permainan .....	82
Gambar 4.22 Implementasi Level 4 Permainan .....	83
Gambar 4.23 Implementasi Level 5 Permainan .....	83
Gambar 4.24 Implementasi Level 6 Permainan .....	84
Gambar 4.25 Implementasi Level 7 Permainan .....	84
Gambar A. 1 Kuesioner Responden Subhan Maulana.....	109
Gambar A. 2 Kuesioner Responden Findryan K P.....	110
Gambar A. 3 Kuesioner Responden Abyan Dafa.....	111
Gambar A. 4 Kuesioner Responden Huda F M.....	112
Gambar A. 5 Kuesioner Responden Muhammad Adib A.....	113
Gambar A. 6 Kuesioner Responden Alfindio M A.....	114
Gambar A. 7 Kuesioner Responden Yuga Mitra Hakiki .....	115
Gambar A. 8 Kuesioner Responden N M H Habib Yahya.....	116
Gambar A. 9 Kuesioner Responden Baskara .....	117
Gambar A. 10 Kuesioner Responden Ronald Andrean .....	118
Gambar A. 11 Kuesioner Responden GD Wahyu Nugraha .....	119
Gambar A. 12 Kuesioner Responden Rogo Jagad Alit.....	120
Gambar A. 13 Kuesioner Responden Ivan Fadhila .....	121
Gambar A. 14 Kuesioner Responden Hidayatul Munawaroh ...	122
Gambar A. 15 Kuesioner Responden Hero Akbar A.....	123
Gambar A. 16 Kuesioner Responden Narendra H B .....	124
Gambar B. 1 Foto Pengujian Responden Abyan Dafa.....	125
Gambar B. 2 Foto Pengujian Responden M Adib Arinanda .....	126
Gambar B. 3 Foto Pengujian Responden Baskara.....	127
Gambar B. 4 Foto Pengujian Responden Yuga Mitra H.....	128
Gambar B. 5 Foto Pengujian Responden Alfindio M A .....	129

Gambar B. 6 Foto Pengujian Responden Hidayatul M.....	130
Gambar B. 7 Foto Pengujian Responden Hero Akbar A .....	131
Gambar B. 8 Foto Pengujian Responden GD Wahyu .....	132
Gambar B. 9 Foto Pengujian Responden Subhan Maulana.....	133

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 2.1 Spesifikasi HTC Vive .....	12
Tabel 2.2 Minimum Spesifikasi Komputer .....	13
Tabel 3.1 Kebutuhan Fungsional .....	19
Tabel 3.2 Kebutuhan Non-Fungsional .....	19
Tabel 4.1 Spesifikasi Lingkungan Implementasi.....	35
Tabel 5.1 Lingkungan Perangkat Keras .....	85
Tabel 5.2 Skenario Uji Coba Fungsionalitas .....	86
Tabel 5.3 Hasil Uji Coba pada Menu Permainan .....	86
Tabel 5.4 Hasil Uji Coba dalam Permainan .....	87
Tabel 5.5 Hasil Uji Coba Akhir Permainan.....	91
Tabel 5.6 Rentang Nilai .....	93
Tabel 5.7 Tujuan Pertanyaan.....	93
Tabel 5.8 Hasil Pengujian Pengguna .....	96
Tabel 5.9 Hasil Akhir Pengujian Pengguna .....	98
Tabel 5.10 Kritik dan Saran Pengguna .....	100
Tabel 5.11 Rekapitulasi Hasil Uji Fungsionalitas.....	102
Tabel 5.12 Rangkuman Hasil Kuesioner Pengujian Aplikasi ...	102

*[Halaman ini sengaja dikosongkan]*

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Kontrol User Interface Menu Utama.....	38
Kode Sumber 4.2 Kontrol User Interface Cara Bermain .....	40
Kode Sumber 4.3 Kontrol User Interface Menu Pause.....	44
Kode Sumber 4.4 Kontrol User Interface Win Condition .....	47
Kode Sumber 4.5 Kontrol User Interface Game Over .....	50
Kode Sumber 4.6 Proses Pembuatan Maze.....	59
Kode Sumber 4.7 Proses Pembuatan Lantai Maze .....	60
Kode Sumber 4.8 Proses Pembuatan Bola .....	61
Kode Sumber 4.9 Proses Pembuatan Titik Finish .....	63
Kode Sumber 4.10 Proses Inisiasi Permainan .....	66
Kode Sumber 4.11 Pembuatan Level Permainan .....	67
Kode Sumber 4.12 Perpindahan Level Permainan .....	68
Kode Sumber 4.13 Kontrol Game Over .....	69
Kode Sumber 4.14 Kontrol Suara Permainan .....	69
Kode Sumber 4.15 Implementasi Kontrol Rotasi Maze .....	73
Kode Sumber 4.16 Proses Penyimpanan Posisi Jebakan .....	75
Kode Sumber 4.17 Proses Pembuatan Jebakan Trampolin .....	77
Kode Sumber 4.18 Proses Penerapan Efek Trampolin .....	78
Kode Sumber 4.19 Proses Pembuatan Jebakan Lubang .....	80
Kode Sumber 4.20 Proses Pembuatan Timer .....	81

*[Halaman ini sengaja dikosongkan]*



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Permainan merupakan salah satu media hiburan yang sudah ada sejak lama dan juga pembelajaran yang efektif bagi anak-anak. Memainkan permainan merupakan salah satu cara untuk mengisi waktu luang dan juga mendorong berkembangnya keterampilan, fisik, sosial, dan intelektual pada anak. Salah satu permainan yang sederhana namun dapat melatih anak berpikir secara kritis yaitu jenis permainan *maze*. *Maze* merupakan permainan sederhana yang bertujuan menentukan jalur yang tepat untuk mencapai tujuan yang telah ditetapkan. *Maze* berisi bagian-bagian yang berupa kotak yang berisi jalur yang akan dilewati pada setiap baris atau kolom [1]. Salah satu *game* dari jenis *Maze* adalah *Marble Maze Game*. Permainan ini diciptakan pada tahun 1946 dikembangkan di Swedia oleh BRIO [2]. Pemain mengatur bidang permainan sehingga kelereng dapat menuju *finish* dengan melewati rintangan yang ada. Permainan ini sangat menarik karena dapat melatih logika dan kecepatan berpikir anak. Karena harus memikirkan jalur yang tepat untuk memenangkan permainan.

Dalam perkembangannya, banyak permainan yang dibuat dalam bentuk permainan digital. Teknologi yang digunakan dalam pembuatan permainan digital telah berkembang. Teknologi permainan yang sedang menarik perhatian banyak orang adalah permainan berbasis realitas virtual. Realitas virtual adalah teknologi yang dapat membuat pengguna berinteraksi dengan suatu lingkungan buatan yang disimulasikan oleh komputer, suatu lingkungan sebenarnya yang ditiru atau benar-benar suatu lingkungan yang hanya ada dalam imajinasi [3].

Salah satu contoh alat realitas virtual adalah HTC Vive. HTC Vive merupakan sebuah alat realitas virtual yang diciptakan oleh HTC dan Valve Corporation. HTC Vive dibangun di atas komputer dengan sistem operasi Windows 10, Windows 8 dan Windows 7. Perangkat

HTC Vive sendiri berupa *headset* dengan *OLED panel* dengan resolusi 1080 x 1200. HTC Vive dilengkapi dengan *wireless controller* yang biasa disebut dengan Vive Controller, terdiri dari dua unit setiap *controller* terdapat *track pad*, *grip buttons*, tombol menu, tombol *system*, dan dua pelatuk [4]. Dengan kehadiran *controller* memberikan keleluasaan untuk berinteraksi dengan dunia virtual lewat klik pada tombol atau gerakan. HTC Vive juga mendukung beberapa fungsi lain diluar untuk pembuatan permainan. Seperti contohnya untuk edukasi, hubungan sosial, dan hiburan.

Dalam hal ini penulis ingin membuat sebuah permainan yang berjudul Trapball. Trapball merupakan sebuah permainan *Marble Maze Game* berbasis realitas virtual yang menggunakan teknologi HTC Vive. *Maze* dirancang diatas papan dengan tambahan rintangan berupa jebakan, dimana pemain ditantang untuk mencari jalan keluar dari *maze* tersebut dengan mengatur kemiringan papan sehingga bola berhasil mencapai garis *finish* tanpa terkena jebakan.

## 1.2 Rumusan Masalah

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

1. Bagaimana implementasi permainan *Marble Maze Game* pada permainan Trapball didukung dengan konsep realitas virtual?
2. Bagaimana implementasi kontrol permainan *Marble Maze Game* pada permainan Trapball didukung dengan konsep realitas virtual?
3. Bagaimana implementasi dari permainan *Marble Maze Game* dihubungkan dengan teknologi HTC Vive?
4. Bagaimana implementasi dari permainan *Marble Maze Game* diselesaikan dengan Game Engine Unity?

### 1.3 Batasan Masalah

Berikut batasan masalah pada tugas akhir ini adalah:

1. Permainan yang dibuat merupakan permainan yang bekerja di komputer dengan sistem operasi Windows 10.
2. Permainan yang dibuat merupakan aplikasi realitas virtual yang membutuhkan HTC Vive untuk realitas virtual.
3. Lingkungan pengembangan yang digunakan menggunakan aplikasi Unity 3D lisensi gratis dan bahasa pemrograman C#.
4. Permainan yang dibuat hanya bisa dimainkan oleh *single player* atau satu orang.

### 1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah untuk mengimplementasikan permainan *Marble Maze Game* kedalam dunia realitas virtual dengan menggunakan teknologi HTC Vive .

### 1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini adalah:

1. Memberikan suasana *gameplay* yang menarik dengan mengimplementasikan permainan *Marble Maze Game* yang dilengkapi jebakan pada permainan realitas virtual.
2. Memberikan pengalaman baru pada pemain *game* realitas virtual.
3. Mengasah kemampuan berpikir dan ketangkasan tangan pemain untuk menyelesaikan/memenangkan permainan.
4. Sebagai sarana hiburan untuk para pemain.

### 1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

**A. Penyusunan proposal tugas akhir**

Proposal tugas akhir berisi deskripsi pendahuluan tugas akhir yang 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 subbab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

**B. Studi literatur**

Tahap studi literatur merupakan tahap pencarian informasi, pembelajaran dan studi literatur sejumlah referensi yang digunakan untuk mengimplementasikan tugas akhir. Tahap ini diawali dengan pengumpulan informasi dan pelatihan-pelatihan untuk pemahaman dasar teori yang digunakan pada topik tugas akhir, seperti Unity 3D, Bahasa Pemrograman C#, Marble Maze Game, Maze, Realitas Virtual, HTC Vive dan Virtual Reality Toolkit.

**C. Perancangan perangkat lunak**

Pada tahap ini diawali dengan melakukan analisis awal terhadap permasalahan utama yang muncul pada topik tugas akhir. Kemudian dilakukan perancangan perangkat lunak yang meliputi penentuan data yang digunakan dan proses-proses yang dilaksanakan, yang meliputi perancangan objek 3D dari objek area permainan dan perancangan *Gameplay* untuk membuat permainan semakin menarik.

**D. Implementasi perangkat lunak**

Aplikasi ini dibangun menggunakan *game engine* Unity 3D *free license* dengan bahasa pemrograman C#. Untuk pembuatan

*environment* dibuat menggunakan asset dari Asset Store Unity. Selain itu, menggunakan perangkat keras HTC Vive.

**E. Uji coba dan evaluasi**

Pada tahap ini dilakukan uji coba dengan indikator untuk mengetahui apakah aplikasi sudah memenuhi semua kebutuhan fungsional selain itu juga dilakukan pengujian non-fungsionalitas apakah aplikasi sudah bisa memberi kenyamanan dan kesan *immersive* kepada pengguna. Pengujian aplikasi ini dilakukan dengan metode kuesioner kepada pengguna yang sebelumnya telah diminta untuk menjalankan aplikasi secara langsung.

**F. 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 KESIMPULAN DAN SARAN**

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

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Unity 3D**

Unity adalah aplikasi pengembangan permainan yang terintegrasi kuat dengan satu set lengkap alat intuitif dan alur kerja yang cepat untuk membuat 3D interaktif dan konten 2D [5]. Unity merupakan *easy multiplatform publishing*. Unity memiliki *Web Asset* yang menyediakan *Asset* untuk diunduh 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* dan *EasyMoving*, *rotating*, dan *scaling objects* hanya perlu sebaris kode. Begitu juga dengan *Duplicating*, *removing*, dan *changing properties*. *Visual Properties Variables* yang didefinisikan dengan *scripts* ditampilkan pada editor. Bisa digeser, di *drag and drop*, bisa memilih warna dengan *color picker*. Berbasis .NET artinya perjalanan program dilakukan dengan Open Source .NET platform, Mono. Serta mendukung pengembangan aplikasi Microsoft, SONY, Qualcomm, BlackBerry, Samsung, Nintendo, Oculus Rift dan Intel.

#### **2.2 Bahasa Pemrograman C#**

Bahasa pemrograman C# (dibaca: *c sharp*) merupakan bahasa pemrograman yang dikembangkan oleh perusahaan ternama Microsoft dengan proses pengembangan yang dikepalai oleh Anders Hejlsberg. Karena dikembangkan oleh perusahaan yang sama dengan pengembang .NET, maka bahasa pemrograman C# menjadi bahasa pemrograman yang diintegrasikan sepenuhnya dengan library .NET [6].

*Net framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur yang terdapat pada bahasa-bahasa pemrograman lainnya seperti

Java, Delphi, Visual Basic dan lain-lain dengan beberapa penyederhanaan.

### 2.3 Marbled Maze Game

*Marble Maze Game* adalah permainan yang membutuhkan keterampilan fisik dan pemikiran cepat dari pemain untuk mengatur kemiringan papan dan mengarahkan bola untuk melalui jalur yang tepat dan menghindari jebakan agar dapat mencapai garis *finish*. Permainan ini diciptakan oleh BRIO di swedia pada tahun 1946, lalu diperkenalkan ke amerika serikat pada tahun 1950 [2]. Pada awalnya permainan ini dikemas dengan bentuk yang besar. Saat ini banyak terdapat variasi dari permainan ini.

Papan digerakan dengan cara mengatur 2 tuas yang berada di samping papan. Satu tuas digunakan untuk mengatur kemiringan pada sumbu horisontal dan satu tuas lainnya digunakan untuk mengatur kemiringan pada sumbu vertikal. Jebakan yang terdapat pada papan digolongkan dalam jebakan dalam skala kecil, sedang, dan besar. Jebakan juga digolongkan berdasarkan jebakan yang terlihat dan tidak terlihat. Oleh karena itu, tantangan dari permainan ini adalah untuk menemukan jalur yang tepat sembari melewati jebakan yang ada. Bentuk permainan *Marble Maze Game* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Permainan Marbled Maze Game

(Sumber : <https://www.amazon.it/Hey-Play-Labirinto-labirinto-acciaio/dp/B072QNFNN3>)



## 2.4 Maze

*Maze* merupakan *game* sederhana yang bertujuan menentukan jalur yang tepat untuk mencapai tujuan yang telah ditetapkan. Dalam permainan *maze*, pemain harus menemukan jalur pada bagian-bagian *maze* yang berupa kotak-kotak yang dilewati untuk tiap baris atau tiap kolom. Terdapat beberapa bentuk *maze* seperti *maze* untuk anak-anak seperti berbentuk lingkaran, mencari jejak rumah, menemukan jalan keluar untuk ke sekolah dan lain-lain. Permainan *maze* merupakan permainan edukatif dengan jalan sempit yang berliku dan berbelok dan kadang kala merupakan jalan buntu ataupun jalan yang mempunyai halangan, dapat juga dikatakan permainan mencari jalan keluar.

## 2.5 Realitas Virtual

Realitas Virtual merupakan sebuah teknologi yang berisikan simulasi komputer mengenai keadaan suatu lingkungan dan membuat penggunanya seakan-akan dapat berinteraksi dengan lingkungan tersebut [3]. Ketika seseorang melengkapi dirinya dengan berbagai macam peralatan realitas virtual, seakan-akan panca inderanya mampu merasakan benar-benar berada dalam lingkungan tersebut, meski jika peralatan dilepas, akan terasa bahwa sebetulnya dunia tersebut hanyalah dunia virtual yang sangat mirip kenyataan.

Sebuah sistem realitas virtual biasanya dilengkapi dengan berbagai peralatan yang mendukung agar penggunanya lebih merasakan sensasi lingkungan buatan tersebut, misalnya dengan menggunakan headset dan berbagai macam properti fisik lainnya yang bergantung pada detail lingkungan pada realitas virtual tersebut.

Realitas virtual bekerja dengan memanipulasi otak manusia sehingga seolah-olah merasakan berbagai hal yang virtual terasa seperti hal yang nyata. Bisa dibayangkan, realitas virtual merupakan proses penghapusan dunia nyata di sekeliling manusia, kemudian

membuat si pengguna merasa tergiring masuk ke dunia virtual yang sama sekali tak bersentuhan dengan dunia nyata. Ada beberapa perangkat tambahan yang dapat digunakan dalam teknologi realitas virtual. Contohnya seperti, Microsoft Hololens, Oculust Rift, HTC Vive atau Google Card Board [7].

## 2.6 HTC Vive

HTC Vive adalah *Virtual Reality Headset* yang dikembangkan oleh HTC dan Valve Corporation. HTC menggunakan teknologi pelacakan *room scale* atau skala ruangan yang memungkinkan pengguna bergerak dalam ruang 3D dan menggunakan pengendali genggam yang melacak gerakan untuk berinteraksi dengan lingkungan. HTC Vive terdiri atas beberapa komponen yaitu Vive Headset, Vive Controllers, Vive Base Stations, Vive Tracker, dan Vive Deluxe Audio Strap [4].

Vive Headset merupakan *head mounted device* (HMD) yang merupakan bagian dari HTC Vive. Perangkat ini digunakan pada kepala pemain dengan menggunakan *strap* khusus. *Strap* yang terdapat pada Vive Headset dapat diatur sesuai kenyamanan pengguna. Bentuk dari Vive Headset dapat dilihat pada Gambar 2.2.



Gambar 2.2 Vive Headset

(Sumber : <https://www.vive.com/sg/product/>)

Vive Controller merupakan bagian dari HTC Vive yang digunakan sebagai alat bantu permainan atau *control* pemain. Vive Controller di desain khusus untuk permainan *virtual reality*, dalam 1 paket HTC Vive dilengkapi dengan 2 buah *controller*. Terdapat beberapa metode input yaitu *track pad*, tombol *grip*, *dual-stage trigger*. Bentuk dari Vive Controller dapat dilihat pada Gambar 2.3.



Gambar 2.3 *Vive Controller*

(Sumber : <https://www.vive.com/sg/product/>)

Vive Base Stations juga dikenal sebagai sistem pelacakan *Lighthouse* yang merupakan dua kotak hitam yang menciptakan ruang virtual 360 derajat hingga radius 15x15 kaki. Terdapat belasan sensor yang digunakan untuk melacak pergerakan Vive Headset dan Vive Controller. Dalam setiap paketnya terdapat 2 buah Vive Base Stations. Alat ini dipasang secara diagonal pada wilayah permainan yang diinginkan. Bentuk Vive Base Stations dapat dilihat pada Gambar 2.4.



*Gambar 2.4 Vive Base Stations*

(Sumber : <https://www.vive.com/sg/product/>)

### 2.6.1 Spesifikasi HTC Vive

Teknologi ini menggunakan berbagai macam sensor, seperti *SteamVR Tracking*, *G-sensor*, *gyroscope*, *proximity*. Untuk melihat spesifikasi HTC Vive lebih lengkapnya dapat dilihat pada Tabel 2.1.

*Tabel 2.1 Spesifikasi HTC Vive*

<b>Headset</b>	
<b>Item</b>	<b>Keterangan</b>
<i>Screen :</i>	<i>Dual AMOLED 3.6'' diagonal</i>
<i>Resolution:</i>	<i>1080 x 1200 pixels per eye (2160 x 1200 pixels combined)</i>
<i>Refresh rate :</i>	<i>90 Hz</i>
<i>Field of view :</i>	<i>110 degrees</i>
<i>Safety features :</i>	<i>Chaperone play area boundaries and front-facing camera</i>
<i>Sensors :</i>	<i>SteamVR Tracking, G-sensor, gyroscope, proximity</i>
<i>Connections :</i>	<i>HDMI, USB 2.0, stereo 3.5 mm headphone jack, Power, Bluetooth</i>

<i>Input :</i>	<i>Integrated microphone</i>
<i>Eye Relief :</i>	<i>Interpupillary distance and lens distance adjustment</i>
<b>Controller</b>	
<b>Item</b>	<b>Keterangan</b>
<i>Sensors</i>	<i>SteamVR Tracking</i>
<i>Input :</i>	<i>Multifunction trackpad, Grip buttons, dual-stage trigger, System button, Menu button</i>
<i>Use per charge :</i>	<i>Use per charge:Approx. 6 hours</i>
<i>Connections :</i>	<i>Connections:Micro-USB charging port</i>
<b>Tracked area requirements</b>	
<b>Item</b>	<b>Keterangan</b>
<i>Standing / seated :</i>	<i>No min. space requirements</i>
<i>Room-scale :</i>	<i>6'6'' x 5' min. room size, 16'4'' max. between base stations</i>

## 2.6.2 Spesifikasi Komputer yang dibutuhkan

Untuk menggunakan HTC Vive dibutuhkan minimum spesifikasi komputer seperti pada Tabel 2.2.

*Tabel 2.2 Minimum Spesifikasi Komputer*

<b>Item</b>	<b>Keterangan</b>
<i>Graphics :</i>	<i>NVIDIA® GeForce® GTX 1060 or AMD Radeon™ RX 480, equivalent or better.</i>
<i>Processor :</i>	<i>Intel® Core™ i5-4590 or AMD FX™ 8350, equivalent or better</i>
<i>Memory :</i>	<i>4 GB RAM or more</i>
<i>Video out :</i>	<i>HDMI 1.4, DisplayPort 1.2 or newer</i>

<i>USB ports :</i>	<i>1x USB 2.0 or better port</i>
<i>Operating system :</i>	<i>Windows® 7 SP1, Windows® 8.1 or later, Windows® 10</i>

### 2.6.3 Penerapan HTC Vive dalam Pemrograman Unity

Untuk pengembangan *game virtual reality* menggunakan alat HTC Vive dalam pemrograman Unity wajib menggunakan *library* atau *plugin* SteamVR Plugin. SteamVR Plugin adalah salah satu *asset* yang terdapat di dalam *unity asset store*. *Plugin* ini menyediakan semua *script* dan *prefab* dasar yang dibutuhkan untuk membangun *game virtual reality* menggunakan HTC Vive.

Untuk mendukung pengembangan *game virtual reality* lebih mudah dan produktif kita dapat menggunakan Virtual Reality Toolkit atau yang biasa disebut dengan VRTK. Penggunaan VRTK tidak terbatas hanya untuk HTC Vive namun juga dapat digunakan untuk *device virtual reality* lainnya, seperti Oculus Rift, Google Daydream, bahkan Google Cardboard. VRTK berisi berbagai macam *script*, contoh, dan konsep yang berguna untuk membangun solusi dalam *game virtual reality* yang umum digunakan. Seperti *locomotion* dalam ruang virtual, interaksi objek menggunakan *controller*, interaksi dengan *element* UI Unity 3D melalui *pointer*, dan masih banyak lainnya.

