

Perancangan dan Pembuatan *Action Game* dengan *Artificial Intelligence* dan *Machine Learning*

Samuel Vincentius Lidiyanto¹, Gregorius Satia Budhi², Rolly Intan³

Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121 – 131, Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-mail: samuel.lidiyanto@gmail.com¹, greg@petra.ac.id², rintan@petra.ac.id³

ABSTRAK

Bermain *game* merupakan salah satu alternatif untuk menghindari seseorang dari tekanan. Namun pada kenyataannya, banyak orang cepat merasa bosan dalam memainkan beberapa *game* yang telah tersedia. Untuk mengatasi hal tersebut, dalam skripsi ini akan dibuat sebuah *game* yang dimana *enemy* dapat memberikan respon yang cepat pada saat bertarung dan dapat mengembangkan kemampuan AI-nya.

Game yang dibuat adalah berjenis *action game*. AI yang akan dibuat dalam *game* ini menggunakan 2 (dua) metode, yaitu *fuzzy state machine* dan *machine learning*. *Fuzzy state machine* merupakan sebuah metode untuk memilih respon terbaik yang dapat diberikan oleh karakter *enemy*. *Machine learning* digunakan untuk melakukan *update* pada *file rules* yang aktif pada karakter *enemy*.

Hasil pengujian menunjukkan bahwa *action game* yang dibuat dengan *fuzzy state machine* dan *machine learning* dapat membuat karakter *enemy* memberikan respon lebih baik setelah dimainkan oleh pemain.

Kata Kunci: *Fuzzy State Machine, Fuzzy Rules, Game Based Learning, Action Game*

ABSTRACT

Playing games is a way for someone to relieve their stress. However, based on reality, most people are easily to get bored after playing several rounds of game. To overcome this problem, this thesis will create a game where the enemy can give fast respons when fighting the player and may develop the AI's capabilities.

The genre of this game is action. The AI in this game will use two methods: fuzzy state machine and machine learning. Fuzzy state machine is a method to choose the best response towards the enemy. Machine learning is used to update the active file rules of the enemy.

The result from this experiment stated that action game with fuzzy state machine and machine learning can make the enemy to give better respons after being played by the player.

Keywords: *Fuzzy State Machine, Fuzzy Rules, Game Based Learning, Action Game*

1. PENDAHULUAN

Personal Computer Games (PC Games) merupakan salah satu hiburan yang umum dimainkan oleh semua orang. Salah satu jenis permainan yang digemari dan sedang berkembang saat ini adalah permainan berjenis *Action*. Dalam permainan berjenis *Action*, *enemy* yang dihadapi oleh pemain digerakkan secara otomatis dengan menggunakan metode *Rule-Based*. *Rule-Based System* (Sistem Berbasis Aturan) merupakan sebuah sistem yang memproses sebuah informasi dengan pengetahuan yang tersimpan di dalam memori untuk menghasilkan informasi yang baru dalam menyelesaikan permasalahan yang ada. Salah satu sifat dari *Rule-Based System* adalah sistem akan selalu mengelolah dan menghasilkan informasi hanya berdasarkan pada pengetahuan atau *rules* yang terdapat dalam sistem. Apabila sistem menerima *input* informasi yang sama, maka informasi yang dihasilkanpun akan sama. Hal ini menyebabkan timbulnya rasa bosan karena strategi yang sama dapat digunakan berulang kali untuk mengalahkan *enemy*.

Dengan adanya permasalahan di atas, dapat disimpulkan bahwa perlu adanya pengembangan sistem dalam permainan berjenis *Action Game* agar pemain tidak cepat merasa bosan ketika bermain *game*. *Fuzzy Rule-Based System* ditambahkan dengan *machine learning* merupakan solusi tepat yang dapat digunakan untuk menjaga ketertarikan seseorang dalam bermain *game* berjenis *Action*. Dalam implementasinya, *machine learning* digunakan untuk mempelajari strategi pemain dan melakukan *update rule* pada sistem apabila *enemy* kalah pada saat bermain melawan pemain sehingga *enemy* dapat memberikan respon yang lebih baik.

2. LANDASAN TEORI

2.1 Teori Game

Teori *game* adalah teori yang membahas komponen-komponen yang diperlukan untuk membuat suatu permainan dengan jenis permainan yang telah ditentukan sebelumnya. Teori *game* ini membagi *game* tergantung dari jumlah *player* yang memainkan, *objective* yang ingin dicapai oleh *game*, dan informasi yang dimiliki oleh setiap *player* dalam *game* tersebut [6].

2.2 Gamification Design

Gamification merupakan konsep yang sering digunakan oleh programmer dalam membuat sebuah permainan. Menurut Deterding, S., Dixon, D., Khaled, R., dan Nacke, L.E., *gamification* adalah sebuah proses dengan menggunakan *game thinking* dan *game mechanic* untuk menyelesaikan permasalahan-permasalahan dan berinteraksi dengan *user* [2].

