



TUGAS AKHIR - KI1502

PEMBANGKIT LABIRIN DINAMIS PADA GAME BERHITUNG BERBASIS ANDROID DENGAN SENSOR *ACCELEROMETER*

SARIANTI PUNDI ABDYANTARY
NRP 5111100114

Dosen Pembimbing I
Imam Kuswardayan, S.Kom., M.T.

Dosen Pembimbing II
Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT- KI1502
DYNAMIC LABYRINTH GENERATOR FOR
ARITHMETIC GAME BASED ON ANDROID
DEVICE WITH *ACCELEROMETER* SENSOR

SARIANTI PUNDI ABDYANTARY
NRP 5111100114

Supervisor I
Imam Kuswardayan, S.Kom., M.T.

Supervisor II
Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2015

LEMBAR PENGESAHAN

PEMBANGKIT LABIRIN DINAMIS PADA GAME BERHITUNG BERBASIS ANDROID DENGAN SENSOR ACCELEROMETER

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Interaksi, Grafika dan Seni
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

SARIANTI PUNDI ABDYANTARY
NRP : 5111 100 114

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Imam Kuswardayan, S.Kom. (Pembimbing 1)
NIP: 197612152003121001
2. Dr.Eng. Nanik Suciati, S.Kom., M.Kom. (Pembimbing 2)
NIP: 197104281994122001



SURABAYA
JUNI, 2015

PEMBANGKIT LABIRIN DINAMIS PADA GAME BERHITUNG BERBASIS ANDROID DENGAN SENSOR ACCELEROMETER

Nama Mahasiswa : SARIANTI PUNDI ABDYANTARY
NRP : 5111100114
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Imam Kuswardayan, S.Kom., M.T.
Dosen Pembimbing 2 : Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

ABSTRAK

Dewasa ini, perkembangan teknologi sangat meningkat, khususnya pada teknologi smartphome. Begitu banyak aplikasi pada smartphome yang dapat dengan mudah diakses dan dimanfaatkan dalam kehidupan sehari-hari. Salah satunya adalah pendukung dalam bidang edukasi, selain itu pengguna aplikasi smartphome telah menjangkau segala usia termasuk remaja dan anak-anak sehingga memunculkan aplikasi permainan yang hanya bertujuan untuk menarik pengguna serta mengesampingkan unsur pembelajaran yang bermanfaat.

Game yang dibangun merupakan penggabungan metode pembangkit labirin dinamis hasil pengembangan Algoritma Prim's yang bertujuan untuk mengatur bentuk labirin yang berbeda disetiap level yang dipilih, sensor accelerometer untuk bantuan control permainan yang terdapat pada smartphome android serta berfokus pada pembelajaran berhitung. Dimensi labirin merupakan salah satu faktor tantangan dalam game ini, sehingga semakin tinggi level yang dipilih semakin besar pula dimensi labirin yang dibangun namun tetap diperhitungkan dengan faktor tantangan yang lain seperti soal, jawaban dan musuh.

Berdasarkan pengujian yang dilakukan, game mampu memberikan perancangan aturan main, membangkitkan labirin dinamis dengan susunan yang beragam serta dimensi labirin mampu menyesuaikan dengan perhitungan bobot faktor tantangan

pada setiap level yang tersedia sehingga, game yang dibangun pada tugas akhir ini mampu menjadi permainan yang menarik, tidak hanya berfokus memberikan hiburan bagi pemain namun juga memberikan unsur pembelajaran terutama pembelajaran berhitung.

Kata kunci: Smartphone, Algoritma Prims, Pembangkit Labirin Dinamis, Sensor Accelerometer.

**DYNAMIC LABYRINTH GENERATOR FOR
ARITHMETIC GAME BASED ON ANDROID DEVICE
WITH ACCELEROMETER SENSOR**

Student Name : Sarianti Pundi Abdyantary
NRP : 5111 100 114
Major : Teknik Informatika FTIf-ITS
Advisor I : Imam Kuswardayan, S.Kom., M.T.
Advisor II : Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

ABSTRACT

Today, technological developments highly increased, especially in smartphone technology. So many applications on a smartphone that can be easily accessed and used in daily life. One of which is support in the educating, other than the smartphone application users have been reach all ages included teens and children that led to game applications which is intended to attract users and rule out elements of educating.

This game is the combination of dynamic labyrinth generator method results from Prim's algorithm development which aims to set up a maze of different designs in each level is selected, accelerometer sensor for support game control and games that focus on learning math. Dimensional maze is one of the challenge factors in this game, so the higher level is selected the bigger dimensional maze was built but still calculated by other challenge factors such as questions, answers and enemies.

Based on testing performed, the game is able to provide designing rules of the game, generate dynamic labyrinth with a variety of designs and the dimensions of labyrinth is able to adjust the weight of calculate challenge factors at every level is available so that, games was built in this final project is able to be an interesting game, not focused only provide entertainment for the players but also give an element of learning, especially learning math.

Keywords: Education Game, Prim's Algorithm, Dynamic Level Generator, Accelerometer Sensor

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur, kehadiran Allah Subhanahu wa ta'ala yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “PEMBANGKIT LABIRIN DINAMIS PADA GAME BERHITUNG BERBASIS ANDROID DENGAN SENSOR *ACCELEROMETER*”.

Dalam pelaksanaan dan pembuatan tugas akhir ini tentunya sangat banyak bantuan-bantuan yang penulis terima dari berbagai pihak. Melalui lembar ini, penulis ingin secara khusus menyampaikan ucapan terima kasih kepada:

1. Allah SWT atas limpahan rahmat dan rezeki-Nya sehingga penulis dapat menyelesaikan Tugas Akhir.
2. Ayah penulis, Ir Sarwono, MMT dan Ibu penulis Sugiyati yang selalu memberikan dukungan, doa, perhatian, dan kasih sayang.
3. Kakak penulis, Mbak Arti, Mas Bima, Mbak Wening, Mas Joko, Mbak Liya, Mas Zaenal yang selalu memberikan motivasi dan bimbingan selama menyelesaikan studi penulis.
4. Keponakan penulis, Wijdan, Danish, Donita dan seluruh keluarga dekat yang telah membantu dan senantiasa memberi semangat untuk menyelesaikan studi penulis.
5. Bapak Imam Kuswardayan selaku dosen pembimbing Tugas Akhir pertama dan yang telah memberikan arahan dalam pengerjaan Tugas Akhir ini.
6. Ibu Nanik Suciati selaku dosen pembimbing Tugas Akhir kedua yang dengan sabar membimbing penulis dalam pengerjaan Tugas Akhir ini.
7. Bapak Raden Vinantius Hari Ginardi, selaku dosen wali yang berkenan memberi saran dan arahan selama penulis menjalani studi S1.
8. Dosen-dosen Teknik Informatika yang dengan sabar mendidik dan memberikan pengalaman baru kepada penulis

selama di Teknik Informatika dan Staf TU Teknik Informatika ITS yang senantiasa memudahkan segala urusan penulis di jurusan.

9. Rekan-rekan dan sahabat-sahabat penulis angkatan 2011 terutama kabinet bersahabat HMTC 2013-2014, Administrator LabProg 2009, 2010, 2011, 2012 dan 2013, Arum, Ruri, Septy, Tini, Fitri yang memberikan dorongan motivasi dan bantuan kepada penulis.
10. Andrie Prasetyo Utomo yang telah memberikan semangat, pencerahan, dukungan, dan motivasi kepada penulis
11. Kakak-kakak pengader C19 dan C1A, terutama kabinet perjuangan 2011-2012 dan kabinet bersatu 2012-2013
12. Pihak-pihak lain yang tidak sengaja terlewat dan tidak dapat penulis sebutkan satu per satu.

Penulis telah berusaha sebaik mungkin dalam menyusun tugas akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2015
Penulis

Sarianti Pundi Abdyantary

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi	3
1.7 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Sistem Operasi Android	7
2.2 Unity.....	7
2.3 Classic Labyrinth.....	8
2.4 Sensor Accelerometer	9
2.5 Algoritma Prims	9
BAB III ANALISIS DAN PERANCANGAN.....	13
3.1 Analisis Sistem	13
3.2 Perancangan Sistem.....	14
3.2.1 Deskripsi Umum Perangkat Lunak.....	14
3.2.2 Karakteristik Pengguna.....	15
3.2.3 Perancangan Pembangkit Labirin Dinamis.....	16
3.2.4 Perancangan Diagram Kasus Penggunaan.....	21
3.2.5 Perancangan Alur Permainan.....	32
3.2.6 Perancangan Penyimpanan Data.....	37
3.2.7 Perancangan Antarmuka Pengguna	37
BAB IV IMPLEMENTASI.....	41
4.1 Lingkungan Implementasi	41

4.2	Implementasi Permainan	41
4.2.1	Implementasi Pembuatan Pembangkit Labirin	42
4.2.2	Implementasi Membuka Level	43
4.2.3	Implementasi Pembuatan Soal dan Jawaban	44
4.2.4	Implementasi Tampilan Nilai Tertinggi.....	44
4.2.5	Implementasi Menembak Jawaban dan Musuh	45
4.2.6	Implementasi Menggerakkan Karakter	47
4.2.7	Implementasi Alur Permainan	49
4.2.8	Implementasi Penyimpanan Data	51
4.2.9	Tampilan Antarmuka Pengguna.....	53
BAB V PENGUJIAN DAN EVALUASI		57
5.1	Lingkungan Uji Coba	57
5.2	Pengujian Aturan Main.....	57
5.2.1	Skenario Uji Aturan Main.....	57
5.2.2	Hasil Pengujian Aturan Main.....	58
5.3	Pengujian Fungsionalitas dengan metode <i>black-box</i>	64
5.3.1	Skenario Pengujian Fungsionalitas	64
5.4	Pengujian Pengguna	71
5.4.1	Skenario Uji Coba Pengguna	71
5.4.2	Daftar Penguji Perangkat Lunak	71
5.4.3	Hasil Uji Coba Pengguna.....	72
BAB VI KESIMPULAN DAN SARAN.....		75
6.1.	Kesimpulan	75
6.2.	Saran	76
DAFTAR PUSTAKA.....		77
LAMPIRAN HASIL KUESIONER.....		79
BIODATA PENULIS.....		85

DAFTAR TABEL

Tabel 3.1 Karakteristik Pengguna	15
Tabel 3.2 Skenario Kasus Penggunaan	22
Tabel 3.3 Skenario Kasus Penggunaan Membuka Level	23
Tabel 3.4 Skenario Kasus Penggunaan Melihat Soal	25
Tabel 3.5 Skenario Kasus Penggunaan Melihat Nilai Tertinggi ..	26
Tabel 3.6 Skenario Kasus Penggunaan Menembak Jawaban dan Musuh.....	29
Tabel 3.7 Skenario Kasus Penggunaan Menggerakkan Karakter Utama dengan Sensor Accelerometer	31
Tabel 3.8 Detil Perancangan Bobot Faktor	33
Tabel 3.9 Detil Perancangan Bobot Level.....	34
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak (bagian 1)	41
Tabel 4.2 Lingkungan Implementasi Perangkat Lunak (bagian 2)	41
Tabel 5.1 Lingkungan Ujicoba Perangkat Lunak (bagian 2).....	57
Tabel 5.2 Skenario Pengujian Aturan Main	58
Tabel 5.3 Bobot Faktor Tantangan Percobaan I.....	59
Tabel 5.4 Pengujian Permainan.....	64
Tabel 5.5 Hasil Pengujian Fungsionalitas	70
Tabel 5.6 Daftar Nama Penguji Coba Aplikasi	72
Tabel 5.7 Penilaian Permainan (Pengujian I).....	72

DAFTAR GAMBAR

Gambar 2.1 Classic Labyrinth.....	8
Gambar 2.2 Accelerometer pada Android.....	9
Gambar 2.3 Pembobotan Graf dan Implementasi Algoritma Prims	11
Gambar 2.4 <i>Pseudocode</i> Algoritma Prims	12
Gambar 2.5 Algoritma Prims	12
Gambar 3.1 Diagram Alir Pembangkit Labirin Dinamis	17
Gambar 3.2 Perancangan Pembangkit Labirin Dinamis	20
Gambar 3.3 Diagram kasus aplikasi.....	21
Gambar 3.4 Diagram aktivitas Membuka Level	24
Gambar 3.5 Diagram aktivitas Melihat Soal	25
Gambar 3.6 Diagram aktivitas Melihat Nilai Tertinggi	26
Gambar 3.7 Diagram aktivitas Menembak Musuh.....	27
Gambar 3.8 Diagram aktivitas Menembak Jawaban.....	28
Gambar 3.9 Diagram Aktivitas Menggerakkan Karakter Utama	30
Gambar 3.10 Grafik Bobot Level.....	35
Gambar 3.11 State diagram permainan	36
Gambar 3.12 Tampilan Menu	37
Gambar 3.13 Tampilan Level dan Nilai Tertinggi	38
Gambar 3.14 Tampilan Menu How To	39
Gambar 3.15 Tampilan Menu Permainan	39
Gambar 3.16 Tampilan Kontrol Permainan <i>Virtual Joystick</i>	40
Gambar 4.1 <i>Pseudocode</i> Pembangkit Labirin Dinamis	43
Gambar 4.2 <i>Pseudocode</i> Membuka Level.....	44
Gambar 4.3 <i>Pseudocode</i> Pembuatan Soal dan Jawaban	44
Gambar 4.4 <i>Pseudocode</i> Menampilkan Nilai Tertinggi.....	45
Gambar 4.5 <i>Pseudocode</i> Mengeluarkan Musuh.....	46
Gambar 4.6 <i>Pseudocode</i> Menembak Musuh.....	46
Gambar 4.7 <i>Pseudocode</i> Menembak Jawaban atau menjawab Soal	47
Gambar 4.8 <i>Pseudocode</i> Menggerakkan Karakter	49
Gambar 4.9 <i>Pseudocode</i> Pembobotan Level.....	49
Gambar 4.10 <i>Pseudocode</i> Pembobotan Faktor perLevel	50

Gambar 4.11 <i>Pseudocode</i> Implementasi Permainan Berakhir	51
Gambar 4.12 Potongan Kode Sumber Penyimpanan Nilai Tertinggi di Setiap level	52
Gambar 4.13 Potongan Kode Sumber Pengambilan Data.....	52
Gambar 4.14 Implementasi Tampilan Menu Utama	53
Gambar 4.15 Implementasi Tampilan Level dan Nilai Tertinggi	54
Gambar 4.16 Implementasi Tampilan How-To.....	55
Gambar 4.17 Implementasi Tampilan Halaman Permainan	55
Gambar 4.18 Implementasi Tampilan Kontrol Permainan.....	56
Gambar 5.1 Tampilan Level 1 (Percobaan 1).....	61
Gambar 5.2 Tampilan Level 1 (Percobaan 2).....	61
Gambar 5.3 Tampilan Level 1 (Percobaan 3).....	62
Gambar 5.4 Karakter Utama Kondisi Kalah	62
Gambar 5.5 Karakter Utama Memenangkan Permainan	63
Gambar 5.6 Level Selanjutnya Terbuka	63
Gambar 5.7 Tampilan Menu Utama	65
Gambar 5.8 Tampilan layar Pemilihan Level.....	65
Gambar 5.9 Tampilan Halaman Permainan	66
Gambar 5.10 Tampilan Layar 'How To'	66
Gambar 5.11 Tampilan Menu Keluar Permainan.....	67
Gambar 5.12 Tampilan Menu Permainan.....	68
Gambar 5.13 Tampilan berkas penyimpanan data	69
Gambar 5.14 Tampilan Isi Penyimpanan Data.....	70
Gambar 9.1 Kuesioner Responden 1	79
Gambar 9.2 Kuesioner Responden 2	80
Gambar 9.3 Kuesioner Responden 3	81
Gambar 9.4 Kuesioner Responden 4	82
Gambar 9.5 Kuesioner Responden 5	83

BAB I PENDAHULUAN

Bab ini memaparkan garis besar Tugas Akhir yang meliputi latar belakang, tujuan dan manfaat pembuatan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1 Latar Belakang

Smartphone atau ponsel cerdas merupakan hal yang umum dalam kehidupan masyarakat saat ini. Berbagai macam fitur serta kelebihan pada berbagai perangkat keras canggih ini secara langsung meningkatkan keinginan bagi setiap orang untuk memilikinya. Perangkat keras ini didukung dengan aplikasi yang menunjang hampir seluruh kebutuhan pengguna, tidak hanya kebutuhan utama seperti penunjang berkomunikasi, perangkat cerdas ini mampu memberikan kebutuhan sekunder seperti hiburan musik, komik, video, film, hingga permainan.

Bermain *game* dengan *smartphone* dirasa mampu memberikan kepuasan tersendiri, karena mampu dilakukan dimana saja dan kapan saja. Penggunaan *smartphone* yang cukup mudah, maka dipastikan kategori pengguna perangkat canggih ini tersebar dalam segala umur. Peminat *game* yang cenderung cukup banyak, memunculkan ide bagi sebagian orang untuk memanfaatkan kesempatan dalam mendapatkan keuntungan materi seperti menciptakan aplikasi yang bersifat adiktif dan hanya bertujuan untuk materi. Sehingga, banyak aplikasi permainan dalam perangkat canggih ini mengesampingkan unsur pembelajaran yang bermanfaat, sedangkan sebagian besar pengguna *smartphone* adalah usia remaja dan pelajar.

Bermain *game* adalah sebuah aktivitas bermain yang dilakukan pada lingkungan digital dan dimainkan minimal oleh satu orang pemain, mempunyai aturan, kondisi menang serta kalah, serta konsep penunjang. Setiap *game* memiliki konsep dengan latar belakang yang bermacam-macam untuk menarik minat bermain

terhadap pengguna seperti konsep latar belakang jalan raya dengan contoh permainan pada *Crossy Road*, konsep dengan latar belakang luar angkasa dengan contoh permainan seperti *Space Cadet Defender Invaders*, konsep dengan latar belakang labirin seperti *Quick Maze 2D* dan masih banyak lagi konsep yang digunakan dalam *game*.

Pada *game* yang akan dirancang pada Tugas Akhir ini, merupakan *game* yang memiliki konsep dengan latar belakang labirin, posisi labirin pada tiap level serta posisi labirin pada level yang sama haruslah berbeda namun harus tetap menjaga keseimbangan letak antara jalan dan dinding, sehingga permainan tetap memungkinkan untuk dimenangkan oleh pemain serta membuat pemain merasa tertarik.

Tugas Akhir ini bertujuan untuk membuat suatu permainan yang tidak hanya memberikan hiburan serta ketertarikan bagi pemain namun juga memberikan unsur pembelajaran terutama pembelajaran berhitung. Permainan ini akan dijalankan pada system operasi Android, serta memanfaatkan sensor *accelerometer* sebagai alat bantu kontrol dalam bermain.

