

Implementasi Kecerdasan Buatan dalam Menentukan Aksi Karakter pada Game RPG dengan Logika Fuzzy Tsukamoto

Dhemma Ratanajaya*, Helmie Arif Wibawa

Program Studi Informatika

Universitas Diponegoro

Semarang

*ratanajayaa@gmail.com

Abstrak-*Turn-based Role Playing Game* (RPG) adalah salah satu genre *video game* yang menggunakan sistem pertarungan antara dua kubu yang salah satunya dikontrol oleh pemain dan kubu lainnya oleh kecerdasan buatan. Pada RPG yang beredar di pasaran masih banyak *game* yang memiliki sistem kecerdasan buatan yang masih belum dapat mengambil keputusan yang paling baik untuk memenangkan pertarungan. Pada artikel ini dibahas tentang potongan dari *game* RPG yang didasarkan pada konvensi genre yang ditemukan pada *game* serupa. Setelah itu dibuat sistem kecerdasan buatan untuk mengontrol kubu musuh yang mampu melakukan pengambilan keputusan dengan tepat dalam pertarungan pada *game* tersebut. Logika *fuzzy* digunakan pada sistem kecerdasan buatan sebagai fungsi untuk melakukan pembobotan atas pilihan keputusan yang dapat dilakukan. Sistem kecerdasan buatan akan menggunakan metode inferensi Tsukamoto dan metode defuzzifikasi centroid. *Game* dibuat menggunakan *engine* Unity3D dan bahasa pemrograman C#. Proses pengembangan dilakukan dengan metode *extremme programming*. Hasil yang didapatkan menunjukkan bahwa implementasi logika *fuzzy* Tsukamoto dalam *game* RPG telah mampu mengoptimalkan *game* tersebut yaitu karakter dalam *game* mampu memilih aksi yang harus dilakukan terhadap target yang dihadapi.

Kata kunci: *Game*; RPG; *Fuzzy* Tsukamoto; Kecerdasan Buatan; Unity3D; *Extremme Programming*

1. Pendahuluan

Role Playing Game (RPG) adalah *game* yang pemainnya berperan sebagai seorang tokoh di dalam *game*, biasanya pada *setting* fantasi atau fiksi ilmiah, yang dapat berinteraksi dengan dunia di dalam *game* tersebut [1][2][3]. Genre ini adalah salah satu genre *video game* yang paling populer dan telah melahirkan beberapa sub-genre serta variasi regional seperti Japanese RPG (JRPG) dan Western RPG (WRPG).

Perbedaan utama antara JRPG dan WRPG terletak pada bagaimana battle atau pertarungan antara karakter pemain dan karakter musuh yang dikendalikan oleh komputer dilakukan. JRPG pada umumnya menggunakan *turn-based battle system*, yaitu sistem pertarungan antara dua kubu yang masing-masing kubu saling bergantian melakukan aksi dengan menunggu *input* dari pemain seperti permainan catur. Sedangkan WRPG pada umumnya menggunakan *real-time battle system*, yaitu sistem pertarungan yang setiap karakternya, baik karakter pemain ataupun karakter musuh melakukan aksi secara bersamaan [4].

Pada *real-time battle*, karakter musuh yang dikontrol oleh komputer memiliki keuntungan yang besar seperti akurasi yang sempurna dan kecepatan reaksi yang instan dibandingkan dengan karakter pemain yang dikontrol oleh manusia. Sehingga yang perlu diperhatikan dalam mendesain karakter musuh dalam *game* seperti ini adalah bagaimana membuatnya lebih mudah dikalahkan dengan memberikan sifat-sifat “manusiawi” seperti kemampuan

untuk melakukan kesalahan. Namun lain halnya pada *game turn-based*. Kecepatan berpikir dan akurasi tidak diperhitungkan sehingga karakter pemain yang dikontrol oleh manusia dapat dengan mudah mengalahkan karakter musuh. Untuk menambah tantangan bagi pemain dalam *game* jenis *turn-based* ini, karakter musuh pada *game* perlu ditingkatkan kemampuannya. Karakter musuh yang dikontrol oleh komputer harus lebih mampu membaca keadaan lawan yang dikontrol oleh pemain. Selain itu karakter musuh harus mampu menentukan aksi yang harus dilakukan terkait kondisi lawan yang dihadapi. Oleh karena itu dibutuhkan sebuah kecerdasan buatan pada karakter musuh. Teknologi sekarang ini dapat digunakan untuk mendesain kecerdasan buatan *game turn-based* [5]. Logika *fuzzy* sebagai salah satu metode di dalam memberikan kecerdasan buatan pada karakter musuh yang mana dapat digunakan untuk mengontrol karakter dalam *game* [6].

Logika *fuzzy* diperkenalkan pertama kali untuk pengembangan *game* pada 1996 pada *Game Developer Magazine* oleh Larry O’Brien dan sejak saat itu telah banyak dieksplorasi dan dikembangkan oleh pengembangan-pengembang lain. Bahkan Zarozinski menyatakan bahwa logika *fuzzy* selalu ada pada setiap *game*. Di dalam *game*, logika *fuzzy* pada umumnya diterapkan pada saat *game* atau karakter di dalam *game* perlu melakukan pengambilan keputusan berdasarkan data-data yang nilainya tidak tentu [7]. Shaout, *et al.* menggunakan model Mamdani untuk membuat ulang *game* Pac-Man [8]. Demikian juga Li, menggunakannya pada pembuatan ulang *game* Battle City [9].