Ada beberapa elemen yang berpengaruh dalam desain sebuah permainan, diurutkan dari yang konkret hingga abstrak terdapat 5 (lima) *levels* yang berpengaruh:

1. *Interface design pattern*
2. *Game design pattern* atau *Game mechanic*
3. *Design principles*
4. *Conceptual model of game design unit*
5. *Game design method* dan *design process*

Tabel 1. Pembagian level Gamification

Level	Deskripsi	Contoh
<i>Game interface design pattern</i>	Umum, dari masalah yang ada pada game, dapat didapatkan interaksi yang tepat antara komponen desain dengan desain solusi	<i>Badge, Leaderboard, level</i>
<i>Game design and mechanics</i>	Umumnya digunakan untuk desain game yang mementingkan <i>gameplay</i>	Permainan yang menggunakan giliran
<i>Game design principles and heuristics</i>	Mengevaluasi <i>guideline</i> untuk menganalisa masalah pada desain atau solusi	Menahan permainan, tujuan yang jelas, <i>game style</i> yang bervariasi
<i>Game Models</i>	Konsep model dari komponen <i>game</i> atau pengalaman <i>game</i>	MDA, tantangan, fantasi, desain <i>game atoms, CEGE</i>
<i>Game design methods</i>	Desain <i>game</i> dengan latihan dan proses yang spesifik	<i>Playtesting</i> , desain <i>playcentric</i> , menghargai nilai dari desain <i>game</i>

Pembagian *level* pada sebuah *game* dapat dilihat pada Tabel 1. sebuah *game* yang akan dibuat dapat diposisikan hingga game elemen yang akan dibuat dapat disesuaikan sehingga *game* yang dibuat dapat memberikan *output* yang diharapkan secara maksimal. [8]

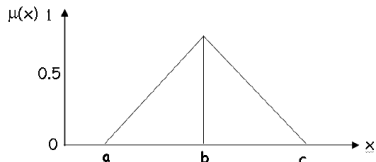
2.3 Game Based Learning

Menurut Epper, R.M., Derryberry, A., dan Jackson, S., menyatakan bahwa *game* dapat dijadikan sebuah media baru untuk berinteraksi dengan siswa. Pembelajaran pada siswa dengan pendekatan menggunakan *game* ini disebut *game based learning*. Dalam implementasinya, terdapat beberapa faktor yang mempengaruhi pembuatan *game based learning* seperti ekspektasi dari murid dimana murid pada zaman sekarang telah tumbuh dari kecil dengan teknologi sehingga *game* yang dibuat harus dapat menarik perhatian mereka; integrasi antara *game* dan simulasi, dimana dengan data yang ada. [3]

2.4. Fuzzy Set

Berikut akan dijelaskan mengenai 2 (dua) macam tipe fungsi yang dimiliki oleh *fuzzy set* [1].

1. Triangular Function



Gambar 1. Grafik Triangular Fuzzy Set

Gambar 1 merupakan grafik *triangular fuzzy set* yang memiliki 3 titik koordinat yang menunjukkan daerah hasil dari fungsi tersebut. Titik a menunjukkan posisi awal, titik b menunjukkan

posisi puncak, dan titik c menunjukkan posisi akhir dari fungsi tersebut. Apabila titik-titik tersebut dihubungkan akan membentuk sebuah daerah hasil fungsi berbentuk segitiga.

$$\mu(x) = 0, \text{ untuk } x < a$$

$$\mu(x) = \frac{x-a}{b-a}, \text{ untuk } a \leq x \leq b$$

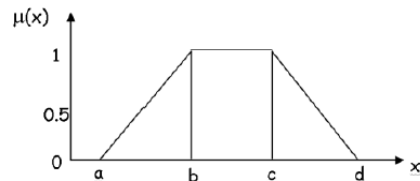
$$\mu(x) = \frac{c-x}{c-b}, \text{ untuk } b < x \leq c$$

$$\mu(x) = 0 \text{ untuk } x > c$$

Gambar 2. Persamaan fuzzy value triangular fuzzy set

Gambar 2 merupakan persamaan yang digunakan untuk mencari *value* dari *fuzzy set* yang menjadi *input* dari fungsi grafik tersebut.

2. Trapezoid Function



Gambar 3. Grafik trapezoid fuzzy set

Gambar 3 merupakan grafik *trapezoid fuzzy set* yang memiliki 4 titik koordinat yang menunjukkan daerah hasil dari fungsi tersebut. Titik a menunjukkan posisi awal, titik b dan c menunjukkan posisi puncak, dan titik d menunjukkan posisi akhir dari fungsi tersebut. Apabila titik-titik tersebut dihubungkan akan membentuk sebuah daerah hasil fungsi berbentuk trapesium.

$$\mu(x) = 0, \text{ untuk } x < a$$

$$\mu(x) = \frac{x-a}{b-a}, \text{ untuk } a \leq x \leq b$$

$$\mu(x) = 1 \text{ untuk } b < x \leq c$$

$$\mu(x) = \frac{d-x}{d-c}, \text{ untuk } c < x \leq d$$

$$\mu(x) = 0 \text{ untuk } x > d$$

Gambar 4. Persamaan fuzzy value trapezoid fuzzy set

Gambar 4 merupakan persamaan yang digunakan untuk mencari *value* dari *fuzzy set* yang menjadi *input* dari fungsi grafik tersebut. Apabila terdapat 2 (dua) macam *Fuzzy Set* dan kedua *Fuzzy Set* tersebut digabungkan, maka akan membentuk sebuah *Fuzzy Set* yang baru.

