

# Penerapan *Adaptive Neuro Fuzzy Inference System* pada *Real Time Strategy Battle Arena Game*

Andy Gunawan<sup>1</sup>, Rolly Intan<sup>2</sup>, Kristo Radion Purba<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra  
Jl. Siwalankerto 121-131, Surabaya 60236

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

E-mail: kwee12lucky@gmail.com<sup>1</sup>, rintan@petra.ac.id<sup>2</sup>, kristo@petra.ac.id<sup>3</sup>

## ABSTRAK

Salah satu *genre game* yang paling banyak diminati saat ini adalah *Real Time Strategy Battle Arena*. Namun, kebanyakan dari *game* dengan *genre* tersebut hanya menarik jika dimainkan dengan manusia lain sebagai lawan sehingga *game* menjadi kurang fleksible karena tidak dapat dimainkan tanpa adanya orang lain atau koneksi internet.

Karena itu, dalam skripsi ini akan dibuat sebuah *game* dengan *genre Real Time Strategy Battle Arena* dengan AI sebagai musuh dimana pergerakan musuh dapat disesuaikan dengan cara berpikir manusia. Musuh pada *game* ini juga dapat belajar dari setiap kesalahan langkah. Metode yang digunakan adalah *Adaptive Neuro Fuzzy Inference System*.

Hasil pengujian menunjukkan bahwa musuh dapat mengambil langkah pergerakan yang terbaik, yang sesuai dengan cara berpikir manusia dengan menggunakan *Fuzzy Inference System*. *Neural System* juga dapat mengubah *Membership Functions* dari beberapa *variable input Fuzzy Inference System* setiap kali musuh dianggap melakukan kesalahan langkah untuk meminimalisir kesalahan serupa saat *game* dimainkan kembali.

**Kata Kunci:** Adaptive Neuro Fuzzy Inference System, Real Time Strategy Battle Arena Game, C#, Unity

## ABSTRACT

Nowadays, *Real Time Strategy Battle Arena* is one of the most popular *game genre*. However, most *games* with that *genre* are only interesting if played together with another human. That makes those *games* less flexible since they can't be played without another human or internet connection.

Because of that, this thesis will create a *game* with *Real Time Strategy Battle Arena genre* with AI as the enemy where the enemy movements can be adjusted to approach human thinking ways. The enemy in this *game* will also be able to learn from each mistake. This thesis will use the method *Adaptive Neuro Fuzzy Inference System*.

Test results show that the enemy was able to choose the best move possible, which suits the way humans think using *Fuzzy Inference System*. *Neural System* also modified the *Membership Functions* of some *Fuzzy Inference System* input variable everytime the enemy is considered to make mistake to minimize similar mistakes when the *game* is played again in the future.

**Keywords:** Adaptive Neuro Fuzzy Inference System, Real Time Strategy Battle Arena Game, C#, Unity

## 1. PENDAHULUAN

Bermain *game* sudah menjadi hal yang umum dilakukan pada masa ini, baik untuk sekedar mengusir kebosanan, maupun untuk mengasah otak dan membantu proses pembelajaran manusia. *Real Time Strategy* dan *Battle Arena* merupakan dua dari sekian banyak *genre game* yang paling banyak diminati sekarang ini. Namun, *game-game* tersebut biasanya baru menarik jika yang menjadi lawan para pemain adalah pemain-pemain lain dari berbagai tempat di dunia karena strategi yang digunakan oleh pemain-pemain tersebut akan berbeda-beda dan tidak mudah ditebak. Hal itu membuat *game-game* tersebut menjadi kurang fleksible karena hanya dapat dimainkan jika ada koneksi internet yang memadai. Karena itu, dibutuhkan *game-game* ber-*genre Real Time Strategy Battle Arena* dengan AI sebagai lawan dimana AI dapat berpikir seperti manusia atau bahkan lebih baik dari manusia, sehingga *game-game* tersebut dapat dimainkan tanpa koneksi internet.

Karena itu, penelitian ini akan mencoba menerapkan *Adaptive Neuro Fuzzy Inference System* yaitu gabungan dari metode *Fuzzy Logic* dan *Artificial Neural Network*. Dengan *Fuzzy Logic*, AI akan mampu mengatur pergerakan lawan secara efektif dan efisien sesuai dengan cara berpikir manusia pada umumnya. Sedangkan dengan *Artificial Neural Network*, AI akan mampu belajar dari kesalahan setiap langkah dan melakukan *update* pada parameter dari *fuzzy membership function* yang sudah ada sesuai dengan kebutuhan tergantung dari hasil setiap pergerakan.