Dibandingkan dengan teknik-teknik lainnya, sistem pengambilan keputusan berbasis logika *fuzzy* relatif mudah untuk didesain dan diimplementasikan. *Output* yang dihasilkannya tidak berbeda dengan sistem pengambilan keputusan berbasis data *crisp*. Perbedaannya hanyalah pada parameter atau *input* yang diterima, dan bagaimana mengolah *input-input* tersebut. Selain itu dari segi beban komputasi, logika *fuzzy* juga termasuk cukup ringan. Pengolahan data *input* dilakukan dengan aritmatika sederhana dan jumlahnya pun proporsional dengan jumlah data *input* [9][10].

Salah satu model inferensi logika *fuzzy* adalah model Tsukamoto. Pada model ini proses inferensi dilakukan dengan mendefinisikan beberapa *rules* dengan masing-masing *rule* menggunakan dua atau lebih variabel. Pada setiap *rule*, nilai alfa diambil dari nilai minimum atau maksimum dari kedua variabel. Setelah itu dilakukan proses defuzifikasi dengan metode rata-rata terbobot. Jika dibandingkan dengan model inferensi lainnya, model Tsukamoto mempunyai kemampuan yang lebih tinggi dalam pengambilan keputusan untuk melakukan suatu tindakan [11]. Model Tsukamoto mempunyai *output crisp* sehingga pengaturan hubungan antar nilai *input* dan *output* menjadi lebih optimal yang mana akan sangat membantu dalam pengaturan karakter pada *game* [12]. Penggunaan model Tsukamoto telah memberikan hasil yang baik dalam penentuan *kicking range* pada *game* sepak takraw [12], demikian juga dalam menentukan posisi pemain sepak bola [13]. Oleh karena itu pada artikel ini dibahas tentang penggunaan model Tsukamoto untuk mengatur tindakan karakter pada *game* RPG sehingga karakter musuh dalam *game* mampu melakukan aksi yang lebih optimal.

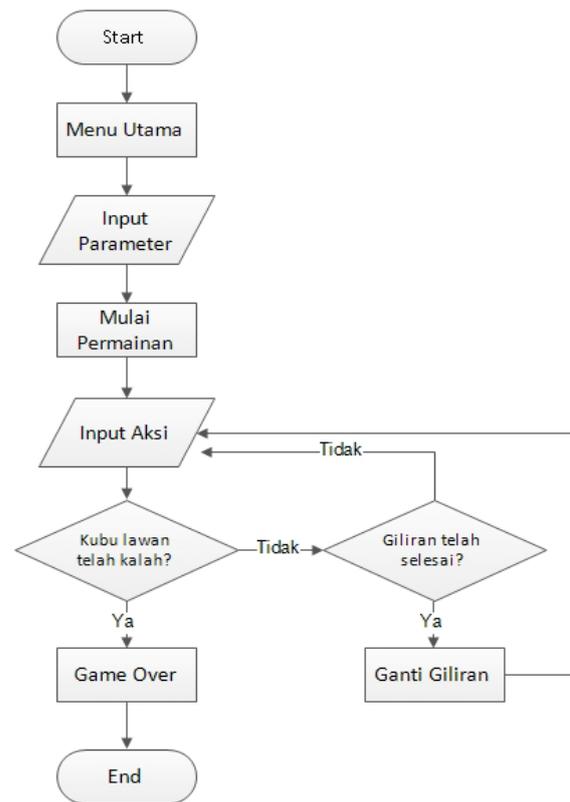
2. Metode

a. Desain Game

Game ini terdiri dari tiga subsistem yaitu Sistem Pertarungan, Sistem Menu, dan Sistem Kecerdasan Buatan, dengan fungsinya masing-masing. Sistem Pertarungan berfungsi untuk mengatur jalannya pertarungan dalam *game*, Sistem Menu berfungsi untuk menjalankan antarmuka sebagai penghubung antara pemain dan Sistem Pertarungan, sedangkan Sistem Kecerdasan Buatan berfungsi untuk menjalankan kubu lawan. Alur sistem *game* ditunjukkan pada gambar 1.

Sebagaimana yang ditunjukkan pada Gambar 1. Pada saat pemain memulai *game*, *scene* pertama yang akan dihadapi adalah menu utama. Pemain dapat memasukkan parameter-parameter untuk menentukan *state* awal pertarungan yang meliputi karakter masing-masing kubu dan giliran pertama. Jika sudah selesai pemain dapat menekan tombol start untuk memulai pertarungan.

Pada *battle scene* pemain dan kecerdasan buatan dapat memberikan perintah aksi kepada sistem pertarungan pada gilirannya masing-masing. Setelah eksekusi setiap aksi akan dilakukan pengecekan apakah kubu lawan sudah kalah, jika ya *game* akan berhenti, pemain dapat memilih untuk kembali ke menu utama. Sedangkan jika tidak akan dilakukan pengecekan apakah giliran dari kubu yang sedang berjalan telah selesai jika ya ganti giliran, jika tidak tunggu *input* aksi dari karakter selanjutnya.



