IMPLEMENTASI LOGIKA *FUZZY* SEBAGAI PENGATUR KECEPATAN PEMBUKA PINTU PADA *ROLE PLAYING GAME BATTERY-MAN*

Yaya Abdul Somad[1]

[1] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : yayasomad@yahoo.co.id

ABSTRAK

Game Battery-man merupakan *game* berjenis *simulation game*. *Game Battery-man* ini memiliki konsep permainan pengatur kecepatan pembuka pintu untuk mengambil keputusan sangat lambat, lambat, sedang, cepat, dan sangat lambat sekali. Kecepatan pembuka pintu akan sangat menentukan kesuksesan pemain dalam *game* ini. Pengambilan baterai akan membutuhkan waktu, jika pemain hanya mengambil baterai dalam jumlah sedikit, maka waktu yang dibutuhkan juga sedikit. Sebaliknya, jika pemain mengambil baterai yang banyak, maka akan membutuhkan waktu yang cukup lama. Tetapi disisi lain, jika baterai tidak sesuai untuk membuka pintu maka kecepatan pintu akan sangat lambat, jika jumlah baterai sesuai dengan yang diperlukan oleh penggerak pintu, maka kecepatannya akan cepat.

Namun pada *game* simulasi *Battery – Man* membutuhkan suatu algoritma yang dapat menentukan nilai untuk 2 variabel yaitu tegangan dan arus. Berdasarkan penjelasan di atas maka dibutuhkan algoritma logika *fuzzy* ke dalam *game Battery-Man* sebagai pengatur kecepatan pembuka pintu untuk mengambil keputusan sangat lambat, lambat, sedang, cepat, dan sangat lambat sekali.

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa implementasi logika *fuzzy* dalam *game Battery-Man* ini telah memenuhi tujuan, yaitu dapat menghasilkan *output* berupa aksi-aksi sesuai dengan kondisinya seperti sangat lambat, lambat, sedang, cepat, sangat lambat sekali berdasarkan variable *fuzzy* tegangan, dan arus.

Kata Kunci: Logika *Fuzzy*, Tegangan, Arus.

1. PENDAHULUAN

1.1 Latar Belakang

Logika *fuzzy* merupakan suatu logika yang memiliki nilai kekaburan atau kesamaran (*fuzzyness*) antara benar atau salah. Namun berapa besar kebenaran dan kesalahan suatu nilai tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* memiliki derajat keanggotaan dalam rentang 0 hingga 1. Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* kedalam suatu ruang *output*, mempunyai nilai kontinyu. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama.[1]

AI *(Artificial Intelligence)* atau kecerdasan buatan adalah bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia [3]. Salah satu cabang dari kecerdasan buatan adalah logika *Fuzzy*. Logika *Fuzzy* digunakan untuk mengatasi masalah-masalah yang memiliki jawaban lebih dari satu. Karenanya logika *Fuzzy* lebih mirip dengan cara berfikir manusia ke dalam bentuk algoritma yang kemudian dijalankan oleh mesin. Penerapan logika *Fuzzy* pada *game Battery-Man* ini untuk mengatur kecepatan pembuka pintu. Berdasarkan karakteristik logika *fuzzy* dapat digunakan untuk permasalahan seperti menentukan pengatur kecepatan pembuka pintu dengan menggunakan variabel – variabel yang telah di tentukan. Adapun variabel masukan pada *game Battery-Man* ini adalah tegangan dan arus.

Game Battery-man merupakan *game* berjenis *simulation game*. *Game Battery-man* ini memiliki konsep permainan pengatur kecepatan pembuka pintu untuk mengambil keputusan sangat lambat, lambat, sedang, cepat, dan sangat lambat sekali. Kecepatan pembuka pintu akan sangat menentukan kesuksesan pemain dalam *game* ini. Pengambilan baterai akan membutuhkan waktu, jika pemain hanya mengambil baterai dalam jumlah sedikit, maka waktu yang dibutuhkan juga sedikit. Sebaliknya, jika pemain mengambil baterai yang banyak, maka akan membutuhkan waktu yang cukup lama. Tetapi disisi lain, jika baterai tidak sesuai untuk membuka pintu maka kecepatan pintu akan sangat lambat, jika jumlah baterai sesuai dengan yang diperlukan oleh penggerak pintu, maka kecepatannya akan cepat. Namun pada *game* simulasi *Battery – Man* membutuhkan suatu algoritma yang dapat menentukan nilai untuk 2

variabel yaitu tegangan dan arus. Berdasarkan penjelasan di atas maka dibutuhkan algoritma logika *fuzzy* ke dalam *game Battery-Man* sebagai pengatur kecepatan pembuka pintu untuk mengambil keputusan sangat lambat, lambat, sedang, cepat, dan sangat lambat sekali.

1.2 Maksud dan Tujuan

Berdasarkan permasalahan yang diteliti, maka maksud dari penulisan Tugas Akhir ini adalah mengimplementasikan Logika *Fuzzy* sebagai Pengatur Kecepatan Pembuka Pintu pada *Role Playing Game Battery-Man.*

Sedangkan tujuan yang akan dicapai dalam penelitian ini adalah kesesuaian Logika *Fuzzy* untuk mengatur kecepatan pembuka pintu untuk mengambil keputusan sangat lambat, lambat, sedang, cepat, dan sangat lambat sekali sesuai dengan masukan pemain dalam *game Battery-Man.*

.

