



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - KI141502  
**IMPLEMENTASI KECERDASAN BUATAN PADA  
GAME EEL JOE MENGGUNAKAN ALGORITMA  
A\***

**HAFIELUDIN YUSUF RIZANA**  
**NRP 5112100024**

Dosen Pembimbing  
**Imam Kuswardayan, S.Kom., M.T.**  
**Ridho Rahman Hariadi, S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**  
**SURABAYA**  
**2016**





**TUGAS AKHIR - KI141502**  
**IMPLEMENTASI KECERDASAN BUATAN PADA**  
**GAME EEL JOE MENGGUNAKAN ALGORITMA**  
**A\***

**HAFIELUDIN YUSUF RIZANA**  
**NRP 5112100024**

**Dosen Pembimbing**  
**Imam Kuswardayan, S.Kom., M.T.**  
**Ridho Rahman Hariadi, S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**  
**SURABAYA**  
**2016**

*(Halaman ini sengaja dikosongkan)*



**FINAL PROJECT- KI141502**  
**IMPLEMENTATION OF ARTIFICIAL**  
**INTELLIGENCE IN GAME EEL JOE USING A\***  
**ALGORITHM**

**HAFIELUDIN YUSUF RIZANA**  
**NRP 5112100024**

**Advisor**  
**Imam Kuswardayan, S.Kom., M.T.**  
**Ridho Rahman Hariadi, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS**  
**FACULTY OF INFORMATION TECHNOLOGY**  
**SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY**  
**SURABAYA**  
**2016**

*(Halaman ini sengaja dikosongkan)*

## LEMBAR PENGESAHAN

### IMPLEMENTASI KECERDASAN BUATAN PADA GAME EEL JOE MENGGUNAKAN ALGORITMA A\*

## TUGAS AKHIR

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

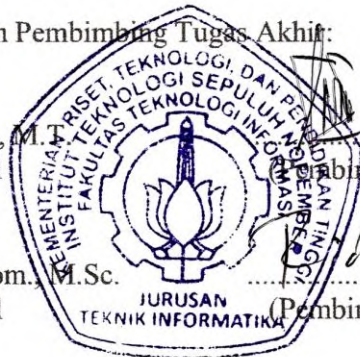
Oleh:

**HAFIELUDIN YUSUF RIZANA**  
**NRP. 5112 100 024**

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Imam Kuswardayan, S.Kom., M.Eng.  
NIP: 19761215 200312 1 001

Ridho Rahman Hariadi, S.Kom., M.Sc.  
NIP: 19870213 201404 1 001



.....  
Pembimbing 1

.....  
Pembimbing 2

**SURABAYA**  
**JUNI, 2016**

*(Halaman ini sengaja dikosongkan)*



# IMPLEMENTASI KECERDASAN BUATAN PADA GAME EEL JOE MENGGUNAKAN ALGORITMA A\*

**Nama** : Hafieludin Yusuf Rizana  
**NRP** : 5112100024  
**Jurusan** : Teknik Informatika FTIf-ITS  
**Dosen Pembimbing 1** : Imam Kuswardayan, S.Kom., M.T.  
**Dosen Pembimbing 2** : Ridho Rahman Hariadi, S.Kom., M.Sc.

## ABSTRAK

*Game merupakan aplikasi yang sangat pesat perkembangannya saat ini. Perkembangan game tidak hanya pada desktop game namun juga telah merambah ke mobile game.*

*Pada perkembangannya, game dituntut untuk bisa memberikan kesenangan dan tantangan jika dimainkan oleh satu orang. Hal inilah yang menyebabkan lahirnya komputer lawan(bot) pada game yang menggunakan kecerdasan buatan (AI).*

*Penggunaan kecerdasan buatan sering ditemui pada game, khususnya dalam game bergenre arcade maze. Hal ini dikarenakan dalam game jenis ini, pemain dituntut untuk mengumpulkan items di dalam maze sementara pemain menghindar dari kejaran bot. kecerdasan buatan digunakan untuk membantu bot dalam mencari jarak terpendek ke posisi pemain di dalam maze. Dengan digunakannya AI, diharapkan kesenangan dan tantangan yang dihadapi pemain saat bermain game lebih besar .*

*Eel Joe adalah sebuah game arcade maze yang dikembangkan dan mengandung kecerdasan buatan didalamnya. Game ini menerapkan algoritma A\* sebagai kecerdasan buatan terhadap bot.*

***Kata kunci: Kecerdasan Buatan, Algoritma A\*, Arcade Maze, Bot***

*(Halaman ini sengaja dikosongkan)*

## IMPLEMENTATION OF ARTIFICIAL INTELLIGENCE IN GAME EEL JOE USING A\* ALGORITHM

**Name** : Hafieludin Yusuf Rizana  
**NRP** : 5112 100 024  
**Department** : Teknik Informatika FTIf-ITS  
**Supervisor I** : Imam Kuswardayan, S.Kom., M.T.  
**Supervisor II** : Ridho Rahman Hariadi, S.Kom., M.Sc.

### ABSTRACT

*Game is application with a rapid development nowadays. The development of the game is not only on desktop games but also has penetrated into the mobile games.*

*In its development, game is required to be able to give pleasure and a challenge if played by one person. This is what causes the birth of enemy bot that uses artificial intelligence (AI).*

*The use of artificial intelligence is often encountered in the game, especially in the genre of arcade maze game. This is because in this type of game, players are required to collect the items in the maze while the player have to escape from Bot. artificial intelligence is used to assist Bot in finding the shortest distance to the position of the player in the maze. With the use of AI, the expected pleasures and challenges faced by the players during game become bigger.*

*Eel Joe is an arcade maze game that was developed and contained artificial intelligence. This game applies the A\* algorithm as the artificial intelligence of the Bot.*

***Keywords: Artificial Intelligence, A\* Algorithm, Arcade Maze, Bot.***

*(Halaman ini sengaja dikosongkan)*

## KATA PENGANTAR

Segala puji dan syukur kepada Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“IMPLEMENTASI KECERDASAN BUATAN PADA GAME EEL JOE MENGGUNAKAN ALGORITMA A\*”**.

Pengerjaan tugas akhir ini adalah momen bagi penulis untuk mengeluarkan seluruh kemampuan, hasrat, dan keinginan yang terpendam di dalam hati mulai dari masuk kuliah hingga lulus sekarang ini, lebih tepatnya di kampus Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya.

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. Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik
2. Ayah penulis, Warsito, dan Ibu penulis, Harnaning Inang Wahyuni, yang selalu memberikan dukungan, doa, perhatian, dan kasih sayang.
3. Adik penulis, Hafiza Mutia Inas, yang senantiasa memberi hiburan dan halangan di sela-sela menyelesaikan studinya.
4. Fajar Setiawan, yang telah menyelamatkan hidup penulis.
5. Bapak Imam Kuswardayan selaku dosen pembimbing Tugas Akhir pertama dan yang telah memberikan arahan serta masukan dalam pengerjaan Tugas Akhir ini.
6. Bapak Ridho Rahman Hariadi selaku dosen pembimbing Tugas Akhir kedua, dosen pembimbing KP, dan yang telah memberikan arahan serta masukan dalam setiap pengerjaan tugas penulis pada masa studinya.
7. Bapak Arya Yudhi Wijaya selaku dosen wali penulis, yang telah banyak membantu penulis dalam masa studinya.

8. Sahabat penulis, Satrio Ramadhana Yudhono, yang telah banyak membantu penulis.
9. Bima Nisrina Madjid, Sofi Izza Sabilla, Dwi Okta Sumadya, Anisa Dyan Pratiwi, yang telah membantu dan senantiasa memberi semangat untuk menyelesaikan studi penulis.
10. Dimas Widdy, Ikrom Aulia Fahdi, Luthfi Soehadak, Fahmy Thoriqul Haq, Bimo Sigit Rusanjaya, Rahma Fida Fadhilah, Wahyu Widyananda, Pinastika Cintya, Imaduddin Al Fikri, Angga Saputra, dan Aditya Putra Ferza, yang telah membantu penulis menyelesaikan tugas akhir ini.
11. Zulwaqor Maulana Achmad, yang senantiasa memberikan cobaan dan tantangan kepada penulis.
12. Bapak Radityo Anggoro selaku dosen koordinator Tugas Akhir yang telah membantu penulis atas segala sesuatu mengenai syarat-syarat dan terlaksananya sidang Tugas Akhir.
13. Bapak Darlis selaku Ketua Jurusan Teknik Informatika ITS yang selama ini memberikan bantuan kepada penulis.
14. Dosen-dosen Teknik Informatika yang dengan sabar mendidik dan memberikan pengalaman baru kepada penulis selama di Teknik Informatika.
15. Staf TU Teknik Informatika ITS yang senantiasa memudahkan segala urusan penulis di jurusan.
16. Rekan-rekan dan pengelola Laboratorium Interaksi, Grafik, dan Seni yang telah memberikan fasilitas dan kesempatan melakukan riset atas Tugas Akhir yang dikerjakan penulis.
17. Rekan-rekan bermain disaat susah dan senang yang telah menemani penulis disaat gundah dalam menyelesaikan tugas akhir ini.
18. Rekan-rekan dan sahabat-sahabat penulis angkatan 2012 yang memberikan dorongan motivasi dan bantuan kepada penulis.
19. 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 2016  
Penulis

Hafieludin Yusuf Rizana

*(Halaman ini sengaja dikosongkan)*



## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK .....	vii
ABSTRACT .....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
DAFTAR KODE SUMBER .....	xxi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Metodologi .....	3
1.7 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 Unity Game Engine .....	7
2.2 <i>Arcade Maze Game</i> .....	7
2.3 Kecerdasan Buatan (AI) .....	8
2.4 Algoritma A*.....	8
2.5 C# .....	11
2.6 Mixamo .....	12
BAB III ANALISIS DAN PERANCANGAN.....	13
3.1 Analisis Sistem .....	13
3.2 Perancangan Permainan.....	14
3.2.1 Deskripsi Umum Perangkat Lunak.....	14
3.2.2 Perancangan Kontrol Permainan .....	14
3.2.3 Perancangan Alur Permainan.....	14
3.2.4 Rancangan Permainan.....	15
3.2.5 Perancangan Antarmuka Pengguna .....	17
3.2.6 Perancangan Karakter .....	21
3.3 Perancangan Algoritma dan Kecerdasan Buatan.....	23

3.3.1	Perancangan Kecerdasan Buatan .....	23
3.3.2	Perancangan Algoritma.....	24
3.3.3	Perancangan Posisi Pemasangan Bom .....	25
3.3.4	Perancangan Level.....	26
BAB IV IMPLEMENTASI.....		29
4.1	Lingkungan Implementasi .....	29
4.2	Implementasi Permainan .....	29
4.2.1	Implementasi <i>Main menu</i> .....	29
4.2.2	Implementasi Level Selector.....	31
4.2.3	Implementasi Kontrol Pemain .....	34
4.2.4	Implementasi Posisi Bom .....	41
4.2.5	Implementasi Kecerdasan Buatan .....	42
4.2.6	Implementasi Menang Atau Kalah .....	51
BAB V PENGUJIAN DAN EVALUASI .....		53
5.1	Lingkungan Uji Coba .....	53
5.2	Pengujian Fungsionalitas .....	53
5.2.1	Pengujian Layar Menu Utama .....	54
5.2.2	Pengujian Layar Level Selector .....	55
5.2.3	Pengujian Layar Permainan .....	55
5.2.4	Pengujian Layar Hasil Permainan.....	58
5.2.5	Hasil Pengujian Fungsional .....	59
5.3	Pengujian Pengguna.....	60
5.3.1	Skenario Uji Coba Pengguna .....	60
5.3.2	Daftar Penguji Perangkat Lunak .....	60
5.3.3	Hasil Uji Coba Pengguna .....	61
5.4	Evaluasi Pengujian.....	64
BAB VI KESIMPULAN DAN SARAN.....		67
6.1.	Kesimpulan .....	67
6.2.	Saran .....	67
DAFTAR PUSTAKA.....		69
LAMPIRAN A. HASIL KUESIONER .....		71
BIODATA PENULIS.....		77

## DAFTAR GAMBAR

Gambar 2.1 Pseudocode algoritma A* .....	9
Gambar 2.2 Diagram Flowchart Algoritma A* .....	10
Gambar 3.1 FSM gerak pemain .....	15
Gambar 3.2 FSM memasang bom .....	16
Gambar 3.3 penempatan ruangan pemasangan bom .....	16
Gambar 3.4 Tampilan <i>Main menu</i> .....	17
Gambar 3.5 Tampilan Pemilihan Level .....	18
Gambar 3.6 Tampilan Permainan .....	19
Gambar 3.7 Tampilan Permainan saat <i>Pause</i> .....	19
Gambar 3.8 Tampilan Permainan Berakhir saat menang .....	20
Gambar 3.9 Tampilan Permainan Berakhir saat kalah .....	21
Gambar 3.10 Karakter Pemain .....	22
Gambar 3.11 Karakter Bot .....	22
Gambar 3.12 Ilustrasi pengambilan keputusan .....	23
Gambar 3.13 Ilustrasi rute node bot ke pemain .....	25
Gambar 3.14 Grafik Tingkat kesulitan Tiap Level .....	27
Gambar 4.1 Implementasi antarmuka <i>Main menu</i> .....	30
Gambar 4.2 Tombol <i>Play</i> .....	30
Gambar 4.3 Tombol <i>Quit</i> .....	31
Gambar 4.4 Implementasi antarmuka Level Selector .....	31
Gambar 4.5 Tombol Komik .....	32
Gambar 4.6 Tombol Level .....	32
Gambar 4.7 Tombol <i>Back</i> .....	33
Gambar 4.8 Tampilan kontrol karakter pemain .....	34
Gambar 4.9 Pemasangan bom .....	39
Gambar 4.10 Bom berhasil dipasang .....	41
Gambar 4.11 Posisi bom .....	41
Gambar 4.12 Bot mengejar pemain .....	50
Gambar 4.13 Tampilan Menang .....	51
Gambar 4.14 Tampilan pemain kalah .....	52
Gambar 5.1 Tampilan layar Menu Utama .....	54
Gambar 5.2 Tampilan layar Level .....	55
Gambar 5.3 Tampilan layar Permainan .....	56

Gambar 5.4 Tampilan layar permainan saat <i>pause</i> .....	56
Gambar 5.5 Pemain bergerak ke ruangan lain.....	57
Gambar 5.6 Bot mengejar pemain .....	57
Gambar 5.7 Tampilan hasil permainan ketika menang .....	58
Gambar 5.8 Tampilan hasil permainan ketika kalah .....	59
Gambar A.1 Kuesioner Pengguna 1 .....	71
Gambar A.2 Kuesioner Pengguna 2 .....	72
Gambar A.3 Kuesioner Pengguna 3 .....	73
Gambar A.4 Kuesioner Pengguna 4 .....	74
Gambar A.5 Kuesioner Pengguna 5 .....	75