2.5 Fuzzy Logic

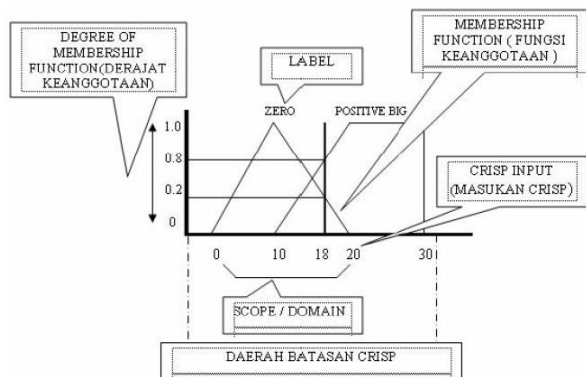
Fuzzy Model Sugeno merupakan varian dari model Mamdani dan memiliki bentuk aturan seperti :

$$\text{IF } x1 \text{ is } A1 \text{ AND.. } xn \text{ is } An \text{ THEN } y = f(x1, x2.. xn) (I)$$

Dimana *x* merupakan parameter *input*, *A* merupakan nilai dari parameter, *f* merupakan sembarang fungsi dari variabel masukan yang nilainya berada dalam *interval* variabel keluaran [4].

Konsep dasar logika *fuzzy* secara ringkas dapat dilihat pada Gambar 5. *Crisp Input* merupakan *input* berupa suatu nilai. *Scope* atau *Domain* adalah jangkauan dari fungsi keanggotaan. Sedangkan Daerah Batasan *Crisp* merupakan jangkauan seluruh nilai yang dapat diaplikasikan pada sistem. Label merupakan

penamaan dari grafik yang ada. *Membership Function* merupakan suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya. *Degree of Membership Function* merupakan nilai keanggotaan suatu fungsi yang memiliki nilai antara 0 hingga 1.



Gambar 5. Konsep Dasar Logika Fuzzy

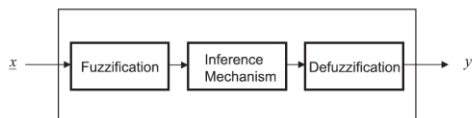
2.6 Membership Functions

Membership functions adalah sebuah kurva yang mendefinisikan seberapa besar nilai yang dihasilkan berdasarkan fungsi yang ada apabila diberi sebuah nilai *input*.

Apabila terdapat sebuah subset dari *fuzzy set* referensial X dengan fungsi keanggotaan (*membership functions*) $A : X \rightarrow L$, untuk setiap $x \in X$, $A(x)$ akan dihasilkan sebuah nilai yang disebut *degree of membership* dari x di dalam A . L dalam fungsi tersebut merupakan *unit interval* yang memiliki nilai antara 0 (nol) hingga 1 (satu) [4].

2.7 Fuzzy System

Sebuah *Fuzzy System* menghubungkan *input membership functions*, di mana *functions* tersebut mewakili *input* terhadap *controller*, untuk menghasilkan *output membership functions* yang mewakili *action* yang akan dilakukan [10].



Gambar 6. Struktur dari fuzzy system

Gambar 6 merupakan struktur dari sebuah *fuzzy system*. *Inputs x* dan *output y* adalah *crisp* (angka sebenarnya, tidak dalam bentuk *Fuzzy Set*). Pada bagian *Fuzzification*, *crisp input* yang telah diterima diubah menjadi *fuzzy sets*. *Inference Mechanism* menggunakan *rules* yang terdapat pada *rule base* untuk mengolah *input* yang telah diubah menjadi *fuzzy sets* untuk diubah kembali menjadi *output* dalam bentuk *fuzzy sets* sesuai dengan *rule* yang terpenuhi. Setelah itu, *Defuzzification* mengubah kembali hasil *output* menjadi sebuah *crisp* [5].

2.8 Fuzzy Rule-Based System

Fuzzy Rule-Based System merupakan salah satu dari bermacam-macam cara yang ada untuk mengimplementasikan pengetahuan manusia ke dalam sebuah sistem kecerdasan buatan. *Fuzzy Model Sugeno* merupakan varian dari model Mamdani dan memiliki bentuk aturan seperti : *IF x1 is A1 AND.. xn is An*

THEN y = f(x1, x2.. xn) dimana x merupakan parameter input, A merupakan nilai dari parameter, f merupakan sembarang fungsi dari variabel-variabel masukan yang nilainya berada dalam interval variabel keluaran [9].