2. ISI PENELITIAN

2.1 Teori

2.1.1 Logika *Fuzzy*

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

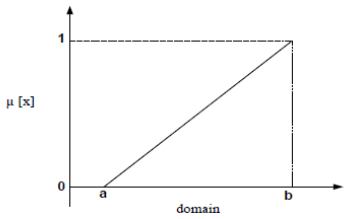
Logika *Fuzzy* merupakan seuatu logika yang memiliki nilai kekaburan atau kesamaran (fuzzyness) antara benar atau salah. Dalam teori logika fuzzy suatu nilai bias bernilai benar atau salah secara bersama. Namun berapa besar keberadaan dan kesalahan suatu tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* memiliki derajat keanggotaan dalam rentang 0 hingga 1. Berbeda dengan logika *digital* yang hanya memiliki dua nilai 1 atau 0. Logika *fuzzy* digunakan untuk menterjemahkan suatu besaran yang diekspresikan menggunakan bahasa (*linguistic*), misalkan besaran kecepatan laju kendaraan yang diekspresikan dengan pelan, agak cepat, cepat, dan sangat cepat. Dan logika *fuzzy* menunjukan sejauh mana suatu nilai itu benar dan sejauh mana suatu nilai itu salah. Tidak seperti logika klasik (*scrisp*)/ tegas, suatu nilai hanya mempunyai 2 kemungkinan yaitu merupakan suatu anggota himpunan atau tidak. Derajat keanggotaan 0 (nol) artinya nilai bukan merupakan anggota himpunan dan 1 (satu) berarti nilai tersebut adalah anggota himpunan.

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang output, mempunyai nilai kontinyu. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama (Kusumadewi. 2004) Logika Fuzzy memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keabuan dan juga hitam dan putih, dan dalam bentuk linguistik, konsep tidak pasti seperti "sedikit", "lumayan" dan "sangat" (Zadeh 1965). Kelebihan dari teori logika fuzzy adalah kemampuan dalam proses penalaran secara bahasa (*linguistic* reasoning). Sehingga dalam perancangannya tidak memerlukan persamaan matematik dari objek yang akan dikendalikan.[3]

1. Fungsi Keanggotaan

Fungsi Keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang digunakan, diantaranya adalah:

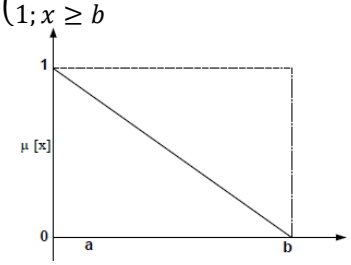
a. Representasi Linear



Gambar 2.1 Kurva Linear Naik [3]

Fungsi keanggotaan Linear Naik:

$$\mu[x] = \begin{cases} 0; x \le a \\ \frac{x-a}{b-a}; \quad a \le x \le b \\ 1; x \ge b \end{cases}$$

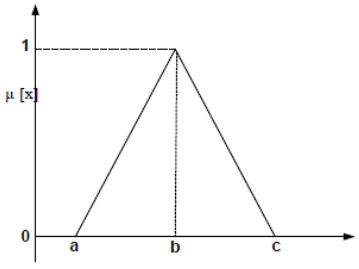


Gambar 2.2 Kurva Linear Turun [3]

Fungsi Keanggotaan Linear Turun:

$$\mu[x] = \begin{cases} \frac{b-x}{b-a}; a \le x \le b \\ 0; x \ge b \end{cases}$$

b. Representasi Segitiga

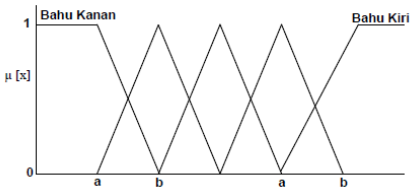


Gambar 2.3 Kurva Segitiga[3]

Fungsi keanggotaan kurva segitiga:

$$\mu[x] = \begin{cases} 0; x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}; a \leq x \leq b \\ \frac{b-x}{c-b}; b \leq x \leq c \end{cases}$$

c. Representasi Bahu



Gambar 2.4 Kurva bahu [3]

Fungsi keanggotaan bahu kanan:

$$\mu[x] = \begin{cases} 1; x \leq a \\ \frac{(b-x)}{(b-a)}; \ a \leq x \leq b` \\ 0; x \geq b \end{cases}$$

Fungsi keanggotaan bahu kiri:

$$\mu[x] = \begin{cases} 0; x \leq a \\ \frac{(x-a)}{(b-a)}; \ a \leq x \leq b \\ 1; x \geq b \end{cases}$$

2. Sistem Inferensi *Fuzzy* metode Tsukamoto

Pada Metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-Then harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan α-predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

Agregasi untuk menentukan keluaran semua *rules* dan dikombinasikan menjadi sebuah *fuzzy set* tunggal. Defuzzifikasi untuk menghasilkan nilai kesehatan menggunakan rata-rata terbobot.

$$z = \frac{\alpha 1 z 1 + \alpha 2 z 2 \ldots + \alpha n z n}{\alpha 1 + \alpha 2 \ldots + \alpha n}$$

1z = nilai crisp
α1 = derajat keanggotaan masing-masing variabel
z1 = nilai domain pada variabel linguistik z.