## DAFTAR TABEL

Tabel 3.1 Presentase dari Faktor Tingkat Kesulitan Level.....	26
Tabel 3.2 Total Nilai Tiap Level .....	27
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak.....	29
Tabel 5.1 Lingkungan Ujicoba Perangkat Lunak.....	53
Tabel 5.2 Pengujian Permainan.....	54
Tabel 5.3 Hasil Pengujian Fungsionalitas .....	59
Tabel 5.4 Daftar Nama Penguji.....	61
Tabel 5.5 Penilaian Antarmuka .....	62
Tabel 5.6 Penilaian Performa Sistem .....	62
Tabel 5.7 Penilaian Kecerdasan Buatan .....	63
Tabel 5.8 Rekapitulasi Hasil Uji Coba Pengguna .....	63

*(Halaman ini sengaja dikosongkan)*

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Tombol <i>Play</i> .....	30
Kode Sumber 4.2 Tombol <i>Quit</i> .....	31
Kode Sumber 4.3 Tombol <i>Komik</i> .....	32
Kode Sumber 4.4 Tombol <i>Level</i> .....	33
Kode Sumber 4.5 Tombol <i>Back</i> .....	33
Kode Sumber 4.6 Kontrol pemain .....	39
Kode Sumber 4.7 Pemasangan bom .....	40
Kode Sumber 4.8 Peletakan tempat pemasangan bom .....	42
Kode Sumber 4.9 Representasi map ke dalam <i>list</i> .....	43
Kode Sumber 4.10 Mendapatkan posisi pemain .....	44
Kode Sumber 4.11 Mendapatkan posisi bot .....	44
Kode Sumber 4.12 Mendapatkan nilai F dari <i>node</i> kanan .....	45
Kode Sumber 4.13 Mendapatkan nilai F dari <i>node</i> kiri .....	46
Kode Sumber 4.14 Mendapatkan nilai F dari <i>node</i> atas .....	47
Kode Sumber 4.15 Mendapatkan nilai F dari <i>node</i> bawah .....	48
Kode Sumber 4.16 Mencari <i>route node</i> .....	49
Kode Sumber 4.17 Mengganti <i>route node</i> .....	50
Kode Sumber 4.18 Aturan Menang .....	51
Kode Sumber 4.19 Aturan Kalah .....	52

*(Halaman ini sengaja dikosongkan)*



# 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

Aplikasi permainan (*Video Game*) merupakan aplikasi yang sangat digemari saat ini. Perkembangan *game* begitu pesat dengan jenis yang beragam. Salah satu jenis dari *game* tersebut adalah *arcade maze*. *Arcade maze* adalah jenis *game* yang mengasah otak pemain untuk memenangkan *game* tersebut. Namun, *game maze* yang ada saat ini tidak memiliki *gameplay* yang baru untuk ditawarkan kepada pemain.

Melihat permasalahan diatas, ditawarkan sebuah inovasi *gameplay* untuk *game* berjenis *arcade maze*. Inovasi yang akan ditawarkan yaitu inovasi terhadap interaksi pemain ke *game*. Pemain mengontrol karakter pemain dengan cara memilih *path* yang muncul di layar. Pemain hanya bisa memilih satu *path* pada setiap langkah. Pemain harus mengecoh dan menghindari komputer lawan(bot) untuk meletakkan bom dan memenangkan level.

Game dengan jenis *maze* memiliki kecerdasan buatan dengan strategi tertentu untuk mengalahkan pemain. Salah satu algoritma yang digunakan pada kecerdasan buatan ini adalah Algoritma A\*. Algoritma ini biasanya digunakan untuk menyelesaikan permasalahan jarak terpendek (*shortest path algorithm*). Pada aplikasi permainan yang akan dibangun nanti, Algoritma A\* akan digunakan untuk mengimplementasikan kecerdasan buatan ke bot. Bot tersebut harus mencari rute terpendek ke pemain untuk menangkapnya sebelum pemain memenangkan permainan dengan meletakkan bom dalam *maze*.

## 1.2 Rumusan Masalah

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

1. Bagaimana menyusun aturan main, skenario, dan level pada aplikasi permainan menjadi lebih menantang
2. Bagaimana perancangan strategi untuk menang terhadap bot dalam aplikasi permainan ?
3. Bagaimana mengimplementasikan rancangan strategi dengan Algoritma A\*?

## 1.3 Batasan Masalah

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

1. Aplikasi yang dibangun dengan menggunakan IDE Unity dan bahasa pemrograman `c#`
2. Maksimal 3 bot pada setiap level
3. Batas maksimal luas *maze* adalah 100

## 1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini, antara lain:

1. Membangun aplikasi permainan *Eel Joe* berbasis perangkat bergerak
2. Membuat kecerdasan buatan pada bot *game Eel Joe* yang memiliki strategi untuk menang
3. Mengimplementasikan Algoritma A\* pada kecerdasan buatan bot

## 1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini antara lain:

1. Menjadi referensi penelitian selanjutnya untuk *game* berjenis *arcade maze*
2. Memberikan tantangan kepada pemain dengan komputer yang lebih pintar
3. Memberikan media hiburan bagi para pengguna.

## 1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

### A. 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. Penggunaan Unity Game Engine;
2. Algoritma A\*;

### B. Analisis dan desain perangkat lunak

Perangkat lunak yang akan dibangun merupakan aplikasi untuk perangkat bergerak. Pada tahap ini dilakukan analisa awal dan pendefisian 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. Perancangan strategi menang untuk kecerdasan buatan lawan.
2. Analisis Algoritma A\* dalam membangun aplikasi permainan *Eel Joe* Analisis kebutuhan non fungsional.
3. Perancangan aturan main, skenario, dan level yang menantang bagi pemain.

### C. 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 IDE Unity.

#### D. Uji coba dan evaluasi

Pada tahap ini akan dilakukan pengujian terhadap perangkat lunak menggunakan data atau skenario yang telah dipersiapkan sebelumnya yakni dengan pengujian usabilitas dengan cara melakukan survei. Responden dalam survei ini adalah beberapa pengguna di sekitar lingkungan Teknik Informatika ITS yang suka bermain game. Survei ini dilakukan untuk mengukur tingkat kegunaan dari aplikasi yang dibuat dalam membantu pengguna.

#### E. Penyusunan laporan tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan tugas akhir.

### 1.7 Sistematika Penulisan

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

#### **BAB I PENDAHULUAN**

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

#### **BAB II TINJAUAN PUSTAKA**

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

#### **BAB III ANALISIS DAN PERANCANGAN**

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas

rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, arsitektur, data, dan antarmuka.

#### **BAB IV IMPLEMENTASI**

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembangkitan area permainan, dan antarmuka permainan.

#### **BAB V PENGUJIAN DAN EVALUASI**

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

#### **BAB VI PENUTUP**

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

*(Halaman ini sengaja dikosongkan)*

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan tugas akhir. Teori-teori tersebut adalah Unity Game Engine, C#, Kecerdasan Buatan, Algoritma Genetika, dan Pembobotan.

#### **2.1 Unity Game Engine**

Unity adalah salah satu perangkat lunak untuk mengembangkan game 2D dan 3D, serta animasi tiga dimensi. Unity lebih tepat dijelaskan sebagai salah satu perangkat lunak untuk mengembangkan video game atau disebut juga *game engine*, yang dikembangkan oleh Unity Technologies dan digunakan untuk membuat game berbasis PC, *console*, perangkat *mobile* dan *website*. *Graphics engine* yang digunakan pada Unity adalah Direct3D, OpenGL, OpenGL ES, dan proprietary APIs. Unity juga dapat mengambil format desain dari berbagai macam aplikasi desain terkenal yang dapat ditambahkan ke game project dan diatur melalui *graphical user interface* Unity. *Script game engine* pada Unity dibuat dengan Visual Studio, sebuah implementasi open-source dari .NET Framework. Dalam Unity disediakan berbagai pilihan bahasa pemrograman untuk mengembangkan game, antara lain JavaScript, C#, dan BooScript. Unity juga menyertakan Server Unity Asset, yaitu sebuah solusi terkontrol untuk developer game asset dan script [1].

#### **2.2 Arcade Maze Game**

*Maze* merupakan salah satu jenis dalam *game* dimana pemain harus bermain dalam *labyrinth/maze* untuk kabur dari *monster*, mengumpulkan *item*, atau menangkap penjahat. Beberapa contoh game *arcade maze* yang terkenal adalah Pacman, Labyrinth, dan Maze war. Pada game tugas akhir ini akan dirancang *gameplay* yang memberikan pemain tantangan dan kesenangan lebih ketika bermain *game* berjenis *arcade maze*.

### 2.3 Kecerdasan Buatan (AI)

Kecerdasan buatan atau kecerdasan yang ditambahkan kepada suatu sistem yang dapat diatur dalam konteks ilmiah (sering disebut Intelegensi Artifisial) didefinisikan sebagai kecerdasan entitas ilmiah. Penerapan kecerdasan buatan pada game merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Aplikasi atau program kecerdasan buatan dapat ditulis dalam semua bahasa komputer [2].

Kecerdasan buatan digunakan di berbagai bidang cukup berbeda dalam sebuah permainan. Mencari jalan adalah salah satu kegunaan kecerdasan buatan pada sebuah *game* yang sering dijumpai pada permainan *arcade maze*. Mencari jalan adalah metode untuk menentukan bagaimana untuk mendapatkan nilai dari satu titik pada peta lain dengan mempertimbangkan medan, hambatan, dan suasana permainan [3].

### 2.4 Algoritma A\*

Algoritma A\* merupakan salah satu dari algoritma yang menerapkan teknik heuristik (teknik pencarian terbimbing). Algoritma A\* (A-Star) merupakan salah satu algoritma yang dipelajari untuk menyelesaikan permasalahan. Algoritma A\* menyelesaikan masalah yang menggunakan graf untuk perluasan ruang statusnya. Dengan kata lain digunakan untuk menyelesaikan permasalahan yang bisa direpresentasikan dengan graf. Algoritma A\* adalah sebuah algoritma yang telah diperkaya dengan menerapkan suatu heuristik, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan [4].

Algoritma A\* membangkitkan *node* yang paling mendekati solusi (*next node*). *Node* ini kemudian disimpan suksesornya ke dalam *list(open list)* sesuai dengan urutan yang paling mendekati solusi terbaik. Kemudian, *node* pertama pada list diambil dan



menjadi *current node*, dibangkitkan suksesornya dan kemudian suksesor ini disimpan ke dalam list sesuai dengan urutan yang terbaik untuk solusi. *Node list* ini disebut dengan *route node*. Ketika *current node* telah mencapai *end node*, *route node* dieksekusi dengan membuat rute dari *start node* ke *end node* sesuai *node* yang ada di *route node* [5]. Pseudocode Algoritma A\* dapat dilihat pada Gambar 2.1. Ketika Algoritma A\* menerapkan teknik heuristik dalam membantu penyelesaian persoalan. Heuristik adalah penilai yang memberi harga pada tiap simpul yang memandu A\* mendapatkan solusi yang diinginkan. Dengan heuristik yang benar, maka A\* pasti akan mendapatkan solusi (jika memang ada solusinya) yang dicari. Dengan kata lain, heuristik adalah fungsi optimasi yang menjadikan Algoritma A\* lebih baik dari pada algoritma lainnya. Namun heuristik masih merupakan estimasi / perkiraan biasa saja Sama sekali tidak ada rumus khususnya. Artinya, setiap kasus memiliki fungsi heuristik yang berbeda-beda. Algoritma A\* ini bisa dikatakan mirip dengan algoritma Dijkstra, namun pada algoritma Dijkstra, nilai fungsi heuristiknya selalu 0 (nol) sehingga tidak ada fungsi yang mempermudah pencarian solusinya [6].

Dalam aplikasi yang akan dibuat, algoritma A\* akan diiterapkan pada bot. Algoritma A\* digunakan untuk menyusun strategi computer lawan untuk mencari rute ke pemain. Gambar flowchart Algoritma A\* dapat dilihat pada Gambar 2.2.

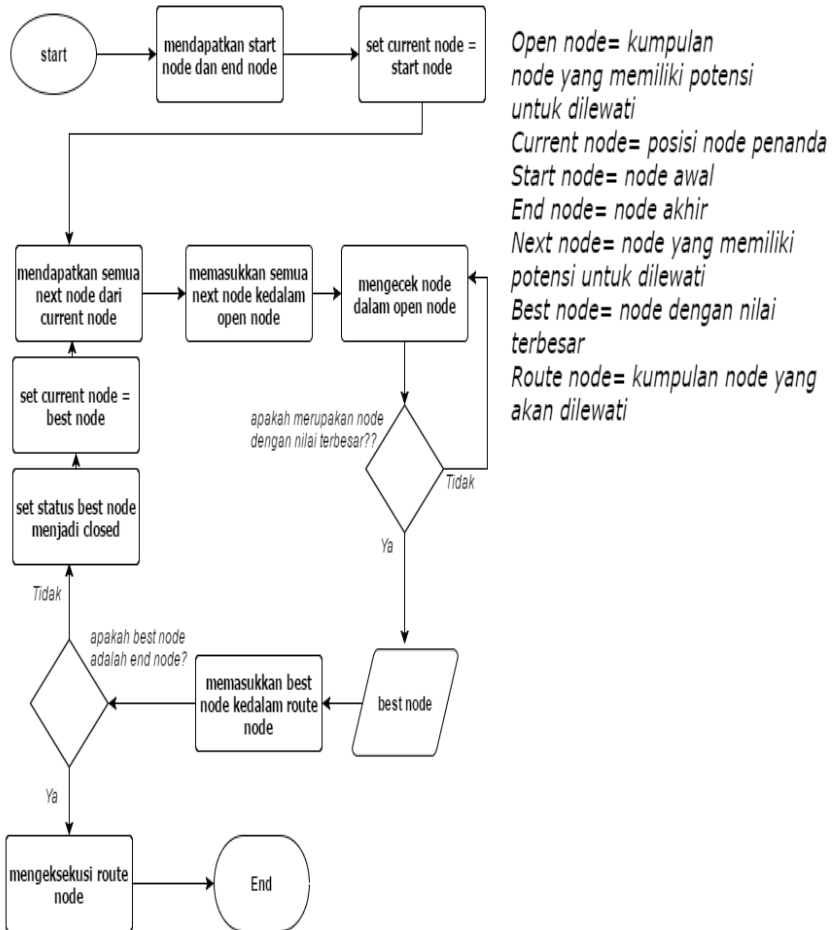
```

// A*
1: initialize the open list
2: initialize the closed list
3: put the starting node on the open list (you can leave its f at zero)
-
4: while the open list is not empty
5:   find the node with the least f on the open list, call it "q"
6:   pop q off the open list
7:   generate q's 8 successors and set their parents to q
8:   for each successor
9:     if successor is the goal, stop the search
10:    successor.g = q.g + distance between successor and q
11:    successor.h = distance from goal to successor
12:    successor.f = successor.g + successor.h
-
13:    if a node with the same position as successor is in the OPEN list \
-    which has a lower f than successor, skip this successor
14:    if a node with the same position as successor is in the CLOSED list \
-    which has a lower f than successor, skip this successor
15:    otherwise, add the node to the open list
16:  end
17:  push q on the closed list
18: end

```

**Gambar 2.1 Pseudocode algoritma A\***

*Algoritma A\**



**Gambar 2.2 Diagram Flowchart Algoritma A\***

## 2.5 C#

Microsoft C# (disebut C sharp) adalah sebuah bahasa pemrograman yang didesain untuk membangun jangkauan aplikasi enterprise yang berjalan di atas framework .NET. Sebuah evolusi Microsoft C dan Microsoft C++, C# sederhana, modern, aman dan Object Oriented. C# dikenal sebagai visual C# dalam Visual Studio .Net. Dukungan untuk Visual C# termasuk proyek template, desainer, halaman properti, kode, model objek dan fitur lain dari lingkungan pengembangan. Library untuk pemrograman visual c# adalah .NET Framework.

