

GAME RESCUE IDOL DENGAN MENGGUNAKAN METODE FINITE STATE MACHINE (FSM)

Danar Pradiktia Yunanta

Teknik Informatika, Institut Teknologi Nasional Malang
pradiktiadanar@gmail.com

ABSTRAKS

Terdapat media untuk bermain *game* yaitu konsol *game computer* dan *smartphone*. *Computer* salah satunya media yang paling banyak digunakan para pengembang industri *game* untuk mengeluarkan produknya. Dalam *game*, musuh dan karakter merupakan salah satu elemen yang paling penting dalam *video game*, untuk menjadikan *game* semakin menarik, karakter dan NPC (*Non Playable Character*), salah satunya musuh dalam *game* harus dapat meniru sifat atau perilaku manusia dalam dunia nyata. Untuk melakukan hal tersebut diperlukan kecerdasan buatan pada karakter NPC (*Non Playable Character*) pada musuh yaitu kecerdasan buatan FSM (*Finite State Machine*) yang menggunakan kecerdasan buatan FSM (*Finite State Machine*) pada karakter musuh agar dapat mendeteksi keberadaan player, dan juga kecerdasan buatan *Fuzzy Logic* yang digunakan untuk *output* aksi serangan musuh terhadap player.

Dalam perancangan *game* ini, penulis menggunakan aplikasi berupa *Unity3D* dengan bahasa pemrograman C#. Penulis dalam membuat *game* ini menerapkan metode kecerdasan buatan *Finite State Machine* yang menggunakan 3 hal yaitu keadaan, kejadian, dan aksi. Penulis juga menerapkan kecerdasan buatan *Fuzzy logic* yang digunakan pada karakter bos (*Non Playable Character*) untuk menentukan aksi dengan menggunakan 2 masukan berupa *health point* pada player dan jarak player ke bos.

Dari pembuatan *game* yang telah dilakukan, implementasi FSM (*Finite State Machine*) dapat diterapkan pada *game* 2 dimensi bergenre *adventure* dengan indikasi musuh dapat mengejar dan menyerang player dengan kondisi tertentu. Kecerdasan buatan yang diterapkan pada musuh sudah berjalan seluruhnya. Semua fungsi tombol pada menu utama sudah berjalan seluruhnya. Semua fungsi tombol yang diterapkan untuk melakukan aksi pada player sudah berjalan. Pengujian *game* di 3 laptop yaitu Asus, Lenovo, dan Acer dan juga 2 OS yaitu windows 8 dan 10 dapat berjalan dan sesuai. Pengujian user dengan 20 orang di Malang dapat dikatakan bahwa 42% mengatakan baik, 40% mengatakan cukup, dan 18% mengatakan kurang. Pengujian *cheat* yang sudah dibuat dapat berjalan seluruhnya.

Kata kunci : *Game, Unity3D, Finite State Machine, Fuzzy Logic, NPC*

1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dan informasi yang semakin pesat khususnya pada teknologi *computer* memberikan pengaruh besar pada berbagai bidang kehidupan, salah satunya adalah perkembangan *game*. Teknologi tidak harus dibatasi hanya untuk penggunaan mesin. Bersamaan dengan perkembangan teknologi dibidang elektronika munculah media baru yaitu komputer. Hal inilah yang membuat perkembangan *game* di *computer* sangat cepat. *Game* bukan hanya sekedar permainan untuk hiburan atau meningkatkan kreatifitas dan tingkat intelektual para penggunanya, pada saat ini tidak sedikit yang menjadikan *game* sebagai sebuah lapangan pekerjaan.

Metode yang dipakai dalam pembuatan *game* "Rescue Idol" ini menggunakan metode FSM atau *Finite State Machine* dan *Fuzzy Logic*. Dimana metode tersebut bagian dari metode kecerdasan buatan atau *artificial intelligence* untuk pengambilan keputusan karakter NPC (*Non Playable Character*) yaitu karakter yang digerakan oleh kecerdasan buatan yang digunakan untuk mendukung *game* tersebut

Pada *game* biasanya FSM dan *fuzzy logic* digunakan pada karakter NPC (*Non Playable Character*) yaitu karakter yang digerakan oleh kecerdasan buatan yang digunakan untuk mendukung *game* tersebut. Seperti karakter musuh, *fuzzy logic* dapat digunakan untuk menentukan gerakan dan aksi dari musuh tanpa melibatkan pengguna *game*. (Utomo, 2015).