1.2 Rumusan Masalah

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

1. Bagaimana membuat aturan bermain *game* berhitung dengan menggunakan pembangkit labirin dinamis?
2. Bagaimana menggunakan Algoritma Prims dalam membangkitkan labirin yang dinamis namun masih memungkinkan untuk dimenangkan oleh pemain?
3. Bagaimana memanfaatkan sensor *Accelerometer* yang telah disediakan perangkat keras Android untuk bantuan kontrol pada permainan?

1.3 Batasan Masalah

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

1. Aplikasi ini akan dijalankan pada *smartphone* berbasis Android dengan versi minimum 2.2

2. Aplikasi ini menggunakan sensor *Accelerometer*.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah merancang dan membangun pembangkit labirin dinamis untuk *game* belajar berhitung berbasis Android serta memanfaatkan sensor *Accelerometer*

1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini antara lain :

1. Memberikan alternatif pembelajaran berhitung serta pengetahuan dari *game* berhitung bagi pengguna.
2. Memberikan media hiburan bagi para pengguna

1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

1. Studi literatur

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

1. Pembangkit Labirin Dinamis
2. Unity
3. Instalasi pada *Platform* Perangkat Mobile Android
4. Cara Bermain yang sesuai
5. Penyimpanan

2. Perancangan perangkat lunak

Pada tahap ini dilakukan analisa awal dan pendefinisian kebutuhan sistem untuk mengetahui permasalahan yang sedang dihadapi. Selanjutnya, dirumuskan rancangan sistem yang dapat

memberi solusi terhadap permasalahan tersebut. Langkah yang akan digunakan pada tahap ini adalah sebagai berikut:

1. Pencarian dan pendataan materi yang akan digunakan.
2. Perancangan pembangkit labirin dinamis
3. Perancangan sistem dan mekanisme permainan.
4. Analisis kebutuhan fungsional.
5. Analisis algoritma dan formula dalam membangun permainan.

3. Implementasi dan pembuatan sistem

Tahap implementasi merupakan tahap untuk membangun aplikasi permainan beserta sistem yang terkait. Aplikasi ini akan dibangun dengan bahasa pemrograman C# dengan *game engine* Unity. Aplikasi yang akan dibangun berbasis perangkat mobile Android dengan memanfaatkan sensor *accelerometer*.

4. Uji coba dan evaluasi

Pada tahap ini akan dilakukan pengujian terhadap perangkat lunak menggunakan data atau skenario yang telah dipersiapkan sebelumnya yakni sebagai berikut:

1. Pengujian *blackbox*

Pengujian *blackbox* adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, pengguna dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program. Pengujian ini dilakukan untuk menguji apakah proses kinerja aplikasi *game* ini sudah sesuai dengan kebutuhan pengguna atau tidak.

2. Pengujian Pengguna

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

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

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan

Bab ini berisi tentang desain sistem yang disajikan dalam analisis sistem, desain model permainan dan desain model perancangan antar muka.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode sumber yang digunakan untuk proses implementasi.

Bab V Pengujian Dan Evaluasi

Bab ini menjelaskan kemampuan *game* dengan melakukan pengujian kebenaran dan pengujian kinerja dari *game* yang telah dibuat.

Bab VI Kesimpulan Dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

Bab VII Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Bab VIII Lampiran Hasil Kuesioner

Merupakan data pendukung dari hasil uji coba pada pengguna dari *game* yang telah dibangun.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan algoritma yang diajukan pada pengimplementasian program. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak. Teori yang akan dijelaskan adalah Sistem Operasi Android, Unity, Algoritma Prims, *Classic Labyrinth*, *Sensor Accelerometer*, *Pembangkit Labirin Dinamis*.

2.1 Sistem Operasi Android

Menurut Nazrudin Safaat H (2011 : 1, “Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware, dan aplikasi.”. Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google Inc. Untuk pengembangannya, dibentuklah Open Handset Alliance (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. [1]

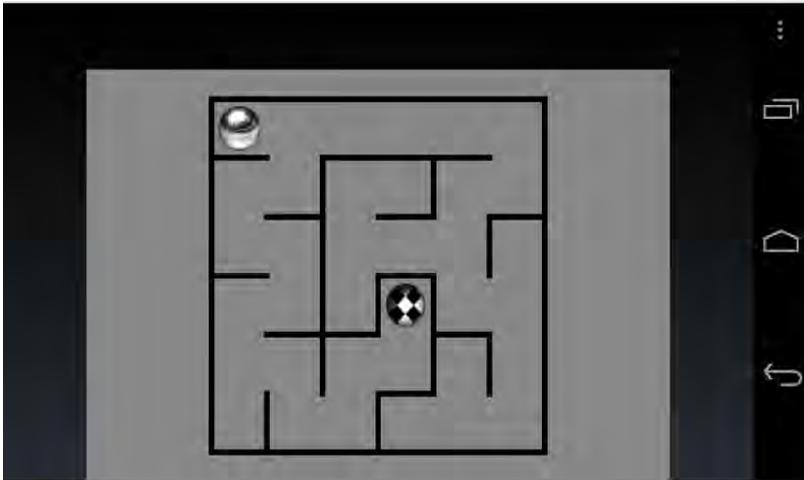
2.2 Unity

Unity adalah salah satu game engine berbasis 3D, dan mendukung beberapa bahasa pemrograman. Memiliki dukungan grafis yang halus untuk membuat environment game anda menjadi sangat cantik [2]. Selain itu IDE unity memiliki berbagai macam asset yang siap untuk digunakan. Selain itu Unity adalah *crossplatform game engine* dengan IDE yang dikembangkan dengan teknologi Unity. Hal ini digunakan untuk mengembangkan

game untuk *website*, platform desktop, dan perangkat seluler. Unity mendukung pengembangan untuk iOS, Android, Windows, Blackberry 10, OS X. Untuk melakukan scripting, pengguna dapat membangunnya di atas Mono, implementasi *open-source* dari .NET Framework. Pemrograman dapat menggunakan *UnityScript*, C# atau Boo [3]

2.3 Classic Labyrinth

Classic Labyrinth merupakan *game* yang tersedia pada *Google Play*. Permainan ini memanfaatkan sensor *accelerometer* untuk mendukung jalannya permainan. Pemain memiliki tujuan untuk menggerakkan bola atau objek yang tersedia kedalam lubang yang telah disediakan.



Gambar 2.1 Classic Labyrinth

Pemain akan menggunakan sensor *accelerometer* untuk menggerakkan bola atau objek kedalam lubang namun pemain juga harus menemukan jalan yang tepat untuk menuju ke lubang tersebut, karena pemain berada pada sebuah labirin seperti yang ditunjukkan pada Gambar 2.1 Pada permainan *classic labyrinth*

labirin yang diciptakan memiliki ukuran yang berbeda-beda sesuai dengan level yang dipilih [4].

2.4 Sensor Accelerometer.

Sensor accelerometer adalah elemen sensing atau sensor yang mengukur percepatan, percepatan adalah laju perubahan kecepatan terhadap waktu. Sensor accelerometer mengukur unit dalam satuan g dimana g adalah pengukuran percepatan gravitasi yang sama dengan 9.82 m/s^2 . *Accelerometer* berkembang dari tabung air sederhana dengan gelembung udara yang menunjukkan arah percepatan ke sirkuit terpadu yang dapat ditempatkan pada papan sirkuit. Sensor Accelerometer dapat mengukur getaran, guncangan, kemiringan, dampak dan gerak dari suatu objek [5] Seperti yang ditunjukkan pada Gambar 2.2



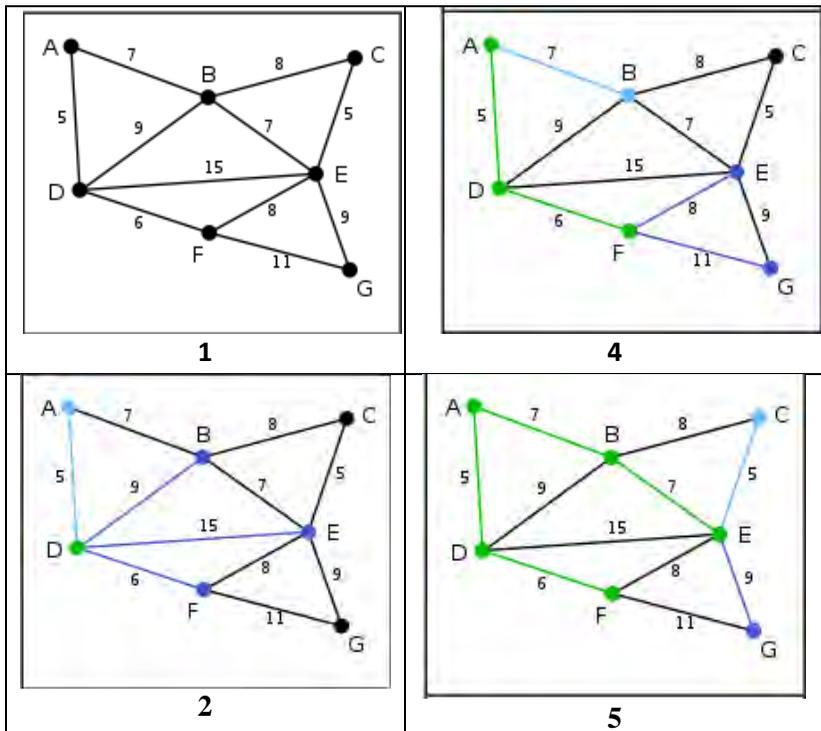
Gambar 2.2 Accelerometer pada Android

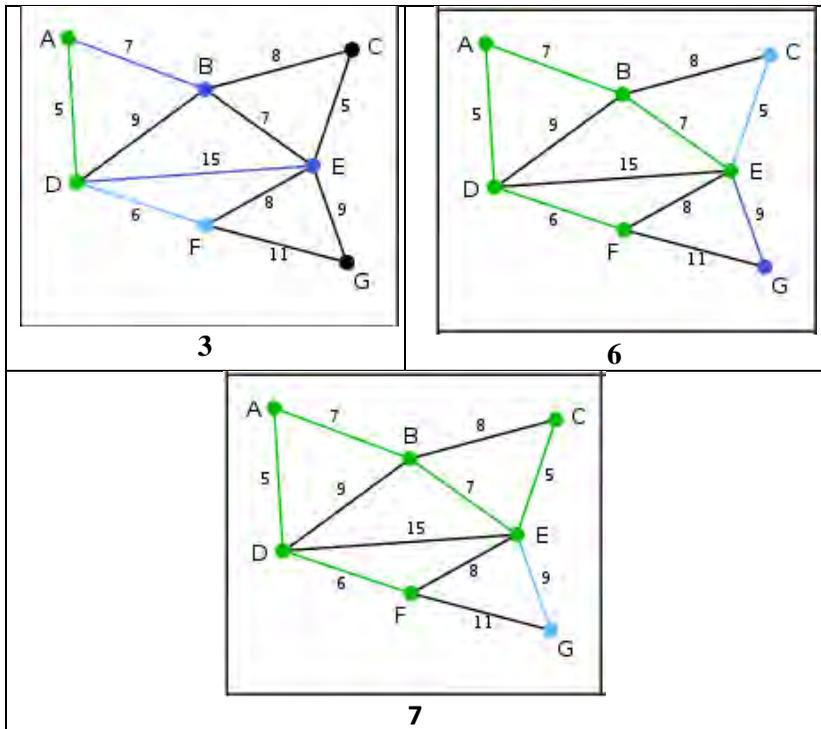
2.5 Algoritma Prims

Penggunaan pemodelan pohon rentang minimum adalah salah satu pemodelan graf yang digunakan untuk mencari nilai minimum dari jalur pada suatu graf atau dengan kata lain

menentukan jarak terpendek suatu titik dengan titik lainnya dengan melewati titik-titik yang ada di antaranya. Untuk mendapatkan hasil tersebut, maka dipilih ruas graf yang memenuhi kriteria dari optimasi yang akan menghasilkan biaya minimum, terdapat beberapa algoritma yang dipakai dalam membangun pohon rentang minimum, di antaranya Algoritma Prim's.

Pohon rentang minimum yang optimal memerlukan pemilihan algoritma yang bagus dan benar. Algoritma yang bagus dan benar adalah algoritma yang tepat atau efisien untuk diterapkan pada graf jenis tertentu. Penentuan pohon rentang minimum menggunakan Algoritma Prim's harus memperhatikan bobot minimum dari masing-masing ruas graf untuk menghasilkan *Minimum Spanning Tree* (MST) dengan bobot minimum [6].





Gambar 2.3 Pembobotan Graf dan Implementasi Algoritma Prims

Algoritma Prims dimulai dari simpul yang berubah-ubah di setiap tingkatnya, diperbolehkan menambah cabang baru untuk membuat susunan pohon baru. Algoritma ini akan tertahan ketika simpul yang sedang dieksplorasi pada graf sudah sampai pada simpul yang dituju. “Strategi yang digunakan adalah strategi Greedy dengan menganggap bahwa pada setiap langkah dari pohon merentangny adalah *augmented* dan dipilih simpul yang nilainya paling kecil dari semua simpul yang ada” (Purwanto, 2008).

Algoritma Prims menitik beratkan pada pemilihan bobot minimum berdasarkan simpul yang diambil. karena tidak perlu mengurutkan terlebih dahulu, Algoritma Prims cocok untuk pohon

dengan jumlah simpul banyak Algoritma Prims akan selalu berhasil menemukan pohon merentang minimum tetapi pohon merentang yang dihasilkan tidak selalu unik, Algoritma Prims pada suatu graf berbobot dapat dilihat Gambar 2.3 [7]. Langkah-langkah dalam algoritma Prims adalah sebagai berikut:

1. Buat sebuah pohon yang terdiri dari satu simpul (node), dipilih secara acak dari graf.
2. Buat sebuah himpunan yang berisi semua cabang di graf.
3. Loop sampai semua cabang di dalam himpunan menghubungkan dua simpul di pohon.
 - a. Hapus dari himpunan satu cabang dengan bobot terkecil yang menghubungkan satu simpul di pohon dengan satu simpul di luar pohon.
 - b. Hubungkan cabang tersebut ke pohon. Algoritma Prims dalam *pseudocode* dapat dilihat pada Gambar 2.4 dan Gambar 2.5 menunjukkan Algoritma Prims.

```

Procedure Prim(input G:graf, output T:pohon) {
  Membentuk pohon merentang minimum T dari graf
  terhubung G.
  Masukan: graf-berbobot terhubung  $G = (V, E)$ , yang mana
   $|V| = n$ 
  Keluaran: pohon merentang minimum  $T = (V, E')$  }

```

Gambar 2.4 Pseudocode Algoritma Prims

Deklarasi

i, p, q, u, v : integer

Algoritma

Cari sisi (p,q) dari E yang berbobot terkecil

$T \leftarrow \{(p,q)\}$

for $i \leftarrow 1$ to $n-1$ do

Pilih sisi (u,v) dari E yang bobotnya terkecil namun bersisian dengan suatu simpul didalam T

$T \leftarrow T \cup \{(u,v)\}$ Endfor

Gambar 2.5 Algoritma Prims

BAB III ANALISIS DAN PERANCANGAN

Pada bab ini akan dibahas mengenai analisis dan perancangan yang akan digunakan untuk menyelesaikan tugas akhir. Bab ini dibagi menjadi dua bagian. Pertama membahas mengenai pembangkit labirin permainan secara dinamis pada semua skenario yang mengandung edukasi. Bagian ini berisi data masukan, proses, serta keluaran yang dihasilkan, serta algoritma yang digunakan.

Bagian kedua berisi perancangan permainan menggunakan pembangkit labirin dinamis yang telah dibuat pada proses sebelumnya yang berisi analisis dan perancangan sistem.

3.1 Analisis Sistem

Aplikasi pada *smartphone* Android berkembang sangat pesat sejalan dengan perkembangan zaman. Masyarakat dengan segala golongan, usia serta dalam jumlah yang tidak sedikit telah memiliki *smartphone* Android dan memiliki ketertarikan yang tinggi terhadap aplikasi penunjang didalamnya. Permainan 'game' merupakan salah satu aplikasi yang terdapat pada *smartphone* Android, *game* yang beragam, tampilan yang baik agar memunculkan ketertarikan pengguna, sehingga memberikan rasa yang menyenangkan ketika bermain serta memberikan efek adiktif dalam memainkannya. Begitu banyak jenis *game* seperti *game* berkonsep labirin. Salah satu faktor yang mempengaruhi tingkat kesenangan pemain dari *game*, terutama *game* berkonsep labirin adalah susunan labirin tersebut. Susunan labirin yang beragam tentunya membutuhkan variasi pada *game*. Pembuatan susunan *game* labirin secara manual dengan *modeling* memakan waktu cukup lama. Dibutuhkan suatu sistem yang dapat membangkitkan susunan labirin pada *game* secara dinamis setiap kali suatu labirin ingin diciptakan.

Aplikasi ini dibangun dengan tujuan membantu para pemain dapat menikmati *game* yang variatif dan otomatis sekaligus dapat menambah ilmu pembelajaran yang edukatif. Susunan labirin pada sistem *game* dibangun secara dinamis karena menerapkan Algoritma

Prims serta berdasarkan tingkat kesulitan masing-masing. Data yang digunakan untuk membangun susunan labirin adalah pembobotan setiap level. Semakin tinggi level yang dipilih maka semakin tinggi pula pembobotan data *game* yang diberikan. Pembobotan level merepresentasikan tingkat kesulitan *game* disetiap level. Aplikasi ini diharapkan dapat memberikan pembelajaran yang menyenangkan kepada para pengguna yaitu mengenai pengerjaan persoalan tanpa meninggalkan unsur menyenangkan pada suatu *game*. Memunculkan perasaan tertarik dari pemain ke *game* juga disisipkan melalui kontrol dalam melakukan permainan tersebut, oleh karenanya pada *game* yang akan dibangun akan memanfaatkan *sensor accelerometer* dalam memberikan kontrol terhadap permainan.

Penulis menggunakan teknologi Unity dengan Bahasa pemrograman C# untuk memfasilitasi pengembangan permainan. Aplikasi yang dibangun dengan pembangkit labirin dinamis, *game* yang sudah dibangkitkan sebelumnya yang membuat penulis tidak perlu lagi mendesain secara manual. Serta dapat fokus pada fitur lain dari permainan misalkan *gameplay* dan cerita.