Gambar 1. Alur sistem *game*

Tabel 1. Data atribut

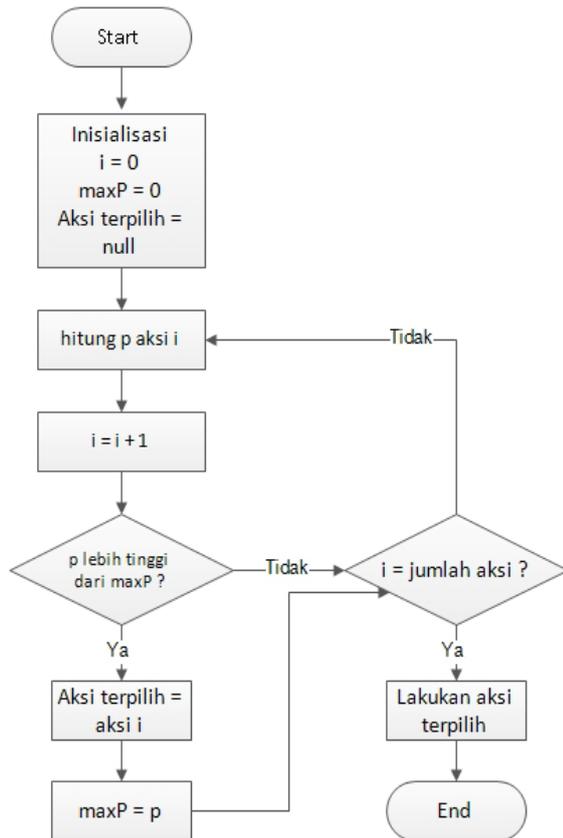
Atribut	Arti	Kegunaan
hp	Health Point	Kemampuan karakter untuk bertahan hidup
ep	Energy Point	Kemampuan karakter untuk melakukan aksi
atk	Attack	Kekuatan karakter dalam melakukan serangan fisik
def	Defense	Pertahanan karakter terhadap serangan fisik
mag	Magic	Kekuatan karakter dalam melakukan serangan mistis
res	Resistance	Pertahanan karakter terhadap serangan mistis
actions	Actions	Aksi-aksi yang dapat dilakukan oleh karakter

Tabel 2. Definisi atribut

Aksi	Cost	Target	Fungsi
Slash	12	Lawan	Mengurangi hp target
Punch	15	Lawan	Mengurangi hp dan ep target
Heal	40	Kawan	Menambah hp target
Fire	20	Lawan	Mengurangi hp target
Shoot	15	Lawan	Mengurangi hp target
Guard	0	Kawan	Menambah DEF dan RES target sebanyak 25% untuk satu giliran

Tabel 3. Data aksi

Karakter	HP	EP	ATK	DEF	MAG	RES	Actions
Knight	120	45	15	30	5	25	Slash, Guard
Fighter	100	60	30	20	5	15	Punch, Guard
Medic	90	90	5	15	30	30	Heal, Guard
Wizard	90	90	5	15	30	30	Fire, Guard
Archer	100	60	30	20	5	15	Shoot, Guard



Gambar 2. Alur proses kecerdasan buatan

b. Data Karakter dan Aksi

Pada tahap ini dirancang karakter-karakter dan aksi-aksi yang dapat dilakukan karakter tersebut. Desain dari karakter-karakter tersebut didasarkan pada konvensi RPG pada umumnya. Adapun game komersial yang dijadikan referensi untuk mendesain atribut-atribut karakter *game* ini yaitu Final Fantasy V dan Fire Emblem VII.

Pada game ini terdapat lima macam karakter, Knight, Fighter, Medic, Wizard, dan Archer. Masing-masing karakter memiliki 5 atribut yaitu Health Point (hp), Energy Point (ep) Attack (atk), Defense (def), Magic (mag), Resistance(res), dan Action. Atribut-atribut ini disajikan pada tabel 1. Adapun aksi yang dapat dilakukan oleh karakter adalah *slash*, *puch*, *heal*, *fire*, *shoot*, dan *guard*. Fungsi dari masing-masing aksi dapat dilihat pada tabel 2. Dari atribut dan aksi yang ada pada *game* ini maka masing-masing karakter akan memiliki kekuatan dan kelemahan masing-masing yang bersifat unik. *Knight* memiliki pertahanan tinggi dan serangan fisik lemah, *Fighter* memiliki serangan terhadap ep dan pertahanan sedang, *Medic* memiliki

kemampuan untuk memulihkan hp, *Wizard* memiliki serangan mistis kuat, dan *Archer* memiliki serangan fisik kuat serta pertahanan sedang. Detail dari atribut dan aksi yang dimiliki karakter dapat dilihat pada tabel 3.

Sebagian besar aksi yang dapat dilakukan oleh karakter memiliki kegunaan untuk menimbulkan *damage* (pengurangan nilai hp) terhadap karakter target. Masing-masing aksi memiliki rumus yang unik sebagai berikut

1. Aksi Slash

$$damage = \left(m \times \frac{actor.atk}{target.def} + b \right) \times y \quad (1)$$

2. Aksi Punch

$$damage = \left(m \times \frac{actor.atk}{0.5 \times (target.def + target.res)} + b \right) \times y \quad (2)$$

3. Aksi Heal

$$damage = -1 \times actor.mag \quad (3)$$

4. Aksi Fire

$$damage = \left(m \times \frac{actor.mag}{target.res} + b \right) \times y \quad (4)$$

5. Aksi Shoot

$$damage = \left(m \times \frac{actor.atk}{target.def} + b \right) \times y \quad (5)$$

6. Aksi Guard

$$target.def = 1.25 \times target.def \quad (6)$$

$$target.res = 1.25 \times target.res \quad (7)$$

dengan:

actor.atk : Atribut atk dari karakter yang melakukan aksi.

actor.mag : Atribut mag dari karakter yang melakukan aksi.

target.def : Atribut def dari karakter yang menerima aksi.

target.res : Atribut res dari karakter yang menerima aksi.

m : Konstanta yang bernilai 0.266. Merepresentasikan derajat kemiringan fungsi linier yang digunakan sebagai rumus damage aksi.

b : Konstanta yang bernilai -0.032

y : Konstanta yang bernilai 150

c. Desain Kecerdasan Buatan

Pengambilan keputusan pada kecerdasan buatan game ini dilakukan dengan cara membandingkan nilai *priority point* dari setiap aksi yang mungkin. Alur berjalannya kecerdasan buatan dapat dilihat pada Gambar 2.

Pertama-tama program menginisialisasi variabel prioritas maksimum dengan nilai nol, dan aksi terpilih dengan nilai kosong. Setelah itu program akan melakukan iterasi sebanyak jumlah aksi yang mungkin. Pada setiap iterasi satu aksi akan dihitung nilai prioritasnya dan dibandingkan nilai prioritas maksimum. Jika nilai prioritasnya lebih tinggi maka aksi tersebut akan dijadikan aksi terpilih.