Dari uraian di atas penulis ingin mengimplementasikan penggunaan metode FSM dan *Fuzzy Logic* untuk melakukan penelitian dalam pembuatan *game* untuk tugas akhir skripsi dengan judul "Rescue Idol" yang merupakan *game* 2D dengan Genre *Adventures* menggunakan *Unity3D* sebagai *game engine*. *Game* ini bercerita tentang seorang fans yang mempunyai keinginan untuk menyelamatkan idolanya yang diculik oleh penjahat. Dalam *game* ini penulis berharap nantinya *game* ini dapat digunakan untuk media hiburan interaktif dan meningkatkan daya kreatifitas pemain.

1.2 Rumusan Masalah

Dari latar belakang tersebut maka dicari suatu pemecah masalah yaitu :

1. Bagaimana merancang *game Rescue Idol* 2D dengan menggunakan *game engine* Unity3D ?.
2. Bagaimana mengimplementasikan metode *Finite State Machine* pada karakter musuh?.
3. Bagaimana mengimplementasikan metode *Fuzzy Logic* pada tindakan bos.

1.3 Batasan Masalah

Batasan masalah dari pembuatan game ini adalah :

1. Jumlah misi dalam menyelesaikan game ini akan dibuat sebanyak dua level
2. Metode *Finite State Machine* diimplementasikan pada karakter musuh untuk menyerang karakter pemain jika memasuki jarak atau range dari musuh.
3. Metode *Fuzzy Logic* diimplementasikan pada bos musuh pada rintangan terakhir di tiap tahapan level.
4. *Game* ini dibuat dengan model 2D.
5. *Game* ini dibuat menggunakan *tools* dan *game engine* Unity3D.
6. Cerita dari game ini merupakan cerita fiksi yang dibuat oleh penulis.

1.4 Tujuan

Tujuan dari pembuatan game ini adalah Menerapkan metode *Finite State Machine* dan *Fuzzy Logic* untuk menentukan gerakan pada musuh dan tindakan bos di *game Rescue Idol*.

2. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Salah satu unsur yang berperan penting dalam sebuah *game* adalah kecerdasan buatan atau *Artificial Intelligence* (AI) yang merupakan suatu program komputer yang bertindak dan berpikir seperti manusia dan juga bertindak dan berpikir secara rasional pada saat yang bersamaan. Dengan kecerdasan buatan, elemen-elemen dalam *game* dapat berperilaku seperti manusia. *Game playing* (permainan game) merupakan bidang AI yang sangat populer berupa permainan antara manusia melawan mesin yang memiliki intelektual untuk berpikir. Komputer dapat bereaksi dan menjawab tindakan-tindakan yang diberikan oleh lawan mainnya (Utari, Dely. 2012).

Game "Rescue Idol" ini mengadopsi tipe *side scrolling game*, yaitu game dengan sudut pandang kamera dari sebelah samping. Dan game ini menggunakan 2 metode, Untuk karakter musuh menggunakan metode FSM (*Finite State Machine*), 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, 2016).

Untuk karakter bos musuh di *game "Rescue Idol"* menggunakan metode *Fuzzy Logic* untuk

menentukan gerakan dan aksi yang dikeluarkan berdasarkan *health point* dan jarak, *Fuzzy Logic* sendiri adalah peningkatan dari logika Boolean yang berhadapan dengan konsep kebenaran sebagian. Saat logika klasik menyatakan bahwa segala hal dapat diekspresikan dalam istilah biner (0 atau 1, hitam atau putih, ya atau tidak), logika fuzzy menggantikan kebenaran boolean dengan tingkat kebenaran. (Wardani, K. E. 2015).

2.2 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 pemain. Beberapa *game* yang beredar saat ini terdapat unsur mendidik, ketangkasan dan ada unsur kekerasan, maka ketika game itu diperjual belikan terdapat batasan umur pemakainya (Ardi, 2012)

2.3 Kecerdasan Buatan (Artificial Intelligence)

Kecerdasan buatan (*artificial intelligence*) merupakan salah satu bagian dari ilmu computer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik dari pada yang dilakukan manusia.