3.2 Perancangan Sistem

3.2.1 Deskripsi Umum Perangkat Lunak

Tugas akhir yang akan dikembangkan adalah sebuah permainan 3D bergenre TPS (*Third Person Shooter*) dan edukasi yang dikombinasikan. Susunan labirin dalam permainan dibangkitkan secara dinamis dan variatif dengan menggunakan pembobotan yang terarah dan meningkat. Permainan ini akan dijalankan dan dimainkan pada perangkat *smartphone* Android agar dapat dimainkan secara fleksibel. Pengguna utama dari permainan ini adalah semua orang yang ingin bermain. Pemain berperan sebagai karakter utama. Pemain dapat memilih salah satu dari sepuluh level yang ditawarkan. Pemain memiliki misi untuk menyelesaikan soal mengenai perhitungan, soal akan ditampilkan pada sisi layar bagian atas, kemudian untuk memilih jawaban yang sesuai, dengan cara mengarahkan tembakan pada jawaban yang diinginkan, yang muncul diantara beberapa pilihan jawaban yang diberikan.

Pemain memiliki hambatan berupa menemukan jalan yang sesuai dalam mendapat jawaban yang sesuai, musuh yang akan menghadang di beberapa tempat serta keterbatasan waktu yang diberikan. Pembobotan pada setiap level yang ditawarkan, diberikan secara terarah dan meningkat. Pemain akan mendapatkan nilai dengan cara membunuh musuh serta menjawab pertanyaan dengan waktu secepat mungkin. Dalam sistem permainan ini, pemain mengontrol satu karakter utama dengan memanfaatkan sensor *accelerometer* yang terdapat dalam perangkat keras Android. Pemain dapat menyelesaikan alur sesuai dengan skenario yang sudah ditentukan. Pemain juga dapat melihat nilai tertinggi di setiap level yang ditawarkan.

Pemain dinyatakan mampu memenangkan permainan jika mampu pemain mampu mengumpulkan skor lebih dari seratus sebelum batas waktu berakhir atau nyawa karakter utama habis, dengan memenangkan permainan, maka pemain dapat membuka level selanjutnya sehingga menambah ketersediaan level yang dipilih sedangkan, pemain akan dinyatakan kalah apabila nyawa karakter utama telah habis atau waktu bermain telah habis sebelum pemain mengumpulkan skor sebanyak lebih dari seratus.

3.2.2 Karakteristik Pengguna

Tabel 3.1 Karakteristik Pengguna

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

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

3.2.3 Perancangan Pembangkit Labirin Dinamis

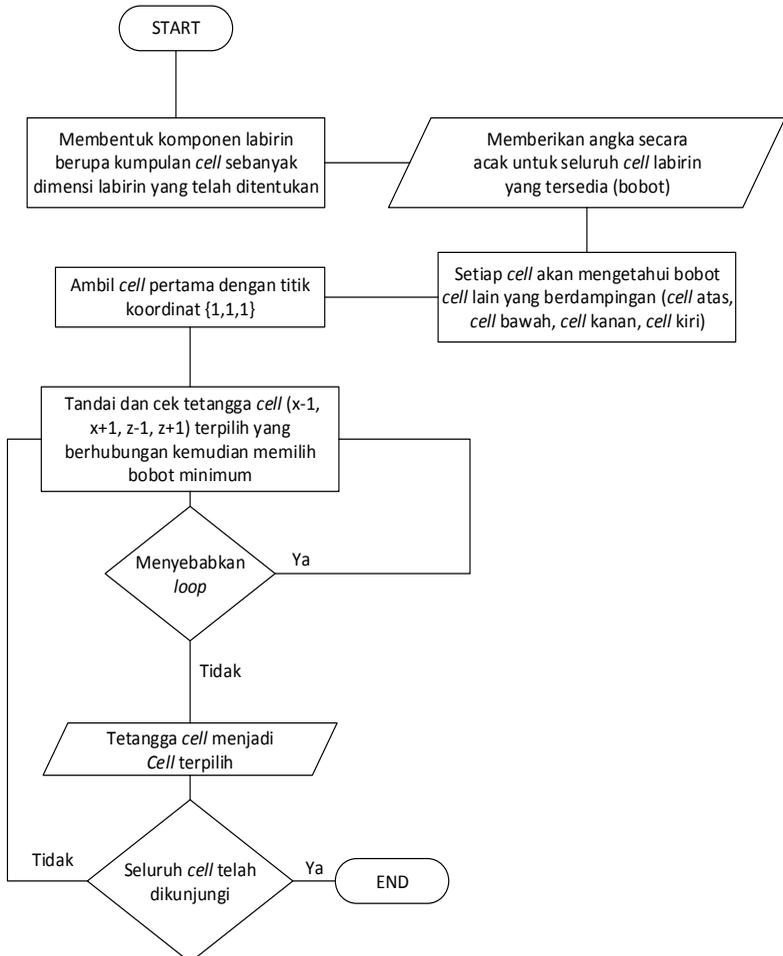
Pembangkit labirin dinamis yang akan dibangun akan memanfaatkan Algoritma Prim's. Algoritma Prim's digunakan untuk menentukan keseimbangan dari dinding labirin dan jalanan labirin, sehingga perancangan labirin dinamis yang akan didapatkan seperti yang ditunjukkan pada Gambar 3.2 dapat dikembangkan menjadi konsep dengan latar belakang permainan berhitung.

Penentuan dimensi atau panjang dan lebar dari labirin akan selalu ditentukan secara acak namun tetap disesuaikan dengan level yang dipilih, sehingga semakin besar level yang dipilih dimensi atau panjang dan lebar labirin akan mempengaruhi bobot tantangan yang akan dihadapi oleh pemain. Algoritma Prim's yang digunakan pada pembangkit labirin dinamis selain memiliki tujuan untuk mengatur keseimbangan juga bertujuan untuk memastikan bahwa pada labirin yang dibangun selalu memiliki jalanan utama diantara jalanan yang terdapat pada labirin. Jalanan utama yaitu jalanan yang tidak terhalang dengan dinding labirin. Tahapan dalam membangun pembangkit labirin dinamis adalah sebagai berikut :

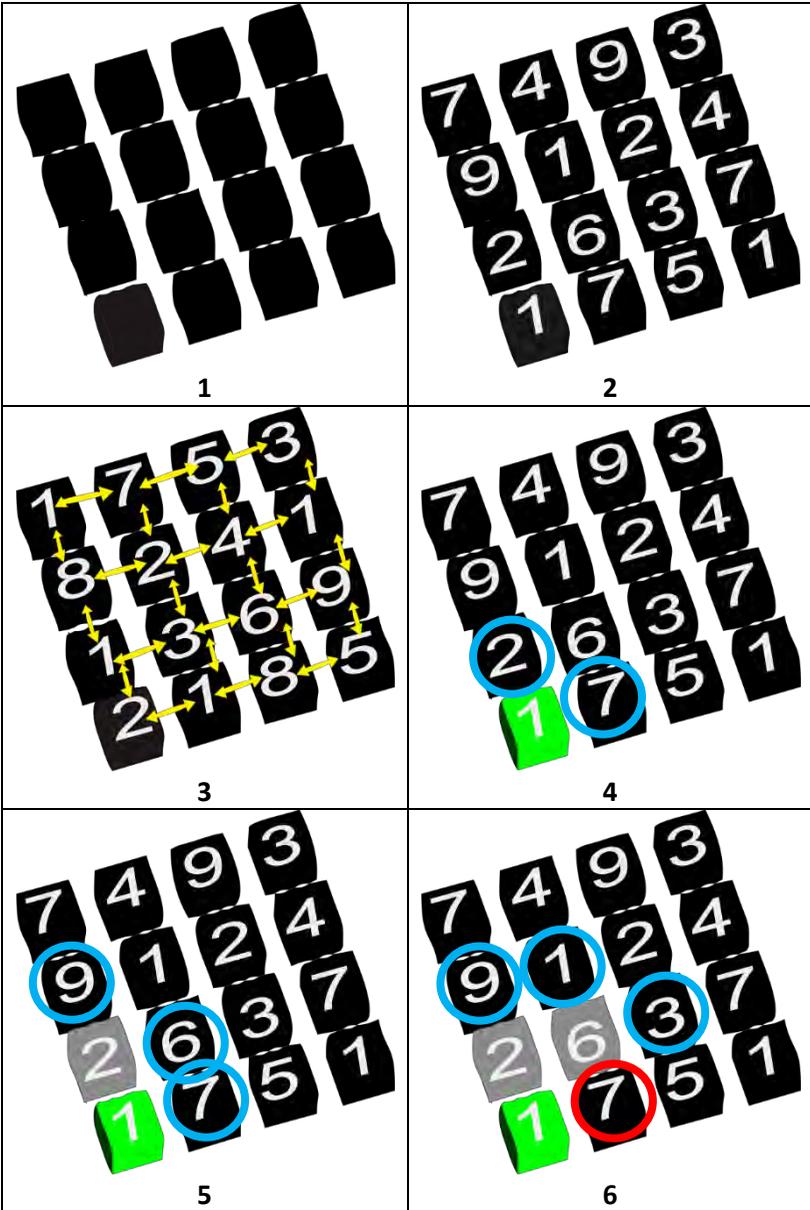
1. Membangun *cell* penyusun labirin sebanyak dimensi yang telah ditentukan.
2. Memberikan angka secara acak untuk seluruh *cell* penyusun labirin yang tersedia, selanjutnya akan disebut sebagai bobot
3. Setiap *cell* akan mengetahui bobot *cell* lain yang berdampingan (*cell* atas, *cell* bawah, *cell* kanan, *cell* kiri)
4. Memilih salah satu *cell*, pada labirin yang akan dibangun *cell* dengan koordinat $\{1,1,1\}$ akan selalu menjadi *cell* awal
5. Selanjutnya dari *cell* yang terpilih akan melakukan pengecekan terhadap *cell* yang berdampingan/tetangga $(x-1, x+1, z-1, z+1)$ berdasarkan bobot *cell* tersebut serta apakah *cell* tersebut menyebabkan *loop* atau tidak
6. Jika salah satu *cell* yang berdampingan memiliki bobot minimum dibanding *cell* lain yang berdampingan dan terpilihnya *cell* tersebut tidak menyebabkan *loop* maka *cell* tersebut akan menjadi *cell* terpilih dan menjadi jalan dan *cell* yang pernah berdampingan dengan *cell* terpilih akan selalu

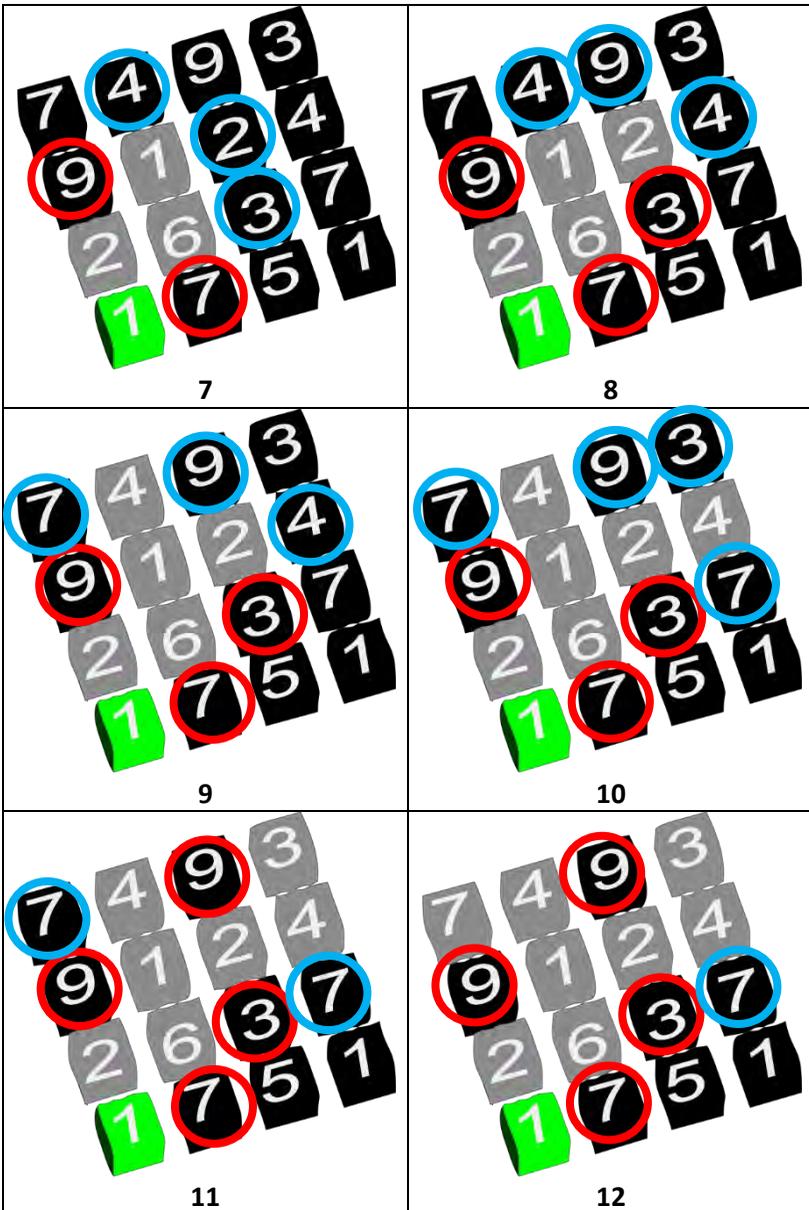
tersimpan, dimana dapat digunakan kembali sebagai *cell* pembanding selama *cell* tersebut tidak menyebabkan *loop*

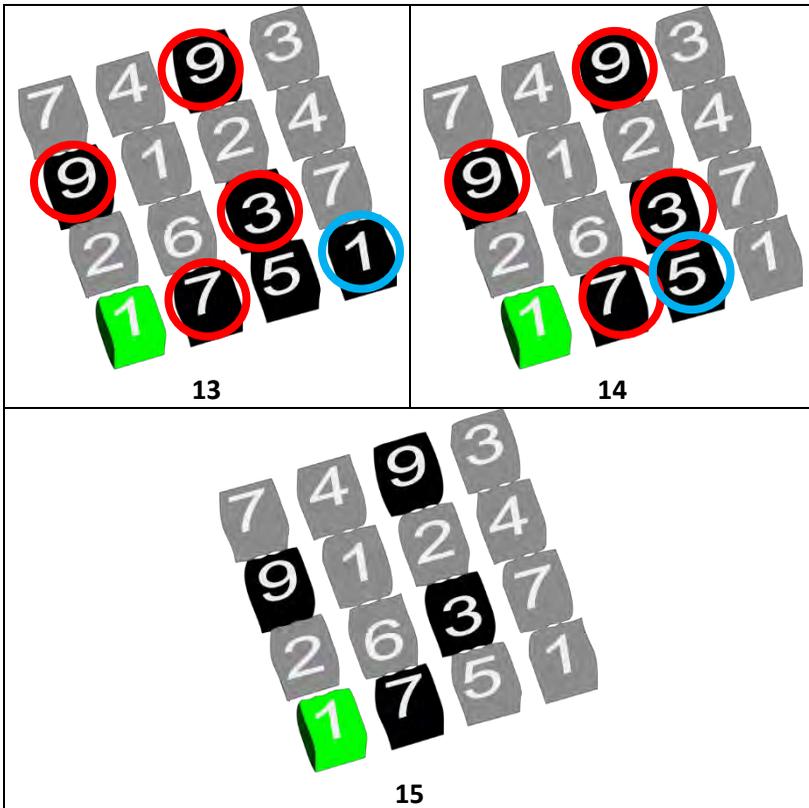
7. *Cell* yang tidak terpilih karena dapat menyebabkan *loop* akan menjadi dinding pada labirin yang akan dibangun.



Gambar 3.1 Diagram Alir Pembangkit Labirin Dinamis



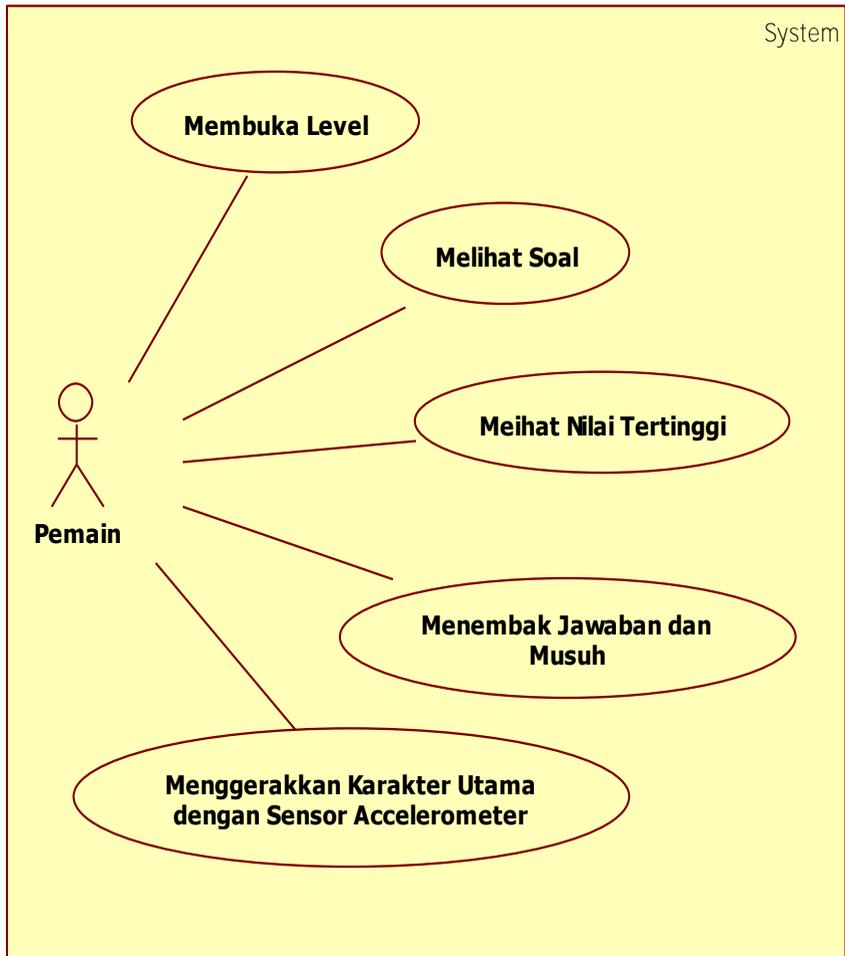




Gambar 3.2 Perancangan Pembangkit Labirin Dinamis

Pada Gambar 3.1 menunjukkan diagram alir dalam pembuatan pembangkit labirin dinamis sedangkan pada Gambar 3.2 menunjukkan perancangan pembangkit labirin dinamis yang akan dibangun, dimana terdapat lingkaran biru merupakan kemungkinan *cell* yang akan dipilih dan lingkaran merah merupakan *cell* yang tidak akan dipilih karena menyebabkan *loop* sehingga dari perancangan tahapan dapat dilihat *cell* yang terpilih karena bobot minimum dan tidak menyebabkan *loop* akan menjadi jalan pada labirin, sedangkan *cell* yang tidak terpilih akan menjadi dinding pada labirin.