Bahasa pemrograman ini dibuat berbasis bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti *Java*, *Delphi*, *Visual Basic*, dan lain-lain) dengan beberapa penyederhanaan. Menurut standar ECMA-334 *C# Language Specification*, nama C# terdiri atas sebuah huruf Latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam keyboard standar.

Bahasa pemrograman C# ditujukan untuk digunakan dalam mengembangkan komponen perangkat lunak yang mampu mengambil keuntungan dari lingkungan terdistribusi. C# ditujukan agar cocok digunakan untuk menulis program aplikasi baik dalam sistem klien-server (*hosted system*) maupun sistem embedded (*embedded system*), mulai dari perangkat lunak yang sangat besar yang menggunakan sistem operasi yang canggih hingga kepada perangkat lunak yang sangat kecil yang memiliki fungsi-fungsi terdedikasi. Meskipun aplikasi C# ditujukan agar bersifat 'ekonomis' dalam hal kebutuhan pemrosesan dan memori komputer, bahasa C# tidak ditujukan untuk bersaing secara langsung dengan kinerja dan ukuran perangkat lunak yang dibuat dengan menggunakan bahasa pemrograman C dan bahasa rakitan [7].

## 2.6 Mixamo

Mixamo adalah sebuah perusahaan teknologi komputer 3D yang ada di San Francisco. Mixamo memberikan fasilitas kepada *developer* maupun *3D artist* untuk membuat 3D karakter ataupun memberikan animasi kepada karakter yang telah dibuat melalui *online platform* mereka. Teknologi Mixamo menggunakan metode *machine learning* untuk mengotomasi langkah proses animasi karakter, meliputi *3D modeling* untuk *rigging* dan animasi 3D. Mixamo juga menyediakan karakter 3D untuk *developer* yang akan membangun aplikasi 3D. Setiap karakter 3D dikelompokkan sesuai tema dari model 3D dan animasinya.

## **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 pembuatan level permainan secara statis pada semua skenario. Bagian ini berisi data masukan, proses, serta keluaran yang dihasilkan, serta algoritma dan *pseudocode* yang digunakan.

Bagian kedua berisi perancangan permainan, perancangan kecerdasan buatan dan perancangan algoritma  $A^*$  yang akan digunakan.

#### **3.1 Analisis Sistem**

*Video game* berkembang sangat pesat saat ini. Perkembangan ini dikarenakan *video game* semakin menyenangkan ketika dimainkan. Salah satu faktor yang mempengaruhi tingkat kesenangan pemain dari *video game* yaitu level permainan. Level permainan yang beragam tentunya membutuhkan variasi tingkat kesulitan pada permainan. Salah satu cara untuk meningkatkan tingkat kesulitan permainan adalah dengan menyediakan lawan yang menantang.

Aplikasi ini dibangun dengan tujuan menyediakan tantangan agar para pemain dapat menikmati *game* dengan tingkat kesulitan yang berbeda. kecerdasan buatan pada *game* umumnya dibangun dengan aturan – aturan tertentu yang sudah ditentukan oleh pembuat.

Penulis menggunakan *IDE* Unity Game Engine dan bahasa pemrograman C# untuk memfasilitasi pengembangan permainan. Kecerdasan buatan yang dibangun pada aplikasi menggunakan algoritma  $A^*$  dengan fungsi heuristik jarak manhattan.

## **3.2 Perancangan Permainan**

### **3.2.1 Deskripsi Umum Perangkat Lunak**

Tugas akhir yang akan dikembangkan adalah sebuah permainan 3D bergenre *arcade maze*. Kecerdasan buatan pada *game* dibuat dengan memanfaatkan algoritma A\* untuk menentukan langkah yang akan diambil oleh AI.

Pengguna utama dari *game* ini adalah semua orang yang ingin bermain. Pemain berperan sebagai teroris yang ingin memasang bom pada ruangan tertentu dalam *maze*. Pemain dapat memilih level yang akan dimainkan pada Layar Pemilihan Level. Pada Layar Pemilihan Level disediakan beberapa level yang dapat dimainkan oleh pemain. Semakin tinggi level yang dipilih kesulitan yang dihadapi pemain untuk menyelesaikan level tersebut menjadi bertambah sulit. Serta pemain diharuskan memilih level pertama terlebih dahulu dan harus memenangkan level tersebut untuk bisa memilih level selanjutnya.

Dalam sistem permainan ini, pemain menghindari dari kejaran tim penyerang teroris(AI) sementara pemain juga harus mencari ruang pemasangan bom dan memasang bom. Pemain dinyatakan menang ketika semua bom telah dipasang.

### **3.2.2 Perancangan Kontrol Permainan**

Terdapat satu jenis kontrol dalam permainan yaitu menggunakan *mouse* dengan klik kiri. Semua interaksi dari user hanya menggunakan satu kontrol tersebut.

### **3.2.3 Perancangan Alur Permainan**

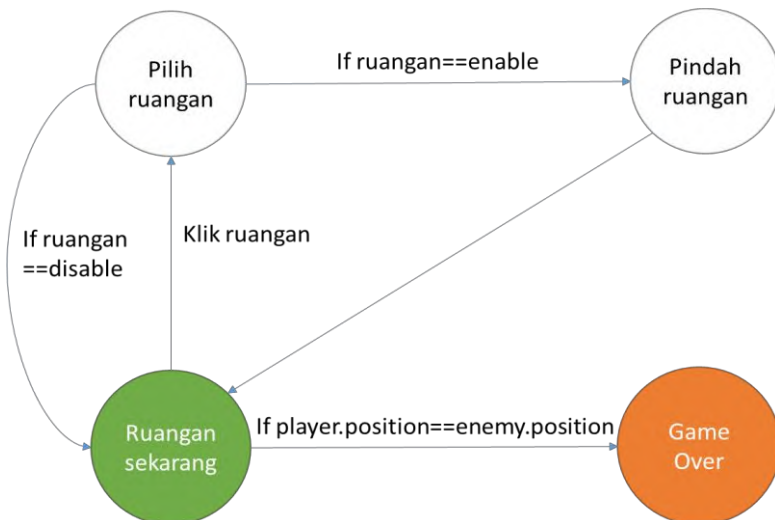
Alur permainan merupakan serangkaian proses yang harus diikuti pemain untuk memperoleh kemenangan. Dalam permainan ini pemain berperan sebagai teroris yang ditugaskan memasang bom pada beberapa ruangan dalam bangunan. Sementara itu, pemain akan diburu oleh tim penyerang teroris yang harus menemukan pemain sebelum pemain memasang seluruh bom yang ada.

Pemain dapat dikatakan menang jika pemain sudah memasang seluruh bom yang harus dipasang dalam bangunan.

### 3.2.4 Rancangan Permainan

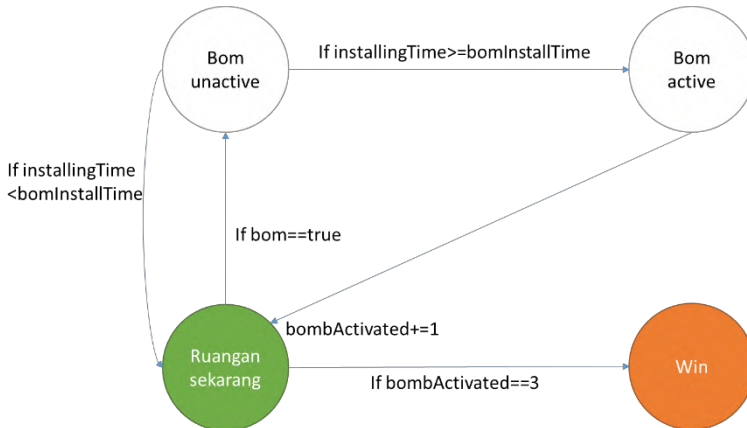
Rancangan simulasi pada permainan ini menggunakan pemodelan *finite state machine* (FSM). FSM gerak pemain adalah model FSM yang menunjukkan perpindahan ruangan yang dilakukan oleh pemain di dalam *maze*. FSM gerak pemain dapat dilihat pada Gambar 3.1.

Alur FSM ini dimulai dari halaman permainan setelah memilih level permainan. Pemain akan memilih ruangan lain dalam *maze* untuk mencari ruangan pemasangan bom. Pemain hanya bisa memilih ruangan atas, bawah, kanan, kiri dari posisi ruangnya saat ini. Jika pemain memilih ruangan selain dari ruangan yang bisa dilewati, karakter pemain tidak akan bergerak ke ruangan tersebut. Pemain harus berpindah dari satu ruangan ke ruangan lain untuk menghindari musuh. Jika posisi pemain sama dengan posisi musuh, maka permainan berakhir.



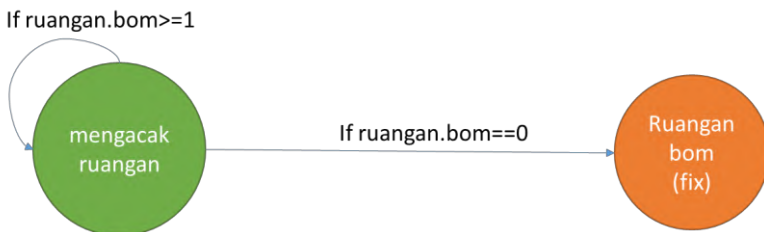
**Gambar 3.1** FSM gerak pemain

Pada saat pemain bergerak dari satu ruangan ke ruangan lain, pemain akan menemukan tanda bahwa pemain harus memasang bom pada ruangan tersebut. FSM memasang bom adalah model FSM yang menunjukkan proses pemain memasang bom untuk memenangkan permainan. FSM memasang bom dapat dilihat pada Gambar 3.2.



**Gambar 3.2 FSM memasang bom**

Setiap bom pada *maze* terdapat dalam ruangan yang berbeda. Dalam satu ruangan tidak pernah ada lebih dari satu bom. FSM posisi bom adalah FSM yang menunjukkan proses penempatan ruangan pemasangan bom. FSM penempatan ruangan pemasangan bom dapat dilihat pada Gambar 3.3.



**Gambar 3.3 penempatan ruangan pemasangan bom**



### 3.2.5 Perancangan Antarmuka Pengguna

Subbab ini membahas bagaimana rancangan antarmuka pengguna yang akan digunakan untuk tugas akhir. Rancangan antarmuka yang dibahas meliputi ketentuan masukan dan rancangan halaman tampilan. Dalam aplikasi ini terdapat beberapa tampilan, yaitu *Main menu*, Pemilihan Level, Permainan, Pemain Kalah, dan Pemain Menang.

#### 3.2.5.1 Tampilan *Main menu*

Tampilan *Main menu* merupakan tampilan yang pertama kali muncul ketika aplikasi dijalankan pertama. Pada tampilan ini terdapat judul game, tombol *Play* dan tombol *Quit*. Tampilan layar *Main menu* ditampilkan pada Gambar 3.4.



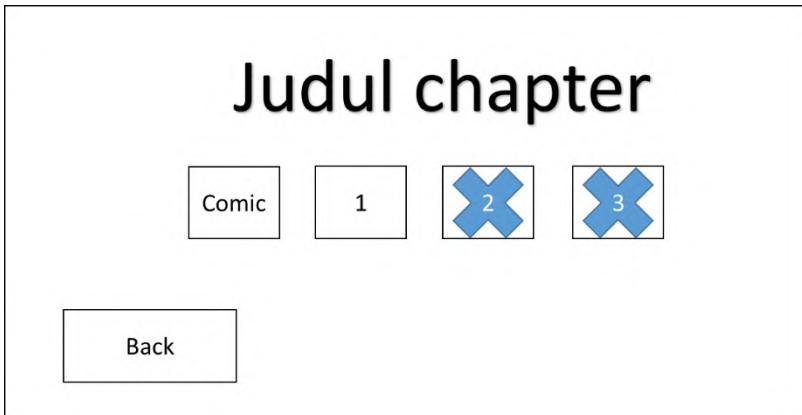
**Gambar 3.4 Tampilan *Main menu***

#### 3.2.5.2 Tampilan Pemilihan Level

Layar Pemilihan Level ditampilkan sesudah pemain menekan tombol *Play*. Dalam tampilan ini dapat dilihat level – level yang sudah dibuka oleh pemain.

Pada saat aplikasi permainan pertama kali dijalankan pilihan level yang terbuka hanya story dan level 1 saja. Seiring pemain mengalahkan level-level yang ada, level-level yang lebih tinggi akan terbuka.

*Icon* level yang masih belum terbuka akan tertutup Gambar silang dan tidak dapat dipilih. Tampilan layar pemilihan level ditampilkan pada Gambar 3.5.



**Gambar 3.5 Tampilan Pemilihan Level**

### **3.2.5.3 Tampilan Permainan**

Tampilan permainan merupakan tampilan dimana pemain telah memilih level permainan. Dalam tampilan ini akan ditunjukkan sebuah *maze* berupa kumpulan ruangan dan sebuah map. Ruangan-ruangan tersebut berisi karakter pemain, karakter musuh, dan bom. Tampilan permainan dapat dilihat pada Gambar 3.6. Pada tampilan ini disajikan beberapa objek 3D berlatar belakang sesuai dengan *chapter level*. Objek 3D penyergap teroris mewakili karakter musuh, objek 3D teroris mewakili karakter pemain, dan objek 3D bom dengan mewakili bom. Bom memiliki lampu warna, jika lampu warna bom berwarna merah maka bom

belum terpasang, jika lampu bom berwarna hijau maka bom telah terpasang.

Dalam satu buah *maze* hanya berisi 3 bom yang harus dipasang oleh pemain. Ruang pemasangan bom tersebut posisinya selalu acak. Pemain dituntut mencari ruang pemasangan bom dengan bergerak dari satu ruang ke ruangan lain. Pemain hanya bisa bergerak ke ruangan atas, bawah, kanan, dan kiri dari ruangan pemain.