Menurut McCarthy (1956), AI adalah untuk mengetahui dan memodelkan prose-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. manusia cerdas dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki tentu lebih mampu menyelesaikan permasalahan. Tapi bekal pengetahuan tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil keputusan berdasarkan pengetahuan dan pengalaman yang dimiliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan pengalaman yang memadai manusia tidak bisa menyelesaikan masalah dengan baik (Dahria, 2008)

2.4 FSM (Finite State Machine)

Finite state machine adalah sebuah metodologi perancangan system control yang menggambarkan tingkah laku atau prinsip kerja dengan menggunakan tiga hal yang meliputi : *State* (keadaan), *Event* (Kejadian) dan *Action* (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, *system* akan berada pada salah satu state yang aktif. *System* 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 (misal interupsi timer).

Transisi keadaan ini umumnya juga disertai aksi oleh aksi yang dilakukan oleh system ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relative kompleks. Berdasarkan sifatnya metode FSM ini sangat cocok untuk digunakan sebagai basis perancangan perangkat lunak pengendalian yang reaktif dan real time. Salah satu keuntungan nyata penggunaan FSM adalah kemampuannya dalam mendekomposisi aplikasi yang relatif besar dengan hanya menggunakan sejumlah kecil *item state*.

Selain untuk bidang *control*, penggunaan metode ini pada kenyataannya juga umum digunakan sebagai basis untuk perancangan protokol-protokol komunikasi, perancangan perangkat lunak game, aplikasi WEB dan sebagainya, dalam bahasa pemrograman procedural seperti bahasa C, FSM umumnya direalisasikan menggunakan *statemen control switch case* atau */ dan if Then*. Dengan menggunakan *statemen-statemen kontrol* ini, aliran program secara praktis akan mudah dipahami dan dilacak jika terjadi kesalahan logika (Setiawan, 2006)

2.5 Fuzzy Logic

Fuzzy secara bahasa diartikan sebagai kabur atau samar – samar. Suatu nilai dapat bernilai besar atau salah secara bersamaan, dalam fuzzy dikenal dengan derajat keanggotaan yang memiliki rentang nilai 0 (nol) hingga 1 (satu). Berbeda dengan himpunan tegas yang memiliki nilai 1 atau 0 (ya atau tidak).

Logika *fuzzy* merupakan suatu logika yang memiliki nilai kekaburan atau kesamaran antara benar atau salah. Dalam teori logika *fuzzy* suatu nilai bisa bernilai benar atau salah secara bersamaan. Namun berapa besar keberaddan dan kesalahan suatu tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* memiliki derajat keanggotaan 0 hingga 1. Berbeda dengan logika digital yang hanya memiliki dua nilai 1 atau 0. Logika *fuzzy* digunakan untuk menterjemahkan suatu besaran yang diekpresikan menggunakan bahasa misalkan besaran kecepatan laju kendaraan yang diekpresikan dengan pelan, agak cepat, cepat dan sangat cepat (Hasanah, 2014)

3. ANALISIS DAN PERANCANGAN

3.1 Story Line

Game ini menceritakan tentang seorang fans yang mempunyai idola. Untuk alur cerita game ini yaitu :

1. Alkisah di ceritakan seorang anak yang mempunyai seorang idola yang sangat dia idolakan.
2. Pada suatu hari anak itu mendapat kabar kalau sang idola diculik oleh kawanan penjahat.
3. Pada hari itu juga anak tersebut memulai petualangannya menyelamatkan sang idola yang diculik.
4. Mereka harus menempuh perjalanan yang sulit dan panjang melewati gunung dan hutan.

5. Mereka harus melawan beberapa penjahat dan halang rintang.
6. Hingga akhirnya mereka akan melawan bos dari penjahat yang menyandra idolanya.
7. Setelah berhasil menyelamatkan idolanya yang disandra oleh bos.
8. Mereka kemudian bahagia karena dapat menyelamatkan idolanya yang diculik dari kawanan penjahat.

