

**PERANCANGAN DAN IMPLEMENTASI *FINITE STATE*
MACHINE PADA GAME “*CASTLE OF ILLUSION*”**

SKRIPSI



Disusun Oleh :

Achmad Badruddin

14.18.137

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

2019

LEMBAR PERSETUJUAN

**PERANCANGAN DAN IMPLEMENTASI *FINITE STATE*
MACHINE PADA GAME “*CASTLE OF ILLUSION*”
SKRIPSI**

*Disusun Dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
Sarjana Teknik Informatika Strata Satu (S-1)*

Disusun Oleh :

**Achmad Badruddin
14.18.137**

Diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

**Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002**

Nurlaily Vendyansyah, ST

Mengetahui,

Ketua Jurusan Teknik Informatika S-1

**Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002**

**JURUSAN TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2018**

LEMBAR KEASLIAN

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : Achmad Badruddin

NIM : 14.18.137

Program Studi : Teknik Informatika S-1

Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul :

***PERANCANGAN DAN IMPLEMENTASI FINITE STATE
MACHINE PADA GAME “CASTLE OF ILLUSION”***

Adalah skripsi sendiri bukan duplikasi serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 17 Januari 2019

Yang membuat pernyataan,

Achmad Badruddin

PERANCANGAN DAN IMPLEMENTASI *FINITE STATE MACHINE* PADA GAME “*CASTLE OF ILLUSION*”

Achmad Badruddin (1418137)
Program Studi Teknik Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jalan Raya Karanglo km 2 Tasikmadu, Malang
Email : badruddin283@gmail.com

ABSTRAK

Game adventure Castle Of Illusion merupakan permainan petualangan dimana pemain akan menjalankan karakter seorang kesatria. Dimana game ini menceritakan tentang sebuah kerajaan yang telah di rampok hasil alamnya oleh seorang penyihir jahat, dan seorang kesatria yg ingin berusaha merebut kembali dan menyelamatkan hasil bumi yg telah di curi oleh seorang penyihir jahat dan juga para pengikutnya.

Dalam *game* ini menggunakan *game engine Unity3D*, dengan menerapkan kecerdasan buatan FSM (*Finite State Machine*). FSM (*Finite State Machine*) digunakan pada karakter NPC (*Non Playable Character*) yaitu karakter yang digerakan oleh kecerdasan buatan yang digunakan untuk mendukung *game* tersebut. Seperti karakter *Enemy* dan *Boss* yang menggunakan tiga hal yaitu : Keadaan, Kejadian, dan Aksi.

Dari pengujian yang telah dilakukan hasil dari kecerdasan buatan FSM (*Finite State Machine*) pada *game Adventure “Castle Of Illusion”* ini karakter NPC (*Non Playable Character*) dapat mendeteksi keberadaan *player* untuk menentukan aksi serangan dari musuh tanpa melibatkan pengguna *game*, dari hasil pengujian yang dilakukan pada 10 orang yang berbeda maka mendapatkan hasil sebagai berikut, 2 orang menyatakan cukup, 6 orang menyatakan baik, 2 orang menyatakan baik sekali..

Kata kunci : *game adventure, Castle Of Illusion, finite state machine, Unity3D*

KATA PENGANTAR

Puji syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT atas berkat, rahmat, taufik, dan hidayah-Nya, penyusunan skripsi yang berjudul “Rancang Bangun dan Penerapan Metode *Finite State Machine* pada Game “*Castle Of Illusion*” dapat diselesaikan dengan baik. Shalawat serta salam senantiasa tercurah kepada junjungan Nabi besar Muhammad SAW beserta keluarga, sahabat, kerabat, dan pengikut beliau hingga akhir zaman.

Penulis menyadari bahwa dalam proses penulisan skripsi ini banyak mengalami kendala, namun berkat bantuan, bimbingan, kerjasama dari berbagai pihak dan berkah dari Allah SWT sehingga kendala – kendala yang dihadapi tersebut dapat diatasi. Untuk itu penulis menyampaikan ucapan terima kasih dan penghargaan kepada Bapak dan Ibu yang senantiasa mendoakan, memberikan bantuan moril, materi, dan nasehat selama penulis menjalani pendidikan.

Selanjutnya ucapan terima kasih penulis sampaikan pula kepada :

1. Bapak Dr. Ir. Lalu Mulyadi, MTA, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Dr. Ir. F. Yudi Limpraptono, MT, selaku Dekan Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, ST, MT, selaku Ketua Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.
4. Bapak Suryo Adi Wibowo, ST, MT, selaku Sekertaris Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.
5. Bapak Joseph Dedy Irawan, ST, MT selaku Dosen Pembimbing I, yang selalu memberikan bimbingan dan masukan.
6. Ibu Nurlaily Vendyansyah, ST selaku Dosen Pembimbing II, yang selalu memberikan bimbingan dan masukan.
7. Semua dosen Program Studi Teknik Informatika yang telah membantu dalam penulisan dan masukan.
8. Kedua Orang Tua yang senantiasa memberikan dukungan berupa doa, materi, serta semangat yang tak ternilai dalam menyelesaikan skripsi.
9. Semua Teman – teman yang sudah memberikan doa dan semangat dalam proses penyelesaian skripsi ini.

Dengan segala kerendahan hati, penulis mengharapkan adanya saran dan kritik yang bersifat membangun demi kesempurnaan skripsi ini.

Malang, 17 Januari 2019

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR KEASLIAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metode Penelitian	2
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Terkait	5
2.2 <i>Game</i>	6
2.2.1 Definisi <i>Game</i>	6
2.2.2 Sejarah <i>Game</i>	6
2.2.3 <i>Genre Game</i>	7
2.3 Kecerdasan Buatan (<i>Artificial Intelegence</i>).....	9
2.4 Jenis – Jenis Kecerdasan Buatan.....	9
2.4.1 <i>Decision Making</i>	9
2.4.2 <i>Pathfinding</i>	11

BAB III ANALISIS DAN PERANCANGAN SISTEM	14
3.1 Analisis <i>Game</i>	14
3.1.1 Analisis Target <i>User</i>	14
3.1.2 Analisis Konsep <i>Game</i>	15
3.1.3 Analisis Kebutuhan Perangkat	16
3.1.4 Analisis Kebutuhan Fungsional	17
3.1.5 Analisis Kebutuhan Non Fungsional	18
3.2 Perancangan <i>Game</i>	19
3.2.1 <i>Storyboard</i>	19
3.2.2 <i>Storyline</i>	20
3.2.3 Perancangan Level	21
3.2.4 Alur <i>Game</i>	22
3.2.5 Perancangan Struktur <i>Menu</i>	23
3.2.6 Perancangan Alur <i>Finite State Machine</i>	24
3.2.7 Rancangan Karakter	25
3.2.8 Perancangan <i>Environment</i>	27
3.2.9 Perancangan Kontrol <i>Game</i>	28
3.2.10 Perancangan Antar Muka	29
BAB IV IMPLEMENTASI DAN PENGUJIAN	32
4.1 Implementasi <i>Game</i>	32
4.1.1 Fungsi <i>Controller</i> Pada <i>Player</i>	33
4.1.2 Implementasi Kecerdasan Buatan Pada <i>Enemy</i> Level 1	33
4.1.3 Implementasi Kecerdasan Buatan Pada <i>Boss</i> Level 1	34
4.1.4 Implementasi Kecerdasan Buatan Pada <i>Enemy</i> Level 2	34
4.1.5 Implementasi Kecerdasan Buatan Pada <i>Boss</i> Level 2.....	35
4.1.6 Implementasi Kecerdasan Buatan Pada <i>Enemy</i> Level 3	35

4.1.7 Implementasi Kecerdasan Buatan Pada <i>Boss</i> Level 3.....	36
4.1.8 Implementasi <i>Build Unity</i> Menjadi Aplikasi	36
4.2 Pengujian.....	37
4.2.1 Pengujian <i>Gameplay</i>	37
4.2.2 Pengujian Fungsional	40
BAB V PENUTUP	48
5.1 Kesimpulan	48
5.2 Saran	48
DAFTAR PUSTAKA	49

DAFTAR GAMBAR

Gambar 2.1 Contoh Diagram <i>State</i> Sederhana (Sumber: Setiawan,2006).....	10
Gambar 2.2 Alur Proses Algoritma Dijkstra (Sholichin, 2013)	12
Gambar 3.1 <i>Storyboard</i> Level 1	19
Gambar 3.2 <i>Storyboard</i> Level 2	20
Gambar 3.3 <i>Storyboard</i> Level 3	20
Gambar 3.4 Rancangan <i>Flowchart Game</i>	23
Gambar 3.5 Rancangan Struktur <i>Menu</i>	23
Gambar 3.6 Diagram <i>Finite State Machine</i> Pada <i>Enemy Level</i>	24
Gambar 3.7 Diagram <i>Finite State Machine</i> Pada <i>Boss Level</i>	25
Gambar 3.8 Kontrol <i>Game</i>	28
Gambar 3.9 Rancangan Tampilan <i>Menu</i> Utama	29
Gambar 3.10 Rancangan Tampilan <i>Menu Help</i>	30
Gambar 3.11 Rancangan Tampilan <i>Menu About</i>	30
Gambar 3.12 Rancangan Tampilan <i>Stage Game</i>	31
Gambar 4.1 Tampilan Awal Unity	32
Gambar 4.2 Fungsi Controller Pada Player.....	33
Gambar 4.3 Fungsi Kecerdasan Buatan Pada Enemy Level 1	34
Gambar 4.4 Fungsi Kecerdasan Buatan Pada Boss Level 1	34
Gambar 4.5 Fungsi Kecerdasan Buatan Pada Enemy Level 2	35
Gambar 4.6 Fungsi Kecerdasan Buatan Pada Boss Level 2	35
Gambar 4.7 Fungsi Kecerdasan Buatan Pada Enemy Level 3	36
Gambar 4.8 Fungsi Kecerdasan Pada Boss Level 3	36

Gambar 4.9 Implementasi Build Unity Menjadi .Exe	37
Gambar 4.10 Pengaturan Platform	37
Gambar 4.19 Tampilan Level 1	38
Gambar 4.20 Tampilan Level 2	38
Gambar 4.21 Tampilan Level 3	39
Gambar 4.22 Tampilan Interaksi Enemy Terhadap Player Pada Level 1	39
Gambar 4.23 Tampilan Interaksi Enemy Terhadap Player Pada Level 2	40
Gambar 4.24 Tampilan Interaksi Enemy Terhadap Player Pada Level 3	40
Gambar 4.25 Enemy Patrol	41
Gambar 4.26 Enemy Mengejar Player	41
Gambar 4.27 Enemy Menembak Player	42
Gambar 4.28 Enemy Menyerang Player	42
Gambar 4.29 Boss Patrol.....	43
Gambar 4.30 Boss Mengejar Player	43
Gambar 4.31 Boss Menembak Player.....	44
Gambar 4.32 Boss Menyerang Player	44

DAFTAR TABEL

Tabel 3.1 Rancangan Karakter	25
Tabel 3.2 Rancangan <i>Environment</i>	27
Tabel 3.3 Fungsi Tombol Pada <i>Keyboard</i>	28
Tabel 4.1 Tabel Pengujian AI Pada <i>Enemy</i>	42
Tabel 4.2 Tabel Pengujian AI Pada <i>Boss Level</i>	44
Tabel 4.3 Tabel Pengujian <i>Controller Player</i>	45
Tabel 4.4 Tabel Pengujian Fungsional.....	46
Tabel 4.5 Tabel Pengujian <i>User</i>	47

BAB I

PENDAHULUAN

1.1 Latar Belakang

Game merupakan salah satu sarana *refreshing*, *game* merupakan salah satu sarana hiburan yang banyak diminati baik anak-anak hingga orang dewasa. Perkembangan teknologi informasi yang pesat semakin tidak dapat dihindari oleh masyarakat. Berbeda dengan zaman dahulu. Sekarang perkembangan dari *game* sangatlah pesat, dengan adanya dukungan dari berbagai platform dalam memberikan pengembang *game* suatu media untuk menjual *game*, sehingga saat ini *game-game* yang ada di pasar lokal maupun global sangat banyak, dengan kata lain persaingan dalam industri pengembang *game* masih terbuka luas.

Game adventure Castle Of Illusion merupakan permainan petualangan dimana pemain akan menjalankan karakter seorang kesatria. Dimana *game* ini menceritakan tentang sebuah kerajaan yang telah di rampok hasil alamnya oleh seorang penyihir jahat, dan seorang kesatria yg ingin berusaha merebut kembali dan menyelamatkan hasil bumi yg telah di curi oleh seorang penyihir jahat dan juga para pengikutnya.

Metode yang digunakan dalam *game "Castle Of Illusion"* menggunakan metode FSM (*Finite State Machine*), dimana metode tersebut berfungsi untuk pengambilan keputusan pada karakter NPC (*Non Playable Character*) yaitu karakter yang digerakkan oleh kecerdasan buatan yang digunakan untuk mendukung *game* tersebut seperti karakter musuh agar gerakan dan aksi dari musuh tanpa melibatkan pengguna *game* (Adi, dkk, 2009). Dari uraian diatas penulis ingin mengimplementasikan penggunaan metode FSM (*Finite State Machine*) untuk melakukan pembuatan *game* dengan judul "*Castle Of Illusion*" yang merupakan *game 2D* dengan *genre Adventure game*, dengan tipe *side scroller game* menggunakan *Unity* sebagai *game engine*.

1.2 Rumusan Masalah

Rumusan masalah pada pembuatan *game "Castle Of Illusion"* ini adalah sebagai berikut :

1. Bagaimana merancang *game* “*Castle Of Illusion*” dengan menggunakan *Unity* sebagai *game engine*?
2. Bagaimana mengimplementasikan metode *Finite State Machine* untuk *NPC* (*Non Playable Character*) biasa pada *game* “*Castle Of Illusion*”?