2.9 Fuzzy State Machine

Fuzzy State Machine adalah AI (*Artificial Intelligence*) yang berdasarkan pada *Fuzzy logic* untuk mengambil keputusan. Cara kerja AI ini menyerupai cara kerja *Fuzzy logic*. AI ini biasa digunakan oleh *object* untuk menentukan senjata apa yang digunakan atau memilih melakukan apa sesuai dari grafik *fuzzy* yang dipakai oleh *object*. Cara AI ini melakukan penerjemahan grafik menjadi sebuah perintah adalah mengubah *crisp* yang telah didisain menjadi *fuzzy rules* dan diterjemahkan lagi menjadi perintah yang dapat dibaca oleh AI [7]. Berikut merupakan gambaran *state* yang digunakan dalam *fuzzy state machine*.

3. DESAIN SISTEM

3.1 Interface untuk memulai Game

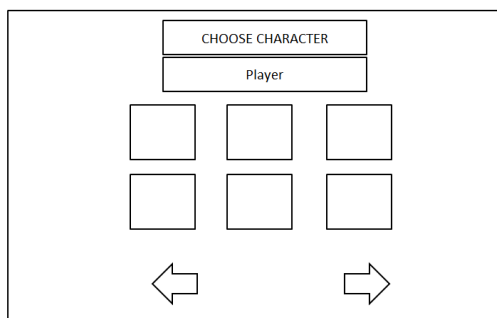
Pada awal permainan saat *game* dijalankan, *player* akan diberikan 2 menu, yaitu *Start Game* dan *Exit*. *Start Game* akan membawa *player* untuk memainkan permainan, dimulai dengan memilih karakter yang akan digunakan oleh *player*, setelah itu memilih karakter yang akan digunakan oleh *enemy* sebagai lawan dari *player*. Setelah memilih karakter, maka *player* akan memulai permainan. Sedangkan menu *Exit* akan membawa *player* untuk keluar dari permainan.

3.2 Interface dalam Game

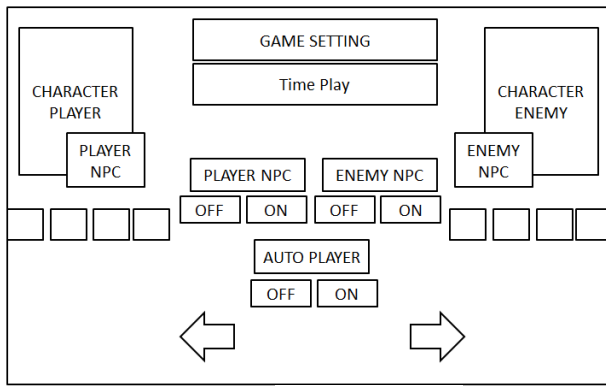


Gambar 7. Halaman Awal Permainan

Gambar 7 merupakan tampilan awal dari aplikasi ini. Pada halaman ini terdapat 2 (dua) opsi yang dapat dipilih oleh *player*, yaitu *start game* dan *exit*. *Start game* adalah opsi untuk memulai permainan, sedangkan *exit* merupakan opsi untuk menutup aplikasi.



Gambar 8. Pemilihan Karakter Player dan Enemy

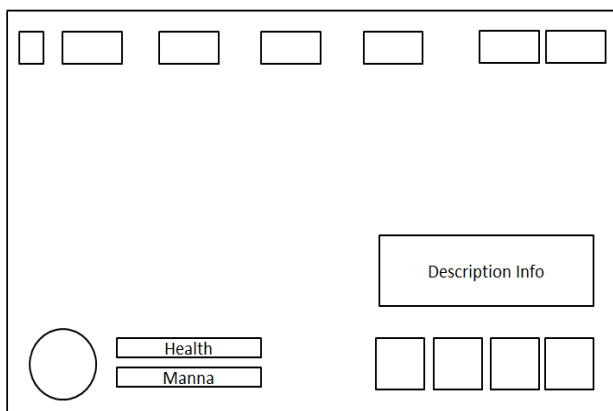


Gambar 9. Setting permainan

Gambar 8 merupakan halaman pemilihan karakter *player* dan *enemy*. Setelah *player* memilih *Start Game*, maka *player* akan dihadapkan pilihan karakter yang dapat digunakan dalam permainan. Selesai memilih karakter, maka *player*, maka pemain akan memilih untuk karakter *enemy*. Setelah itu pemain dapat mengatur *setting arena* pada permainan yang akan digunakan. Tampilan halaman *setting permainan* dapat dilihat pada Gambar 9. Hal yang dapat diatur adalah lama permainan dan *support character* untuk karakter *player* dan *enemy*.

Tampilan permainan *player* dapat dilihat pada Gambar 10. Pada saat permainan dimulai, maka pada layar permainan, *player* akan dapat melihat *status screen* yang menunjukkan keterangan mengenai kondisi karakter yang sedang digunakan, seperti *health*, *mana*, *type*, *attack damage*, *shoot damage*. Selain itu, pada sebelah kanan, *player* dapat melihat beberapa menu *action* yang dapat dilakukan oleh karakter tersebut. Berikut akan dijelaskan mengenai beberapa *action* yang dapat dilakukan oleh karakter dalam permainan.