### 2.6.4 Kelebihan dan Kekurangan HTC Vive

Dengan spesifikasi yang diberikan oleh HTC Vive akan membuat kita merasakan pengalaman yang tidak akan kita dapatkan dari VR *device* lainnya. Vive Controller memiliki 3 macam input yang membuat pengalaman bermain didalam dunia virtual semakin lengkap. Sedangkan Vive Base Station memiliki kemampuan untuk membuat ruang virtual hingga sejauh 15 kaki sehingga wilayah permainan menjadi semakin luas. *Strap* yang terdapat pada Vive Headset menambah kenyamanan saat penggunaan karena dapat diatur kerapatannya. Dan terdapat satu lagi kelebihan dari HTC Vive yaitu desain keseluruhan yang sangat futuristik.

Salah satu kekurangan dari HTC Vive yaitu, kita membutuhkan komputer dengan spesifikasi minimum yang cukup tinggi seperti pada Tabel 2.2. Selain itu harga untuk HTC Vive sendiri juga lebih mahal dibandingkan dengan VR *devices* lainnya. Banyaknya kabel yang digunakan untuk menggunakan HTC Vive berpotensi untuk membuat pengguna tersandung atau jatuh ketika memainkan permainan VR. Vive Headset memiliki ukuran yang cukup besar sehingga menambah berat dari Vive Headset itu sendiri ketika digunakan.

## 2.7 Virtual Reality Toolkit (VRTK)

VRTK adalah *package* Unity 3D yang bertujuan membuat pengembangan untuk platform VR lebih mudah dan lebih produktif di Unity3d 5+ [8]. VRTK memiliki berbagai macam *script* dan konsep yang berguna untuk membangun solusi *virtual reality* yang bisa digunakan. VRTK mencakup sejumlah *script* atau solusi umum seperti, *locomotion* dalam ruang virtual, interaksi seperti menyentuh, meraih dan menggunakan objek, berinteraksi dengan *element* UI Unity 3D melalui *pointer* atau sentuh, kontrol 2D dan 3D seperti tombol, tuas, pintu, laci, dan masih banyak lainnya.



Gambar 2.5 Virtual Reality Toolkit

(Sumber : <https://assetstore.unity.com/packages/tools/integration/vrtoolkit-virtual-reality-toolkit-vr-toolkit-64131>)

*[halaman ini sengaja dikosongkan]*



## **BAB III**

### **ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan tentang analisis dan perancangan aplikasi permainan realitas virtual Trapball. Permainan Trapball merupakan implementasi permainan *Marble maze game*. Pembahasan yang dibahas meliputi analisis fitur yang dibutuhkan dan perancangan perangkat lunak.

#### **3.1 Analisis Perangkat Lunak**

Pada subbab ini akan menjelaskan tentang hasil analisis kebutuhan perangkat lunak *game* Trapball menggunakan teknologi HTC Vive. Tiap-tiap subbab menjelaskan tentang deskripsi perangkat lunak, spesifikasi kebutuhan perangkat lunak, identifikasi pengguna dan perancangan perangkat lunak.

##### **3.1.1 Deskripsi Umum Perangkat Lunak**

Trapball merupakan implementasi dari *Marble Maze Game* dengan menggunakan realitas virtual. Fokus dalam pengerjaan Tugas Akhir ini terletak pada *gameplay*, bagaimana pemain dapat mengarahkan bola dari titik *start* hingga *finish* dengan cara melewati jalur yang tepat dalam *maze* dan menghindari jebakan yang ada. Pemain dapat mengatur laju bola dengan cara mengatur bidang permainan menggunakan dua kontrol. Dua kontrol meliputi satu untuk menggerakkan bidang permainan pada sumbu horizontal dan satu digunakan untuk menggerakkan bidang permainan pada sumbu vertikal.

Untuk bermain Trapball, pemain memerlukan *Personal Computer* yang mumpuni dan perangkat Realitas Virtual HTC Vive (Vive Headset, Vive Controller, Vive Base Stations). Selanjutnya melakukan penyetelan HTC Vive dan *Personal Computer* sehingga dapat digunakan.

Pemain akan memilih menu yang tersedia saat pertama kali memasuki permainan dengan cara menyentuh obyek virtual yang ada. Contohnya adalah memilih 'Play'. Setelah memilih menu

'Play', akan disediakan beberapa level permainan. Level permainan tergantung pada ukuran *maze* dan rintangan. Semakin tinggi level akan semakin besar ukuran *maze*, dan semakin banyak rintangan yang terdapat pada *board* permainan.

Pemain dapat melihat lingkungan virtual sesuai dengan keinginannya dengan menggunakan Vive Headset. Selain itu pemain juga dapat mengontrol objek yang telah disediakan dengan menggunakan Vive controller.

Permainan akan dimulai dengan bola berada di garis *start* didalam *maze* permainan. Pemain harus mengarahkan bola melewati *maze* yang terdapat pada *board* permainan agar dapat sampai di garis *finish*. Jika bola jatuh dari *board* permainan, maka bola direset kembali ke garis *start*. Sehingga pemain harus mengulang dari awal. Level permainan akan berakhir ketika bola sampai di garis *finish*. Tidak ada batasan waktu untuk menyelesaikan sebuah level permainan. Tapi catatan waktu bermain akan ditampilkan saat pemain menyelesaikan level permainan.

Agar dapat menyelesaikan permainan ini pemain memerlukan ketangkasan tangan dan koordinasi yang baik untuk mengontrol *board* permainan agar bola tidak terkena jebakan dan bergulir ke arah yang tepat. Diperlukan kordinasi antara kecepatan tangan dan kecepatan berpikir agar bola dapat keluar dari *maze* dengan jalur yang tepat namun juga harus memperhatikan jebakan yang ada. Selain itu dengan adanya kontrol menggunakan kedua tangan maka diperlukan koordinasi yang baik antara kedua tangan dan otak.

### **3.1.2 Spesifikasi Kebutuhan Sistem**

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

Tabel 3.1 Kebutuhan Fungsional

Kode	Deskripsi
F1	Pemain dapat melihat dan memilih menu utama
F2	Pemain dapat memulai permainan
F3	Pemain dapat keluar permainan
F4	Pemain dapat melihat informasi cara kontrol permainan
F5	Pemain dapat menyentuh tuas yang sudah disediakan
F6	Pemain dapat memilih menu <i>pause</i>
F7	Bola dapat bergerak ketika bidang permainan di kontrol
F8	<i>Maze</i> dapat berganti setiap level permainan dimulai
F9	Bola dapat bergerak keatas saat mengenai jebakan ( <i>trampoline</i> )
F10	Bola dapat jatuh kebawah saat mengenai jebakan ( <i>hole</i> )
F11	Pemain dapat mengubah rotasi <i>maze</i> dengan menggunakan tuas yang telah disediakan
F12	Pemain dapat melihat total waktu ketika level permainan berakhir
F13	Pemain dapat memulai level permainan selanjutnya
F14	Pemain dapat mengulang level permainan sebelumnya
F15	Pemain dapat mati dalam permainan
F16	Pemain dapat memenangkan permainan

Tabel 3.2 Kebutuhan Non-Fungsional

Kode	Deskripsi
NF1	Lingkungan permainan dibangun dengan menarik
NF2	Sistem dapat berjalan dengan lancar

### 3.1.3 Karakteristik Pengguna

Pengguna yang dapat memainkan *game* Trapball adalah siapa saja (umum), sehingga pengguna berhak menggunakan seluruh fungsionalitas yang terdapat pada sistem.

## 3.2 Perancangan Perangkat Lunak

Tahap perancangan pada bab ini dibagi menjadi beberapa bagian yaitu perancangan proses, perancangan skenario aturan main, perancangan antarmuka pengguna, proses perancangan level permainan, dan perancangan objek dalam *game*.

### 3.2.1 Perancangan Proses Permainan

Proses sistem dimulai dengan pemain membuka permainan dan memasuki menu utama. Di dalam menu utama pemain dapat memilih tiga tombol interaksi yaitu tombol *play*, *how to play*, dan *exit*. Tombol *play* digunakan untuk memulai level permainan, tombol *how to play* digunakan untuk menampilkan informasi cara kontrol atau cara bermain dari permainan. Sedangkan tombol *exit* digunakan untuk keluar dari permainan.

Setelah memilih tombol *play*, maka di arena permainan akan muncul dua tuas. Tuas ini digunakan untuk interaksi pemain agar dapat mengubah rotasi dari *board* permainan. Dengan merotasi *board* permainan, maka bola yang pada saat memulai level permainan akan bergerak. Untuk memenangkan sebuah level maka pemain harus mengontrol *board* permainan sehingga bola sampai pada titik *finish*.

Pada level tertentu akan muncul beberapa jebakan (*trap*) yang berfungsi sebagai rintangan untuk pemain. *Trap* akan muncul secara acak pada bidang permainan. Terdapat dua macam *trap* yang terdapat pada permainan. Yang pertama adalah trampolin, ketika bola terkena *trap* ini maka bola akan terlempar keatas. Yang kedua adalah lubang, ketika bola terkena *trap* ini maka bola akan jatuh kebawah. Permainan akan berakhir ketika bola jatuh dari bidang permainan.

Terdapat beberapa level permainan. Semakin tinggi level permainan maka kesulitan yang dihadapi pemain untuk sampai di titik *finish* semakin banyak. Kesulitan pada setiap level permainan berdasarkan pada besar *maze* dan banyak *trap*. Semakin tinggi level maka semakin besar *maze* dan banyak *trap* yang akan dihadapi.

Pada awal permainan bola akan berada pada titik *start*. Pemain mengatur rotasi *board* dengan melakukan interaksi pada tuas yang sudah tersedia. Terdapat dua tuas yang berada di sebelah kanan dan kiri pemain. Tuas kanan berfungsi untuk mengatur rotasi pada sumbu z dan tuas kiri berfungsi untuk mengatur rotasi pada sumbu x. Pemain melakukan kontrol pada tuas dengan cara menyentuh tuas menggunakan *controller*. Selengkapnya alur proses ini dapat dilihat pada Gambar 3.1.

### **3.2.2 Perancangan Aturan Permainan**

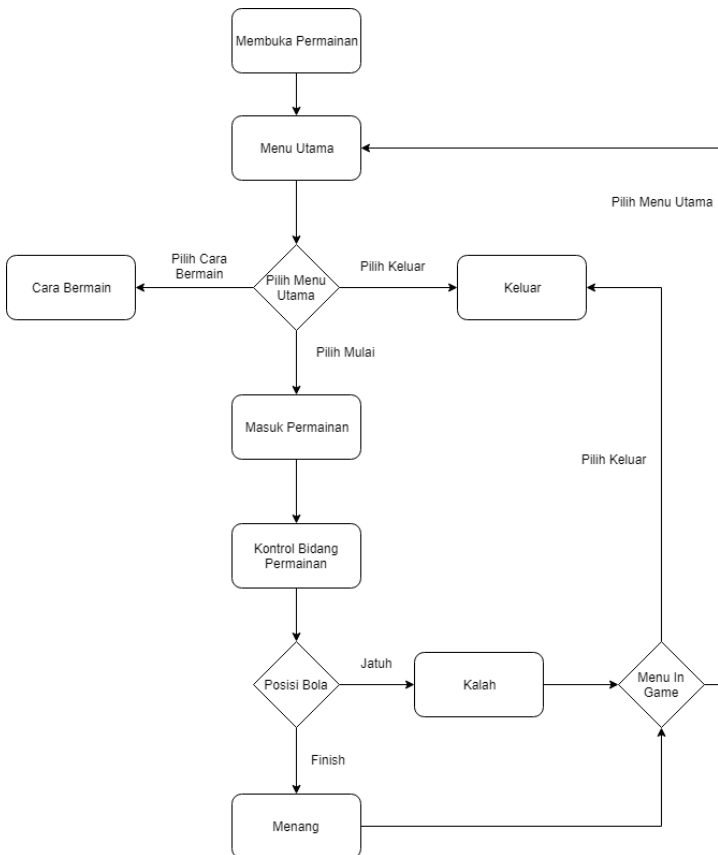
Pada sub bab ini menjelaskan tentang aturan main dalam permainan Trapball untuk menentukan kondisi menang atau kalah.

#### **3.2.2.1 Aturan Permainan**

Aturan permainan dari permainan Trapball antara lain :

1. Setelah pemain menekan tombol *Play*, maka akan muncul 2 tuas kontrol pada area permainan.
2. Pada awal permainan bola akan berada pada titik *start*.
3. Titik *finish* pada *board* permainan terdapat secara diagonal dari titik *start*.
4. Selain level 1 dan 5 terdapat jebakan yang muncul secara *random* pada *board* permainan.
5. Terdapat 2 jenis jebakan, yaitu jebakan trampolin dan jebakan lubang.
6. Ketika bola mengenai jebakan trampolin, maka bola akan terlempar keatas.

7. Ketika bola mengenai jebakan lubang, maka bola akan jatuh kebawah.
8. Pemain dikatakan kalah jika bola jatuh dari *board* permainan.
9. Pemain dikatakan menang jika bola berhasil mencapai titik *finish*.



Gambar 3.1 Diagram Alur Proses Permainan

### 3.2.2.2 Tata Cara Permainan

Tata cara permainan dalam permainan Trapball antara lain :

1. Pemain hanya dapat melakukan interaksi pada tuas kontrol yang telah tersedia.
2. Pemain harus mengatur rotasi *board* permainan agar bola dapat mencapai titik *finish*.
3. Pemain harus mengatur rotasi *board* permainan agar bola tidak terkena jebakan dan jatuh.
4. Tuas kontrol sebelah kanan digunakan untuk mengatur rotasi *board* permainan pada sumbu z.
5. Tuas kontrol sebelah kiri digunakan untuk mengatur rotasi *board* permainan pada sumbu x.

### 3.2.3 Proses Perancangan Level Permainan

Perancangan level permainan pada *game* ini dirancang untuk memberikan tantangan pada pemain dengan menambahkan jebakan dan ukuran *maze*. Terdapat 7 level permainan pada *game* ini, semakin tinggi level maka semakin besar ukuran *maze* dan semakin banyak jumlah jebakan yang muncul.

Jebakan yang ada di dalam *game* ini muncul secara acak pada bidang permainan, sehingga menambah kesulitan pemain dalam mengontrol rotasi *board* permainan. Terdapat dua jebakan, yaitu trampolin dan lubang. Trampolin memiliki efek untuk melontarkan bola keatas, dan lubang memiliki efek untuk membuat bola jatuh dari area permainan.

#### 3.2.3.1 Perancangan Permainan Level 1

Pada level 1 pemain akan dihadapkan pada *board maze* dengan ukuran 5 x 5, tanpa adanya jebakan. Hal ini bertujuan agar pemain terbiasa menggunakan kontrol pada permainan ini. Sehingga waktu yang diperlukan untuk menyelesaikan level ini

tidak akan lama. *Maze* dibuat secara otomatis, sehingga setiap kali level permainan dimulai bentuk *maze* akan terus berganti.

### **3.2.3.2 Perancangan Permainan Level 2**

Pada level 2 pemain akan dihadapkan pada *board maze* dengan ukuran 5 x 5, dengan 1 jebakan trampolin. Hal ini bertujuan agar pemain mulai merasakan tantangan dengan adanya jebakan yang muncul pada level permainan ini. Sehingga waktu yang diperlukan untuk menyelesaikan level ini akan bertambah lebih lama. *Maze* dibuat secara otomatis, sehingga setiap kali level permainan dimulai bentuk *maze* akan terus berganti.

### **3.2.3.3 Perancangan Permainan Level 3**

Pada level 3 pemain akan dihadapkan pada *board maze* dengan ukuran 5 x 5, dengan 2 jebakan trampolin. Hal ini bertujuan agar pemain mulai merasakan tantangan dengan bertambahnya jebakan yang muncul pada level permainan ini. Sehingga waktu yang diperlukan untuk menyelesaikan level ini akan bertambah lebih lama. *Maze* dibuat secara otomatis, sehingga setiap kali level permainan dimulai bentuk *maze* akan terus berganti.

### **3.2.3.4 Perancangan Permainan Level 4**

Pada level 4 pemain akan dihadapkan pada *board maze* dengan ukuran 5 x 5, dengan 2 jebakan trampoline dan 1 jebakan lubang. Hal ini bertujuan agar pemain mulai merasakan tantangan dengan bertambahnya tipe jebakan yang muncul pada level permainan ini. Sehingga waktu yang diperlukan untuk menyelesaikan level ini akan bertambah lebih lama. *Maze* dibuat secara otomatis, sehingga setiap kali level permainan dimulai bentuk *maze* akan terus berganti.

### **3.2.3.5 Perancangan Permainan Level 5**

Pada level 5 pemain akan dihadapkan pada *board maze* dengan ukuran 10 x 10, tanpa jebakan. Hal ini bertujuan agar pemain mulai terbiasa dengan besar *maze* yang ada. Karena dengan



bertambahnya ukuran *maze* tentu kesulitan dalam menyelesaikan level akan bertambah. *Maze* dibuat secara otomatis, sehingga setiap kali level permainan dimulai bentuk *maze* akan terus berganti.

### **3.2.3.6 Perancangan Permainan Level 6**

Pada level 6 pemain akan dihadapkan pada *board maze* dengan ukuran 10 x 10, dengan 3 jebakan trampoline dan 2 jebakan lubang. Hal ini bertujuan agar pemain mulai merasakan tantangan dengan bertambahnya jumlah dan tipe jebakan yang muncul pada level permainan ini. Sehingga waktu yang diperlukan untuk menyelesaikan level ini akan bertambah lebih lama. *Maze* dibuat secara otomatis, sehingga setiap kali level permainan dimulai bentuk *maze* akan terus berganti.

### **3.2.3.7 Perancangan Permainan Level 7**

Pada level 7 pemain akan dihadapkan pada *board maze* dengan ukuran 10 x 10, dengan 4 jebakan trampoline dan 3 jebakan lubang. Hal ini bertujuan agar pada level terakhir pemain dihadapkan pada kesulitan yang tinggi. Oleh karena itu jumlah jebakan yang muncul juga bertambah. Sehingga waktu yang diperlukan untuk menyelesaikan level ini akan cukup lama. *Maze* dibuat secara otomatis, sehingga setiap kali level permainan dimulai bentuk *maze* akan terus berganti.

## **3.2.4 Perancangan Antarmuka Pengguna**

Subbab ini membahas bagaimana rancangan antarmuka pengguna yang akan digunakan untuk tugas akhir. Rancangan antarmuka yang dibahas meliputi ketentuan masukan dan rancangan halaman tampilan. Didalam aplikasi ini terdapat beberapa tampilan, yaitu tampilan menu utama, tampilan cara bermain, tampilan permainan, tampilan menu *pause*, tampilan *win condition*, dan tampilan *game over*.

### 3.2.4.1 Rancangan Antarmuka Menu Utama

Tampilan menu utama permainan terdapat pada lingkungan yang berada disebuah pulau buatan. Pada tampilan ini terdapat tiga tombol yaitu tombol *play*, *how to play*, dan *exit*. Tombol *play* digunakan untuk memulai permainan, tombol *how to play* untuk masuk pada halaman cara bermain, dan tombol *exit* digunakan untuk keluar dari permainan. Tampilan rancangan antarmuka menu utama dapat dilihat pada Gambar 3.2.



Gambar 3.2 Rancangan Antarmuka Menu Utama

### 3.2.4.2 Rancangan Antarmuka Cara Bermain

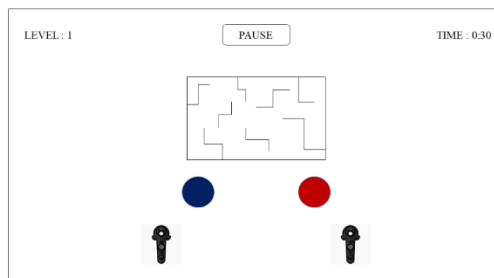
Tampilan ini berisi informasi tentang bagaimana cara bermain *game*, aturan main *game*, dan cara menggunakan *controller* pada *game*. Tampilan rancangan antarmuka cara bermain dapat dilihat pada Gambar 3.3.



Gambar 3.3 Rancangan Antarmuka Cara Bermain

### 3.2.4.3 Rancangan Antarmuka Permainan

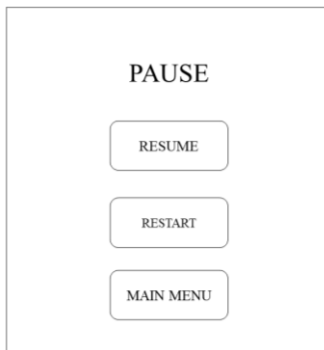
Tampilan permainan merupakan halaman yang muncul ketika pemain memilih tombol *play* pada halaman menu utama. Halaman ini merupakan halaman dimana pemain dapat mengontrol *board* permainan. Pada halaman ini juga terdapat tombol *pause*, dan juga keterangan tentang waktu serta level yang sedang dimainkan. Tombol *pause* digunakan pemain untuk masuk pada halaman menu *pause*. Pemain dapat melakukan interaksi terhadap dunia virtual pada halaman ini. Tampilan rancangan permainan dapat dilihat pada Gambar 3.4



Gambar 3.4 Rancangan Antarmuka Permainan

#### 3.2.4.4 Rancangan Antarmuka Menu *Pause*

Tampilan menu *pause* adalah halaman ketika pemain memilih menu *pause* saat memainkan permainan. Dalam menu ini terdapat tiga tombol, yaitu tombol *resume*, tombol *restart*, dan tombol *main menu*. Tombol *resume* digunakan untuk memulai kembali permainan, tombol *restart* digunakan untuk memulai ulang level permainan, dan tombol *main menu* digunakan untuk kembali ke halaman main menu. Tampilan rancangan antarmuka menu *pause* dapat dilihat pada gambar 3.5.



Gambar 3.5 Rancangan Antarmuka Menu *Pause*

#### 3.2.4.5 Rancangan Antarmuka *Win Condition*

Tampilan ini muncul ketika pemain berhasil mengarahkan bola hingga mencapai titik *finish*. Didalam tampilan ini terdapat beberapa informasi yang ditampilkan seperti keterangan level yang sudah diselesaikan beserta waktu yang dibutuhkan untuk menyelesaikan level. Selain itu terdapat beberapa tombol seperti, tombol *next*, tombol *restart*, dan tombol *main menu*. Tombol *next* digunakan untuk melanjutkan ke level selanjutnya, tombol *restart* digunakan untuk mengulang level yang telah diselesaikan, dan tombol *main menu* digunakan untuk kembali ke halaman menu utama. Tampilan rancangan antarmuka *win condition* dapat dilihat pada Gambar 3.6.



Gambar 3.6 Rancangan Antarmuka Win Condition

#### 3.2.4.6 Rancangan Antarmuka *Game Over*

Tampilan ini muncul ketika bola keluar dan jatuh dari *board* permainan. Bola dapat jatuh ketika pemain mengarahkan bola sehingga mengenai jebakan yang ada. Tampilan *game over* berisi beberapa tombol seperti, tombol *restart* dan tombol *main menu*. Tombol *restart* digunakan untuk mengulang kembali level permainan, sedangkan tombol *main menu* digunakan untuk kembali ke halaman menu utama. Tampilan rancangan antarmuka *game over* dapat dilihat pada Gambar 3.7.



Gambar 3.7 Rancangan Antarmuka Win Condition

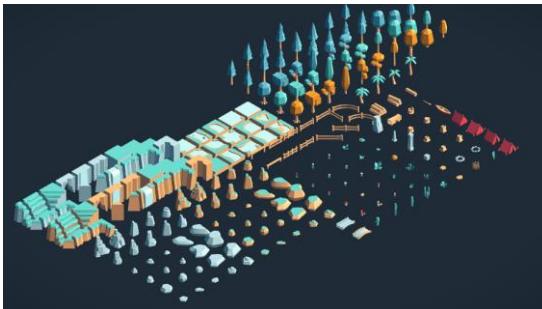
### 3.2.5 Perancangan Objek dalam Game

Perancangan objek dalam *game* ini memakai objek 3D dengan memodelkan objek kedalam bentuk tiga dimensi dan memberikan tekstur pada objek sehingga menyerupai objek aslinya. Pembuatan objek pada *game* ini sebagian besar menggunakan Unity 3D *object* dan beberapa diambil dari *asset store* unity. Selain itu, beberapa model didapat dari beberapa sumber melalui unduhan.

