

## AGEN CERDAS BERBASIS CONTROLLER FUZZY PADA PERMAINAN STRATEGI PERTEMPURAN DENGAN BEHAVIOR TREE

Edwin Meinardi Trianto, Hartarto Junaedi, Hari Sutiksno  
Institut Informatika Indonesia (IKADO) , Sekolah Tinggi Teknik Surabaya (STTS)  
Surabaya, Indonesia  
[edwinmeinardi@gmail.com](mailto:edwinmeinardi@gmail.com), [aikawa@gmail.com](mailto:aikawa@gmail.com), [hari@stts.edu](mailto:hari@stts.edu)

### ABSTRAK

Perkembangan permainan strategi sekarang sudah sangat pesat. Salah satu permasalahan utama dalam genre permainan tersebut adalah bagaimana sebuah NPC (non playable character) dapat bermain dengan menggunakan strategi khusus untuk dapat memenangkan permainan. Untuk dapat membuat kecerdasan buatan pada NPC akan diterapkan logika *fuzzy* yang memudahkan NPC untuk menentukan action apa yang akan dijalankan. Action yang dipilih oleh NPC tersebut berupa behavior tree. Behavior tree tersebut dirancang untuk membuat NPC seolah olah memiliki behavior tertentu untuk sebuah kondisi tertentu. Behavior yang dijalankan akan ditentukan oleh fuzzy controller yang memiliki model fuzzy zero – order sugeno. Ketika diterapkan pada skema permainan, tingkat kemenangan NPC ketika menggunakan fuzzy controller dan behavior tree akan lebih besar dari pada sebuah NPC tanpa fuzzy controller yang hanya menjalankan 1 action saja.

**Kata kunci :** permainan, NPC, kecerdasan buatan, fuzzy controller, behavior tree.

### 1. PENDAHULUAN

Perkembangan permainan strategi yang menggunakan kecerdasan buatan sudah pesat. Salah satu permasalahan dalam perkembangan tersebut adalah bagaimana membuat sebuah NPC (Non Playable Character) yang dapat bermain dengan menggunakan strategi khusus yang dapat memenangkan permainan dengan baik.

Pada penelitian ini akan dikaji bagaimana membuat sebuah kecerdasan buatan untuk NPC dan diterapkan pada permainan. Setiap NPC pada permainan ini akan diterapkan logika fuzzy. Logika fuzzy merupakan metode atau algoritma untuk menerapkan kecerdasan buatan yang diterapkan pada kasus pengambilan keputusan. Logika fuzzy sendiri merupakan suatu metode yang memiliki nilai kesamaran atau kekaburan (fuzziness) antara benar dan salah. Namun penilaian seberapa besar suatu nilai atau kesalahan tergantung pada bobot keanggotaan yang dimilikinya. Dalam penerapannya logika fuzzy merupakan suatu cara untuk memetakan permasalahan dari input menuju output yang diharapkan terutama dalam sistem pengambilan keputusan. Logika fuzzy sendiri mampu beradaptasi dengan perubahan-perubahan, dan ketidakpastian yang menyertai permasalahan. Sebagai contoh jika menggunakan logika fuzzy, NPC pada permainan ini akan memiliki sifat seperti

prajurit pada umumnya, setiap prajurit akan memiliki atribut seperti health yang berfungsi sebagai ketahanan, stamina yang berfungsi sebagai kekuatan untuk melakukan serangan tertentu, tingkat aggressive untuk membuat prajurit itu cenderung menyerang, dan sebagainya. Sebagai contoh jika HP (Hit Point / Health) pada NPC tergolong tinggi, maka NPC tersebut akan lebih agresif (lebih cenderung menyerang), jika HP pada NPC tergolong rendah, Maka NPC tersebut cenderung melarikan diri (dalam kasus ini adalah beristirahat untuk memulihkan HP).

Sampai saat ini sudah banyak sekali metode yang digunakan untuk merancang sebuah smart agent dalam sebuah permainan, salah satunya adalah Behavior Tree. Behavior Tree adalah salah satu metode pemetaan perilaku agen cerdas dasar yang menggunakan task sebagai elemen penyusunnya. Behavior Tree sangat populer digunakan dalam industri game, beberapa permainan seperti Halo 2 dan Crysis menggunakan Behavior Tree dalam perancangan agen cerdasnya.