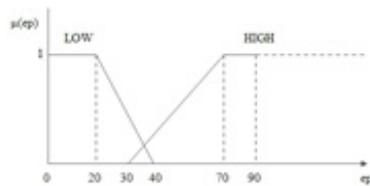
d. Himpunan Fuzzy

Logika *fuzzy* pada game ini merupakan inti dari sistem kecerdasan buatan secara keseluruhan. Pada tahap ini dirancang variabel-variabel *fuzzy*. Variabel-variabel yang digunakan untuk menentukan *priority* (p) suatu aksi mencakup *health point* (hp), *energy point* (ep), *dexterity*, dan *damage*. Variabel hp dan ep diambil secara langsung dari atribut target, variabel *dexterity* merupakan rata-rata dari defense dan resistance target, sedangkan variabel *damage* dihitung menggunakan rumus penghitungan *damage* yang dimiliki masing-masing aksi.

Fungsi keanggotaan masing-masing variabel *fuzzy* ditentukan dengan metode *trial and error* dengan pendekatan goal driven. Adapun fungsi keanggotaan tersebut adalah sebagai berikut:

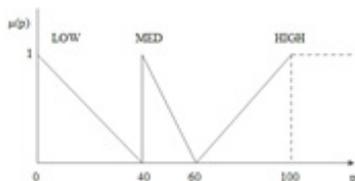
1) Aksi serangan *Slash*, *Fire*, dan *Shoot*

Variabel *damage* pada aksi *slash*



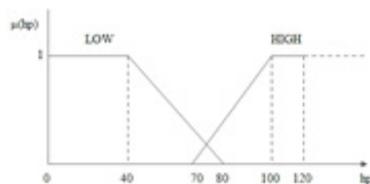
Gambar 3. Fungsi keanggotaan variabel *damage* pada aksi *slash*

Variabel *damage* pada aksi *fire* dan *shoot*



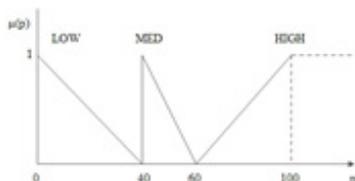
Gambar 4. Fungsi keanggotaan variabel *damage* pada aksi *fire* dan *shoot*

Variabel hp pada aksi *slash*, *fire*, dan *shoot*



Gambar 5. Fungsi keanggotaan variabel hp pada aksi *slash*, *fire*, dan *shoot*

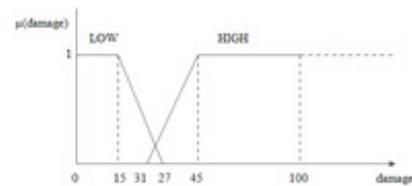
Variabel p pada aksi *slash*, *fire*, dan *shoot*



Gambar 6. Fungsi keanggotaan variabel p pada aksi *slash*, *fire*, dan *shoot*

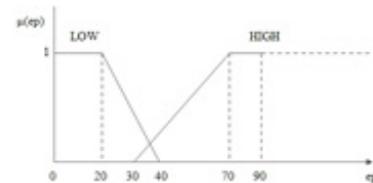
2) Aksi serangan *punch*

Variabel *damage* pada aksi *punch*



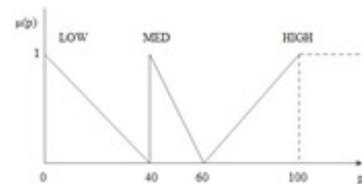
Gambar 7. Fungsi keanggotaan variabel *damage* pada aksi *punch*

Variabel ep pada aksi *punch*



Gambar 8. Fungsi keanggotaan variabel ep pada aksi *punch*

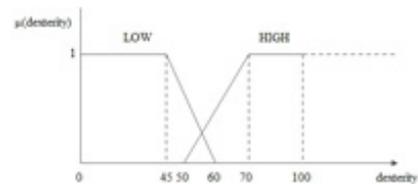
Variabel p pada aksi *punch*



Gambar 9. Fungsi keanggotaan variabel p pada aksi *punch*

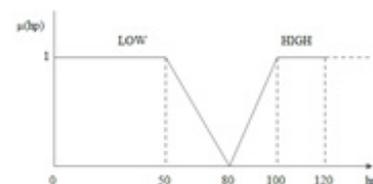
3) Aksi pemulihan (*Heal*)

Variabel *dexterity* pada aksi *heal*



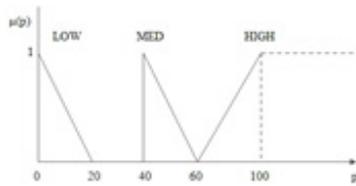
Gambar 10. Fungsi keanggotaan variabel *dexterity* pada aksi *heal*

Variabel hp pada aksi *heal*



Gambar 11. Fungsi keanggotaan variabel hp pada aksi *heal*

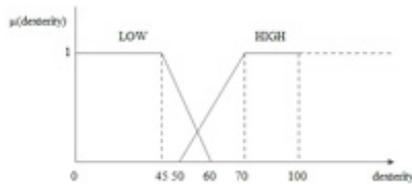
Variabel p pada aksi *beat*



Gambar 12. Fungsi keanggotaan variabel p pada aksi *beat*

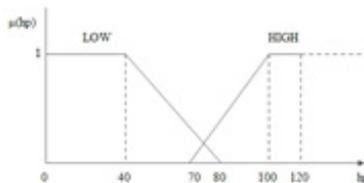
4) Aksi perlindungan (*guard*)