1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut :

1. Merancang *game* “*Castle Of Illusion*” berbasis desktop
2. Mengimplementasikan Metode *Finite State Machine* untuk *NPC* (*Non Playable Character*) biasa pada *game* “*Castle Of Illusion*”

1.4 Batasan Masalah

Agar penulisan dalam penelitian yang dibuat tidak keluar dari topik yang ditentukan, maka penelitian ini terdapat batasan permasalahan sebagai berikut:

1. *Game* “*Castle Of Illusion*” ini dibuat dengan menggunakan *game engine* *Unity*.
2. *Game* “*Castle Of Illusion*” menerapkan tipe *side scroller game*.
3. *Game* “*Castle Of Illusion*” ini bergenre *Adventure game*.
4. *Game* “*Castle Of Illusion*” ini memiliki 3 level yang berbeda dimana disetiap level terdapat bos level.
5. Karakter utama pada *Game* “*Castle Of Illusion*” ini adalah seorang cowboys yang misterius.
6. Kecerdasan dalam pembuatan *game* ini adalah *FSM* (*Finite State Machine*) untuk digunakan sebagai kondisi tindakan pada musuh.
7. *Game* ini dibuat untuk berjalan pada sistem operasi *Windows*.
8. Cerita dari *game* ini merupakan cerita fiksi yang dibuat oleh penulis.

1.5 Metode Penelitian

Untuk dapat mencapai keinginan dalam pembuatan *game* “*Castle Of Illusion*”, maka perlu dilakukan penelitian dengan langkah – langkah sebagai berikut :

1. Studi Literatur

Pada tahap ini dipelajari literatur dan perencanaan serta konsep awal untuk merancang *game* yang akan dibuat yaitu didapat dari referensi buku, internet, maupun sumber – sumber yang lain.

2. Pengumpulan Data

Pada tahap ini adalah proses pengumpulan data yang dibutuhkan untuk pembuatan *game*.

3. Analisis Dan Perancangan Sistem

Tahap selanjutnya adalah melakukan analisa dan perancangan sistem. Pada tahap ini adalah proses perancangan dari *game* yang akan dibuat untuk selanjutnya akan diproses lebih lanjut.

4. Pembuatan *Game*

Tahap selanjutnya adalah pembuatan *game*. Pada tahap ini rancangan yang sebelumnya telah dibuat akan diterapkan pada program yang akan dibuat. Pembuatan *game* ini menggunakan bahasa pemrograman *C#* dan *FSM (Finite State Machine)* sebagai metode penalaran pada program ini.

5. Uji Coba *Game*

Setelah *game* selesai dibuat maka dilakukan pengujian program untuk mengetahui apakah *game* tersebut telah bekerja dengan benar dan sesuai dengan sistem yang telah dibuat sebelumnya.

6. Pembuatan Kesimpulan

Pada tahap akhir ini adalah pembuatan kesimpulan atau ringkasan dari skripsi ini dan kesimpulan tentang program yang telah dibuat.

1.6 Manfaat

Adapun manfaat yang di dapatkan dari *game* yang bergenre petualangan ini yaitu :

1. Untuk sarana hiburan dan juga refreshing.
2. Membuat user mampu berfikir cepat dalam mengambil keputusan dan juga mampu memecahkan suatu teka-teki maupun menyimpulkan rangkaian peristiwa yang terjadi.

1.7 Sistematika Penulisan

Dalam penyusunan skripsi ini agar lebih mudah dipahami maka dibuatlah suatu sistematika dalam penulisan sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, metode penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada bab ini berisi tentang penelitian terkait dan dasar teori mengenai permasalahan yang berhubungan dengan penelitian ini.

BAB III : ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tentang analisis pada sistem dan perancangan sistem yang dibuat.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Pada bab ini membahas tentang paparan implementasi AI (*Artificial Intelligence*) yang diterapkan pada penelitian yaitu metode FSM (*Finite State Machine*) dan analisa hasil pengujian program yang dibuat, meliputi Pengujian AI (*Artificial Intelligence*), Pengujian *Controller Player*, Pengujian Fungsionalitas, dan Pengujian *User*.

BAB V : PENUTUP

Pada bab ini berisi kesimpulan dan saran yang didapat dari ulasan data – data penelitian, menyimpulkan bukti – bukti yang diperoleh dari hasil analisa.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Metode kecerdasan buatan pada *game* dengan menggunakan metode FSM (*Finite State Machine*) juga pernah digunakan oleh peneliti sebelumnya. Maka peneliti juga perlu untuk mempelajari penelitian-penelitian sebelumnya sebagai dasar acuan dalam mengerjakan penelitian ini.

NPC (*Non-Playable Character*) adalah obyek bergerak atau karakter pada dunia *game* yang dijalankan oleh komputer dan bisa berinteraksi dengan *pemain*. Penelitian ini mengenai penggunaan *Finite State Machine* untuk mendapatkan variasi respon NPC dengan *pemain* pada *game*. Dengan adanya variasi respon NPC pada *game* tersebut diharapkan *game* akan menjadi lebih menarik untuk *dimainkan* karena respon NPC lebih sulit untuk diprediksi (Adi, dkk, 2009).

FSM (*Finite State Machine*) merupakan salah satu metoda penerapan AI di *game* yang populer diterapkan untuk pengambilan keputusan yang cerdas. Penerapan FSM yang banyak ditemui adalah untuk membentuk NPC (*Non-Playable Character*) yang cerdas. Sehingga NPC tersebut dapat merespon karakter *pemain* sehingga 'seolah-olah' NPC tersebut bisa berpikir (Smed, Hakoneni, 2006).

Game Adventure "Castle Of Illusion" ini mengadopsi tipe *side scrolling game*, yaitu *game* dengan sudut pandang kamera dari sebelah samping. Dan *game* ini menggunakan 2 metode, Untuk karakter prajurit menggunakan metode *Fuzzy Logic*, yaitu kecerdasan yang berguna untuk menentukan berbagai macam respon NPC berdasarkan interaksi yang dilakukan oleh *pemain*, hal ini disebabkan karena FSM dapat digunakan untuk mendesain dan menentukan respon perilaku yang dilakukan terhadap perubahan kondisi. (Rahadian, Suyatno, Maharani, 2017).

Adapun metodologi yang digunakan adalah metode *Finite State Machine* (FSM), yaitu sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *State* (Keadaan), *Event* (Kejadian) dan *Action* (Aksi). Sebagai sebuah

metodologi perancangan sistem *control*. penerapan *Finite State Machine* (FSM) sangat cocok digunakan pada *game* (Marzian dan Qamal, 2017).

2.2 Game

Penjelasan tentang *game* meliputi Definisi *Game*, Sejarah *Game*, dan *Genre Game*.

2.2.1 Definisi Game

Permainan adalah kegiatan yang kompleks yang di dalamnya terdapat peraturan, *bermain* dan budaya. Sebuah permainan adalah sebuah sistem di mana *pemain* terlibat dalam konflik buatan. Di sini *pemain* berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau buatan. Dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku *pemain* dan menentukan permainan. (Somya, 2016).

2.2.2 Sejarah Game

Dalam buku yang berjudul “Optimalkan Potensi Anak Dengan *Game*” yang ditulis oleh Al. (Tridhonanto dan Beranda Agency, 2011) dijelaskan bahwa *game* sebenarnya sudah ditemukan sejak tiga dasawarsa lalu, kemunculannya pertama kali atas prakarsa Steven Russel dalam proyeknya yang bernama *Computer Games* pada tahun 1962 dengan produk andalannya bernama *Star Wars*.

Delapan tahun kemudian, sekitar tahun 1970-an muncul *game* yang cukup terkenal dikalangan *gamers* saat itu, namanya *game Pong* dengan sistem disket atau *catridge*. Setelah itu pada tahun 1980- an muncul *game* yang cukup populer dengan basis teknologi IBM PC yakni *game Pacman*. Hingga sekarang *game* ini juga masih populer dikalangan *gamers* dimana *Pacman* ini memakan semua *item*, dimana diantara *item* tersebut terdapat *item* penambah tenaga yang dapat digunakan untuk memakan hantu yang selalu menghalanginya.

Pada tahun 1993 tercipta yang juga tidak kalah populer dengan *Pacman* yaitu *game Mortal Combat*. *Game* ini mengusung *genre* aksi dimana dua karakter akan saling beradu jurus dan ilmu untuk saling menjatuhkan. Lima tahun kemudian muncul *game Doom* dengan desain yang lebih canggih karena

teknologinya telah menggunakan basis 3d tetapi sederhana dan sistem suara yang masih terbatas.

Dalam hal lain belum lagi ditambah tampilan gambar *game* dengan tekstur dan model yang bervariasi, lebih berwarna, nyata, dan menarik. Jika *game* yang dimiliki dilengkapi dengan fasilitas 3d maka hasilnya akan sangat mirip dengan kondisi nyata. Hal itu karena didukung dengan teknologi CG (*Computer Graphics*).

Di lihat dari segi komoditas *game* dari tahun 1990-an sampai tahun 2000-an sangatlah berbeda. Pada tahun 1990-an *game* dianggap sebagai komoditas bagi anak-anak sedangkan pada era tahun 2000-an *game* sudah melampaui berbagai bidang seperti hiburan bagi semua kalangan, bisnis, simulasi, edukasi, dan juga pembelajarn virtual. Perputaran uang dalam industri *game* juga sangat besar dan pertumbuhannya sangat pesat.

2.2.3 Genre Game

Game merupakan salah satu hiburan karena mampu mengurangi tingkat kepenatan seseorang dari rutinitas pekerjaan setiap hari. *Game* juga mampu meningkatkan kecerdasan seseorang ketika *game* tersebut memerlukan tingkat ketangkasan dari seorang *player*. Beberapa *game* yang berada saat ini terdapat unsur mendidik, ketangkasan, dan ada unsur kekerasan, maka ketika *game* itu diperjual belikan terdapat batasan umur pemakainya (Ardi, 2012).

Berdasarkan jenisnya, *game* dapat dibagi menjadi beberapa bagian yaitu diantaranya :

1. Shooting (Tembak – tembakan)

Game jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan, juga timing, inti dari *game* jenis ini adalah tembak, tembak dan tembak. Contoh : GTA (*Grand Theft Auto*), dan *Crysis*.

2. *Fighting* (Pertarungan)

Game yang permainannya memerlukan refleks dan koordinasi mata dan tangan dengan cepat, tetapi inti dari *game* ini adalah penguasaan hafalan jurus. Contoh : *Mortal Kombat* dan *Tekken*.

3. *Adventure* (Petualangan)

Game yang lebih menekankan pada jalan cerita dan kemampuan berfikir pemain dalam menganalisa tempat secara visual, memecahkan teka-teki maupun menyimpulkan berbagai peristiwa. Contoh : *Kings Quest*, dan *Space Quest*.

4. Simulasi

Game jenis ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detil berbagai faktor. Contoh : *The Sims*.

5. Strategi

Game jenis ini memerlukan koordinasi dan strategi dalam memainkan permainan ini. Kebanyakan *game strategi* adalah *game* perang. Contoh : *Warcraft*.

6. *Sport* (Olahraga)

Game ini merupakan adaptasi dari kenyataan, membutuhkan kelincahan dan juga strategi dalam memainkannya. Contoh : *Winning Eleven* dan NBA (*National Basketball Association*).

7. *Puzzle* (Teka-teki)

Game teka-teki, pemain diharuskan memecahkan teka-teki dalam *game* tersebut. Contoh : *Tetris*, *Minesweeper* dan *Bejeweled*.

8. *Edugames* (Edukasi)

Game jenis ini dibuat dengan tujuan spesifik sebagai alat pendidikan.

2.3 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan Buatan (*Artificial Intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Menurut John McCarthy, 1956, AI: untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan ditambah pengalaman, penalaran (bagaimana membuat keputusan dan mengambil tindakan), moral yang baik (Dahria, 2008).

Manusia cerdas (pandai) dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki tentu akan lebih mampu menyelesaikan permasalahan. Tapi bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang dimiliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik (Dahria, 2008).

2.4 Jenis – Jenis Kecerdasan Buatan

Adapun beberapa kecerdasan buatan antara lain *Decision Making* dan *Pathfinding*.

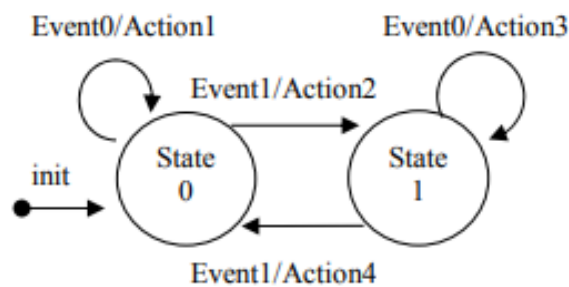
2.4.1 *Decision Making*

Jenis – jenis kecerdasan buatan *Decision Making* meliputi FSM (*Finite State Machine*) dan *Fuzzy Logic*.

1. **FSM (*Finite State Machine*)**

Finite State Machine adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut, *State* (Keadaan), *Event* (kejadian) dan *action* (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan

berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif. Contoh diagram *state* sederhana ditunjukkan pada Gambar 2.1.



Gambar 2.1 Contoh Diagram *State* Sederhana (Sumber: Setiawan,2006)

Diagram tersebut memperlihatkan FSM (*Finite State Machine*) dengan dua buah *state* dan dua buah *input* serta empat buah aksi output yang berbeda: seperti terlihat pada gambar, ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *state0*, pada keadaan ini sistem akan menghasilkan *Action1* jika terjadi masukan *Event0*, sedangkan jika terjadi *Event1* maka *Action2* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State1* dan seterusnya.

Salah satu alternatif implementasi FSM adalah menggunakan pemrograman berorientasi objek (*Object Oriented Programming*) atau yang sering disingkat sebagai OOP (Rahadian, Suyatno, & Maharani, 2017). Kelebihan penggunaan OOP pada FSM adalah fleksibilitasnya yang tinggi dan pemeliharaannya yang mudah baik pada sistem yang sederhana, menengah, maupun sistem yang kompleks. Selain itu juga mendapatkan manfaat dari salah satu kelebihan OOP yaitu penggunaan kembali kode yang telah diketik (*code reusability*) sehingga pengetikan kode menjadi lebih sedikit (Rahadian, Suyatno, & Maharani, 2017).

2. *Fuzzy Logic*

Fuzzy Logic diperkenalkan oleh Prof. Lotfi Zadeh pada tahun 1965. Merupakan metode yang mempunyai kemampuan untuk memproses variabel yang

bersifat kabur atau yang tidak dapat dideskripsikan secara eksak/pasti seperti misalnya tinggi, lambat, bising. Dalam *fuzzy logic* variabel yang bersifat kabur tersebut direpresentasikan sebagai sebuah himpunan yang anggotanya adalah suatu nilai crisp dan derajat keanggotaannya (*membershipfunction*) dalam himpunan tersebut (Hermawanto, 2008).