Pada penelitian ini akan menggunakan kasus permainan strategi dimana permainan ini terdapat 3 kelas utama yaitu: 1.Swordman, 2.Archer, 3.Magician. Swordman adalah kelas yang memiliki HP dan stamina yang paling besar diantara 3 kelas, serta dampak serangan yang kecil. Archer adalah

kelas yang memiliki HP dan stamina yang sedang, dan archer merupakan kelas yang serangannya lebih besar dari swordman. Magician adalah kelas yang memiliki HP dan stamina yang paling kecil tetapi memiliki serangan yang paling besar. Setiap kelas memiliki behavior state dimana setiap state pada setiap kelas akan bervariasi yang membuat permainan strategi ini lebih menarik. Pada setiap prajurit pada setiap kelas akan memiliki 4 atribut dengan nilai yang berbeda – beda, atribut tersebut adalah health, stamina, tingkat aggressive, dan tingkat agility.

## 2. RUANG LINGKUP

Berikut akan dijelaskan detail spesifikasi sebagai dasar penelitian.

### A. Fuzzy Logic

Logika fuzzy adalah salah satu teknik matematika yang dirancang untuk mengatasi daerah bias [2]. Misalnya mobil npc yang bergerak di trafik yang padat, pada behavior yang digunakan, peralihan antar behavior terjadi jika tidak ada mobil lain yang dekat dengan mobil NPC, kemudian mobil NPC akan bergerak. Jika terdapat mobil yang dekat, maka mobil NPC berhenti. Sistem fuzzy memungkinkan kelancaran transisi antar behavior. Sebagai contoh lain, mobil NPC bisa mengurangi kecepatan saat ada mobil lain pada jarak tertentu dengan mobil NPC lainnya.

Dalam penelitian ini, logika fuzzy digunakan untuk memilih perilaku serangan masing-masing NPC berdasarkan health (kesehatannya atau ketahanan), stamina (kekuatan untuk melakukan sesuatu), tingkat aggressive (membuat prajurit itu cenderung menyerang), dan tingkat agility (membuat prajurit itu semakin cepat dalam bergerak). Model zero-order sugeno digunakan dalam penelitian ini, sebuah aturan khas dalam model fuzzy Sugeno memiliki bentuk:

IF Input 1 = x AND Input 2 = y, then Output is z = ax + by + c.

Untuk model zero-order Sugeno, output dari z bernilai tetap : a = b = 0.

### B. Behavior Tree

Behavior tree (BTs) dikembangkan untuk para komunitas gaming sebagai alternatif modular untuk Finite State Machine (FMS). Permasalahan utama dari FMS adalah ketika terjadi penambahan state

atau perubahan transisi. Bahkan setiap state pada FMS tidak dapat digunakan kembali dengan mudah. Kekurangan FMS adalah sulitnya menerapkan konsep modularity. Behavior tree dengan mudah menutupi kekurangan FMS dengan menggunakan konsep modularitas, sehingga setiap state membentuk struktur cabang seperti pohon. Setiap behavior akan dijalankan melalui root menuju ke cabang paling akhir (leaf) dan transversal pada pohon ini bisa sangat dinamis.

### C. Spesifikasi Permainan

Permainan strategi yang digunakan memiliki genre close combat. Detail permainan ini memiliki 2 kelompok yaitu kelompok 1 NPC melawan kelompok 2 NPC. Berikut ini adalah aturan permainannya:

1. Setiap kelompok NPC terdapat dan 30 anggota (prajurit), yang terbagi sama besar untuk 3 kelas.
2. Untuk memvisualisasikan permainan, terdapat matrix 20x20 sebagai lahan untuk posisi dan pergerakan dari setiap prajurit serta detail jumlah sisa pasukan setiap kelompok, detail rata – rata total keseluruhan health pasukan, dan detail rata – rata total keseluruhan stamina pasukan yang dapat ditampilkan setiap updatenya.
3. Setiap prajurit memiliki atribut
  - a. Health : menandakan ketahanan seorang prajurit, jika health mencapai 0, maka prajurit tersebut dinyatakan mati.
  - b. Stamina : menandakan kekuatan untuk melakukan special action (running attack dan running defense)
  - c. Agility : menandakan kegesitan atau kecepatan seorang prajurit untuk melangkah (standar 1 langkah untuk 1 unit pixel)
  - d. Aggressive : menandakan seberapa besar keinginan seorang prajurit untuk menyerang, semakin kecil tingkat keagresifan seorang prajurit, maka semakin sering prajurit itu bertahan.
4. Nilai setiap atribut dari seorang prajurit akan diacak, setiap prajurit pada kelas tertentu akan memiliki health dan stamina berdasarkan kelasnya.

5. Setiap babak dalam permainan pada setiap prajurit kelompok akan memberikan action pada setiap prajurit yang terdapat pada kelompok lawan dan akan mengurangi health pada setiap prajurit pada kelompok lawan.
6. Babak akan berakhir ketika semua pasukan pada salah satu kelompok habis.

Berikut adalah detil pada spesifikasi permainan yang disajikan dalam tabel 1

TABEL I. DETIL SPESIFIKASI PERMAINAN

Kelas	Swordman	Archer	Magician
Health	150	100	60
Stamina	100	80	60
Normal Attack	Dampak 15 poin	Dampak 20 poin	Dampak 25 poin
Jarak Serang	1,5m – 2m	3m – 3,5m	2m – 2,5m
Agility	0,5 – 2,5 poin	0,5 – 2,5 poin	0,5 – 2,5 poin
Attack Delay	1.25 - 2.25 poin	1.25 - 2.25 poin	1.25 - 2.25 poin
Aggressive	0,5 poin – 1 poin	0,5 poin – 1 poin	0,5 poin – 1 poin
Damage Reduction Defense	7 poin	5 poin	2 poin
Damage Reduction Running Defense	5 poin	3 poin	1 poin
Health Regeneration Defense	0,1 poin per update	0,05 poin per update	0,2 poin per update
Health Regeneration Running	0,3 poin per update	0,15 poin per update	0,6 poin per update
Health Regeneration Running & Defense	0,05 poin per update	0,005 poin per update	0,1 poin per update
Stamina Regeneration Defense	2.5 poin per 3 update	1.25 poin per 3 update	5 poin per 3 update
Stamina Regeneration Running	5 poin per 3 update	2.5 poin per 3 update	10 poin per 3 update

Stamina Regeneration Running & Defense	7.5 poin per 3 update	4 poin per 3 update	15 poin per 3 update
--	-----------------------	---------------------	----------------------

### 3. IMPLEMENTASI DAN DESAIN

Pada bab ini akan dijelaskan penerapan dan penjelasan pada penelitian ini.

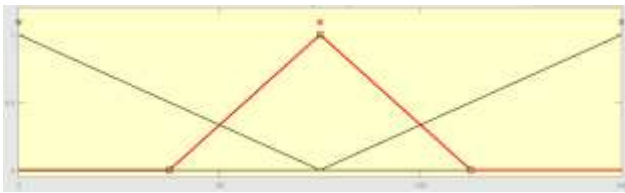
#### A. NPC Fuzzy Controller

Untuk membuat NPC dapat mengambil keputusan yang tepat untuk pemilihan behavior maka setiap anggota pada kelompok akan diterapkan logika fuzzy untuk menghasilkan pemilihan keputusan terhadap behavior itu sendiri. Input dari fuzzy controller ini adalah

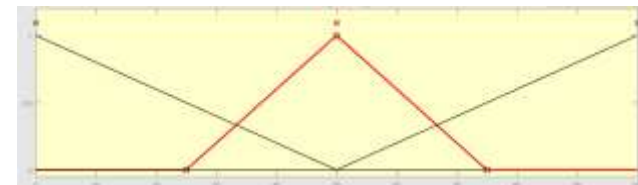
- i. Nilai Health dari prajurit
- ii. Nilai Stamina dari prajurit
- iii. Tingkat Agility dari prajurit
- iv. Tingkat Aggressive dari prajurit