- *Melee-attack* merupakan serangan yang dilakukan karakter dari jarak dekat.
- *Long-range-attack* merupakan serangan yang dilakukan karakter dari jarak jauh.
- *Move* merupakan gerakan untuk berpindah ke area yang lain.
- *Magic* merupakan serangan sihir yang dapat dilakukan oleh karakter. Apabila menggunakan *magic* maka akan mengurangi *mana* yang dimiliki karakter sesuai dengan poin *magic* yang dibutuhkan dan karakter tidak dapat menggunakan *magic* jika *mana* yang dimiliki tidak mencukupi.



Gambar 10. Tampilan permainan *player*

Action yang dilakukan oleh karakter *player*, dapat dilakukan melalui *input* melalui *keyboard* dan *mouse*.

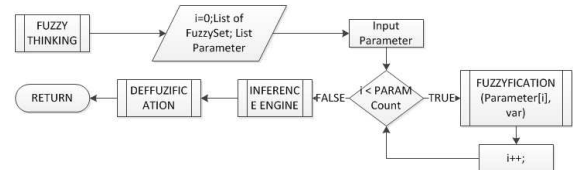
- *Move* dapat dilakukan dengan menekan tombol kanan pada *mouse*.
- *Melee-attack* dengan menekan tombol kiri pada *mouse*.
- *Long-range attack (shoot)* dapat dilakukan dengan menggunakan tombol 'Q'
- *Casting magic* dapat dilakukan dengan tombol 'E', 'R', dan 'F'.

3.3 Desain Sistem *Enemy*

Selama permainan berlangsung, *enemy* akan menghasilkan *output* gerakan berdasarkan dengan parameter *input fuzzy* yang telah di-*set* dalam program. Fungsi yang melakukan perhitungan agar *enemy* dapat menghasilkan sebuah *output* dinamakan *ProsesBehavior()*. Selain itu, sistem juga akan melakukan pengecekan terhadap serangan yang dilakukan oleh pemain. Apabila serangan yang dilakukan oleh pemain membuat *health* dari karakter *enemy* berkurang, maka *rule* yang aktif pada saat *enemy* itu beserta dengan besar nilai serangan pemain akan disimpan oleh sistem. Pada saat permainan berakhir, pada saat *enemy* menang dalam permainan, maka tidak akan dilakukan analisa dan *update* pada *rule* yang digunakan. Apabila *enemy* kalah dalam permainan, maka akan dilakukan analisa terhadap *rule* yang memiliki nilai dampak yang terbesar bagi *health enemy*. Setelah proses analisa oleh sistem selesai, maka *rule* tersebut akan di-*update* dengan *output* yang lebih baik.

3.4 Proses *Fuzzy State Machine*

Metode yang digunakan dalam menjalankan karakter oleh *enemy* yaitu *Fuzzy State Machine*. AI digunakan sepanjang permainan berlangsung. AI bertugas untuk menentukan keputusan yang akan diambil oleh karakter *enemy*. Keputusan yang akan diambil berdasarkan *input* yang telah ditentukan sebelumnya, *input* berupa *health player*, *health enemy*, *mana*, dan jarak antara karakter pemain dengan *enemy*. Untuk mengambil keputusan, AI cukup menggunakan *Fuzzy State Machine* dengan inputan sesuai dengan yang ditentukan. Perhitungan pada AI *Fuzzy State Machine* menggunakan *integer* sehingga perhitungan lebih cepat.



Gambar 11. *Fuzzy State Machine*

Untuk dapat menjalankan karakter *enemy*, diperlukan beberapa *input* berupa *Fuzzy Set* dan *Rules* dalam bentuk *Fuzzy Rules*. *Fuzzy Rules* tiap karakter dibuat dengan menggunakan aplikasi *Fuzzy Logic Control*. Dengan menggunakan aplikasi tersebut, dapat dibuat *variable* beserta dengan *rules* yang akan digunakan oleh *Fuzzy State Machine* dalam mengambil keputusan mengenai tindakan yang dilakukan. Pada Gambar 11 merupakan gambaran mengenai proses yang dijalankan oleh *Fuzzy State Machine*. Dalam menggunakan *Fuzzy Rules* maka proses yang dilalui akan dibagi menjadi 3 bagian, yaitu *Fuzzification*, *Inference Engine*, dan *Defuzzification*. Setelah melalui 3 bagian

proses ini, akan dihasilkan sebuah *output* berupa *crisp value* yang dapat digunakan untuk menentukan keputusan yang diambil oleh AI.

3.5 Desain Machine Learning

Proses perhitungan pada tiap *output rule* yang telah tersimpan dalam variabel *triggeredRules*. Perhitungan dilakukan berdasarkan dengan nilai yang terdapat pada setiap *rule* yang ter-*trigger*. Setelah itu sistem akan melakukan *update* pada *rules* yang ada dalam *database*.