## 2. LANDASAN TEORI

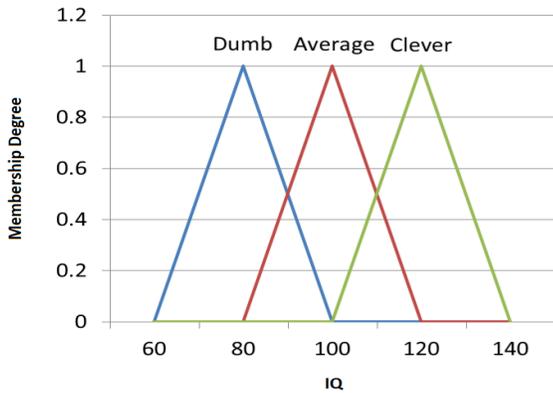
### 2.1 Fuzzy Logic

*Fuzzy Logic* (Logika fuzzy) merupakan perkembangan dari logika Boolean oleh Lotfi Zadeh pada tahun 1965 berdasarkan teori matematis *Fuzzy Sets*, yang merupakan generalisasi dari teori himpunan klasik. Dengan memperkenalkan konsep derajat dalam verifikasi suatu kondisi yang memungkinkan kondisi berada dalam keadaan selain benar atau salah, logika *fuzzy* memberikan fleksibilitas penalaran yang sangat berharga, yang memungkinkan untuk memperhitungkan ketidakakuratan dan ketidakpastian. Salah satu keuntungan logika *fuzzy* adalah kondisi-kondisi yang ada dalam seperangkat aturan ditetapkan dalam *Natural Language* [2]. Berikut merupakan beberapa contoh *Fuzzy Rules*:

- IF Target\_Far AND Energy\_Loads THEN Deploy Troop
- IF Target\_Far AND Energy\_Okay THEN Wait For Energy

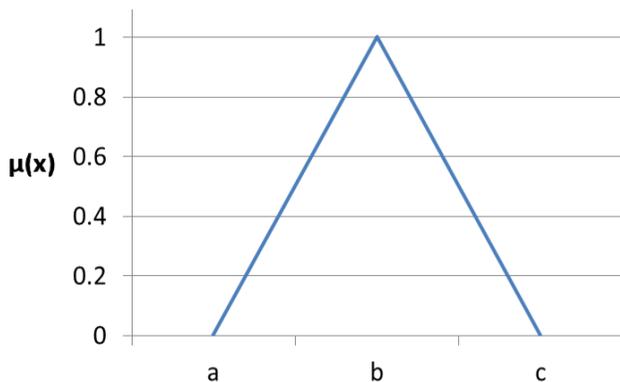
Peraturan-peraturan seperti itu memang sangat deskriptif dan dapat dimengerti dengan mudah oleh manusia, tetapi akan cukup sulit untuk menerjemahkan peraturan-peraturan tersebut ke dalam

bahasa yang dapat dimengerti oleh komputer. Kata-kata seperti “jauh”, “sangat dekat”, dan “halus” tidak memiliki batasan dan angka yang pasti. Karena itu, dalam *Fuzzy Logic* konsep-konsep kategori seperti itu tidak dinyatakan dalam interval diskrit, melainkan dengan *Fuzzy Sets*, yang memungkinkan suatu nilai untuk ditempatkan pada lebih dari satu kategori dalam derajat tertentu untuk setiap kategori. Proses tersebut dikenal sebagai *Fuzzification*. Dengan nilai yang sudah difuzzifikasi, komputer dapat memproses nilai tersebut dan mengaplikasikannya pada peraturan-peraturan (*Fuzzy Rules*) yang sudah ditentukan untuk menghasilkan suatu nilai output dalam bentuk *fuzzy* yang biasanya akan melalui proses *defuzzification*, yaitu mengembalikan nilai tersebut ke dalam bentuk *crisp*. *Crisp* sendiri adalah suatu nilai yang memiliki batasan-batasan pasti [1].



**Gambar 1. Contoh Fuzzy Set pembagian kategori Dumb, Average atau Clever berdasarkan IQ**

Gambar 1 merupakan contoh dari sebuah *Fuzzy Variable* yang terdiri dari 3 buah *Fuzzy Sets* dimana masing-masing *Fuzzy Set* mewakili satu buah kategori yaitu *Dumb*, *Average* dan *Clever*. *Fuzzy Sets* bisa dinyatakan dalam beberapa bentuk yang berbeda, tetapi yang paling sering digunakan adalah *triangular* dan *trapezoidal* dimana *trapezoidal* bisa dibagi lagi menjadi bentuk *left-shoulder* dan *right-shoulder* yang hanya menggunakan setengah bagian dari trapesium, kiri untuk *left-shoulder* dan kanan untuk *right-shoulder* [5].



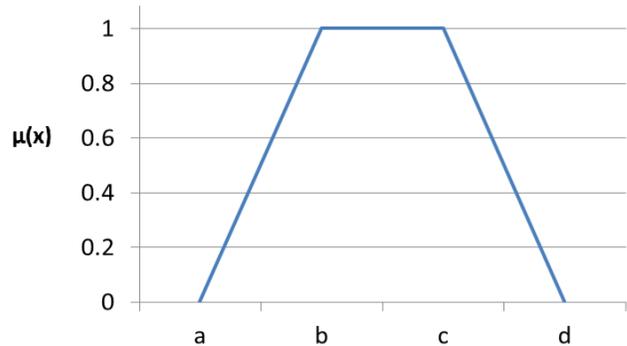
**Gambar 2. Triangular Fuzzy Set**

Gambar 2 merupakan *triangular fuzzy set* yang memiliki 3 titik koordinat untuk menunjukkan daerah hasil dari sebuah *Membership Function*.

$$\begin{aligned} \mu(x) &= 0, \text{ untuk } x < a \\ \mu(x) &= \frac{x-a}{b-a}, \text{ untuk } a \leq x \leq b \\ \mu(x) &= \frac{c-x}{c-b}, \text{ untuk } b < x \leq c \\ \mu(x) &= 0 \text{ untuk } x > c \end{aligned}$$

**Gambar 3. Persamaan Fuzzy Value dari Triangular Fuzzy Set**

Gambar 3 merupakan persamaan yang digunakan untuk mencari *value* dari *triangular fuzzy set*.