			Ruang gerak		pause
Enemy		Ruang gerak	Player	Ruang gerak	
			Ruang gerak		
Map					bomb

**Gambar 3.6 Tampilan Permainan**

			Ruang gerak		pause
Enemy		Ruang gerak	Player	Ruang gerak	resume
			Ruang gerak		restart
Map					quit
					bomb

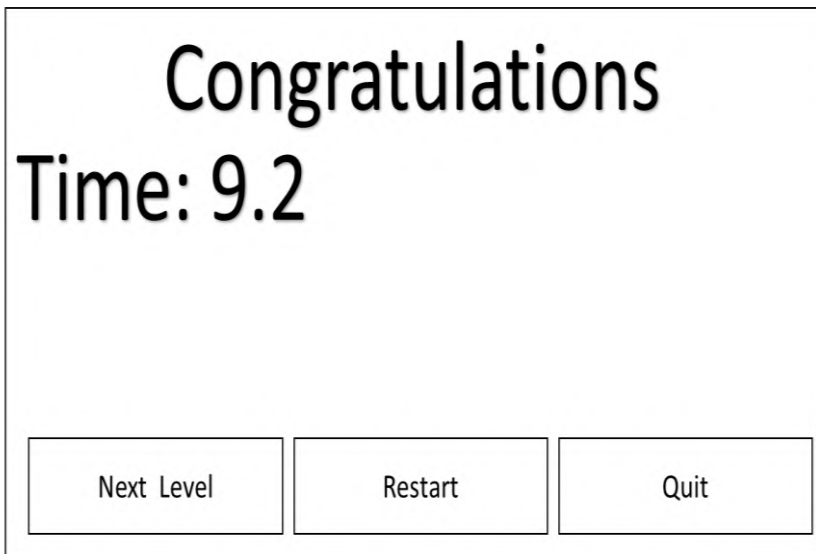
Paused

**Gambar 3.7 Tampilan Permainan saat Pause**

Pada saat bermain, pemain dapat melakukan *pause*. Ketika tombol *Pause* ditekan, karakter pemain dan karakter musuh tidak bisa bergerak dari ruangnya sampai tombol *Resume* ditekan. Ketika permainan dihentikan sementara, pemain juga dapat memilih tombol *Quit* dan *Restart*. Jika pemain memilih tombol *Quit* maupun *Restart*, hasil dari permainan pada level ini akan dianggap sebagai kekalahan untuk pemain. Tampilan permainan dapat dilihat pada Gambar 3.7.

### 3.2.5.4 Tampilan Pemain Menang

Tampilan Pemain Menang merupakan tampilan dari pemain ketika memenangkan permainan. Tampilan ini menampilkan waktu yang dibutuhkan pemain untuk menyelesaikan *game*, tombol *Restart*, dan juga menampilkan tombol Next Level jika belum mencapai level terakhir. Tampilan Pemain Menang ditunjukkan seperti pada Gambar 3.8.



**Gambar 3.8 Tampilan Permainan Berakhir saat menang**

### 3.2.5.5 Tampilan Pemain Kalah

Tampilan Pemain Menang merupakan tampilan dari pemain ketika pemain kalah. Tampilan ini hanya menampilkan tombol *Restart* dan tombol *Quit*. Tampilan Pemain Menang ditunjukkan seperti pada Gambar 3.9.



**Gambar 3.9 Tampilan Permainan Berakhir saat kalah**

### 3.2.6 Perancangan Karakter

Subbab ini membahas mengenai rancangan karakter yang akan digunakan dalam *game*. Karakter dalam *game* ini berbentuk Terdapat 2 macam karakter pada *game* ini yaitu karakter pemain dan karakter bot. Karakter-karakter tersebut diunduh dari *website*. Setiap karakter memiliki bentuk dan animasi yang berbeda. Tampilan karakter pemain diunduh dari *Unity Asset Store* yang kemudian dianimasikan menggunakan *Mixamo*. Tampilan karakter pemain dapat dilihat pada Gambar 3.10. Tampilan karakter bot diunduh dari *tf3dm.com*. Tampilan karakter bot dapat dilihat pada Gambar 3.11



**Gambar 3.10 Karakter Pemain**



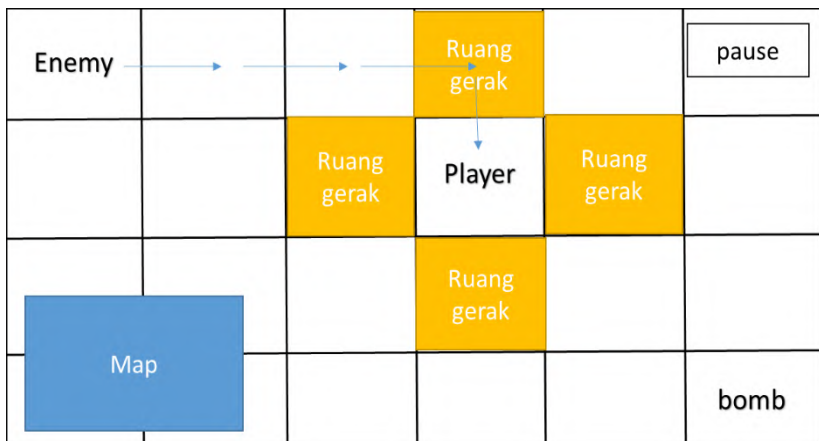
**Gambar 3.11 Karakter Bot**

### 3.3 Perancangan Algoritma dan Kecerdasan Buatan

Dalam subbab ini dibahas perancangan kecerdasan buatan yang akan digunakan dalam permainan dan algoritma yang akan digunakan untuk melakukan optimasi kecerdasan buatan.

#### 3.3.1 Perancangan Kecerdasan Buatan

Dalam tugas akhir ini kecerdasan buatan akan digunakan sebagai lawan dari pemain saat bermain. Kecerdasan buatan pada tugas akhir ini dibangun untuk memilih ruangan mana yang harus dilewati oleh bot. Bot akan bergerak mendekati *player*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.12.



**Gambar 3.12 Ilustrasi pengambilan keputusan**

Pada Gambar 3.12 dapat dilihat bahwa pemain memiliki ruangan yang berbeda dengan musuh. Bot akan bergerak dari satu ruangan ke ruangan yang lain untuk mendekati pemain.

Dalam tugas akhir ini akan dibuat kecerdasan buatan menggunakan algoritma  $A^*$  untuk mencari jarak terdekat bot ke pemain.

### 3.3.2 Perancangan Algoritma

Pada subbab ini akan dibahas tentang algoritma  $A^*$  yang akan diimplementasikan pada bot. Algoritma tersebut akan direpresentasikan ke dalam *array* dan *list*. *Array* tersebut merepresentasikan map pada level. Bot akan mengambil jalan dengan nilai *cost* terkecil, yang berarti jarak terpendek ke pemain.

Jarak tersebut didapatkan dari posisi saat ini pemain dan posisi saat ini musuh. Bot tidak akan berjalan ke karakter pemain sebelum mendapatkan seluruh rute ke pemain. Rute tersebut didapatkan dari perhitungan  $F$  menggunakan *manhattan distance*. Perhitungan  $F$  dijelaskan pada persamaan (3.1).

$$F = G + H \quad (3.1)$$

$G$  = jarak *node* ke Bot,

$H$  = nilai heuristik,

$F$  = nilai *node*

Setiap pencarian *node* dari algoritma, ditentukan  $G$  dan  $H$  dari *node* tersebut. Nilai  $G$  dari *node* didapat dari jarak tempuh *node* tersebut ke posisi *node* bot, sedangkan nilai  $H$  dari *node* didapatkan dari nilai heuristik *node* tersebut terhadap posisi *node* pemain. Perhitungan nilai heuristik *node* didapatkan dari indeks posisi *array* yang merepresentasikan map. Perhitungan nilai  $H$  dijelaskan pada persamaan (3.2).

$$H = (|currentNode.x-player.x|+|currentNode.y-player.y|)*h \quad (3.2)$$

$currentNode.x$  = nilai  $x$  *node* pada bot dari posisi *node* bot

$player.x$  = nilai  $x$  *node* pada pemain dari posisi *node* *player*

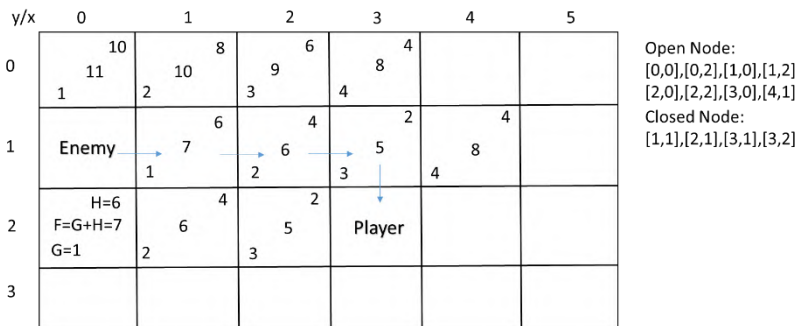
$currentNode.y$  = nilai  $y$  *node* pada bot dari posisi *node* bot

$player.y$  = nilai  $y$  *node* pada bot dari posisi *node* bot

$h$  = koefisien, pada aplikasi ini  $h$  memiliki nilai 2



*Node* selanjutnya dari *node* sekarang akan dimasukkan kedalam *Open Node* yang berupa *list*. *Open node* akan di urutkan agar mendapatkan *node* dengan nilai *F* terkecil. *Node* dengan nilai terkecil akan dimasukkan ke *closed node*, dihapus dari *open node*, dan status dari *node* di *main list* dirubah menjadi *closed* agar tidak dihitung kembali. Proses terus diulang hingga mendapatkan *closed node* berisi rute bot ke pemain. Untuk lebih jelasnya dapat dilihat pada Gambar 3.13.



**Gambar 3.13 Ilustrasi rute node bot ke pemain**

Ketika posisi dari pemain berubah, *open node* dan *close node* akan di hilangkan semua isinya. Status setiap *node* pada *main list* berubah menjadi *unexplored*. Kemudian dilakukan proses *A\** lagi untuk mendapatkan *open node* dan *close node* sesuai dengan posisi bot dan pemain.

### 3.3.3 Perancangan Posisi Pemasangan Bom

Pada setiap permainan terdapat 3 buah ruangan pemasangan bom. Posisi ruangan pemasangan bom didapatkan dari hasil *random* nilai. Meskipun posisi ruangan pemasangan bom di *random*, setiap ruangan pemasangan bom tidak memiliki posisi yang sama atau saling bertumpuk.

### 3.3.4 Perancangan Level

Setiap level memiliki tingkat kesusahan yang berbeda. Semakin tinggi level yang dimainkan maka tingkat kesusahan akan semakin tinggi. Tabel-tabel dibawah menunjukkan detail perancangan tiap level dari level satu sampai tiga dengan kolom yang berisi:

- Level, berisi tingkatan level
- SP, berisi kecepatan gerak pemain dari ruangan satu ke ruangan yang lain
- SE1, berisi kecepatan gerak bot1 dari ruangan satu ke ruangan yang lain
- SE2, berisi kecepatan gerak bot2 dari ruangan satu ke ruangan yang lain
- SE3, berisi kecepatan gerak bot3 dari ruangan satu ke ruangan yang lain.

Nilai minimum, maksimum, dan presentase dari setiap faktor yang mempengaruhi tingkat kesulitan dari level (Persentase F) ditunjukkan pada Tabel 3.1.

**Tabel 3.1 Presentase dari Faktor Tingkat Kesulitan Level**

Faktor	Min	Max	Persentase F
kecepatan karakter pemain	8	6	40
kecepatan karakter bot 1	0	6	20
kecepatan karakter bot 2	0	6	20
kecepatan karakter bot 3	0	6	20

Kemudian, dilakukan perhitungan untuk mencari total nilai tingkat kesulitan tiap level. Perhitungan total nilai tingkat kesulitan level dapat dilihat pada persamaan (3.3). Tabel total nilai tingkat kesulitan tiap level ditunjukkan pada Tabel 3.2.

$$total = \sum_{i=1}^n \frac{F_i - F_{MINi}}{F_{MAXi} - F_{MINi}} * presentase F \quad (3.3)$$

$i$  = index faktor tingkat kesulitan level,

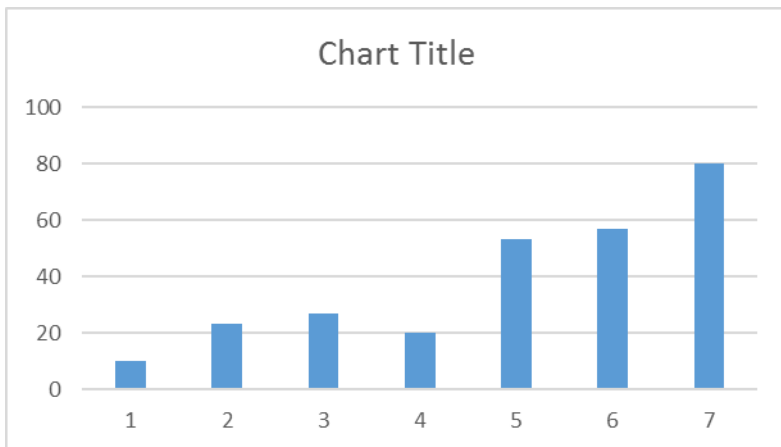
$N$  = Jumlah faktor tingkat kesulitan level

- $F_i$  = Faktor tingkat kesulitan level  
 $FMIN_i$  = Nilai minimum dari faktor tingkat kesulitan level  
 $FMAX_i$  = Nilai maksimum dari faktor tingkat kesulitan level  
 PersentaseF = Persentase pengaruh faktor tingkat kesulitan level

**Tabel 3.2 Total Nilai Tiap Level**

Level	SP	SE1	SE2	SE3	Total
1	8	3	0	0	10
2	8	4	3	0	23.33333
3	8	5	3	0	26.66667
4	8	6	0	0	20
5	7	4	3	3	53.33333
6	7	4	4	3	56.66667
7	6	5	4	3	80

Grafik tingkat kesulitan tiap level dapat dilihat pada Gambar 3.14. Pada grafik tersebut semakin tinggi level maka akan semakin susah. Namun ada beberapa level yang lebih tinggi dari sebelumnya namun lebih mudah. Level tersebut bertujuan untuk memberikan pemain istirahat.



**Gambar 3.14 Grafik Tingkat kesulitan Tiap Level**

*(Halaman ini sengaja dikosongkan)*

## **BAB IV IMPLEMENTASI**

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Di dalamnya mencakup proses penerapan dan pengimplementasian 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.