2.2 Analisi Algoritma dan Perancangan

2.2.1 Analisis Masukan

Berdasarkan definisi, logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*. Menghasilkan *input fuzzy* yang awalnya berdasarkan *crisp input*, *crisp input* adalah nilai masukan analog yang diberikan untuk mencari dan menghitung *degree of membership* atau derajat keanggotaan dengan cara proses *fuzzifikasi*.

Batasan *input* yang diberikan dalam merancang *fuzzy* sistem pada permainan *Battery-Man* berbeda-beda untuk menghasilkan *output* perilaku, berikut ini adalah variabel *Input* dan *Output* yang digunakan.

Tabel 3.2 Variabel *Output* dan *Input*

Output	Output/Input	Input
Kecepatan Pintu	Tegangan	Sangat Kecil
		Kecil
		Sedang
		Besar
		Sangat Besar
	Arus	Sangat Kecil
		Kecil
		Sedang
		Besar
		Sangat Besar

2.2.2.1 Analisis *Fuzzy* Tegangan

1. Variabel Tegangan, terdapat lima himpunan *fuzzy*, yaitu: Sangat kecil, kecil, sedang, besar, sangat besar. Semesta pembicara untuk variabel Tegangan adalah 0 – 100. Domain hipunan *fuzzy* pada variabel Tegangan, yaitu :
   1) Sangat Kecil : 0 – 40
   2) Kecil : 20 – 60
   3) Sedang : 40 – 80
   4) Besar : 60 – 100
   5) Sangat Besar : 80 – 100

2.2.2.2 Analisis *Fuzzy* Arus

1. Variabel Arus, terdapat lima himpunan *fuzzy*, yaitu: Sangat kecil, kecil, sedang, besar, sangat besar. Semesta pembicara untuk variabel Arus adalah 0 – 50. Domain hipunan *fuzzy* pada variabel Arus, yaitu :
   1) Sangat Kecil : 0 – 20
   2) Kecil : 10 – 30
   3) Sedang : 20 – 40
   4) Besar : 30 – 50
   5) Sangat Besar : 40 – 50

2.2.2.3 Analisis *Fuzzy* Kecepatan Pintu

1. Variabel Kecepatan Pintu terdapat lima himpunan fuzzy, yaitu: Sangat Lambat, Lambat, Sedang, Cepat, Sangat Lambat Sekali. Semesta pembicara untuk variable Arus adalah 0 – 120. Domain hipunan *fuzzy* pada variabel Arus, yaitu
   1) Sangat Lambat : 0 – 20
   2) Lambat : 10 – 30
   3) Sedang : 20 – 40
   4) Cepat : 30 – 50
   5) Sangat Lambat Sekali : 40 – 50

2.2.3 Implementasi Logika *Fuzzy*

2.2.3.1 Kasus Tegangan dan Arus

Misalkan didapatkan nilai Tegangan adalah 27,77, maka nilai 27,77 akan di konversi kedalam nilai fuzzy, dimana nilai Tegangan 27,77 berada pada nilai linguistik sangat rendah. Berdasarkan tahapan-tahapan atau proses agregasi yang menentukan keluaran sesuai dengan aturan *fuzzy*.

1. Tahapan Fuzzifikasi

Perhitungan *fuzzifikasi* untuk Tegangan dari setiap variabel linguistik (Sangat kecil, kecil, sedang, besar, sangat besar) apabila nilai tegangan [27,77] adalah sebagai berikut:

μSangatKecil [27,77]= $20 \leq 40 = 0$

μKecil [27,77]= $\frac{60-20}{40} = 1$

μSedag [27,77]= $\frac{80-40}{60} = 0{,}6$

μBesar [27,77]= $\frac{100-60}{80} = 0{,}5$

μSangatBesar [27,7]= $100 \geq 80 = 0$

2. Tahapan Pembentukan *Rule*

Berikut ini merupakan tahapan perancangan pembentukan *rule*.

Tabel 2.2 *Fuzzy Rule*

IF	Tegangan	AND	Arus	THEN	Kecepatan Pintu
IF	Sangat Kecil	AND	Sangat Kecil	THEN	Sangat Lambat
IF	Sangat Kecil	AND	Kecil	THEN	Sangat Lambat
IF	Sangat Kecil	AND	Sedang	THEN	Sangat Lambat
IF	Sangat Kecil	AND	Besar	THEN	Sangat Lambat
IF	Sangat Kecil	AND	Sangat Besar	THEN	Sangat Lambat
IF	Kecil	AND	Sangat Kecil	THEN	Lambat
IF	Kecil	AND	Kecil	THEN	Lambat
IF	Kecil	AND	Sedang	THEN	Lambat
IF	Kecil	AND	Besar	THEN	Sedang
IF	Kecil	AND	Sangat Besar	THEN	Sedang
IF	Sedang	AND	Sangat Kecil	THEN	Sedang
IF	Sedang	AND	Kecil	THEN	Sedang
IF	Sedang	AND	Sedang	THEN	Cepat
IF	Sedang	AND	Besar	THEN	Lambat
IF	Sedang	AND	Sangat Besar	THEN	Lambat
IF	Besar	AND	Sangat Kecil	THEN	Sangat Lambat
IF	Besar	AND	Kecil	THEN	Lambat
IF	Besar	AND	Sedang	THEN	Lambat
IF	Besar	AND	Besar	THEN	Lambat
IF	Besar	AND	Sangat Besar	THEN	Lambat
IF	Sangat Besar	AND	Sangat Kecil	THEN	Lambat
IF	Sangat Besar	AND	Kecil	THEN	Lambat
IF	Sangat Besar	AND	Sedang	THEN	Sangat Lambat Sekali
IF	Sangat Besar	AND	Besar	THEN	Sangat Lambat Sekali
IF	Sangat Besar	AND	Sangat Besar	THEN	Sangat Lambat Sekali

3. Tahapan Mesin Inferensi

Setelah nilai derajat keanggotaan di dapatkan, maka nilai-nilai derajat tersebut harus dibandingkan sesuai dengan *rule base* yang ada, tujuannya untuk mencari nilai Minimum untuk setiap *rule* yang dihitung.