**Gambar 4. Trapezoid Fuzzy Set**

Gambar 4 merupakan *trapezoid fuzzy set* yang memiliki 4 titik koordinat untuk menunjukkan daerah hasil dari sebuah *Membership Function*.

$$\begin{aligned} \mu(x) &= 0, \text{ untuk } x < a \\ \mu(x) &= \frac{x-a}{b-a}, \text{ untuk } a \leq x \leq b \\ \mu(x) &= 1 \text{ untuk } b < x \leq c \\ \mu(x) &= \frac{d-x}{d-c}, \text{ untuk } c < x \leq d \\ \mu(x) &= 0 \text{ untuk } x > d \end{aligned}$$

**Gambar 5. Persamaan Fuzzy Value dari Trapezoid Fuzzy Set**

Gambar 5 merupakan persamaan yang digunakan untuk mencari *value* dari *trapezoid fuzzy set*.

## 2.2 Artificial Neural Network

*Artificial Neural Network* adalah sebuah sistem yang dirancang untuk dapat menyerupai kemampuan otak manusia yang mampu menerima *input* berupa stimulasi, melakukan proses pembelajaran berdasarkan *input* tersebut, dan menghasilkan *output* yang sesuai. Sistem tersebut belajar untuk melakukan sesuatu dari pengalaman, tanpa harus diprogram secara spesifik untuk melakukan hal tersebut [6].

## 2.3 Adaptive Neuro Fuzzy Inference System

*Adaptive Neuro Fuzzy Inference System* (ANFIS) adalah kombinasi dari dua metode yaitu *Artificial Neural Network* dan *Fuzzy Logic*. Logika *Fuzzy* memiliki kemampuan untuk mengubah

aspek kualitatif pengetahuan dan wawasan manusia ke dalam proses analisis kuantitatif yang tepat. Namun, metode ini tidak memiliki metode terdefinisi yang dapat digunakan sebagai panduan dalam proses transformasi pemikiran manusia ke dalam *rule based Fuzzy Inference System* (FIS), dan juga memerlukan waktu yang cukup lama untuk menyesuaikan *Membership Functions*, tidak seperti ANN yang memiliki kemampuan lebih tinggi dalam proses pembelajaran untuk dapat beradaptasi dengan lingkungannya [4]. Oleh karena itu, ANN dapat digunakan untuk menyesuaikan *Membership Functions* secara otomatis dan mengurangi tingkat kesalahan dalam penentuan *Fuzzy Rules* [7].

## 2.4 Real Time Strategy Battle Arena Game

*Game* dengan *genre Real Time Strategy Battle Arena* biasanya merupakan pertandingan antara 2 tim dalam 1 arena. *Gameplay* maupun fitur dari setiap *game* dengan *genre RTS-BA* dapat berbeda-beda, namun tujuan utamanya adalah mempertahankan *base* pemain dan menghancurkan *base* musuh [3].

## 3. DESAIN SISTEM

### 3.1 Desain Game

Judul *game* yang akan dibuat adalah “*War of the Beasts*”. *Game* ini menggunakan *Adaptive Neuro Fuzzy Inference System* untuk mengatur langkah-langkah musuh. Hal itu memungkinkan musuh untuk mengambil langkah-langkah yang paling tepat untuk melawan pemain, yang sesuai dengan *Fuzzy Inference Rules* yang sudah disesuaikan dengan cara berpikir manusia. Apabila musuh kalah, musuh juga dapat menganalisa letak kesalahannya secara otomatis, lalu melakukan perubahan pada *fuzzy membership function* yang berperan dalam menentukan *fuzzy rules* mana yang akan diikuti, sehingga untuk kedepannya musuh dapat mengambil langkah yang lebih tepat jika berada dalam keadaan yang sama dengan sebelumnya.

Pada saat *game* dimulai, pemain dan musuh masing-masing akan memiliki 5 poin energi yang dapat digunakan untuk mengeluarkan pasukan. Energi akan terus bertambah 1 poin setiap 3 detik dan jumlah maksimal energi adalah 10 poin. Jika jumlah energi sudah mencapai 10 poin, maka energi tidak akan bisa bertambah hingga energi digunakan untuk mengeluarkan pasukan. Jumlah energi yang dibutuhkan untuk mengeluarkan pasukan berbeda-beda tergantung dari jenis pasukan. Pada 60 detik terakhir *game*, kecepatan penambahan energi akan meningkat hingga dua kali lipat menjadi 2 poin setiap 3 detik.

Setiap *game* akan berlangsung selama 180 detik. Sebelum waktu habis, pemain atau musuh harus berhasil menghancurkan *turret* utama milik lawannya untuk memenangkan *game*. Jika waktu habis dan belum ada yang berhasil menghancurkan *turret* utama milik lawannya, maka pemenangnya adalah tim yang berhasil menghancurkan lebih banyak *turret* pertahanan milik lawannya. Jika jumlah *turret* pertahanan yang tersisa milik pemain maupun musuh sama saat waktu habis, maka akan ada perpanjangan waktu 60 detik. Selama 60 detik tersebut, tim yang pertama kali berhasil menghancurkan salah satu *turret* milik lawannya akan menang. Namun jika tidak ada yang berhasil menghancurkan *turret* lawannya, maka *game* akan berakhir seri.