Bab ini dibagi ke dalam tiga subbab meliputi perancangan model pulau *game*, perancangan model *maze*, dan perancangan model-model pendukung lainnya.

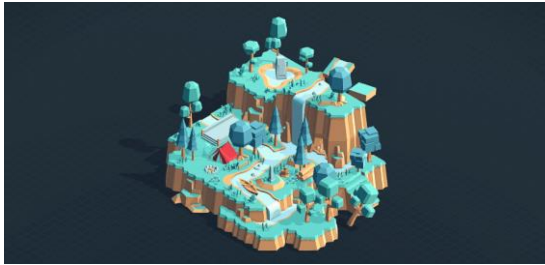
#### 3.2.5.1 Perancangan Model Pulau Game

Pada subbab ini dijelaskan hasil pembuatan model pulau *game* Trapball. Dalam pembuatannya, pembuat menggunakan *asset* yang didapat pada *website* penyedia *asset* gratis. Dari *asset* yang didapat pembuat membentuk objek menjadi pulau dengan mengatur *environment* (pohon, batu, rumput, pagar, rumah, dan objek pendukung lain). *Asset* yang digunakan berupa *low poly model* sehingga cocok untuk dimainkan semua kalangan dan membangun suasana yang terlihat *fun*. Model yang terdapat pada *asset* berjumlah 320 yang dapat digunakan. Tampilan pulau *game* Trapball dapat dilihat pada Gambar 3.8 sampai Gambar 3.10.



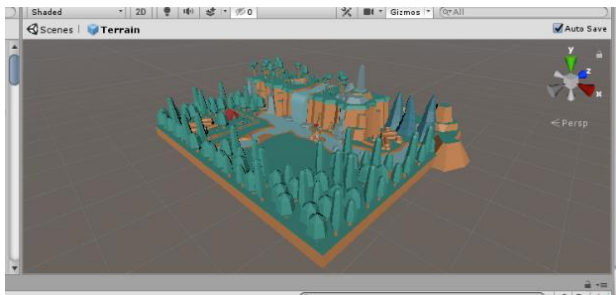
Gambar 3.8 Asset Model Pulau

(Sumber : <https://www.kenney.nl/assets/nature-kit>)



Gambar 3.9 Contoh Model Pulau

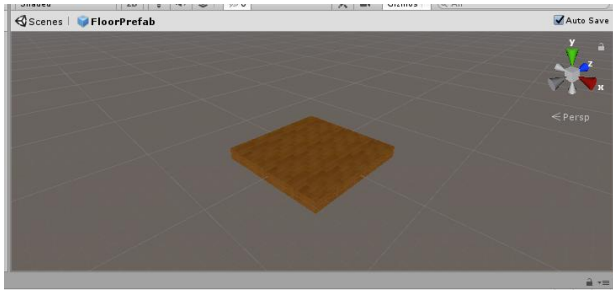
(Sumber : <https://www.kenney.nl/assets/nature-kit>)



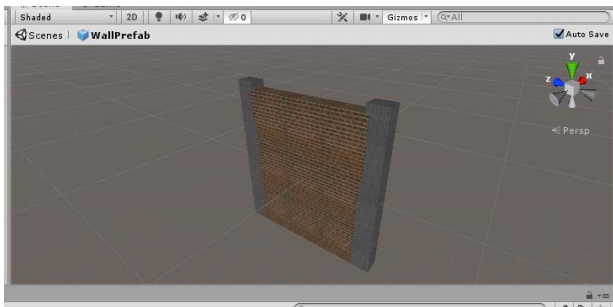
Gambar 3.10 Model Pulau Game

### 3.2.5.2 Perancangan Model Maze

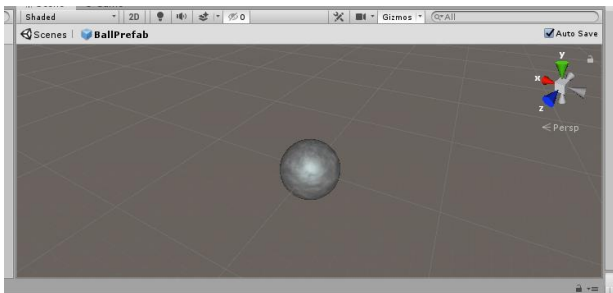
Pada subbab ini dijelaskan hasil pembuatan model *maze*. Sebagian besar model *maze* dibuat menggunakan unity 3D. Pembuatan *maze* dibagi menjadi lima bagian. Yaitu model dinding, model lantai, model bola, model *finish*, dan model trampolin. Pembuatan model *maze* terpisah karena pembuatan *maze* pada *game* ini secara otomatis. Agar *maze* dapat berganti-ganti secara acak setiap kali level dimulai. Setiap model yang diperlukan dalam perancangan model *maze* dapat dilihat pada Gambar 3.11 sampai Gambar 3.15.



*Gambar 3.11 Model Lantai Maze*

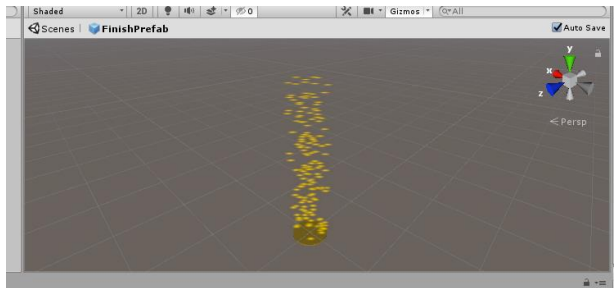


*Gambar 3.12 Model Dinding Maze*

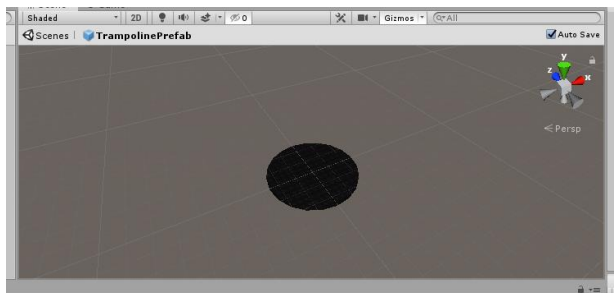


*Gambar 3.13 Model Bola*





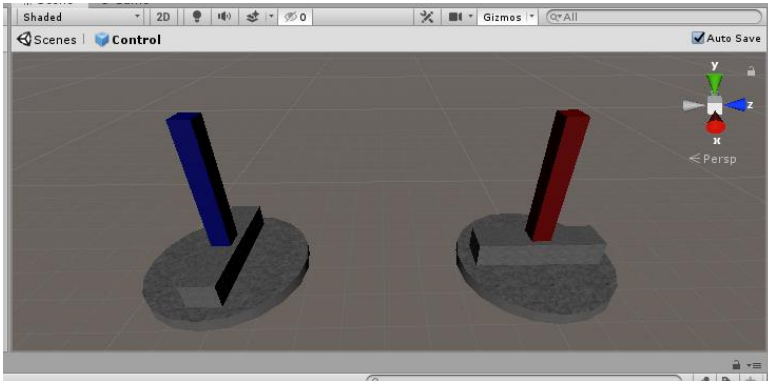
*Gambar 3.14 Model Finish Maze*



*Gambar 3.15 Model Trampolin Maze*

### **3.2.5.3 Perancangan Model-Model Pendukung Lain**

Pada subbab ini dijelaskan pemakaian dan sebab penggunaan model-model 3D lain sebagai pendukung aplikasi. Model-model pendukung yang digunakan pada *game* ini seperti model *controller* HTC Vive yang didapat dari *library* VRTK dan model tuas yang dibuat secara manual menggunakan unity 3D. Setiap model pendukung yang digunakan dalam *game* ini dapat dilihat pada Gambar 3.16 dan Gambar 3.17.



*Gambar 3.16 Model Tuas Kontrol*



*Gambar 3.17 Model Controller HTC Vive*

(Sumber : <https://assetstore.unity.com/packages/tools/integration/vrtk-virtual-reality-toolkit-vr-toolkit-64131>)

## **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 Spesifikasi Lingkungan Implementasi*

<b>Perangkat</b>	<b>Spesifikasi</b>
Perangkat Keras	<ul style="list-style-type: none"><li>• Prosesor Intel(R) Core (TM) i7-770 CPU @ 3.60GHz (8 CPUs), ~3.6 GHz</li><li>• RAM 8 Gigabyte</li><li>• Graphic card NVIDIA GeForce GTX 1060 3GB</li><li>• HTC Vive</li></ul>
Perangkat Lunak	<ul style="list-style-type: none"><li>• Sistem operasi Windows 10 Home Single Language 64-bit</li><li>• Unity 2019.1.1f1 (64-bit)</li></ul>

### **4.2 Implementasi Pembuatan Aplikasi**

Pada tahap ini dibagi menjadi dua subbab yaitu meliputi Implementasi Antarmuka dan Implementasi Permainan.

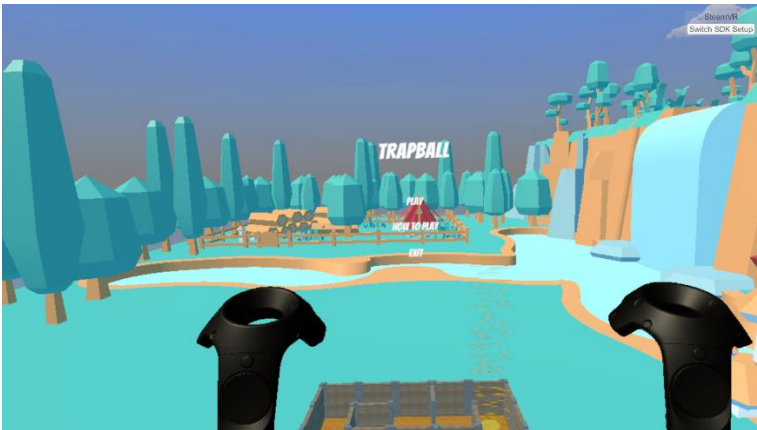
#### **4.2.1 Implementasi Antarmuka**

Implementasi antarmuka dalam *game* Trapball dibagi menjadi enam yaitu implementasi antarmuka menu utama, implementasi antarmuka cara bermain, implementasi antarmuka

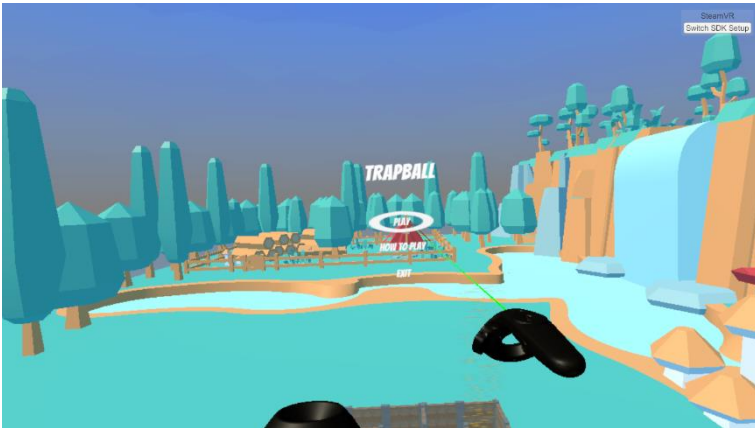
permainan, implementasi antarmuka menu *pause*, implementasi antarmuka *win condition*, dan implementasi antarmuka *game over*.

#### 4.2.1.1 Implementasi Antarmuka Menu Utama

Pada antarmuka ini penulis mengimplementasikan tampilan menu utama dengan isi judul permainan dan tiga tombol interaksi. Yaitu tombol *start*, tombol *how to play*, dan tombol *exit*. Tombol *play* digunakan untuk memulai permainan. Tombol *how to play* digunakan untuk memunculkan halaman cara bermain. Dan tombol *exit* digunakan untuk keluar dari permainan. Tampilan implementasi antarmuka menu utama dapat dilihat pada Gambar 4.1. dan Gambar 4.2. Sedangkan kode sumber untuk mengatur tombol pada antarmuka menu utama dapat dilihat pada Kode Sumber 4.1.



Gambar 4.1 Antarmuka Menu Utama



Gambar 4.2 Kontrol Memilih Menu

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class MainMenuManager : MonoBehaviour
6. {
7.     GameManager gameManager;
8.
9.     private void Start()
10.    {
11.        gameManager = GameObject.Find("Maze").Get
Component<GameManager>();
12.    }
13.
14.    public void StartGame()
15.    {
16.        gameManager.inGame.SetActive(true);
17.        gameManager.mainMenu.SetActive(false);
18.        gameManager.pauseButton.SetActive(true);
19.
20.        gameManager.Resume();
21.        gameManager.UnTransparent();
22.        gameManager.PlaySFX();
22.    }

```

```
23.  
24.     public void HowToPlay()  
25.     {  
26.         gameManager.mainMenu.SetActive(false);  
27.         gameManager.howTo.SetActive(true);  
28.         gameManager.PlaySFX();  
29.     }  
30.  
31.     public void ExitGame()  
32.     {  
33.         Application.Quit();  
34.         gameManager.PlaySFX();  
35.     }  
36. }
```

Kode Sumber 4.1 Kontrol User Interface Menu Utama

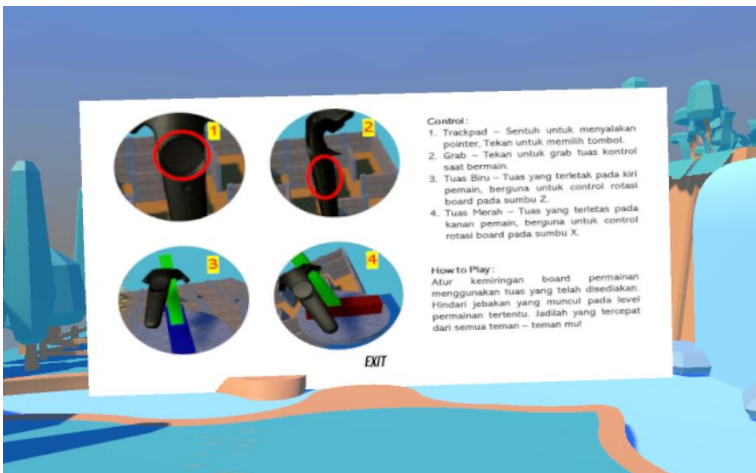
Fungsi *StartGame()* pada baris ke-14 berfungsi sebagai kontrol ketika pemain memilih tombol *start* pada menu utama. Didalam fungsi *StartGame()* terdapat beberapa intruksi antara lain untuk membuka halaman permainan (baris kode ke-16), menutup halaman menu utama (baris kode ke-17), memunculkan tombol *pause* (baris kode ke-18), memunculkan tuas kontrol (baris kode ke-20), memulai *timer* (baris kode ke-19), dan memutar efek suara (baris kode ke-21).

Fungsi *HowToPlay()* pada baris ke-24 berfungsi sebagai kontrol ketika pemain memilih tombol *how to play* pada menu utama. Didalam fungsi *HowToPlay()* terdapat beberapa intruksi antara lain untuk menutup halaman menu utama (baris kode ke-26), membuka halaman cara bermain (baris kode ke-27), dan memutar efek suara (baris kode ke-28).

Fungsi *ExitGame()* pada baris ke-31 berfungsi sebagai kontrol ketika pemain memilih tombol *exit* pada menu utama. Didalam fungsi *ExitGame()* terdapat dua intruksi antara lain untuk menutup aplikasi (baris kode ke-33) dan memutar efek suara (baris kode ke-34).

#### 4.2.1.2 Implementasi Antarmuka Cara Bermain

Pada antarmuka ini berisi tentang tata cara bermain, termasuk didalamnya informasi singkat tentang berbagai macam input dari Vive Controller. Hal ini bertujuan untuk mempermudah pemain saat mengoperasikan *controller*. Selain itu pada halaman ini juga terdapat informasi cara untuk menyelesaikan level permainan. Dan informasi tentang cara untuk mengontrol *board* permainan. Dalam antarmuka cara bermain penulis juga menambahkan tombol *exit* yang digunakan untuk kembali ke halaman menu utama. Tampilan implementasi antarmuka cara bermain dapat dilihat pada Gambar 4.3. Dan kode sumber untuk mengatur tombol pada antarmuka cara bermain dapat dilihat pada Kode Sumber 4.2.



Gambar 4.3 Antarmuka Menu Cara Bermain

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.

```

```

5. public class HowToMenuManager : MonoBehaviour
6. {
7.
8.     GameManager gameManager;
9.
10.    private void Start()
11.    {
12.        gameManager = GameObject.Find("Maze").GetComponent<GameManager>();
13.    }
14.
15.    public void Exit()
16.    {
17.        gameManager.howTo.SetActive(false);
18.        gameManager.mainMenu.SetActive(true);
19.        gameManager.PlaySFX();
20.    }
21. }

```

*Kode Sumber 4.2 Kontrol User Interface Cara Bermain*

Fungsi *Exit()* pada baris kode ke-15 berfungsi sebagai kontrol ketika pemain memilih tombol *exit*. Didalam fungsi *Exit()* terdapat beberapa intruksi, antara lain untuk menutup halaman cara bermain (baris kode ke-17), membuka halaman menu utama (baris kode ke-18), dan memutar efek suara (baris kode ke-19).

### 4.2.1.3 Implementasi Antarmuka Permainan

Antarmuka permainan adalah halaman tempat pemain melakukan interaksi pada objek virtual. Pemain dapat melihat seluruh ruang virtual yang telah dibuat secara lengkap. Selain itu pemain dapat melakukan interaksi dengan tombol dan juga melakukan interaksi pada tuas yang dibuat untuk mengontrol *board* permainan. Setiap tuas memerlukan interaksi dari pemain untuk dapat mengatur rotasi *board* permainan. Tuas bagian kiri digunakan untuk mengatur rotasi *board* pada sumbu z. Dan tuas

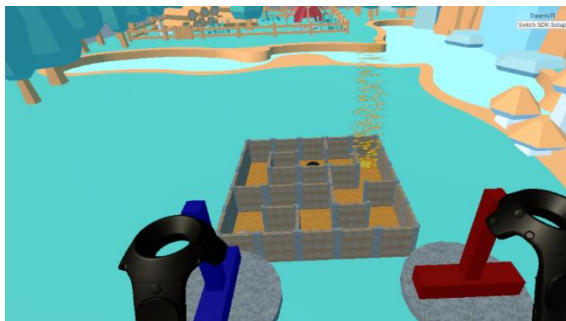


bagian kanan digunakan untuk mengatur rotasi board pada sumbu x.

Pada halaman ini juga terdapat beberapa informasi terkait level permainan seperti informasi level saat ini dan waktu permainan berlangsung. Selain itu terdapat tombol *pause* yang dapat digunakan oleh pemain untuk masuk ke halaman menu *pause*. Tampilan implementasi antarmuka permainan dapat dilihat pada Gambar 4.4 sampai Gambar 4.7.



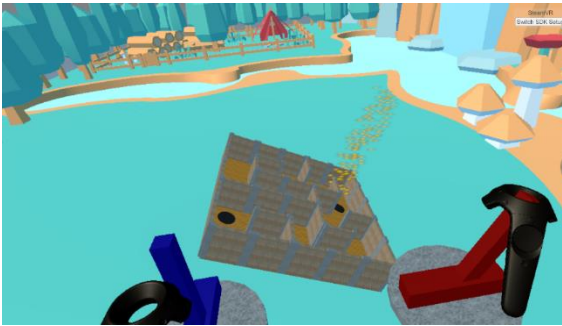
Gambar 4.4 Antarmuka Permainan



Gambar 4.5 Antarmuka Kontrol Permainan



Gambar 4.6 Antarmuka Kontrol Pada Sumbu Z



Gambar 4.7 Antarmuka Kontrol Pada Sumbu X

#### 4.2.1.4 Implementasi Antarmuka Menu *Pause*

Antarmuka menu *pause* muncul ketika pemain memilih tombol *pause* yang terdapat ketika bermain. Pada menu ini terdapat informasi tentang level saat ini dan tiga tombol. Tombol *restart level*, *resume level*, dan *exit level*. Tombol *restart level* digunakan untuk mengulang kembali level permainan saat ini. Tombol *resume level* digunakan untuk kembali ke level permainan. Dan tombol *exit level* digunakan untuk menuju halaman menu utama. Tampilan implementasi antarmuka menu *pause* dapat dilihat pada Gambar 4.8. Dan kode sumber untuk mengatur tombol pada antarmuka menu *pause* dapat dilihat pada Kode Sumber 4.3.



*Gambar 4.8 Antarmuka Menu Pause*

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.SceneManagement;
5. using VRTK;
6.
7. public class PauseMenuManager : MonoBehaviour
8. {
9.
10.     GameManager gameManager;
11.
12.     private void Start()
13.     {
14.         gameManager = GameObject.Find("Maze").GetComponent<GameManager>();
15.     }
16.
17.     public void Pause()
18.     {
19.         gameManager.ResetMazePosition();
20.         gameManager.Transparant();
21.         gameManager.pauseImage.SetActive(true);

```

```

22.         gameManager.Pause();
23.         gameManager.PlaySFX();
24.     }
25.
26.     public void Resume()
27.     {
28.         gameManager.UnTransparant();
29.         gameManager.Resume();
30.         gameManager.pauseImage.SetActive(false);
31.         gameManager.PlaySFX();
32.     }
33.
34.     public void Restart()
35.     {
36.         gameManager.ResetMazePosition();
37.         gameManager.UnTransparant();
38.         gameManager.pauseImage.SetActive(false);
39.         gameManager.GameOver();
40.         gameManager.Resume();
41.         gameManager.PlaySFX();
42.     }
43.
44.     public void Exit()
45.     {
46.         gameManager.ResetMazePosition();
47.         gameManager.level = 0;
48.         gameManager.mainMenu.SetActive(true);
49.         gameManager.pauseImage.SetActive(false);
50.         gameManager.inGame.SetActive(false);
51.         gameManager.pauseButton.SetActive(false);
52.         gameManager.Transparant();
53.         gameManager.PlaySFX();
54.         gameManager.GameOver();
55.     }
56. }

```

*Kode Sumber 4.3 Kontrol User Interface Menu Pause*

Fungsi *Pause()* pada baris ke-17 berfungsi sebagai kontrol ketika pemain memilih tombol *pause* pada halaman permainan.

Didalam fungsi *Pause()* terdapat beberapa intruksi antara lain untuk mengembalikan rotasi *maze* pada posisi normal (baris kode ke-19), menghapus tuas kontrol pada halaman permainan (baris kode ke-20), memunculkan halaman *pause* (baris kode ke-21), menghentikan waktu *timer* (baris kode ke-22), dan memutar efek suara (baris kode ke-23).

Fungsi *Resume()* pada baris ke-26 berfungsi sebagai kontrol ketika pemain memilih tombol *resume level* pada halaman *pause*. Didalam fungsi *Resume()* terdapat beberapa intruksi antara lain untuk menampilkan tuas kontrol (baris kode ke-28), memulai *timer* permainan (baris kode ke-29), menutup halaman menu *pause* (baris kode ke-30), dan memutar efek suara (baris kode ke-31).

Fungsi *Restart()* pada baris ke-34 berfungsi sebagai kontrol ketika pemain memilih tombol *restart level* pada halaman *pause*. Didalam fungsi *Restart()* terdapat beberapa intruksi antara lain untuk mengembalikan rotasi *maze* pada posisi normal (baris kode ke-36), menampilkan tuas kontrol (baris kode ke-37), menutup halaman menu *pause* (baris kode ke-38), memulai kembali *maze* pada level saat ini (baris kode ke-39), memulai *timer* permainan (baris kode ke-40) dan memutar efek suara (baris kode ke-41).

Fungsi *Exit()* pada baris ke-44 berfungsi sebagai kontrol ketika pemain memilih tombol *exit level* pada halaman *pause*. Didalam fungsi *Exit()* terdapat beberapa intruksi antara lain untuk mengembalikan rotasi *maze* pada posisi normal (baris kode ke-36), menghapus tuas kontrol (baris kode ke-52), membuka halaman menu utama (baris kode ke-48) menutup halaman menu *pause*, halaman permainan, dan tombol *pause* (baris kode ke-49 sampai baris kode ke-51), memulai kembali *maze* pada level 1 (baris kode ke-47 dan baris kode ke-54), dan memutar efek suara (baris kode ke-53).

#### 4.2.1.5 Implementasi Antarmuka *Win Condition*

Antarmuka *win condition* muncul ketika pemain berhasil mengarahkan bola hingga titik *finish*. Pada antarmuka ini terdapat

beberapa informasi yang dapat pemain dapatkan. Seperti informasi level permainan yang telah diselesaikan dan waktu yang dibutuhkan pemain untuk menyelesaikan level yang telah diselesaikan. Selain itu terdapat tiga tombol yaitu, tombol *next level*, *restart level*, dan *exit level*. Tombol *next level* digunakan untuk melanjutkan ke level selanjutnya. Tombol *restart level* digunakan untuk mengulang level yang telah diselesaikan dan tombol *exit level* digunakan untuk kembali ke halaman menu utama. Tampilan implementasi antarmuka *win condition* dapat dilihat pada Gambar 4.9. Dan kode sumber untuk mengatur tombol pada antarmuka *win condition* dapat dilihat pada Kode Sumber 4.4.



Gambar 4.9 Antarmuka Win Condition

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.SceneManagement;
5.
6. public class FinishMenuManager : MonoBehaviour
7. {
8.
9.     GameManager gameManager;
10.