1. [R1] *IF* Tegangan *is* Sangat Kecil *AND* Arus *is* Sangat Kecil *THEN* Kecepatan Pintu *is* Sangat Lambat
   α-pred$_1$ = μSangat Kecil[10] ∩ μSangat Kecil[8]
   = MIN (0 ; 0)
   = 0
2. [R2] *IF* Tegangan *is* Sangat Kecil *AND* Arus *is* Kecil *THEN* Kecepatan Pintu *is* Sangat Lambat
   α-pred$_2$ = μSangat Kecil[10] ∩ μKecil[8]
   = MIN (0 ; 1)
   = 0
3. [R3] *IF* Tegangan *is* Sangat Kecil *AND* Arus *is* Sedang *THEN* Kecepatan Pintu *is* Sangat Lambat
   α-pred$_3$ = μSangat Kecil[10] ∩μSedang[8]
   = MIN (0 ; 0,6)
   = 0
4. [R4] *IF* Tegangan *is* Sangat Kecil *AND* Arus *is* Besar *THEN* Kecepatan Pintu *is* Sangat Lambat
   α-pred$_4$ = μSangatRenda[10] ∩ μKuat[8]
   = MIN (0 ; 0,5)
   = 0
5. [R5] *IF* Tegangan *is* Sangat Kecil *AND* Arus *is* Sangat Besar *THEN* Kecepatan Pintu *is* Sangat Lambat
   α-pred$_5$ = μSangat Kecil[10] ∩ μSangatBesar[8]
   = MIN (0 ; 0)
   = 0
6. [R6] *IF* Tegangan *is* Kecil *AND* Arus *is* Sangat Kecil *THEN* Kecepatan Pintu *is* Lambat
   α-pred$_6$ = μKecil[10] ∩ μSangatKecil[8]
   = MIN (1 ; 0)
   = 1
7. [R7] *IF* Tegangan *is* Kecil *AND* Arus *is* Kecil *THEN* Kecepatan Pintu *is* Lambat
   α-pred$_7$ = μKecil[10] ∩ μKecil[8]
   = MIN (1 ; 1)
   = 1
8. [R8] *IF* Tegangan *is* Kecil *AND* Arus *is* Sedang *THEN* Kecepatan Pintu *is* Lambat
   α-pred$_8$ = μKecil[10] ∩ μSedang[8]
   = MIN (1 ; 0,6)
   = 1
9. [R9] *IF* Tegangan *is* Kecil *AND* Arus *is* Besar *THEN* Kecepatan Pintu *is* Sedang
   α-pred$_9$ = μKecil[10] ∩ μBesar[8]
   = MIN (1 ; 0,5)
   = 1
10. [R10] *IF* Tegangan *is* Kecil *AND* Arus *is* Sangat Besar *THEN* Kecepatan Pintu *is* Sedang
    α-pred$_{10}$= μKecil[10] ∩ μSangatBesar[8]
    = MIN (1 ; 0)
    = 1
11. [R11] *IF* Tegangan *is* Sedang *AND* Arus *is* Sangat Kecil *THEN* Kecepatan Pintu *is* Sedang
    α-pred$_{11}$ = μSedang[10] ∩ μSangatKecil[8]
    = MIN (0,6 ; 0)
    = 0,6