*Arena* pada *game* ini dibagi menjadi 2 wilayah, yaitu wilayah pemain dan wilayah musuh. Batas kedua wilayah tersebut merupakan jurang pemisah yang ada di tengah *arena*. Pemain maupun musuh hanya dapat mengeluarkan pasukan di wilayah

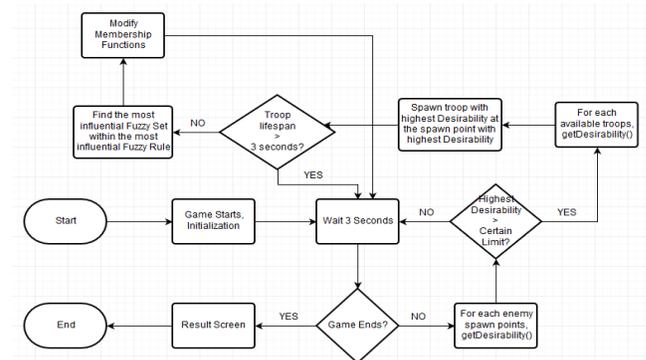
masing-masing. Pasukan-pasukan yang telah dikeluarkan akan berjalan secara otomatis ke wilayah lawannya.

Pemain dapat mengeluarkan pasukan dengan cara menggeser salah satu dari 4 kartu pasukan yang ada di *panel* pasukan ke posisi manapun yang diinginkan di wilayah pemain. Kartu pasukan yang pertama kali muncul pada *panel* tersebut merupakan 4 kartu dengan urutan pertama hasil *random* dari 8 jenis pasukan yang telah dipilih oleh pemain untuk dibawa ke dalam *game*. Urutan tersebut akan bersifat tetap selama *game* berlangsung sehingga jika pemain mengeluarkan salah satu pasukan, maka pasukan dengan urutan berikutnya akan mengisi tempat kartu pasukan yang telah dikeluarkan oleh pemain. Setelah mencapai kartu dengan urutan terakhir, maka yang akan muncul berikutnya merupakan kartu yang pertama kali dikeluarkan oleh pemain.

Strategi yang harus dipikirkan oleh pemain adalah di posisi manakah pemain harus mengeluarkan pasukan untuk mencapai hasil yang paling efektif dalam berbagai macam keadaan. Pemain juga harus memikirkan pasukan manakah yang ada di *panel* pasukan yang paling tepat untuk dikeluarkan di posisi tersebut. Pemain harus mempertimbangkan berbagai aspek untuk dapat menyerang sekaligus bertahan dengan penggunaan energi yang seefisien mungkin. Hal yang sama juga berlaku untuk musuh. Namun, setiap langkah musuh akan diatur oleh *Adaptive Neuro Fuzzy Inference System*.

Selama permainan berlangsung, pemain dapat mengatur posisi, sudut kemiringan, dan perbesaran dari *camera* yang menampilkan gambar permainan. Pemain dapat menggeser kamera ke depan, kiri, belakang dan kanan dengan menekan tombol W, A, S, D pada *keyboard*. Untuk mengatur kemiringan *camera*, pemain dapat menggerakkan mouse sambil menahan tombol *right click mouse*, sedangkan untuk mengatur perbesaran *camera*, pemain dapat menggunakan *scroll wheel* yang ada pada *mouse*.

### 3.2 Desain AI



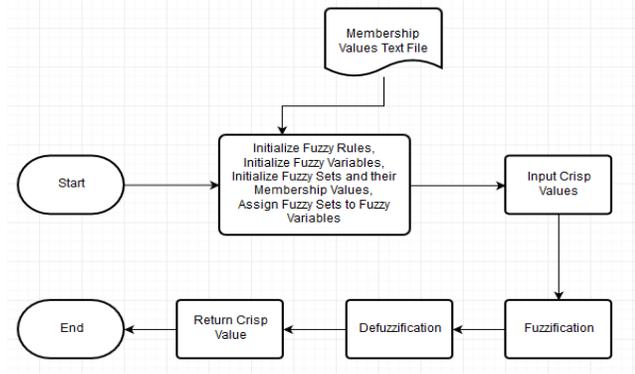
Gambar 6. Flowchart AI

Gambar 6 di atas merupakan alur dari AI yang mengatur langkah-langkah pergerakan musuh dalam *game* ini. Terdapat 8 buah titik di wilayah musuh yang merupakan tempat dimana pasukan musuh dapat muncul. Setiap 3 detik selama *game* berlangsung, program akan melakukan penghitungan *Desirability* pada setiap titik menggunakan *Fuzzy Inference System* untuk mengetahui titik manakah yang paling sesuai untuk mengeluarkan pasukan, yaitu titik dengan nilai *Desirability* yang paling tinggi. *Delay* 3 detik diterapkan untuk mengoptimasi *game* agar proses *Fuzzy Inference System* tidak berjalan tanpa henti dan membuat *game* semakin berat.