**Tabel 4.1 Lingkungan Implementasi Perangkat Lunak**

Perangkat Keras	Prosesor: Intel(R) Core(TM) i5-4200U CPU @1.60GHz(4 CPUs), ~ 2.3GHz Memori: 4 GB
Perangkat Lunak	Sistem Operasi: Microsoft Windows 8.1 64-bit Perangkat Pengembang: Unity Game Engine v5.0.0f4 Direct X Version : DirectX 11

### **4.2 Implementasi Permainan**

Implementasi dari masing-masing fungsi utama dituliskan menggunakan kode sumber berdasarkan antarmuka utama yang ada pada permainan. Penjelasan implementasi hanya berupa antar muka yang berhubungan dengan algoritma  $A^*$ , kecerdasan buatan, gerak pemain, penyimpanan hasil permainan, dan penentuan akhir permainan.

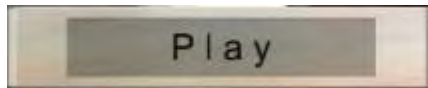
#### **4.2.1 Implementasi *Main menu***

Dalam subbab ini akan dijelaskan tentang pembuatan antarmuka *Main menu*. Tampilan ini memiliki dua tombol yaitu tombol *Play* dan *Quit*. Tampilan dari antarmuka ini ditunjukkan pada Gambar 4.1.



**Gambar 4.1 Implementasi antarmuka *Main menu***

Tombol *Play* digunakan untuk masuk ke antarmuka Level Selector. Tampilan tombol *Play* ditunjukkan pada Gambar 4.2



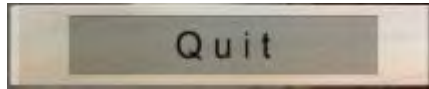
**Gambar 4.2 Tombol *Play***

Implementasi kode sumber untuk tombol *Play* ditunjukkan pada Kode Sumber 4.1.

```
public void GoLevelSelector()
{
    Application.LoadLevel("Level Selector");
}
```

**Kode Sumber 4.1 Tombol *Play***

Tombol *Quit* digunakan untuk keluar dari aplikasi. Tampilan tombol *Quit* ditunjukkan pada Gambar 4.3.



**Gambar 4.3 Tombol *Quit***

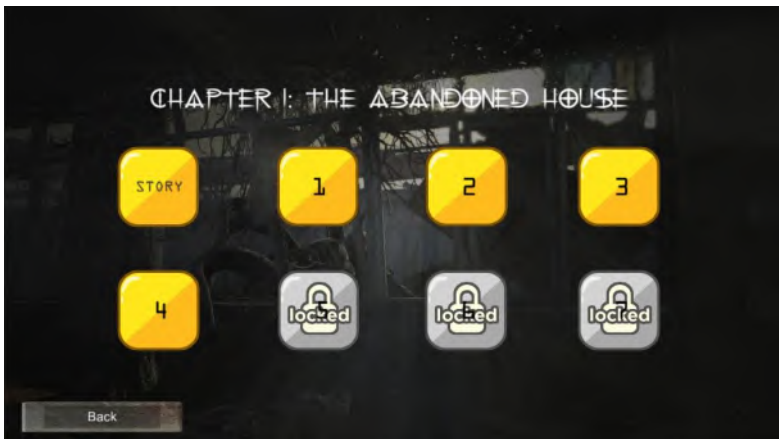
Implementasi kode sumber untuk tombol *Quit* ditunjukkan pada Kode Sumber 4.2.

```
public void Quit()
{
    MasterData.currentLevel = 0;
    Application.Quit();
}
```

**Kode Sumber 4.2 Tombol *Quit***

#### 4.2.2 Implementasi Level Selector

Dalam subbab ini akan dijelaskan tentang pembuatan antarmuka Level Selector. Tampilan ini memiliki tombol Level, tombol *Back*, dan tombol Komik. Ketika pemain pertama kali menjalankan *game* ini, tombol Level hanya terbuka untuk level 1. Pemain harus menyelesaikan level tertinggi untuk mencapai level yang lebih tinggi. Tampilan dari antarmuka ini ditunjukkan pada Gambar 4.4.



**Gambar 4.4 Implementasi antarmuka Level Selector**

Tombol Komik digunakan untuk masuk ke antarmuka Komik. Tampilan tombol Komik ditunjukkan pada Gambar 4.5. Implementasi kode sumber untuk tombol Komik ditunjukkan pada Kode Sumber 4.3.



**Gambar 4.5 Tombol Komik**

```
public void GoToComic()
{
    Application.LoadLevel("Comic"+currentChapter);
}
```

**Kode Sumber 4.3 Tombol Komik**

Tombol Level digunakan untuk memilih level permainan. semakin tinggi level yang dipilih maka permainan akan semakin menantang. Untuk membuka tombol Level yang terkunci, pemain harus menyelesaikan level sebelumnya. Tampilan tombol Level ditunjukkan pada Gambar 4.6. Implementasi kode sumber untuk tombol Level ditunjukkan pada Kode Sumber 4.4.



**Gambar 4.6 Tombol Level**

```
void Start () {
    for (int a = 1; a <= PLayerPrefs.GetInt("Level
Max");a++)
    {
        GameObject.Find("Level"+a).GetComponent<Image>().sprite = open;
    }
}
```



```

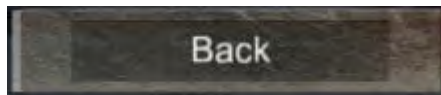
    }
    }
    // Update is called once per frame
    void Update () {

    }
    public void Level1()
    {
        MasterData.currentLevel = 1;
        Application.LoadLevel("Level 1");
    }
    public void Level2()
    {
        if (PlayerPrefs.GetInt("Level Max") >= 2)
        {
            MasterData.currentLevel = 2;
            Application.LoadLevel("Level 2");
        }
    }
    public void Level3()
    {
        if (PlayerPrefs.GetInt("Level Max") >= 3)
        {
            MasterData.currentLevel = 3;
            Application.LoadLevel("Level 3");
        }
    }
}

```

**Kode Sumber 4.4 Tombol Level**

Tombol *Back* digunakan untuk kembali ke antarmuka *Main menu*. Tampilan tombol *Back* ditunjukkan pada Gambar 4.7. Implementasi kode sumber untuk tombol *Back* ditunjukkan pada Kode Sumber 4.5.



**Gambar 4.7 Tombol Back**

```

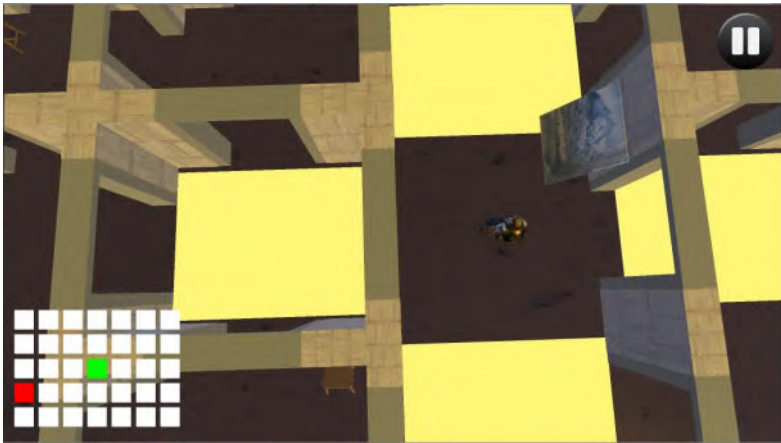
public void GoToMainMenu()
{
    Application.LoadLevel("Main menu");
}

```

**Kode Sumber 4.5 Tombol Back**

### 4.2.3 Implementasi Kontrol Pemain

Dalam subbab ini akan dijelaskan mengenai kontrol pemain. Kontrol pada karakter pemain hanya dengan menggunakan *mouse*. Pada antarmuka permainan akan muncul ruangan dengan lantai kuning menunjukkan bahwa pemain bisa menggerakkan karakter pemain ke ruangan tersebut. Tampilan kontrol pemain dapat dilihat pada Gambar 4.8. Implementasi kode sumber untuk kontrol pemain ditunjukkan pada Kode Sumber 4.6.



**Gambar 4.8 Tampilan kontrol karakter pemain**

```

void Update () {
    if (run)
    {
        pPlayer.transform.position =
        Vector3.MoveTowards(pPlayer.transform.position,
        targetPlayer, moveSpeed * Time.deltaTime);
        if (pPlayer.transform.position !=
        targetPlayer)
        {
            pPlayerAnimator.Play("run");
        }
    }
    if(pPlayer.transform.position==targetPlayer)
  
```

```

        {
            targetPlayer=target;
        }
        if (currentSpot.xSpot != 0 &&
currentSpot.xSpot != (longMaze - 1) &&
currentSpot.ySpot != 0 && currentSpot.ySpot !=
(wideMaze - 1))
        {
            spotUp =
GameObject.Find(map[currentSpot.xSpot - 1,
currentSpot.ySpot]);
            spotUp.transform.localScale = new
Vector3(4, 0.1f, 3);
            spotDown =
GameObject.Find(map[currentSpot.xSpot + 1,
currentSpot.ySpot]);
            spotDown.transform.localScale = new
Vector3(4, 0.1f, 3);
            spotLeft =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot - 1]);
            spotLeft.transform.localScale = new
Vector3(4, 0.1f, 3);
            spotRight =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot + 1]);
            spotRight.transform.localScale = new
Vector3(4, 0.1f, 3);
        }
        else if (currentSpot.xSpot == 0 &&
currentSpot.ySpot == 0)
        {
            spotRight =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot + 1]);
            spotRight.transform.localScale = new
Vector3(4, 0.1f, 3);
            spotDown =
GameObject.Find(map[currentSpot.xSpot + 1,
currentSpot.ySpot]);

```

```

        spotDown.transform.localScale = new
Vector3(4, 0.1f, 3);
    }
    else if (currentSpot.xSpot == (longMaze -
1) && currentSpot.ySpot == (wideMaze - 1))
    {
        spotUp =
GameObject.Find(map[currentSpot.xSpot - 1,
currentSpot.ySpot]);
        spotUp.transform.localScale = new
Vector3(4, 0.1f, 3);
        spotLeft =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot - 1]);
        spotLeft.transform.localScale = new
Vector3(4, 0.1f, 3);
    }
    else if (currentSpot.ySpot == 0 &&
currentSpot.xSpot == longMaze - 1)
    {
        spotUp =
GameObject.Find(map[currentSpot.xSpot - 1,
currentSpot.ySpot]);
        spotUp.transform.localScale = new
Vector3(4, 0.1f, 3);
        spotRight =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot + 1]);
        spotRight.transform.localScale = new
Vector3(4, 0.1f, 3);
    }
    else if (currentSpot.xSpot == 0 &&
currentSpot.ySpot == (wideMaze - 1))
    {
        spotLeft =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot - 1]);
        spotLeft.transform.localScale = new
Vector3(4, 0.1f, 3);
    }

```

```

        spotDown =
GameObject.Find(map[currentSpot.xSpot + 1,
currentSpot.ySpot]);
        spotDown.transform.localScale = new
Vector3(4, 0.1f, 3);
    }
    else if (currentSpot.ySpot == 0)
    {
        spotUp =
GameObject.Find(map[currentSpot.xSpot - 1,
currentSpot.ySpot]);
        spotUp.transform.localScale = new
Vector3(4, 0.1f, 3);
        spotRight =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot + 1]);
        spotRight.transform.localScale = new
Vector3(4, 0.1f, 3);
        spotDown =
GameObject.Find(map[currentSpot.xSpot + 1,
currentSpot.ySpot]);
        spotDown.transform.localScale = new
Vector3(4, 0.1f, 3);
    }
    else if (currentSpot.xSpot == 0)
    {
        spotLeft =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot - 1]);
        spotLeft.transform.localScale = new
Vector3(4, 0.1f, 3);
        spotDown =
GameObject.Find(map[currentSpot.xSpot + 1,
currentSpot.ySpot]);
        spotDown.transform.localScale = new
Vector3(4, 0.1f, 3);
        spotRight =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot + 1]);
        spotRight.transform.localScale = new

```

```

Vector3(4, 0.1f, 3);
    }
    else if (currentSpot.ySpot == (wideMaze -
1))
    {
        spotUp =
GameObject.Find(map[currentSpot.xSpot - 1,
currentSpot.ySpot]);
        spotUp.transform.localScale = new
Vector3(4, 0.1f, 3);
        spotDown =
GameObject.Find(map[currentSpot.xSpot + 1,
currentSpot.ySpot]);
        spotDown.transform.localScale = new
Vector3(4, 0.1f, 3);
        spotLeft =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot - 1]);
        spotLeft.transform.localScale = new
Vector3(4, 0.1f, 3);
    }
    else if (currentSpot.xSpot == (longMaze -
1))
    {
        spotUp =
GameObject.Find(map[currentSpot.xSpot - 1,
currentSpot.ySpot]);
        spotUp.transform.localScale = new
Vector3(4, 0.1f, 3);
        spotLeft =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot - 1]);
        spotLeft.transform.localScale = new
Vector3(4, 0.1f, 3);
        spotRight =
GameObject.Find(map[currentSpot.xSpot,
currentSpot.ySpot + 1]);
        spotRight.transform.localScale = new
Vector3(4, 0.1f, 3);
    }

```

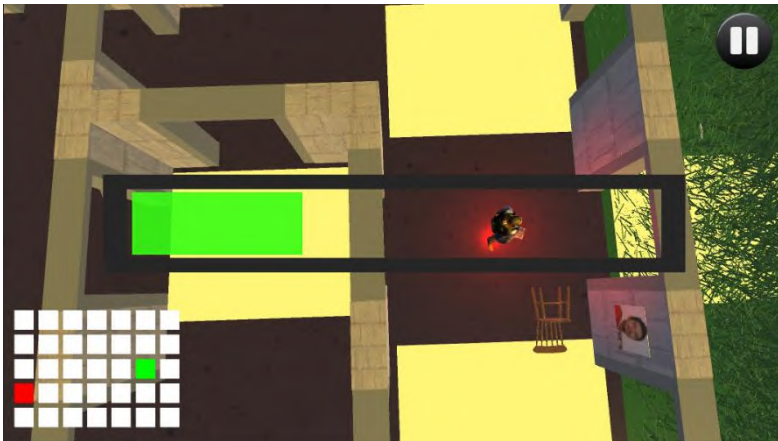
```

    }
    else
    {
        foreach (Spot x in spots)
        {
            GameObject temporaryGb =
GameObject.Find(x.spotName);
            temporaryGb.transform.localScale =
new Vector3(0, 0, 0);
        }
    }

```

**Kode Sumber 4.6 Kontrol pemain**

Pemain dituntut menemukan semua tempat pemasangan bom dan memasang bom untuk memenangkan permainan. Setelah menemukan tempat pemasangan bom, pemain harus menunggu selama waktu yang ditentukan di ruangan tersebut agar bom terpasang. Jika pemain pindah dari ruangan sebelum bom terpasang maka waktu pemain untuk memasang bom di ruangan tersebut dimulai dari awal. Tampilan pemasangan bom dapat dilihat pada Gambar 4.9. Implementasi kode sumber untuk pemasangan bom ditunjukkan pada Kode Sumber 4.7.