12. [R12] *IF* Tegangan *is* Sedang *AND* Arus *is* Kecil *THEN* Kecepatan Pintu *is* Sedang
    α-pred$_{12}$ = µSedang[10] ∩ µKecil[8]
    = MIN (0,6 ; 1)
    = 0,6
13. [R13] *IF* Tegangan *is* Sedang *AND* Arus *is* Sedang *THEN* Kecepatan Pintu *is* Cepat
    α-pred$_{13}$ = µSedang[10] ∩ µSedang[8]
    = MIN (0,6 ; 0,6)
    = 0,6
14. [R14] *IF* Tegangan *is* Sedang *AND* Arus *is* Besar *THEN* Kecepatan Pintu *is* Lambat
    α-pred$_{14}$ = µSedang[10] ∩ µBesar[8]
    = MIN (0,6 ; 0,5)
    = 0,6
15. [R15] *IF* Tegangan *is* Sedang *AND* Arus *is* Sangat Besar *THEN* Kecepatan Pintu *is* Lambat
    α-pred$_{15}$ = µSedang[10] ∩ µSangaBesar[8]
    = MIN (0,6 ; 0)
    = 0,6
16. [R16] *IF* Tegangan *is* Besar *AND* Arus *is* Sangat Kecil *THEN* Kecepatan Pintu *is* Sangat Lambat
    α-pred$_{16}$ = µBesar[10] ∩ µSangaKecil[8]
    = MIN (0,5 ; 0)
    = 0,5
17. [R17] *IF* Tegangan *is* Besar *AND* Arus *is* Kecil *THEN* Kecepatan Pintu *is* Lambat
    α-pred$_{17}$ = µBesar[10] ∩ µKecil[8]
    = MIN (0,5 ; 1)
    = 0,5
18. [R18] *IF* Tegangan *is* Besar *AND* Arus *is* Sedang *THEN* Kecepatan Pintu *is* Lambat
    α-pred$_{18}$ = µBesar[10] ∩ µSedang[8]
    = MIN (0,5 ; 0,6)
    = 0,5
19. [R19] *IF* Tegangan *is* Besar *AND* Arus *is* Besar *THEN* Kecepatan Pintu *is* Lambat
    α-pred$_{19}$ = µBesar[10] ∩ µBesar[8]
    = MIN (0,5 ; 0,5)
    = 0,5
20. [R20]*IF* Tegangan *is* Besar *AND* Arus *is* Sangat Besar *THEN* Kecepatan Pintu *is* Lambat
    α-pred$_{20}$ = µBesar[10] ∩ µSangatBesar[8]
    = MIN (0,5 ; 0)
    = 0,5
21. [R21] *IF* Tegangan *is* Sangat Besar *AND* Arus *is* Sangat Kecil *THEN* Kecepatan Pintu *is* Lambat
    α-pred$_{21}$ = µSangatBesar[10] ∩ µSangaKecil[8]
    = MIN (0 ; 0)
    = 0
22. [R22] *IF* Tegangan *is* Sangat Besar *AND* Arus *is* Kecil *THEN* Kecepatan Pintu *is* Lambat
    α-pred$_{22}$= µSangatBesar[10] ∩ µKecil[8]
    = MIN (0 ; 1)
    = 0
23. [R23] *IF* Tegangan *is* Sangat Besar *AND* Arus *is* Sedang *THEN* Kecepatan Pintu *is* Sangat Lambat Sekali
    α-pred$_{23}$ = µSangatBesar[10] ∩ µSedang[8]
    = MIN (0 ; 0,6)
    = 0
24. [R24] *IF* Tegangan *is* Sangat Besar *AND* Arus *is* Besar *THEN* Kecepatan Pintu *is* Sangat Lambat Sekali
    α-pred$_{24}$ = µSangatBesar[10] ∩ µBesar[8]
    = MIN (0 ; 0,5)
    = 0
25. [R25] *IF* Tegangan *is* Sangat Besar *AND* Arus *is* Sangat Besar *THEN* Kecepatan Pintu *is* Sangat lambat sekali
    α-pred$_{25}$ = µSangatBesar[10] ∩ µSangaBesar[8]
    = MIN (0 ; 0)
    = 0

4. Tahapan Defuzzifikasi

1. Rule 1 = Sangat Lambat $\frac{10-z}{10} = 0$
   Z1 = 10 – 0 = 10
2. Rule 2 = Sangat Lambat $\frac{10-z}{10} = 0$
   Z1 = 10 – 0 = 10
3. Rule 3 = Sangat Lambat $\frac{10-z}{10} = 0$
   Z1 = 10 - 0 = 10
4. Rule 4 = Sangat Lambat $\frac{10-z}{10} = 0$
   Z1 = 10 – 0 = 10
5. Rule 5 = Sangat Lambat $\frac{10-z}{10} = 0$
   Z1 = 10 – 0 = 10
6. Rule 6 = Lambat $\frac{z-10}{10} = 1$
   Z1 = 10 – 10 = 0
7. Rule 7 = Lambat $\frac{z-10}{10} = 1$
   Z1 = 10 – 10 = 0
8. Rule 8 = Lambat $\frac{z-10}{10} = 1$
   Z1 = 10 – 10 = 0
9. Rule 9 = Sedang $\frac{z-20}{10} = 1$
   Z1 = 20 – 20 = 0
10. Rule 10 = Sedang $\frac{z-20}{10} = 1$
    Z1 = 20 – 20 = 30
11. Rule 11 = Sedang $\frac{z-20}{10} = 0{,}6$
    Z1 = 12 + 20 = 32
12. Rule 12 = Sedang $\frac{z-20}{10} = 0{,}6$
    Z1 = 12 + 20 = 32
13. Rule 13 = Cepat $\frac{z-30}{10} = 0{,}6$
    Z1 = 18 + 20 = 38
14. Rule 14 = Lambat $\frac{z-20}{10} = 0{,}6$
    Z1 = 12 + 20 = 32

15. Rule 15 = Lambat $\frac{z-20}{10} = 0,6$
    Z1 = 12 + 20 = 32
16. Rule 16 = Sangat Lambat $\frac{z-40}{10} = 0,5$
    Z1 = 20 + 40 = 60
17. Rule 17 = Lambat $\frac{z-10}{10} = 0,5$
    Z1 = 5 + 10 = 15
18. Rule 18 = Lambat $\frac{z-10}{10} = 0,5$
    Z1 = 5 – 10 = 15
19. Rule 19 = Lambat $\frac{z-10}{20} = 0,5$
    Z1 = 5 – 10 = 15
20. Rule 20 = Lambat $\frac{z-10}{10} = 0,5$
    Z1 = 1 + 80 = 15
21. Rule 21 = Lambat $\frac{z-40}{10} = 0$
    Z1 = 0 + 40 = 40
22. Rule 22 = Lambat $\frac{z-40}{10} = 0$
    Z1 = 0 + 40 = 40
23. Rule 23 = Sangat Lambat Sekali $\frac{z-40}{10} = 0$
    Z1 = 0 + 40 = 40
24. Rule 24 = Sangat Lambat Sekali $\frac{z-40}{10} = 0$
    Z1 =0 + 40 = 40
25. Rule 25 = Sangat Lambat Sekali $\frac{z-40}{10} = 0$
    Z1 = 0 + 40 = 40