3.2 Desain Level

a. Tujuan

Tujuan dari *game* “Rescue Idol” yaitu menyelamatkan idola dari musuh-musuh yang ada pada tiap level.

b. Start

1. *Player* memulai dari level 1 dengan memiliki *health point* (hp) dalam bentuk bar.
2. *Player* memiliki 3 nyawa.
3. *Player* mempunyai aksi berupa serangan.
4. Jika *health point player* habis, *player* mengulang di level 1.
5. Jika nyawa habis (*game over*), *player* mengulang dari level 1.
6. *Player* mengalahkan musuh-musuh yang diterapkan kecerdasan buatan FSM dengan serangan dekat.
7. *Player* dapat menghindari dari jebakan yang ada.
8. *Player* dapat menambah *health point* jika menyentuh *item* yang dapat menambah *health point player*.
9. *Player* dapat menambah skor jika menyentuh item koin.

c. Middle

1. *Player* menuju level 2 untuk menyelamatkan idolanya dari tangan bos.
2. *Player* juga memiliki *health point* dalam bentuk bar
3. *Player* memiliki 3 nyawa.
4. *Player* mempunyai aksi berupa serangan.
5. *Player* akan mati jika *health point* habis.
6. Jika *health point player* habis, *player* mengulang di level 2.
7. Jika nyawa habis (*game over*), *player* mengulang dari level 1.
8. *Player* dapat menghindari dari jebakan yang ada.
9. *Player* mengalahkan musuh-musuh yang diterapkan kecerdasan buatan FSM dengan serangan dekat dan serangan jauh.
10. *Player* mengalahkan bos yang diterapkan kecerdasan buatan *Fuzzy Logic*.
11. *Player* dapat menambah *health point* jika menyentuh *item* yang dapat menambah *health point player*.
12. *Player* dapat menambah skor jika menyentuh item koin.

d. Ending

Player berhasil mengalahkan bos dan menyelamatkan idolanya.

3.3 Perancangan Karakter dan Jebakan

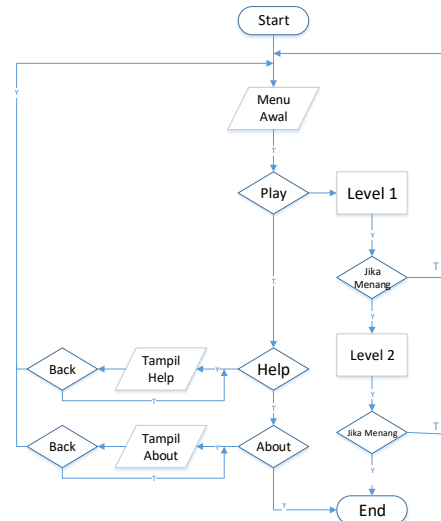
Perancangan karakter merupakan pembahasan mengenai karakter yang terlibat dalam *Game Rescue Idol*. Karakter pada *game Rescue Idol* pada Tabel 3.1.

Tabel 3.1 Keterangan karakter dan jebakan

No	Gambar	Keterangan
1.		Player
2.		Musuh pada level 1 dan 2 dengan Health Point 30
3.		Musuh penjahat pada level 2 dan bos pada level 1 dengan Health Point 50
4.		Bos penjahat pada level terakhir yaitu di level 2 dengan Health Point 100
5.		Idola yang diselamatkan
6.		3 tipe jebakan pada level 1 dan 2

3.4 Flowchart Game

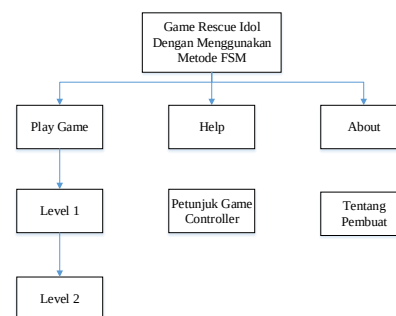
Pada tahap perancangan diagram alir game berfungsi untuk mengetahui alur proses dari program dimulai dari *start* program hingga *end* program. Program dimulai dari *start* kemudian masuk ke 3 menu utama yaitu *Play*, *Help*, dan *About*. Jika permainan memulai *Play* maka langsung saja menuju level 1, jika selesai maka melanjutkan ke level 2. Jika permainan memilih *Help* maka akan muncul bantuan dalam bermain. Jika permainan memilih *About* maka akan muncul tampilan profil pembuat game. Flowchart game seperti Gambar 3.1