**Gambar 4.9 Pemasangan bom**

```

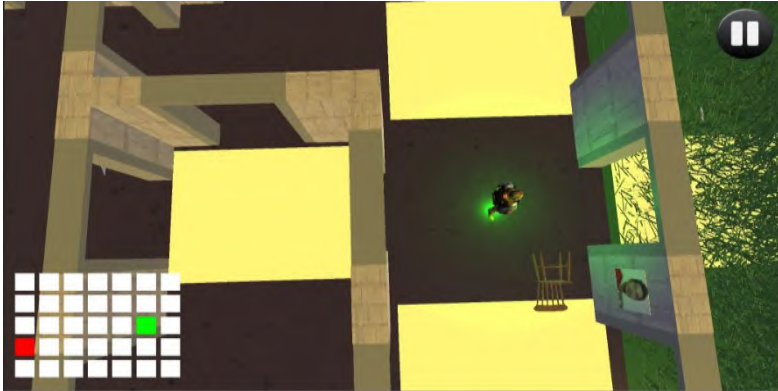
if (exitOnBomb == false)
    {
        if ((currentTime + 0.75f) <=
Time.time)
            {
                GameObject.Find("b11").transform.localScale = new
                Vector3(1, 1, 1);
            }
        if ((currentTime + 1.5f) <=
Time.time)
            {
                GameObject.Find("b12").transform.localScale = new
                Vector3(1, 1, 1);
            }
        if ((currentTime + 2.25f) <=
Time.time)
            {
                GameObject.Find("b13").transform.localScale = new
                Vector3(1, 1, 1);
            }
        if ((currentTime + 2.5f) <=
Time.time)
            {
                bombPlanted += 1;
                exitOnBomb=true;
                closeBombLoading();
                Light currentBombLight =
                currentBomb.GetComponent<Light>();
                currentBombLight.color =
                Color.green;
            }
    }

```

**Kode Sumber 4.7 Pemasangan bom**

Ketika pemain berhasil memasang bom, warna lampu disekitar objek bom akan berubah menjadi warna hijau. Tampilan objek bom berhasil dipasang dapat dilihat pada Gambar 4.10.

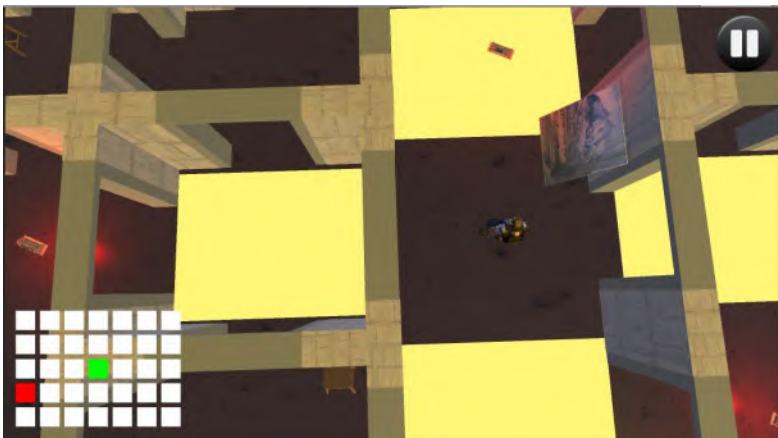




**Gambar 4.10 Bom berhasil dipasang**

#### **4.2.4 Implementasi Posisi Bom**

Dalam subbab ini akan dijelaskan tentang penempatan posisi pemasangan bom. Dalam setiap level terdiri dari tiga tempat pemasangan bom yang diacak posisinya. Setiap bom memiliki posisi yang berbeda. Posisi bom dapat dilihat pada Gambar 4.11. Implementasi kode sumber untuk peletakan tempat pemasangan bom ditunjukkan pada Kode Sumber 4.8



**Gambar 4.11 Posisi bom**

```

    void RandomPosition()
    {
        random = Random.Range(1, 35);
        x = GameObject.Find("Spot" +
random).transform.localPosition.x;
        z = GameObject.Find("Spot" +
random).transform.localPosition.z;
        this.transform.localPosition = new Vector3(x
+ 4.6f, -4.85f, z + 2.81f);
    }
    void SeparateBomb()
    {
        foreach (GameObject bomb in bombs)
        {
            if (bomb.name != this.name)
            {
                if (bomb.transform.localPosition ==
this.transform.localPosition)
                {
                    RandomPosition();
                }
            }
        }
    }
}

```

**Kode Sumber 4.8 Peletakan tempat pemasangan bom**

#### 4.2.5 Implementasi Kecerdasan Buatan

Dalam subbab ini akan dijelaskan tentang implementasi kecerdasan buatan dalam permainan dan fungsi-fungsi yang berhubungan dengan hal tersebut. Hal pertama yang harus dilakukan adalah merepresentasikan map kedalam sebuah *list*. Representasi map ke dalam *list* dapat dilihat pada Kode Sumber 4.9.

```

for (int a = 0; a < longMaze; a++)
    for (int b = 0; b < wideMaze; b++)
    {
        map[a, b] =

```

```

enemySpot.transform.GetChild(child).name;
    if
(enemySpot.transform.GetChild(child).name != "wall")
    {
        enemySpots.Add(new EnemySpot()
        {
            enemySpotName =
enemySpot.transform.GetChild(child).name,
            arrayX = a,
            arrayY = b,
            isWalkAble = true,
            enemySpotPosition =
enemySpot.transform.GetChild(child).position,
            enemySpotStatus =
EnemySpotStatus.untested
        });
    }
    else
    {
        enemySpots.Add(new EnemySpot()
        {
            enemySpotName =
enemySpot.transform.GetChild(child).name,
            arrayX = a,
            arrayY = b,
            isWalkAble = false,
            enemySpotPosition =
enemySpot.transform.GetChild(child).position,
            enemySpotStatus =
EnemySpotStatus.closed
        });
    }
    child++;
}

```

**Kode Sumber 4.9 Representasi map ke dalam list**

Kemudian, mendapatkan posisi pemain sebagai *end node*, dan posisi bot sebagai *start node*. Mendapatkan posisi pemain sebagai *end node* dapat dilihat pada Kode Sumber 4.10.

```

pLayer =
GameObject.Find("Player").GetComponent<Player>();
pLayerGB = GameObject.Find("Player");
endX = pLayer.currentSpot.xSpot;
endY = pLayer.currentSpot.ySpot;

```

#### Kode Sumber 4.10 Mendapatkan posisi pemain

Mendapatkan posisi bot sebagai *start node* dapat dilihat pada Kode Sumber 4.11.

```

void OnTriggerEnter(Collider gb)
{
    if (gb.gameObject.tag == "EnemySpot")
    {
        currentEnemySpot = enemySpots.Find(x =>
x.enemySpotName == gb.name);
    }
}

```

#### Kode Sumber 4.11 Mendapatkan posisi bot

Untuk mendapatkan *node* selanjutnya yang harus dilewati, dilakukan perhitungan F setiap *node* disekitar posisi bot. Perhitungan nilai F dari *node* sebelah kanan dari posisi bot dapat dilihat pada Kode Sumber 4.12.

```

public void getRight()
{
    //calculate F
    int ha = Math.Abs(currentNode.arrayX - endX);
    int hb = Math.Abs((currentNode.arrayY+1) -
endY);
    int h = (ha + hb) * 2;
    float thisF = (currentNode.G + 1f) + h;
    EnemySpot nextNode = enemySpots.Find(x =>
x.arrayX == currentNode.arrayX &&
x.arrayY==(currentNode.arrayY+1));

    if (nextNode != null)
    {
        //Debug.Log("right " +

```

```

nextNode.enemySpotName);
    if (nextNode.arrayY < wideMaze &&
nextNode.isWalkAble == true &&
nextNode.enemySpotStatus != EnemySpotStatus.closed)
    {
        openNode.Add(new EnemySpot()
        {
            enemySpotName =
map[nextNode.arrayX, nextNode.arrayY],
            arrayX = nextNode.arrayX,
            arrayY = nextNode.arrayY,
            F = thisF,
            G = currentNode.G + 1,
            parentNode = currentNode,
            isWalkAble = true,
            enemySpotStatus =
EnemySpotStatus.untested,
            enemySpotPosition =
GameObject.Find(map[nextNode.arrayX,
nextNode.arrayY]).transform.position
        });
        UpdateMainList(nextNode, enemySpots);
    }
}
}
}

```

### Kode Sumber 4.12 Mendapatkan nilai F dari *node* kanan

Perhitungan nilai F dari *node* sebelah kiri dari posisi bot dapat dilihat pada Kode Sumber 4.13.

```

public void getLeft()
{
    int ha = Math.Abs(currentNode.arrayX - endX);
    int hb = Math.Abs((currentNode.arrayY - 1) -
endY);
    int h = (ha + hb) * 2;
    float thisF = (currentNode.G + 1f) + h;
    EnemySpot nextNode = enemySpots.Find(x =>

```

```

x.arrayX == currentNode.arrayX && x.arrayY ==
(currentNode.arrayY - 1));
    if (nextNode != null)
    {
        //Debug.Log("left " +
nextNode.enemySpotName);
        if (nextNode.arrayY >= 0 &&
nextNode.isWalkAble == true &&
nextNode.enemySpotStatus != EnemySpotStatus.closed)
        {
            openNode.Add(new EnemySpot()
            {
                enemySpotName =
map[nextNode.arrayX, nextNode.arrayY],
                arrayX = nextNode.arrayX,
                arrayY = nextNode.arrayY,
                F = thisF,
                G = currentNode.G + 1,
                parentNode = currentNode,
                isWalkAble = true,
                enemySpotStatus =
EnemySpotStatus.untested,
                enemySpotPosition =
GameObject.Find(map[nextNode.arrayX,
nextNode.arrayY]).transform.position
            });
            UpdateMainList(nextNode, enemySpots);
        }
    }
}

```

**Kode Sumber 4.13 Mendapatkan nilai F dari *node* kiri**

Perhitungan nilai F dari *node* sebelah atas dari posisi bot dapat dilihat pada Kode Sumber 4.14.

```

public void getTop()
{
    int ha = Math.Abs((currentNode.arrayX-1) -
endX);
    int hb = Math.Abs(currentNode.arrayY - endY);
}

```

```

        int h = (ha + hb) * 2;
        float thisF = (currentNode.G + 1f) + h;
        EnemySpot nextNode = enemySpots.Find(x =>
x.arrayX == (currentNode.arrayX-1) && x.arrayY ==
currentNode.arrayY);
        if (nextNode != null)
        {
            //Debug.Log("top " + nextNode.enemySpotName);
            if (nextNode.arrayX >= 0 &&
nextNode.isWalkAble == true &&
nextNode.enemySpotStatus != EnemySpotStatus.closed)
                {
                    openNode.Add(new EnemySpot()
                    {
                        enemySpotName =
map[nextNode.arrayX, nextNode.arrayY],
                        arrayX = nextNode.arrayX,
                        arrayY = nextNode.arrayY,
                        F = thisF,
                        G = currentNode.G + 1,
                        parentNode = currentNode,
                        isWalkAble = true,
                        enemySpotStatus =
EnemySpotStatus.untested,
                        enemySpotPosition =
GameObject.Find(map[nextNode.arrayX,
nextNode.arrayY]).transform.position
                    });
                    UpdateMainList(nextNode, enemySpots);
                }
        }
    }
}

```

**Kode Sumber 4.14 Mendapatkan nilai F dari *node* atas**

Perhitungan nilai F dari *node* sebelah bawah dari posisi bot dapat dilihat pada Kode Sumber 4.15.

```

public void getBottom()
{
    int ha = Math.Abs((currentNode.arrayX + 1) -

```

```

endX);
    int hb = Math.Abs(currentNode.arrayY - endY);
    int h = (ha + hb) * 2;
    float thisF = (currentNode.G + 1f) + h;
    EnemySpot nextNode = enemySpots.Find(x =>
x.arrayX == (currentNode.arrayX + 1) && x.arrayY ==
currentNode.arrayY);
    if (nextNode != null)
    {
        //Debug.Log("bottom " +
nextNode.enemySpotName);
        if (nextNode.arrayX < longMaze &&
nextNode.isWalkAble == true &&
nextNode.enemySpotStatus != EnemySpotStatus.closed)
            {
                openNode.Add(new EnemySpot()
                {
                    enemySpotName =
map[nextNode.arrayX, nextNode.arrayY],
                    arrayX = nextNode.arrayX,
                    arrayY = nextNode.arrayY,
                    F = thisF,
                    G = currentNode.G + 1,
                    parentNode = currentNode,
                    isWalkAble = true,
                    enemySpotStatus =
EnemySpotStatus.untested,
                    enemySpotPosition =
GameObject.Find(map[nextNode.arrayX,
nextNode.arrayY]).transform.position
                });
                UpdateMainList(nextNode, enemySpots);
            }
    }
}
}
}

```

#### Kode Sumber 4.15 Mendapatkan nilai F dari *node* bawah

Nilai F dari node yang telah didapat akan dimasukkan ke dalam *open node*. Node dengan nilai F terkecil akan dijadikan



posisi *node* terbaru dari bot. Posisi *node* sebelumnya dari bot akan dirubah statusnya menjadi *closed* agar tidak dimasukkan kedalam *open node* untuk dijadikan pertimbangan *node* selanjutnya. Proses pencarian *node* dengan nilai F terkecil diulang hingga mendapatkan *end node*. Setiap *node* yang menjadi posisi *node* dari bot akan dimasukkan ke dalam *route node* yang berisi *node* dengan informasi ruangan yang akan dilewati bot pada permainan. Implementasi proses tersebut dapat dilihat pada Kode Sumber 4.16.

```

public void Astar()
{
    if
(currentNode.enemySpotName.Substring(currentNode.enemySpotName.Length - 2) ==
player.currentSpot.spotName.Substring(player.currentSpot.spotName.Length - 2))
    {
        collide = true;
    }
    getTop();
    getRight();
    getBottom();
    getLeft();
    List<EnemySpot> sortedList = openNode.OrderBy(o
=> o.F).ToList();
    currentNode.enemySpotStatus =
EnemySpotStatus.closed;
    UpdateMainList(currentNode, enemySpots);
    routeNode.Add(currentNode);
    currentNode = sortedList[0];
}

```

**Kode Sumber 4.16 Mencari *route node***

*Route node* akan dihapus ketika karakter pemain berpindah ruangan. Kemudian dilakukan proses pencarian *route node* kembali dengan *end node* posisi terbaru karakter pemain.