$$Z = \frac{0 \times 10 + 0 \times 10 + 0 \times 10 + 0 \times 10 + 0 \times 10 + 1 \times 10 + 1 \times 10 + 1 \times 10 + 1 \times 10 + 1 \times 10 + 0,6 \times 32 + 0,6 \times 32 + 0,6 \times 38 + 0,6 \times 32 + 0,6 \times 32 + 0,5 \times 60 + 0,5 \times 15 + 0,5 \times 15 + 0,5 \times 15 + 0,5 \times 15 + 0 \times 40 + 0 \times 40 + 0 \times 40 + 0 \times 40 + 0 \times 40}{0 + 0 + 0 + 0 + 0 + 1 + 1 + 1 + 1 + 1 + 0,6 + 0,6 + 0,6 + 0,6 + 0,6 + 0,5 + 0,5 + 0,5 + 0,5 + 0,5 + 0 + 0 + 0 + 0 + 0}$$

$$= \frac{0 + 0 + 0 + 0 + 0 + 10 + 10 + 10 + 10 + 10 + 19,2 + 19,2 + 22,8 + 19,2 + 19,2 + 30 + 7,5 + 7,5 + 7,5 + 7,5 + 0 + 0 + 0 + 0 + 0}{10,5}$$

$$= \frac{202,1}{10,5} = 19,72$$

Kesimpulan :

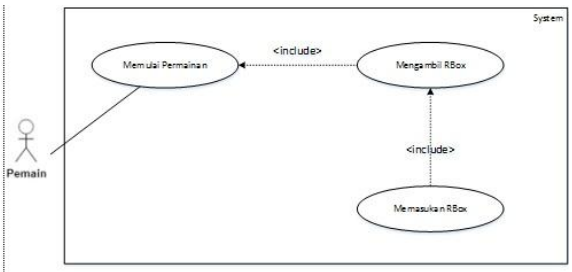
Nilai yang didapat dari proses *defuzzyfication* tersebut selanjutnya digunakan untuk penentuan keputusan dengan aturan sebagai berikut :

1. Sangat Lambat : 0 – 20
2. Lambat : 10 – 30
3. Sedang : 20 – 40
4. Cepat : 30 – 50
5. Sangat Lambat Sekali : 40 – 50

Maka hasil nilai dari 19,72 adalah Lambat.

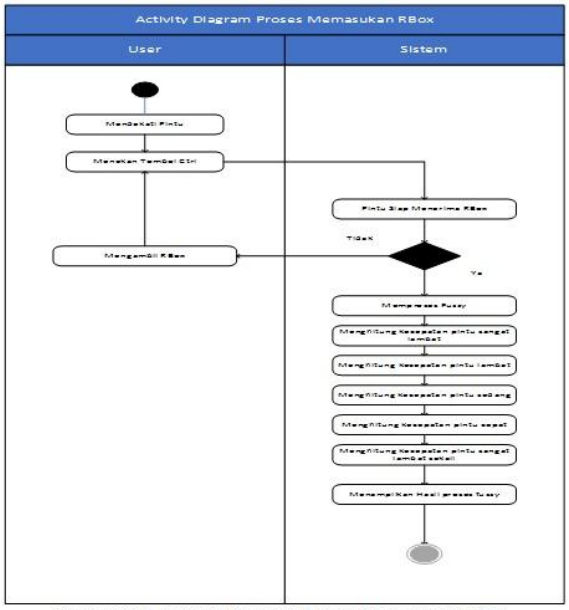
2.2.4 Analisis Kebutuhan Fungsional

2.2.4.1 Use Case



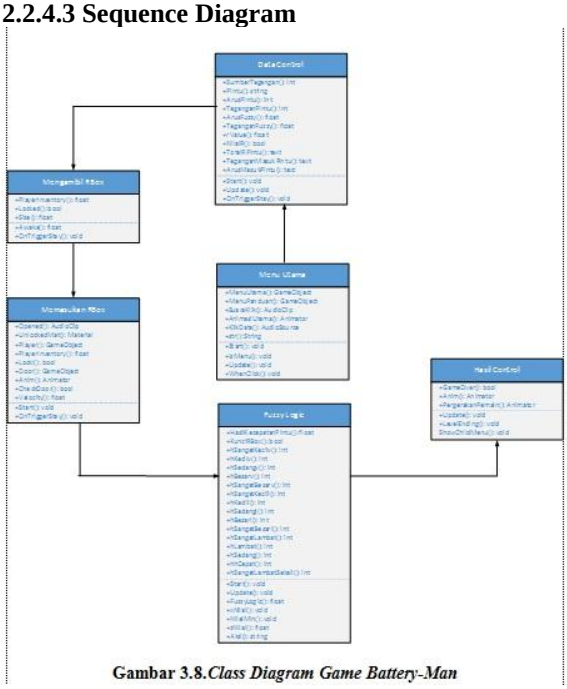
Gambar 3.4.*Use Case Diagram Game Battery-Man*