3.2.4 Perancangan Diagram Kasus Penggunaan



Gambar 3.3 Diagram kasus aplikasi

Kasus penggunaan yang terdapat di dalam sistem dicantumkan pada Gambar 3.3. seperti, Membuka level, Melihat Soal,

Melihat Nilai Tertinggi, Menjawab Soal, Menembak Jawaban dan Musuh, dan Menggerakkan Karakter Utama dengan Memanfaatkan sensor *accelerometer*. Penjelasan dari masing-masing kasus penggunaan dicantumkan pada Tabel 3.2. Tabel tersebut berisi penjelasan skenario yang akan dilakukan ketika pengujian.

Tabel 3.2 Skenario Kasus Penggunaan

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan
1	UC-001	Membuka Level
2	UC-002	Melihat Soal
3	UC-003	Melihat Nilai Tertinggi
5	UC-004	Menembak Jawaban dan Musuh
6	UC-005	Menggerakkan Karakter Utama dengan Sensor Accelerometer

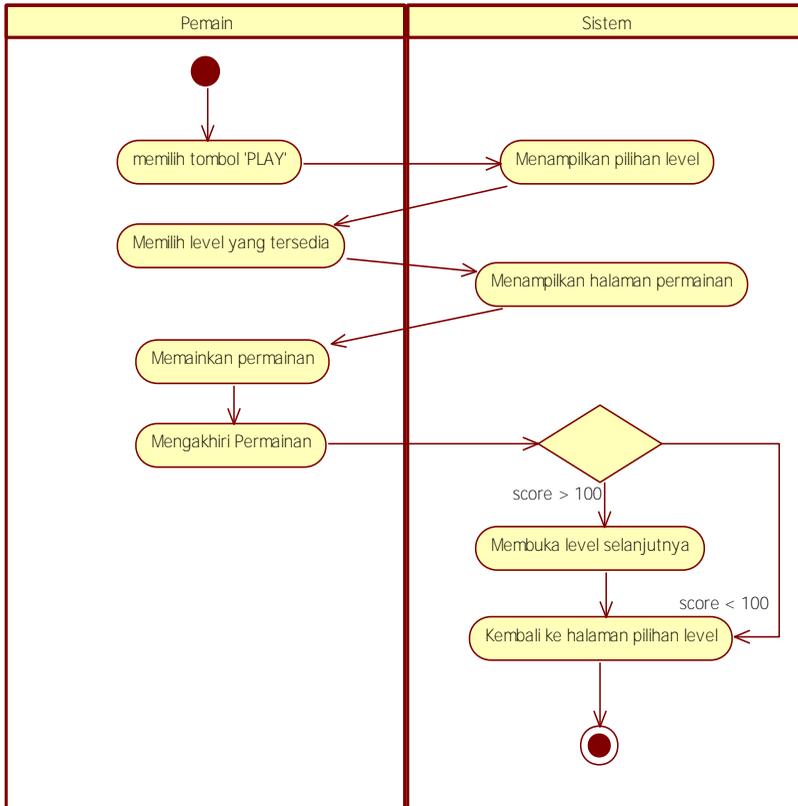
3.2.4.1 Kasus Penggunaan Membuka Level

Penjelasan kasus penggunaan permainan untuk skenario UC-001 yakni Membuka Level ditampilkan pada Tabel 3.3 dan Diagram Aktivitas Membuka Level ditampilkan pada Gambar 3.4 yaitu, diawali dengan pemain memilih tombol ‘PLAY’ yang terdapat pada menu utama, system menampilkan pilihan level.

Pemain memilih level yang tersedia dan memainkan hingga mendapatkan score atau nilai lebih dari seratus. Setelah permainan berakhir pemain dapat melihat pada halaman pemilihan level, bahwa level selanjutnya telah terbuka dan siap untuk dimainkan

Tabel 3.3 Skenario Kasus Penggunaan Membuka Level

Nama Kasus Penggunaan	Membuka Level
Kode	UC-001
Deskripsi	Memberikan tantangan pada pemain, bahwa seluruh level, kecuali level pertama dalam keadaan terkunci sehingga untuk membuka level mewajibkan kepada pemain memenuhi syarat yang telah ditentukan.
Aktor	Pemain.
Kondisi Awal	Pemain sudah masuk ke aplikasi permainan
Alur Normal	<ol style="list-style-type: none"> 1. Pemain memilih tombol 'PLAY' dan menekan klik. 2. Sistem menampilkan pilihan level. 3. Pemain memainkan hingga mendapatkan nilai lebih dari 100 4. Pemain mengakhiri permainan. 5. Sistem kembali menampilkan pilhan level serta level selanjutnya yang telah terbuka, sehingga dapat dimainkan oleh pemain
Alur Alternatif	<p>3.A Pemain tidak mendapatkan nilai lebih dari 100.</p> <p>3.A.1 Pemain mengakhiri permainan.</p> <p>3.A.2 Sistem kembali menampilkan pilhan level, level selanjutnya masih dalam keadaan terkunci.</p>



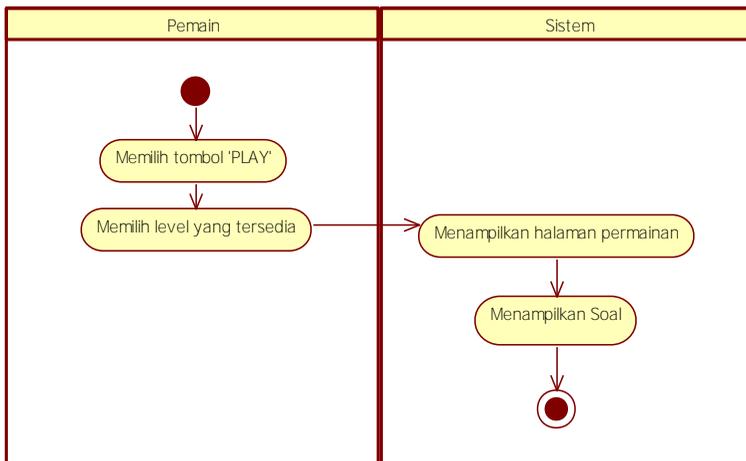
Gambar 3.4 Diagram aktivitas Membuka Level

3.2.4.2 Kasus Penggunaan Melihat Soal

Selanjutnya penjelasan kasus penggunaan permainan untuk skenario UC-002 yakni Melihat Soal ditunjukkan pada Tabel 3.4 dan diagram aktivitas Melihat Soal ditunjukkan pada Gambar 3.5. Yaitu, pemain memilih menu 'PLAY', kemudian memilih level yang tersedia, kemudian sistem akan menampilkan halaman permainan dan soal pada sisi kanan atas layar.

Tabel 3.4 Skenario Kasus Penggunaan Melihat Soal

Nama Kasus Penggunaan	Melihat Soal
Kode	UC-002
Deskripsi	Memberikan fitur pada pemain dalam menampilkan soal terbaru selama permainan berlangsung
Aktor	Pemain.
Kondisi Awal	Pemain sudah masuk ke aplikasi
Alur Normal	<ol style="list-style-type: none"> 1. Pemain memilih tombol 'PLAY' 2. Pemain memilih level yang tersedia 3. Sistem menampilkan halaman permainan 4. Sistem menampilkan soal secara random pada sudut kanan atas layar.

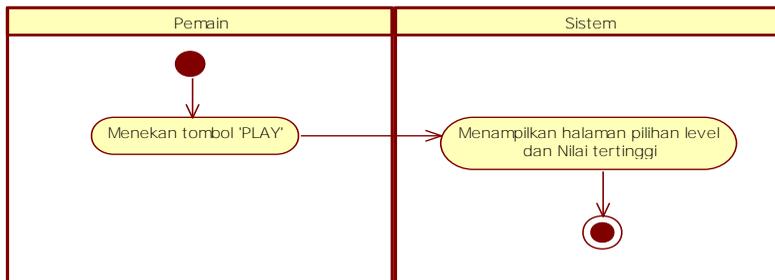
**Gambar 3.5 Diagram aktivitas Melihat Soal**

3.2.4.3 Kasus Penggunaan Melihat Nilai Tertinggi

Kemudian penjelasan kasus penggunaan permainan untuk skenario UC-003 Melihat Nilai Tertinggi dijelaskan pada Tabel 3.5 dan Diagram aktivitas Melihat Nilai Tertinggi ditunjukkan pada Gambar 3.6. Pemain memilih tombol ‘PLAY’ pada menu utama permainan, kemudian sistem menampilkan halaman untuk memilih level dan nilai tertinggi perlevel.

Tabel 3.5 Skenario Kasus Penggunaan Melihat Nilai Tertinggi

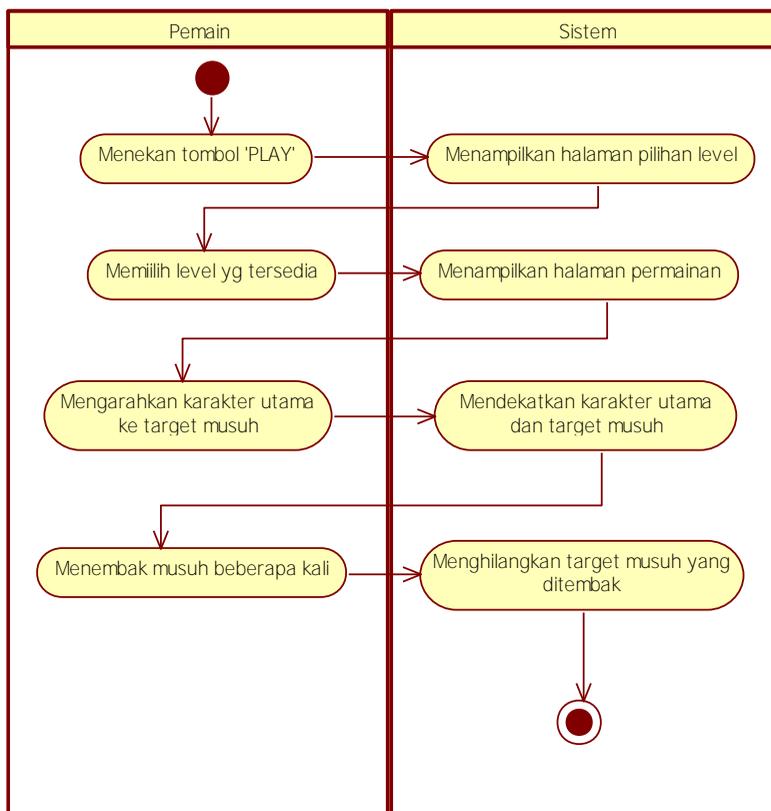
Nama Kasus Penggunaan	Melihat Nilai Tertinggi
Kode	UC-003
Deskripsi	Memberikan fitur pada pemain untuk mengetahui informasi terbaru seputar nilai tertinggi pada setiap level
Aktor	Pemain.
Kondisi Awal	Pemain sudah masuk ke aplikasi
Alur Normal	<ol style="list-style-type: none"> 1. Pemain memilih tombol ‘PLAY’ dan menekan klik. 2. Sistem menampilkan pilihan level serta nilai tertinggi disetiap level.



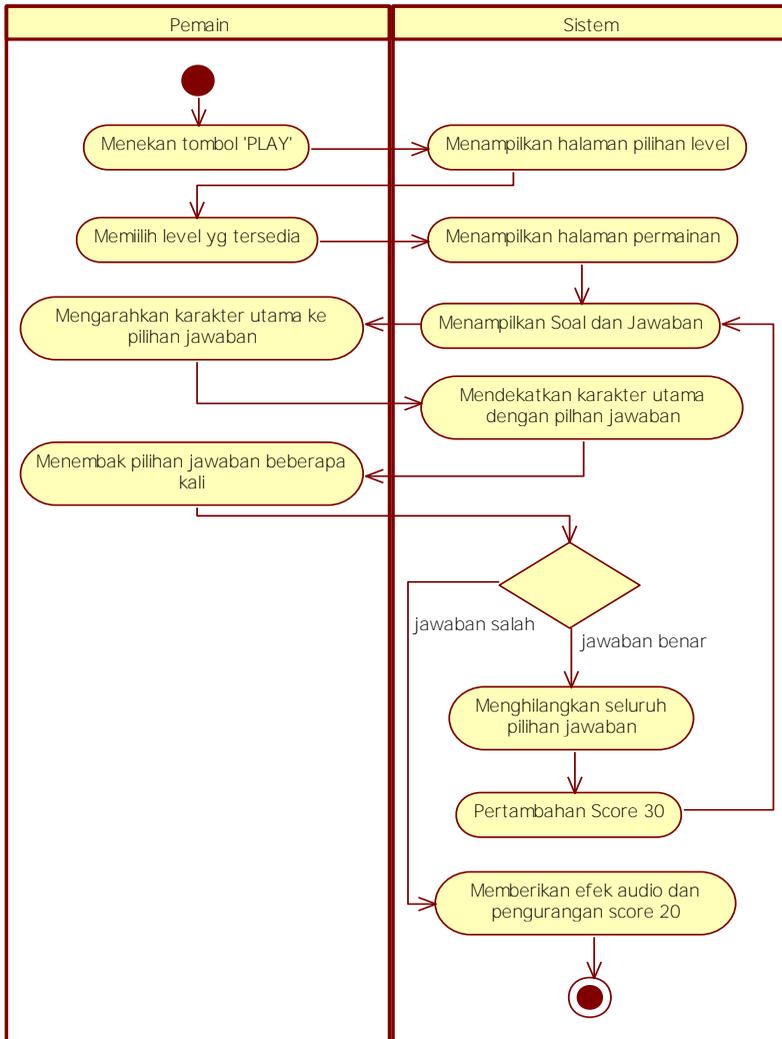
Gambar 3.6 Diagram aktivitas Melihat Nilai Tertinggi

3.2.4.4 Kasus Penggunaan Menembak Jawaban dan Musuh

Penjelasan kasus penggunaan permainan untuk skenario UC-004 yakni Menembak Jawaban atau Menjawab soal dan Musuh ditunjukkan pada Tabel 3.6, Diagram Aktivitas Menembak Musuh ditunjukkan pada Gambar 3.7 serta Diagram Aktivitas Menembak Jawaban ditunjukkan pada Gambar 3.8.



Gambar 3.7 Diagram aktivitas Menembak Musuh



Gambar 3.8 Diagram aktivitas Menembak Jawaban

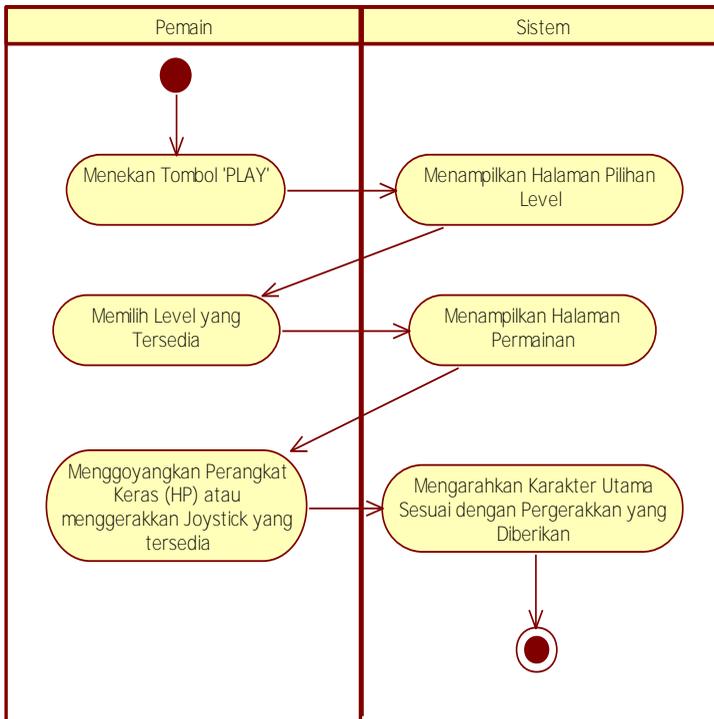
Tabel 3.6 Skenario Kasus Penggunaan Menembak Jawaban dan Musuh

Nama Kasus Penggunaan	Menembak Jawaban dan Musuh
Kode	UC-004
Deskripsi	Memberikan fitur tembakan untuk karakter utama dalam menembak musuh dan memilih pilihan jawaban yang diinginkan.
Aktor	Pemain.
Kondisi Awal	Pemain sudah masuk ke aplikasi
Alur Normal	<ol style="list-style-type: none"> 1 Pemain memilih tombol 'PLAY' dan menekan klik. 2 Sistem menampilkan pilihan level 3 Pemain memilih level yang tersedia 4 Sistem menampilkan halaman permainan, seperti soal, pilihan jawaban dan musuh yang tersedia pada labirin, 5 Pemain mendekati pilihan jawaban atau musuh dan mengarahkan beberapa tembakan pada target tersebut dengan cara menekan tombol yang telah disediakan. 6 Sistem merespon dengan audio dan hilangnya karakter musuh ketika pemain menembak musuh sebanyak 5x 7 Sistem merespon dengan hilangnya seluruh pilihan jawaban ketika pemain menembak jawaban yang sesuai sebanyak 3x 8 Sistem merespon dengan audio dan berkurangnya nyawa karakter utama ketika pemain menembak jawaban yang tidak sesuai sebanyak 3x

Diagram aktivitas dan skenario diawali dengan pemain memilih tombol 'PLAY' pada menu utama, setelah sistem

menampilkan pilihan level, pemain memilih level yang tersedia, kemudian sistem mengarahkan pemain ke halaman permainan dengan level yang telah dipilih, ketika soal dan pilihan jawaban telah muncul, pemain akan mengarahkan karakter utama untuk mendekati ke pilihan jawaban dan melakukan beberapa kali tembakan sebagai cara untuk menjawab pertanyaan yang telah ditampilkan, pada permainan juga tersedia tantangan berupa musuh yang akan mengejar pemain sehingga pemain juga dapat menghindari serangan dari musuh dengan cara menekan tombol yang telah disediakan ke musuh dengan jarak tembak yang sesuai.