Implementasi proses tersebut dapat dilihat pada Kode Sumber 4.17.

```

if (player.prevPositionPlayer !=
player.currentSpot.spotName )
    {
        player.prevPositionPlayer =
player.currentSpot.spotName;
        endX = player.currentSpot.xSpot;
        endY = player.currentSpot.ySpot;
        routeNode.Clear();
        openNode.Clear();
        currentIndexOfRoute = 0;
        collide = false;
        currentNode = currentEnemySpot;
        foreach(EnemySpot status in enemySpots){
            status.enemySpotStatus =
EnemySpotStatus.untested;
        }
    }

```

**Kode Sumber 4.17 Mengganti route node**

Bot akan bergerak mengejar pemain seperti yang terlihat pada Gambar 4.12.

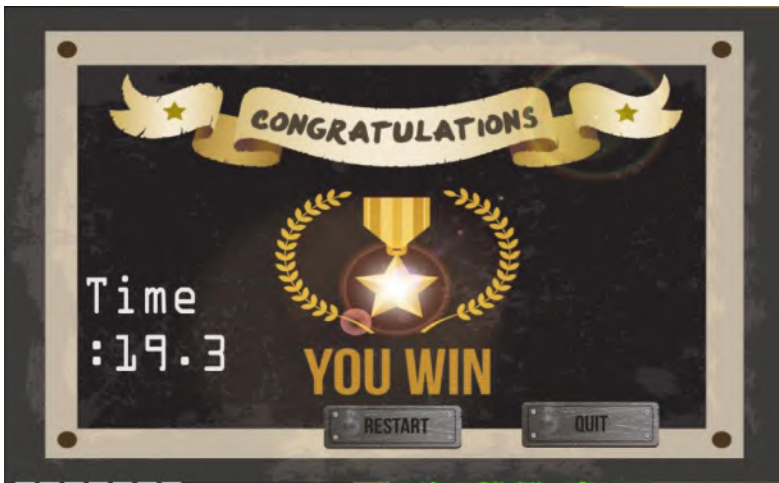


**Gambar 4.12 Bot mengejar pemain**

#### 4.2.6 Implementasi Menang Atau Kalah

Dalam sebuah *game* perlu adanya cara untuk mengetahui menang atau kalah. Kode sumber di bawah ini adalah kode sumber yang digunakan untuk melakukan pengecekan apakah pemain menang atau kalah.

Jika pemain berhasil mengaktifkan tiga bom dalam *maze* maka pemain akan dianggap menang. Tampilan pemain menang dapat dilihat pada Gambar 4.13.



Gambar 4.13 Tampilan Menang

Implementasi pemain menang dapat dilihat pada Kode Sumber 4.18.

```

if (bombPlanted == 3) {
    inGameMenu.Win();
    if(MasterData.currentLevel+1<MasterData.maxLevel){
        GameObject.Find("Nextlevel").transform.localScale =
        new Vector3(1, 1, 1);
    }
    else {
        GameObject.Find("Nextlevel").transform.localScale =
        new Vector3(0, 0, 0);}}

```

Kode Sumber 4.18 Aturan Menang

Jika pemain tertangkap oleh bot maka pemain kalah. Tampilan pemain kalah dapat dilihat pada Gambar 4.14.



**Gambar 4.14 Tampilan pemain kalah**

Implementasi pemain kalah dapat dilihat pada Kode Sumber 4.19.

```
void OnTriggerEnter(Collider gb)
{
if (gb.gameObject.name == "Player")
{
    enemyAnimator.Play("Idle Firing");
    player.playerAnimator.Play("killed");
    run = false;
    player.run = false;
    gameOverAnimator.Play("Game Over");
}}
}
```

**Kode Sumber 4.19 Aturan Kalah**

## 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**

Perangkat Keras	Prosesor: Intel(R) Core(TM) i5-4200U CPU @1.60GHz(4 CPUs), ~ 2.3GHz Memori: 4 GB
Perangkat Lunak	Sistem Operasi: Microsoft Windows 8.1 64-bit Unity Game Engine v5.0.0f4

### 5.2 Pengujian Fungsionalitas

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* [8].

Skenario Pengujian fungsionalitas digunakan untuk memberikan tahap-tahap dalam pengujian sistem. Skenario ini tertera pada Tabel 5.2.

**Tabel 5.2 Pengujian Permainan**

Kondisi Awal	Pengguna berada pada layar Judul 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.

### 5.2.1 Pengujian Layar Menu Utama

Pengujian dimulai ketika pemain telah masuk ke layar Menu Utama seperti yang tercantum pada Gambar 5.1. Pemain dapat memilih untuk menekan salah satu dari beberapa tombol yang disediakan untuk berpindah ke layar yang lain.

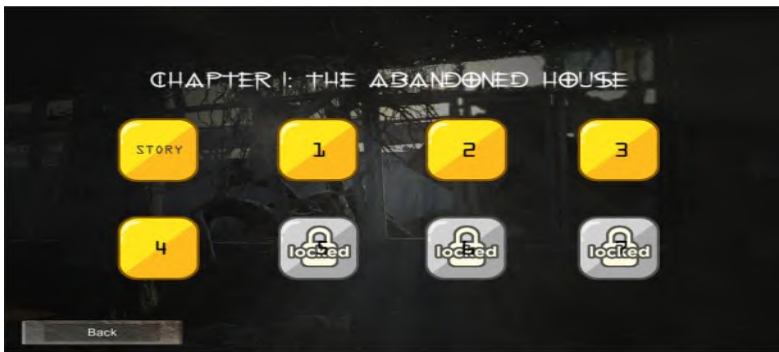


**Gambar 5.1 Tampilan layar Menu Utama**

Setelah memulai permainan, layar Menu Utama akan muncul. Tampilan dan semua tombol pada tampilan ini berfungsi dengan baik.

### 5.2.2 Pengujian Layar Level Selector

Pengujian dimulai ketika pemain masuk ke layar Pemilihan Level dengan cara menekan tombol Start pada layar Menu Utama.



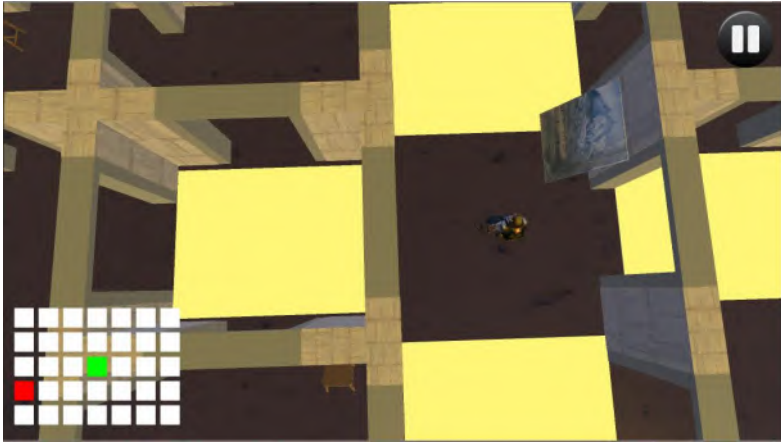
**Gambar 5.2 Tampilan layar Level**

Pada tampilan ini disajikan level-level yang dapat dimainkan oleh pemain. Pada awalnya level yang terbuka hanya level pertama. Pemain hanya dapat memilih level yang terbuka.

Kesimpulan fungsionalitas pada layar ini adalah berhasil dikarenakan tombol, sistem, dan skenario sudah diuji dan hasilnya sesuai dengan yang diharapkan.

### 5.2.3 Pengujian Layar Permainan

Skenario pengujian dimulai ketika pemain sudah memilih level yang ingin dimainkan. Pemain akan dibawa ke layar permainan seperti terlihat pada Gambar 5.3. Pada layar ini pemain dapat melakukan *pause* dengan menekan tombol *Pause*. Tampilan layar permainan saat *pause* dapat dilihat pada Gambar Gambar 5.4.



**Gambar 5.3** Tampilan layar Permainan



**Gambar 5.4** Tampilan layar permainan saat *pause*

Pada layar ini juga pemain dapat memilih ruangan untuk berpindah tempat. Ketika pemain memilih ruangan berlantai kuning, pemain akan berpindah tempat. Hal tersebut dapat dilihat pada Gambar 5.5.





**Gambar 5.5 Pemain bergerak ke ruangan lain**

Pada layar ini bot (kecerdasan buatan) berhasil diimplementasikan dengan algoritma  $A^*$  karena bot berhasil mengejar pemain. Hal ini dibuktikan dari Gambar 5.6.

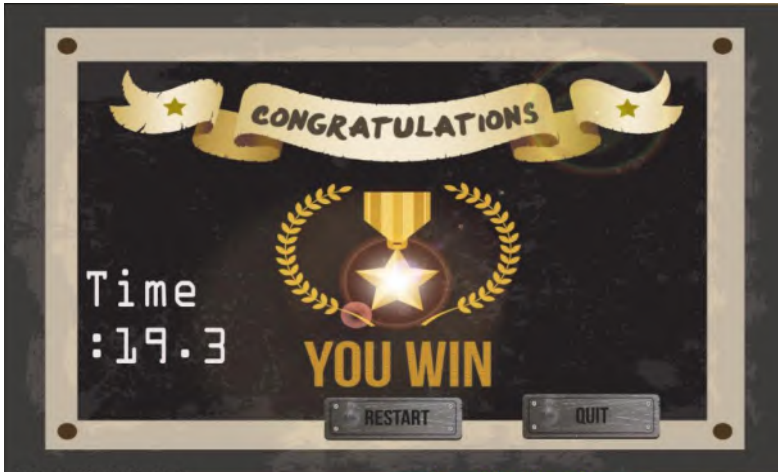


**Gambar 5.6 Bot mengejar pemain**

Kesimpulan untuk fungsionalitas pada layar Permainan adalah berhasil dikarenakan tombol, sistem, teks, dan skenario sudah diuji dan hasilnya sesuai dengan yang diharapkan. Tetapi, implementasi algoritma A\* pada bot belum optimal sehingga ada waktu dimana kecepatan gerak pemain lebih cepat daripada kecepatan gerak bot. Hal ini mengakibatkan ada waktu dimana bot tidak mendapatkan jarak terpendek ke pemain.

#### 5.2.4 Pengujian Layar Hasil Permainan

Pengujian dimulai ketika pemain sudah berhasil mengalahkan sebuah level atau kalah pada sebuah level. Sistem akan membawa pemain ke layar Menang atau layar Kalah. Tampilan pada layar ini bergantung pada hasil permainan. Tampilan menang berhasil ditampilkan ketika pemain menang seperti pada Gambar 5.7.



**Gambar 5.7 Tampilan hasil permainan ketika menang**

Tampilan Kalah berhasil ditampilkan ketika pemain kalah seperti pada Gambar 5.8.



**Gambar 5.8** Tampilan hasil permainan ketika kalah

### 5.2.5 Hasil Pengujian Fungsional

Rekap hasil pengujian fungsionalitas dicantumkan pada Tabel 5.3.

**Tabel 5.3** Hasil Pengujian Fungsionalitas

No	Pengujian	Hasil Pengujian
1.	Pengujian Layar Menu Utama	Berhasil
2.	Pengujian Layar Level Selector	Berhasil
3.	Pengujian Layar Permainan	Berhasil
4.	Pengujian Layar Hasil Permainan	Berhasil

### **5.3 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.

#### **5.3.1 Skenario Uji Coba Pengguna**

Dalam melakukan pengujian perangkat lunak, penguji diminta mencoba menggunakan perangkat lunak untuk mencoba semua fungsionalitas dan fitur yang ada.

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 lima 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.

#### **5.3.2 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 5.4.

**Tabel 5.4 Daftar Nama Penguji**

<b>Nomor</b>	<b>Nama</b>	<b>Pekerjaan</b>
1	Ignatius Abraham Susanto	Mahasiswa Teknik Informatika ITS
2	Demsey Iman M	Mahasiswa Teknik Informatika ITS
3	Dimas Riskahadi	Mahasiswa Teknik Informatika ITS
4	Aditya Putra Ferza	Mahasiswa Teknik Informatika ITS
5	Zulwaqor Maulana	<i>Offshore Engiineer</i>

### **5.3.3 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 empat menunjukkan skala tertinggi. Penilaian akhir kemudian dilakukan dengan menghitung berapa banyak penguji yang memilih suatu skala tertentu dan kemudian dicari nilai rata-ratanya. Hasil uji coba dipaparkan secara lengkap dengan disertai Tabel yang dapat dilihat pada subbab.

#### **5.3.3.1 Hasil Penilaian Antarmuka**

Penilaian antarmuka difokuskan pada penilaian pengguna terhadap kemudahan penggunaan antarmuka dan sifat-sifat lain yang perlu dimiliki. Pengguna akan menilai apakah tampilan aplikasi memudahkan pengguna dalam memainkan aplikasi. Hasil penilaian pengguna terhadap antarmuka aplikasi dapat dilihat pada Tabel 5.5.

**Tabel 5.5 Penilaian Antarmuka**

No.	Antarmuka	Penilaian				
		Sangat tidak setuju	Tidak setuju	Cukup setuju	Setuju	Sangat setuju
1	Aplikasi mudah digunakan	0	0	0	5	0
2	Antarmuka aplikasi memiliki desain bagus	0	0	1	3	1
3	Kecepatan pemilihan menu/fitur cukup	0	0	0	4	1
4	Tema <i>game</i> dengan tema yang diinginkan pengembang cukup sesuai	0	0	1	3	1
5	Adanya ketertarikan untuk bermain <i>game</i>	0	0	0	3	2

### 5.3.3.2 Hasil Penilaian Performa Sistem, Kenyamanan, dan *Fun*

Penilaian performa sistem difokuskan pada penilaian pengguna terhadap kemampuan aplikasi dalam menghasilkan performa dari interaksi pengguna. Penilaian ini juga ditujukan untuk mendapatkan tingkat kecepatan dan kelancaran sistem atas interaksi yang dibuat oleh pengguna. Hasil penilaian performa sistem dapat dilihat pada Tabel 5.6.

**Tabel 5.6 Penilaian Performa Sistem**

No	Performa Sistem	Penilaian				
		Sangat tidak setuju	Tidak setuju	Cukup setuju	Setuju	Sangat setuju
1	Animasi lancar ketika <i>game</i> dimainkan	0	0	1	2	2
2	Perpindahan menu lancar	0	0	1	2	2
3	Pemain menikmati <i>game</i>	0	0	0	2	3

### 5.3.3.3 Hasil Penilaian Kecerdasan Buatan

Penilaian kecerdasan buatan difokuskan pada penilaian pengguna terhadap kemampuan bot dalam memberikan tantangan lebih kepada pemain. Penilaian ini juga ditujukan untuk mendapatkan tingkat akurasi bot dalam mencari rute terpendek. Hasil penilaian performa bot dapat dilihat pada Tabel 5.7.