2.2.4.2 Activity Diagram



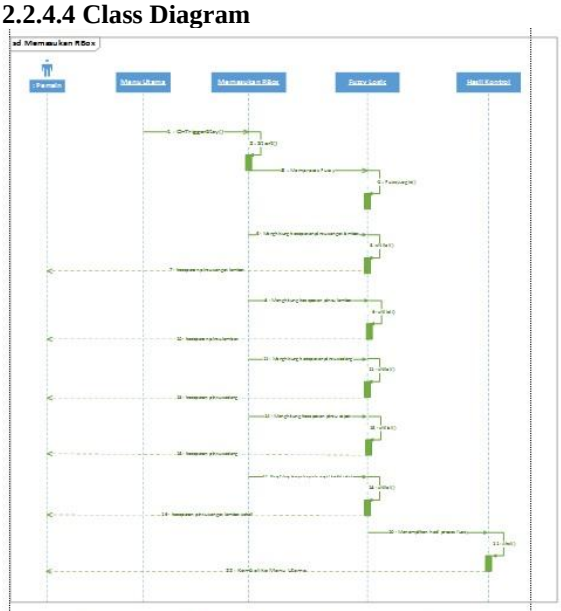
Gambar 3.7. *Activity Diagram* Proses Memasukan RBox

2.2.4.3 Sequence Diagram



Gambar 3.8.*Class Diagram Game Battery-Man*

2.2.4.4 Class Diagram



Gambar 3.11. *Diagram Sequence* Memasukan RBox

2.3 Implementasi dan Pengujian

2.3.1 Pengujian Sistem

Pengujian adalah hal yang perlu dilakukan untuk mengetahui kekurangan atau kesalahan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

2.3.2 Rencana Pengujian

Pada bagian rencana pengujian, digunakan rencana pengujian fungsionalitas yaitu dengan menggunakan pengujian *black box* yang berfokus pada persyaratan fungsional perangkat lunak serta pengujian simulsi yang ditujukan pada tingkat kesesuaian nilai defuzzifikasi dengan perilaku yang sudah ditetapkan. Berikut adalah rencana pengujian yang dilakukan pada tabel 4.3 :

Tabel 4.3 Pengujian *Black Box*

Tabel 4.3 Pengujian *Black Box*

No	Komponen yang di uji	Jenis Pengujian
1.	Menu Utama	Black Box
2.	Mulai Permainan	Black Box
3.	Panduan Game	Black Box
4.	Hasil Permainan	Black Box

2.3.3 Hasil Pengujian

Hasil pengujian didapat dari pengujian algoritma logika fuzzy dan *black box*. Kesesuaian item mana saja yang akan menggunakan pengujian algoritma logika fuzzy dan *black box* dilakukan sesuai dengan rencana pengujian.

2.3.3.1 Hasil Pengujian Black Box

Berikut ini adalah hasil dari pengujian perangkat lunak yang sudah dibangun menggunakan metode *black box* berdasarkan pengamatan yang sudah dilakukan dan menghasilkan kesimpulan dari pengujian tersebut.

1. Pengujian Menu Utama

Pengujian menu utama dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.4 :

Tabel 4.4 Pengujian Menu Utama

Kasus yang akan di uji	Hasil Uji	
	Hasil yang diharapkan	Kesimpulan
Menu Utama	Sistem menampilkan menu permainan	[√] Berhasil [ ] Tidak Behasil

2. Pengujian Mulai Permainan

Pengujian mulai permainan dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.5 :

Tabel 4.5 Pengujian Mulai Permainan

Kasus yang akan di uji	Hasil Uji	
	Hasil yang diharapkan	Kesimpulan
Memulai Permainan	Sistem menampilkan layar permainan	[√] Berhasil [ ] Tidak Behasil

3. Pengujian Panduan Game

Pengujian panduan game dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.6 :

Tabel 4.6 Pengujian Panduan Game

Kasus yang akan di uji	Hasil Uji	
	Hasil yang diharapkan	Kesimpulan
Panduan Game	Sistem menampilkan panduan game	[√] Berhasil [ ] Tidak Behasil

4. Pengujian Hasil Permainan

Pengujian hasil permainan dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.7 :

Tabel 4.7 Pengujian Hasil Permainan

Kasus yang akan di uji	Hasil Uji	
	Hasil yang diharapkan	Kesimpulan
Hasil Permainan	Sistem menampilkan hasil permainan	[√] Berhasil [ ] Tidak Behasil

2.3.3.2 Hasil Pengujian Simulasi

Pengujian simulasi untuk memastikan *rule-rule* yang dibuat dapat berjalan dengan benar. Pengujian yang dilakukan adalah pengujian pada variabel *input* tegangan dan arus, variabel *output* kecepatan pintu karena pada bagian inilah terdapat *rule-rule* yang sudah dibuat berdasarkan aturan Logika *Fuzzy*. Setiap bagian dari *rule-rule* Logika

Fuzzy ini akan diuji kebenarannya, apakah program aplikasi yang sudah dibuat dapat berjalan dengan baik atau tidak. *Rule* yang akan diuji berjumlah 25 jenis *rule*, dan 5 jenis kecepatan pintu. Pengujian dilakukan oleh karakter utama untuk mengimplementasi algoritma logika *fuzzy*.