3.2.4.5 Kasus Penggunaan Menggerakkan Karakter Utama



Gambar 3.9 Diagram Aktivitas Menggerakkan Karakter Utama

Tabel 3.7 Skenario Kasus Penggunaan Menggerakkan Karakter Utama dengan Sensor Accelerometer

Nama Kasus Penggunaan	Menggerakkan Karakter Utama dengan Sensor <i>Accelerometer</i>
Kode	UC-006
Deskripsi	Memberikan fitur kepada pemain, kemampuan untuk menggerakkan karakter utama
Aktor	Pemain
Kondisi Awal	Pemain sudah masuk ke aplikasi
Alur Normal	<ol style="list-style-type: none"> 1 Pemain memilih tombol 'PLAY' dan menekan klik. 2 Sistem menampilkan pilihan level 3 Pemain memilih level yang tersedia 4 Sistem menampilkan halaman permainan, seperti soal, pilihan jawaban dan musuh yang tersedia pada labirin, 5 Pemain melakukan kontrol terhadap karakter utama ketika mendekati target dan menghindari dengan cara menggerakkan <i>smartphone</i> atau mengatur arah wajah karakter utama dengan menggunakan <i>joystick</i> 6 Sistem merespon dengan mengarahkan karakter utama sesuai dengan pergerakan yang pemain lakukan

Penjelasan kasus penggunaan atau kontrol permainan untuk skenario UC-005 dan diagram aktivitas Menggerakkan Karakter Utama yang ditunjukkan pada Gambar 3.9 yakni Menggerakkan Karakter Utama dengan sensor *accelerometer* dan *joystick* dijelaskan pada Tabel 3.7. Skenario Kasus Penggunaan Menggerakkan Karakter Utama dengan sensor *accelerometer* diawali dengan Pemain memilih tombol 'PLAY' pada menu utama, setelah sistem menampilkan

pilihan level, pemain memilih level yang tersedia, kemudian sistem mengarahkan ke halaman permainan dengan level yang telah dipilih. Selanjutnya, pemain dapat menggerakkan karakter utama ke arah kanan, kiri, depan dan belakang dengan memanfaatkan sensor *accelerometer* yang terdapat pada Android serta disediakan *joystick* untuk mengatur arah wajah karakter utama.

3.2.5 Perancangan Alur Permainan

Alur permainan merupakan serangkaian proses yang harus diikuti pemain untuk memperoleh kemenangan. Permainan ini memiliki tujuan pemain dapat menjawab soal yang diberikan. Pemain akan mengarahkan karakter utama untuk mendekat kepilihan jawaban yang dirasa sesuai dengan soal yang telah diberikan, pemain dapat menembak jawaban yang dipilih, setiap jawaban yang benar akan memberikan *score* 30 dan diberikan soal terbaru selama waktu masih tersisa. Apabila, pemain menembak jawaban yang tidak sesuai dengan soal yang ditampilkan, *score* pemain akan dikurangi 20 dan nyawa karakter utama akan berkurang. Ketika waktu telah habis atau nyawa karakter utama telah habis, *score* saat ini akan dibandingkan dengan *score* sebelumnya pada level yang sama. Sehingga, *score* tertinggi dapat selalu muncul pada halaman memilih level.

Selama permainan berlangsung, pemain dapat mendekati pilihan jawaban, menghindari musuh, menembak pilihan jawaban dan menembak musuh. Kemudian untuk level pertarungan dibangun secara meningkat sehingga semakin tinggi level yang ditempuh maka semakin tinggi pula bobot tantangan yang harus dihadapi oleh pemain. Memenangkan level akan membuka level selanjutnya, sehingga level yang dapat dimainkan akan bertambah. Tabel 3.8 menunjukkan kondisi yang mencantumkan detil perancangan bobot faktor tantangan dan jarak minimum maksimum yang akan diberikan ke pemain dengan kolom yang berisi:

- Besar angka soal : yang dimaksudkan adalah, besarnya bilangan yang akan dijadikan soal seperti : $2 + 2$, $4 * 8$, dan seterusnya.
- Banyaknya pilihan jawaban yang muncul pada tiap level yang dimainkan.

- Jumlah musuh banyaknya musuh yang muncul pada tiap level yang dimainkan.
- Kecepatan musuh, kecepatan pergerakan musuh ketika mengejar karakter utama.
- Jumlah labirin pada sumbu X dan jumlah labirin pada sumbu Y.

Tabel 3.8 Detil Perancangan Bobot Faktor

No	Faktor Tantangan	Minimal	Maksimal	Bobot (%)
1.	Angka Soal	2	30	25
2.	Pilihan Jawaban	5	22	20
3.	Jumlah Musuh	2	10	15
4.	Kecepatan Musuh	3	8	30
5.	X	5	15	5
6.	Y	5	15	5

(3.1)

$$bobot\ level = \sum \frac{faktor\ tantangan - Minimum}{(Maksimum - Minimum)} * Bobot$$

(3.2)

$$Selisih\ bobot\ Level = \frac{100}{Jumlah\ level}$$

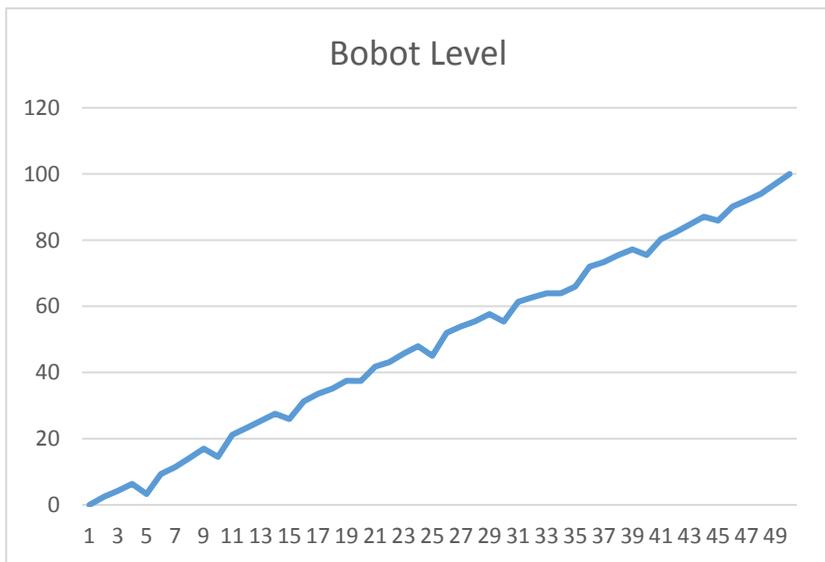
Selanjutnya pada Tabel 3.9 merupakan hasil perancangan bobot yang diberikan pada tiap level berdasarkan bobot yang telah ditampilkan pada Tabel 3.8, Rumus 3.1 dan Rumus 3.2. Pada Rumus 3.1 faktor-faktor tantangan seperti soal, pilihan jawaban, musuh, kecepatan musuh, panjang labirin serta lebar dari labirin akan dibangun setelah ditentukan nilainya serta dapat disesuaikan dengan level permainan yang telah dipilih oleh pemain, sedangkan Rumus 3.2

digunakan untuk mengatur jarak kesulitan tiap level sehingga kesulitan yang dihadapi pemain akan bertahap. Diharapkan dengan adanya perancangan pembobotan ini, ketertarikan pemain dalam memainkan permainan ini lebih meningkat.

Tabel 3.9 Detil Perancangan Bobot Level

Level	Angka Soal	Pilihan Jawaban	Jumlah Musuh	Speed Musuh	P	L	Bobot Level
1	2	5	2	3	5	5	0
2	3	5	2	3	8	5	2.39285714
3	3	5	3	3	6	7	4.26785714
4	4	6	3	3	7	6	6.33718487
5	4	5	2	3	6	7	3.28571429
6	6	5	4	3	8	6	9.32142857
7	8	6	3	3	8	8	11.4086134
8	6	6	3	4	7	6	14.1228992
9	6	6	4	4	7	8	16.9978992
10	7	6	3	4	6	6	14.5157563
11	9	6	4	4	10	8	21.1764706
12	10	7	4	4	9	9	23.2457983
13	11	7	6	3	11	14	25.3886555
14	11	7	5	4	11	10	27.5136555
15	10	6	5	4	11	11	25.9443277
16	12	7	6	4	12	11	31.2815126
17	13	7	7	4	10	12	33.5493697
18	16	10	8	3	10	11	35.1323529
19	16	12	8	3	10	11	37.4852941
20	16	14	7	3	10	10	37.4632353
25	16	13	5	5	11	10	45.0367647

Level	Angka Soal	Pilihan Jawaban	Jumlah Musuh	Speed Musuh	P	L	Bobot Level
30	18	17	6	5	13	12	55.4033613
35	21	20	7	5	15	15	65.9863445
40	25	19	10	6	10	11	75.5063025
41	27	19	10	6	12	15	80.2920168
42	28	20	10	6	12	15	82.3613445
43	29	19	9	7	11	13	84.7027311
44	29	21	9	7	11	13	87.0556723
45	29	20	9	7	11	13	85.8792017
50	30	22	10	8	15	15	100

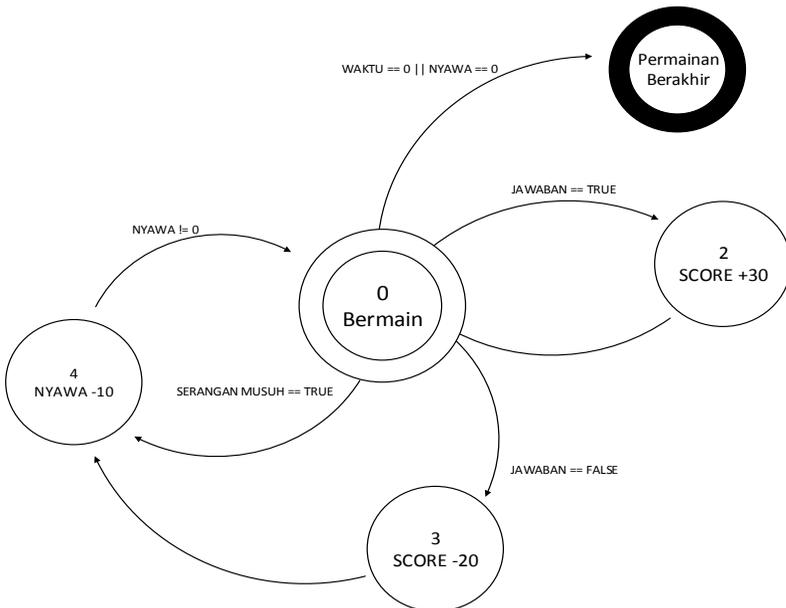


Gambar 3.10 Grafik Bobot Level

Gambar 3.10 menunjukkan Grafik Bobot Level permainan dimana bobot akan meningkat sesuai dengan besarnya level yang

dipilih. Namun, bobot level pun juga dapat menurun pada pilihan level tertentu. Hal ini bertujuan untuk menghindari rasa bosan pada pemain. Sedangkan alur permainan juga dapat dilihat dengan *Finite State Machine* (FSM) yang tercantum pada Gambar 3.11.

Finite State Machine (FSM) diawali dengan pemain yang telah memilih level yang tersedia dan masuk ke halaman permainan, kemudian pemain akan memainkan permainan, apabila pemain memilih jawaban yang sesuai dengan soal yang ditampilkan, maka nilai pemain akan bertambah sebanyak tiga puluh, apabila pemain memilih pilihan jawaban yang tidak sesuai, maka nilai pemain akan berkurang sebanyak dua puluh serta nyawa dari karakter utama akan berkurang sebanyak sepuluh dan nyawa karakter juga akan berkurang. Apabila nyawa karakter utama atau waktu permainan telah berakhir maka permainan juga akan berakhir.



Gambar 3.11 State diagram permainan

3.2.6 Perancangan Penyimpanan Data

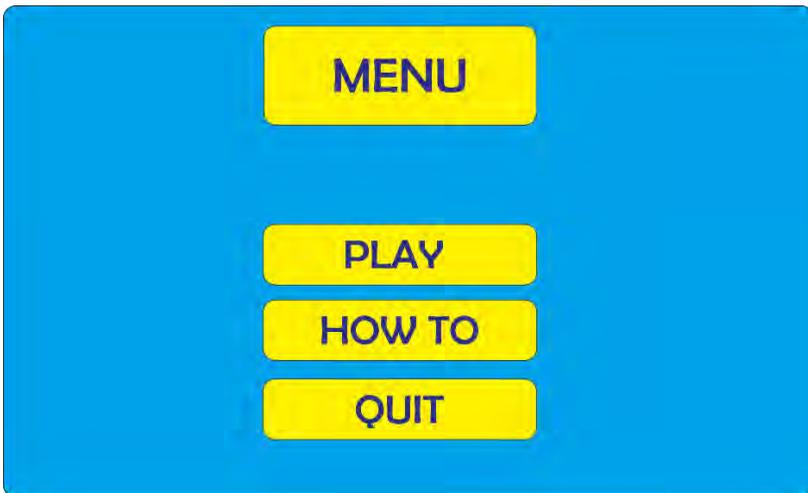
Fitur penyimpanan data berfungsi untuk mencatat perkembangan yang terjadi di dalam permainan. Pemain dapat menyimpan nilai tertinggi terakhir karakter utama dalam permainan dan level terakhir yang terbuka.

3.2.7 Perancangan Antarmuka Pengguna

Subbab ini membahas bagaimana rancangan antarmuka pengguna yang akan digunakan untuk tugas akhir. Rancangan antarmuka yang dibahas meliputi ketentuan masukan dan rancangan jendela tampilan. Dalam aplikasi ini terdapat beberapa tampilan, yaitu **Menu, Level dan Nilai Tertinggi, How To, dan Halaman Permainan**

3.2.7.1 Tampilan Menu

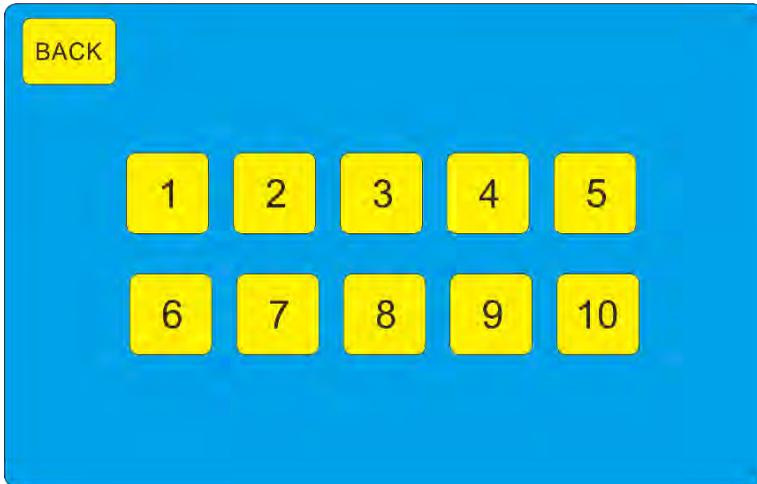
Tampilan awal menu merupakan tampilan yang pertama kali muncul ketika aplikasi dijalankan untuk yang pertama kalinya. Seperti yang ditampilkan pada Gambar 3.12. Tampilan ini menampilkan pilihan 'PLAY', 'HOW TO', dan 'QUIT'



Gambar 3.12 Tampilan Menu

3.2.7.2 Tampilan Level dan Nilai Tertinggi

Layar Tampilan Level dan Nilai Tertinggi ditampilkan secara bersamaan ketika pemain memilih tombol 'PLAY' pada layar menu Gambar 3.13 sehingga, ketika pemain memilih level yang tersedia, pemain akan langsung diarahkan ke halaman permainan berdasarkan level yang telah dipilih.



Gambar 3.13 Tampilan Level dan Nilai Tertinggi

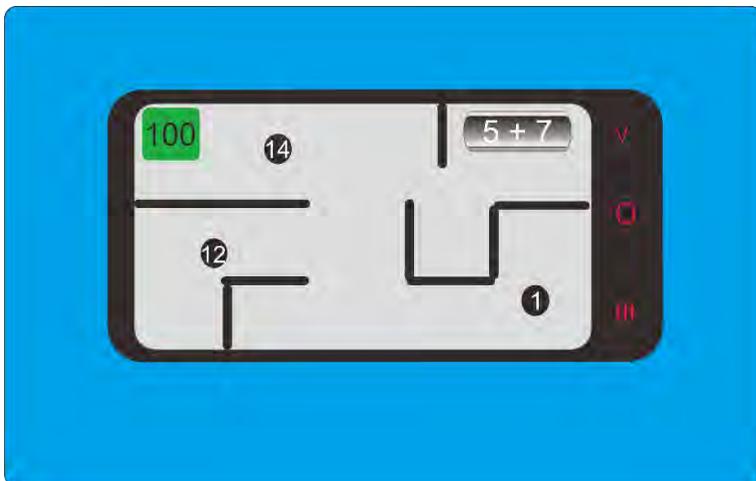
3.2.7.3 Tampilan How To

Tampilan 'HOW TO' merupakan tampilan yang ditampilkan ketika pemain memilih tombol 'HOW TO' pada halaman awal permainan. Pada halaman 'HOW TO' berfungsi untuk melihat deskripsi permainan dan cara bermain, tombol yang terdapat pada sisi kanan atas layar merupakan tombol yang berfungsi untuk mengarahkan ke halaman petunjuk selanjutnya, hingga seluruh halaman instruksi selesai ditampilkan. Setelah pemain selesai memahami cara bermain yang diberikan, pemain akan langsung diarahkan ke halaman level, untuk memilih bobot permainan sesuai dengan level yang diinginkan. Tampilan ini ditunjukkan pada Gambar 3.14.



Gambar 3.14 Tampilan Menu How To

3.2.7.4 Tampilan Permainan



Gambar 3.15 Tampilan Menu Permainan

Tampilan ini merupakan menu penting. Dari tampilan ini pengguna telah memilih bobot permainan berdasarkan level yang dipilih. Pada tampilan ini akan diberikan bentuk labirin, soal yang terletak pada kanan atas layar, pilihan jawaban yang terdapat pada jalanan labirin, batas waktu yang terdapat pada kiri atas layar, penilaian pada bagian tengah atas layar, jumlah musuh dan kecepatan musuh yang sesuai dengan bobot pada level tersebut. Tampilan ini ditunjukkan pada Gambar 3.15.

3.2.7.5 Tampilan Kontrol Permainan

Terdapat dua jenis kontrol dalam permainan yaitu memanfaatkan sensor *accelerometer* dan *joystick* untuk mengarahkan arah wajah karakter utama dan menembak ketika permainan dijalankan seperti yang ditunjukkan pada Gambar 3.16.



Gambar 3.16 Tampilan Kontrol Permainan *Virtual Joystick*

BAB IV IMPLEMENTASI

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

4.1 Lingkungan Implementasi

Lingkungan implementasi dari tugas akhir dijelaskan pada Tabel 4.1. dan Tabel 4.2

**Tabel 4.1 Lingkungan Implementasi Perangkat Lunak
(bagian 1)**

Perangkat Keras	Prosesor: Intel(R) Core(TM) i5-3317U CPU @1.70GHz 1.70GHz Memori: 8.00 GB
Perangkat Lunak	Sistem Operasi: Microsoft Windows 8.1 64-bit Perangkat Pengembang: Unity 4 Version 4.6.2f1

**Tabel 4.2 Lingkungan Implementasi Perangkat Lunak
(bagian 2)**

Perangkat Keras	Alat : <i>Smarthphone</i> Memori: 2 GB
Perangkat Lunak	Sistem Operasi: Android Perangkat Pengembang: Unity

4.2 Implementasi Permainan

Implementasi dari masing-masing fungsi utama dituliskan menggunakan kode sumber berdasarkan antarmuka utama yang ada pada permainan.