Variabel *dexterity* pada aksi *guard*



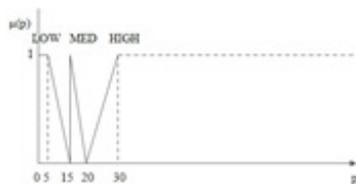
Gambar 13. Fungsi keanggotaan variabel *dexterity* pada aksi *guard*

Variabel *hp* pada aksi *guard*



Gambar 14. Fungsi keanggotaan variabel *hp* pada aksi *guard*

Variabel p pada aksi *guard*



Gambar 15. Fungsi keanggotaan variabel p pada aksi *guard*

e. Aturan Fuzzy

Berdasarkan peraturan *game* dan aksi-aksi yang telah ditentukan dapat didesain *fuzzy rules* sebagai basis dari pengambilan keputusan sistem kecerdasan buatan. Seperti pada fungsi keanggotaan *fuzzy* pada bab sebelumnya, perancangan *rules* juga dilakukan dengan metode *trial and error* dengan pendekatan *goal-driven*. Dalam artian aksi-aksi yang paling logis untuk dipilih pada berbagai situasi telah diketahui sehingga *fuzzy rules* perlu dirancang sedemikian rupa agar sistem kecerdasan buatan dapat memilih aksi-aksi tersebut.

Berdasarkan *fuzzy rules* yang digunakan, aksi dapat dikategorikan menjadi empat macam. Adapun keempat macam aksi tersebut beserta *fuzzy rules* yang digunakan adalah sebagai berikut:

1. Aksi serangan *Slash*, *Fire*, dan *Shoot*
Aksi *Slash*, *Fire*, dan *Shoot* berfungsi untuk mengurangi *hp* target (*damage*). Proses penghitungan *damage* dari ketiga aksi tersebut berbeda tetapi tujuannya sama. Tujuan dari kelompok aksi ini adalah untuk menimbulkan *damage* sebesar-besarnya atau mematikan target yang memiliki *hp* kecil. Sehingga *fuzzy rules* yang digunakan yaitu:
 - a. JIKA *damage* HIGH ATAU *hp* LOW MAKA p HIGH.
 - b. JIKA *damage* HIGH DAN *hp* HIGH MAKA p MED.
 - c. JIKA *damage* LOW ATAU *hp* LOW MAKA p MED.
 - d. JIKA *damage* LOW ATAU *hp* HIGH MAKA p LOW.
2. Aksi serangan *Punch*
Aksi *Punch* berfungsi untuk mengurangi *hp* dan *ep* target. Tujuan utama dari aksi ini adalah untuk menghambat atau menghentikan target dalam melakukan aksi pada giliran selanjutnya dengan cara mengurangi *ep* target di samping mengurangi *hp* seperti aksi serangan lainnya. Sehingga *fuzzy rules* yang digunakan yaitu:
 - a. JIKA *damage* HIGH DAN *ep* HIGH MAKA p HIGH.
 - b. JIKA *damage* HIGH DAN *ep* LOW MAKA p MED.
 - c. JIKA *damage* LOW ATAU *ep* HIGH MAKA p MED.
 - d. JIKA *damage* LOW DAN *ep* LOW MAKA p LOW.
3. Aksi pemulihan (*Heal*)
Aksi *Heal* berfungsi untuk menambah *hp* target. Tujuan dari aksi ini adalah untuk menyelamatkan karakter target dari kematian yang mungkin terjadi pada giliran selanjutnya. Sehingga *fuzzy rules* yang digunakan yaitu:
 - a. JIKA *dexterity* HIGH ATAU *hp* HIGH MAKA p LOW.
 - b. JIKA *dexterity* HIGH DAN *hp* LOW MAKA p HIGH.
 - c. JIKA *dexterity* LOW ATAU *hp* HIGH MAKA p LOW.
 - d. JIKA *dexterity* LOW DAN *hp* LOW MAKA p HIGH.
4. Aksi perlindungan (*Guard*)
Aksi *Guard* berfungsi untuk menambah pertahanan target untuk satu giliran. Tujuan dari aksi ini adalah untuk mengurangi *damage* dari serangan lawan yang mungkin terjadi pada giliran selanjutnya. Sehingga *fuzzy rules* yang digunakan yaitu:
 - a. JIKA *dexterity* HIGH DAN *hp* HIGH MAKA p LOW.
 - b. JIKA *dexterity* HIGH DAN *hp* LOW MAKA p MED.
 - c. JIKA *dexterity* LOW DAN *hp* LOW MAKA p MED.
 - d. JIKA *dexterity* LOW DAN *hp* LOW MAKA p HIGH.

Tabel 4. Pengujian karakter *knight* sampel *default*

Kubu	Karakter	hp	ep	Damage	P Slash	P Guard
Biru	Knight	120	45	16	20	-
	Wizard	90	90	36	40	-
	Archer	100	60	26	21.5	-
Merah	Knight	120	45	-	-	9.8
	Wizard	90	90	-	-	10
	Archer	100	60	-	-	10

Tabel 5. Pengujian karakter *knight* sampel *ekstrem*

Kubu	Karakter	hp	ep	Damage	P Slash	P Guard
Biru	Knight	40	33	16	46.9	-
	Wizard	90	70	36	40	-
	Archer	100	45	26	21.5	-
Merah	Knight	120	45	-	-	9.8
	Wizard	54	90	-	-	13.3
	Archer	100	60	-	-	10

Tabel 6. Pengujian karakter *fighter* sampel *default*

Kubu	Karakter	hp	ep	Damage	P Punch	P Guard
Biru	Knight	120	45	39	63.8	-
	Wizard	90	90	49	70	-
	Archer	100	60	64	67.5	-
Merah	Knight	120	45	-	-	9.8
	Wizard	90	90	-	-	10
	Fighter	100	60	-	-	10