2.4.2 Pathfinding

Jenis – jenis kecerdasan buatan *Pathfinding* meliputi Algoritma A Star (A*), Algoritma Djikstra, Algoritma *Min Max*, *Breadth First Search* dan *Depth Search* *Fist*.

1. Algoritma A Star (A*)

Algoritma A *Star* (A*) adalah algoritma pencarian terbaik dalam mencari jalur terpendek dengan perhitungan terkecil pada jalur dengan simpul awal menuju simpul akhir. Algoritma ini pertama kali dideskripsikan pada tahun 1968 oleh Peter Hart, Nils Nilsson, dan Bertram Raphael. Dalam makalah mereka, disebut dengan Algoritma A. Lalu dengan optimasi heuristik, disebut dengan A *Star* (A*). Perhitungan pada Algoritma A *Star* (A*) dapat ditentukan sebagai berikut :

$$F(x) = G(x) + H(x)$$

Dimana :

- a. G(x) adalah nilai pada pergerakan simpul awal menuju simpul berikutnya.
- b. H(x) adalah perkiraan nilai pergerakan simpul awal menuju tujuan akhir simpul. Fungsi ini seringkali disebut dengan fungsi heuristik, dinamakan heuristik karena perhitungan tersebut berdasarkan perkiraan (*guess*).
- c. F(x) adalah jumlah nilai dari fungsi G(x) dan H(x). dengan nilai terkecil F(x) adalah jalur terpendek menuju tujuan akhir. (Maulana, Sofwan, Isnanto, 2011)

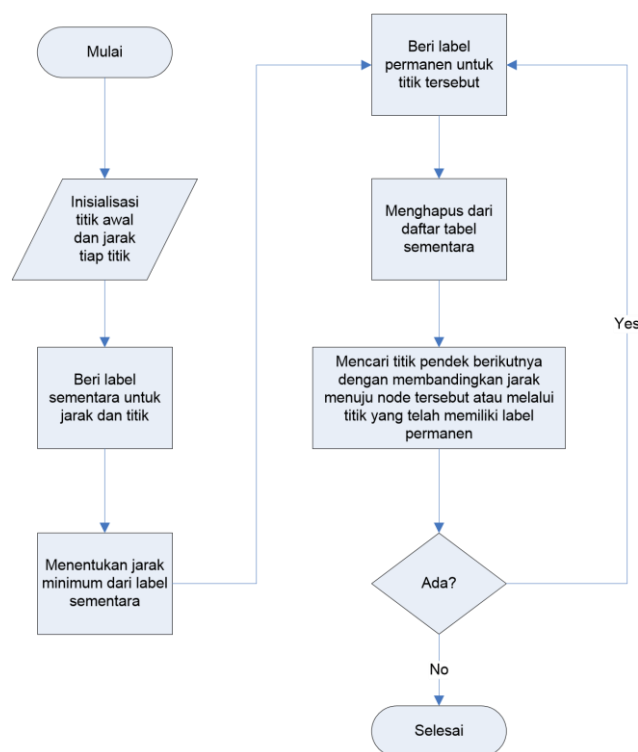
Terdapat ketentuan pada grafik agar algoritma A *Star* (A*) ini bila diterapkan akan selalu mendapatkan jalan yang terpendek. ketentuan tersebut yang harus dipenuhi pada grafik yaitu.

1. Setiap simpul (*node*) dalam grafik memiliki jumlah terbatas pada area pencariannya.
2. Pada pencarian terdapat jalan yang dilalui untuk mencapai tujuan.

3. Fungsi $F(x)$ pada grafik bernilai rendah daripada fungsi $F(x)$ pada pencarian sebelumnya.

2. Algoritma Dijkstra

Prinsip dari algoritma Dijkstra adalah dengan pencarian dua lintasan yang paling kecil. Algoritma Dijkstra memiliki iterasi untuk mencari titik yang jaraknya dari titik awal adalah paling pendek. Pada setiap iterasi, jarak titik yang diketahui (dari titik awal) diperbarui bila ternyata didapat titik yang baru yang memberikan jarak terpendek. Syarat algoritma ini adalah bobot sisinya yang harus non-negatif (Sholichin, 2012). Seperti terlihat pada Gambar 2.3



Gambar 2.2 Alur Proses Algoritma Dijkstra (Sholichin, 2013)

3. Algoritma *Min Max*

Algoritma *Min Max* adalah sebuah algoritma pencarian khusus yang berfungsi sebagai sebuah taktik untuk menentukan langkah terbaik berikutnya pada sebuah *game* yang melibatkan dua pemain, di mana hanya ada satu pengguna yang bermain melawan komputer. Algoritma *Min Max* bekerja secara rekursif, yaitu dengan mencari langkah yang akan meminimalkan peluang kemenangan lawan. Semua strategi lawan akan dihitung menggunakan algoritma.

Artinya, pada langkah pertama komputer akan menganalisis seluruh pohon permainan dan untuk setiap langkahnya komputer akan memilih langkah yang membuat lawan mendapatkan keuntungan yang paling minimum sedangkan komputer sendiri mendapatkan keuntungan maksimum. Pada pohon permainan, *Max* adalah pemain pertama yang bergerak dengan poin terbesar dan *Min* adalah pemain kedua yang bergerak dengan poin terkecil. Tentunya, *Max* harus membangun sebuah strategi, suatu strategi yang menentukan langkah *Max* pada simpul awal. Langkah *Max* dihasilkan dari semua tanggapan yang mungkin *Min* buat. Sebuah strategi optimal dalam kasus ini membawa kepada hasil terbaik, setidaknya sebaik strategi – strategi lainnya ketika bermain melawan musuh tangguh. (Chandra, 2014).

4. BFS (*Breadth First Search*)

Breadth First Search adalah penelusuran yang menggunakan pengetahuan akan suatu masalah untuk melakukan panduan pencarian ke arah *node* tempat dimana solusi berada. Pencarian seperti ini juga dikenal sebagai pencarian heuristik. Keuntungan jenis penelusuran ini adalah mengurangi beban komputasi karena hanya solusi yang memberikan harapan saja yang diuji dan akan berhenti apabila solusi sudah mendekati yang terbaik. (Diarsono, 2009).

5. DSF (*Depth Search First*)

Pada algoritma DFS, pencarian dilakukan pada satu *node* dalam setiap level dari yang paling kiri. Jika pada level yang paling dalam solusi belum ditemukan, maka pencarian dilanjutkan pada *node* sebelah kanan. *Node* yang di kiri dapat dihapus dari memori. Jika pada level yang paling dalam belum ditemukan solusi, maka pencarian dilanjutkan ke level sebelumnya. Demikian seterusnya sampai ditemukannya solusi.

Algoritma pencarian *Depth First Search* diawali dengan membangkitkan antrian untuk menjadi masukan. Kemudian antrian pertama diproses dengan cara mencocokkan nilai masukan terhadap nilai tujuan. Jika iya, maka masukan pertama merupakan solusi dan proses berakhir. Jika tidak, maka antrian masukan pertama dihapus dan dimasukan antrian kedua. Proses ini diulang sampai masukan sama dengan tujuan. (Fajar, 2015).

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Game

Dalam *game adventure*, setiap *pemain* harus menyelesaikan semua *stage* yang ada didalam *game*. Setiap *stage* memiliki tingkat kesulitan tertentu dan berbeda seperti menyelesaikan misi disetiap *stage* untuk dapat *menuju* ke *stage* selanjutnya. Disetiap *game adventure*, akan terasa kurang menarik apabila (*player*) hanya melawan musuh – musuh kecil yang mudah mati. Oleh karena itu dibuatlah karakter *boss* yang tidak mudah dikalahkan *player* dan memiliki status diatas rata – rata bahkan melebihi karakter *player*. *Boss* berada disetiap *stage* akhir *game* dan untuk *pemain* secara tidak langsung dituntut untuk mengasah kemampuan analisa menyerang dan bertahan serta menghindar.

Dalam *game “Castle Of Illusion”* menggunakan *view side scroller*, yaitu dimana tampilan *game* dari samping dengan desain 2D. Pada *game “Castle Of Illusion”*, *pemain* dituntut untuk mengumpulkan *coins* dan *item* yang ada pada setiap *stage*. Berikut ini adalah analisis *game* yang ada pada *game “Castle Of Illusion”* :

1. Menggunakan Grafik 2D.
2. *Game* ini bergenre *Adventure*.
3. Interaksi menggunakan *Keyboard*.

3.1.1 Analisis Target User

Game “Castle Of Illusion” memiliki target *user* yaitu semua kalangan dengan usia diatas 13 tahun keatas dan dapat mengoperasikan komputer, apabila *user* berusia dibawah 13 tahun maka harus dengan bimbingan dari orang tua karena adanya adegan kekerasan dalam *game* ini yang kurang sesuai bagi kalangan dibawah usia 13 tahun kebawah. Dikhawatirkan apabila tanpa bimbingan dari orang tua, *pemain* yang masih berusia 13 dibawah tahun akan memperagakan dan menirukan adegan kekerasan pada *game* ini di dunia nyata.

3.1.2 Analisis Konsep *Game*

Analisis konsep *game* meliputi *Genre Game*, *Tujuan Game*, *Start Game*, *Middle Game*, dan *Ending Game*.

1. *Genre Game*

Genre pada *game* “*Castle Of Illusion*” menggunakan *view side scroller*, yaitu *game* terlihat dari samping dengan desain 2D.

2. *Tujuan Game*

Didalam sebuah *game* pasti memiliki *goal* atau tujuan yang harus dicapai agar dapat menyelesaikan setiap *stage* pada *game*. Dalam *game* “*Castle Of Illusion*” *goal* atau tujuan yang harus dicapai adalah *pemain (player)* harus melewati setiap rintangan, mengumpulkan semua *coins* dan sebuah *item* yang terdapat dalam setiap levelnya.

3. *Start Game*

Start game adalah kondisi awal *pemain (player)* saat *game* baru dimulai. Kondisi *start game* pada *game* “*Castle Of Illusion*” adalah dimulai dengan cerita awal *game* dan *pemain (player)* dibekali dengan status *full health point* serta *coins* dan *item* awal adalah 0.

4. *Middle Game*

Middle game pada *game* “*Castle Of Illusion*” ini *pemain* harus menjelajahi *stage* dan mengalahkan semua musuh. *Pemain (player)* harus melewati 3 *stage* yang dimana terdapat *boss level* disetiap *stagenya* untuk menuju ke *ending game*, ketiga *stage* tersebut yaitu sebagai berikut.

a. *Stage 1*

Goal pada *stage 1 game* “*Castle Of Illusion*” ini adalah musuh memiliki tingkat kesulitan yang mudah atau *easy*. Dimana musuh memiliki *health point* yang sedikit, karakter utama harus mengalahkan semua musuh dan juga harus mengumpulkan *coins* sebanyak 30 *coins* dan sebuah *item* yang didapat dari mengalahkan *boss level* pada *stage 1* untuk dapat melanjutkan ke *stage* yang berikutnya.

b. *Stage 2*

Goal pada *stage 2 game "Castle Of Illusion"* ini adalah musuh memiliki tingkat kesulitan yang sedang atau *medium*. Dimana musuh memiliki *health point* yang lebih banyak dari pada *stage 1*, karakter utama harus mengalahkan semua musuh dan juga harus mengumpulkan *coins* sebanyak 35 *coins* dan sebuah *item* yang didapat dari mengalahkan *boss* level pada *stage 2* untuk dapat melanjutkan ke *stage* yang berikutnya.

c. *Stage 3*

Goal pada *stage 3 game "Castle Of Illusion"* ini adalah musuh memiliki tingkat kesulitan yang sulit atau *hard*. Dimana musuh memiliki *health point* yang lebih banyak dari pada *stage 1* dan *stage 2*, karakter utama harus mengalahkan semua musuh dan juga harus mengumpulkan *coins* sebanyak 40 *coins* dan sebuah *item* yang didapat dari mengalahkan *boss* level pada *stage 3* untuk dapat menyelesaikan *game*.

5. *Ending Game*

Ending dari *game "Castle Of Illusion"* ini adalah berada pada level 3. Pada level 3 terdapat *Alien Boss* dan *pemain* harus mengalahkannya, setelah mengalahkan *boss* maka *game* akan lanjut ke *story ending* lalu menuju *scene* tamat.

3.1.3 Analisis Kebutuhan Perangkat

Pada *game "Castle Of Illusion"* diperlukan perangkat keras maupun perangkat lunak untuk mengembangkan maupun memainkannya yang meliputi :

1. Analisis Kebutuhan Perangkat Keras

Spesifikasi *hardware* atau perangkat keras yang dibutuhkan oleh pengguna atau *pemain (player)* adalah sebagai berikut :

- a. Laptop / Notebook / PC (Personal Computer)
- b. Ram 2 GB
- c. Monitor
- d. *Mouse* dan *Keyboard*
- e. *Speaker*

2. Analisis Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak bagi pengembang yang digunakan dalam membangun game “*Castle Of Illusion*” adalah :

- a. Sistem Operasi *Windows 7, Windows 8*
- b. *Software Unity 5.6.1f1*
- c. *Software Corel Draw x4*
- d. *Software Adobe Photoshop*

Sedangkan spesifikasi perangkat lunak bagi pengguna atau pemain (*player*) yang memainkan game “*Castle Of Illusion*” adalah sistem operasi *Windows 7* atau *Windows 8*.