Gambar 3.1 flowchart game

3.5 Struktur Menu

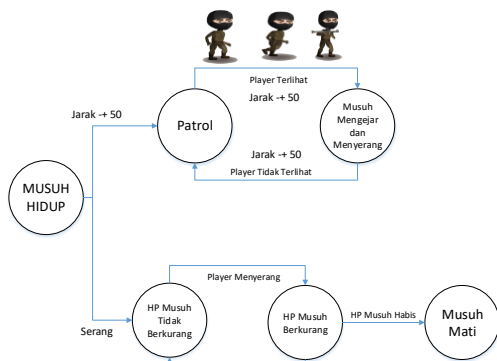
Pada *Game Rescue Idol* Dengan Menggunakan Metode FSM terdiri dari beberapa menu, yaitu *Play*, *Help*, dan *About*. Diagram struktur menu seperti gambar 3.2



Gambar 3.2 struktur menu

3.6 Penerapan Finite State Machine pada Game

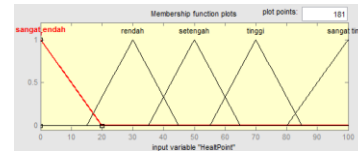
Finite state machine merupakan salah satu logika penalaran yang memperlihatkan perilaku system dengan berdasarkan tiga hal, yaitu *state* (keadaan), *event* (kejadian), dan *action* (aksi). Pada suatu saat, system akan berada pada salah satu state yang aktif. Sistem dapat beralih atau bertransisi menuju state lain jika mendapatkan masukan atau event tertentu. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh system ketika menanggapi masukan yang terjadi. Pada game *Rescue Idol* ini metode *Finite State Machine* diterapkan pada karakter musuh. Alur kecerdasan buatan *finite state machine* seperti gambar 3.3

Gambar 3.3 alur *finite state machine* pada musuh

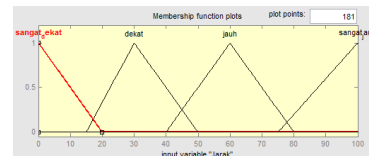
3.7 Fuzzy Logic Rule, Input, dan Output

Fuzzy logic pada game Petualangan Rescue Idol di terapkan pada karakter bos, *fuzzy rule* pada game ini sebagai berikut :

1. **JIKA** health_point Sangat Rendah **DAN** Jarak Sangat Dekat **MAKA** Pukul Dekat.
2. **JIKA** health_point Sangat Rendah **DAN** Jarak Dekat **MAKA** Pukul Dekat.
3. **JIKA** health_point Sangat Rendah **DAN** Jarak Jauh **MAKA** Pukul Dekat.
4. **JIKA** health_point Sangat Rendah **DAN** Jarak Sangat Jauh **MAKA** Pukul Dekat.
5. **JIKA** health_point Rendah **DAN** Jarak Sangat Dekat **MAKA** Pukul Jauh.
6. **JIKA** health_point Rendah **DAN** Jarak Dekat **MAKA** Pukul Jauh.
7. **JIKA** health_point Rendah **DAN** Jarak Jauh **MAKA** Tembak Jauh.
8. **JIKA** health_point Rendah **DAN** Jarak Sangat Jauh **MAKA** Tembak Jauh.
9. **JIKA** health_point Sedang **DAN** Jarak Sangat Dekat **MAKA** Pukul Dekat.
10. **JIKA** health_point Sedang **DAN** Jarak Dekat **MAKA** Pukul Dekat.
11. **JIKA** health_point Sedang **DAN** Jarak Jauh **MAKA** Tembak Jauh.
12. **JIKA** health_point Setengah **DAN** Jarak Sangat Jauh **MAKA** Tembak Jauh.
13. **JIKA** health_point Tinggi **DAN** Jarak Sangat Dekat **MAKA** Pukul Dekat.
14. **JIKA** health_point Tinggi **DAN** Jarak Dekat **MAKA** Pukul Dekat.
15. **JIKA** health_point Tinggi **DAN** Jarak Jauh **MAKA** Tembak Jauh.
16. **JIKA** health_point Tinggi **DAN** Jarak Sangat Jauh **MAKA** Tembak Jauh.
17. **JIKA** health_point Sangat Tinggi **DAN** Jarak Sangat Dekat **MAKA** Pukul Dekat.
18. **JIKA** health_point Sangat Tinggi **DAN** Jarak Dekat **MAKA** Pukul Dekat.
19. **JIKA** health_point Sangat Tinggi **DAN** Jarak Jauh **MAKA** Tembak Jauh.
20. **JIKA** health_point Sangat Tinggi **DAN** Jarak Sangat Jauh **MAKA** Tembak Jauh.