Jika nilai *Desirability* yang paling tinggi mencapai batas tertentu, maka program akan melakukan penghitungan yang sama pada setiap pasukan musuh yang dapat dikeluarkan untuk mengetahui pasukan manakah yang paling sesuai untuk dikeluarkan di titik tersebut. Namun jika *Desirability* setiap titik tidak ada yang mencapai batas tersebut, maka musuh tidak akan mengeluarkan pasukan hingga 3 detik berikutnya.

Setelah suatu pasukan dikeluarkan, program akan menghitung berapa lama pasukan tersebut dapat bertahan di *arena*. Jika pasukan langsung mati dalam waktu 3 detik atau kurang, maka hal itu akan dihitung sebagai kesalahan langkah oleh AI dan *Neural System* akan mencari *Fuzzy Rule* manakah yang paling mempengaruhi keputusan sistem untuk mengeluarkan pasukan di titik tersebut. Setelah menemukan *Fuzzy Rule* yang paling berpengaruh, sistem akan mencari *Fuzzy Set* yang paling berpengaruh dalam *Fuzzy Rule* tersebut lalu menggeser *Membership Function* dari *Fuzzy Set* tersebut sebesar *learning rate* yang sudah ditentukan.

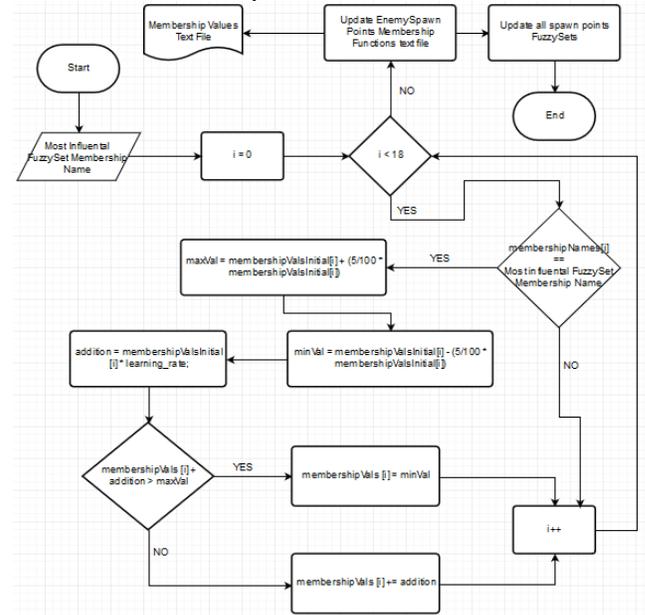
### 3.2.1 Alur Fuzzy Inference System



Gambar 7. Alur Fuzzy Inference System

Gambar 7 di atas merupakan alur dari proses penghitungan nilai *Desirability* menggunakan *Fuzzy Inference System* pada *game* ini. Pertama, program akan menginisialisasi seluruh *Fuzzy Rules* dan *Fuzzy Variables* yang akan digunakan, termasuk *Fuzzy Variable Desirability* sebagai *variable* output yang akan menjadi tolok ukur nilai *Desirability*. Seluruh *Fuzzy Sets* beserta masing-masing *Membership Values*-nya juga diinisialisasi oleh program sebelum kemudian dimasukkan ke dalam *Fuzzy Variables* yang sesuai. Setelah itu, program akan menerima input berupa *Crisp Values*. *Crisp Values* akan dimasukkan ke dalam proses *Fuzzification* dimana *Crisp Values* diolah menggunakan *Fuzzy Inference Engine* untuk menghasilkan *Fuzzy Variable* baru yang *Fuzzy Sets*-nya sudah dipotong sesuai dengan *Confidence Value* dari setiap *Fuzzy Sets*. Selanjutnya, *Fuzzy Variable* tersebut akan dimasukkan ke dalam proses *Defuzzification* untuk menghasilkan sebuah *Crisp Value* berupa nilai *Desirability*. Proses *Defuzzification* dalam *game* ini dilakukan dengan menggunakan metode *Centroid*.

### 3.2.2 Alur Neural System



Gambar 8. Alur Neural System

Gambar 8 di atas merupakan alur dari proses pengubahan *Membership Function* dari suatu *Fuzzy Set* menggunakan *Neural System* dalam *game* ini. Jika musuh dianggap melakukan sebuah kesalahan langkah, maka program akan mencari *Fuzzy Set* manakah yang berpengaruh paling besar dalam kesalahan tersebut. Setelah itu, *Membership Functions* dari *Fuzzy Set* tersebut akan diubah sebesar 10% dengan harapan kemungkinan terjadinya kesalahan yang sama menjadi lebih kecil. Batas pergeseran *Membership Functions* dari suatu *Fuzzy Set* adalah sebesar 50% dari *Initial Membership Function Fuzzy Set* tersebut.

## 4. PENGUJIAN SISTEM

### 4.1 Pengujian Aplikasi

Pengujian ini dilakukan untuk mengetahui proses permainan secara keseluruhan ketika seseorang memainkan *game* ini. Pengujian ini dilakukan dengan memainkan *game* yang telah dibuat. Pada saat program dijalankan, maka akan muncul tampilan awal, yaitu tampilan *Main Menu* seperti dapat dilihat pada Gambar 9.