4. PENGUJIAN SISTEM

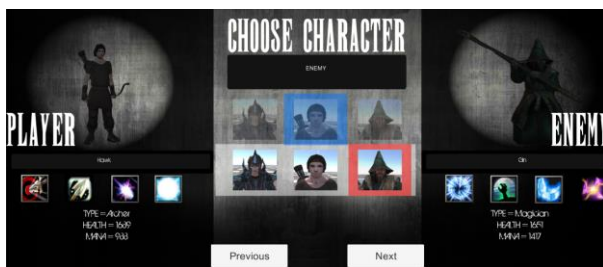
4.1 Pengujian Aplikasi

Pengujian ini dilakukan untuk mengetahui proses permainan secara keseluruhan ketika seseorang bermain aplikasi ini. Pengujian ini dilakukan dengan memainkan program aplikasi yang telah dibuat. Pada saat seseorang akan memainkan permainan ini, maka pemain tersebut akan masuk ke dalam halaman awal. Tampilan halaman awal dapat dilihat pada Gambar 12.



Gambar 12. Halaman Awal Permainan

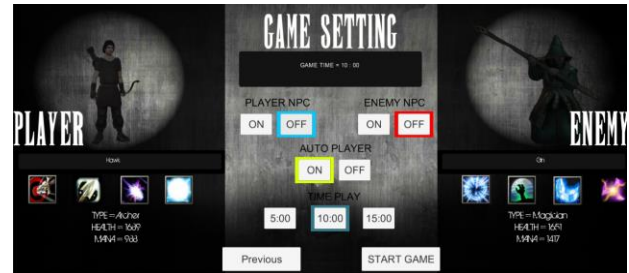
Pada Gambar 12 pemain dihadapkan 2 (dua) opsi pilihan, yaitu opsi *start game* dan opsi *exit*. *Start game* adalah opsi untuk memulai permainan dan *exit* adalah opsi untuk keluar dari permainan. Pemain memilih opsi *start game* untuk memulai permainan. Selanjutnya pemain akan masuk ke dalam halaman untuk memilih karakter. Tampilan dalam memilih karakter dapat dilihat pada Gambar 13



Gambar 13. Halaman Pemilihan Karakter

Gambar 13 merupakan halaman pemilihan karakter. Pada kolom sebelah kiri merupakan kolom karakter *player* sedangkan kolom sebelah kanan merupakan kolom karakter *enemy*. Pada saat pemilihan karakter, pemain memiliki 3 (tiga) karakter yang dapat digunakan dalam permainan. Pada awalnya pemain akan memilih karakter untuk *player* dan setelah selesai memilih, maka pemain dapat memilih opsi *next* untuk memilih karakter *enemy*. Apabila pemain memilih opsi *previous*, maka pemain akan kembali ke halaman awal permainan. Dalam permainan ini pemain memilih karakter *archer* sebagai karakter *player*, dan *magician* sebagai karakter *enemy*. Ketika pemain melakukan pemilihan karakter, pemain dapat melihat informasi mengenai

kemampuan karakter yang dipilihnya di kolom kiri dan kanan pada *user interface* yang ada. Pada Gambar 13 dapat dilihat informasi karakter *player* yang digunakan, yaitu *type archer*, *health* 1689, *mana* 988, beserta dengan 4 *magic* yang dapat digunakan karakter *archer*. Pada kolom kanan juga dapat dilihat informasi mengenai karakter *enemy*, yaitu *type magician*, *health* 1651, *mana* 1417, beserta dengan 4 *magic* yang dapat digunakan oleh karakter *magician*. Selanjutnya pemain dapat memilih opsi *next* untuk melanjutkan ke halaman *setting* permainan. Halaman *setting* permainan dapat dilihat pada Gambar 14.



Gambar 14. Halaman Pemilihan Karakter

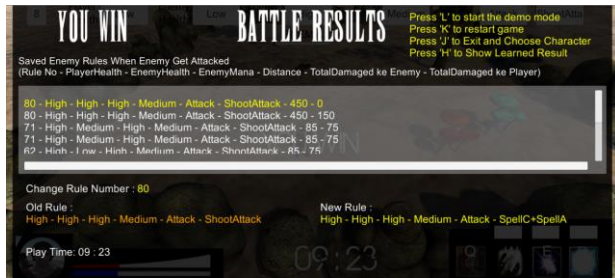
Gambar 14 merupakan halaman untuk melakukan *setting* (pengaturan) pada permainan. Pada halaman ini pemain dapat melakukan *setting* untuk *support character* karakter *player* dan karakter *enemy*, lama waktu permainan, dan *auto player* pada karakter *player*. Dalam permainan ini pemain melakukan *setting* permainan seperti yang dilihat pada Gambar 14. *Setting* yang dilakukan pemain adalah *support character* untuk kedua karakter adalah *off*, lama permainan adalah 10 (sepuluh) menit, dan *auto player* untuk karakter *player* di-*set on*. Selanjutnya pemain dapat menekan tombol *start game* untuk memulai permainan. Tampilan halaman permainan dapat dilihat pada Gambar 15.