**Tabel 5.7 Penilaian Kecerdasan Buatan**

No	Performa Sistem	Penilaian				
		Sangat tidak setuju	Tidak Setuju	Cukup setuju	Setuju	Sangat setuju
1	Bot memberikan tantangan lebih kepada pemain	0	0	0	2	3
2	Rute yang dilewati bot ke pemain adalah rute terpendek	0	0	0	2	3

### 5.3.3.4 Hasil Pengujian Pengguna

Evaluasi pengujian pengguna dilakukan dengan menampilkan data rekapitulasi perangkat lunak yang telah dipaparkan. Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Tabel 5.8. Pada rekapitulasi ini mayoritas pengguna menilai banyak aspek yang perlu ditingkatkan pada aplikasi ini, khususnya antarmuka aplikasi. Antarmuka aplikasi dinilai masih kurang bagus dan kurang memudahkan pengguna dalam memainkan aplikasi permainan.

**Tabel 5.8 Rekapitulasi Hasil Uji Coba Pengguna**

No.	Nama Pengujian		Rata-Rata
1	Penilaian Antarmuka	Aplikasi mudah digunakan	4

No.	Nama Pengujian		Rata-Rata
1	Penilaian Antarmuka	Antarmuka aplikasi memiliki desain yang bagus	4
		Kecepatan pemilihan menu/fitur cukup	4,2
		Tema <i>game</i> dengan tema yang diinginkan oleh pengembang cukup sesuai	4
		Adanya ketertarikan untuk bermain <i>game</i>	4,4
2	Penilaian Performa, Kenyamanan, dan <i>fun</i>	Animasi lancar ketika <i>game</i> dimainkan	4,2
		Perpindahan menu lancar	4,2
		Kelancaran animasi	4,6
3	Penilaian Kecerdasan Buatan	Bot memberikan tantangan lebih kepada pemain	4,6
Rute yang dilewati bot ke pemain adalah rute terpendek		4,6	

#### 5.4 Evaluasi Pengujian

Subbab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Dalam hal ini, evaluasi menunjukkan data rekapitulasi dari hasil pengujian yang telah dilakukan sebelumnya. Berdasarkan hasil pengujian fungsional oleh penulis, pengujian kegunaan oleh pengguna berdasarkan kuesioner dan pendataan performa yang telah dilaksanakan dijelaskan pada subbab sebelumnya, maka didapatkan evaluasi sebagai berikut:



1. Berdasarkan kuesioner, mayoritas pengguna menilai tampilan aplikasi cukup baik namun dapat ditingkatkan kembali. Hal ini dapat dibuktikan dari hasil kuesioner pada Tabel 5.8. Pengguna menambahkan perlunya tampilan tutorial untuk aplikasi ini.
2. Mayoritas pengguna menilai tingkat kenyamanan saat memainkan permainan ini cukup. Hal ini dapat dibuktikan dari hasil kuesioner pada Tabel 5.8.
3. Mayoritas pengguna menilai implementasi algoritma A\* pada kecerdasan buatan pada permainan ini sangat baik. Hal ini dapat dibuktikan dari hasil kuesioner pada Tabel 5.8.
4. Pada permainan ini masih banyak hal yang harus diperbaiki seperti animasi yang perlu diperhalus kembali, optimasi pada kecerdasan buatan agar lebih cepat waktu memproses *path*, dan *chapter* yang perlu ditambah agar pemain tidak bosan bermain dalam lingkungan yang selalu sama.
5. Cerita bergambar pada permainan ini perlu ditingkatkan kembali karena hal tersebut dapat menjadi salah satu daya tarik bagi pemain untuk terus memainkan permainan ini hingga mengetahui akhir dari cerita.
6. Skor untuk pemain perlu ditingkatkan lagi seperti menambahkan koin pada setiap level.

*(Halaman ini sengaja dikosongkan)*

## LAMPIRAN A. HASIL KUESIONER



**KUESIONER TUGAS AKHIR – 5112100024 HAFIELUDIN YUSUF RIZANA**

**IMPLEMENTASI KECERDASAN BUATAN PADA GAME EEL JOE MENGGUNAKAN ALGORITMA A\***

**Identitas Responden**

Nama Lengkap : Ignatius Abraham S.      Usia : 22 Tahun  
 Pekerjaan : Mahasiswa      Jenis Kelamin : O/P

Isilah tabel dibawah ini dengan menggunakan tanda (v)

Dalam rentang 1 – 5

1 = sangat tidak setuju

5 = sangat setuju

No	Parameter Antarmuka	1	2	3	4	5
1	Aplikasi mudah digunakan				✓	
2	Antarmuka aplikasi memiliki desain yang bagus			✓		
3	Kecepatan pemilihan menu/fitur cukup				✓	
4	Tema game dengan tema yang diinginkan oleh pengembang cukup sesuai				✓	
5	Adanya ketertarikan untuk bermain game				✓	
<b>Parameter Performa, Kenyamanan, dan Fun</b>						
6	Animasi lancar ketika game dimainkan			✓		
7	Perpindahan menu lancar				✓	
8	Pemain menikmati game				✓	
<b>Parameter Kecerdasan Buatan</b>						
9	Bot memberikan tantangan lebih kepada pemain					✓
10	Rute yang dilewati bot ke pemain adalah rute terpendek					✓

**A. KRITIK DAN SARAN**

Font main menu agak kurang besar  
 Pause menu ~~itu~~ dibuat lebih menarik  
 .....  
 .....

Surabaya, 11 Juli 2016

Ignatius Abraham S.

**Gambar A.1 Kuesioner Pengguna 1**



KUESIONER TUGAS AKHIR – 5112100024 HAFIELUDIN YUSUF RIZANA

IMPLEMENTASI KECERDASAN BUATAN PADA GAME EEL JOE MENGGUNAKAN ALGORITMA A\*

Identitas Responden

Nama Lengkap : Denny Iman M Usia : 21 Tahun  
 Pekerjaan : Mahasiswa Jenis Kelamin : L/P

Isilah tabel dibawah ini dengan menggunakan tanda (v)

Dalam rentang 1 – 5

1 = sangat tidak setuju

5 = sangat setuju

No	Parameter Antarmuka	1	2	3	4	5
1	Aplikasi mudah digunakan					
2	Antarmuka aplikasi memiliki desain yang bagus				✓	
3	Kecepatan pemilihan menu/fitur cukup					✓
4	Tema game dengan tema yang diinginkan oleh pengembang cukup sesuai				✓	
5	Adanya ketertarikan untuk bermain game					✓
<b>Parameter Performa, Kenyamanan, dan Fun</b>						
6	Animasi lancar ketika game dimainkan					✓
7	Perpindahan menu lancar					✓
8	Pemain menikmati game					✓
<b>Parameter Kecerdasan Buatan</b>						
9	Bot memberikan tantangan lebih kepada pemain					✓
10	Rute yang dilewati bot ke pemain adalah rute terpendek					✓

A. KRITIK DAN SARAN

.....  
 Grafisnya di perbaiki lagi  
 .....  
 .....

Surabaya, 11 Juli ..... 2016

\_\_\_\_\_  
 Denny Iman M

Gambar A.2 Kuesioner Pengguna 2



ITS  
Institut Teknologi  
Sepuluh Nopember

KUESIONER TUGAS AKHIR – 5112100024 HAFIELUDIN YUSUF RIZANA

IMPLEMENTASI KECERDASAN BUATAN PADA GAME EEL JOE MENGGUNAKAN ALGORITMA A\*

Identitas Responden

Nama Lengkap

Zulwarof M

Usia

21 Tahun

Pekerjaan

IT Engineer

Jenis Kelamin

♂/P

Pertamina Shipping

Isilah tabel dibawah ini dengan menggunakan tanda (v)

Dalam rentang 1 – 5

1 = sangat tidak setuju

5 = sangat setuju

No	Parameter Antarmuka	1	2	3	4	5
1	Aplikasi mudah digunakan				✓	
2	Antarmuka aplikasi memiliki desain yang bagus				✓	
3	Kecepatan pemilihan menu/fitur cukup				✓	
4	Tema game dengan tema yang diinginkan oleh pengembang cukup sesuai					✓
5	Adanya ketertarikan untuk bermain game				✓	
<b>Parameter Performa, Kenyamanan, dan Fun</b>						
6	Animasi lancar ketika game dimainkan				✓	
7	Perpindahan menu lancar				✓	
8	Pemain menikmati game				✓	
<b>Parameter Kecerdasan Buatan</b>						
9	Bot memberikan tantangan lebih kepada pemain					✓
10	Rute yang dilewati bot ke pemain adalah rute terpendek					✓

A. KRITIK DAN SARAN

Kurang variasi, kurang power up, bot  
terlalu super power.

Surabaya, 7-11-2016

Gambar A.3 Kuesioner Pengguna 3



KUESIONER TUGAS AKHIR – 5112100024 HAFIELUDIN YUSUF RIZANA

IMPLEMENTASI KECERDASAN BUATAN PADA GAME EEL JOE MENGGUNAKAN ALGORITMA A\*

Identitas Responden

Nama Lengkap : Dimas Rizkadi ..... Usia : 23 Tahun  
 Pekerjaan : Mahasiswa ..... Jenis Kelamin : ♂ / P

Isilah tabel dibawah ini dengan menggunakan tanda (v)  
 Dalam rentang 1 – 5  
 1 = sangat tidak setuju  
 5 = sangat setuju

No	Parameter Antarmuka	1	2	3	4	5
1	Aplikasi mudah digunakan				✓	
2	Antarmuka aplikasi memiliki desain yang bagus					✓
3	Kecepatan pemilihan menu/fitur cukup				✓	
4	Tema <i>game</i> dengan tema yang diinginkan oleh pengembang cukup sesuai			✓		
5	Adanya ketertarikan untuk bermain <i>game</i>				✓	
<b>Parameter Performa, Kenyamanan, dan Fun</b>						
6	Animasi lancar ketika <i>game</i> dimainkan					✓
7	Perpindahan menu lancar					✓
8	Pemain menikmati <i>game</i>					✓
<b>Parameter Kecerdasan Buatan</b>						
9	Bot memberikan tantangan lebih kepada pemain				✓	
10	Rute yang dilewati bot ke pemain adalah rute terpendek				✓	

A. KRITIK DAN SARAN

- lebih diperbaiki tema *game* nya / tempatnya  
 - kalau bisa jangan kelat segi animasi dan enak lebih gres  
 .....  
 .....

Surabaya, ..... 11 Juli ..... 2016

Gambar A.4 Kuesioner Pengguna 4





KUESIONER TUGAS AKHIR – 5112100024 HAFIELUDIN YUSUF RIZANA

IMPLEMENTASI KECERDASAN BUATAN PADA GAME EEL JOE MENGGUNAKAN ALGORITMA A\*

Identitas Responden

Nama Lengkap : Aditya Putra Fecza  
 Pekerjaan : Mahasiswa  
 Usia : 21 Tahun  
 Jenis Kelamin : L/P

Isilah tabel dibawah ini dengan menggunakan tanda (v)

Dalam rentang 1 – 5

1 = sangat tidak setuju

5 = sangat setuju

No	Parameter Antarmuka	1	2	3	4	5
1	Aplikasi mudah digunakan				✓	
2	Antarmuka aplikasi memiliki desain yang bagus				✓	
3	Kecepatan pemilihan menu/fitur cukup				✓	
4	Tema game dengan tema yang diinginkan oleh pengembang cukup sesuai				✓	
5	Adanya ketertarikan untuk bermain game					✓
<b>Parameter Performa, Kenyamanan, dan Fun</b>						
6	Animasi lancar ketika game dimainkan				✓	
7	Perpindahan menu lancar			✓		
8	Pemain menikmati game					✓
<b>Parameter Kecerdasan Buatan</b>						
9	Bot memberikan tantangan lebih kepada pemain				✓	
10	Rute yang dilewati bot ke pemain adalah rute terpendek				✓	

A. KRITIK DAN SARAN

- Perpindahan menu pada story sedikit kurang jelas. Tombol panah ke atas malah dianggap keluar dari menu story
- Level 2 saat permainan dimulai agak bingung karena musuh langsung bergerak. Kalau bisa tiap saku musuh hanya bisa menempati satu ruangan
- Permainan ini sangat menarik untuk saya

Surabaya, 11 Juli 2016

*Aditya Putra Fecza*  
 Aditya Putra Fecza

Gambar A.5 Kuesioner Pengguna 5

*(Halaman ini sengaja dikosongkan)*



## **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. Aturan main dan skenario telah dirancang dengan baik sehingga membuat permainan lebih menantang bagi pemain.
2. Algoritma A\* telah berhasil diimplementasikan kepada bot dalam permainan.
3. Kecepatan gerak pemain mempengaruhi performa Algoritma A\*.
4. Aplikasi ini telah menyediakan kenyamanan yang cukup nyaman dalam bermain dan kesenangan untuk bermain bagi pemain.

#### **6.2. Saran**

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

1. Sebaiknya kecerdasan buatan dioptimasi kembali agar memiliki proses eksekusi kode yang lebih cepat.
2. Sebaiknya tampilan grafis dan animasi pada permainan dibuat lebih bagus agar menambah daya tarik dari permainan.

*(Halaman ini sengaja dikosongkan)*

## DAFTAR PUSTAKA

- [1] "Unity Manual," Unity, [Online]. Available: <http://docs.unity3d.com/Manual/index.html>. [Accessed 12 5 2015].
- [2] "/r/artificial," [Online]. Available: <https://www.reddit.com/r/artificial>. [Accessed 18 11 2015].
- [3] Millington, "Artificial Intelligence for Games," 2005.
- [4] W. Zeng and R. L. Church, "Finding shortest paths on real road networks: the case for A," 2009.
- [5] Ian Millington, John Funge, Artificial Intelligence for Games 2nd Edition, CRC Press , 2009.
- [6] S. J. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach," 2010.
- [7] "C#.NET Programming," [Online]. Available: <http://brainmatics.com/c-net-programming/>. [Accessed 21 12 2014].
- [8] Foraker Labs, "Usability Testing," Foraker Labs, Maret 2002. [Online]. Available: <http://www.usabilityfirst.com/usability-methods/usability-testing>. [Accessed 26 September 2014].

*(Halaman ini sengaja dikosongkan)*

## BIODATA PENULIS



Penulis dilahirkan di Surabaya, 12 September 1995, merupakan anak pertama dari 2 bersaudara. Penulis telah menempuh pendidikan formal yaitu TK Muhammadiyah Surabaya(1999-2001), SD Negeri Kebraon 1 Surabaya (2001-2007), SMP Negeri 16 Surabaya (2007-2010), SMA Negeri 1 Surabaya (2010-2012), dan mahasiswa S1

Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya rumpun mata kuliah Interaksi, Grafika, dan Seni (2011-2016). Penulis yang memiliki hobi bermain *game* merupakan mahasiswa yang aktif dalam organisasi intra kampus maupun ekstra kampus diantaranya Himpunan Mahasiswa Teknik Computer-Informatika ITS, dan Surabaya Youth. Penulis dapat dihubungi melalui *surel* [hafieluyr@gmail.com](mailto:hafieluyr@gmail.com).