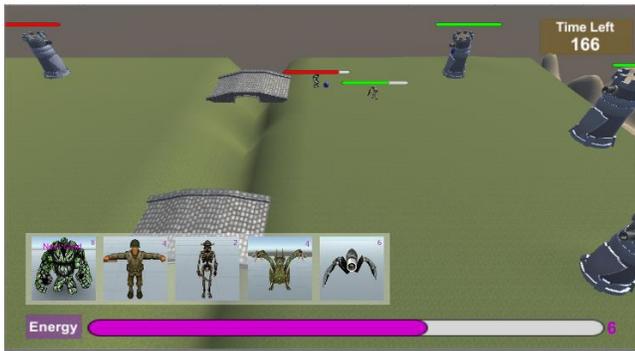
Gambar 9. Tampilan Main Menu

Pada tampilan *Main Menu*, terdapat 2 buah tombol yang dapat ditekan oleh pemain, yaitu tombol *Start Game* dan tombol *Quit Game*. Tombol *Start Game* berfungsi untuk memulai *game*, sedangkan tombol *Quit Game* berfungsi untuk keluar dari *game*. Jika pemain memilih untuk memulai *game*, maka pemain akan di arahkan ke tampilan pemilihan pasukan seperti dapat dilihat pada Gambar 10.



Gambar 10. Tampilan pemilihan pasukan

Pada Gambar 10, dapat dilihat bahwa tampilan pemilihan pasukan memiliki 2 buah *panel* yang berisi beberapa *slot* kartu pasukan. *Panel* yang di atas dengan 8 *slot* kartu pasukan merupakan *panel* tempat kartu pasukan yang sudah dipilih dan akan dibawa ke dalam *game*. Sedangkan *panel* yang bawah dengan 10 *slot* kartu pasukan merupakan panel tempat pemain dapat memilih kartu untuk digeser ke *panel* yang di atas. Tampilan ini juga memiliki 2 buah tombol yaitu tombol *Main Menu* untuk kembali ke tampilan *Main Menu* dan tombol *Start Game* untuk memulai *game* apabila pemain sudah memilih 8 unit pasukan. Jika pemain menekan tombol *Start Game* tanpa memilih 8 unit pasukan, maka akan ada pemberitahuan bahwa pemain harus memilih 8 pasukan untuk dapat mulai. Pada tampilan ini, pemain juga dapat mengarahkan *mouse* ke salah satu kartu pasukan untuk melihat *stats* dari pasukan tersebut yang akan tampil di bagian kanan tampilan.



Gambar 11. Tampilan *game*

Setelah pemain memilih 8 unit pasukan dan menekan tombol *Start Game* pada tampilan pemilihan pasukan, maka pemain akan di arahkan di tampilan *game* seperti dapat dilihat pada Gambar 11. Pada tampilan ini, pemain dapat melihat waktu *game* yang tersisa pada bagian kanan atas. Pemain juga dapat melihat jumlah energi yang dimiliki pada bagian bawah tampilan. Di atas *bar* energi, terdapat sebuah panel berisi 5 buah *slot* kartu pasukan. *Slot* pertama berfungsi untuk menampilkan pasukan yang akan muncul selanjutnya, sedangkan keempat *slot* lainnya berisi kartu pasukan yang tersedia dan dapat digeser ke *arena* oleh pemain untuk mengeluarkan pasukan tersebut.

Pada tampilan ini, dapat dilihat bahwa *arena* terbagi menjadi 2 buah wilayah yang dipisah oleh 2 buah jembatan di tengah. Wilayah sebelah kanan merupakan wilayah milik pemain, sedangkan wilayah sebelah kiri merupakan wilayah milik musuh. Setiap unit yang dapat berinteraksi seperti pasukan dan *turret* akan memiliki sebuah *bar* di bagian atas unit yang menunjukkan sisa *Hit Points* dari unit tersebut. *Bar* milik unit pemain akan berwarna hijau sedangkan *bar* milik unit musuh akan berwarna merah. Setelah *game* selesai, maka pemain akan diarahkan ke tampilan hasil *game* seperti dapat dilihat pada Gambar 12.



Gambar 12. Tampilan hasil *game*

## 4.2 Pengujian *Fuzzy Inference System*

Pengujian ini dilakukan untuk mengetahui apakah *Fuzzy Inference System* yang diterapkan pada *game* ini sudah tepat atau tidak. Pengujian ini dilakukan dengan memainkan *game* yang telah dibuat, lalu menganalisa *output* yang dihasilkan oleh penghitungan *Fuzzy Inference System* dalam *game* untuk mengetahui apakah *output* tersebut sudah sesuai dengan *Fuzzy Rules* yang berlaku atau belum.

Tabel 1. Daftar koordinat titik tempat menaruh pasukan musuh pada *arena game*

Spawner Name	Position		
	X	Y	Z
Spawner 1	711.0	440.5	1112.0
Spawner 2	165.0	440.5	1324.0
Spawner 3	380.0	440.5	1723.7
Spawner 4	880.0	440.5	1723.7
Spawner 5	80.0	440.5	1723.7
Spawner 6	241.0	440.5	1112.0
Spawner 7	610.0	440.5	1723.7
Spawner 8	825.0	440.5	1324.0

Tabel 1 merupakan daftar titik tempat menaruh pasukan musuh. Setiap 3 detik, masing-masing titik tersebut akan dihitung nilai *Desirability*-nya menggunakan *Fuzzy Inference System*. Setelah itu, pasukan musuh akan ditempatkan di titik dengan nilai *Desirability* yang paling tinggi. Nilai *Desirability* dari suatu titik ditentukan oleh *input* berupa jarak titik dengan pasukan lawan yang terdekat dengan titik tersebut (*Enemy Range*) dan juga jumlah energi yang dimiliki saat penghitungan (*Energy Amount*).