Tabel 7. Pengujian karakter *fighter* sampel *ekstrem*

Kubu	Karakter	hp	ep	Damage	P Punch	P Guard
Biru	Knight	40	33	39	63.3	-
	Wizard	8	21	49	51.9	-
	Archer	100	45	64	66.3	-
Merah	Knight	68	33	-	-	14.5
	Wizard	54	70	-	-	13.3
	Fighter	100	45	-	-	10

Tabel 8. Pengujian karakter *medic* sampel *default*

Kubu	Karakter	hp	ep	Damage	P Heal	P Guard
Biru	Knight	120	45	-	-	-
	Wizard	90	90	-	-	-
	Archer	100	60	-	-	-
Merah	Knight	120	45	-45	0	9.8
	Medic	90	90	-45	3.3	10
	Archer	100	60	-45	0	10

Tabel 9. Pengujian karakter *medic* sampel *ekstrrm*

Kubu	Karakter	hp	ep	Damage	P Heal	P Guard
Biru	Knight	40	33	-	-	-
	Wizard	54	70	-	-	-
	Archer	100	45	-	-	-
Merah	Knight	120	33	-45	0	9.8
	Medic	90	90	-45	3.3	10
	Archer	18	60	-45	50	15

Tabel 10. Pengujian karakter *wizard* sampel *default*

Kubu	Karakter	hp	ep	Damage	P Fire	P Guard
Biru	Knight	120	45	44	21.1	-
	Wizard	90	90	36	28.6	-
	Archer	100	60	75	46.7	-
Merah	Knight	120	33	-	-	9.8
	Wizard	15	90	-	-	10
	Archer	100	60	-	-	10

Tabel 11. Pengujian karakter *wizard* sampel *ekstrem*

Kubu	Karakter	hp	ep	Damage	P Fire	P Guard
Biru	Knight	60	33	44	58.3	-
	Wizard	90	70	36	54.9	-
	Archer	100	45	75	46.7	-
Merah	Knight	120	45	-	-	9.8
	Wizard	54	90	-	-	15
	Archer	100	60	-	-	10

f. Defuzzifikasi

Defuzzifikasi pada model *Fuzzy* Tsukamoto dilakukan dengan menggunakan metode rata-rata terbobot. Rata-rata terbobot dihitung berdasarkan *fire strength* dan nilai hasil inferensi yang telah diperoleh pada tahap sebelumnya. Pada program ini inferensi *fuzzy* menggunakan empat rules sehingga rumus defuzzifikasi adalah sebagai berikut:

$$p = \frac{w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4}{w_1 + w_2 + w_3 + w_4} \quad (8)$$

Dimana

p : nilai prioritas hasil rata-rata terbobot

z : nilai hasil inferensi

w : fire strength

Hasil akhir p merupakan nilai prioritas dari aksi yang sedang dihitung. Nilai inilah yang kemudian digunakan sebagai pembandingan antara satu aksi dengan lainnya.

3. Hasil dan Pembahasan

Pada bagian ini akan dibahas efektivitas kinerja kecerdasan buatan yang dihasilkan. Untuk itu perlu dilakukan pengujian dengan menganalisis nilai prioritas (p) setiap aksi oleh satu karakter pada situasi-situasi tertentu. Dari nilai prioritas tersebut dapat dilihat aksi yang dipilih dan dianalisis apakah pemilihan aksi tersebut sudah tepat atau tidak. Tepat tidaknya suatu aksi dinilai berdasarkan nalar dengan merujuk pada peran masing-masing karakter yang telah dijabarkan sebelumnya.

Data pengujian masing-masing karakter diambil dari dua macam sampel, yaitu sampel awal dan sampel ekstrem. Sampel awal adalah data yang diambil pada saat pertarungan pertama kali dimulai. Sedangkan sampel ekstrem diambil pada saat terdapat satu karakter yang diperkirakan akan memiliki prioritas tinggi sebagai target suatu aksi, misalkan karakter musuh yang memiliki HP rendah.

Tabel 11. Pengujian karakter *archer* sampel default

Kubu	Karakter	hp	ep	Damage	P Shoot	P Guard
Biru	Knight	120	45	36	15.9	-
	Wizard	90	90	75	60	-
	Archer	100	60	56	31.5	-
Merah	Knight	120	45	-	-	9.8
	Wizard	90	90	-	-	10
	Archer	100	60	-	-	10

Tabel 12. Pengujian karakter *archer* sampel ekstrem

Kubu	Karakter	hp	ep	Damage	P Shoot	P Guard
Biru	Knight	40	33	36	61	-
	Wizard	90	70	75	60	-
	Archer	100	45	56	31.5	-
Merah	Knight	120	33	-	-	9.8
	Wizard	54	90	-	-	15
	Archer	100	60	-	-	10

a. Pengujian Karakter Knight

Karakter Knight adalah karakter yang memiliki atribut pertahanan kuat dan serangan lemah. Peran utama karakter ini adalah untuk melakukan aksi *slash* terhadap karakter lawan yang lemah terhadap serangan fisik. Aksi *guard* hanya dilakukan pada saat karakter tidak memiliki ep yang cukup untuk melakukan serangan. Hasil pengujian ditunjukkan pada tabel 4 dan tabel 5.