Gambar 15. Halaman Pertarungan Awal

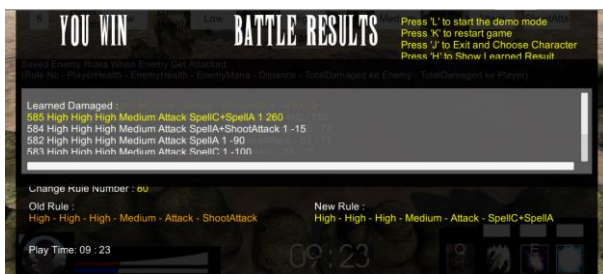
Gambar 15 merupakan halaman pertarungan antara karakter *player* dan karakter *enemy*. Dalam permainan, pemain dapat melihat *status* karakternya pada *user interface* yang disediakan. Pada bagian bawah *user interface* merupakan bagian status yang dimiliki oleh karakter *player*. Pada bagian bawah kiri terdapat *image* yang digunakan oleh karakter *player*, *health bar*, dan juga *mana bar*. Pada bagian tengah terdapat lama permainan berlangsung. Pada bagian kanan bawah terdapat *icon* untuk *magic* yang dapat digunakan. Bagian atas pada *user interface* merupakan keterangan mengenai *rule* yang sedang ter-*trigger* oleh sistem, diantaranya terdapat nomer *rule*, kondisi *player health*, kondisi *enemy health*, kondisi *enemy mana*, dan *distance* antara 2 (dua) karakter, *output* dan juga *action* yang dilakukan. Pada bagian kiri terdapat sebuah *text area* yang menampilkan mengenai dampak serangan dari karakter *player* yang mengenai karakter *enemy*. Keterangan yang disimpan adalah nomer *rule*, kondisi *player health*, kondisi *enemy health*,

kondisi *enemy mana*, *distance*, dampak serangan karakter *player* ke karakter *enemy*, dan dampak serangan karakter *enemy* ke karakter *player*. Pada sebelah kanan terdapat *text area* yang menampilkan proses penghitungan sebelum *rule* yang terkena dampak serangan tersebut disimpan. 'P' untuk serangan karakter *enemy* ke karakter *player*, dan 'E' untuk serangan karakter *player* ke karakter *enemy*. Karena mode *auto* yang dipilih oleh pemain, maka pemain dapat menunggu hingga kedua karakter selesai bertarung hingga salah satunya memiliki *health* 0 (nol). Setelah selesai, maka pemain akan masuk ke halaman hasil pertarungan. Tampilan halaman hasil pertarungan dapat dilihat pada Gambar 16.



Gambar 16. Halaman Hasil Pertarungan Awal

Gambar 16 merupakan halaman hasil pertarungan yang telah selesai. Pada halaman ini, pemain dapat melihat hasil dari pertarungan yang telah dimainkannya. Pada bagian atas kiri terdapat keterangan *player* menang (*You Win*) ataupun kalah (*You Lose*). Pada bagian tengah merupakan *rule-rule* yang tersimpan oleh sistem sebagai dampak serangan karakter *player* yang mengenai karakter *enemy*. Pada tulisan yang berwarna kuning merupakan *rule* yang dinilai oleh sistem sebagai *rule* yang kurang baik, didapatkan dari selisih terbesar antara dampak serangan ke *enemy* dikurangkan dengan dampak serangan ke *player*. Pada bagian kiri bawah terdapat keterangan mengenai *rule* yang dianggap kurang baik, beserta dengan sisa waktu pada pertarungan sebelumnya, sedangkan pada sebelah kanan merupakan *rule* baru yang menggantikan *rule* yang dianggap kurang baik oleh sistem. Pada halaman ini pemain juga dapat melihat hasil akhir dari penghitungan *rule* baru yang dipilih untuk menggantikan *rule* yang lama dengan menekan tombol 'H'. Tampilan halaman penghitungan *rule* dapat dilihat pada Gambar 17.



Gambar 17. Halaman Perhitungan Learning Damaged

Gambar 17 merupakan halaman yang menampilkan hasil perhitungan sistem yang menyimpulkan *rule* yang terbaik yang dapat menggantikan *rule* yang dianggap kurang baik. Pada bagian tengah pemain dapat melihat hasil perhitungan beberapa *rule* yang diajukan oleh sistem untuk menggantikan *rule* yang kurang baik. *Rule* yang berwarna kuning merupakan *rule* terpilih yang dinilai *rule* yang lebih baik oleh sistem. Selain itu, pemain juga diberikan beberapa pilihan halaman untuk berpindah.