Pengujian *Fuzzy Inference System* ini dilakukan sebanyak 2 kali, masing-masing dengan cara menempatkan 1 unit pasukan di arena. Setelah 3 detik, setiap titik menghitung jaraknya dengan pasukan tersebut untuk dijadikan salah satu *input* penghitungan nilai *Desirability*-nya. Tabel 2 dan 3 merupakan hasil pengujian yang berupa data *input* dan *output* penghitungan nilai *Desirability* setiap titik menggunakan *Fuzzy Inference System*.

**Tabel 2. Hasil pengujian pertama *Fuzzy Inference System***

Spawner Name	Input		Output	Match with Fuzzy Rules
	Enemy Range	Energy Amount	Desirability	
Spawner 1	552.945 (Medium)	5.005 (Medium)	47.352 (Desirable)	Yes
Spawner 2	488.167 (Medium)	5.005 (Medium)	50.000 (Desirable)	Yes
Spawner 3	896.285 (Far)	5.005 (Medium)	30.185 (Undesirable)	Yes
Spawner 4	1097.173 (Far)	5.005 (Medium)	30.000 (Undesirable)	Yes
Spawner 5	896.130 (Far)	5.005 (Medium)	30.193 (Undesirable)	Yes
Spawner 6	272.164 (Close)	5.005 (Medium)	50.000 (Desirable)	Yes
Spawner 7	961.998 (Far)	5.005 (Medium)	30.000 (Undesirable)	Yes
Spawner 8	767.279 (Far)	5.005 (Medium)	36.636 (Undesirable)	Yes
<b>Chosen Spawner</b>	Spawner 2			

**Tabel 3. Hasil pengujian kedua *Fuzzy Inference System***

Spawner Name	Input		Output	Match with Fuzzy Rules
	Enemy Range	Energy Amount	Desirability	
Spawner 1	600.532 (Medium)	7.003 (High)	68.517 (VeryDesirable)	Yes
Spawner 2	570.300 (Medium)	7.003 (High)	69.295 (VeryDesirable)	Yes
Spawner 3	978.765 (Far)	7.003 (High)	67.309 (VeryDesirable)	Yes
Spawner 4	1166.781 (Far)	7.003 (High)	67.309 (VeryDesirable)	Yes
Spawner 5	977.723 (Far)	7.003 (High)	67.309 (VeryDesirable)	Yes
Spawner 6	355.255 (Close)	7.003 (High)	70.727 (VeryDesirable)	Yes
Spawner 7	1039.925 (Far)	7.003 (High)	67.309 (VeryDesirable)	Yes
Spawner 8	824.333 (Far)	7.003 (High)	66.645 (VeryDesirable)	Yes
<b>Chosen Spawner</b>	Spawner 6			

### 4.3 Pengujian *Neural System*

Pengujian ini dilakukan untuk mengetahui apakah *Neural System* yang diterapkan pada *game* ini sudah tepat atau tidak. Pengujian ini dilakukan dengan cara membiarkan *game* ini dimainkan oleh 10 orang yang berbeda, masing-masing sebanyak 10 kali. Pola kemenangan musuh setelah 10 kali bertanding melawan 10 orang

tersebut akan dianalisa untuk mengetahui apakah perubahan *Membership Function* yang dilakukan cukup signifikan atau tidak dalam mempengaruhi menang atau kalahnya musuh. Tabel 4 dan 5 merupakan data-data pengujian oleh 2 orang dengan perubahan *Membership Function* yang paling signifikan.

**Tabel 4. Hasil pengujian *Neural System* oleh orang ketiga**

No.	Membership Values																W		
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P		Q	R
1	0	300	500	300	500	900	500	900	1500	0	3	5	3	5	8	5	8	10	P1
2	0	300	500	300	500	900	550	990	1650	0	3	5	3	5	8	5	8	10	AI
3	0	300	500	300	500	900	550	990	1650	0	3	5	3	5	8	5	8	10	AI
4	0	300	500	300	500	900	550	990	1650	0	3	5	3	5	8	5	8	10	P1
5	0	300	500	300	500	900	550	990	1650	0	3	5	3	5	8	5	8	10	P1
6	0	330	550	300	500	900	550	990	1650	0	3	5	3	5	8	5	8	10	AI
7	0	330	550	300	500	900	550	990	1650	0	3.3	5.5	3	5	8	5	8	10	P1
8	0	330	550	300	500	900	600	1080	1800	0	3.3	5.5	3	5	8	5	8	10	AI
9	0	330	550	300	500	900	600	1080	1800	0	3.3	5.5	3	5	8	5.5	8.8	11	AI
10	0	330	550	300	500	900	600	1080	1800	0	3.3	5.5	3	5	8	5.5	8.8	11	AI