3.1.4 Analisis Kebutuhan Fungsional

Kebutuhan Fungsional adalah kebutuhan yang berisi proses – proses apa saja atau fitur apa saja yang nantinya harus disediakan oleh sistem, mencakup bagaimana sistem harus bereaksi pada *input* tertentu dan bagaimana perilaku sistem pada situasi tertentu. Adapun Analisis Kebutuhan Fungsional pada game adalah sebagai berikut :

1. Dalam game *Castle Of Illusion*, *Player* dapat terkena serangan *Enemy*
2. Dalam game *Castle Of Illusion*, *Enemy* dapat terkena serangan *Player*
3. Dalam game *Castle Of Illusion*, tombol *Play Game* untuk memulai permainan
4. Dalam game *Castle Of Illusion*, tombol *Help* merupakan petunjuk *control player*
5. Dalam game *Castle Of Illusion*, tombol *About* dapat menampilkan profil pembuat game
6. Dalam game *Castle Of Illusion*, tombol *Exit* dapat keluar dari permainan
7. Dalam game *Castle Of Illusion*, *Player* dapat menggunakan serangan jarak jauh (*shoot*)
8. Dalam game *Castle Of Illusion*, *Player* memiliki *control jump* (melompat) dengan menekan tombol spasi (*space*) berlari (*run*) dengan menekan tombol

arah kiri dan kanan, memukul (*attack*) dengan menekan tombol *shift*, menghindar (*slide*) dengan menekan tombol *Ctrl*, dan menembak (*shoot*) dengan menekan tombol *V* pada *keyboard*, terkena serangan musuh (*damage*), dan mati (*dead*)

9. Dalam *game Castle Of Illusion*, pintu untuk menuju selanjutnya akan terbuka ketika *player* berhasil menyelesaikan misi pada setiap level
10. Dalam *game Castle Of Illusion*, permainan berakhir apabila *health point player* habis atau bernilai 0
11. Dalam *game Castle Of Illusion*, *coin* akan bertambah ketika *player* mengambil item *coin*
12. Dalam *game Castle Of Illusion*, *health point* akan berkurang apabila *player* terkena serangan *enemy (damage)*
13. Dalam *game Castle Of Illusion*, *health point enemy* akan berkurang apabila *enemy* terkena serangan *player (damage)*

3.1.5 Analisis Kebutuhan Non Fungsional

Kebutuhan Non Fungsional adalah kebutuhan yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Kebutuhan non fungsional juga sering disebut sebagai batasan fitur atau fungsi yang ditawarkan sistem seperti batasan waktu, batasan pengembangan proses, standarisasi dan lain – lain. Adapun kebutuhan non fungsional pada *game* adalah sebagai berikut :

1. Sistem *single player*
2. Grafik *game* 2 dimensi
3. Interaksi menggunakan *keyboard*
4. Tindakan pada karakter musuh diimplementasikan menggunakan metode *finite state machine*
5. *Finite state machine* diimplementasikan pada karakter *enemy* dan *boss* level
6. *Game* ini dibuat dengan menggunakan *software Unity3D* dengan bahasa pemrograman *C#*

3.2 Perancangan *Game*

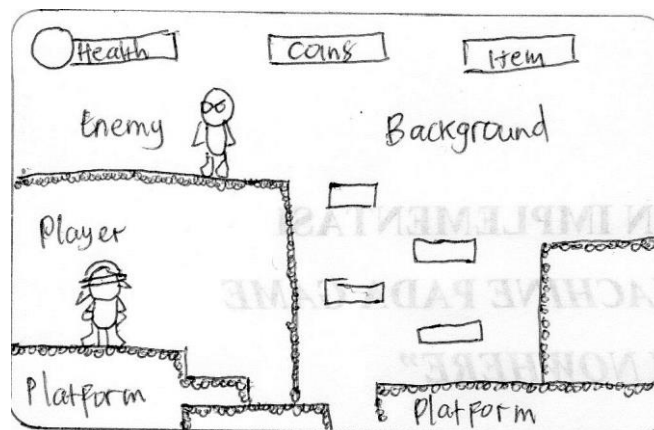
Perancangan *game* adalah suatu bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan untuk memberikan gambaran secara terperinci tentang *game* “*Castle Of Illusion*” adalah meliputi *Storyboard*, *Storyline*, Perancangan Level, Kontrol *Game*, Alur *Game*, Perancangan Struktur *Menu*, Perancangan Diagram FSM (*Finite State Machine*), Perancangan Karakter, Perancangan *Environment*, Perancangan Antar Muka.

3.2.1 *Storyboard*

Storyboard adalah gambaran ide dari aplikasi yang akan dibangun, sehingga dapat memberikan gambaran dari aplikasi yang akan dihasilkan. *Storyboard* dapat dikatakan juga *visual script* yang akan dijadikan *outline* dari sebuah proyek, ditampilkan *shot by shot* yang biasa disebut dengan istilah *scene*.

1. *Storyboard* Level 1 *Game*

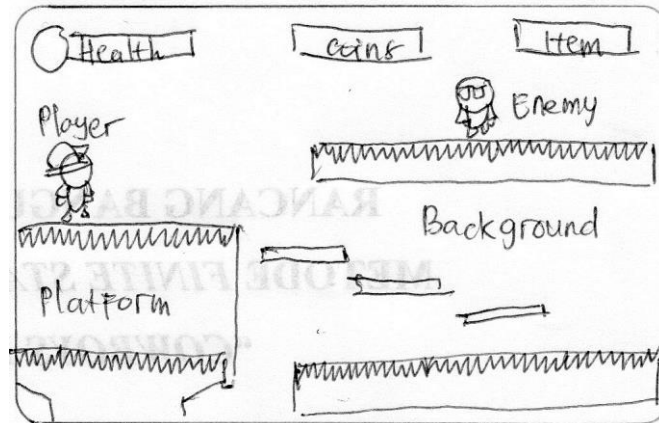
Arena Level 1 merupakan perancangan arena pertama yang harus diselesaikan oleh *player*. Rancangan *storyboard* level 1 dapat dilihat seperti gambar yang ditunjukkan Gambar 3.23



Gambar 3.1 *Storyboard* Level 1

2. *Storyboard* Level 2

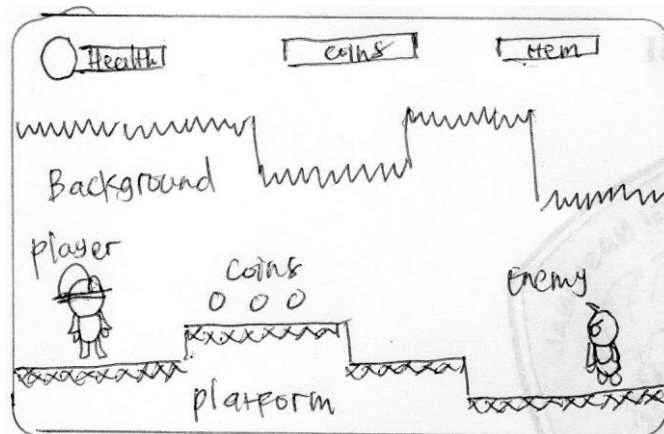
Arena Level 2 merupakan perancangan arena pertama yang harus diselesaikan oleh *player*. Rancangan *storyboard* level 2 dapat dilihat seperti gambar yang ditunjukkan Gambar 3.24



Gambar 3.2 *Storyboard* Level 2

3. *Storyboard* Level 3 Game

Arena Level 3 merupakan perancangan arena pertama yang harus diselesaikan oleh *player*. Rancangan *storyboard* level 3 dapat dilihat seperti gambar yang ditunjukkan Gambar 3.25



Gambar 3.3 *Storyboard* Level 3

3.2.2 *Storyline*

Berikut ini adalah alur cerita dalam game “*Castle Of Illusion*” :

1. Pada suatu zaman di sebuah kerajaan hidup seorang *kesatria* yang sangat pemberani yang bernama samba, ia merupakan sosok yang baik hati, kuat dan pandai.

2. Pada suatu ketika datanglah seorang penyihir jahat dan para anak buahnya yang ingin menguasai kerajaan tersebut
3. Penyihir tersebut berhasil merampas hampir seluruh harta yang ada di dalam kerajaan tersebut dan di bawanya ke dalam hutan
4. Kemudian sang kesatria tersebut bertekad mengembara ke dalam hutan untuk mengambil kembali harta kerajaan yang telah dicuri oleh penyihir jahat dan para anak buahnya tersebut
5. Sesampainya ia di dalam hutan, ternyata penyihir tersebut telah mempersiapkan penjagaan yg sangat kuat
6. Dan tidak ada pilihan lain kesatria tersebut harus melawan semua penjaga harta tersebut yang telah di siapkan oleh si penyihir jahat tersebut untuk mengambil kembali harta kerajaan yang telah di curi oleh si penyihir jahat.

3.2.3 Perancangan Level

Game ini terbagi menjadi tiga level utama. Masing – masing level memiliki latar, musuh, *boss* level, dan tingkat kesulitan yang berbeda – beda. Beberapa aturan dari *game* ini yaitu :

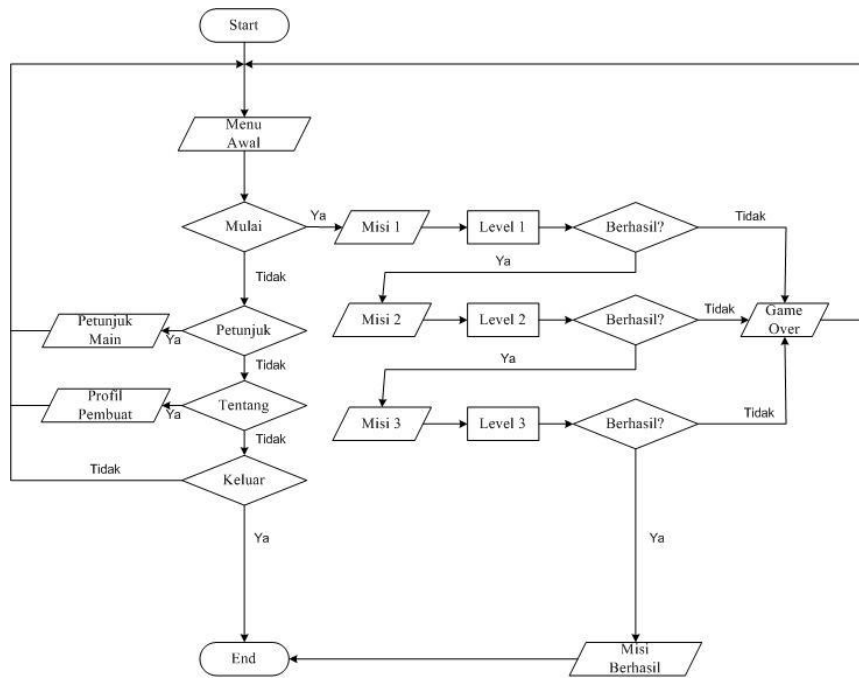
1. Perancangan Level 1
 - a. *Player* memiliki *health point* 50 point
 - b. *Player* akan mati apabila *health point* 0
 - c. Saat *player* mati maka *game over* atau permainan selesai
 - d. *Player* harus mengulang permainan dari awal
 - e. *Player* harus mengumpulkan 30 *coins* untuk menyelesaikan level
 - f. *Player* harus mengambil *item* untuk dapat melanjutkan ke level berikutnya
 - g. *Enemy* pada level 1 berjumlah 5
 - h. Terdapat *boss* level pada setiap akhir level

2. Perancangan Level 2
 - a. *Player* memiliki *health point* 60 point
 - b. *Player* akan mati apabila *health point* 0
 - c. Saat *player* mati maka *game over* atau permainan selesai
 - d. *Player* harus mengulang permainan dari awal

- e. *Player* harus mengumpulkan 35 *coins* untuk menyelesaikan level
 - f. *Player* harus mengambil *item* untuk dapat melanjutkan ke level berikutnya
 - g. *Enemy* pada level 2 berjumlah 5
 - h. Terdapat *boss* level pada setiap akhir level
3. Perancangan Level 3
- a. *Player* memiliki *health point* 70 point
 - b. *Player* akan mati apabila *health point* 0
 - c. Saat *player* mati maka *game over* atau permainan selesai
 - d. *Player* harus mengulang permainan dari awal
 - e. *Player* harus mengumpulkan 40 *coins* untuk menyelesaikan level
 - f. *Player* harus mengambil *item* untuk dapat melanjutkan ke level berikutnya
 - g. *Enemy* pada level 3 berjumlah 5
 - h. Terdapat *boss* level pada setiap akhir level

3.2.4 Alur Game

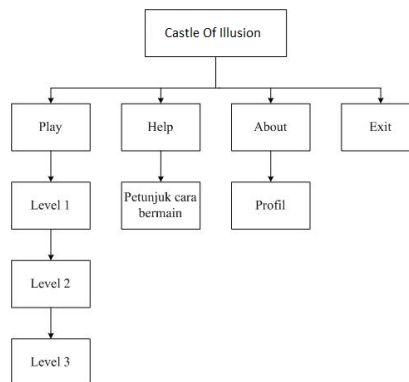
Perancangan alur *game* berfungsi untuk mengetahui alur proses awal program dijalankan sampai akhir dari program. Pada tampilan awal *game* “*Castle Of Illusion*” menampilkan *menu* utama yang ada dalam *game* yang terdiri dari *Play Game*, *Help*, *About*, dan *Exit Game*. Jika memilih *menu Play Game*, maka akan menampilkan prolog cerita *game*, kemudian misi yang harus diselesaikan pada Level 1. Jika Level 1 berhasil menyelesaikan misi maka akan dilanjutkan ke misi selanjutnya tetapi jika misi pada Level 1 tidak terpenuhi atau *game over* maka akan kembali ke *menu* awal. Kemudian terdapat *menu Help* yang berisi tentang informasi tombol apa saja yang digunakan pada *keyboard* untuk bermain *game* “*Castle Of Illusion*”. Pada *menu About* berisi tentang *Profile* pembuat *game* “*Castle Of Illusion*”. Pada *menu Exit* maka *game* akan berhenti dan keluar dari *game*. Alur dari *game* “*Castle Of Illusion*” ditunjukkan dengan *flowchart* pada Gambar 3.5



Gambar 3.4 Rancangan *Flowchart Game*

3.2.5 Perancangan Struktur Menu

Pada game “*Castle Of Illusion*” ini terdiri dari beberapa menu yaitu *Play*, *Help*, *About*, dan *Exit*. Diagram struktur menu dapat dilihat pada gambar 3.6



Gambar 3.5 Rancangan Struktur Menu

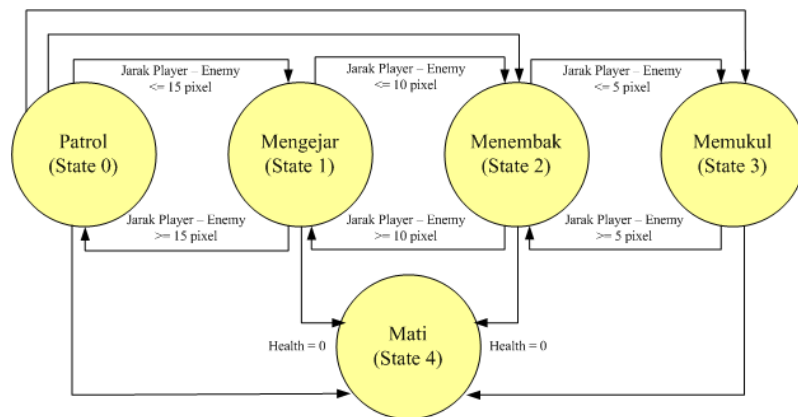
Pada game “*Castle Of Illusion*” terdapat 4 menu utama yaitu *Play*, *Help*, *About*, dan *Exit*. Untuk memulai game, pemain harus memilih menu *Play*. Pada menu *Help* akan ditampilkan tombol – tombol apa saja yang dapat digunakan dalam game ini. Pada menu *About* akan ditampilkan *profile* pembuat game ini dan menu *Exit* untuk keluar dari game.