Pilihan halaman dapat dilihat pada bagian kanan atas pada *user interface*. Untuk menguji kembali *rule* yang telah diubah, maka pemain menekan tombol 'L' pada *keyboard* untuk masuk ke halaman *demo*. Halaman *demo* dapat dilihat pada Gambar 18



Gambar 18. Halaman Pertarungan Mode Demo

Gambar 18 merupakan halaman *demo* untuk melakukan pengecekan *rule* pada karakter *enemy* telah mengalami perubahan atau tidak, dengan langkah-langkah yang diambil oleh karakter *player* adalah langkah-langkah yang digunakan dalam permainan sebelumnya. Pada halaman ini, maka pemain dapat melihat pergerakan karakter *player* dan karakter *enemy* sesuai dengan permainan sebelumnya, hingga sistem menemukan kondisi *input rule* sama dengan kondisi *rule* yang telah diganti. Pada Gambar 18 diperlihatkan pada saat kondisi untuk *rule* yang tergantikan oleh sistem muncul dalam permainan. Sistem akan melakukan *pause* pada permainan sehingga pemain dapat melakukan pengecekan *rule* yang dalam sistem sudah diubah atau belum. Hal tersebut dapat dilihat pada bagian tengah yang merupakan *rule* lama sebelum sistem merubah *rule* dari karakter *enemy* dan pada bagian atas merupakan *rule* baru yang telah diubah oleh sistem. Untuk melanjutkan permainan, pemain dapat menekan tombol 'P'. Setelah ini, maka karakter *enemy* akan kembali digerakkan dengan menggunakan perhitungan *fuzzy*, tetapi karakter *player* akan tetap digerakkan dengan *action* pada permainan sebelumnya. Pemain dapat mengecek apakah *rule* yang baru lebih efektif atau tidak melalui respon yang diberikan oleh sistem. Setelah mode *demo* selesai, maka pemain dapat melihat hasilnya pada Gambar 19.



Gambar 19. Halaman Hasil Pertarungan Mode Demo

Pada Gambar 19 pemain dapat membandingkan dampak serangan yang diterima oleh karakter *enemy*. Pada bagian kiri atas merupakan hasil pertandingan pada saat mode *demo*. Pada bagian tengah merupakan dampak serangan karakter *player* yang mengenai karakter *enemy* dengan *rule* yang berwarna kuning merupakan *rule* yang memiliki kondisi yang sama dengan *rule* yang diubah oleh sistem. Pada bagian kanan bawah merupakan *rule* baru yang diubah oleh sistem sebelumnya beserta dengan besar dampak serangan karakter *player* yang mengenai karakter *enemy*.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian pada bab sebelumnya, dapat diambil kesimpulan sebagai berikut:

- Dengan desain permainan yang dibuat, pemain tidak akan mudah bosan
- Pembuatan *game* dengan aplikasi *unity* dapat menghasilkan grafik permainan dengan baik
- Karakter *enemy* dalam *game* dapat digerakkan dengan menggunakan metode *fuzzy rule base system* dengan menambahkan *library fuzzyClass.dll*.
- Penggunaan *fuzzy* dalam permainan membuat karakter *enemy* dapat memberikan respon dengan cepat
- Dengan menambahkan *machine learning* dengan membandingkan tindakan *enemy* sebelum dan setelah *rule* diubah oleh aksi *player*, dapat dilihat bahwa *enemy* dapat memberikan respon yang lebih baik.

5.2 Saran

Beberapa saran untuk pengembangan *game* kedepannya adalah:

- *Game* ini dapat dikembangkan untuk beberapa *platform* seperti *mobile devices* dengan OS Android atau iOS sehingga *game* ini dapat dimainkan secara luas.
- *Machine Learning* dalam *game* ini dapat dikembangkan dengan melakukan *update* pada *membership function* pada tiap variabel dalam *game*.

6. DAFTAR REFERENSI

- [1] Bede, B. 2013. *Mathematics of Fuzzy Sets and Fuzzy Logic*. Heidelberg: Springer.
- [2] Deterding, S, Dixon, D, Khaled, R, dan Nacke, L.E. 2011. Gamification: Toward to Definition. *CHI 2011*, pp. 1-4.
- [3] Epper, R.M., Derryberry, A., dan Jackson, S. 2012. *Game-Based Learning: Developing an Institutional Strategy*. Retrieved August 9, 2012, Louisville, CO: EDUCAUSE Center for Applied Research Web Site: <http://www.educause.edu/ecar>.
- [4] Hullermeier, E. 2011. *Fuzzy Logic in Machine Learning*.
- [5] Lilly, J. H. 2010. *Fuzzy Control and Identification*. Hoboken: John Wiley & Sons, Inc.
- [6] Millington, I. 2006. *Artificial Intelligence for Games*. San Fransico: Elsevier.
- [7] Pirovano, M. 2012. The use of Fuzzy Logic for Artificial Intelligence in Games. pp. 1-8.
- [8] Porter, N. 2012. *Component-Based Game Object System*.
- [9] Purba, K. R., Hasanah, R. N., & Muslim, M. A. 2013. Implementasi Logika Fuzzy Untuk Mengatur Perilaku Musuh dalam Game Bertipe Action-RPG. *Jurnal EECCIS*, Vol. 7 (No. 1), pp. 15-20.
- [10] Ross, T. J. 2004. *Fuzzy Logic with Engineering Applications* (2th ed.). Chichester: John Wiley & Sons Ltd.