1. Pengujian Rule ke 1

Berikut ini adalah hasil pengujian yang telah dilakukan pada *rule* 1 untuk kecepatan pintu, dapat dilihat pada Tabel 4.8 dan gambar 4.5 di bawah ini.

Tabel 4.8 Hasil Pengujian *Rule* 1

		Tegangan		Arus		Kecepatan Pintu
Rule 1	IF	Sangat Kecil	AND	Sangat Kecil	THEN	Sangat Lambat
Hasil Pengujian						
No	Parameter		Nilai	Nilai z	Kecepatan Pintu	
1	Tegangan		21.73913	14.18679	Sangat Lambat	
2	Arus		10.86957			



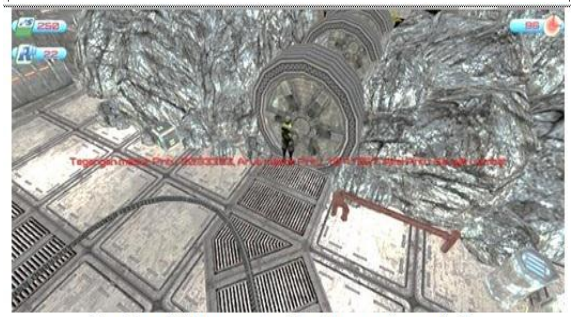
Gambar 4.5 Representasi hasil pengujian *rule* 1

2. Pengujian Rule k2

Berikut ini adalah hasil pengujian yang telah dilakukan pada *rule* 2 untuk kecepatan pintu, dapat dilihat pada Tabel 4.9 dan gambar 4.6 di bawah ini.

Tabel 4.9 Hasil Pengujian *Rule* 2

		Tegangan		Arus		Kecepatan Pintu
Rule 2	IF	Sangat Kecil	AND	Kecil	THEN	Sangat Lambat
Hasil Pengujian						
No	Parameter		Nilai	Nilai z	Kecepatan Pintu	
1	Tegangan		20.83333	12.24359	Sangat Lambat	
2	Arus		10.41667			



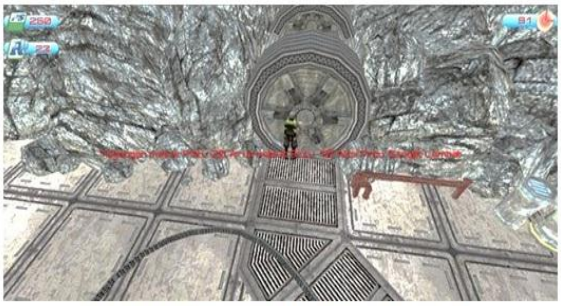
Gambar 4.6 Representasi hasil pengujian *rule* 2

3. Pengujian Rule ke 3

Berikut ini adalah hasil pengujian yang telah dilakukan pada *rule* 3 untuk kecepatan pintu, dapat dilihat pada Tabel 4.10 dan gambar 4.7 di bawah ini.

Tabel 4.10 Hasil Pengujian *Rule* 3

		Tegangan		Arus		Kecepatan Pintu
Rule 3	IF	Sangat Kecil	AND	Sedang	THEN	Sangat Lambat
Hasil Pengujian						
No	Parameter		Nilai	Nilai z	Kecepatan Pintu	
1	Tegangan		20	10	Sangat Lambat	
2	Arus		10			



Gambar 4.7 Representasi hasil pengujian *rule* 3

2.3.4 Kesimpulan Pengujian

Setelah dilakukan pengujian maka di dapatkan kesimpulan bahwa rule – rule yang di uji telah sesuai dengan perancangan dan logika *fuzzy* yang dapat menghasilkan keputusan dengan baik dan tepat.

3.PENUTUP

3.1 Kesimpulan

Berdasarkan hasil yang didapatkan dalam penelitian dan penyusunan skripsi ini serta disesuaikan dengan tujuan, maka diperoleh kesimpulan bahwa implementasi logika *fuzzy* dalam *game Battery-Man* ini telah memenuhi tujuan, yaitu dapat menghasilkan *output* berupa aksi-aksi sesuai dengan kondisinya seperti sangat lambat, lambat, sedang, cepat, sangat lambat sekali berdasarkan variabel *fuzzy* tegangan dan arus.

3.2 Saran

Berdasarkan hasil dari keseluruhan implementasi dan pengujian, maka dapat diberikan saran-saran yaitu penambahan parameter dan perilaku dalam algoritma *Logika Fuzzy Tsukamoto* dalam menentukan perilaku yang tepat dalam kecepatan pintu dan diharapkan dapat dilakukan pembulatan nilai *linguistic* agar ketepatan nilai defuzzifikasi antara proses dalam aplikasi sesuai dengan perhitungan manual.

DAFTAR PUSTAKA

[1] Kusumadewi, S. 2003. *Artificial Interlligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.

[2] Nasution. H. 2012. *Jurnal Implementasi Logika Fuzzy Pada Sistem Kecerdasan Buatan.* Universitas Tanjungpura.

[3] Widhiastiwi, Y. 2007. *Model Fuzzy Dengan Metode Tsukamoto.* Bina Widia, Vol. 18 No. 02.

[4] Rollings, Andrew; Morris, Dave. 2004. *Game Architecture and Design: A New Edition*. New Riders.

[5] Marsic. I. 2012. *Software Engineering*. Rutgers University.