Tabel 4 menunjukkan bahwa pada saat karakter knight merah melakukan aksi pertama menghadapi musuh *knight*, *wizard* dan *archer*, aksi yang terpilih adalah slash terhadap *wizard* biru dengan nilai $p = 40$. Aksi tersebut memiliki prioritas tinggi karena menimbulkan *damage* yang paling tinggi. Tabel 5 menunjukkan bahwa pada situasi yang berbeda dengan karakter-karakter yang sama, aksi yang terpilih adalah *slash* terhadap karakter *knight* biru dengan nilai $p = 46.9$. Aksi tersebut terpilih karena *knight* biru sudah hampir mati dengan nilai $ep = 40$, meskipun *damage* yang dihasilkan lebih kecil daripada aksi *slash* terhadap *wizard* biru. Sedangkan untuk aksi *guard*, prioritas tertinggi adalah terhadap karakter *wizard* merah dengan nilai $p = 13.3$, pada kasus lain saat karakter *knight* merah tidak memiliki ep untuk melakukan aksi *slash*, aksi inilah yang akan dipilih. Prinsip yang sama terhadap penggunaan aksi *guard* juga berlaku bagi semua karakter lainnya kecuali *medic*.

b. Pengujian Karakter Fighter

Karakter Fighter adalah karakter yang memiliki atribut pertahanan sedang dan serangan yang mampu mengurangi ep target. Peran utama karakter ini adalah untuk melakukan aksi *punch* terhadap karakter yang memiliki ep tinggi, dimaksudkan agar karakter target tidak dapat banyak melakukan aksi pada giliran-giliran selanjutnya. Aksi *guard* hanya dilakukan pada saat karakter tidak memiliki ep yang cukup untuk melakukan serangan. Hasil pengujian karakter *fighter* dapat dilihat pada tabel 6 dan tabel 7.

Tabel 6 menunjukkan bahwa pada saat karakter *fighter* merah melakukan aksi pertama menghadapi musuh *knight*, *wizard*, dan *archer*, aksi yang terpilih adalah *punch* terhadap *wizard* biru dengan nilai $p = 70$. Aksi tersebut memiliki prioritas tinggi karena karakter target memiliki ep yang paling tinggi sebanyak 90 *point*. Tabel 7 menunjukkan

bahwa pada situasi yang berbeda dengan karakter-karakter yang sama, aksi yang terpilih adalah *punch* terhadap *archer* biru dengan nilai $p = 66.3$. Aksi tersebut terpilih karena *archer* biru memiliki nilai ep yang paling tinggi sehingga perlu dibungkam agar tidak dapat melakukan aksi pada giliran selanjutnya sebagaimana yang telah dilakukan terhadap *wizard* merah.

c. Pengujian Karakter Medic

Karakter Medic adalah karakter yang memiliki atribut pertahanan lemah dan aksi yang mampu memulihkan hp karakter kawan atau diri sendiri. Peran utama karakter ini adalah untuk memulihkan hp karakter kawan yang telah berkurang akibat serangan musuh dengan aksi *heal*. Aksi *guard* dilakukan pada saat tidak ada karakter kawan yang memerlukan pemulihan hp. Hasil pengujian terhadap karakter *medic* ini ditunjukkan pada tabel 8 dan tabel 9.

Tabel 8 menunjukkan bahwa pada saat karakter *medic* merah melakukan aksi pertama dengan karakter kawan *knight* dan *archer*, aksi yang terpilih adalah *guard* terhadap diri sendiri dengan nilai $p = 10$. Aksi tersebut didahulukan atas aksi *guard* terhadap *archer* dengan nilai p yang sama karena pada iterasi program karakter *medic* merah dicek lebih awal. Dapat dilihat juga bahwa aksi *heal* memiliki prioritas yang sangat kecil atau pun nol terhadap karakter manapun dikarenakan hp setiap karakter masih penuh.

Tabel 9 menunjukkan bahwa pada situasi yang berbeda dengan karakter-karakter yang sama, aksi yang terpilih adalah *heal* terhadap *archer* merah dengan nilai $p = 50$. Aksi tersebut terpilih karena *archer* merah hampir mati dan memerlukan bantuan pemulihan hp.

d. Pengujian Karakter Wizard

Karakter Wizard adalah karakter yang memiliki atribut pertahanan lemah dan serangan mistis kuat. Peran utama karakter ini adalah untuk melakukan aksi *fire* terhadap karakter lawan yang lemah terhadap serangan mistis. Aksi *guard* hanya dilakukan pada saat karakter tidak memiliki ep yang cukup untuk melakukan serangan.

Pengujian terhadap karakter *wizard* dapat dilihat pada tabel 10 dan tabel 11.

Tabel 10 menunjukkan bahwa pada saat karakter *wizard* merah melakukan aksi pertama menghadapi musuh *knight*, *wizard* dan *archer*, aksi yang terpilih adalah *fire* terhadap *archer* biru dengan nilai $p = 46.7$. Sedangkan tabel 11 menunjukkan bahwa pada situasi yang berbeda dengan karakter-karakter yang sama, aksi yang terpilih adalah *fire* terhadap karakter *knight* biru dengan nilai $p = 58.3$. Aksi tersebut terpilih karena *knight* biru sudah hampir mati dengan nilai $hp = 60$, meskipun *damage* yang dihasilkan lebih kecil daripada aksi *fire* terhadap *archer* biru.

e. Pengujian Karakter Archer

Karakter Archer adalah karakter yang memiliki atribut pertahanan sedang dan serangan fisik kuat. Peran utama karakter ini adalah untuk melakukan aksi *shoot* terhadap karakter lawan yang lemah terhadap serangan fisik. Aksi *guard* hanya dilakukan pada saat karakter tidak memiliki ep yang cukup untuk melakukan serangan. Hasil pengujian dapat dilihat pada tabel 12 dan tabel 13.

Tabel 12 menunjukkan bahwa pada saat karakter *archer* merah melakukan aksi pertama menghadapi musuh *knight*, *wizard*, dan *archer*, aksi yang terpilih adalah *shoot*

terhadap *wizard* biru dengan nilai $p = 60$. Aksi tersebut memiliki prioritas tinggi karena menimbulkan *damage* yang paling tinggi.