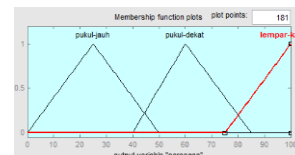
Tabel 3.2 fungsi keanggotaan *health point*

Fungsi Keanggotaan	Domain	Parameter
Sangat Rendah	0 – 20	[0,20]
Rendah	15 – 45	[15,30,45]
Sedang	35 – 65	[35,50,65]
Tinggi	55 – 85	[55,70,85]
Sangat Tinggi	80 – 100	[80,100]



Tabel 3.3 fungsi keanggotaan jarak

Fungsi Keanggotaan	Domain	Parameter
Sangat Dekat	0 – 20	[0, 20]
Dekat	15 – 50	[15, 30, 50]
Jauh	40 – 80	[40, 60, 80]
Sangat Jauh	75 - 100	[75,100]



Tabel 3.4 fungsi keanggotaan aksi

Fungsi Keanggotaan	Domain	Parameter
Pukul Jauh	0 – 50	[10, 25, 50]
Pukul Dekat	40 – 85	[40, 60, 85]
Tembak Jauh	75 - 100	[75, 100]

4. HASIL DAN PEMBAHASAN

4.1 Tampilan Menu Utama

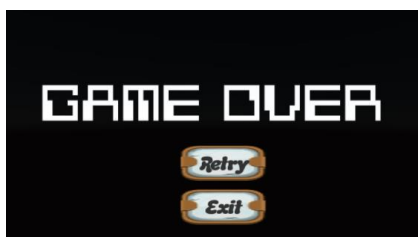
Tampilan menu utama adalah tampilan awal yang akan muncul pada saat pemain menjalankan *Game Rescue Idol*. Pada tampilan awal ini berisikan tombol *Play Game*, *Help*, *About..* *Play Game* untuk memulai game baru, Tombol *Help* berisi cara *controller player*, tombol *about* berisi tentang pembuat *game*. tampilan menu utama seperti gambar 4.1



Gambar 4.1 tampilan menu utama

4.2 Tampilan Game Over

Tampilan *game over* adalah tampilan yang muncul ketika *player* mati pada suatu level. Jika mengeklik *Retry* maka otomatis akan menuju level 1 dan jika mengeklik *exit* maka akan kembali ke menu seperti gambar 4.2



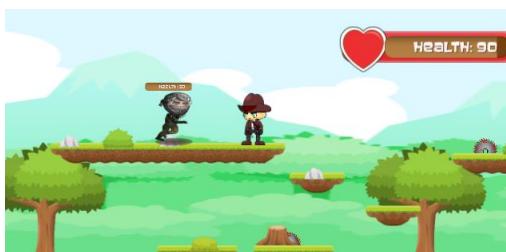
Gambar 4.2 tampilan game over

4.3 Pengujian Gameplay

4.3.1 Level 1

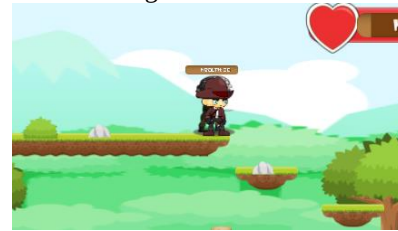
Pengujian *gameplay* adalah pengujian bagaimana game tersebut berjalan sesuai dengan rancangan *sistem* yang telah dibuat. Tampilan pengujian *game* menggambarkan interaksi antara karakter dengan lingkungan didalam *game*. Pada *game* ini karakter memiliki beberapa aksi yaitu jalan, loncat, *attack*. Pada awal permainan *player* akan melihat penjahat yang sudah terintegrasi oleh kecerdasan buatan FSM (*Finite State Machine*) yang mempunyai beberapa kondisi, dan dapat menyerang *player* dengan jarak dekat maupun jarak jauh.

Pada gambar 4.3 menggambarkan interaksi antara karakter dengan musuh dimana musuh melihat keberadaan *player* dan melakukan *output* serangan jarak dekat, jika dari hasil serangan musuh mengenai *player* maka *health bar* *player* akan berkurang.