3.2.6 Perancangan Alur *Finite State Machine*

Dalam game “*Castle Of Illusion*” ini memiliki 2 jenis FSM (*Finite State Machine*), yaitu metode FSM pada *Enemy Level* dan metode FSM pada *Boss Level*.

1. Penerapan FSM (*Finite State Machine*) pada *Enemy Level*

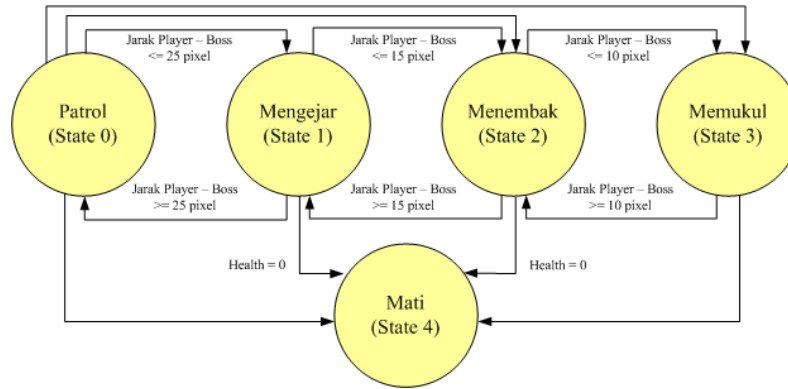
Penerapan alur FSM pada karakter *Enemy Level* memiliki sifat jika karakter *Player* mendekat pada jarak $< 15 \text{ pixel}$ dari *Enemy*, maka *Enemy* akan mengejar, jika *Player* mendekat lagi pada jarak $< 10 \text{ pixel}$ maka *Enemy* akan menembak *Player*, dan jika *Player* mendekat pada jarak $< 5 \text{ pixel}$ maka *Enemy* akan menyerang *Player*, apabila *player* menjauh dari *Enemy* dengan jarak $> 15 \text{ pixel}$ maka *Enemy* akan kembali *Patrol* pada tempat semula *Enemy* berada, seperti yang terlihat pada Gambar 3.6



Gambar 3.6 Diagram *Finite State Machine* Pada *Enemy Level*

2. Penerapan FSM (*Finite State Machine*) pada *Boss Level*

Penerapan alur FSM pada karakter *Boss Level* memiliki sifat jika karakter *Player* mendekat pada jarak $< 25 \text{ pixel}$ dari *Boss*, maka *Boss* akan mengejar, jika *Player* mendekat lagi pada jarak $< 15 \text{ pixel}$ maka *Boss* akan menembak *Player*, dan jika *Player* mendekat pada jarak $< 10 \text{ pixel}$ maka *Boss* akan menyerang *Player*, apabila *player* menjauh dari *Boss* dengan jarak $> 25 \text{ pixel}$ maka *Boss* akan kembali *Patrol* pada tempat semula *Boss* berada, seperti yang terlihat pada Gambar 3.7










Gambar 3.7 Diagram *Finite State Machine* Pada *Boss Level*

3.2.7 Rancangan Karakter

Desain karakter merupakan pembahasan mengenai karakter – karakter yang terlibat dalam *game “Castle Of Illusion”*. Karakter pada *game “Castle Of Illusion”* dapat dilihat pada Tabel 3.1

Tabel 3.1 Rancangan Karakter






No	Karakter	Keterangan
1		Karakter Utama (<i>Kesatria</i>) <i>Sumber : www.gameart2d.com</i>
2		<i>Enemy Level</i> yang ada di Level 1 (<i>Goblin</i>) <i>Sumber : www.gameart2d.com</i>

No	Karakter	Keterangan
3		<p><i>Enemy Level yang ada di Level 2</i> <i>(Ogre)</i> <i>Sumber : www.gameart2d.com</i></p>
4		<p><i>Enemy Level yang ada di Level 3</i> <i>(Orc)</i> <i>Sumber www.gameart2d.com</i></p>
5		<p><i>Boss Level yang ada di Level 1</i> <i>(Girl Zombi)</i> <i>Sumber : www.gameart2d.com</i></p>
6		<p><i>Boss Level yang ada di Level 2</i> <i>(Men Zombi)</i> <i>Sumber : www.gameart2d.com</i></p>
7		<p><i>Boss Level yang ada di Level 3</i> <i>(Manusia Labu)</i> <i>Sumber : www.gameart2d.com</i></p>

3.2.8 Perancangan *Environment*

Rancangan *Environment* adalah pembahasan mengenai lingkungan yang terdapat pada game “*Castle Of Illusion*”. Berikut ini adalah *Environment* yang dapat dilihat pada Tabel 3.2

Tabel 3.2 Rancangan *Environment*

No	Nama	Gambar	Keterangan
1	<i>Ground</i> Level 1		<i>Ground</i> yang digunakan pada Level 1
2	<i>Ground</i> Level 2		<i>Ground</i> yang digunakan pada Level 2
3	<i>Ground</i> Level 3		<i>Ground</i> yang digunakan pada Level 3
4	<i>Coin</i>		<i>Item</i> yang harus dikumpulkan oleh <i>Player</i> untuk menyelesaikan misi
5	<i>Door</i>		Pintu yang digunakan untuk menuju Level selanjutnya

3.2.9 Perancangan Kontrol *Game*

Game ini dapat dimainkan dengan menggunakan *Gamepad* dengan *control mapping* sebagai berikut seperti Gambar 3.2.1



Gambar 3.2.1 Kontrol *Game*

Penjelasan dari gambar diatas akan dijelaskan pada Tabel 3.3 dibawah ini :

Tabel 3.3 Fungsi Tombol Pada *Keyboard*

No	<i>Keyboard</i>	Fungsi	<i>Output</i>
1	<i>Shift</i>	Menyerang (<i>Attack</i>)	<i>Player Menyerang</i>
2	<i>Ctrl</i>	Menghindar (<i>Sliding</i>)	<i>Player Menghindar</i>
3	V	Menembak (<i>Shoot</i>)	<i>Player Menembak</i>
4	<i>Space</i>	Melompat (<i>Jump</i>)	<i>Player Melompat</i>
5	⇒	Bergerak ke Kanan (<i>Moving Right</i>)	<i>Player Bergerak ke Kanan</i>
6	⇐	Bergerak ke Kiri (<i>Moving Left</i>)	<i>Player Bergerak ke Kiri</i>

Dari tabel 3.3 ketika *pemain* menekan tombol arah kanan, arah kiri, dan *space* pada *keyboard* karakter dapat bergerak sesuai dengan arahnya yaitu, arah kanan untuk bergerak maju, arah kiri untuk berkerak mundur, dan *space* untuk melompat ke atas. Ketika *pemain* menekan tombol V maka karakter akan menembak. Ketika *pemain* menekan tombol *Shift* maka karakter akan menyerang, dan ketika *pemain* menekan tombol *Ctrl* maka karakter akan menghindar dengan cara *sliding* dari serangan musuh.

3.2.10 Perancangan Antar Muka

Perancangan antar muka bertujuan untuk memberikan gambaran bagaimana tampilan keseluruhan dari game “*Castle Of Illusion*”.

1. Perancangan Tampilan *Main Menu*

Perancangan tampilan *menu* utama pada game “*Castle Of Illusion*” terdapat 4 *menu* utama, yaitu tombol *Play Game*, maka *pemain* akan diarahkan *menuju* tampilan prolog, dan kemudian tampilan misi, dan *menuju* ke permainan Level 1. Jika *pemain* memilih tombol *Help*, maka *pemain* akan diarahkan *menuju* tampilan yang menunjukkan bagaimana cara memainkan game “*Castle Of Illusion*”. Jika *pemain* memilih *About* maka *pemain* akan diarahkan *menuju* tampilan *Profile* pembuat game. Dan jika *pemain* memilih *Exit*, maka *pemain* akan keluar dari game. Rancangan *menu* utama dari game “*Castle Of Illusion*” ditunjukkan pada Gambar 3.9.



Gambar 3.8 Rancangan Tampilan *Menu* Utama

2. Perancangan Tampilan *Menu Help*

Perancangan antarmuka *menu Help* merupakan perancangan tampilan *menu Help* yang terdapat di *menu* utama. Di dalam *menu Help* ini berisi tentang informasi tombol *keyboard* yang digunakan dan cara bermain game ini. Rancangan antar muka *menu Help* terlihat seperti pada gambar 3.10.



Gambar 3.9 Rancangan Tampilan *Menu Help*

3. Perancangan Tampilan *Menu About*

Perancangan antarmuka *menu About* merupakan perancangan tampilan *menu About* yang terdapat di *menu* utama. Di dalam *menu About* ini berisi tentang informasi profile pembuat *game "Castle Of Illusion"*. Rancangan antar muka *menu About* terlihat seperti pada gambar 3.11.

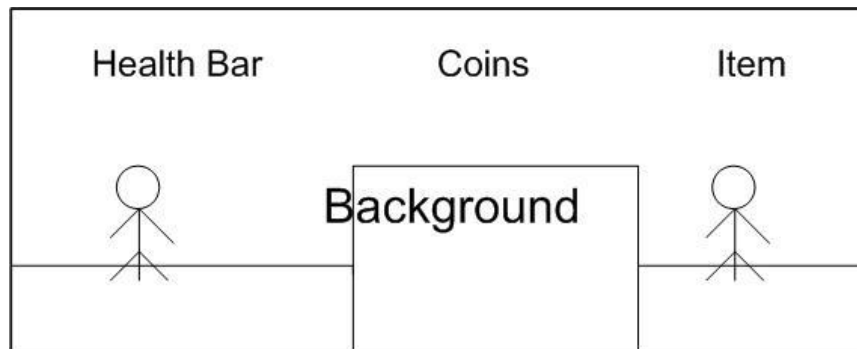


Gambar 3.10 Rancangan Tampilan *Menu About*

4. Perancangan Tampilan *Level Game*

Perancangan antarmuka *level game* merupakan perancangan tampilan level dari *game "Castle Of Illusion"*. Pada setiap level terdapat *user interface* dari karakter pemain yaitu *health point bar*, *Coins*, dan *Item*. *Health point bar* bertujuan mengetahui darah dari pemain, jika darah pemain mencapai 0 maka pemain akan diarahkan menuju tampilan *Game over*. *Item* berfungsi untuk menampilkan banyaknya *Item* yang sudah didapat pemain, jika jumlah *Item* yang dikumpulkan belum memenuhi target misi, maka karakter tidak bisa membuka portal untuk ke level selanjutnya. *Coins* berfungsi untuk menampilkan banyaknya *Coins* yang didapat pemain, jika pemain membunuh musuh dalam *game* maka

pemain akan memperoleh *coins*. Rancangan antarmuka *level game* dari “*Castle Of Illusion*” ditunjukkan pada Gambar 3.12.



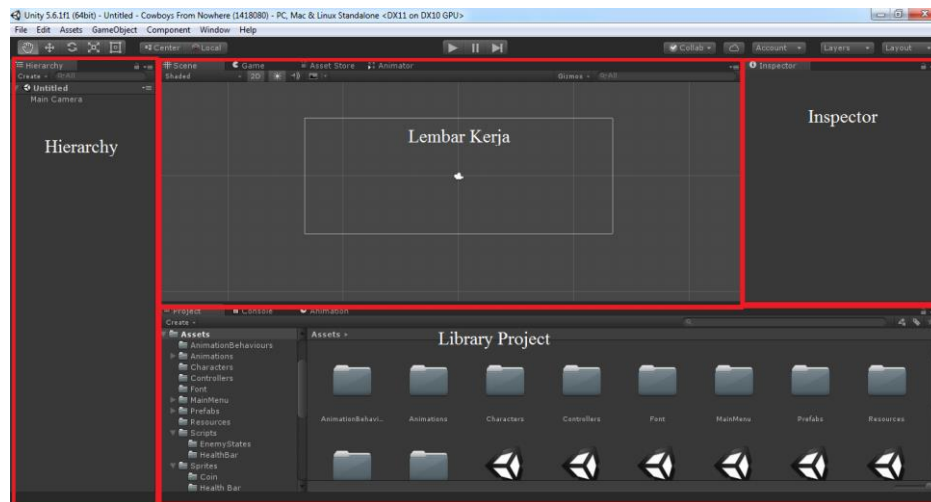
Gambar 3.11 Rancangan Tampilan *Stage Game*

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi *Game*

Implementasi sistem adalah proses penerapan rancangan sistem yang telah dibuat menjadi suatu aplikasi yang bisa dijalankan pada kenyataanya. Disamping itu, implementasi ini juga berfungsi untuk mengetahui tingkat keberhasilan dari rancangan yang telah dibuat.

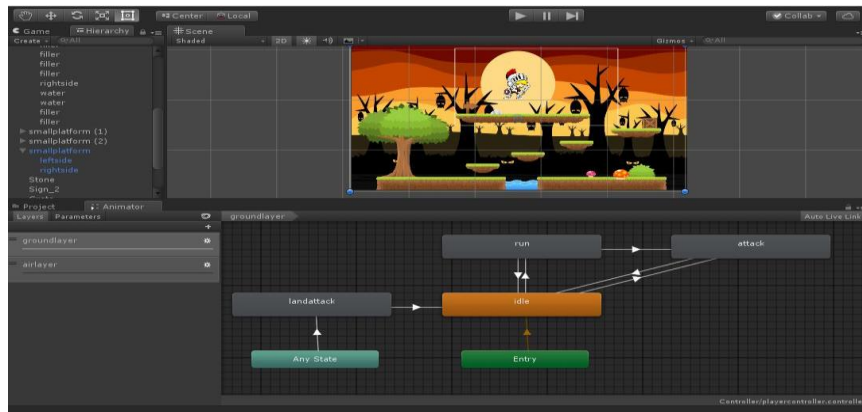


Gambar 4.1 Tampilan Awal *Unity*

Pada Gambar 4.1 terdapat beberapa jendela dengan fungsi yang berbeda beda seperti *hierarchy* dimana berfungsi untuk inialisasi objek pada *game*. Halaman kerja tempat meletakkan objek *game* yang nantinya akan dimanipulasi. *Inspector* merupakan jendela dimana pada jendela ini developer dapat memanipulasi objek yang ada pada *game* seperti *rigidbody*, *collider* dan lain sebagainya. *Library project* berfungsi menyimpan *file* dari *game* yang akan dibuat seperti *script*, karakter, *background* dan lain-lain. Untuk implementasi *game* dibagi menjadi beberapa bagian yang meliputi Implementasi *Control* Pada *Player*, Implementasi Kecerdasan Buatan Pada *Enemy* Level 1, 2, dan 3, Implementasi Kecerdasan Buatan Pada *Boss* Level 1, 2, dan 3.