4.2.1 Implementasi Pembuatan Pembangkit Labirin

Fungsi pembuatan pembangkit labirin dinamis ditampilkan pada Gambar 4.1 merupakan fungsi yang digunakan untuk pembuatan labirin secara dinamis disetiap level yang dipilih oleh pemain dengan memanfaatkan Algoritma Prims. Fungsi ini diawali dengan pembuatan *cell* dasar pembuatan labirin. Pada implementasi disini, *cell* dasar yang dimaksud adalah *game object 3D 'cube'* yang terdapat pada Unity 4. *Cell* akan disusun sesuai dengan dimensi labirin yang diinginkan.

Selanjutnya *cell* yang telah tersusun, secara acak akan diberikan nilai yang kemudian dimanfaatkan untuk menjadi *weight* atau bobot. Setelah seluruh *cell* mendapatkan nilai, Algoritma Prims dijalankan. Dengan cara mengenali terlebih dahulu *cell* yang berdampingan (sisi kiri, sisi kanan, sisi bawah serta sisi atas) pada setiap *cell* penyusun labirin tersebut, nilai dari *cell* yang berdampingan tersebut akan tersimpan dengan urutan terkecil hingga terbesar.

Kemudian, dipilih '*node*' awal atau salah satu dari *cell* penyusun labirin, dari node tersebut Algoritma Prims akan mencari *cell* yang berdampingan dengan bobot terkecil diawali dari posisi (x-1, x+1, z-1 dan z+1), sehingga *cell* yang terpilih akan menjadi '*edge*' atau jalan pada labirin dan yang tidak terpilih akan menjadi dinding labirin. Hal ini akan dilakukan terus menerus hingga seluruh *cell* penyusun labirin telah dikunjungi.

```
// Fungsi pembuatan labirin//
void CreateGrid(){
    For each x in ( 0, Size.x ){
        For each z in ( 0, Size.z ){
            Transform newCell;
            newCell <- Instantiate(Cell);}}

void SetAdj(){
    For each x in ( 0, Size.x ){
        For each z in ( 0, Size.z ){
            Mengecek setiap cell yang berhubungan (x-1,
            x+1, z-1, z+1);
```

<pre> Mengurutkan cell berhubungan; } } int LowAdj(Transform A, Transform B){ Membandingkan bobot antar cell yang terhubung; } </pre>
<pre> void SetStart(){ Menyimpan cell dengan koordinat (1,1,1); } </pre>
<pre> void AddToSet(Transform toAdd){ menyimpan cell terpilih; foreach aj in toAdd < banyaknya tetangga pada cell { Menandai cell yang terhubung ; if(cell belum tersimpan && bobot yang dikunjungi belum tersimpan) Menyimpan bobot cell terhubung; } } </pre>
<pre> void FindNext() { do { Memilih cell yang terhubung dengan bobot minimum; } while(cell terhubung telah ditandai >= 2 cell tidak menjadi tepi labirin); AddToSet (next); FindNext (); } </pre>

Gambar 4.1 Pseudocode Pembangkit Labirin Dinamis

4.2.2 Implementasi Membuka Level

Pada fungsi yang ditampilkan pada Gambar 4.2, merupakan fungsi yang digunakan untuk mengecek *score* level sebelumnya apakah melebihi seratus atau tidak, diberikan pengecualian terhadap level satu. Sehingga, setiap level akan dicek

apakah *score* level sebelumnya tidak lebih dari seratus maka level selanjutnya akan dikunci.

```
// membuka level
button_level[now] = unlocked;
IF level != "1"
THEN
  IF level[now-1].score <=100;
  THEN button level[now] = locked;
```

Gambar 4.2 Pseudocode Membuka Level

4.2.3 Implementasi Pembuatan Soal dan Jawaban

Fungsi pembuatan soal dan jawaban ditampilkan pada Gambar 4.3 merupakan fungsi untuk membuat soal berdasarkan jarak angka yang ditentukan berdasarkan level yang telah dipilih oleh pemain. Kemudian, jawaban dari soal yang diberikan ditampilkan pada jalan labirin yang telah dibuat sebelumnya serta menampilkan soal selama permainan berlangsung.

```
// membuat soal dan jawaban//

void RandOption(int n)
{
  int randomAns <- random pilihan jawaban
  int ans <- jawaban soal;
  for (int i=0; i<n; i++)
  {
    Random letak posisi seluruh pilihan jawaban;
  }
  Penggambaran koordinat pilihan jawaban;
  Menentukan salah satu pertanyaan dari pilihan jawaban;
  Mengatur posisi tampilan pertanyaan;
}
```

Gambar 4.3 Pseudocode Pembuatan Soal dan Jawaban

4.2.4 Implementasi Tampilan Nilai Tertinggi

Pada Gambar 4.4 ditunjukkan untuk menampilkan Nilai tertinggi disetiap level, adalah dengan melakukan penyimpanan

terlebih dahulu terhadap nilai yang didapat oleh pemain sebelumnya. Jika pemain belum memulai permainan, maka nilai awal yang disimpan adalah nol. Selanjutnya adalah membandingkan nilai terbaru dengan nilai yang telah tersimpan, apabila nilai baru lebih tinggi dari nilai tersimpan, maka nilai baru tersebut akan disimpan. Dan ditampilkan pada tombol yang sama saat memilih level.

<pre>//Tampilan Nilai Tertinggi void SaveToFile() { IF SaveScore.Level[i] < ScoreNow.Level[i] THEN SaveScore.Level[i] <- ScoreNow.Level[i]; } // LevelManager.cs newButton ("ScoreText").GetComponent<Text>().text = SaveScore.Level[i].ToString();</pre>

Gambar 4.4 Pseudocode Menampilkan Nilai Tertinggi

4.2.5 Implementasi Menembak Jawaban dan Musuh

Implementasi menembak musuh akan diperlihatkan pada Gambar 4.5, dimana pada gambar tersebut ditampilkan fungsi *spawn*, dimana berfungsi untuk memunculkan jumlah musuh sesuai dengan perhitungan yang dilakukan untuk menentukan bobot faktor tantangan tiap level. Setelah musuh dimunculkan musuh akan mengejar karakter utama, dengan cara mencari tau terlebih dahulu posisi karakter utama, jangkauan untuk mengejar serta karakter utama masih memiliki nyawa. Untuk implementasi dimana musuh mengejar karakter utama, penulis menggunakan *library* dari A* Pathfinding Project 3.6 serta dapat diunduh pada <http://www.arongranberg.com/astar>.

Pada Gambar 4.6 apabila pemain akan menembak musuh, terlebih dahulu sistem akan melakukan pengecekan terhadap nyawa musuh, sehingga apabila nyawa musuh masih ada atau musuh masih mengejar maka pemain dapat melakukan tembakan untuk mengalahkan musuh dan menambah *score*. Musuh yang telah

tertembak oleh pemain dan mengalami konsisi kalah akan menghilang dari halaman permainan.

```
//memunculkan musuh
void Spawn ()
{
    if (enemyCount < maxEnemy)
    {
        spawnEnemy;
        enemyCount++;
    }
}
```

Gambar 4.5 Pseudocode Mengeluarkan Musuh

```
// menembak musuh //
public void TakeShoot (int amount, Vector3 hitPoint)
{
    IF musuh == exist
    THEN
        IF tembakan > 5x
        THEN
            Score += 10;
            Destroy Objek;
}
```

Gambar 4.6 Pseudocode Menembak Musuh

Pada permainan ini, cara menjawab pertanyaan adalah dengan cara menembak sebanyak empat kali pada jawaban yang dirasa sesuai oleh pemain, sehingga diberikan fungsi untuk menembak pilihan jawaban yang diberikan. Apabila, jawaban yang ditembak oleh pemain salah, maka *score* pemain akan berkurang sebanyak 20 dan nyawa karakter utama juga akan berkurang sebanyak 10. Namun jika pemain menembak pada jawaban yang sesuai dengan soal, maka nilai pemain akan bertambah sebanyak 30, kemudian seluruh pilihan jawaban akan dihilangkan untuk diganti dengan soal baru. Kode sumber ini ditampilkan pada Gambar 4.7.

```
// menembak jawaban //

public void TakeShoot (int amount, Vector3 hitPoint)
{
    IF objek tembak == jawaban
    THEN
        IF tembakan > 3x
        THEN
            Score += 30;
            Foreach gameobject in pilihanJawaban
                Destroy;
        ELSE
        THEN
            IF tembakan > 2x
            THEN
                Score -= 20;
                NyawaKarakterUtama -= 10;
                Foreach gameobject in pilihanJawaban
                    Destroy;
            }
}
```

Gambar 4.7 Pseudocode Menembak Jawaban atau menjawab Soal

4.2.6 Implementasi Menggerakkan Karakter

Beberapa fungsi yang terdapat pada Gambar 4.8 merupakan fungsi-fungsi yang digunakan untuk menggerakkan karakter utama selama permainan berlangsung. Fungsi untuk pengaturan sensor *accelerometer* diawali dengan mengatur *input* derajat kemiringan dimana pengaturan *input* yang lebih jauh telah disediakan oleh Unity, lalu mengatur derajat kemiringan perangkat keras agar dalam menggerakkan karakter utama tidak terlalu besar dan tidak terlalu kecil derajat kemiringannya.

Selanjutnya untuk *Virtual Joystick*, pada penggunaan control *Virtual Joystick* ini menggunakan pengembangan dari *Virtual Joystick CNControls* sehingga pengaturan control ini hanya dilakukan pada beberapa bagian tertentu yang dirasa dibutuhkan untuk menunjang jalannya *game* yang dibangun.

```

// Fungsi pergerakan karakter utama//
void FixedUpdate()
{
    h = Input.acceleration.x;
    v = Input.acceleration.y;
    Move (h, v);
    Animating (h, v);
}

// sensor accelerometer
void Move(float h, float v)
{
    if (Mathf.Abs(h)>=0.02f || Mathf.Abs(v)>=0.02f)
    {
        if ((h >= 0.20f) || (h <= -0.20f) || (v >= 0.20f) ||
            (v <= -0.20f))
        {
            Karakter utama bergerak dg derajat kemiringan
            maksimum h dan v;
        }
        else
        {
            Karakter utama bergerak;}}
}

// virtual joystick
void Start()
{
    _playerTransform <- "Player";
}

// joystick rotate karakter utama
void Update()
{
    var movement <- ("Horizontal",0f, "Vertical");
}
private void CommonMovementMethod(Vector3 movement)
{
    arah muka karakter utama <- movement;
}

// joystick menembak
IF joystick.touchpad == touch
THEN
    protected override void TweakControl(Vector2
        touchPosition ){

```

<pre>Tembak = true;}</pre>
<pre>// playerShooting.cs</pre>
<pre>IF tembak == true THEN shoot ();</pre>

Gambar 4.8 Pseudocode Menggerakkan Karakter

4.2.7 Implementasi Alur Permainan

Implementasi permainan yang akan diperlihatkan pada Gambar 4.9, adalah seputar pembobotan faktor level, tantangan bermain, pembobotan perlevel dan berakhirnya permainan atau *game over*. Pembobotan diawali dengan menentukan level mana yang memiliki bobot tantangan meningkat dan level mana yang memiliki bobot tantangan yang menurun, digunakan fungsi *random* yang bertujuan untuk menghasilkan nilai yang lebih beragam.

Pada Gambar 4.10 merupakan fungsi untuk penyesuaian bobot faktor tantangan terhadap level yang terpilih dan disesuaikan dengan rumus yang telah dijelaskan pada subbab perancangan alur permainan. Selanjutnya, pada Gambar 4.11 merupakan pengaturan yang menunjukkan permainan berakhir apabila waktu telah habis atau nyawa karakter utama telah habis, sehingga ketika permainan telah berakhir, sistem akan melakukan menyimpan nilai terakhir dan disiapkan untuk membandingkan dengan nilai yang tersimpan sebelumnya, kemudian menjalankan animasi permainan berakhir dan mengarahkan pemain kembali ke halaman pemilihan level.

<pre>// Fungsi pembobotan level // public void Bobot(int level) { IF(level%8 == 0 level%5 == 0) bobotSekarang = bobotLevelNow - randBobot; ELSE bobotSekarang = bobotLevelNow + randBobot; }</pre>

Gambar 4.9 Pseudocode Pembobotan Level

```

// Fungsi penentuan bobot faktor tantangan
IF level > 25
{
do {
faktorAngkaSoal = (((level / totalLevel) *
(Angka.max - Angka.min)) + Angka.min, Angka.max +
1);
faktorJawaban = (((level / totalLevel) *
(Jawaban.max - Jawaban.min)) + Jawaban.min,
Jawaban.max + 1);
faktorJumlahMusuh = (jumMusuh.min, jumMusuh.max +
1);
faktorKecepatanMusuh = (((level / totalLevel) *
(kecMusuh.max - kecMusuh.min)) + kecMusuh.min,
kecMusuh.max + 1);
faktorDimensiX = (((level / totalLevel) * (X.max -
X.min)) + X.min, X.max + 1);
faktorDimensiY = (((level / totalLevel) * (Y.max -
Y.min)) + Y.min, Y.max + 1);
}
while( bobotSekarang - RumusPerhitungan()) > cek;
}
ELSE {
do {
faktorAngkaSoal = (Angka.min, (Angka.max + 1)/2);
faktorJawaban = (Jawaban.min, (Jawaban.max +
1)/2);
faktorJumlahMusuh = (jumMusuh.min, (jumMusuh.max +
1)/2);
faktorKecepatanMusuh = (((level / totalLevel) *
(kecMusuh.max - kecMusuh.min)) + kecMusuh.min,
kecMusuh.max + 1);
faktorDimensiX = (((level / totalLevel) * (X.max -
X.min)) + X.min, (X.max + 1)/2);
faktorDimensiY = (((level / totalLevel) * (Y.max -
Y.min)) + Y.min, Y.max + 1);
}
while( Mathf.Abs( bobotSekarang - RumusPerhitungan())
> cek);
}

```

Gambar 4.10 Pseudocode Pembobotan Faktor perLevel

```
// score //

void Update()
{
    if (playerHealth.currentHealth <= 0)
    {
        SaveToFile();
        GameOver();
        if (restartTimer >= restartDelay)
        {
            Application.LoadLevel ("1_level");
        }
    }
    else if (timer == 1)
    {
        SaveToFile();
        GameOver();
        if (restartTimer >= restartDelay)
        {
            Application.LoadLevel ("1_level");
        }
    }
}}
```

Gambar 4.11 Pseudocode Implementasi Permainan Berakhir

4.2.8 Implementasi Penyimpanan Data

Fitur ini digunakan untuk menyimpan data nilai tertinggi setiap level dan level terakhir yang telah dibuka oleh pemain, Implementasi fitur ini ditampilkan pada Gambar 4.12. Sehingga penyimpanan data hanya akan dijalankan jika didapatkan nilai yang lebih tinggi dari nilai sebelumnya dan di level yang sama.

```
// fungsi penyimpanan file //

void SaveToFile()
{
    using (System.IO.StreamWriter )
    {
        FOR (int i = 0; i < SaveScore.Length; i++)
        {
            IF (i == levelNow)
            {
                IF (SaveScore[i] < ScoreManager.score[i])
            }
        }
    }
}
```

```

        {
            SaveScore[i] = ScoreManager.score;
        }
    }
    file.WriteLine(LevelManager.lines[i]);
}
}
}

```

Gambar 4.12 Potongan Kode Sumber Penyimpanan Nilai Tertinggi di Setiap level

Fungsi 'loadData' pada Gambar 4.13 adalah mengambil data-data dari file penyimpanan untuk dimuat ke dalam data sebelum permainan selanjutnya dimulai seperti, level terakhir yang terbuka serta nilai tertinggi yang tersimpan akan dimunculkan halaman pemilihan level.

```

// function for load file/

void CreateLevelButton ()
{
    try
    {
        string[] temp =
            System.IO.File.ReadAllLines("\\Score.txt");
        Using System.IO.StreamWriter("\\Score.txt")
        {
            for (int a=0; a<temp.Length; a++)
            {
                lines [a] = temp [a];
            }
            for(int a=0; a<lines.Length; a++)
            {
                file.WriteLine(lines[a]);
            }
        }
    }
}
}

```

Gambar 4.13 Potongan Kode Sumber Pengambilan Data

4.2.9 Tampilan Antarmuka Pengguna

Subbab ini membahas bagaimana implementasi antarmuka pengguna yang telah dibangun pada tugas akhir ini. Implementasi tampilan antarmuka pengguna meliputi pilihan utama seperti **Menu**, **Level** dan **Nilai Tertinggi**, **How To**, dan **Halaman Permainan**

4.2.9.1 Tampilan Menu

Tampilan awal menu merupakan tampilan yang pertama kali muncul ketika aplikasi dijalankan untuk yang pertama kalinya. Seperti yang ditampilkan pada Gambar 4.14. Tampilan ini menampilkan pilihan 'PLAY' yang akan mengarahkan ke halaman pemilihan level, 'HOW TO' yang akan mengarahkan ke halaman instruksi dan 'QUIT' yang akan mengarahkan pemain ke halaman konfirmasi untuk keluar dari permainan.

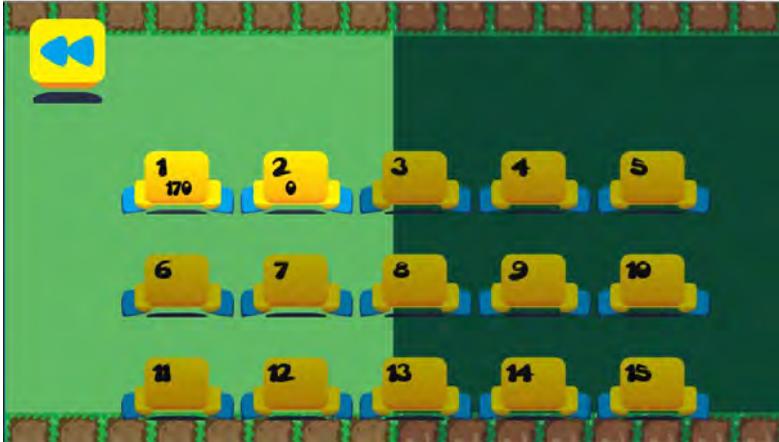


Gambar 4.14 Implementasi Tampilan Menu Utama

4.2.9.2 Tampilan Level dan Nilai Tertinggi

Implementasi layar Tampilan Level dan Nilai Tertinggi ditampilkan secara bersamaan ketika pemain memilih tombol

‘PLAY’ pada layar menu Gambar 4.15 sehingga, ketika pemain memilih level yang tersedia, pemain langsung diarahkan ke halaman permainan berdasarkan level yang telah dipilih



Gambar 4.15 Implementasi Tampilan Level dan Nilai Tertinggi

4.2.9.3 Tampilan How To

Tampilan ‘HOW TO’ merupakan tampilan yang ditampilkan ketika pemain memilih tombol ‘HOW TO’ pada halaman awal permainan. Pada halaman ‘HOW TO’ berfungsi untuk melihat deskripsi permainan dan cara bermain, tombol yang terdapat pada sisi kanan atas layar merupakan tombol yang berfungsi untuk mengarahkan ke halaman petunjuk selanjutnya, hingga seluruh halaman instruksi selesai ditampilkan. Setelah pemain selesai memahami cara bermain yang diberikan, pemain akan langsung diarahkan ke halaman level, untuk memilih bobot permainan sesuai dengan level yang diinginkan. Tampilan ini ditunjukkan pada Gambar 4.16.