```

```
11.     private void Start()
12.     {
13.         gameManager = GameObject.Find("Maze").GetComponent<GameManager>();
14.     }
15.
16.     public void Next()
17.     {
18.         gameManager.ResetMazePosition();
19.         gameManager.UnTransparant();
20.         gameManager.NextLevel();
21.         gameManager.finishImage.SetActive(false);
22.         gameManager.PlaySFX();
23.     }
24.
25.     public void Restart()
26.     {
27.         gameManager.ResetMazePosition();
28.         gameManager.UnTransparant();
29.         gameManager.GameOver();
30.         gameManager.finishImage.SetActive(false);
31.         gameManager.PlaySFX();
32.     }
33.
34.     public void Menu()
35.     {
36.         gameManager.ResetMazePosition();
37.         gameManager.level = 0;
38.         gameManager.mainMenu.SetActive(true);
39.         gameManager.finishImage.SetActive(false);
40.         gameManager.pauseImage.SetActive(false);
41.         gameManager.inGame.SetActive(false);
42.         gameManager.pauseButton.SetActive(false);
43.         gameManager.Transparant();
44.         gameManager.PlaySFX();
45.         gameManager.GameOver();
46.         gameManager.Pause();
47.     }
48. }
```

*Kode Sumber 4.4 Kontrol User Interface Win Condition*

Fungsi *Next()* pada baris ke-16 berfungsi sebagai kontrol ketika pemain memilih tombol *next level* pada halaman *win condition*. Didalam fungsi *Next()* terdapat beberapa intruksi antara lain untuk menghapus tuas kontrol (baris kode ke-19), mengembalikan rotasi *maze* pada posisi normal (baris kode ke-18), memulai level selanjutnya (baris kode ke-20), menutup halaman menu *win condition* (baris kode ke-21), dan memutar efek suara (baris kode ke-22).

Fungsi *Restart()* pada baris ke-25 berfungsi sebagai kontrol ketika pemain memilih tombol *restart level* pada halaman *win condition*. Didalam fungsi *Restart()* terdapat beberapa intruksi antara lain untuk mengembalikan rotasi *maze* pada posisi normal (baris kode ke-27), menampilkan tuas kontrol (baris kode ke-28), menutup halaman *win condition* (baris kode ke-30), memulai kembali *maze* pada level saat ini (baris kode ke-29), dan memutar efek suara (baris kode ke-41).

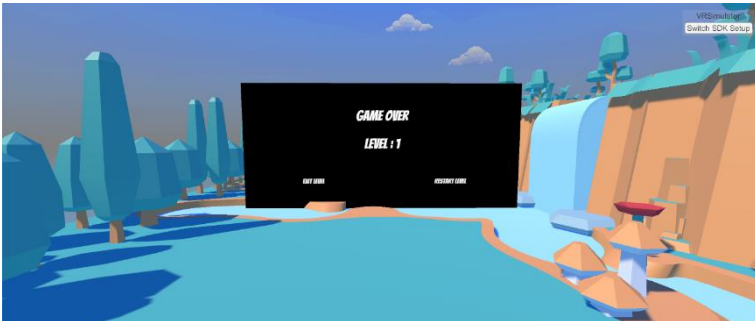
Fungsi *Menu()* pada baris ke-34 berfungsi sebagai kontrol ketika pemain memilih tombol *exit level* pada halaman *win condition*. Didalam fungsi *Menu()* terdapat beberapa intruksi antara lain untuk mengembalikan rotasi *maze* pada posisi normal (baris kode ke-36), menghapus tuas kontrol (baris kode ke-43), menutup halaman menu *pause*, halaman *win condition*, halaman permainan, dan tombol *pause* (baris kode ke-39 sampai baris kode ke-42), memulai kembali *maze* pada level 1 (baris kode ke-37 dan baris kode ke-45), membuka halaman menu utama (baris kode ke-38), menghentikan *timer* (baris kode ke-46), dan memutar efek suara (baris kode ke-44).

#### 4.2.1.6 Implementasi Antarmuka *Game Over*

Antarmuka *game over* muncul ketika pemain gagal dalam mencapai titik finish yaitu ketika bola jatuh dari *board* permainan. Pada antarmuka ini terdapat informasi tentang level yang sedang dimainkan dan terdapat dua tombol. Yang pertama tombol *restart level*, dan tombol *exit level*. Tombol *restart level* digunakan untuk



mengulang level yang telah dimainkan. Tombol *exit level* digunakan untuk kembali ke halaman menu utama. Tampilan implementasi antarmuka *game over* dapat dilihat pada Gambar 4.10. Dan kode sumber untuk mengatur tombol pada antarmuka *game over* dapat dilihat pada Kode Sumber 4.5.



Gambar 4.10 Antarmuka Game Over

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.SceneManagement;
5.
6. public class GameOverMenuManager : MonoBehaviour
7. {
8.
9.     GameManager gameManager;
10.
11.     private void Start()
12.     {
13.         gameManager = GameObject.Find("Maze").GetComponent<GameManager>();
14.     }
15.
16.     public void Restart()
17.     {
18.         gameManager.UnTransparant();
19.         gameManager.GameOver();

```

```

20.         gameManager.gameoverImage.SetActive(false);
21.         gameManager.inGame.SetActive(true);
22.         gameManager.pauseButton.SetActive(true);
23.         gameManager.PlaySFX();
24.     }
25.
26.     public void Menu()
27.     {
28.         gameManager.ResetMazePosition();
29.         gameManager.level = 0;
30.         gameManager.mainMenu.SetActive(true);
31.         gameManager.finishImage.SetActive(false);
32.         gameManager.pauseImage.SetActive(false);
33.         gameManager.inGame.SetActive(false);
34.         gameManager.pauseButton.SetActive(false);
35.         gameManager.gameoverImage.SetActive(false);
36.
37.         gameManager.Transparant();
38.         gameManager.PlaySFX();
39.         gameManager.GameOver();
40.         gameManager.Pause();
41.     }

```

*Kode Sumber 4.5 Kontrol User Interface Game Over*

Fungsi **Restart()** pada baris ke-16 berfungsi sebagai kontrol ketika pemain memilih tombol *restart level* pada halaman *game over*. Didalam fungsi **Restart()** terdapat beberapa intruksi antara lain untuk menampilkan tuas kontrol (baris kode ke-18), menutup halaman *game over* (baris kode ke-20), memulai kembali *maze* pada level saat ini (baris kode ke-19), membuka halaman permainan dan tombol *pause* (baris kode ke-21 dan baris kode ke-22), dan memutar efek suara (baris kode ke-23).

Fungsi **Menu()** pada baris ke-26 berfungsi sebagai kontrol ketika pemain memilih tombol *exit level* pada halaman *game over*. Didalam fungsi **Menu()** terdapat beberapa intruksi antara lain

untuk mengembalikan rotasi *maze* pada posisi normal (baris kode ke-28), menghapus tuas kontrol (baris kode ke-36), menutup halaman menu *pause*, halaman *win condition*, halaman *game over*, halaman permainan, dan tombol *pause* (baris kode ke-31 sampai baris kode ke-35), memulai kembali *maze* pada level 1 (baris kode ke-29 dan baris kode ke-38), membuka halaman menu utama (baris kode ke-30), menghentikan *timer* (baris kode ke-39), dan memutar efek suara (baris kode ke-37).

#### 4.2.2 Implementasi Permainan

Implementasi pada subbab ini meliputi implementasi *maze*, implementasi *game manager*, implementasi realitas virtual, implementasi kontrol *maze*, implementasi bola, implementasi jebakan trampolin, implementasi jebakan lubang, implementasi *timer*, dan implementasi level permainan.

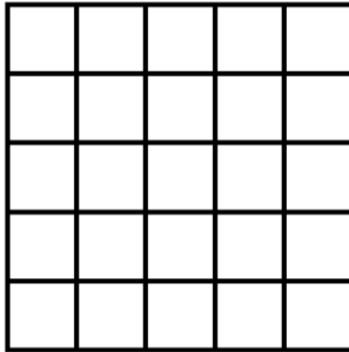
##### 4.2.2.1 Implementasi Maze

Implementasi pembuatan *maze* pada permainan ini dilakukan secara otomatis, sehingga *maze* yang dihasilkan setiap kali fungsi dijalankan akan selalu berubah atau *random*. Sehingga dibutuhkan beberapa objek secara terpisah seperti, objek lantai, objek dinding, objek bola dan objek *finish*.

Dalam pembuatan *maze* untuk permainan ini penulis menggunakan algoritma *Depth First Search* atau DFS. DFS adalah salah satu algoritma yang digunakan untuk pencarian jalur. Dengan cara melakukan perhitungan secara terurut dari urutan terakhir. Setelah menghabiskan semua kemungkinan dari titik terakhir baru mundur ke titik-titik sebelumnya sampai titik pertama [9]. Pembuatan *maze* pada permainan ini menggunakan DFS namun dengan sedikit modifikasi.

Pertama sistem membuat objek dinding terlebih dahulu dengan melakukan pembuatan objek dinding secara berurut sehingga menjadi seperti pada Gambar 4.11. Setiap kotak pada gambar tersebut merupakan sebuah sel dan setiap garis yang ada merepresentasikan dinding dari sel. Lalu setelah itu penulis

mengelompokkan setiap dinding pada sel untuk memisahkan dinding sel satu dari sel yang lain. Selain itu penulis perlu menyimpan data sel mana yang sudah pernah kita lewati. Jika *maze* memiliki ukuran 5 x 5 maka terdapat 25 sel pada *maze* tersebut.



Gambar 4.11 Pembuatan Dinding Maze

Kedua, sistem memilih secara acak atau *random* sel yang terdapat pada *maze* yang telah dibuat dan menandai sel tersebut dengan tanda sudah dilewati atau *visited*. Setelah itu proses berlanjut dengan mencari tetangga dari sel yang belum dikunjungi secara *random*. Jika proses menemukan sel tersebut, maka proses akan menghapus dinding antara sel yang sekarang dengan tetangga sel yang secara *random* telah dipilih. Lalu proses berlanjut ke sel tetangga yang dipilih secara *random* sebelumnya. Dan menandai sel tetangga menjadi *visited*. Jika proses tidak menemukan sel tetangga maka proses akan kembali ke sel sebelumnya sampai terdapat sel tetangga yang belum memiliki status *visited*. Sistem akan mengulangi proses ini sampai semua sel telah dikunjungi. Proses pembuatan *maze* secara *random* dapat dilihat pada Kode Sumber 4.6.

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

```
3. using UnityEngine;
4. using Random = UnityEngine.Random;
5.
6. public class MazeManager : MonoBehaviour
7. {
8.
9.     [System.Serializable]
10.    public class Cell
11.    {
12.        public bool visited;
13.        public GameObject up;
14.        public GameObject left;
15.        public GameObject right;
16.        public GameObject down;
17.    }
18.
19.    [HideInInspector] public float wallLength = 0f;
20.    [HideInInspector] public int rows = 0, columns =
    0, randomPoint;
21.    [HideInInspector] public GameObject Ball, Finish
    , Floor, Wall, Trampoline, Control;
22.
23.    public GameObject wall, floor, finish, trampolin
    e, control, ball;
24.    private List<GameObject> floors = new List<GameO
    bject>();
25.
26.    private Cell[] cells;
27.    private List<int> cellStep;
28.    private int totalCell, currentCell = 0, visitedC
    ell = 0, currentNeighbor = 0, backingUp = 0, wallToB
    reak = 0;
29.    private bool startedBuild = false;
30.
31.    private Vector3 initialPos;
32.    private GameObject instance;
33.    private List<Vector3> positionStore = new List<V
    ector3>();
34.    private int trampolineCount, floorCount;
35.    private float spawnTime = 1.0f, floorTimer = 2.0
    f;
36.
```

```
37.     void ToAssign(int row, int column, float length)
38.     {
39.         wallLength = length;
40.         rows = row;
41.         columns = column;
42.     }
43.
44.     void CreateWalls(int row, int column, float length)
45.     {
46.         ToAssign(row, column, length);
47.         Wall = new GameObject("Wall");
48.
49.         Vector3 currentPos = initialPos;
50.         initialPos = new Vector3((-
row / 2) + length / 2, 0.0f, (-
column / 2) + length / 2);
51.
52.
53.         for (int x = 0; x < column; x++)
54.         {
55.             for (int y = 0; y <= row; y++)
56.             {
57.                 currentPos = new Vector3(initialPos.
x + (y * length) - length / 2, 0.0f, initialPos.z +
(x * length) - length / 2);
58.                 GameObject instance = Instantiate(wa
ll, currentPos, Quaternion.identity) as GameObject;
59.                 instance.transform.SetParent(Wall.tr
ansform);
60.                 Wall.transform.parent = transform;
61.             }
62.         }
63.
64.         for (int x = 0; x <= column; x++)
65.         {
66.             for (int y = 0; y < row; y++)
67.             {
68.                 currentPos = new Vector3(initialPos.
x + (y * length), 0.0f, initialPos.z + (x * length)
- length);
```

```

69.         GameObject instance = Instantiate(wa
    ll, currentPos, Quaternion.Euler(0.0f, 90.0f, 0.0f))
    as GameObject;
70.         instance.transform.SetParent(Wall.tr
    ansform);
71.         Wall.transform.parent = transform;
72.     }
73. }
74.
75.     CreateCells();
76. }
77.
78. void CreateCells()
79. {
80.     cellStep = new List<int>();
81.     cellStep.Clear();
82.     totalCell = rows * columns;
83.     int horizontalProcess = 0, childProcess = 0,
    countProcess = 0;
84.
85.     GameObject[] allWalls;
86.     int children = Wall.transform.childCount;
87.     allWalls = new GameObject[children];
88.     cells = new Cell[rows * columns];
89.
90.     for (int x = 0; x < children; x++)
91.     {
92.         allWalls[x] = Wall.transform.GetChild(x)
    .gameObject;
93.     }
94.
95.     for (int process = 0; process < cells.Length
    ; process++)
96.     {
97.         if (countProcess == rows)
98.         {
99.             horizontalProcess++;
100.            countProcess = 0;
101.        }
102.
103.        cells[process] = new Cell();
104.        cells[process].left = allWalls[horizo
    ntalProcess];

```

```

105.         cells[process].down = allWalls[childP
    rocess + (rows + 1) * columns];
106.
107.         horizontalProcess++;
108.         countProcess++;
109.         childProcess++;
110.
111.         cells[process].right = allWalls[hORIZ
    ONTALProcess];
112.         cells[process].up = allWalls[(childPr
    ocess + (rows + 1) * columns) + rows - 1];
113.     }
114.
115.     CreateMaze();
116. }
117.
118. void CreateMaze()
119. {
120.     while (visitedCell < totalCell)
121.     {
122.         if (startedBuild)
123.         {
124.             SetNeighbor();
125.             if (cells[currentNeighbor].visite
    d == false && cells[currentCell].visited == true)
126.             {
127.                 BreakWall();
128.                 cells[currentNeighbor].visite
    d = true;
129.                 visitedCell++;
130.                 cellStep.Add(currentCell);
131.                 currentCell = currentNeighbor
    ;
132.
133.                 if (cellStep.Count > 0)
134.                 {
135.                     backingUp = cellStep.Coun
    t - 1;
136.                 }
137.             }
138.         }
139.     }
140. }

```



```
141.         currentCell = Random.Range(0, tot
    alCell);
142.         cells[currentCell].visited = true
;
143.         visitedCell++;
144.         startedBuild = true;
145.     }
146. }
147. }
148.
149. void BreakWall()
150. {
151.     switch (wallToBreak)
152.     {
153.         case 1: Destroy(cells[currentCell].up
); break;
154.         case 2: Destroy(cells[currentCell].le
ft); break;
155.         case 3: Destroy(cells[currentCell].ri
ght); break;
156.         case 4: Destroy(cells[currentCell].do
wn); break;
157.     }
158. }
159.
160. void SetNeighbor()
161. {
162.     int length = 0, corner = 0;
163.     int[] neighbors = new int[4];
164.     int[] connectingWall = new int[4];
165.     corner = ((currentCell + 1) / rows);
166.     corner -= 1;
167.     corner *= rows;
168.     corner += rows;
169.
170.     if (currentCell + 1 < totalCell && (curre
ntCell + 1) != corner)
171.     {
172.         if (cells[currentCell + 1].visited ==
false)
173.         {
174.             neighbors[length] = currentCell +
1;
1;
```

```
175.             connectingWall[length] = 3;
176.             length++;
177.         }
178.     }
179.
180.     if (currentCell - 1 >= 0 && currentCell !=
= corner)
181.     {
182.         if (cells[currentCell - 1].visited ==
false)
183.         {
184.             neighbors[length] = currentCell -
1;
185.             connectingWall[length] = 2;
186.             length++;
187.         }
188.     }
189.
190.     if (currentCell + rows < totalCell)
191.     {
192.         if (cells[currentCell + rows].visited
== false)
193.         {
194.             neighbors[length] = currentCell +
rows;
195.             connectingWall[length] = 1;
196.             length++;
197.         }
198.     }
199.
200.     if (currentCell - rows >= 0)
201.     {
202.         if (cells[currentCell - rows].visited
== false)
203.         {
204.             neighbors[length] = currentCell -
rows;
205.             connectingWall[length] = 4;
206.             length++;
207.         }
208.     }
209.
210.     if (length != 0)
```

```

211.         {
212.             int chosenNeighbor = Random.Range(0,
                length);
213.             currentNeighbor = neighbors[chosenNe
                ighbor];
214.             wallToBreak = connectingWall[chosenN
                eighbor];
215.         }
216.         else
217.         {
218.             if (backingUp > 0)
219.             {
220.                 currentCell = cellStep[backingUp]
                ;
221.                 backingUp--;
222.             }
223.         }
224.     }

```

Kode Sumber 4.6 Proses Pembuatan Maze

Pada Kode Sumber 4.6 terdapat beberapa fungsi, seperti fungsi **CreateWalls()** pada baris kode ke-44 berfungsi untuk memanggil atau membangun objek dinding pada *maze*. Objek dinding dibuat secara berurutan sebanyak besar *maze*. Fungsi **CreateCell()** pada baris kode ke-78 digunakan untuk mengelompokkan dinding yang sudah terbuat menjadi sel-sel yang akan digunakan pada proses selanjutnya. Selanjutnya fungsi **SetNeighbor()** pada baris kode ke-160 digunakan untuk memilih secara *random* sel tetangga yang akan dikunjungi. Fungsi **BreakWall()** pada baris kode ke-149 digunakan untuk menghapus dinding antara sel yang telah dikunjungi dan sel tetangga yang akan dikunjungi. Yang terakhir fungsi **CreateMaze()** pada baris kode ke-118 berfungsi untuk memulai proses pembuatan *maze*.

Untuk proses pembuatan lantai, sama seperti pembuatan dinding. Objek lantai dibuat secara berurut sebesar ukuran *maze*. Objek lantai yang dibuat merupakan objek model yang telah

disiapkan sebelumnya. Proses pembuatan lantai *maze* menggunakan fungsi *CreateFloors()* dapat dilihat pada Kode Sumber 4.7.

```

1. void CreateFloors(int row, int column, float length)
2.     {
3.
4.         ToAssign(row, column, length);
5.         Floor = new GameObject("Floor");
6.
7.         Vector3 currentPos = initialPos;
8.         initialPos = new Vector3((-
row / 2) + length / 2, 0.0f, (-
column / 2) + length / 2);
9.
10.        for (int x = 0; x < column; x++)
11.            {
12.                for (int y = 0; y < row; y++)
13.                    {
14.                        currentPos = new Vector3(initialPos.
x + (y * length), -
0.5f, initialPos.z + (x * length) - length / 2);
15.                        GameObject instance = Instantiate(fl
oor, currentPos, Quaternion.Euler(0, 0, 0)) as GameO
bject;
16.                        instance.transform.SetParent(Floor.t
ransform);
17.                        Floor.transform.parent = transform;
18.                        floors.Add(instance);
19.                    }
20.            }
21.    }

```

Kode Sumber 4.7 Proses Pembuatan Lantai Maze

Untuk pembuatan objek bola proses yang dilakukan dengan cara menyimpan titik pojok setiap *maze*. Bola akan dibuat

pada titik yang berlawanan secara diagonal dengan posisi titik *finish* dibuat. Hal ini dilakukan untuk meningkatkan tingkat kesulitan saat menyelesaikan *maze*. Selain itu posisi bola pada *maze* juga dibuat secara *random*. Proses pembuatan objek bola menggunakan fungsi **CreateBall()** dapat dilihat pada Kode Sumber 4.8.