4.1.1 Fungsi *Controller* Pada *Player*

Fungsi *Controller* pada *player* bisa berupa bergerak, menyerang, menembak, melompat dan menghindar. Merupakan contoh *script C#* yang akan diimplementasikan ke objek *player* agar nantinya dapat dikontrol oleh pemain. Untuk implementasi *script* pada *player* dapat dilihat pada Gambar 4.2



Gambar 4.2 Fungsi *Controller* Pada *Player*

Pada Gambar 4.2 merupakan implementasi dari *script player*. Di mana untuk memasukkan *script* lakukan langkah 1 seperti yang ada pada Gambar 4.2 yaitu memilih objek yang akan dimasukkan *script* dalam hal ini *player*, langkah 2 cari *inspector* yang ada pada sisi bagian kanan jendela kerja. *Inspector* berfungsi untuk memanipulasi segala objek yang ada pada laman kerja seperti memberikan *rigidbody*, *collider*, *script* dan lain sebagainya. Langkah 3 masukkan *script player* ke dalam *inspector*.

4.1.2 Implementasi Kecerdasan Buatan Pada *Enemy Level 1*

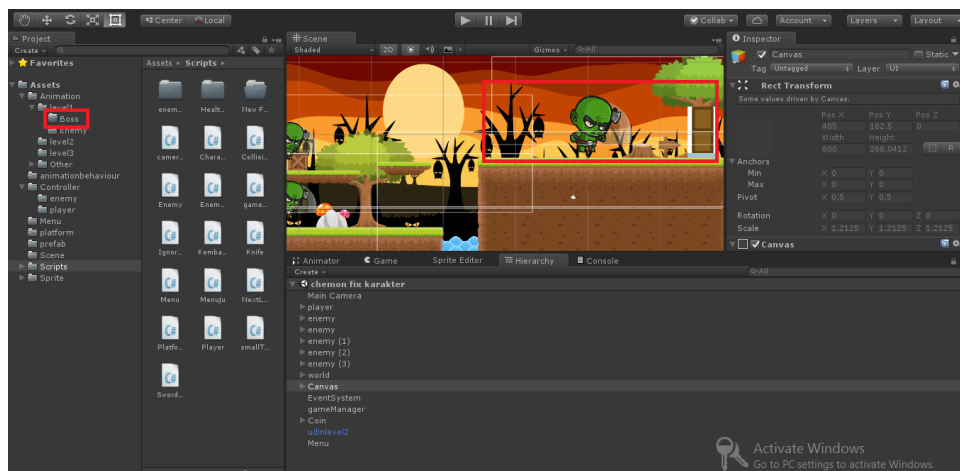
Fungsi kecerdasan buatan pada *enemy Level 1* merupakan fungsi agar *enemy* dapat berperilaku ketika ada aksi dari *player*. Misalnya *player* berada pada jangkauan *sight enemy* maka aksi dari *enemy* bisa mengejar, menyerang dan menembak sesuai aksi yang telah diterapkan pada *enemy*. Untuk fungsi kecerdasan buatan *enemy* dapat dilihat pada Gambar 4.3



Gambar 4.3 Fungsi Kecerdasan Buatan Pada *Enemy Level 1*

4.1.3 Implementasi Kecerdasan Buatan Pada *Boss Level 1*

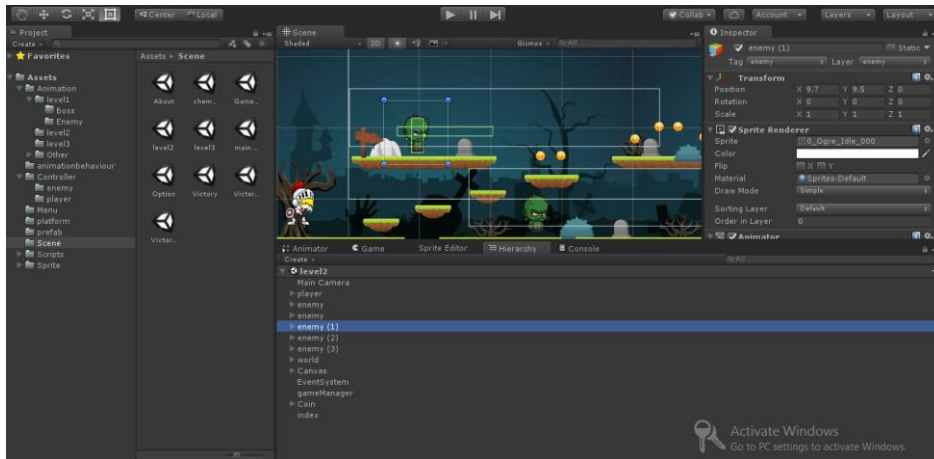
Fungsi kecerdasan buatan pada *Boss Level 1* merupakan fungsi agar *Boss Level 1* dapat berperilaku ketika ada aksi dari *player*. Apabila *player* berada pada jangkauan *sight Boss Level 1* maka aksi dari *Boss Level 1* bisa mengejar, menyerang dan menembak sesuai aksi yang telah diterapkan pada *Boss Level 1*. Untuk fungsi kecerdasan buatan *Boss Level 1* dapat dilihat pada Gambar 4.4



Gambar 4.4 Fungsi Kecerdasan Buatan Pada *Boss Level 1*

4.1.4 Implementasi Kecerdasan Buatan Pada *Enemy Level 2*

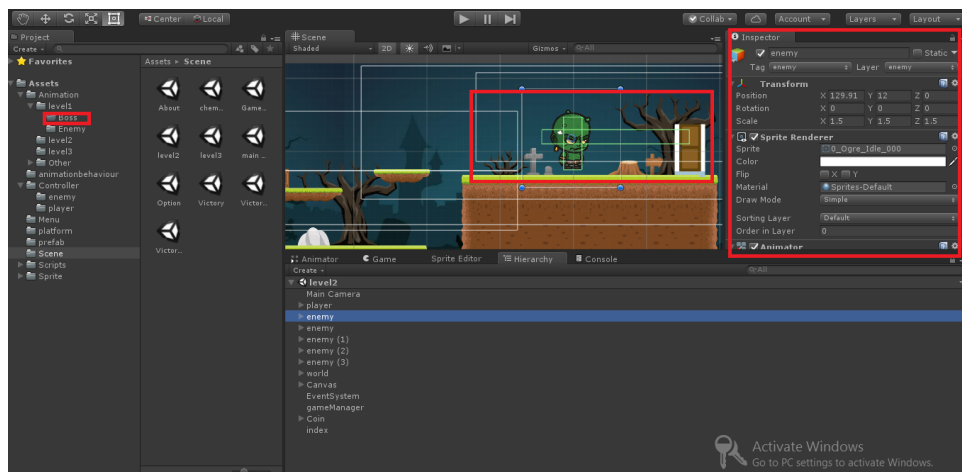
Fungsi kecerdasan buatan pada *enemy Level 2* merupakan fungsi agar *enemy* dapat berperilaku ketika ada aksi dari *player*. Misalnya *player* berada pada jangkauan *sight enemy* maka aksi dari *enemy* bisa mengejar, menyerang dan menembak sesuai aksi yang telah diterapkan pada *enemy*. Untuk fungsi kecerdasan buatan *enemy* dapat dilihat pada Gambar 4.5



Gambar 4.5 Fungsi Kecerdasan Buatan Pada *Enemy* Level 2

4.1.5 Implementasi Kecerdasan Buatan Pada *Boss* Level 2

Fungsi kecerdasan buatan pada *Boss* Level 2 merupakan fungsi agar *Boss* Level 2 dapat berperilaku ketika ada aksi dari *player*. Apabila *player* berada pada jangkauan *sight* *Boss* Level 2 maka aksi dari *Boss* Level 2 bisa mengejar, menyerang dan menembak sesuai aksi yang telah diterapkan pada *Boss* Level 2. Untuk fungsi kecerdasan buatan *Boss* Level 2 dapat dilihat pada Gambar 4.6

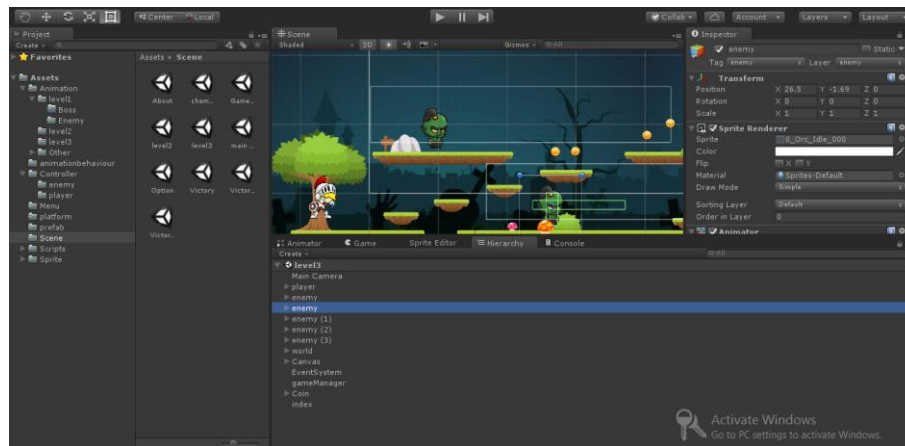


Gambar 4.6 Fungsi Kecerdasan Buatan Pada *Boss* Level 2

4.1.6 Implementasi Kecerdasan Buatan Pada *Enemy* Level 3

Fungsi kecerdasan buatan pada *enemy* Level 3 merupakan fungsi agar *enemy* dapat berperilaku ketika ada aksi dari *player*. Misalnya *player* berada pada jangkauan *sight* *enemy* maka aksi dari *enemy* bisa mengejar, menyerang dan

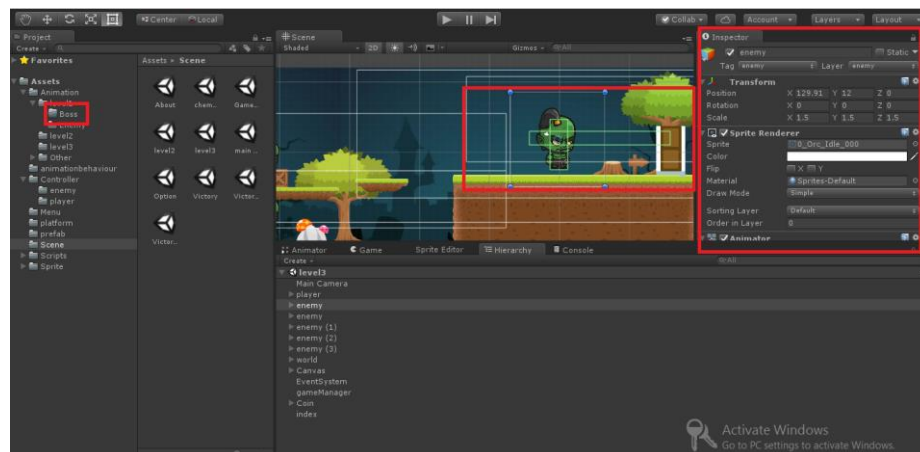
menembak sesuai aksi yang telah diterapkan pada *enemy*. Untuk fungsi kecerdasan buatan *enemy* dapat dilihat pada Gambar 4.7



Gambar 4.7 Fungsi Kecerdasan Buatan Pada *Enemy* Level 3

4.1.7 Implementasi Kecerdasan Buatan Pada *Boss* Level 3

Fungsi kecerdasan buatan pada *Boss* Level 3 merupakan fungsi agar *Boss* Level 3 dapat berperilaku ketika ada aksi dari *player*. Apabila *player* berada pada jangkauan *sight* *Boss* Level 3 maka aksi dari *Boss* Level 3 bisa mengejar, menyerang dan menembak sesuai aksi yang telah diterapkan pada *Boss* Level 3. Untuk fungsi kecerdasan buatan *Boss* Level 3 dapat dilihat pada Gambar 4.8



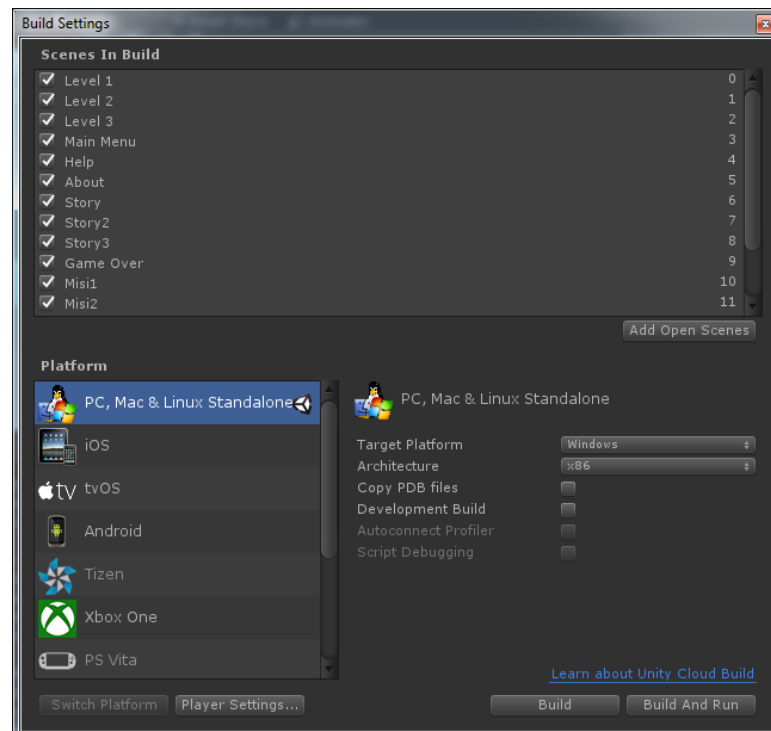
Gambar 4.8 Fungsi Kecerdasan Pada *Boss* Level 3

4.1.8 Implementasi *Build Unity* Menjadi Aplikasi

Implementasi *file unity* menjadi *file .exe* berfungsi agar pengembang dari *game adventure "Castle Of Illusion"* tidak perlu untuk mempublikasikan *file* mentahnya, cukup dengan *file.exe* (*file* jadi) dari *game adventure "Castle Of*

Illusion". Untuk implementasinya dapat dilihat seperti yang ditunjukkan Gambar 4.9 dan Gambar 4.10.