Gambar 4.16 Implementasi Tampilan How-To

4.2.9.4 Tampilan Permainan



Gambar 4.17 Implementasi Tampilan Halaman Permainan

Tampilan ini merupakan menu penting. Dari tampilan ini pengguna telah memilih bobot permainan berdasarkan level yang dipilih. Pada tampilan ini telah memberikan bentuk labirin, soal

yang terletak pada kanan atas layar, pilihan jawaban yang terdapat pada jalanan labirin, batas waktu yang terdapat pada kiri atas layar, penilaian pada bagian tengah atas layar, jumlah musuh dan kecepatan musuh yang sesuai dengan bobot pada level tersebut berdasarkan perancangan yang telah dilakukan sebelumnya. Tampilan ini ditunjukkan pada Gambar 4.17.

4.2.9.5 Tampilan Kontrol Permainan



Gambar 4.18 Implementasi Tampilan Kontrol Permainan

Pada Gambar 4.18 menunjukkan kontrol yang terdapat pada permainan yaitu telah disediakan *Virtual JoyStick*. Pada *virtual joystick* sisi kiri pengguna merupakan control untuk mengatur arah hadap karakter utama, sedangkan pada sisi kanan pengguna terdapat fitur untuk melakukan tembakan oleh karakter utama selama permainan berlangsung.

BAB V

PENGUJIAN DAN EVALUASI

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

5.1 Lingkungan Uji Coba

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

Tabel 5.1 Lingkungan Ujicoba Perangkat Lunak (bagian 2).

Perangkat Keras	Alat : <i>Smarthphone</i> Memori: 2 GB
Perangkat Lunak	Sistem Operasi: Android Perangkat Pengembang: Unity

5.2 Pengujian Aturan Main

Pengujian dilakukan untuk menguji apakah aturan main yang diberikan berjalan dengan baik atau tidak, terutama penentuan kondisi menang dan kalah

5.2.1 Skenario Uji Aturan Main.

Skenario Pengujian aturan main digunakan untuk memberikan tahap-tahap dalam pengujian sistem dalam hal aturan main yang telah diberikan. Skenario ini tertera pada Tabel 5.2.

Tabel 5.2 Skenario Pengujian Aturan Main

Deskripsi	Bertujuan untuk mengetahui respon sistem dalam menjalankan aturan main yang diberikan
Langkah pengujian	<ol style="list-style-type: none"> 1. Memilih salah satu level permainan. 2. Melakukan pengecekan terhadap faktor tantangan disetiap level. 3. Melakukan pengecekan adanya faktor tantangan mempengaruhi kondisi menang atau kondisi kalah karakter utama. 4. Mengulangi langkah 1,2,3.

Pengujian dilakukan dengan cara memilih satu level yang sama dengan percobaan beberapa kali. Faktor bobot tantangan akan dicatat disetiap pengujian dan respon sistem terhadap faktor bobot tantangan mempengaruhi kondisi menang atau kondisi kalah karakter utama.

5.2.2 Hasil Pengujian Aturan Main

Hasil pengujian performa dikhususkan untuk menghitung faktor bobot yang dibutuhkan dalam mendukung aturan main yang telah diberikan, dimana ditampilkan pada Tabel 5.3 semua nilai pada tabel dihitung dalam satuan, sedangkan untuk melihat respon sistem terhadap aturan main yang diberikan dapat dilihat pada Gambar 5.1, Gambar 5.2 dan Gambar 5.3 terlihat bahwa dengan parameter yang sama atau level yang sama, sistem akan memberikan faktor tantangan yang sesuai sehingga hasil pengujian aturan main dapat dikatakan berhasil.

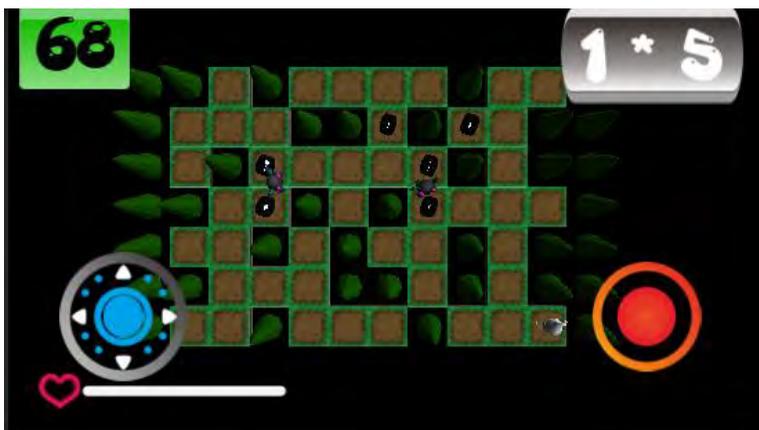
Tabel 5.3 Bobot Faktor Tantangan Percobaan I

Level	Angka Soal	Pilihan Jawaban	Jumlah Musuh	Speed Musuh	P	L	Bobot Level
1	2	5	2	3	5	10	2.5
2	4	5	2	3	6	12	5.785714286
3	5	5	4	3	6	10	9.428571429
4	8	5	3	3	5	10	9.732142857
5	3	5	4	3	5	11	7.642857143
6	2	5	4	4	7	10	13.25
7	6	5	3	4	5	12	14.94642857
8	4	6	3	4	6	10	13.83718487
9	14	6	4	3	5	15	20.6407563
10	2	6	5	4	7	13	17.80147059
11	2	8	2	6	6	10	24.52941176
12	14	9	4	3	7	15	25.17016807
13	11	7	4	5	5	9	28.13865546
14	10	6	4	5	7	13	29.06932773
15	5	7	3	6	5	7	25.90651261
16	2	9	2	7	6	8	30.70588235
17	3	10	4	7	7	6	36.02521008
18	12	10	2	6	6	15	38.31092437
19	10	6	3	7	6	15	39.69432773
20	8	10	4	7	7	5	39.9894958
21	11	8	5	7	5	6	41.69012605
22	8	7	4	8	7	10	44.96008403
23	11	10	3	8	6	6	46.79306723
24	13	6	2	8	5	12	44.49789916
25	15	7	2	8	7	14	49.46008403

Level	Angka Soal	Pilihan Jawaban	Jumlah Musuh	Speed Musuh	P	L	Bobot Level
26	13	7	5	8	7	15	53.79936975
27	7	10	7	8	10	13	56.22163866
28	10	11	10	7	9	7	56.20168067
29	20	15	7	6	10	6	58.21113445
30	19	15	3	7	13	6	57.31827731
31	13	21	3	7	15	15	64.51995798
32	2	21	8	8	10	6	63.07352941
33	10	13	10	8	8	14	67.55462185
34	17	19	9	7	7	10	70.48844538
35	9	22	5	8	15	5	66.875
36	30	8	8	8	7	12	74.27941176
37	29	12	7	8	6	14	76.71743697
38	26	7	10	8	15	12	77.28151261
39	28	18	7	8	6	12	81.88340336
40	24	21	6	8	11	5	78.96638655
41	27	22	10	7	7	7	83.32142857
42	28	19	10	8	9	7	87.68487395
43	26	20	9	8	9	12	87.70063025
44	30	20	7	8	13	13	90.02205882
45	24	22	8	8	12	10	86.89285714
46	28	21	10	8	15	7	93.03781513
47	26	22	10	8	14	11	93.92857143
48	27	22	10	8	15	9	94.32142857
49	30	22	10	8	14	14	99
50	30	22	9	8	15	15	98.125



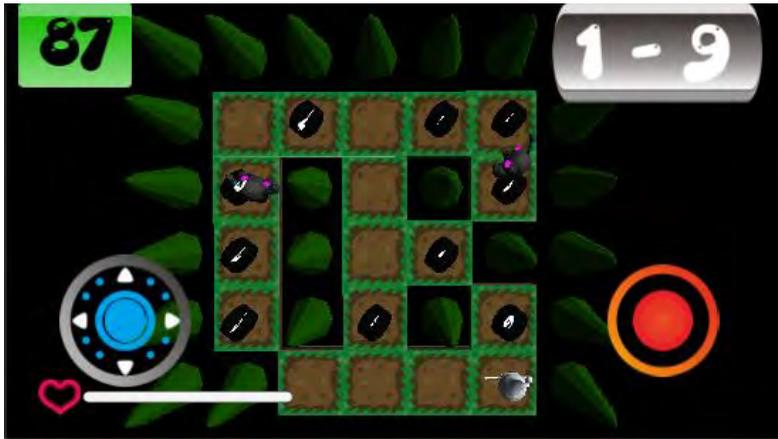
Gambar 5.1 Tampilan Level 1 (Percobaan 1)



Gambar 5.2 Tampilan Level 1 (Percobaan 2)

Pengujian kondisi menang dan kondisi kalah, pengujian pertama diawali dengan pemain pertama kali memainkan permainan pada level 1 dan kondisi level dua masih terkunci, pemain memainkan *game* hingga mendapatkan nilai 60 namun karena adanya serangan dari musuh mengenai pemain secara terus menerus, maka nyawa karakter berkurang sehingga karakter utama

tidak memiliki nyawa seperti yang ditunjukkan pada Gambar 5.4 maka pada kondisi ini pemain dinyatakan kalah



Gambar 5.3 Tampilan Level 1 (Percobaan 3)



Gambar 5.4 Karakter Utama Kondisi Kalah

Pengujian kedua diawali dengan pertama kali memainkan permainan pada level 1 dan kondisi level dua masih terkunci, pemain memainkan *game* hingga mendapatkan nilai 140, masih

5.3 Pengujian Fungsionalitas dengan metode *black-box*

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

5.3.1 Skenario Pengujian Fungsionalitas

Skenario Pengujian fungsionalitas digunakan untuk memberikan tahap-tahap dalam pengujian sistem. Skenario ini tertera pada Tabel 5.4.

Tabel 5.4 Pengujian Permainan

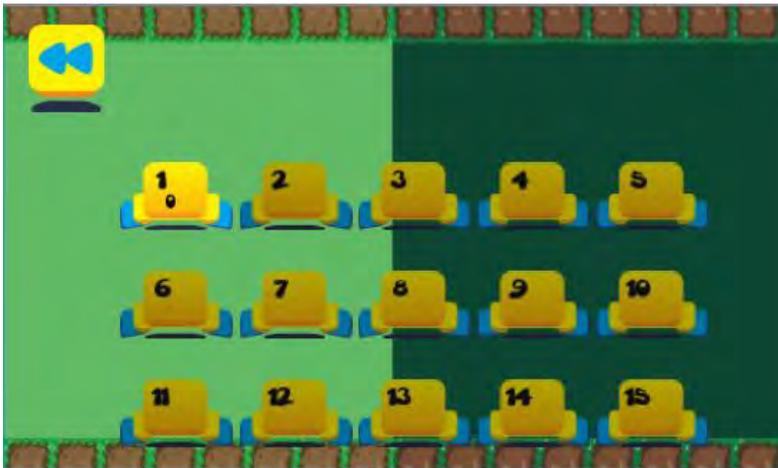
	UFG01
Kondisi Awal	Pengguna berada pada layar Menu Utama Permainan
Prosedur Pengujian	Pengguna memainkan permainan hingga selesai dari beberapa fungsionalitas yang ingin diuji
Hasil yang diharapkan	Pengguna berhasil menyelesaikan permainan dan fungsionalitas permainan berjalan dengan lancar.
Hasil yang diperoleh	Pengguna berhasil menyelesaikan permainan dan fungsionalitas berjalan lancar.
Kesimpulan.	Pengujian berhasil.

5.3.1.1 PF01: Pengujian Layar Menu Utama

Pengujian dimulai ketika pemain telah masuk ke permainan dan Sistem akan menampilkan tiga menu pilihan, yaitu 'PLAY' 'HOW TO' 'QUIT' seperti pada Gambar 5.7



Gambar 5.7 Tampilan Menu Utama



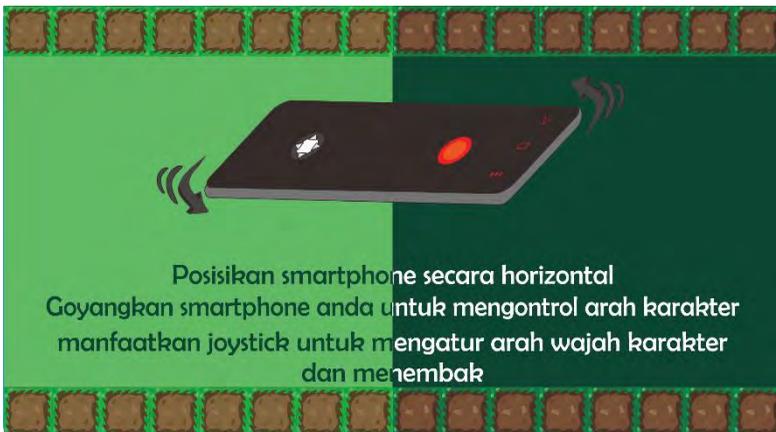
Gambar 5.8 Tampilan layar Pemilihan Level

Setelah menekan tombol 'PLAY' maka sistem menampilkan layar Pemilihan Level seperti yang dicantumkan pada Gambar 5.8.

Setelah layar muncul, pemain diminta untuk memilih level yang tersedia. Apabila level yang terpilih tersedia dan siap untuk dimainkan, maka sistem akan mengarahkan pemain untuk menuju ke Halaman Permainan. Seperti yang ditunjukkan pada Gambar 5.9. Jika pemain memilih level yang tidak tersedia atau terkunci, sistem tidak akan merespon permintaan pemain.

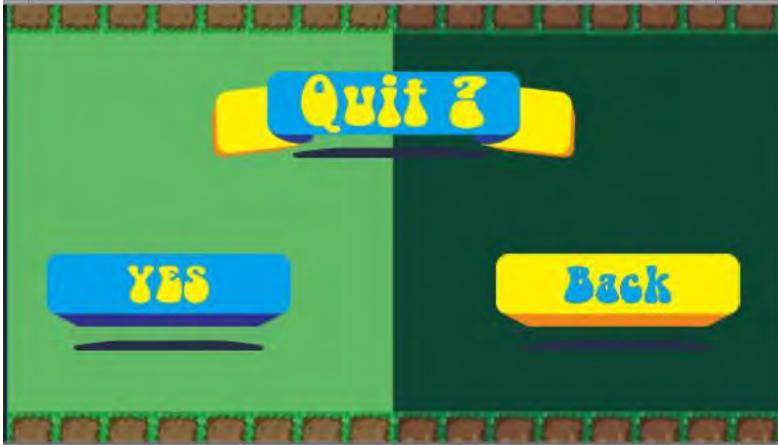


Gambar 5.9 Tampilan Halaman Permainan



Gambar 5.10 Tampilan Layar 'How To'

Sedangkan, apabila pemain memilih tombol ‘HOW TO’ pada Menu Utama atau yang ditunjukkan pada Gambar 5.10 maka pemain akan diarahkan untuk melihat cara bermain, seperti yang ditunjukkan pada



Gambar 5.11 Tampilan Menu Keluar Permainan

Pada halaman terakhir, ketika pemain menekan tombol selanjutnya. Yang terdapat pada sisi kanan atas layar, pemain akan otomatis diarahkan menuju halaman untuk memilih level, seperti yang ditunjukkan pada Gambar 5.9.

Untuk selanjutnya, apabila pada menu utama pemain memilih tombol ‘QUIT’ maka sistem akan mengarahkan ke halaman untuk menanyakan kepada pemain apakah ingin keluar dari permainan atau tidak. Seperti yang ditunjukkan Gambar 5.11.

5.3.1.2 PF02: Pengujian Layar Menu Permainan

Skenario pengujian dimulai ketika pemain sudah masuk ke Menu Permainan seperti yang tercantum pada Gambar 5.12 Pada layar ini terdapat empat informasi permainan, seperti : informasi waktu bermain yang terdapat pada sisi kiri atas layar permainan dimana waktu akan dimulai dari detik ke-100 dan akan terus berkurang hingga waktu menunjukkan detik ke-0 sehingga sistem

akan memberikan efek untuk menandakan permainan berakhir, informasi nyawa karakter utama yang terdapat pada sisi kiri bawah permainan dimana diawali dengan garis yang penuh namun akan berkurang ketika pemain memilih pilihan jawaban yang salah atau karakter utama diserang oleh musuh, informasi nilai atau *score* yang terdapat pada sisi atas bagian tengah layar permainan yang diawali dengan nilai 0 dan akan bertambah apabila pemain mampu membunuh musuh atau memilih pilihan jawaban dengan tepat dan informasi soal yang terdapat pada sisi kanan atas layar yang mampu berubah jika pemain telah menjawab soal dengan benar.

Selanjutnya, pilihan jawaban akan terdapat pada jalanan labirin, dan pilihan jawaban tersebut memungkinkan untuk ditembak, lalu musuh yang datang juga akan selalu berada di jalanan labirin dan selalu mengejar karakter utama dan juga memungkinkan untuk ditembak dan dikalahkan. Karakter utama yang akan selalu berada di jalanan labirin dan memiliki kemampuan untuk bergerak dan menembak.



Gambar 5.12 Tampilan Menu Permainan

Kesimpulan untuk fungsionalitas pada **Menu Permainan** adalah berhasil dikarenakan seluruh informasi seperti informasi

waktu bermain, informasi nilai atau *score*, informasi nyawa karakter, informasi soal, penempatan dan fungsi dari pilihan jawaban, musuh dan karakter utama sudah diuji dan hasilnya sesuai dengan tujuan yang telah disebutkan sebelumnya.