Sedangkan pada situasi yang berbeda dengan karakter-karakter yang sama sebagaimana ditunjukkan tabel 13, aksi yang terpilih adalah *shoot* terhadap karakter *knight* biru dengan nilai $p = 61$. Aksi tersebut terpilih karena *knight* biru sudah hampir mati dengan nilai $hp = 40$, meskipun *damage* yang dihasilkan lebih kecil daripada aksi *shoot* terhadap *wizard* biru.

f. Analisis Pengujian

Berdasarkan hasil uji yang telah dijabarkan, dapat diamati bahwa sistem *fuzzy* yang dirancang dengan pendekatan *goal-driven* dapat menghasilkan nilai keluaran yang diinginkan. Untuk karakter-karakter penyerang, *knight*, *wizard*, dan *archer*, *behavior* yang diinginkan dapat dicapai dengan memastikan nilai p dari aksi serangan harus selalu lebih tinggi dari nilai p aksi *guard*. Selanjutnya agar aksi serangan tersebut, diarahkan terhadap target yang tepat, maka perlu keseimbangan pembobotan antara pengaruh variabel *damage* dan hp target. Pada *state* awal permainan, variabel *damage*-lah yang lebih berperan dalam menentukan pemilihan serangan, sebagaimana yang ditunjukkan pada Tabel 4, Tabel 10, dan Tabel 12. Sedangkan jika ada satu karakter musuh memiliki hp dengan nilai ekstrem rendah, variabel hp -lah yang lebih berperan dalam menentukan pemilihan serangan, sebagaimana yang ditunjukkan pada Tabel 5, Tabel 11, dan Tabel 13.

Untuk karakter *fighter*, nilai p aksi *punch* dipengaruhi oleh variabel *damage* dan ep . Dapat dilihat bahwa pada semua kasus variabel ep -lah yang lebih berperan dalam menentukan aksi yang terpilih sebagaimana yang ditunjukkan pada Tabel 4 dan Tabel 5. Variabel *damage* berperan sebagai pembeda pada kasus saat ada dua karakter yang memiliki nilai ep berdekatan Selanjutnya untuk karakter *medic* yang cenderung lebih sering menggunakan aksi *guard*, aksi *heal* perlu didesain agar nilai p -nya selalu lebih rendah daripada aksi *guard* pada saat karakter target masih memiliki hp penuh. Nilai p aksi *heal* akan naik secara drastis melampaui nilai p maksimum aksi *guard* jika ada karakter yang telah berkurang hp -nya.

4. Kesimpulan

Dari hasil uji kinerja sistem cerdas pengimplementasian *fuzzy* Tsukamoto didapatkan bahwa semua karakter telah menunjukkan *behavior* sesuai dengan perannya masing-masing. Semua karakter mampu memutuskan aksi yang tepat sesuai dengan kondisi lawan yang dihadapi. Dengan demikian diambil kesimpulan bahwa dengan diimplementasikannya logika Fuzzy Tsukamoto dihasilkan sistem kecerdasan buatan yang mampu menangani setiap kombinasi karakter dan setiap *range* situasi yang dapat ditemui pada jalannya permainan.

5. Daftar Pustaka

- [1] A. Drachen, M. Copier, M. Hitchens, M. Montola, M. P. Eladhari, and J. Stenros, "Role-Playing Games: The State of Knowledge," *Break. New Gr. Innov. Games, Play. Pract. Theory. Proc. DiGRA 2009*, 2009.
- [2] M. Hitchens, A. Drachen, and others, "The many faces of role-playing games," *Int. J. role-playing*, vol. 1, no. 1, pp. 3–21, 2008.
- [3] J. Pulkkinen, "Design Values of Digital Role-Playing Games," no. April, 2014.
- [4] J. H. Lee, N. Karlova, R. I. Clarke, and K. Thornton, "Facet Analysis of Video Game Genres," *iConference 2014 Proc.*, 2014.
- [5] A. S. S. L. H, and P. Gonz, "Game AI for a Turn-based Strategy Game with Plan Adaptation and AI Planning," *Proceeding DIMEA '08 Proc. 3rd Int. Conf. Digit. Interact. Media Entertain. Arts*, no. 0642882, pp. 295–302, 2008.
- [6] U. Kose, "Developing a Fuzzy Logic Based Game System," *Comput. Technol. Appl.*, vol. 3, no. Game Development, pp. 510–517, 2012.
- [7] M. Pirovano, I. Elettronica, and P. Milano, "The use of Fuzzy Logic for Artificial Intelligence in Games The current state of Game AI," 2012.
- [8] A. Shaout, B. King, and L. Reisner, "Real-Time Game Design of Pac-Man Using Fuzzy Logic," *Int. Arab J. Inf. Technol.*, vol. 3, no. 4, pp. 315–325, 2006.
- [9] Yifan Li, P. Musilek, and L. Wyard-Scott, "Fuzzy logic in agent-based game design," *IEEE Annu. Meet. Fuzzy Information, 2004. Process. NAFIPS '04.*, p. 734–739 Vol.2, 2004.
- [10] P. Tozour, "The Evolution of Game AI," *Ai Game Program. Wisdom*, vol. 1, pp. 3–15, 2002.
- [11] A. Saepullah and S. W. Romi, "Comparative Analysis of Mamdani, Sugeno And Tsukamoto Method of Fuzzy Inference System for Air Conditioner Energy Saving," *J. Intell. Syst.*, vol. 1, no. 2, pp. 143–147, 2015.
- [12] A. Maselena and M. Hasan, "Finding Kicking Range of Sepak Takraw Game A Fuzzy Logic Approach – Maselena – TELKOMNIKA Indonesian Journal of Electrical Engineering FT," vol. 14, no. 3, pp. 557–564, 2015.
- [13] Y. A. Gerhana *et al.*, "Decision support system for football player's position with tsukamoto fuzzy inference system," *Aasec*, vol. 03014, pp. 1–6, 2018.