Gambar 4.9 Implementasi *Build Unity* Menjadi *.Exe*



Gambar 4.10 Pengaturan *Platform*

4.2 Pengujian

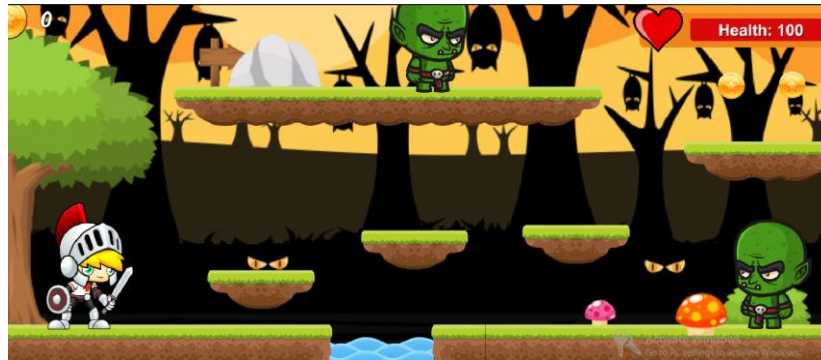
Pengujian adalah tahapan yang dilakukan untuk mengetahui apakah penelitian yang dilakukan telah sesuai dengan perencanaan yang telah dibuat sebelumnya. Pengujian yang dilakukan meliputi pengujian *gameplay* dan pengujian fungsional.

4.2.1 Pengujian *Gameplay*

Pengujian *gameplay* bertujuan untuk menguji bagaimana *game* tersebut berjalan sesuai dengan rancangan *system* yang telah dibuat. Untuk pengujian *gameplay* dilakukan beberapa tahapan pengujian yang meliputi Tampilan *Menu* Utama, *Menu Help*, *Menu About*, *Game Over*, Misi Level 1, Misi Level 2, Misi Level 3, *Result Game*, Tampilan Level 1, Level 2, Level 3, dan Interaksi *Enemy* terhadap *Player* pada Level 1, 2, dan 3.

1. Tampilan Level 1 *Game*

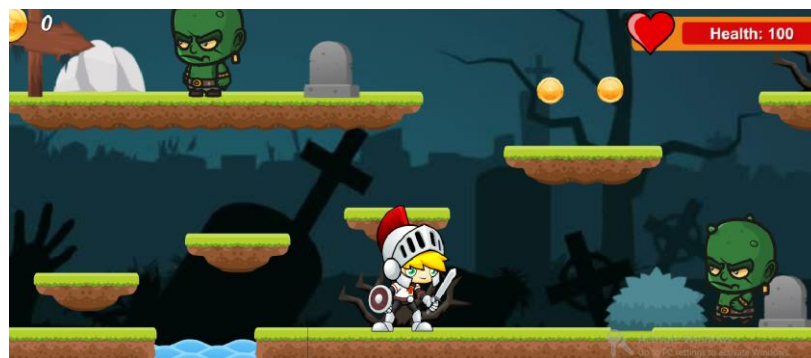
Tampilan lingkungan yang ada pada level 1 *game* “*Castle Of Illusion*” adalah dimana petualangan *cowboys* dimulai, *player* akan berinteraksi langsung dengan karakter ataupun objek yang ada didalam *game*. Tampilan Level 1 *game* “*Castle Of Illusion*” ditunjukkan pada Gambar 4.19.



Gambar 4.11 Tampilan Level 1

2. Tampilan Level 2 *Game*

Tampilan lingkungan yang ada pada level 2 *game* “*Castle Of Illusion*” ini adalah dimana karakter utama yaitu *Cowboys* sudah berada dekat dengan tempat dimana *Boss Alien* berada. Tampilan Level 2 *game* “*Castle Of Illusion*” ditunjukkan pada Gambar 4.20



Gambar 4.12 Tampilan Level 2

3. Tampilan Level 3 *Game*

Tampilan lingkungan yang ada pada level 3 adalah merupakan level dimana *Alien Boss* berada *player* harus mengalahkan *Boss* level yang ada pada level 3 ini untuk dapat menyelesaikan petualangan. Tampilan Level 3 akan ditunjukkan pada Gambar 4.21



Gambar 4.13 Tampilan Level 3

4. Tampilan Interaksi *Enemy Terhadap Player* Pada Level 1

Tampilan Interaksi *Enemy Terhadap Player* pada level 1 ini adalah *Enemy* akan berinteraksi dengan *Player* apabila *Player* menjangkau *sight* dari *Enemy* dimana *Enemy* akan berpatroli apabila *Player* tidak menjangkau *sight Enemy* dan *Enemy* akan menyerang apabila *Player* menjangkau *sight Enemy*. Tampilan interaksi *Enemy* terhadap *Player* Pada Level 1 ditunjukkan pada Gambar 4.22



Gambar 4.14 Tampilan Interaksi *Enemy Terhadap Player* Pada Level 1

5. Tampilan Interaksi *Enemy Terhadap Player* Pada Level 2

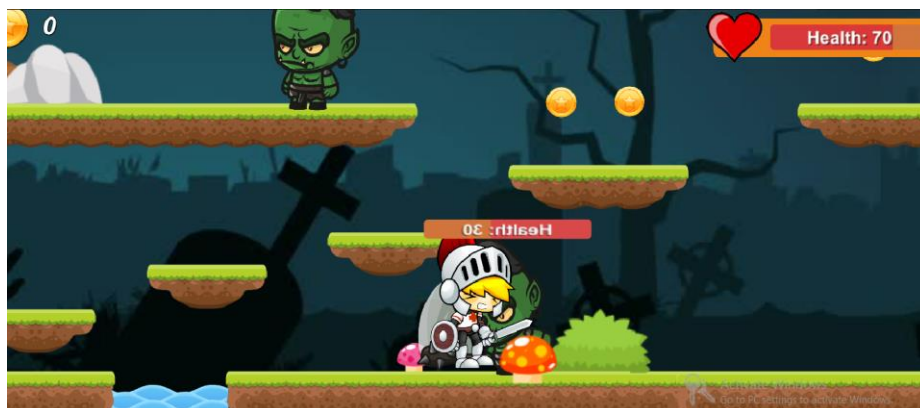
Tampilan Interaksi *Enemy Terhadap Player* pada level 2 ini adalah *Enemy* akan berinteraksi dengan *Player* apabila *Player* menjangkau *sight* dari *Enemy* dimana *Enemy* akan berpatroli apabila *Player* tidak menjangkau *sight Enemy* dan *Enemy* akan menyerang apabila *Player* menjangkau *sight Enemy*. Tampilan interaksi *Enemy* terhadap *Player* Pada Level 2 ditunjukkan pada Gambar 4.23



Gambar 4.15 Tampilan Interaksi *Enemy* Terhadap *Player* Pada Level 2

6. Tampilan Interaksi *Enemy* Terhadap *Player* Pada Level 3

Tampilan Interaksi *Enemy* Terhadap *Player* pada level 3 ini adalah *Enemy* akan berinteraksi dengan *Player* apabila *Player* menjangkau *sight* dari *Enemy* dimana *Enemy* akan berpatroli apabila *Player* tidak menjangkau *sight* *Enemy* dan *Enemy* akan menyerang apabila *Player* menjangkau *sight* *Enemy*. Tampilan interaksi *Enemy* terhadap *Player* Pada Level 3 ditunjukkan pada Gambar 4.24



Gambar 4.16 Tampilan Interaksi *Enemy* Terhadap *Player* Pada Level 3

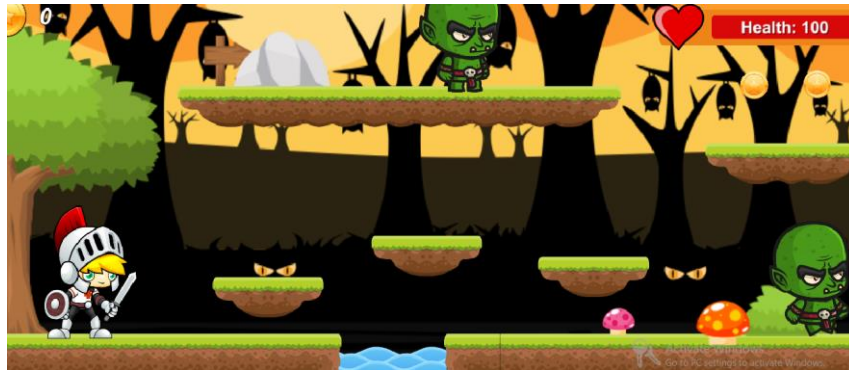
4.2.2 Pengujian Fungsional

Pada tahapan pengujian penulis menampilkan hasil dari pengujian yang telah dilakukan, dimana pengujian tersebut meliputi pengujian AI (*Artificial Intelligence*) *Enemy*, Pengujian *Controller Player*, Pengujian Fungsionalitas, dan Pengujian *User*. Untuk mengetahui apakah semua fungsi dalam game “*Castle Of Illusion*” telah berfungsi semua atau belum, selanjutnya adalah pengujian

terhadap *user*, apakah *game* yang dibuat telah memenuhi aspek-aspek yang dikehendaki oleh penulis.

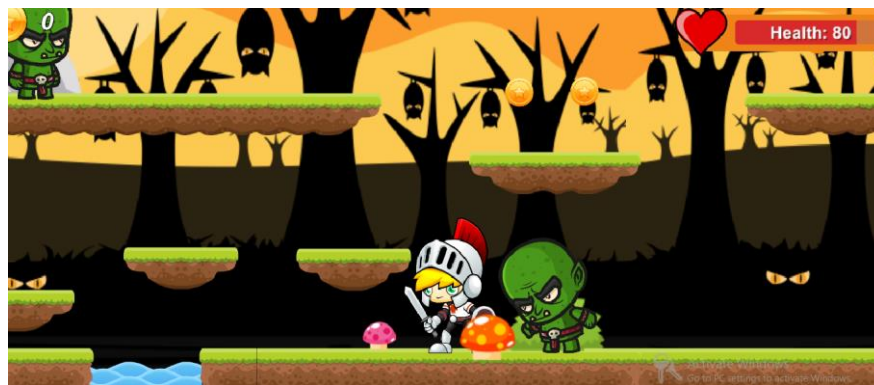
1. Pengujian AI (*Artificial Intelligence*) *Enemy*

Pada bagian ini merupakan hasil pengujian metode *finite state machine* pada *enemy* telah berjalan dengan baik, dimana jika *player* menjangkau *sight enemy*, maka *enemy* akan mengejar dan menyerang *player*.



Gambar 4.17 *Enemy Patrol*

Pada Gambar 4.25 *enemy* akan melakukan *patrol*, apabila *player* tidak menjangkau *sight enemy* yaitu pada jarak > 15 *pixel*.



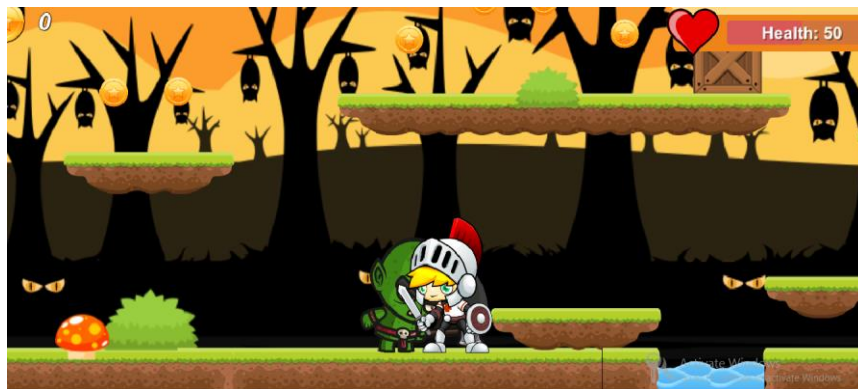
Gambar 4.18 *Enemy Mengejar Player*

Pada Gambar 4.26 *Enemy* akan mengejar *player* pada saat *player* masuk kedalam jangkauan *sight enemy* yaitu pada jarak < 15 *pixel* kemudian saat masuk kedalam *sight enemy* pada jarak < 10 *pixel* maka *enemy* akan menembak *player*, kemudian pada jarak *sight enemy* < 5 *pixel* maka *enemy* akan menyerang *player*.



Gambar 4.19 *Enemy Menembak Player*

Pada Gambar 4.27 *enemy* akan menembak *player* pada saat *player* masuk kedalam jangkauan *enemy* atau *sight enemy* pada jarak $< 10 \text{ pixel}$.



Gambar 4.20 *Enemy Menyerang Player*

Pada Gambar 4.28 *enemy* akan menyerang *player* pada saat *player* masuk kedalam jangkauan *sight enemy* yaitu pada jarak $< 5 \text{ pixel}$.

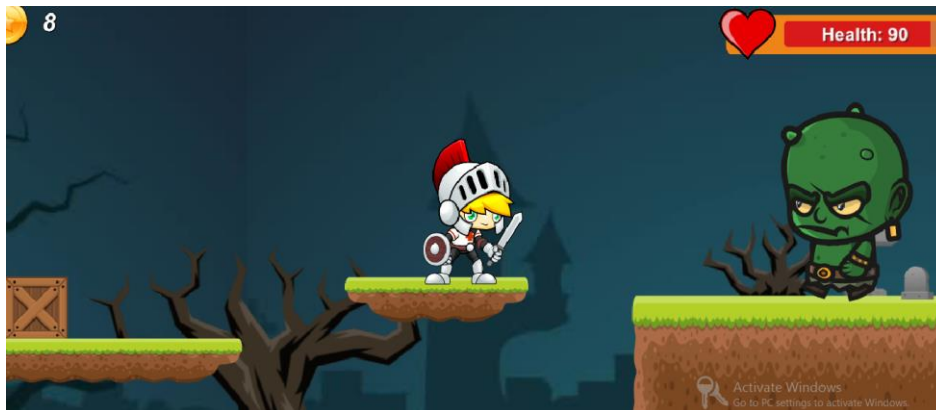
Tabel 4.1 Tabel Pengujian AI Pada *Enemy*

No	Keadaan	Kondisi	Aksi	Keterangan
1	Patroli	$\leq 15 \text{ px}$	Mengejar	Sesuai
2	Mengejar	$\geq 15 \text{ px}$	Patroli	Sesuai
3	Mengejar	$\leq 10 \text{ px}$	Menembak	Sesuai
4	Menembak	$\geq 10 \text{ px}$	Mengejar	Sesuai
5	Menembak	$\leq 5 \text{ px}$	Memukul	Sesuai
6	Memukul	$\geq 5 \text{ px}$	Menembak	Sesuai

Hasil dari pengujian Kecerdasan Buatan yang diterapkan pada *enemy* berjalan dengan baik dan sesuai dengan tingkat prosentase 100%.