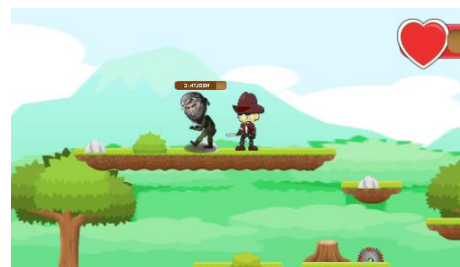
Gambar 4.3 karakter musuh melakukan patrol dan menyerang *player*

Pada gambar 4.4 menggambarkan interaksi antara karakter dengan musuh dimana musuh melihat keberadaan *player* yang sangat dekat posisinya dan melakukan *output* serangan pukul dekat, jika hasil

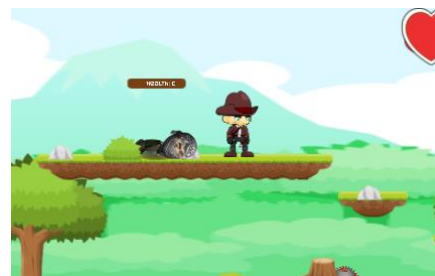
serangan musuh mengenai *player* maka *health bar* *player* akan berkurang.

Gambar 4.4 karakter musuh menyerang *player* dengan pukul dekat.

Pada gambar 4.5 menggambarkan ketika *player* menyerang musuh dengan pedang, jika serangan *player* mengenai musuh, maka *health bar* musuh akan berkurang.

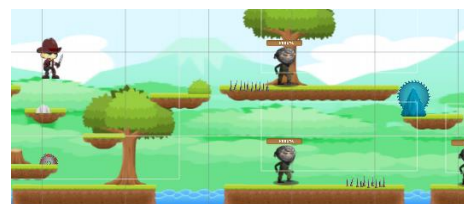
Gambar 4.5 karakter *player* menyerang musuh dengan pedang

Gambar 4.6 menggambarkan ketika karakter *player* menyerang musuh dan *health bar* musuh habis maka musuh akan mati, dalam jangka waktu 3 detik karakter musuh akan hilang.



Gambar 4.6 karakter musuh mati

Gambar 4.7 menggambarkan adanya beberapa macam jebakan yang jika *player* menyentuh jebakan tersebut maka *health point* *player* akan berkurang



Gambar 4.7 beberapa jebakan

4.3.2 Level 2

Pada gambar 4.8 menggambarkan interaksi antara karakter dengan musuh dimana musuh melihat keberadaan *player* dan melakukan output serangan jarak dekat atau jarak jauh, jika dari hasil serangan musuh mengenai *player* maka *health bar* *player* akan berkurang.



Gambar 4.8 karakter musuh melakukan patrol dan menyerang *player*

Pada gambar 4.9 menggambarkan interaksi antara karakter dengan musuh dimana musuh melihat keberadaan *player* yang dekat atau jauh posisinya dan melakukan output serangan pukul dekat atau menembak, jika hasil serangan musuh mengenai *player* maka *health bar* *player* akan berkurang.



Gambar 4.9 karakter musuh menyerang *player* dengan pukul dekat.



Gambar 4.10 karakter musuh menyerang *player* dengan menembak.

Pada gambar 4.11 menggambarkan ketika *player* menyerang musuh dengan pedang, jika serangan *player* mengenai musuh, maka *health bar* musuh akan berkurang.



Gambar 4.11 karakter *player* menyerang musuh dengan pedang

Gambar 4.12 menggambarkan ketika karakter *player* menyerang musuh dan *health bar* musuh habis maka musuh akan mati, dalam jangka waktu 3 detik karakter musuh akan hilang.



Gambar 4.12 karakter musuh mati

Gambar 4.13 menggambarkan adanya beberapa macam jebakan yang jika *player* menyentuh jebakan tersebut maka *health point* *player* akan berkurang



Gambar 4.13 beberapa jebakan

4.4 Pengujian AI (Artificial Intelligence)

Pengujian *artificial intelligence* adalah pengujian mengenai fungsi yang berkaitan dengan *artificial intelligence* yang ada pada *Game Rescue Idol*. Hasil pengujian dari *artificial intelligence* pada game ini seperti pada tabel 4.1