```

1. void CreateBall(int row, int column, float length)
2.     {
3.         ToAssign(row, column, length);
4.         Ball = new GameObject("Ball");
5.
6.         Vector3[] cornerPos = new Vector3[]
7.         {
8.             new Vector3((-row / 2) + length / 2, -
9.             0.45f, (-column / 2)),
10.            new Vector3((-row / 2) + length / 2, -
11.            0.45f, length * (column - 1) - (column / 2)),
12.            new Vector3((-
13.            row / 2) + length / 2 + length * (row - 1), -
14.            0.45f, (-column / 2)),
15.            new Vector3((-
16.            row / 2) + length / 2 + length * (row - 1), -
17.            0.45f, length * (column - 1) - (column / 2))
18.        };
19.
20.        GameObject instance = Instantiate(ball, new
21.        Vector3(0, 0, 0), Quaternion.identity) as GameObject
22.        ;
23.        instance.transform.SetParent(Ball.transform)
24.        ;
25.        instance.transform.position = (cornerPos[4 -
26.        randomPoint] * transform.localScale.x) + setPositio
27.        n;
28.        instance.transform.localScale = transform.lo
29.        calScale / 2;
30.    }

```

Kode Sumber 4.8 Proses Pembuatan Bola

Yang terakhir untuk pembuatan titik *finish* dilakukan secara diagonal dari titik bola dibuat. Titik pojok setiap *maze* telah disimpan terlebih dahulu dan dipilih secara acak setiap kali *maze* dibuat. Hal ini untuk menambah kesulitan untuk memecahkan *maze* yang telah dibuat. Proses pembuatan titik *finish* menggunakan fungsi **CreateFinish()** dapat dilihat pada Kode Sumber 4.9.

```

1. void CreateFinish(int row, int column, float length)
2.     {
3.
4.         ToAssign(row, column, length);
5.         Finish = new GameObject("Finish");
6.
7.         Vector3[] cornerPos = new Vector3[]
8.         {
9.             new Vector3((-row / 2) + length / 2, -
10.                0.45f, (-column / 2)),
11.             new Vector3((-row / 2) + length / 2, -
12.                0.45f, length * (column - 1) - (column / 2)),
13.             new Vector3((-
14.                row / 2) + length / 2 + length * (row - 1), -
15.                0.45f, (-column / 2)),
16.             new Vector3((-
17.                row / 2) + length / 2 + length * (row - 1), -
18.                0.45f, length * (column - 1) - (column / 2))
19.         };
20.
21.         GameObject instance = Instantiate(finish, new
22.             Vector3(0, 0, 0), Quaternion.identity) as GameObje
23.             ct;
24.         instance.transform.SetParent(Finish.transfor
25.             m);
26.         Finish.transform.parent = transform;
27.
28.         randomPoint = Random.Range(1, 4);
29.         instance.transform.position = cornerPos[rand
30.             omPoint - 1];
31.

```

```
22.     }
```

*Kode Sumber 4.9 Proses Pembuatan Titik Finish*

#### 4.2.2.2 Implementasi *Game Manager*

Tujuan dari penulis dalam pembuatan *game manager* adalah untuk mengatur jalannya permainan. Didalam *game manager* terdapat fungsi untuk membuat *maze* secara keseluruhan, mengatur banyaknya level permainan, mengatur perpindahan setiap level permainan, mengatur *user interface* pada permainan, dan mengatur suara pada permainan. Berikut ini beberapa fungsi yang terdapat pada *game manager*, dapat dilihat pada Kode Sumber 4.10 sampai Kode Sumber 4.14.

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5. using TMPro;
6.
7. public class GameManager : MonoBehaviour
8. {
9.
10.     [System.Serializable]
11.     public class Difficulty
12.     {
13.         public int mazeSize, trampolineCount, floorC
14.         ount;
15.         public Difficulty(int mazeSize, int trampoli
16.         neCount, int floorCount)
17.         {
18.             this.mazeSize = mazeSize;
19.             this.trampolineCount = trampolineCount;
20.
21.             this.floorCount = floorCount;
22.         }
23.     }
24. }
```

```
22.     public static GameManager instance = null;
23.     private MazeManager mazeManager;
24.     public int level = 0;
25.     private float timer;
26.     private float finishTime;
27.     private bool pauseTimer = true;
28.
29.     private TextMeshProUGUI levelText;
30.     private TextMeshProUGUI timeText;
31.
32.     [HideInInspector]public GameObject finishImage;
33.     [HideInInspector] public GameObject gameoverImage;
34.     [HideInInspector] public GameObject pauseImage;
35.     [HideInInspector] public GameObject inGame;
36.     [HideInInspector] public GameObject mainMenu;
37.     [HideInInspector] public GameObject howTo;
38.     [HideInInspector] public GameObject pauseButton;
39.     [HideInInspector] public GameObject congratulation;
40.     private GameObject leverLeft;
41.     private GameObject baseLeft;
42.     private GameObject leverRight;
43.     private GameObject baseRight;
44.     private ControlHandler control;
45.     private TextMeshProUGUI finishlevelText;
46.     private TextMeshProUGUI finishTimeText;
47.     private TextMeshProUGUI gameoverLevelText;
48.     private TextMeshProUGUI pauseLevelText;
49.
50.     public AudioSource sfxSource;
51.     public AudioSource musicSource;
52.
53.     void Awake()
54.     {
55.         if (instance == null)
56.         {
57.             instance = this;
58.         }
59.
```



```
60.     else if (instance != this)
61.     {
62.         Destroy(gameObject);
63.     }
64.
65.     level = 0;
66.     mazeManager = GetComponent<MazeManager>();
67.     InitVar();
68.     DontDestroyOnLoad(gameObject);
69.     InitGame();
70. }
71.
72. void InitVar()
73. {
74.     levelText = GameObject.Find("Level").GetComp
75.     onent<TextMeshProUGUI>();
76.     timeText = GameObject.Find("Time").GetCompon
77.     ent<TextMeshProUGUI>();
78.     finishImage = GameObject.Find("FinishImage")
79.     ;
80.     gameoverImage = GameObject.Find("GameOverIma
81.     ge");
82.     pauseImage = GameObject.Find("PauseImage");
83.
84.     inGame = GameObject.Find("InGame");
85.     mainMenu = GameObject.Find("MainMenu");
86.     howTo = GameObject.Find("HowToPlay");
87.     congratulation = GameObject.Find("Congratula
88.     tions");
89.     pauseButton = GameObject.Find("Pause Button"
90.     );
91.     finishlevelText = GameObject.Find("LevelText
92.     ").GetComponent<TextMeshProUGUI>();
93.     finishTimeText = GameObject.Find("TimeText")
94.     .GetComponent<TextMeshProUGUI>();
95.     gameoverLevelText = GameObject.Find("GameOve
96.     rLevelText").GetComponent<TextMeshProUGUI>();
97.     pauseLevelText = GameObject.Find("PauseLevel
98.     Text").GetComponent<TextMeshProUGUI>();
99.
100.     leverLeft = GameObject.Find("LeverLeft");
101.     baseLeft = GameObject.Find("BaseLeverLeft");
```

```

91.
92.     leverRight = GameObject.Find("LeverRight");
93.     baseRight = GameObject.Find("BaseLeverRight"
94. );
95.     control = GetComponent<ControlHandler>();
96.
97.     pauseButton.SetActive(false);
98.     howTo.SetActive(false);
99.     inGame.SetActive(false);
100.    finishImage.SetActive(false);
101.    gameOverImage.SetActive(false);
102.    pauseImage.SetActive(false);
103.    congratulation.SetActive(false);
104.
105.    Transparant();
106. }
107.
108. public void InitGame()
109. {
110.     PlayMusic();
111.     timer = 0;
112.     Difficulty difficulty = difficulties[leve
113. l];
114.     mazeManager.SetupScene(difficulty.mazeSiz
115. e, difficulty.trampolineCount, difficulty.floorCount
116. );
117. }
118. }

```

Kode Sumber 4.10 Proses Inisiasi Permainan

Fungsi *InitVar()* digunakan untuk mengatur semua *variable* yang berkaitan dengan objek 3D pada *game*. Sedangkan fungsi *InitGame()* dipanggil pertama ketika *game* dimulai didalamnya terdapat intruksi untuk memutar lagu *background* permainan, mengatur *timer* pada angka nol, dan membuat *maze* pada level pertama.

```

1. [System.Serializable]
2.     public class Difficulty
3.     {
4.         public int mazeSize, trampolineCount, floorC
           ount;
5.         public Difficulty(int mazeSize, int trampoli
           neCount, int floorCount)
6.         {
7.             this.mazeSize = mazeSize;
8.             this.trampolineCount = trampolineCount;
9.             this.floorCount = floorCount;
10.        }
11.    }
12.
13. private Difficulty[] difficulties = new Difficulty[]
14. {
15.     new Difficulty(5, 0, 0),
16.     new Difficulty(5, 1, 0),
17.     new Difficulty(5, 2, 0),
18.     new Difficulty(5, 2, 1),
19.     new Difficulty(10, 0, 0),
20.     new Difficulty(10, 3, 2),
21.     new Difficulty(10, 4, 3)
22. };

```

Kode Sumber 4.11 Pembuatan Level Permainan

Pada Kode Sumber 4.11 dapat diketahui bahwa pembuatan level permainan yang terdapat pada *game manager* berkaitan dengan ukuran *maze*, banyaknya jebakan trampolin dan banyaknya jebakan lubang. *Array* pada baris kode ke-13 menunjukkan bahwa terdapat 7 level permainan. Parameter pertama berisi ukuran *maze*, parameter kedua berisi banyak jebakan trampolin, dan parameter ketiga berisi banyak jebakan lubang.

```

1. public void NextLevel()

```

```

2.     {
3.         pauseTimer = false;
4.         timer = 0;
5.         level++;
6.         if (level < difficulties.Length)
7.         {
8.             Difficulty difficulty = difficulties[lev
9. el];
10.            mazeManager.RestartGame();
11.            mazeManager.SetupScene(difficulty.mazeSi
12. ze, difficulty.trampolineCount, difficulty.floorCoun
13. t);
14.        }
15.        else
16.        {
17.            congratulation.SetActive(true);
18.            Transparant();
19.            Pause();
20.        }
21.    }

```

Kode Sumber 4.12 Perpindahan Level Permainan

Fungsi *NextLevel()* pada Kode Sumber 4.12 terdapat beberapa intruksi seperti mengatur *timer* pada angka nol, dan membuat *maze* pada level selanjutnya. Jika level permainan telah habis atau berada pada level terakhir, maka halaman *congratulation* akan muncul dan tuas kontrol akan hilang (baris kode ke-12 sampai baris kode ke-17). Yang menandakan permainan berhasil diselesaikan.

```

1. public void GameOver()
2.     {
3.         timer = 0;
4.         Difficulty difficulty = difficulties[level];
5.         mazeManager.RestartGame();

```

```

6.         mazeManager.SetupScene(difficulty.mazeSize,
           difficulty.trampolineCount, difficulty.floorCount);
7.     }

```

*Kode Sumber 4.13 Kontrol Game Over*

Fungsi **GameOver()** pada Kode Sumber 4.13 terdapat beberapa intruksi seperti mengatur *timer* pada angka nol, dan melakukan pembuatan *maze* kembali pada level dimana fungsi ini digunakan. Fungsi ini banyak dipanggil ketika beberapa halaman *user interface* muncul.

```

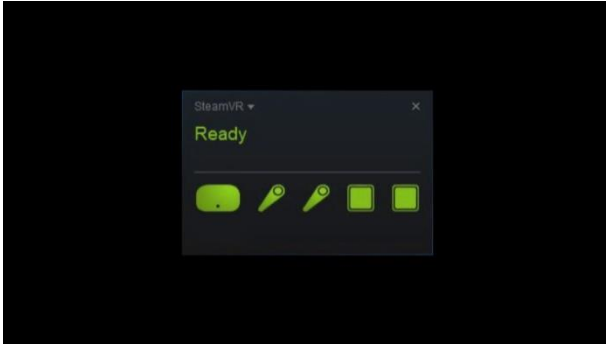
1. public void PlayMusic()
2. {
3.     musicSource.Play();
4. }
5.
6. public void PlaySFX()
7. {
8.     sfxSource.Play();
9. }

```

*Kode Sumber 4.14 Kontrol Suara Permainan*

#### 4.2.2.3 Implementasi Realitas Virtual pada HTC Vive

Implementasi penggunaan realitas virtual pada alat HTC Vive memerlukan plugin SteamVR. Plugin ini dapat diunduh di asset store unity. Sebelum mengunduh SteamVR, perlu dipastikan alat HTC Vive telah tersambung dengan benar dengan komputer. Dengan cara mengunduh aplikasi SteamVR pada komputer. Ketika aplikasi SteamVR dijalankan pada komputer akan terdapat beberapa indikator seperti pada gambar 4.12 dibawah.

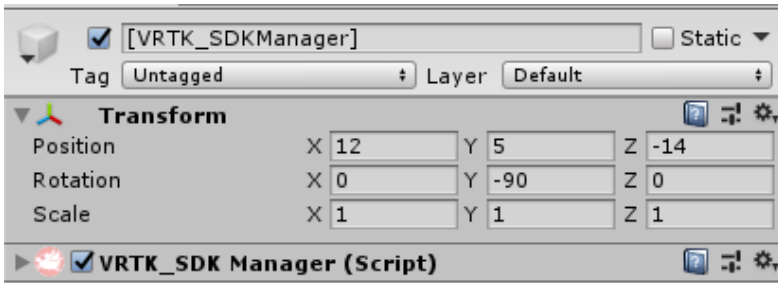


Gambar 4.12 Indikator SteamVR

(Sumber : <https://www.youtube.com/watch?v=2mAIUn-Ykz0>)

Pada gambar diatas terdapat 5 indikator yang setiap indikator berfungsi menunjukan status dari setiap komponen alat HTC Vive. Pada indikator pertama yaitu pada indikator paling kiri menunjukan status dari Vive Headset, lalu yang kedua dan ketiga yaitu indikator untuk Vive Controller. Untuk indikator kelima dan keenam menunjukkan status pada Vive Base Stations. Jika indikator tidak berwarna hijau maka cek kembali koneksi pada kabel yang terhubung ke komputer atau yang terhubung ke HTC Vive.

Setelah memastikan alat HTC Vive telah tersambung secara benar dan tidak terdapat *error*, maka dilanjutkan dengan mengunduh *plugin* SteamVR. Setelah mengunduh *plugin* SteamVR hal kedua yang diperlukan adalah *plugin* VRTK. Untuk dapat menggunakan fitur dari Vive Headset maka perlu dilakukan *setup* pada unity. *Setup* VRTK dapat dilakukan dengan cara menggunakan *asset* yang terdapat pada library VRTK seperti pada Gambar 4.13.

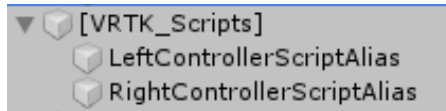


Gambar 4.13 Setup Asset VRTK

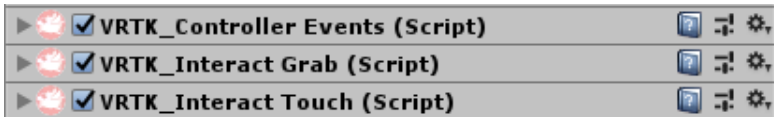
Untuk dapat menggunakan Vive Headset maka harus menambahkan *script VRTK SDK Manager* pada permainan. *Script* ini digunakan untuk menampung *plugin* dari SteamVR yang sebelumnya telah dimasukan kedalam *project*. Di dalamnya terdapat *script* untuk membangkitkan *prefabs controller* yang akan ditampilkan saat memulai permainan. Selain itu, juga terdapat *prefabs* untuk *headset rig*, yaitu digunakan agar Vive Headset dapat menjalankan fungsinya untuk menampilkan *game* dengan konsep realitas virtual.

#### 4.2.2.4 Implementasi Kontrol Maze

Untuk mengatur kontrol *maze* pada permainan ini maka pemain harus melakukan interaksi pada tuas yang sudah disediakan. Interaksi yang dilakukan adalah dengan memegang tuas menggunakan Vive Controller. Untuk melakukan implementasi kontrol *maze*, penulis menggunakan *asset VRTK* yang dapat diunduh dari *asset store unity*. Untuk dapat melakukan interaksi menggunakan *controller* maka diperlukan beberapa komponen *script*, yaitu *VRTK Controller Events*, *VRTK Interact Grab* dan *VRTK Interact Touch*. *Script* ini dimasukan kedalam objek *controller* yang telah dibuat sebelumnya seperti pada Gambar 4.14 dan Gambar 4.15.

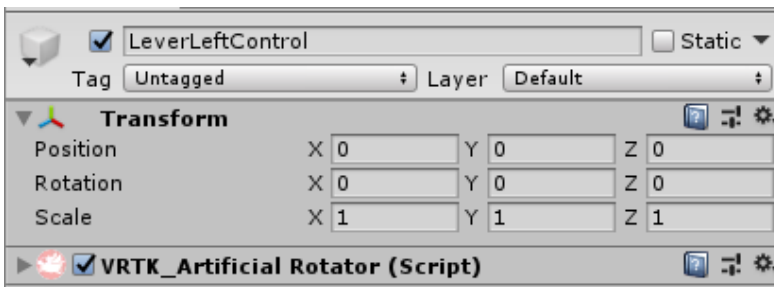


Gambar 4.14 Objek Controller



Gambar 4.15 Implementasi Interaksi Grab Controller

Selain memberikan *script* kepada *controller* HTC Vive, *script* juga diperlukan pada objek tuas kontrol. Karena penulis menggunakan objek kontrol berbentuk tuas maka komponen *script* yang digunakan yaitu VRTK *Artificial Rotator*. Untuk implementasi *script* pada tuas dapat dilihat pada Gambar 4.16.



Gambar 4.16 Implementasi Pada Tuas Kontrol

```

1. using UnityEngine;
2. using UnityEngine.UI;
3.
4. public class ControlHandler : MonoBehaviour
5. {
6.     GameManager gameManager;
7.

```



```
8.     public ControlHandlerLeft handlerLeft;
9.     public ControlHandlerRight handlerRight;
10.
11.    private float smooth = 5.0f;
12.
13.    private void Start()
14.    {
15.        gameManager = GetComponent<GameManager>();
16.
17.        Initial();
18.
19.    }
20.
21.    private void Initial()
22.    {
23.        handlerLeft = GameObject.Find("LeverLeftControl").GetComponent<ControlHandlerLeft>();
24.        handlerRight = GameObject.Find("LeverRightControl").GetComponent<ControlHandlerRight>();
25.    }
26.
27.    private void Update()
28.    {
29.        float tiltAroundZ = handlerRight.rightValue;
30.        float tiltAroundX = handlerLeft.leftValue;
31.
32.        Quaternion target = Quaternion.Euler(tiltAroundZ, 0, tiltAroundX);
33.
34.        transform.rotation = Quaternion.Slerp(transform.rotation, target, Time.deltaTime * smooth);
35.    }
36.
37. }
```

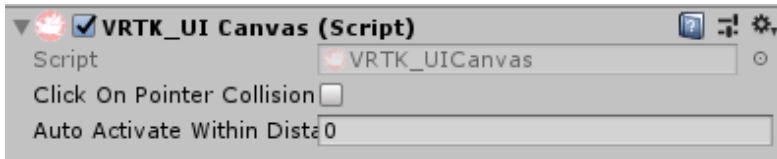
*Kode Sumber 4.15 Implementasi Kontrol Rotasi Maze*

Pada Kode Sumber 4.15 dilakukan proses pengambilan nilai yang muncul ketika pemain berinteraksi dengan tuas kontrol.

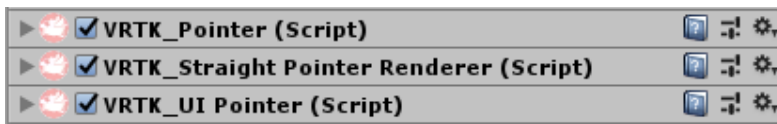
Pada fungsi *Initial()* di kode baris ke-21 berfungsi mengambil nilai dari dua tuas kontrol. Fungsi *Update()* pada kode baris ke-27 berisi intruksi untuk mengatur rotasi *maze* (baris kode ke-34). Menggunakan fungsi pada *library* Unity *transform.rotation* dan *Quaternion.Slerp*.

#### 4.2.2.5 Implementasi Kontrol *User Interface*

Agar dapat memilih menu yang ada didalam permainan diperlukan beberapa komponen *script* pendukung dari VRTK. Komponen *script* digunakan dalam objek *controller* dan juga objek *canvas* yang merupakan *user interface* dalam permainan yang sudah dibuat sebelumnya. *Script* yang dibutuhkan yaitu VRTK *Pointer*, VRTK *Straight Pointer Renderer*, VRTK *UI Pointer*, dan VRTK *UI Canvas*. Implementasi *user interface* pada permainan *virtual reality* menggunakan VRTK dapat dilihat pada Gambar 4.17 dan Gambar 4.18.



Gambar 4.17 Implementasi Pada Objek Canvas



Gambar 4.18 Implementasi Pointer Untuk User Interface

#### 4.2.2.6 Implementasi Jebakan Trampolin

Jebakan trampolin dibuat secara *random* atau acak dan akan menghilang setelah beberapa detik lalu muncul kembali pada lokasi lain secara *random* dalam *board* permainan. Trampolin tidak

akan muncul pada empat lokasi yaitu lokasi dimana objek bola dan objek *finish* akan muncul. Ketika bola menyentuh objek jebakan trampolin. Maka bola akan terdorong keatas dan jatuh secara *random*. Bola dapat jatuh kembali ke *board* permainan atau jatuh keluar *board* permainan. Jika bola jatuh keluar dari *board* permainan maka halaman *game over* akan muncul. Untuk lebih lengkapnya, proses pembuatan jebakan trampolin dapat dilihat pada Kode Sumber 4.15 sampai Kode Sumber 4.17.

```

1. void UpdateTrapPosition()
2. {
3.     Vector3 center = new Vector3((-
        rows / 2) + wallLength / 2, -0.45f, (-
        columns / 2));
4.     Vector3 currentPos = center;
5.     positionStore.Clear();
6.
7.     for (int x = 0; x < columns; x++)
8.     {
9.         for (int y = 0; y < rows; y++)
10.        {
11.            currentPos = new Vector3(center.x + (y *
                wallLength), center.y, center.z + (x * wallLength))
12.            ;
13.            positionStore.Add(currentPos);
14.        }
15.    }

```

Kode Sumber 4.16 Proses Penyimpanan Posisi Jebakan

Fungsi *UpdateTrapPosition()* pada Kode Sumber 4.16 berfungsi untuk menyimpan setiap kemungkinan posisi jebakan pada *maze* seperti pada kode sumber ke-7 sampai kode sumber ke-11. Dan kemudian semua posisi jebakan disimpan pada *variable positionStore* seperti pada kode sumber ke-12. *Variable* tersebut akan digunakan untuk membuat jebakan.

```
1. private void InitTrampoline()
2. {
3.     Trampoline = new GameObject("Trampoline");
4.     Trampoline.transform.parent = transform;
5. }
6. void TrampolineSpawner(GameObject prefab)
7. {
8.     int floorSum = rows * columns;
9.     int number;
10.    bool shouldCreateTrap = Trampoline.transform.childCount == 0;
11.
12.    if (shouldCreateTrap)
13.    {
14.
15.        if (spawnTime <= 0)
16.        {
17.            List<int> takenCoordinates = new List<int>();
18.
19.            for (int i = 0; i < trampolineCount; i++)
20.            {
21.
22.                while(true)
23.                {
24.                    number = Random.Range(0, (floorSum - 1));
25.                    bool isTaken = takenCoordinates.Contains(number);
26.                    bool notCorner = number != 0 && number != (floorSum - 1) && number != (columns - 1) && number != (rows * (columns - 1));
27.
28.                    if (notCorner && !isTaken)
29.                        break;
30.                }
31.
32.                takenCoordinates.Add(number);
33.
34.                Vector3 trapPosition = floors[number].transform.position;
35.                trapPosition.y += 0.05f;
```

```

36.         instance = Instantiate(prefab, trapP
osition, Quaternion.identity) as GameObject;
37.         instance.transform.SetParent(Trampol
ine.transform);
38.         instance.transform.rotation = transf
orm.rotation;
39.         instance.transform.localScale = new
Vector3(transform.localScale.x, 0.01f, transfo
rm.localScale.z);
40.         Trampoline.transform.parent = transf
orm;
41.         Destroy(instance, 2.0f);
42.
43.     }
44.     spawnTime = 1.0f;
45. }
46. }
47.
48.     spawnTime -= Time.deltaTime;
49. }

```

*Kode Sumber 4.17 Proses Pembuatan Jebakan Trampoline*

Fungsi *TrampolineSpawner()* digunakan untuk membuat jebakan pada *maze*. Posisi jebakan dibuat pada lokasi *random* dalam *maze*, dengan cara melakukan pemilihan lokasi yang telah disimpan pada *variable* secara acak dapat dilihat pada kode baris ke-34. Setiap jebakan yang dibuat akan dihapus setiap dua detik lalu akan muncul kembali pada posisi *random* lain setelah satu detik dapat dilihat pada kode baris ke-41.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class PlayerBehaviour : MonoBehaviour
6. {
7.     [HideInInspector] public GameManager gameManager
;

```

```

8.     private Rigidbody rb;
9.
10.    private void OnTriggerEnter(Collider other)
11.    {
12.        if (other.gameObject.tag == "Trap")
13.        {
14.            rb.AddForce(0, 10.0f, 0, ForceMode.Impul
15.            se);
16.        }
17.    }
18.
19.    private void Start()
20.    {
21.        gameManager = GameObject.Find("Maze").GetCom
22.        ponent<GameManager>();
23.        rb = GetComponent<Rigidbody>();
24.    }

```

*Kode Sumber 4.18 Proses Penerapan Efek Trampolin*

Pada Kode Sumber 4.18 di kode baris ke-14 efek trampolin diterapkan pada bola. Dengan menggunakan fungsi *AddForce* ketika bola menyentuh jebakan trampolin. Jadi ketika bola menyentuh jebakan trampolin, maka bola akan terlempar ke atas searah sumbu Y.

#### **4.2.2.7 Implementasi Jebakan Lubang**

Jebakan lubang dibuat dengan cara menghapus lantai *maze* secara *random* atau acak. Setelah lantai terhapus maka dalam beberapa detik lantai *maze* yang sudah dihapus akan dimunculkan lagi, lalu proses akan menghapus lantai *maze* lain secara *acak*. Proses tidak akan menghapus lantai pada empat titik. Yaitu titik dimana objek bola dan objek *finish* akan dibuat. Ketika bola melewati jebakan lubang maka bola akan terjatuh kebawah, lalu halaman *game over* akan muncul. Untuk lebih lengkap proses

pembuatan jebakan lubang pada *maze* dapat dilihat pada Kode Sumber 4.18.