5.3.1.3 PF03: Pengujian Penyimpanan Data Lokal

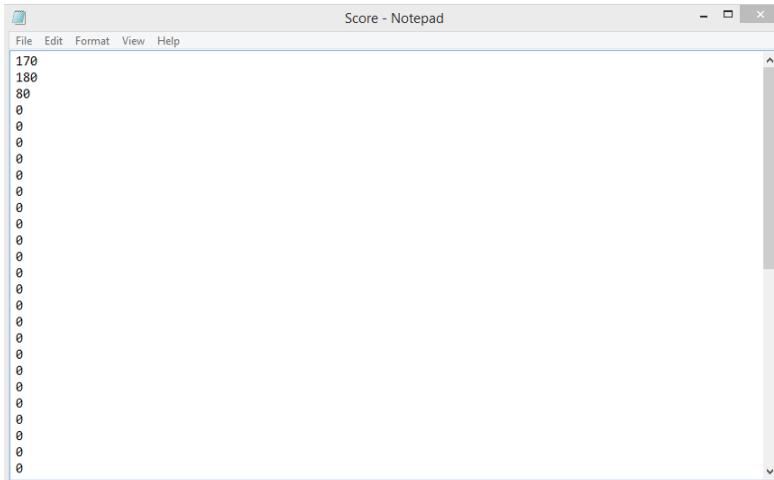
Pengujian ini dilakukan untuk uji coba penyimpanan data nilai dan level terakhir yang telah dimainkan oleh pemain. Uji coba ini digunakan untuk menyimpan data agar sistem hanya menampilkan nilai tertinggi disetiap level dan level terakhir yang terbuka. Penyimpanan ini menggunakan *persitent data path* pada Unity.

Hasil pengujian yang diharapkan adalah, pada lingkungan pengujian I file penyimpanan dapat dilihat dalam bentuk .txt, dan data yang tersimpan adalah nilai tertinggi disetiap level yang sesuai.



Gambar 5.13 Tampilan berkas penyimpanan data

Dapat dilihat pada Gambar 5.13 bahwa pada pengujian penyimpanan data, data telah tersimpan dengan format .txt pada lingkungan pengujian I, dan pada Gambar 5.14 dapat dilihat, nilai yang tersimpan adalah hanya nilai tertinggi dan urutan penyimpanan sesuai dengan urutan level yang telah dimainkan dan sesuai dengan nilai tertinggi yang telah dicapai.



Gambar 5.14 Tampilan Isi Penyimpanan Data

Dengan ini pengujian untuk penyimpanan data-data telah diuji dengan baik dan dinyatakan berhasil karena telah sesuai dengan hasil yang disebutkan sebelumnya.

5.3.1.4 Hasil Pengujian Fungsional

Hasil uji fungsionalitas yang sudah dilakukan berdasarkan pada PF01, PF02, PF03 dan PF04. Empat uji coba yang telah dilakukan menunjukkan bahwa semua fungsionalitas permainan berjalan dengan baik dan sesuai dengan sebagaimana mestinya skenario dan alur yang telah dibuat pada perancangan. Rekap hasil pengujian fungsionalitas dicantumkan pada Tabel 5.5

Tabel 5.5 Hasil Pengujian Fungsionalitas

No	Kode Pengujian	Hasil Pengujian
1	PF01	Berhasil
2	PF02	Berhasil
3	PF03	Berhasil

5.4 Pengujian Pengguna

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

5.4.1 Skenario Uji Coba Pengguna

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

Jumlah pengguna yang terlibat dalam pengujian perangkat lunak sebanyak dua puluh orang. Dalam melakukan pengujian, pengguna melakukan percobaan lebih dari satu kali penggunaan untuk masing-masing pengguna. Dalam memberikan penilaian dan tanggapan, penguji diberikan formulir pengujian perangkat lunak. Formulir pengujian perangkat lunak ini memiliki beberapa aspek penilaian dan pada bagian akhir terdapat saran untuk perbaikan fitur.

5.4.2 Daftar Penguji Perangkat Lunak

Pada subbab ini ditunjukkan daftar pengguna yang bertindak sebagai penguji coba aplikasi yang dibangun. Daftar nama penguji aplikasi ini dapat dilihat pada Tabel 5.6.

Tabel 5.6 Daftar Nama Penguji Coba Aplikasi

No	Nama	Pekerjaan
1.	Mahendra Harsa Wardhana	Mahasiswa Teknik Informatika
2.	Rizaldi Tri Yanuar	Mahasiswa Teknik Informatika
3.	Yoga Pratama Aliarham	Mahasiswa Teknik Informatika
4.	Mohammad Aprialdi Rizky Pratama	Mahasiswa Teknik Informatika
5.	Andrie Prasetyo Utomo	Mahasiswa Teknik Informatika

5.4.3 Hasil Uji Coba Pengguna

Uji coba yang dilakukan terhadap beberapa pengguna memiliki beberapa aspek yang dipisahkan berdasarkan antarmuka dan fungsionalitas yang dimiliki. Sistem penilaian didasarkan pada skala penghitungan satu sampai lima di mana skala satu menunjukkan nilai terendah dan skala lima 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 pengguna yang dipaparkan dapat dilihat pada bab LAMPIRAN HASIL KUESIONER.

5.4.3.1 Hasil Penilaian Permainan dan Pengujian Pengguna

Penilaian Permainan merupakan hasil uji coba pemain setelah memainkan mulai dari level pertama hingga level kelima. Hasil penilaian pengguna terhadap antarmuka aplikasi dapat dilihat pada Tabel 5.7.

Tabel 5.7 Penilaian Permainan (Pengujian I)

No.	Nama Pengujian	Responden					Rata-Rata
		1	2	3	4	5	
1.	Keindahan Tampilan	4	4	3	4	4	3.8
2.	Kesesuaian tema	3	4	3	4	4	3.6

3.	Ketertarikan Bermain	3	4	3	4	4	3.6
4.	Kemudahan Materi Berhitung	5	5	5	5	5	5
5.	Kemudahan Menyelesaikan Level	3	3	4	3	3	3.2
6.	Kemudahan Memenangkan Permainan	3	4	5	4	5	4.2
7.	Kesesuaian dimensi labirin dengan kesulitan level yang dipilih	4	4	3	4	4	3.8
8.	Kemudahan Menggunakan Sensor <i>Accelerometer</i>	2	2	4	4	4	3.2
9.	Kemudahan Menggunakan <i>Joystick</i>	3	3	2	4	5	3.4
							3.8

Dari data diketahui bahwa aplikasi telah memenuhi unsur yang seharusnya seperti perancangan dimana nilai rata-rata keseluruhan hasil dari responden memiliki nilai 3,8 dari jenjang skala yang diberikan antara satu hingga lima, dimana satu bernilai pemain sangat kurang setuju dan lima bernilai sangat setuju, sedangkan nilai rata-rata yang dibulatkan menjadi nilai empat bernilai cukup baik. Permainan yang dibuat telah mendapatkan

apresiasi oleh pengguna sebenarnya. Sehingga pula dapat disimpulkan bahwa pengguna secara tidak langsung cukup menyetujui dan memberikan komentar baik kepada aplikasi permainan yang dibuat.

BAB VI

KESIMPULAN DAN SARAN

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

6.1. Kesimpulan

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

1. Berdasarkan pengujian aturan main yang diberikan, *game* memberikan respon pada pemain berupa kondisi menang apabila pemain mampu menjawab pertanyaan serta mendapatkan *score* lebih dari seratus sebelum waktu habis atau nyawa dari karakter utama habis, serta *game* memberikan respon pada pemain berupa kondisi kalah apabila pemain tidak mampu menjawab soal dan mengumpulkan *score* lebih dari seratus seperti yang telah dijelaskan pada subbab pengujian aturan main.
2. Dalam pengujian bobot faktor tantangan, *game* yang dibangun berhasil menerapkan seluruh faktor bobot tantangan, dimana faktor bobot selalu meningkat sesuai dengan tingkat level yang dipilih, namun pada level tertentu menurun untuk mencegah rasa bosan yang akan dialami oleh pemain.
3. Algoritma Prims dalam membangun pembangkit labirin dinamis dapat dikatakan berhasil, terbukti mampu menyusun labirin dengan pola yang berbeda-beda namun tetap mengikuti pembobotan yang berlaku meskipun pengujian dilakukan dengan parameter atau level yang sama.
4. Berdasarkan uji coba pengguna melalui kuesioner yang didapatkan menunjukkan bahwa pemain mampu

memenangkan permainan dengan bentuk labirin yang beragam.

5. Berdasarkan uji coba, sensor *accelerometer* telah berhasil digunakan pada *game* yang dibangun untuk menggerakkan karakter utama berdasarkan derajat kemiringan.
6. Berdasarkan uji coba pengguna melalui kuesioner, pemain setuju bahwa sensor *accelerometer* dapat membantu dalam melakukan kontrol terhadap karakter utama, sehingga dapat disimpulkan bahwa pemanfaatan sensor *accelerometer* telah diimplementasikan dengan baik, namun para pemain kurang terbiasa dengan cara bermain *game* yang telah dibangun.

6.2. Saran

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

1. Dimensi labirin yang luas dapat dimanfaatkan agar permainan dapat melakukan *multiplayer*.
2. Sensor *accelerometer* yang disisipkan menjadi bantuan kontrol dalam permainan dengan jenis *Third Person Shooter (TPS)* masih kurang menarik dikarenakan pemain yang kurang terbiasa, oleh karenanya diperlukan pendekatan lebih jauh terhadap menentukan bantuan kontrol dalam permainan dengan jenis *Third Person Shooter (TPS)*

DAFTAR PUSTAKA

- [1] F. H. M. Ichwan, "PENGUKURAN KINERJA GOODREADS APPLICATION PROGRAMMING," *JURNAL INFORMATIKA*, vol. 2, no. 2, pp. 13-21, May - August 2011.
- [2] Rolliawati, "Game Engine," Universitas Narotama, 13 September 2013. [Online]. Available: <http://rolliawati.dosen.narotama.ac.id/2013/09/13/game-engine/>. [Accessed 24 December 2014].
- [3] M. Habibie, "HabibieSystem," 21 April 2012. [Online]. Available: <http://bloghabibie.blogspot.com/2012/04/unity3d-cross-platform-game-engine.html>. [Accessed 2015 January 6].
- [4] GFAGAME, "GooglePlay," 20 June 2014. [Online]. Available: <https://play.google.com/store/apps/details?id=com.gfagame.labyrinth>. [Accessed 25 June 2015].
- [5] SENSR, "Practical Guide to Accelerometers," [Online]. Available: www.sensr.com. [Accessed 20 December 2014].
- [6] Hadiyanto and M. Alhan, "Penentuan Pohon Rentang Minimum," July 2012. [Online]. Available: <http://download.portalgaruda.org/article.php?article=256905&val=6943&title=PENENTUAN%20ALGORITMA%20PRIM>. [Accessed 4 June 2015].
- [7] D. W. Nugraha, "APLIKASI ALGORITMA PRIM UNTUK MENENTUKAN MINIMUM SPANNING TREE SUATU GRAF BERBOBOT," *Jurnal Ilmiah Foristek*, vol. Vol 1, no. 2, pp. 70-79, 2011.

BAB VI

KESIMPULAN DAN SARAN

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

6.1. Kesimpulan

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

1. Berdasarkan pengujian aturan main yang diberikan, *game* memberikan respon pada pemain berupa kondisi menang apabila pemain mampu menjawab pertanyaan serta mendapatkan *score* lebih dari seratus sebelum waktu habis atau nyawa dari karakter utama habis, serta *game* memberikan respon pada pemain berupa kondisi kalah apabila pemain tidak mampu menjawab soal dan mengumpulkan *score* lebih dari seratus seperti yang telah dijelaskan pada subbab pengujian aturan main.
2. Dalam pengujian bobot faktor tantangan, *game* yang dibangun berhasil menerapkan seluruh faktor bobot tantangan, dimana faktor bobot selalu meningkat sesuai dengan tingkat level yang dipilih, namun pada level tertentu menurun untuk mencegah rasa bosan yang akan dialami oleh pemain.
3. Algoritma Prims dalam membangun pembangkit labirin dinamis dapat dikatakan berhasil, terbukti mampu menyusun labirin dengan pola yang berbeda-beda namun tetap mengikuti pembobotan yang berlaku meskipun pengujian dilakukan dengan parameter atau level yang sama.
4. Berdasarkan uji coba pengguna melalui kuesioner yang didapatkan menunjukkan bahwa pemain mampu

memenangkan permainan dengan bentuk labirin yang beragam.

5. Berdasarkan uji coba, sensor *accelerometer* telah berhasil digunakan pada *game* yang dibangun untuk menggerakkan karakter utama berdasarkan derajat kemiringan.
6. Berdasarkan uji coba pengguna melalui kuesioner, pemain setuju bahwa sensor *accelerometer* dapat membantu dalam melakukan kontrol terhadap karakter utama, sehingga dapat disimpulkan bahwa pemanfaatan sensor *accelerometer* telah diimplementasikan dengan baik, namun para pemain kurang terbiasa dengan cara bermain *game* yang telah dibangun.

6.2. Saran

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

1. Dimensi labirin yang luas dapat dimanfaatkan agar permainan dapat melakukan *multiplayer*.
2. Sensor *accelerometer* yang disisipkan menjadi bantuan kontrol dalam permainan dengan jenis *Third Person Shooter (TPS)* masih kurang menarik dikarenakan pemain yang kurang terbiasa, oleh karenanya diperlukan pendekatan lebih jauh terhadap menentukan bantuan kontrol dalam permainan dengan jenis *Third Person Shooter (TPS)*

LAMPIRAN HASIL KUESIONER

①

KUISIONER TUGAS AKHIR

Nama Lengkap : Mahendra Hursa W.

Profesi : Mahasiswa

No.	Nama Pengujian	Penilaian				
		1	2	3	4	5
1.	Keindahan Tampilan				✓	
2.	Kesesuaian tema			✓		
3.	Ketertarikan Bermain			✓		
4.	Kemudahan Materi Berhitung					✓
5.	Kemudahan Menyelesaikan Level			✓		
6.	Kemudahan Memenangkan Permainan			✓		
7.	Kesesuaian dimensi labirin dengan kesulitan level dipilih				✓	
8.	Kemudahan Menggunakan Sensor <i>Accelerometer</i>		✓			
9.	Kemudahan Menggunakan <i>Joystick</i>			✓		


(MAHENDRA HURSA W.)

✕

Gambar 8.1 Kuesioner Responden 1

②

KUISIONER TUGAS AKHIR

Nama Lengkap : Rizaldi Tri Yanuar

Profesi : Mahasiswa

No.	Nama Pengujian	Penilaian				
		1	2	3	4	5
1.	Keindahan Tampilan				✓	
2.	Kesesuaian tema				✓	
3.	Ketertarikan Bermain				✓	
4.	Kemudahan Materi Berhitung					✓
5.	Kemudahan Menyelesaikan Level			✓		
6.	Kemudahan Memenangkan Permainan				✓	
7.	Kesesuaian dimensi labirin dengan kesulitan level dipilih				✓	
8.	Kemudahan Menggunakan Sensor <i>Accelerometer</i>		✓			
9.	Kemudahan Menggunakan <i>Joystick</i>			✓		


 (... Rizaldi Tri Y.)

Gambar 8.2 Kuesioner Responden 2

KUISIONER TUGAS AKHIR

Nama Lengkap : *Yoga Pratama A.*
 Profesi : *Mahasiswa.*

No.	Nama Pengujian	Penilaian				
		1	2	3	4	5
1.	Keindahan Tampilan			✓		
2.	Kesesuaian tema			✓		
3.	Ketertarikan Bermain			✓		
4.	Kemudahan Materi Berhitung					✓
5.	Kemudahan Menyelesaikan Level				✓	
6.	Kemudahan Memenangkan Permainan					✓
7.	Kesesuaian dimensi labirin dengan kesulitan level dipilih			✓		
8.	Kemudahan Menggunakan Sensor <i>Accelerometer</i>				✓	
9.	Kemudahan Menggunakan <i>Joystick</i>		✓			

Yoga
(YOGA P. A.)

Gambar 8.3 Kuesioner Responden 3

④

KUISIONER TUGAS AKHIR

Nama Lengkap : ANDRIE PRASETYO UTOMO

Profesi : MAHASISWA

No.	Nama Pengujian	Penilaian				
		1	2	3	4	5
1.	Keindahan Tampilan				✓	
2.	Kesesuaian tema				✓	
3.	Ketertarikan Bermain				✓	
4.	Kemudahan Materi Berhitung					✓
5.	Kemudahan Menyelesaikan Level			✓		
6.	Kemudahan Memenangkan Permainan				✓	
7.	Kesesuaian dimensi labirin dengan kesulitan level dipilih				✓	
8.	Kemudahan Menggunakan Sensor <i>Accelerometer</i>				✓	
9.	Kemudahan Menggunakan <i>Joystick</i>				✓	


 (.....)
 ANDRIE P.U.

Gambar 8.4 Kuesioner Responden 4

5

KUISIONER TUGAS AKHIR

Nama Lengkap : M. Aprialdi Rizky P

Profesi : Mahasiswa

No.	Nama Pengujian	Penilaian				
		1	2	3	4	5
1.	Keindahan Tampilan				✓	
2.	Kesesuaian tema				✓	
3.	Ketertarikan Bermain				✓	
4.	Kemudahan Materi Berhitung					✓
5.	Kemudahan Menyelesaikan Level			✓		
6.	Kemudahan Memenangkan Permainan					✓
7.	Kesesuaian dimensi labirin dengan kesulitan level dipilih				✓	
8.	Kemudahan Menggunakan Sensor <i>Accelerometer</i>				✓	
9.	Kemudahan Menggunakan <i>Joystick</i>					✓

()
M. Aprialdi Rizky P

Gambar 8.5 Kuesioner Responden 5

BIODATA PENULIS



Penulis dilahirkan di Surabaya, 15 Oktober 1993, merupakan anak keempat dari empat bersaudara. Penulis telah menempuh pendidikan formal yaitu TK Dharma Wanita ITS, Surabaya (1997-1999), SD Negeri Klampis Ngasem I/246 Surabaya (1999-2005), SMP Negeri 19 Surabaya (2005-2008), SMA Negeri 16 Surabaya (2008-2011), dan mahasiswa S1 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya rumpun mata kuliah Interaksi, Grafika, dan Seni (2011-2015). Selama menjadi mahasiswa penulis pernah berperan sebagai asisten Teori Graph dan Otomata (2012-2013), Analisa Perancangan dan Sistem (2013-2014), Manajemen Basis Data (2014-2015) serta Administrator Laboratorium Pemrograman (2012-2015). Penulis juga aktif berorganisasi menjadi Staff HMTC (2012-2013), Sekertaris Departemen Pengembangan Profesi Kabinet Bersahabat HMTC(2013-2014), Kepemanduan Bakor FTIf ITS – AVENGER(2012). Penulis dapat dihubungi melalui *surel* pundipun7@gmail.com.