Tabel 4.1 pengujian *artificial intelligence*

NO	Kedaaan	Kejadian	Tindakan	Hasil
1	Player Tidak Terlihat	Jika player tidak masuk daerah musuh	Maka musuh akan tetap pada posisi diam atau patrol	Sesuai
	Player Terlihat	Jika player masuk pada daerah musuh	Maka musuh akan mengejar dan menyerang player	Sesuai
	Player Tidak Menyerang	Jika player tidak masuk daerah musuh dan tidak menyerang	Maka musuh akan tetap pada posisi diam atau patrol dan HP musuh tidak berkurang	Sesuai
	Player Menyerang	Jika player berada pada daerah musuh	Maka musuh akan menyerang dengan serangan dekat atau jauh	Sesuai
	HP Musuh Habis	Jika player menyerang musuh dan HP musuh habis	Maka musuh akan mati	Sesuai

Berdasarkan dari tabel 4.1,dapat diambil kesimpulan bahwa kecerdasan buatan yang diterapkan pada musuh sudah berhasil seluruhnya.

4.5 Pengujian Menu Utama

Pengujian *menu* utama adalah pengujian setiap fungsi dari tombol yang sudah diterapkan di *menu* utama. Hasil pengujian *menu* pada tabel 4.2

Tabel 4.2 pengujian *menu* utama

No	Tombol	Fungsi	Hasil
1	Play	Memulai permainan	Sesuai
2	Help	Menampilkan kontrol pada player	Sesuai
3	About	Menampilkan profil pembuat game	Sesuai

Berdasarkan pengujian tabel 4.2,dapat disimpulkan bahwa semua fungsi tombol pada *menu* utama sudah berhasil seluruhnya.

4.6 Pengujian Control Player

Pengujian *control player* adalah pengujian setiap fungsi dari tombol yang sudah diterapkan untuk menggerakkan karakter utama. Hasil pengujian *player* pada tabel 4.3

Tabel 4.3 pengujian *control player*

No	Tombol	Fungsi	Hasil
1	←	Menggerakkan ke kiri	Sesuai
2	→	Menggerakkan ke kanan	Sesuai
3	S	Player meloncat	Sesuai
4	D	Player melakukan aksi berupa serangan (<i>attack</i>)	Sesuai

Berdasarkan pengujian tabel 4.3,dapat disimpulkan bahwa semua fungsi tombol yang diterapkan untuk melakukan aksi pada *player* sudah sesuai.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah pembuatan *Game Rescue Idol*, maka penulis dapat mengambil kesimpulan :

1. Berdasarkan tabel 4.1 implementasi FSM (*Finite State Machine*) dapat diterapkan pada game "Rescue Idol" dengan indikasi musuh dapat mengejar dan menyerang *player* dengan kondisi tertentu.
2. Berdasarkan tabel 4.2 fungsi tombol pada *menu* utama bisa berjalan seperti mengklik *menu* "play" akan memulai permainan, mengklik *menu* "about" akan menuju tampilan pembuat, dan mengklik *menu* "help" akan menuju tampilan *control player*.
3. Berdasarkan tabel 4.3 fungsi tombol yang diterapkan di *player* dapat berjalan dengan cara menekan tombol kiri (←), kanan (→), S (meloncat), dan D (serang)..

5.2 Saran

Setelah dilakukan pengujian terhadap *Game Rescue Idol* maka masih ada kekurangan sehingga untuk pengembangan lebih lanjut disarankan :

1. Dapat dikembangkan sebagai mobile game berbasis *platform android* atau ios.
2. *Fuzzy logic* hanya dapat diimplementasikan pada karakter bos, selanjutnya *fuzzy logic* dapat diimplementasikan pada semua karakter game.

DAFTAR PUSTAKA

- [1] Ardi, R., 2012. Pembuatan Game First Person Shooter (FPS)"Operation Zygma Force" menggunakan FPS Creator X10".Yogyakarta : STMIK AMIKOM.
- [2] Dahria, M., 2008. Kecerdasan Buatan *Jurnal SAINTIKOM*, p.185
- [3] Hasanah, S.N dan Widiastuti, N.I., 2014. Representasi Emosi Menggunakan Fuzzy Pada Permainan Bonny's Tooth Booth.
- [4] Setiawan, I., 2016. Perancangan Software Embedded System Berbasis FSM Online. Online :<http://www.elektro.undip.ac.id/iwan/Perancangan>, p.20