```
1. void FloorRemover()
2.     {
3.         int floorSum = rows * columns;
4.         int number;
5.
6.         if (floorTimer <= 1)
7.         {
8.             foreach (GameObject floor in floors)
9.             {
10.                floor.SetActive(true);
11.            }
12.        }
13.
14.        if (floorTimer <= 0)
15.        {
16.            List<int> takenCoordinates = new List<int>();
17.
18.            for (int i = 0; i < floorCount; i++)
19.            {
20.                while (true)
21.                {
22.                    number = Random.Range(0, (floorSum - 1));
23.                    bool isTaken = takenCoordinates.Contains(number);
24.                    bool notCorner = number != 0 && number != (floorSum - 1) && number != (columns - 1) && number != (rows * (columns - 1));
25.
26.                    if (notCorner && !isTaken)
27.                        break;
28.                }
29.
30.                takenCoordinates.Add(number);
31.
32.                floors[number].SetActive(false);
33.
34.            }
```

```

35.
36.         floorTimer = 2.0f;
37.     }
38.
39.         floorTimer -= Time.deltaTime;
40.     }

```

Kode Sumber 4.19 Proses Pembuatan Jebakan Lubang

Fungsi **FloorRemover()** pada Kode Sumber 4.19 dengan cara menghapus posisi lantai pada *maze* secara *random* yang telah disimpan sebelumnya seperti pada kode baris ke-32. Setiap lantai akan dihapus setiap dua detik dan akan muncul setiap satu detik setelah dihapus seperti pada kode baris ke-6 dan kode baris ke-36.

#### 4.2.2.8 Implementasi Timer

*Timer* digunakan sebagai informasi kecepatan pemain dalam menyelesaikan suatu level. Implementasi *timer* dilakukan didalam *game manager*. *Timer* akan berhenti ketika pemain berhasil menyelesaikan level, pemain gagal menyelesaikan level atau *game over*, dan pemain memilih tombol *pause*. *Timer* akan diulang kembali atau reset setiap kali level permainan baru dimulai. Hasil *timer* akan ditampilkan pada halaman *win condition* dan halaman *game over*. Untuk lebih lengkapnya proses implementasi timer dapat dilihat pada Kode Sumber 4.19.

```

1. void Update()
2. {
3.     if(pauseTimer == false)
4.     {
5.         timer += Time.deltaTime;
6.     }
7.     string minutes = ((int)timer / 60).ToString("f0"
8. );
9.     string seconds = (timer % 60).ToString("f0");
10. }

```