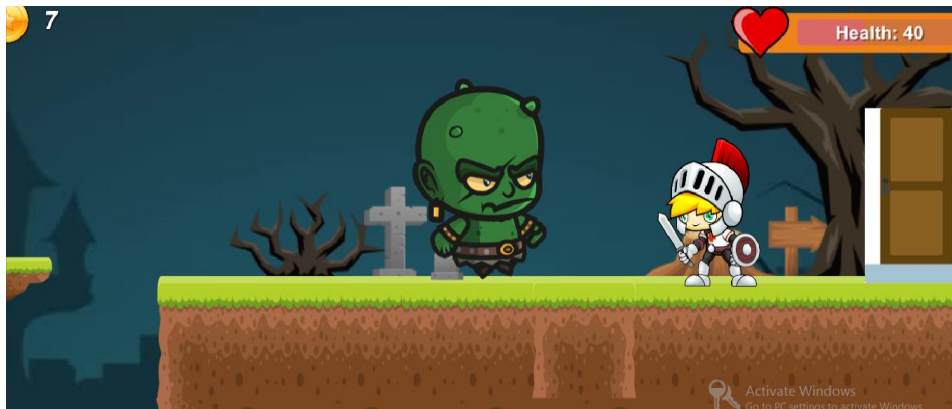
2. Pengujian AI (*Artificial Intelligence*) Pada *Boss Level*

Pada bagian ini merupakan hasil pengujian metode *finite state machine* pada *Boss Level* telah berjalan dengan baik, dimana jika *player* menjangkau *sight Boss Level*, maka *Boss Level* akan mengejar dan menyerang *player*.



Gambar 4.21 *Boss Patrol*

Pada Gambar 4.29 *Boss Level* akan *patrol* pada saat *player* tidak masuk kedalam jangkauan *sight* dari *boss level* yaitu pada jarak > 25 *pixel*.



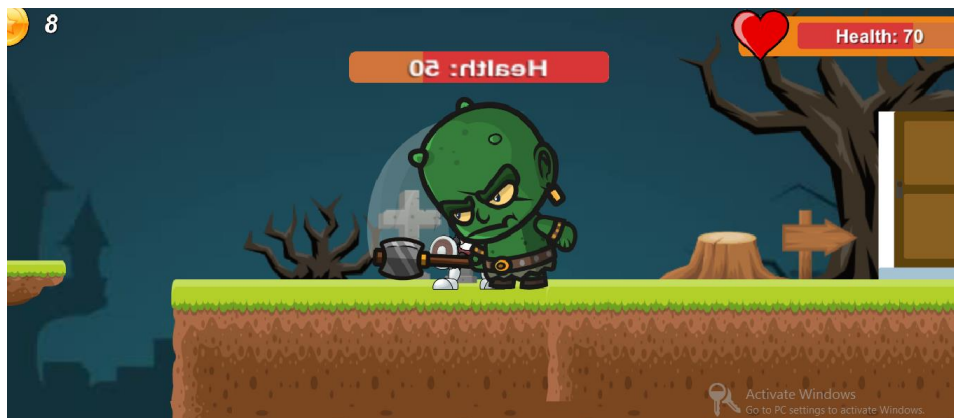
Gambar 4.22 *Boss Mengejar Player*

Pada Gambar 4.30 *Boss Level* akan mengejar *player* apabila *player* masuk kedalam jangkauan *sight* dari *boss level* yaitu pada jarak < 15 *pixel*.



Gambar 4.23 Boss Menembak Player

Pada Gambar 4.31 Boss Level akan menembak *player* apabila *player* masuk kedalam jangkauan atau *sight boss* level yaitu pada jarak < 10 pixel.



Gambar 4.24 Boss Menyerang Player

Pada Gambar 4.32 Boss Level akan menyerang *player* pada saat *player* masuk kedalam jangkauan atau *sight* dari boss level, yaitu pada jarak < 5 pixel.

Tabel 4.2 Tabel Pengujian AI Pada Boss Level

No	Keadaan	Kondisi	Aksi	Keterangan
1	Patroli	≤ 25 px	Mengejar	Sesuai
2	Mengejar	≥ 25 px	Patroli	Sesuai
3	Mengejar	≤ 15 px	Menembak	Sesuai
4	Menembak	≥ 15 px	Mengejar	Sesuai
5	Menembak	≤ 10 px	Memukul	Sesuai
6	Memukul	≥ 10 px	Menembak	Sesuai

Hasil dari pengujian Kecerdasan Buatan yang diterapkan pada *Boss Level* berjalan dengan baik dan sesuai dengan tingkat prosentase 100%.

3. Pengujian *Controller Player*

Pengujian *Controller* pada *Player* berfungsi untuk mengetahui apakah semua *button* yang ada pada *keyboard* yang digunakan untuk bermain dalam game “*Castle Of Illusion*” berfungsi sesuai dengan yang diharapkan atau tidak. Tabel Pengujian *Controller Player* ditunjukkan pada Tabel 4.3

Tabel 4.3 Tabel Pengujian *Controller Player*

No	Keyboard	Fungsi	Output	Berhasil	Gagal
1	<i>Shift</i>	Menyerang (<i>Attack</i>)	<i>Player</i> Menyerang	√	-
2	V	Menembak (<i>Shoot</i>)	<i>Player</i> Menembak	√	-
3	<i>Space</i>	Melompat (<i>Jump</i>)	<i>Player</i> Melompat	√	-
4	⇒	Bergerak ke Kanan (<i>Moving Right</i>)	<i>Player</i> Bergerak ke Kanan	√	-
5	⇐	Bergerak ke Kiri (<i>Moving Left</i>)	<i>Player</i> Bergerak ke Kiri	√	-

Hasil dari pengujian kontrol pada *player* berfungsi dengan baik dan sesuai dengan tingkat prosentase 100%.

4. Pengujian Fungsional

Pada pengujian fungsional ini, *game* dicoba dijalankan pada sistem operasi *windows*. Hasil pengujian fungsional dapat dilihat pada Tabel 4.4.

Tabel 4.4 Tabel Pengujian Fungsional

No	Item Uji	Windows 7	Windows 8
1	<i>Main Menu</i>	√	√
2	<i>Button Play</i>	√	√
3	<i>Button Help</i>	√	√
4	<i>Button About</i>	√	√
5	<i>Button Exit</i>	√	√
6	<i>Level 1 Game</i>	√	√
7	<i>Level 2 Game</i>	√	√
8	<i>Level 3 Game</i>	√	√
9	<i>Health Bar Player</i>	√	√
10	<i>Health Bar Enemy</i>	√	√
11	<i>Backsound Main Menu</i>	√	√
12	<i>Backsound Level 1 Game</i>	√	√
13	<i>Backsound Level 2 Game</i>	√	√
14	<i>Backsound Level 3 Game</i>	√	√
15	<i>Collected Coins</i>	√	√
16	<i>Next Level</i>	√	√
17	<i>Sound Effect</i>	√	√

Hasil dari pengujian Fungsional yang dilakukan pada 4 sistem operasi yang berbeda yaitu *Windows 7* dan *Windows 8* berjalan dengan baik dan sesuai dengan tingkat prosentase 100% semua item yang diujikan yaitu *Main Menu*, *Button Play*, *Button Help*, *Button About*, *Level 1 Game*, *Level 2 Game*, *Level 3 Game*, *Health Bar Player*, *Health Bar Enemy*, *Backsound Main Menu*, *Backsound Level 2 Game*, *Backsound Level 3 Game*, *Collected Coins*, *Next Level*, *Sound Effect*.

5. Pengujian *User*

Pengujian *User* dilakukan untuk mengetahui apakah aplikasi yang dibuat sudah berjalan dengan baik atau tidak. Pengujian *user* dilakukan pada 10 mahasiswa ITN Malang. Berikut ini pengujian *user* ditunjukkan pada Tabel 4.5

Tabel 4.5 Tabel Pengujian *User*

No	Pertanyaan	Penilaian		
		Kurang	Cukup	Baik
1.	Apakah tampilan <i>game adventure</i> “ <i>Castle Of Illusion</i> ” sudah menarik untuk <i>user</i> ?	1 Orang	9 Orang	-
2.	Apakah <i>game adventure</i> “ <i>Castle Of Illusion</i> ” mudah dipahami untuk <i>user</i> ?	2 Orang	4 Orang	4 Orang
3.	Apakah <i>Controller Player</i> dalam <i>game adventure</i> “ <i>Castle Of Illusion</i> ” sudah dapat digunakan dengan penggunaan <i>user</i> ?	1 Orang	4 Orang	5 Orang
4.	Apakah <i>enemy</i> sudah mempunyai kecerdasan buatan mengejar saat <i>player</i> mendekat dengan jarak tertentu saat digunakan dengan penggunaan <i>user</i> ?	3 Orang	6 Orang	1 Orang
5.	Apakah <i>enemy</i> sudah mempunyai kecerdasan buatan menyerang saat <i>player</i> mendekat dengan jarak tertentu saat digunakan dengan penggunaan <i>user</i> ?	2 Orang	6 Orang	2 Orang
Jumlah		9 (18%)	29 (58%)	12 (24%)

Berdasarkan Tabel 4.5, didapatkan hasil pengujian *user* sebagai berikut, kurang Sebesar 18%, Cukup Sebesar 58%, dan Baik Sebesar 24%. Oleh karena itu hasil pengujian terhadap 10 orang responden cenderung menilai *game* dengan nilai cukup, yaitu dengan nilai prosentase 58%.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah pembuatan *Game Adventure "Castle Of Illusion"*, maka penulis dapat mengambil kesimpulan :

1. Hasil dari pengujian Kecerdasan Buatan yang diterapkan pada *enemy* berjalan dengan baik dan sesuai dengan tingkat prosentase 100%.
2. Hasil dari pengujian kontrol pada *player* berfungsi dengan baik dan sesuai dengan tingkat prosentase 100%.
3. Hasil dari pengujian Fungsional yang dilakukan pada 4 sistem operasi yang berbeda yaitu *Windows 7* dan *Windows 8* berjalan dengan baik dan sesuai dengan tingkat prosentase 100%.
4. Berdasarkan Tabel 4.5, didapatkan hasil pengujian *user* sebagai berikut, Kurang Sebesar 18%, Cukup Sebesar 58%, dan Baik Sebesar 24%. Oleh karena itu hasil pengujian terhadap 10 orang responden cenderung menilai *game* dengan nilai cukup, yaitu dengan nilai 58%.

5.2 Saran

Setelah dilakukan pengujian terhadap *game adventure "Castle Of Illusion"* maka masih ada kekurangan sehingga untuk pengembangan lebih lanjut disarankan :

1. Harapan penulis semoga aplikasi ini dapat di kembangkan dengan penambahan fitur- fitur yang lebih menarik.
2. *Game adventure "Castle Of Illusion"* dapat dijalankan pada berbagai macam *platform* yang berbeda.
3. Dapat mengembangkan lagi *game adventure "Castle Of Illusion"* dengan menambah level, *rank point*, *save game*, dan *multiplayer game*

DAFTAR PUSTAKA

- Adi Wijaya, Surya., Susi Juniastuti, Supeno Mardi SN, dan Moch. Hariadi. 2009. *Desain Fuzzy State Machine Untuk Menghasilkan Variasi Respon NPC (Non-Playable Character) Pada Sebuah Game*. Program Studi MMT-ITS.
- AI. Tridhonanto dan Beranda Agency. 2011. "OPTIMALKAN POTENSI ANAK DENGAN GAME". Jakarta. PT Elex Media Komputindo.
- Ardi, R., 2012. *Pembuatan Game First Person Shooter (FPS) "Operation Zygma Force" Menggunakan FPS Creator X10*. Yogyakarta : STMIK AMIKOM.
- Chandra, Y., Sabloak, S., & Angreni, R. (2014). Penerapan Algoritma Minimax dan Breadth First Search pada Permainan Othello Menggunakan Framework Phonegap.
- Dahria, M., 2008. Kecerdasan Buatan (*Artificial Intelligence*). Jurnal Saintikom, 5(2), p.1.
- Fajar, M., Ningrum, I. P., & Ramadhan, R. (2015). MEMBANGUN APLIKASI GAME "DUA SATU (21)" BERBASIS ANDROID MENGGUNAKAN METODE DEPTH FIRST SEARCH. *semanTIK*, 1(2).
- Marzian, Fadel., Mukti Qamal. (2017) *Game RPG "The Royal Sword" Berbasis Desktop Dengan Menggunakan Metode Finite State Machine (FSM)*. Teknik Informatika Universitas Malikussaleh Lhokseumawe.
- Maulana, P., Sofwan, A., & Isnanto, R. R. (2011). *KECERDASAN BUATAN DALAM PERMAINAN SNAKE 3D MENGGUNAKAN VISUAL BASIC. NET DAN DIRECTX* (Doctoral dissertation, Jurusan Teknik Elektro Fakultas Teknbik Undip).
- Rahadian, M.F., Suyatno, A. and Maharani, S., 2017. PENERAPAN METODE *FINITE STATE MACHINE* PADA GAME "THE RELATIONSHIP".
- Setiawan, I. 2006. Perancangan *Software Embedded System* Berbasis FSM. Semarang: Universitas Diponegoro.

Sholichin, R., Yasindan, M., & Oktoviana, L. T. (2012). Implementasi Algoritma Dijkstra Dalam Pencarian Lintasan Terpendek Lokasi Rumah Sakit, Hotel Dan Terminal Kota Malang Berbasis Web. *Jurnal Online Universitas Negeri Malang,(Online)*.