**Tabel 5. Hasil pengujian *Neural System* oleh orang kesepuluh**

No.	Membership Values																W		
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P		Q	R
1	0	300	500	300	500	900	500	900	1500	0	3	5	3	5	8	5	8	10	P1
2	0	300	500	300	500	900	500	900	1500	0	3	5	3	5	8	5	8	10	AI
3	0	300	500	300	500	900	500	900	1500	0	3	5	3.3	5.5	8.8	5	8	10	AI
4	0	300	500	300	500	900	500	900	1500	0	3	5	3.3	5.5	8.8	5	8	10	P1
5	0	300	500	300	500	900	500	900	1500	0	3	5	3.3	5.5	8.8	5	8	10	P1
6	0	300	500	300	500	900	500	900	1500	0	3	5	3.3	5.5	8.8	5	8	10	P1
7	0	300	500	300	500	900	500	900	1500	0	3	5	3.3	5.5	8.8	5	8	10	P1
8	0	330	550	300	500	900	500	900	1500	0	3	5	3.3	5.5	8.8	5	8	10	AI
9	0	330	550	300	500	900	500	900	1500	0	3	5	3.3	5.5	8.8	5	8	10	AI
10	0	360	600	300	500	900	500	900	1500	0	3	5	3.3	5.5	8.8	5	8	10	AI

Keterangan tabel 4 dan 5:

- A = Target Close Left Point
- B = Target Close Peak Point
- C = Target Close Right Point
- D = Target Mid Left Point
- E = Target Mid Peak Point
- F = Target Mid Right Point
- G = Target Far Left Point
- H = Target Far Peak Point
- I = Target Far Right Point
- J = Energy Low Left Point
- K = Energy Low Peak Point
- L = Energy Low Right Point
- M = Energy Med Left Point
- N = Energy Med Peak Point
- O = Energy Med Right Point
- P = Energy High Left Point
- Q = Energy High Peak Point
- R = Energy High Right Point
- W = Winner
- P1 = Player

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Setelah dilakukan perancangan sistem, pengimplementasian, pengujian, dan analisa terhadap aplikasi yang telah dibuat, dapat ditarik kesimpulan sebagai berikut:

- Musuh dalam *game* dapat menentukan posisi yang terbaik untuk mengeluarkan pasukan dan juga pasukan yang paling sesuai untuk dikeluarkan dengan menggunakan metode *Fuzzy Inference System*.
- Setiap keputusan yang diambil oleh musuh dalam *game* dapat dilakukan dengan mengubah seluruh parameter yang digunakan untuk *input* dari bentuk *crisp values* ke bentuk *fuzzy values*, lalu diolah dengan *Fuzzy Inference Engine* untuk menghasilkan *output* yang paling sesuai.
- Perubahan *Membership Function* untuk meminimalisir kesalahan langkah oleh musuh dapat dilakukan dengan menggunakan metode *Neural System*.
- Pengecekan berapa lama suatu unit musuh dapat bertahan di dalam arena sebagai faktor yang menentukan salah atau tidaknya langkah yang diambil musuh masih kurang efektif karena pengaruh dari perubahan *Membership Function* yang dihasilkan tidak terlalu signifikan, dan kesalahan langkah oleh musuh yang ditentukan dengan faktor tersebut masih cukup jarang terjadi.
- *Real Time Strategy Battle Arena Game* dengan AI yang digerakkan menggunakan *Adaptive Neuro Fuzzy Inference System* sebagai musuh cukup menarik dan menantang untuk dimainkan, meskipun pergerakan AI masih belum bisa menyamai pergerakan musuh manusia secara keseluruhan.

### 5.2 Saran

Beberapa saran untuk pengembangan aplikasi kedepannya adalah:

- *Game* ini dapat dikembangkan untuk beberapa *platform* lain seperti Android dan iOS sehingga *game* dapat dimainkan melalui *mobile devices*.
- *Fuzzy Inference System* dalam *game* ini dapat dikembangkan dengan menambah beberapa faktor lain untuk menentukan langkah pergerakan musuh.
- *Neural System* dalam *game* ini dapat dikembangkan agar tidak hanya mengecek dan melakukan *update* pada *membership function* dari *variable Enemy Range* saja, tetapi dari semua *variable* yang digunakan oleh *game* ini.

## 6. REFERENCES

- [1] Buckland, M. 2005. Programming Game AI by Example. Plano: Wordware Publishing, Inc.
- [2] Derroncourt, F. 2013. Introduction to Fuzzy Logic. Cambridge: Massachusetts Institute of Technology.
- [3] Gavrilova, V. 2013. Cognitive Load and Flow in Multiplayer Online Battle Arena Games. Limassol: Cyprus University of Technology.
- [4] Jang, J.S.R. 1993. ANFIS: Adaptive-Network-Based Fuzzy Inference System. Berkeley: University of California.
- [5] Lidianto, S.V., Budhi, G.S. & Intan, R. 2016. Perancangan dan Pembuatan Action Game dengan Artificial Intelligence dan Machine Learning. Surabaya: Universitas Kristen Petra.
- [6] Shalev-Shwartz, S. & Ben-David, S. 2014. Understanding Machine Learning. Cambridge: Cambridge University Press.
- [7] Suparta, W. & Alhasa K.M. 2016. Modeling of Tropospheric Delays Using ANFIS.