```

11.
12. public void Pause()
13. {
14.     pauseTimer = true;
15. }
16.
17.
18. public void Resume()
19. {
20.     pauseTimer = false;
21. }

```

*Kode Sumber 4.20 Proses Pembuatan Timer*

Pada kode baris ke-3 di Kode Sumber 4.20 menunjukkan bahwa *timer* akan berjalan ketika permainan tidak dalam menu *pause*. Lalu pada kode baris ke-7 dan kode baris ke-8 digunakan untuk mencetak timer didalam permainan.

#### 4.2.2.9 Implementasi Level Permainan

Pada permainan ini tersedia 7 level permainan yang dapat dimainkan oleh pemain. Setiap level permainan memiliki kesulitan yang berbeda berdasarkan jebakan dan besarnya *board* permainan. Setiap level dibangkitkan maka proses akan melakukan *generate* ulang *maze*, sehingga meskipun pada level permainan yang sama setiap *maze* yang muncul pasti berbeda.



*Gambar 4.19 Implementasi Level 1 Permainan*

Pada Gambar 4.19 diatas, dapat dilihat pada level 1 permainan maze berukuran 5 x 5. Pada level ini masih belum terdapat jebakan, hal ini bertujuan agar pemain terbiasa dengan kontrol yang tersedia.



*Gambar 4.20 Implementasi Level 2 Permainan*

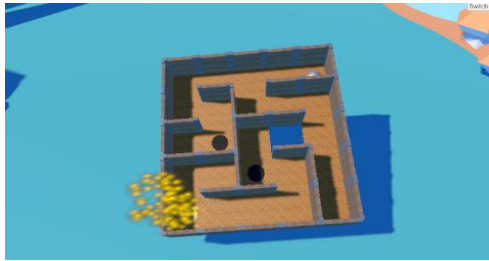
Pada Gambar 4.20 diatas, dapat dilihat pada level 2 permainan maze masih berukuran 5 x 5. Pada level ini sudah terdapat jebakan. Yaitu jebakan trampolin dan hanya berjumlah 1, hal ini bertujuan agar pemain terbiasa dengan munculnya jebakan pada permainan.



*Gambar 4.21 Implementasi Level 3 Permainan*

Pada Gambar 4.21 diatas, dapat dilihat pada level 3 permainan maze masih berukuran 5 x 5. Pada level ini terdapat jebakan, yaitu

jebakan trampolin dan berjumlah 2, hal ini bertujuan untuk menambah tingkat kesulitan pada level permainan ini.



*Gambar 4.22 Implementasi Level 4 Permainan*

Pada Gambar 4.22 diatas, dapat dilihat pada level 4 permainan maze masih berukuran 5 x 5. Pada level ini terdapat 2 jenis jebakan, yaitu jebakan trampolin dan lubang. Jebakan trampolin masih sama dengan level sebelumnya yaitu berjumlah 2, dan jebakan lubang berjumlah 1. Hal ini bertujuan untuk menambah tingkat kesulitan pada level permainan ini.



*Gambar 4.23 Implementasi Level 5 Permainan*

Pada Gambar 4.23 diatas, dapat dilihat pada level 5 permainan maze berukuran 10 x 10. Pada level ini tidak terdapat jebakan. Hal ini bertujuan agar pemain dapat terbiasa dengan besar maze 10 x 10.



*Gambar 4.24 Implementasi Level 6 Permainan*

Pada Gambar 4.24 diatas, dapat dilihat pada level 6 permainan maze masih berukuran 10 x 10. Pada level ini terdapat 2 jenis jebakan, yaitu jebakan trampoline dan lubang. Jebakan trampoline berjumlah 3, dan jebakan lubang berjumlah 2. Hal ini bertujuan untuk menambah tingkat kesulitan pada level permainan ini.



*Gambar 4.25 Implementasi Level 7 Permainan*

Pada Gambar 4.25 diatas, dapat dilihat pada level 7 permainan maze masih berukuran 10 x 10. Pada level ini terdapat 2 jenis jebakan, yaitu jebakan trampoline dan lubang. Jebakan trampoline berjumlah 4, dan jebakan lubang berjumlah 3. Hal ini bertujuan untuk menambah tingkat kesulitan pada level permainan ini.

## BAB V PENGUJIAN DAN EVALUASI

Pada bab ini akan dijelaskan mengenai rangkaian uji coba dan evaluasi yang dilakukan. Proses pengujian dilakukan menggunakan metode *blackbox* 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 dijelaskan mengenai tiap-tiap lingkungan uji coba aplikasi.

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

*Tabel 5.1 Lingkungan Perangkat Keras*

Perangkat	Spesifikasi
Perangkat Keras	<ul style="list-style-type: none"><li>• Prosesor Intel(R) Core (TM) i7-770 CPU @ 3.60GHz (8 CPUs), ~3.6 GHz</li><li>• RAM 8 Gigabyte</li><li>• Graphic card NVIDIA GeForce GTX 1060 3GB</li><li>• HTC Vive</li></ul>
Perangkat Lunak	<ul style="list-style-type: none"><li>• Sistem operasi Windows 10 Home Single Language 64-bit</li><li>• Unity 2019.1.1f1 Personal (64-bit)</li></ul>

### 5.2 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 3.1.2. Skenario uji coba fungsionalitas yang dilakukan terhadap aplikasi yang dibangun dijelaskan pada Tabel 5.2.

*Tabel 5.2 Skenario Uji Coba Fungsionalitas*

<b>Kode Uji Coba</b>	<b>Deskripsi Uji Coba</b>
UF-001	Uji coba pada menu permainan
UF-002	Uji coba dalam permainan
UF-003	Uji coba akhir permainan

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

### 5.2.1 Uji Coba pada Menu Permainan

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

*Tabel 5.3 Hasil Uji Coba pada Menu Permainan*

<b>ID</b>	<b>UF-001</b>
Nama	Uji coba pada menu permainan
Tujuan uji coba	Pengguna mengetahui fungsionalitas tombol yang ada pada menu permainan
Kondisi awal	Pemain berada pada halaman menu utama
<b><i>Skenario 1</i></b>	<b><i>Pemain memilih tombol Play</i></b>
Masukan	Memilih tombol <i>play</i> pada dunia virtual
Keluaran yang diharapkan	Tuas muncul pada area permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada halaman permainan
<b><i>Skenario 2</i></b>	<b><i>Pemain memilih tombol How To Play</i></b>

Masukan	Menekan tombol <i>how to play</i> pada dunia virtual
Keluaran yang diharapkan	Halaman penjelasan cara bermain muncul
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada halaman cara bermain
<b><i>Skenario 3</i></b>	<b><i>Pemain memilih tombol exit</i></b>
Masukan	Menekan tombol <i>exit</i> pada dunia virtual
Keluaran yang diharapkan	Keluar dari aplikasi permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Permainan berakhir

Hasil uji dari skenario 1 berpindah ke halaman permainan yang dapat dilihat pada Gambar 4.4 sampai dengan Gambar 4.7, skenario 2 berpindah ke halaman cara bermain yang dapat dilihat pada Gambar 4.3, dan skenario 3 pemain akan keluar dari permainan.

### 5.2.2 Uji Coba dalam Permainan

Pada subbab ini dijelaskan mengenai skenario yang dilakukan dan hasil yang didapatkan dari pengujian fungsionalitas pada permainan. Penjelasan disajikan dengan menampilkan kondisi awal, masukan, keluaran, hasil yang dicapai, dan kondisi akhir. Skenario dan hasil pengujian secara lengkap dapat dilihat pada Tabel 5.4.

Tabel 5.4 Hasil Uji Coba dalam Permainan

<b>ID</b>	<b>UF-002</b>
Nama	Uji coba dalam permainan
Tujuan uji coba	Pengguna mengetahui fungsionalitas interaksi yang ada pada permainan
Kondisi awal	Pemain berada pada halaman permainan
<b><i>Skenario 1</i></b>	<b><i>Pemain dapat menyentuh tuas kontrol</i></b>

Masukan	Pemain mengarahkan <i>controller</i> pada tuas kontrol yang tersedia
Keluaran yang diharapkan	Pemain dapat menyentuh tuas kontrol dalam permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain dapat memahami kontrol <i>maze</i>
<b><i>Skenario 2</i></b>	<b><i>Pemain dapat melakukan grab pada tuas kontrol</i></b>
Masukan	Pemain melakukan <i>grab</i> pada <i>controller</i> ketika menyentuh tuas kontrol yang tersedia
Keluaran yang diharapkan	Pemain dapat mengerti cara kontrol <i>board</i> permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain dapat mengatur rotasi <i>maze</i> menggunakan tuas kontrol
<b><i>Skenario 3</i></b>	<b><i>Pemain dapat menyelesaikan level permainan</i></b>
Masukan	Pemain mengatur rotasi <i>maze</i> menggunakan tuas kontrol yang tersedia
Keluaran yang diharapkan	Bola mencapai titik <i>finish</i> pada <i>board</i> permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Tuas kontrol hilang dan muncul halaman <i>win condition</i> yang berisi <i>next level</i> , <i>restart level</i> , dan <i>exit level</i> .
<b><i>Skenario 4</i></b>	<b><i>Pemain dapat kalah dalam permainan</i></b>
Masukan	Pemain mengarahkan bola melewati jebakan yang ada
Keluaran yang diharapkan	Bola jatuh atau keluar dari <i>board</i> permainan
Hasil uji coba	Berhasil



Kondisi Akhir	Tuas kontrol hilang dan muncul halaman <i>game over</i> yang berisi <i>restart level</i> , dan <i>exit level</i>
<b>Skenario 5</b>	<b><i>Pemain dapat menghentikan permainan sementara</i></b>
Masukan	Pemain memilih tombol <i>pause</i> yang tersedia dalam permainan
Keluaran yang diharapkan	Permainan berhenti sementara dan muncul halaman <i>pause</i> yang berisi <i>restart level</i> , <i>resume level</i> , dan <i>exit level</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Permainan berhenti sementara dan muncul halaman <i>pause</i>
<b>Skenario 6</b>	<b><i>Pemain dapat melanjutkan permainan yang sudah dihentikan sementara</i></b>
Masukan	Pemain memilih tombol <i>resume level</i> yang terdapat pada halaman <i>pause</i>
Keluaran yang diharapkan	Pemain dapat melanjutkan kembali permainan dan halaman <i>pause</i> hilang
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain dapat melanjutkan kembali permainan dan halaman <i>pause</i> hilang
<b>Skenario 7</b>	<b><i>Pemain dapat mengulang kembali level permainan</i></b>
Masukan	Pemain memilih tombol <i>restart level</i> yang terdapat pada halaman <i>pause</i>
Keluaran yang diharapkan	Pemain dapat mengulang kembali level permainan yang sedang dimainkan
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain dapat mengulang kembali level permainan yang sedang dimainkan
<b>Skenario 8</b>	<b><i>Pemain dapat keluar ke halaman menu utama</i></b>
Masukan	Pemain memilih tombol <i>exit level</i> yang terdapat pada halaman <i>pause</i>

Keluaran yang diharapkan	Pemain berpindah ke halaman menu utama
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada pada halaman menu utama
<b><i>Skenario 9</i></b>	<b><i>Pemain dapat melanjutkan level permainan</i></b>
Masukan	Pemain memilih tombol <i>next level</i> yang terdapat pada halaman <i>win condition</i>
Keluaran yang diharapkan	Pemain berpindah ke level permainan selanjutnya
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain dapat bermain pada level permainan selanjutnya
<b><i>Skenario 10</i></b>	<b><i>Bola dapat terdorong keatas</i></b>
Masukan	Pemain mengarahkan bola melewati jebakan trampolin
Keluaran yang diharapkan	Bola terdorong keatas menjauhi <i>board</i> permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Posisi bola berpindah akibat dorongan keatas
<b><i>Skenario 11</i></b>	<b><i>Bola dapat jatuh kebawah</i></b>
Masukan	Pemain mengarahkan bola melewati jebakan lubang
Keluaran yang diharapkan	Bola jatuh kebawah menjauhi <i>board</i> permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Posisi bola berpindah kebawah <i>board</i> permainan

Hasil uji dari skenario 1 yaitu pemain dapat memahami kontrol *board* permainan. Selanjutnya skenario 2 yaitu pemain dapat mengerti cara kontrol tuas. Selanjutnya skenario 3 yaitu pemain dapat menyelesaikan permainan dengan cara mengarahkan

bola sampai ke titik *finish*. Untuk skenario 4 pemain akan kalah dalam level permainan jika menjatuhkan bola keluar *board* permainan. Skenario 10 bola akan terlempar ke udara ketika pemain mengarahkan bola melewati jebakan trampolin. Bola akan jatuh dari udara dan mendarat ke *board* permainan atau keluar *board* permainan. Lalu skenario 11 bola akan jatuh kebawah *board* permainan ketika pemain mengarahkan bola melewati jebakan lubang. Bola akan jatuh dari *board* permainan dan halaman *game over* akan muncul.

### 5.2.3 Uji Coba Akhir Permainan

Pada subbab ini dijelaskan mengenai skenario yang dilakukan dan hasil yang didapatkan dari pengujian fungsionalitas akhir permainan. Penjelasan disajikan dengan menampilkan kondisi awal, masukan, keluaran, hasil yang dicapai, dan kondisi akhir. Skenario dan hasil pengujian secara lengkap dapat dilihat pada Tabel 5.5.

Tabel 5.5 Hasil Uji Coba Akhir Permainan

ID	UF-003
Nama	Uji Coba akhir permainan
Tujuan uji coba	Pengguna mengetahui fungsionalitas tombol yang ada pada akhir permainan
Kondisi awal	Pemain berada pada halaman permainan dalam kondisi menang/kalah
<b><i>Skenario 1</i></b>	<b><i>Pemain memilih tombol exit level</i></b>
Masukan	Memilih tombol <i>exit level</i> pada panel informasi kekalahan / kemenangan
Keluaran yang diharapkan	Pemain berpindah ke halaman menu utama permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berpindah ke halaman menu utama permainan
<b><i>Skenario 2</i></b>	<b><i>Pemain memilih tombol exit</i></b>

Masukan	Menekan tombol <i>exit</i> pada halaman menu utama
Keluaran yang diharapkan	Keluar dari aplikasi permainan
Hasil uji coba	Berhasil
Kondisi Akhir	Permainan berakhir

Hasil uji dari skenario 1 berpindah ke halaman menu utama permainan yang dapat dilihat pada Gambar 4.1 dan skenario 2 keluar dari permainan, pemain akan keluar dari permainan.

### 5.3 Pengujian Pengguna

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

#### 5.3.1 Skenario Uji Coba Pengguna

Dalam melakukan pengujian aplikasi ini, penguji diminta untuk mencoba menggunakan aplikasi yang bersangkutan dan mencoba semua fungsionalitas serta fitur yang tersedia. Selain itu, pengguna juga diminta memberikan saran untuk pengembangan aplikasi selanjutnya.

Pengujian aplikasi oleh pengguna dilakukan dengan sebelumnya memberi informasi seputar aplikasi, kegunaan, dan fitur-fitur yang dimiliki. Setelah informasi tersampaikan, pengguna kemudian diarahkan untuk langsung mencoba aplikasi dengan spesifikasi lingkungan yang sama persis dengan yang telah diuraikan pada uji coba fungsionalitas.

Jumlah pengguna yang terlibat dalam pengujian aplikasi ini sebanyak 16 orang. Dalam melakukan pengujian, pengguna melakukan percobaan sebanyak satu kali penggunaan.

Ketika memberikan penilaian dan tanggapan terhadap aplikasi, penguji diberikan formulir pengujian aplikasi. Formulir pengujian aplikasi ini memiliki beberapa aspek penilaian dan pada bagian akhir terdapat permintaan saran serta kritik untuk perbaikan fitur aplikasi. Nilai yang diberikan rentang nilai 1 hingga 5 dengan rincian pada Table 5.6. Pada bagian akhir terdapat saran untuk perbaikan fitur. Detil kuesioner pengguna dapat dilihat pada Tabel 5.7.

*Tabel 5.6 Rentang Nilai*

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

### 5.3.2 Tujuan Pertanyaan Kuesioner

Kuesioner ini dibuat untuk mengetahui tanggapan dari pemain dan untuk mengukur apakah aplikasi *game* Trapball sudah berjalan sesuai dengan rancangan. Tujuan dari pertanyaan kuisisioner dapat dilihat pada Tabel 5.7.

*Tabel 5.7 Tujuan Pertanyaan*

No.	Isi Pertanyaan	Tujuan Pertanyaan
1.	Aplikasi permainan memiliki objek dengan desain, tampilan, dan warna yang menarik	Untuk mengetahui aplikasi permainan sudah memiliki objek dengan desain, tampilan dan warna yang

		menarik untuk pemain
2.	Aplikasi permainan memiliki antarmuka yang mudah dimengerti	Untuk mengetahui aplikasi permainan sudah memiliki antarmuka yang mudah dipahami dan dimengerti oleh pemain
3.	Aplikasi permainan memiliki informasi dan instruksi yang mudah dipahami	Untuk mengetahui aplikasi permainan sudah memiliki informasi dan instruksi cara bermain sehingga pemain mudah memahami kontrol permainan
4.	Aplikasi memiliki area permainan dengan tampilan menarik	Untuk mengetahui apakah aplikasi permainan sudah memiliki area permainan yang menarik bagi pemain
5.	Aplikasi permainan nyaman ketika dimainkan	Untuk mengetahui apakah pemain merasa nyaman ketika bermain aplikasi permainan
6.	Aplikasi permainan memiliki kontrol yang mudah dimengerti	Untuk mengetahui apakah aplikasi permainan memiliki kontrol yang dapat

		digunakan dengan mudah oleh pemain
7.	Saya dapat berinteraksi dengan tuas kontrol dengan mudah	Untuk mengetahui apakah aplikasi permainan memiliki tuas kontrol yang mudah untuk digunakan
8.	Saya merasa tertantang untuk menyelesaikan permainan	Untuk mengetahui apakah aplikasi permainan memiliki level permainan yang menarik untuk diselesaikan
9.	Aplikasi permainan memiliki jebakan yang menantang	Untuk mengetahui apakah aplikasi permainan memiliki jebakan yang menambah kesulitan dari level permainan
10.	Saya merasa tertarik ingin memainkan permainan ini kembali	Untuk mengetahui apakah aplikasi permainan memiliki <i>gameplay</i> yang menarik bagi pemain
11.	Permainan dapat berjalan dengan lancar tanpa adanya <i>lag</i> dan <i>crash</i>	Untuk mengetahui apakah aplikasi permainan dapat berjalan dengan lancar

12.	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan	Untuk mengetahui apakah aplikasi permainan memiliki level permainan yang sesuai dengan rancangan
-----	---	--

### 5.3.3 Karakteristik Pengujian Pengguna

Pada subbab ini ditunjukkan karakteristik pengujian pengguna yang bertindak sebagai partisipan untuk menguji permainan yang telah dibangun. Partisipan yang melakukan pengujian pada permainan ini berjumlah 16 orang. Semua partisipan yang melakukan pengujian memiliki pekerjaan sebagai mahasiswa. Partisipan yang melakukan pengujian memiliki rentan usia pada umur 22 tahun sampai 23 tahun. Seluruh partisipan mengisi kuesioner yang telah disediakan setiap kali selesai melakukan pengujian pada permainan.

### 5.3.4 Hasil Pengujian Pengguna

Sistem penilaian didasarkan pada skala perhitungan satu sampai enam dimana skala satu menunjukkan nilai terendah dan skala enam menunjukkan skala tertinggi. Penilaian akhir kemudian dilakukan dengan menghitung berapa banyak penguji yang memilih suatu skala tertentu dan kemudian dicari nilai rata-ratanya. Hasil uji coba dipaparkan secara lengkap dengan disertai tabel yang dapat dilihat pada Tabel 5.8 dan Tabel 5.9

*Tabel 5.8 Hasil Pengujian Pengguna*

No	Parameter	Penilaian					Rata - rata
		1	2	3	4	5	
1	Aplikasi permainan memiliki objek	0	0	0	8	8	4.5



	dengan desain, tampilan, dan warna yang menarik						
2	Aplikasi permainan memiliki antarmuka yang mudah dimengerti	0	0	1	5	9	4.2
3	Aplikasi permainan memiliki informasi dan instruksi yang mudah dipahami	0	0	0	10	6	4.3
4	Aplikasi memiliki area permainan dengan tampilan menarik	0	0	1	8	7	4.3
5	Aplikasi permainan nyaman ketika dimainkan	0	1	2	8	5	4.1
6	Aplikasi permainan memiliki kontrol yang mudah dimengerti	0	0	3	6	7	4.2
7	Saya dapat berinteraksi dengan tuas kontrol dengan mudah	0	3	3	9	1	3.5
8	Saya merasa tertantang untuk menyelesaikan permainan	0	0	1	3	12	4.6
9	Aplikasi permainan memiliki jebakan yang menantang	0	3	3	4	6	3.8
10	Saya merasa tertarik ingin memainkan	0	0	1	8	7	4.3

	permainan ini kembali						
11	Permainan dapat berjalan dengan lancar tanpa adanya <i>lag</i> dan <i>crash</i>	0	0	0	4	12	4.7
12	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan	1	0	0	7	8	4.1

Tabel 5.9 Hasil Akhir Pengujian Pengguna

No	Pernyataan	Rata-rata	Total	Total (%)
<b>Parameter Antarmuka</b>				
1	Aplikasi permainan memiliki objek dengan desain, tampilan, dan warna yang menarik	4.5	4.3	86.6%
2	Aplikasi permainan memiliki antarmuka yang mudah dimengerti	4.2		
3	Aplikasi permainan memiliki informasi dan instruksi yang mudah dipahami	4.3		
<b>Parameter Immersive</b>				
4	Aplikasi memiliki area permainan dengan tampilan menarik	4.3	4.2	84.4%

No	Pernyataan	Rata-rata	Total	Total (%)
5	Saya merasa tertarik ingin memainkan permainan ini kembali	4.3		
6	Saya merasa tertantang untuk menyelesaikan permainan	4.6		
7	Aplikasi permainan memiliki jebakan yang menantang	3.8		
8	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan	4.1		
<b>Parameter Controller</b>				
9	Aplikasi permainan memiliki kontrol yang mudah dimengerti	4.2	3.8	77%
10	Saya dapat berinteraksi dengan tuas kontrol dengan mudah	3.5		
<b>Parameter Kenyamanan</b>				
11	Aplikasi permainan nyaman ketika dimainkan	4.1	4.4	88%
12	Permainan dapat berjalan dengan lancar tanpa adanya <i>lag</i> dan <i>crash</i>	4.7		

### 5.3.5 Kritik dan saran pengguna

Dalam memberikan penilaian dan tanggapan, pengguna diberikan kuesioner pengujian permainan. Kuesioner pengujian permainan ini terdapat bagian kritik dan saran untuk perbaikan fitur

kedepannya. Kritik dan saran penggunaan dapat dilihat pada Tabel 5.10.

Tabel 5.10 Kritik dan Saran Pengguna

No.	Partisipan	Kritik dan Saran
1	Partisipan 1	Saat permainan selesai, dan muncul <i>pop up</i> , permainan tetap berjalan, sehingga bikin bingung. Tambahkan suara saat jebakan muncul.
2	Partisipan 2	Posisi kamera agak susah untuk melihat beberapa sudut tertentu.
3	Partisipan 3	Jebakan terlalu <i>shocking</i> di level akhir. Jebakan kalau bisa jangan tepat di area bola sekarang.
4	Partisipan 4	Panel <i>game over</i> kurang me- <i>notif user</i> kalau <i>game over</i> . Sudut / <i>viewpoint</i> kurang jelas, harus memiringkan <i>maze</i> untuk melihat keseluruhan <i>maze</i> .
5	Partisipan 5	Permainan menarik, tantangan kurang banyak.
6	Partisipan 6	<i>Trap</i> untuk bola jatuh kalau bisa <i>random</i> -nya diatur di 1 kotak sekeliling bola.
7	Partisipan 7	Tampilan <i>maze</i> terlalu jauh, sehingga beberapa memerlukan <i>effort</i> lebih agar bisa dilihat.
8	Partisipan 8	Tuas kontrol terkadang susah saat digunakan untuk interaksi. Jebakan yang ada kurang menantang meskipun sudah mencapai level atas, sebaiknya ditambah lagi jebakannya.

9	Partisipan 9	Selain kontrol kanan uang kadang – kadang sulit di kontrol, semua bagus. Menantang dan seru.
10	Partisipan 10	<i>Maze</i> terlalu dibawah jadi capek, <i>overall</i> bagus.
11	Partisipan 11	Saya rasa lebih mudah bila arah tuas dan pergerakan <i>maze</i> berlawanan agar tidak bingung.
12	Partisipan 12	-
13	Partisipan 13	Mantap.
14	Partisipan 14	Sebaiknya tampilan trampolin diganti, karena lebih mirip lubang daripada trampoline.
15	Partisipan 15	Bolanya bias diganti-ganti tidak hanya satu saja.
16	Partisipan 16	Tampilan disetiap level itu-itu saja, tambah tampilan yang berbeda disetiap level.

## 5.4 Evaluasi Pengujian

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

### 5.4.1 Evaluasi Pengujian Fungsionalitas

Evaluasi pengujian fungsionalitas dilakukan dengan menampilkan data rekapitulasi aplikasi yang telah dipaparkan pada subbab 5.2. Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Tabel 5.11. Dari data yang terdapat pada tabel tersebut, diketahui bahwa aplikasi yang dibuat telah memenuhi kasus penggunaan yang telah ditentukan.

Tabel 5.11 Rekapitulasi Hasil Uji Fungsionalitas

ID	Deskripsi	Skenario	Perilaku Terlaksana
UF-001	Uji Coba pada Menu Permainan	Skenario 1	Ya
		Skenario 2	Ya
		Skenario 3	Ya
UF-002	Uji Coba dalam Permainan	Skenario 1	Ya
		Skenario 2	Ya
		Skenario 3	Ya
		Skenario 4	Ya
		Skenario 5	Ya
		Skenario 6	Ya
		Skenario 7	Ya
		Skenario 8	Ya
		Skenario 9	Ya
		Skenario 10	Ya
		Skenario 11	Ya
UF-003	Uji Coba Akhir Permainan	Skenario 1	Ya
		Skenario 2	Ya

#### 5.4.2 Evaluasi Pengujian Pengguna

Evaluasi pengujian pengguna dapat dilihat pada Tabel 5.12. Hasil yang terlihat merupakan hasil rata-rata dari semua kuesioner.

Tabel 5.12 Rangkuman Hasil Kuesioner Pengujian Aplikasi

No.	Pernyataan	Nilai
1	Aplikasi permainan memiliki objek dengan desain, tampilan, dan warna yang menarik	4.5
2	Aplikasi permainan memiliki antarmuka yang mudah dimengerti	4.2

No.	Pernyataan	Nilai
3	Aplikasi permainan memiliki informasi dan instruksi yang mudah dipahami	4.3
4	Aplikasi memiliki area permainan dengan tampilan menarik	4.3
5	Aplikasi permainan nyaman ketika dimainkan	4.1
6	Aplikasi permainan memiliki kontrol yang mudah dimengerti	4.2
7	Saya dapat berinteraksi dengan tuas kontrol dengan mudah	3.5
8	Saya merasa tertantang untuk menyelesaikan permainan	4.6
9	Aplikasi permainan memiliki jebakan yang menantang	3.8
10	Saya merasa tertarik ingin memainkan permainan ini kembali	4.3
11	Permainan dapat berjalan dengan lancar tanpa adanya <i>lag</i> dan <i>crash</i>	4.7
12	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan	4.1

Berdasarkan Tabel 5.12, dapat diketahui nilai rata-rata adalah 4.2. Berdasarkan nilai maksimal 5, nilai ini sudah melebihi dari 4 yang memiliki arti “setuju” dengan pernyataan-pernyataan yang dibuat, sehingga dapat disimpulkan bahwa aplikasi permainan ini secara keseluruhan sudah cukup bagus, cukup nyaman, dan cukup mudah digunakan. Namun masih perlu adanya pengembangan karena aplikasi permainan ini masih belum dapat mencapai nilai tinggi.

*[halaman ini sengaja dikosongkan]*



## **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. Implementasi permainan *Marble Maze Game* pada permainan Trapball berhasil diterapkan didukung dengan konsep realitas virtual. Berdasarkan pada hasil kuesioner didapat nilai 84.4% (nilai 4,2 dari nilai maksimal 5) pada parameter *immersive*.
2. Implementasi kontrol permainan *Marble Maze Game* pada permainan Trapball berhasil diterapkan didukung dengan konsep realitas virtual. Berdasarkan pada hasil kuesioner didapat nilai 77% (nilai 3,8 dari nilai maksimal 5) pada parameter *controller*.
3. Implementasi permainan *Marble Maze Game* berhasil diterapkan menggunakan teknologi HTC Vive. Berdasarkan pada hasil kuesioner didapat nilai 88% (nilai 4,4 dari nilai maksimal 5) pada parameter kenyamanan.
4. Implementasi permainan *Marble Maze Game* berhasil diselesaikan menggunakan *game engine* Unity.

## 6.2. Saran

Berikut merupakan beberapa saran untuk perbaikan dan pengembangan aplikasi di masa akan datang:


1. Ditambahkan sebuah jebakan atau *trap* baru untuk menambah kesulitan dalam level permainan. Selain itu, setiap kali jebakan muncul ditambah efek suara agar mempermudah pemain.
2. Mengubah posisi pemain dan *maze* untuk menambah kenyamanan pemain ketika bermain permainan. Selain itu, ketika berada dalam halaman permainan ditambahkan layar kecil untuk menampilkan tampilan *maze* dari atas.
3. Mengubah posisi tuas kontrol agar pemain lebih nyaman dalam melakukan interaksi pada tuas.
4. Menambahkan *haptic feedback* pada Vive Controller pada saat pemain berhasil memegang tuas kontrol.
5. Mengubah posisi halaman interaksi lebih mendekat agar pemain dapat mengetahui setiap perpindahan halaman yang terjadi.
6. Menambah lebih banyak level permainan agar permainan lebih menantang bagi pemain.

## DAFTAR PUSTAKA

- [1] Sudono, Anggani, Sumber Belajar dan Alat Permainan, Jakarta: Grasindo, 2000.
- [2] “10 of the Best Marble Maze Toys for Developing Creativity and Logic,” [Daring]. Tersedia pada: <https://www.fractuslearning.com/marble-maze-toys-creativity-logic/>. [Diakses 8 November 2018].
- [3] Herumurti, D., Yuniarti, A., Kuswardayan, I., & Khotimah, W. N. (2018). The identification and observation of star constellations using virtual reality and Google card board. In Proceedings - 2017 International Seminar on Application for Technology of Information and Communication: Empowering Technology for a Better Human Life, iSemantic 2017 (Vol. 2018–January, pp. 187–190). Institute of Electrical and Electronics Engineers Inc.
- [4] “VIVE Virtual Reality System,” [Daring]. Tersedia pada: <https://www.vive.com/us/product/vive-virtual-reality-system/>. [Diakses 1 Desember 2018].
- [5] “Unity,” [Daring]. Tersedia pada: <https://docs.unity3d.com/Manual/index.html>. [Diakses 19 November 2018].
- [6] “Kelebihan Bahasa Pemograman C#,” [Daring]. Tersedia pada: <https://teknajurnal.com/kelebihan-bahasa-pemrograman-c-sharp/>. [Diakses 10 November 2018].
- [7] Herumurti, D., Yuniarti, A., Kuswardayan, I., Nurul, W., Hariadi, R. R., Suciati, N., & Manggala, M. G. (2017). Mixed reality in the 3D virtual room arrangement. In 2017 11th International Conference on Information Communication Technology and System (ICTS).
- [8] “VRTK – Virtual Reality Toolkit,” [Daring]. Tersedia pada: <https://vrtoolkit.readme.io/>. [Diakses 18 Juni 2019].

- [9] “Algoritma DFS (Depth First Search),” [Daring]. Tersedia pada: <https://piptools.net/algoritma-dfs-depth-first-search/>. [Diakses 16 Juni 2019]






**ITS**  
Institut Teknologi Sepuluh Nopember

**KUESIONER TUGAS AKHIR**  
05.1114000025 – Aluram dan/or Peninggi

**IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL MENGGUNAKAN TEKNOLOGI HTC VIVE**

Identitas Responden  
 Nama Lengkap : FINDRYAN K P  
 Pekerjaan : MAHASISWA  
 Usia : 22

Surabaya, Juni 2019  


**A. KARAKTERISTIK RESPONDEN**  
 Jawablah pertanyaan di bawah ini dengan menggunakan tanda silang (x)

- Apakah anda tahu tentang *Game Virtual Reality* ?  
 Ya  
 Tidak
- Apakah anda tahu tentang permainan *Marble Maze Game* ?  
 Ya  
 Tidak
- Apakah anda pernah memainkan game yang berbasis realitas virtual ?  
 Pernah  
 Tidak Pernah
- Apakah anda tahu tentang HTC Vive ?  
 Ya  
 Tidak
- Apakah anda pernah memainkan game yang berbasis realitas virtual menggunakan HTC Vive ?  
 Pernah  
 Tidak Pernah

**B. PENILAIAN TERHADAP PERMAINAN**  
 Isilah tabel dibawah ini dengan menggunakan tanda silang (x)

Keterangan  
 TS = Tidak Setuju      KS = Kurang Setuju      N = Netral  
 S = Setuju              SS = Sangat Setuju

Parameter Antarmuka		TS	KS	N	S	SS
1	Aplikasi permainan memiliki objek dengan desain, tampilan, dan warna yang menarik					✓
2	Aplikasi permainan memiliki antarmuka yang mudah dimengerti			✓		
3	Aplikasi permainan memiliki informasi dan petunjuk yang mudah dipahami				✓	
Parameter Immersivity		TS	KS	N	S	SS
4	Aplikasi memiliki area permainan dengan tampilan menarik					✓
5	Saya merasa tertarik ingin memainkan permainan ini kembali					✓
6	Saya merasa tertantang untuk menyelesaikan permainan					✓
7	Aplikasi permainan memiliki jobakan yang menantang			✓		
8	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan				✓	
Parameter Controller		TS	KS	N	S	SS
9	Aplikasi permainan memiliki kontrol yang mudah dimengerti					✓
10	Saya dapat berinteraksi dengan luas kontrol dengan mudah				✓	
Parameter Kenyamanan		TS	KS	N	S	SS
11	Aplikasi permainan nyaman ketika dimainkan					✓
12	Permainan dapat berjalan dengan lancar tanpa adanya lag dan crash					✓

**C. KRITIK DAN SARAN**  
 Dengan ini saya berkritik - saran  
 sebagai berikut

Gambar A. 2 Kuesioner Responden Findryan K P



**KUESIONER TUGAS AKHIR**  
05.1154000029—Andreas Inuar Pratomo

**IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL  
MENGUNAKAN TEKNOLOGI HTC VIVE**

Identitas Responden : Surabaya, 16 Juni 2019

Nama Lengkap : Abyan Dafa

Pekerjaan : Marketing

Usia : 30

*(Signature)*  
Abyan Dafa

**A. KARAKTERISTIK RESPONDEN**

Jawablah pertanyaan di bawah ini dengan menggunakan tanda silang (x)

1. Apakah anda tahu tentang *Game Virtual Reality* ?  
 Ya  Tidak
2. Apakah anda tahu tentang permainan *Marble Maze Game* ?  
 Ya  Tidak
3. Apakah anda pernah memainkan *game* yang berbasis realitas virtual ?  
 Pernah  Tidak Pernah
4. Apakah anda tahu tentang HTC Vive ?  
 Ya  Tidak
5. Apakah anda pernah memainkan *game* yang berbasis realitas virtual menggunakan HTC Vive ?  
 Pernah  Tidak Pernah

**B. PENILAIAN TERHADAP PERMAINAN**


Isilah tabel dibawah ini dengan menggunakan tanda silang (x)

Keterangan  
TS = Tidak Setuju  
S = Setuju  
KS = Kurang Setuju  
SS = Sangat Setuju  
N = Netral

	Parameter Antarmuka	TS	KS	N	S	SS
1	Apakah permainan memiliki obyek dengan desain, tampilan, dan warna yang menarik					<input checked="" type="checkbox"/>
2	Apakah permainan memiliki antarmuka yang mudah dimengerti					<input checked="" type="checkbox"/>
3	Apakah permainan memiliki informasi dan instruksi yang mudah dipahami					<input checked="" type="checkbox"/>
	Parameter Immersivity	TS	KS	N	S	SS
4	Apakah memiliki area permainan dengan tampilan menarik					<input checked="" type="checkbox"/>
5	Seberapa mudah melihat/memilihkan permainan ini kembali					<input checked="" type="checkbox"/>
6	Seberapa tertantang untuk menyelesaikan permainan					<input checked="" type="checkbox"/>
7	Apakah permainan memiliki jebakan yang menantang			<input checked="" type="checkbox"/>		
8	Seberapa tinggi level permainan, semakin sulit untuk menyelesaikan level permainan					<input checked="" type="checkbox"/>
	Parameter Controller	TS	KS	N	S	SS
9	Apakah permainan memiliki kontrol yang mudah dimengerti					<input checked="" type="checkbox"/>
10	Seberapa dapat berinteraksi dengan tuas kontrol dengan mudah					<input checked="" type="checkbox"/>
	Parameter Kenyamanan	TS	KS	N	S	SS
11	Apakah permainan nyaman ketika dimainkan					<input checked="" type="checkbox"/>
12	Permainan dapat berjalan dengan lancar tanpa adanya lag dan crash					<input checked="" type="checkbox"/>

C. KRITIK DAN SARAN  
 Saran: Tampilan yang diwart...  
 karena lebih baik...  
 yang...  
 ...

Gambar A. 3 Kuesioner Responden Abyan Dafa



**KUESIONER TUGAS AKHIR**  
08113-600032 - Aplikasi Game Realitas

Surabaya, 18 Juni 2019

*[Signature]*  
Huda F M

*[Signature]*  
Huda F M

**IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL MENGGUNAKAN TEKNOLOGI HTC VIVE**

**Identitas Responden**

Nama Lengkap : *Huda F M*

Pekerjaan : *Mahasiswa*

Usia : *22*

---

**A. KARAKTERISTIK RESPONDEN**  
*Jawaban pertanyaan di bawah ini dengan menggunakan tanda silang (x)*

- Apakah anda tahu tentang Game Virtual Reality ?  
 Ya  
 Tidak
- Apakah anda tahu tentang permainan Marble Maze Game ?  
 Ya  
 Tidak
- Apakah anda pernah memainkan game yang berbasis realitas virtual ?  
 Pernah  
 Tidak Pernah
- Apakah anda tahu tentang HTC Vive ?  
 Ya  
 Tidak
- Apakah anda pernah memainkan game yang berbasis realitas virtual menggunakan HTC Vive ?  
 Pernah  
 Tidak Pernah

**B. PENILAIAN TERHADAP PERMAINAN**  
*Isian tabel dibawah ini dengan menggunakan tanda silang (x)*

Keterangan  
 TS = Tidak Setuju      KS = Kurang Setuju      N = Netral  
 S = Setuju              SS = Sangat Setuju

Parameter Antarmuka		TS	KS	N	S	SS
1	Apliasi permainan memiliki objek dengan desain, tampilan, dan warna yang menarik					<input checked="" type="checkbox"/>
2	Apliasi permainan memiliki antarmuka yang mudah dimengerti					<input checked="" type="checkbox"/>
3	Apliasi permainan memiliki informasi dan feedback yang mudah dipahami				<input checked="" type="checkbox"/>	
Parameter Immersivity		TS	KS	N	S	SS
4	Apliasi memiliki area permainan dengan tampilan menarik					<input checked="" type="checkbox"/>
5	Saya merasa tertarik ingin memainkan permainan ini kembali					<input checked="" type="checkbox"/>
6	Saya merasa tertantang untuk menyelesaikan permainan					<input checked="" type="checkbox"/>
7	Apliasi permainan memiliki jobatan yang menantang					<input checked="" type="checkbox"/>
8	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan					<input checked="" type="checkbox"/>
Parameter Controller		TS	KS	N	S	SS
9	Apliasi permainan memiliki kontrol yang mudah dimengerti					<input checked="" type="checkbox"/>
10	Saya dapat berinteraksi dengan tuas kontrol dengan mudah					<input checked="" type="checkbox"/>
Parameter Kenyamanan		TS	KS	N	S	SS
11	Apliasi permainan nyaman ketika dimainkan					<input checked="" type="checkbox"/>
12	Permainan sangat nyaman dengan lancar tanpa adanya lag dan crash					<input checked="" type="checkbox"/>

**C. KRITIK DAN SARAN**  
*[Signature]*

Gambar A. 4 Kuesioner Responden Huda F M



IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL  
MENGGUNAKAN TEKNOLOGI HTC VIVE

Surabaya, 18 Juni 2019

Identitas Responden

Nama Lengkap : Muhammad Adib Armandu

Pekerjaan : Mahasiswa

Usia : 23

A. KARAKTERISTIK RESPONDEN

Jawablah pertanyaan di bawah ini dengan menggunakan tanda silang (x)

1. Apakah anda tahu tentang Game Virtual Reality ?  
 Ya  
 Tidak
2. Apakah anda tahu tentang permainan Marble Maze Game ?  
 Ya  
 Tidak
3. Apakah anda pernah memainkan game yang berbasis realitas virtual ?  
 Pernah  
 Tidak Pernah
4. Apakah anda tahu tentang HTC Vive ?  
 Ya  
 Tidak
5. Apakah anda pernah memainkan game yang berbasis realitas virtual menggunakan HTC Vive ?  
 Pernah  
 Tidak Pernah

B. PENILAIAN TERHADAP PERMAINAN

Isilah tabel dibawah ini dengan menggunakan tanda silang (x)

Keterangan

TS = Tidak Setuju

S = Sangat Setuju

KS = Kurang Setuju

N = Netral


SS = Sangat Setuju

	Parameter Antarmuka	TS	KS	N	S	SS
1	Aplikasi permainan memiliki objek dengan desain, tampilan, dan warna yang menarik					<input checked="" type="checkbox"/>
2	Aplikasi permainan memiliki antarmuka yang mudah dimengerti					<input checked="" type="checkbox"/>
3	Aplikasi permainan memiliki informasi dan instruksi yang mudah dipahami					<input checked="" type="checkbox"/>
	Parameter Immersivity	TS	KS	N	S	SS
4	Aplikasi memiliki area permainan dengan tampilan menarik					<input checked="" type="checkbox"/>
5	Saya merasa terdapat, tinggi memainkan permainan ini kembali					<input checked="" type="checkbox"/>
6	Saya merasa tertantang untuk menyelesaikan permainan					<input checked="" type="checkbox"/>
7	Aplikasi permainan memiliki jabatan yang menantang					<input checked="" type="checkbox"/>
8	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan	<input checked="" type="checkbox"/>				
	Parameter Controller	TS	KS	N	S	SS
9	Aplikasi permainan memiliki kontrol yang mudah dimengerti					<input checked="" type="checkbox"/>
10	Saya dapat berinteraksi dengan luas kontrol dengan mudah					<input checked="" type="checkbox"/>
	Parameter Kenyamanan	TS	KS	N	S	SS
11	Aplikasi permainan nyaman ketika dimainkan					<input checked="" type="checkbox"/>
12	Permainan dapat berjalan dengan lancar tanpa adanya lag dan crash					<input checked="" type="checkbox"/>

C. KRITIK DAN SABAN

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

Gambar A. 5 Kuesioner Responden Muhammad Adib A



**KUESIONER TUGAS AKHIR**  
 051124030019 - Alfindio Januar Pratomo

**IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL MENGGUNAKAN TEKNOLOGI HTC VIVE**

Surabaya, 16 Juni 2019

**Identitas Responden**

Nama Lengkap : Alfindio M. A.  
 Pekerjaan : Mahasiswa  
 Usia : 21 Tahun

Alfindio M. A.  
 051124030019

**A. KARAKTERISTIK RESPONDEN**  
 Jawaban pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda tahu tentang Game Virtual Reality ?  
 Ya  Tidak
- Apakah anda tahu tentang permainan Marble Maze Game ?  
 Ya  Tidak
- Apakah anda pernah memainkan game yang berbasis realitas virtual ?  
 Pernah  Tidak Pernah
- Apakah anda tahu tentang HTC Vive ?  
 Ya  Tidak
- Apakah anda pernah memainkan game yang berbasis realitas virtual menggunakan HTC Vive ?  
 Pernah  Tidak Pernah

**B. PENILAIAN TERHADAP PERMAINAN**  
 Isian tabel dibawah ini dengan menggunakan tanda silang (X)

Keterangan  
 TS = Tidak Setuju N = Netral  
 S = Setuju SS = Sangat Setuju

Parameter Antarmuka		TS	KS	N	S	SS
1	Aplikasi permainan memiliki objek dengan desain, tampilan, dan warna yang menarik				<input checked="" type="checkbox"/>	
2	Aplikasi permainan memiliki antarmuka yang mudah dimengerti				<input checked="" type="checkbox"/>	
3	Aplikasi permainan memiliki informasi dan petunjuk yang mudah dipahami				<input checked="" type="checkbox"/>	
Parameter Immersivity		TS	KS	N	S	SS
4	Aplikasi memiliki area permainan dengan tampilan menarik			<input checked="" type="checkbox"/>		
5	Saya merasa tertarik ingin memainkan permainan ini kembali				<input checked="" type="checkbox"/>	
6	Saya merasa tertantang untuk menyelesaikan permainan				<input checked="" type="checkbox"/>	
7	Aplikasi permainan memiliki jobatan yang menantang			<input checked="" type="checkbox"/>		
8	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan				<input checked="" type="checkbox"/>	
Parameter Controller		TS	KS	N	S	SS
9	Aplikasi permainan memiliki kontrol yang mudah dimengerti			<input checked="" type="checkbox"/>		
10	Saya dapat berinteraksi dengan luas kontrol dengan mudah				<input checked="" type="checkbox"/>	
Parameter Kenyamanan		TS	KS	N	S	SS
11	Aplikasi permainan nyaman ketika dimainkan			<input checked="" type="checkbox"/>		
12	Permainan dapat berjalan dengan lancar tanpa adanya lag dan crash				<input checked="" type="checkbox"/>	

**C. KRITIK DAN SARAN**

Saya rasa masih banyak yang harus diperbaiki agar tidak mengalami lag dan crash

Gambar A. 6 Kuesioner Responden Alfindio M A

**IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL**  
MENGUNAKAN TEKNOLOGI HTC VIVE

Surabaya, 14 Juni 2019

Identitas Responden

Nama Lengkap : Yuga Mitra Hakiki

Pekerjaan : Mahasiswa

Usia : 22 Tahun

**A. KARAKTERISTIK RESPONDEN**

Jawablah pertanyaan di bawah ini dengan menggunakan tanda silang (x)

- Apakah anda tahu tentang *Game Virtual Reality* ?  
 Ya  
 Tidak
- Apakah anda tahu tentang permainan *Marble Maze Game* ?  
 Ya  
 Tidak
- Apakah anda pernah memainkan *game* yang berbasis realitas virtual ?  
 Pernah  
 Tidak Pernah
- Apakah anda tahu tentang HTC Vive ?  
 Ya  
 Tidak
- Apakah anda pernah memainkan *game* yang berbasis realitas virtual menggunakan HTC Vive ?  
 Pernah  
 Tidak Pernah

**B. PENILAIAN TERHADAP PERMAINAN**

Isilah tabel dibawah ini dengan menggunakan tanda silang (x)

Keterangan

TS = Tidak Setuju

S = Kurang Setuju

KS = Sangat Kurang Setuju

N = Netral


SS = Sangat Setuju

Parameter Antarmuka		TS	KS	N	S	SS
1	Aplikasi permainan memiliki obyek dengan desain, tampilan, dan warna yang menarik					X
2	Aplikasi permainan memiliki antarmuka yang mudah dimengerti					X
3	Aplikasi permainan memiliki informasi dan instruksi yang mudah dipahami					X
Parameter Immersivity		TS	KS	N	S	SS
4	Aplikasi memiliki area permainan dengan tampilan menarik					X
5	Saya merasa terdapat <i>light</i> memisahkan permainan ini dengan permainan lainnya					X
6	Saya merasa terdorong untuk menyelesaikan permainan					X
7	Aplikasi permainan memiliki jobakan yang menantang					X
8	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan					X
Parameter Controller		TS	KS	N	S	SS
9	Aplikasi permainan memiliki kontrol yang mudah dimengerti					X
10	Saya dapat berinteraksi dengan luas kontrol dengan mudah					X
Parameter Kenyamanan		TS	KS	N	S	SS
11	Aplikasi permainan nyaman ketika dimainkan					X
12	Permainan dapat berjalan dengan lancar tanpa adanya lag dan crash					X

**C. KRITIK DAN SARAN**

- Saat permainan selesai, dan muncul *pop up* permainan, *close* ke bagian *setting* akan bingung
- Penempatan *score* saat *end* akan muncul

Gambar A. 7 Kuesioner Responden Yuga Mitra Hakiki



**ITS**  
Institut Teknologi Sepuluh Nopember

**KUESIONER TUGAS AKHIR**  
051154000025 - Alifien Zamir Pratomo

**IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL MENGGUNAKAN TEKNOLOGI HTC VIVE**

Identitas Responden : Alifien Zamir Pratomo Surabaya, 17 Juni 2019  
 Nama Lengkap : Alifien Zamir Pratomo  
 Pekerjaan : Manajemen Sistem Informasi  
 Usia : 20

**A. KARAKTERISTIK RESPONDEN**  
*Jawablah pertanyaan di bawah ini dengan menggunakan tanda silang (x)*

- Apakah anda tahu tentang *Game Virtual Reality* ?  
 Ya  
 Tidak
- Apakah anda tahu tentang permainan *Marble Maze Game* ?  
 Ya  
 Tidak
- Apakah anda pernah memainkan game yang berbasis realitas virtual ?  
 Pernah  
 Tidak Pernah
- Apakah anda tahu tentang *HTC Vive* ?  
 Ya  
 Tidak
- Apakah anda pernah memainkan game yang berbasis realitas virtual menggunakan *HTC Vive* ?  
 Pernah  
 Tidak Pernah

**B. PENJAJAN TERHADAP PERMAINAN**  
*(Isilah tabel dibawah ini dengan menggunakan tanda silang (x))*

Keinginan	TS	KS	N	S	SS
TS = Tidak Setuju					
KS = Kurang Setuju					
S = Netral					
SS = Sangat Setuju					

Parameter Anamnata	TS	KS	N	S	SS
1 Aplikasi permainan memiliki objek dengan desain, tampilan, dan warna yang menarik					X
2 Aplikasi permainan memiliki antarmuka yang mudah dimengerti					X
3 Aplikasi permainan memiliki informasi dan instruksi yang mudah dipahami					X
Parameter Immersivity	TS	KS	N	S	SS
4 Aplikasi memiliki area permainan dengan tampilan menarik					X
5 Saya merasa tertarik ingin memainkan permainan ini kembali					X
6 Saya merasa tertantang untuk menyelesaikan permainan					X
7 Aplikasi permainan memiliki jobban yang menantang					X
8 Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan					X
Parameter Controller	TS	KS	N	S	SS
9 Aplikasi permainan memiliki kontrol yang mudah dimengerti					X
10 Saya dapat berinteraksi dengan tugas kontrol dengan mudah					X
Parameter Kenyamanan	TS	KS	N	S	SS
11 Aplikasi permainan nyaman ketika dimainkan					X
12 Permainan dapat berjalan dengan lancar tanpa adanya lag dan crash					X

**C. KRITIK DAN SARAN**  
 Untuk meningkatkan daya tarik game ini, sebaiknya tampilan objek dan karakternya dibuat lebih menarik.

Gambar A. 8 Kuesioner Responden N M H Habib Yahya



**KUISIONER TUGAS AKHIR**  
051154000023 - Andrian Januar Pratomo

**IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL  
MENGUNAKAN TEKNOLOGI HTC VIVE**

Identitas Responden : Surabaya, Juni 2019  
 Nama Lengkap : BASKARA  
 Pekerjaan : MAHASISWA  
 Usia : 20

**A. KARAKTERISTIK RESPONDEN**

Jawablah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

1. Apakah anda tahu tentang *Game Virtual Reality* ?  
 Ya  Tidak
2. Apakah anda tahu tentang permainan *Marble Maze Game* ?  
 Ya  Tidak
3. Apakah anda pernah memainkan game yang berbasis realitas virtual ?  
 Pernah  Tidak Pernah
4. Apakah anda tahu tentang HTC Vive ?  
 Ya  Tidak
5. Apakah anda pernah memainkan game yang berbasis realitas virtual menggunakan HTC Vive ?  
 Pernah  Tidak Pernah

**B. PENILAIAN TERHADAP PERMAINAN**

Isilah tabel dibawah ini dengan menggunakan tanda silang (X)


Keterangan  
 TS = Tidak Setuju N = Netral  
 S = Setuju SS = Sangat Setuju

Parameter Antarmuka		TS	KS	N	S	SS
1	Aplikasi permainan memiliki objek dengan desain, tampilan, dan warna yang menarik					X
2	Aplikasi permainan memiliki antarmuka yang mudah dimengerti					X
3	Aplikasi permainan memiliki informasi dan instruksi yang mudah dipahami					X
Parameter Immersivity		TS	KS	N	S	SS
4	Aplikasi memiliki area permainan dengan tampilan menarik					X
5	Saya merasa terstimulasi/memilihkan permainan ini kembali			X		
6	Saya merasa tertantang untuk menyelesaikan permainan					
7	Aplikasi permainan memiliki jabatan yang menantang				X	
8	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan					X
Parameter Controller		TS	KS	N	S	SS
9	Aplikasi permainan memiliki kontrol yang mudah dimengerti					X
10	Saya dapat berinteraksi dengan luas kontrol dengan mudah				X	
Parameter Kenyamanan		TS	KS	N	S	SS
11	Aplikasi permainan nyaman ketika dimainkan					X
12	Permainan dapat berjalan dengan lancar tanpa adanya lag dan crash					X

**C. KRITIK DAN SARAN**

Apakah ada feedback di awal - awal akan  
 diberikan ketika sudah selesai game, atau  
 jika ada feedback maka bisa langsung di  
 kirim ke admin

Gambar A. 9 Kuesioner Responden Baskara



**ITS**  
Institut Teknologi Sepuluh Nopember

**KUESIONER TUGAS AKHIR**  
0511140000035 – Akademi Sains dan Teknologi

**IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL MENGGUNAKAN TEKNOLOGI HTC VIVE**

Surabaya, 17 Juni 2019

Identitas Responden : Ronald Andron  
 Nama Lengkap : Ronald Andron  
 Pekerjaan : Programmer  
 Usia : 22

---

**A. KARAKTERISTIK RESPONDEN**  
*Jawablah pertanyaan di bawah ini dengan menggunakan tanda silang (x)*

- Apakah anda tahu tentang permainan *Marble Maze Reality* ?  
 Ya  
 Tidak
- Apakah anda tahu tentang permainan *Marble Maze Game* ?  
 Ya  
 Tidak
- Apakah anda pernah memainkan game yang berbasis realitas virtual ?  
 Pernah  
 Tidak Pernah
- Apakah anda tahu tentang HTC Vive ?  
 Ya  
 Tidak
- Apakah anda pernah memainkan game yang berbasis realitas virtual menggunakan HTC Vive ?  
 Pernah  
 Tidak Pernah

**B. PENILAIAN TERHADAP PERMAINAN**  
*Silahkan tanda di bawah ini dengan menggunakan tanda silang (x)*

Keterangan :  
 TS = Tidak Setuju    KS = Kurang Setuju    N = Netral  
 S = Setuju    SS = Sangat Setuju

Parameter Antarmuka		TS	KS	N	S	SS
1	Aplikasi permainan memiliki obyek dengan desain, tampilan, dan warna yang menarik				X	
2	Aplikasi permainan memiliki antarmuka yang mudah dimengerti			X		
3	Aplikasi permainan memiliki informasi dan feedback yang mudah dipahami				X	
Parameter Immersivity		TS	KS	N	S	SS
4	Aplikasi memiliki area permainan dengan tampilan menarik				X	
5	Saya merasa tertarik ingin memainkan permainan ini kembali				X	
6	Saya merasa tertantang untuk menyelesaikan permainan				X	
7	Aplikasi permainan memiliki jobkahan yang menantang				X	
8	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan					X
Parameter Controller		TS	KS	N	S	SS
9	Aplikasi permainan memiliki kontrol yang mudah dimengerti				X	
10	Saya dapat berinteraksi dengan tuas kontrol dengan mudah		X			
Parameter Kenyamanan		TS	KS	N	S	SS
11	Aplikasi permainan nyaman ketika dimainkan				X	
12	Permainan dapat berjalan dengan lancar tanpa adanya lag dan crash				X	

**C. KRITIK DAN SARAN**  
 Untuk game ini, kurang menarik bila game over. Model permainan kurang jelas, perlu penambahan banyak variasi. [www.kelompok.com](http://www.kelompok.com)

Gambar A. 10 Kuesioner Responden Ronald Andron





**KUESIONER TUGAS AKHIR**  
051174000025 - Analisis Usman Pratiwi

**IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL  
MENGUNAKAN TEKNOLOGI HTC VIVE**

Surabaya, 17 Juni 2019

Identitas Responden  
Nama Lengkap : Roko Mahid ALFI  
Pekerjaan : Freelance Wk  
Usia : 23

*Roko Mahid ALFI*  
Roko Mahid ALFI

- A. KARAKTERISTIK RESPONDEN**  
Jawaban pertanyaan di bawah ini dengan menggunakan tanda silang (x)
1. Apakah anda tahu tentang *Game Virtual Reality*?  
 Ya  
 Tidak
  2. Apakah anda tahu tentang permainan *Marble Maze Game*?  
 Ya  
 Tidak
  3. Apakah anda pernah memainkan *game* yang berbasis realitas virtual?  
 Pernah  
 Tidak Pernah
  4. Apakah anda tahu tentang HTC Vive?  
 Ya  
 Tidak
  5. Apakah anda pernah memainkan *game* yang berbasis realitas virtual menggunakan HTC Vive?  
 Pernah  
 Tidak Pernah

**B. PENILAIAN TERHADAP PERMAINAN**  
Isilah tabel dibawah ini dengan menggunakan tanda silang (x)

Skoranggot  
TS = Tidak Setuju KS = Kurang Setuju N = Netral  
S = Setuju SS = Sangat Setuju

Parameter Antarmuka		TS	KS	N	S	SS
1	Aplikasi permainan memiliki objek dengan desain, tampilan, dan warna yang menarik					X
2	Aplikasi permainan memiliki antarmuka yang mudah dimengerti					X
3	Aplikasi permainan memiliki informasi dan instruksi yang mudah dipahami					X
Parameter Immersivity		TS	KS	N	S	SS
4	Aplikasi memiliki area permainan dengan tampilan menarik					X
5	Saya merasa tertarik ingin memainkan permainan ini kembali					X
6	Saya merasa tertantang untuk menyelesaikan permainan					X
7	Aplikasi permainan memiliki jejak yang menantang					X
8	Semua tantangan level permainan, semakin sulit untuk menyelesaikan level permainan					X
Parameter Controller		TS	KS	N	S	SS
9	Aplikasi permainan memiliki kontrol yang mudah dimengerti					X
10	Saya dapat berinteraksi dengan luas kontrol dengan mudah					X
Parameter Kenyamanan		TS	KS	N	S	SS
11	Aplikasi permainan nyaman ketika dimainkan					X
12	Permainan dapat berjalan dengan lancar tanpa adanya lag dan crash					X

**C. KRITIK DAN SARAN**  
 Kritik yang akan saya sampaikan disini di:  
 1. Desain game yang menarik  
 2. Desain game yang menarik  
 3. Desain game yang menarik  
 4. Desain game yang menarik  
 5. Desain game yang menarik  
 6. Desain game yang menarik  
 7. Desain game yang menarik  
 8. Desain game yang menarik  
 9. Desain game yang menarik  
 10. Desain game yang menarik  
 11. Desain game yang menarik  
 12. Desain game yang menarik

Gambar A. 12 Kuesioner Responden Rogo Jagad Alit







**IMPLEMENTASI PERMAINAN MARBLE MAZE GAME BERBASIS REALITAS VIRTUAL  
MENGUNAKAN TEKNOLOGI HTC VIVE**

Identitas Responden : Hero Akbar Amwadi Surabaya, 17 Juni 2019  
 Nama Lengkap : Hero Akbar Amwadi  
 Pekerjaan : Mobostwa  
 Usia : 21

**A. KARAKTERISTIK RESPONDEN**

Jawablah pertanyaan di bawah ini dengan menggunakan tanda silang (x)

1. Apakah anda tahu tentang Game Virtual Reality ?  
 Ya  Tidak
2. Apakah anda tahu tentang permainan Marble Maze Game ?  
 Ya  Tidak
3. Apakah anda pernah memainkan game yang berbasis realitas virtual ?  
 Pernah  Tidak Pernah
4. Apakah anda tahu tentang HTC Vive ?  
 Ya  Tidak
5. Apakah anda pernah memainkan game yang berbasis realitas virtual menggunakan HTC Vive ?  
 Pernah  Tidak Pernah

**B. PENILAIAN TERHADAP PERMAINAN**

Isilah tabel dibawah ini dengan menggunakan tanda silang (x)

Keterangan:  
 TS = Tidak Setuju KS = Kurang Setuju N = Netral  
 S = Setuju SS = Sangat Setuju

Parameter Antarmuka		TS	KS	N	S	SS
1	Aplikasi permainan memiliki obyek dengan desain, tampilan, dan warna yang menarik				<input checked="" type="checkbox"/>	
2	Aplikasi permainan memiliki antarmuka yang mudah dimengerti					<input checked="" type="checkbox"/>
3	Aplikasi permainan memiliki informasi dan instruksi yang mudah dipahami					<input checked="" type="checkbox"/>
Parameter Immersibility		TS	KS	N	S	SS
4	Aplikasi memiliki area permainan dengan tampilan menarik				<input checked="" type="checkbox"/>	
5	Saya merasa tertarik ingin memainkan permainan ini kembali					<input checked="" type="checkbox"/>
6	Saya merasa tertantang untuk menyelesaikan permainan					<input checked="" type="checkbox"/>
7	Aplikasi permainan memiliki jabatan yang menantang					<input checked="" type="checkbox"/>
8	Semakin tinggi level permainan, semakin sulit untuk menyelesaikan level permainan				<input checked="" type="checkbox"/>	
Parameter Controller		TS	KS	N	S	SS
9	Aplikasi permainan memiliki kontrol yang mudah dimengerti					<input checked="" type="checkbox"/>
10	Saya dapat berinteraksi dengan bus kontrol dengan mudah			<input checked="" type="checkbox"/>		
Parameter Kenyamanan		TS	KS	N	S	SS
11	Aplikasi permainan nyaman ketika dimainkan					<input checked="" type="checkbox"/>
12	Permainan dapat berjalan dengan lancar tanpa adanya lag dan crash					<input checked="" type="checkbox"/>

**C. KRITIK DAN SARAN** Selain kontrol, elemen yang paling banyak mendapat kritik di kontrol, terutama bagian mekanis dan game.

Gambar A. 15 Kuesioner Responden Hero Akbar A



**B. FOTO PENGUJIAN**

*Gambar B. 1 Foto Pengujian Responden Abyan Dafa*



*Gambar B. 2 Foto Pengujian Responden M Adib Arinanda*



*Gambar B. 3 Foto Pengujian Responden Baskara*





*Gambar B. 4 Foto Pengujian Responden Yuga Mitra H*





*Gambar B. 5 Foto Pengujian Responden Alfindio M A*



*Gambar B. 6 Foto Pengujian Responden Hidayatul M*



*Gambar B. 7 Foto Pengujian Responden Hero Akbar A*



*Gambar B. 8 Foto Pengujian Responden GD Wahyu*



*Gambar B. 9 Foto Pengujian Responden Subhan Maulana*

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Penulis lahir di Surabaya pada tanggal 29 Januari 1997. Dalam perjalanan hidupnya penulis pernah menempuh pendidikan tingkat dasar di SDN Kauman 1, dilanjutkan Pendidikan tingkat menengah SMPN 1 Srengat, Pendidikan tingkat atas SMAN 1 Srengat dan terakhir sedang menempuh Pendidikan di perguruan tinggi negeri Institut Teknologi Sepuluh Nopember Surabaya (ITS Surabaya) jurusan Departemen Infomatika. Selama menjadi mahasiswa penulis pernah mengikuti pelatihan pengembangan diri mahasiswa dari tingkat Pra-dasar, Dasar, dan beberapa pelatihan lainnya. Penulis juga pernah menjadi bagian dalam salah satu Big Event yaitu Schematics ITS 2015 dan 2016. Dalam bidang organisasi, penulis pernah bergabung menjadi pengurus Himpunan Mahasiswa Teknik Computer-Informatika (HMTIC) ITS 2015/2016 dan 2016/2017 serta menjadi Wakil Koordinator 3 Web dan Kesekretariatan Schematics 2015 dan Wakil Koordinator 1 Web dan Kesekretariatan Schematics 2016. Selain itu, penulis juga pernah tergabung menjadi pengurus Badan Eksekutif Mahasiswa Fakultas (BEMF) FTIK 2015/2016. Penulis dapat dihubungi melalui email [andreanjanuarpriatmojo@gmail.com](mailto:andreanjanuarpriatmojo@gmail.com)