Health dari kelas swordman memiliki range 0 – 150 poin, sedangkan kelas archer memiliki range health antara 0 – 100 poin, dan kelas terakhir adalah kelas magician yang memiliki range health antara 0 – 60 poin. Stamina dari kelas swordman memiliki range 0 – 100 poin, sedangkan kelas archer memiliki range stamina antara 0 – 80 poin, dan kelas terakhir adalah kelas magician yang memiliki range stamina antara 0 – 60 poin. Untuk tingkat agility dan tingkat aggressive setiap kelas akan memiliki range yang sama yaitu range 0.5 – 2.5 poin untuk tingkat agility dan range 0.5 – 1 poin untuk tingkat aggressive. Setiap nilai atribut akan diproses oleh NPC fuzzy controller untuk menghasilkan nilai derajat keanggotaan ( $\mu$ ) dan akan menentukan behavior apa yang tepat untuk pasukan (1 pasukan dikontrol oleh 1 NPC fuzzy controller).

Pada kelas swordman untuk atribut health dan stamina terdapat 3 kategori fuzzy membership health yaitu **weak (W)**, **medium (M)**, dan **strong (S)**. Gambar 1. menunjukkan membership health W, M, S untuk kelas swordman dan Gambar 2. menunjukkan membership stamina W, M, S untuk kelas swordman.

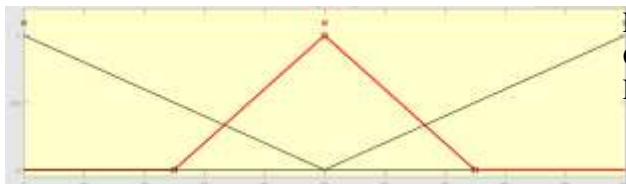


Gambar 1. Input Health Membership Function NPC Fuzzy Controller, Kelas Swordman

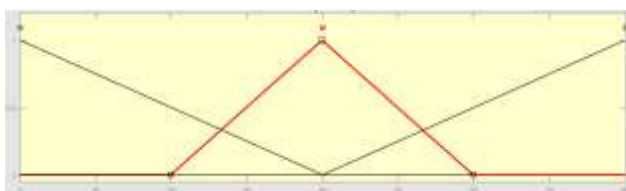


Gambar 2. Input Stamina Membership Function NPC Fuzzy Controller, Kelas Swordman

Pada kelas archer untuk atribut health dan stamina terdapat 3 kategori fuzzy membership health yaitu **weak (W)**, **medium (M)**, dan **strong (S)**. Gambar 3. menunjukkan membership health W, M, S untuk kelas swordman dan Gambar 4. menunjukkan membership stamina W, M, S untuk kelas archer.

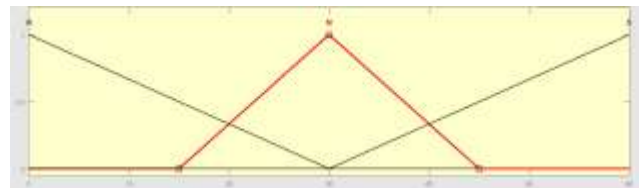


Gambar 3. Input Health Membership Function NPC Fuzzy Controller, Kelas Archer

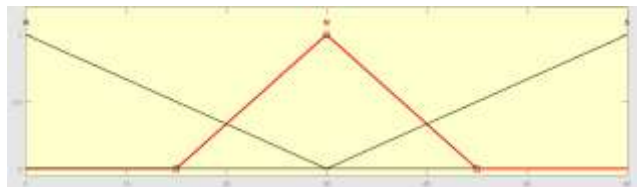


Gambar 4. Input Stamina Membership Function NPC Fuzzy Controller, Kelas Archer

Pada kelas mage untuk atribut health dan stamina terdapat 3 kategori fuzzy membership health yaitu **weak (W)**, **medium (M)**, dan **strong (S)**. Gambar 5. menunjukkan membership health W, M, S untuk kelas mage dan Gambar 6. menunjukkan membership stamina W, M, S untuk kelas mage.

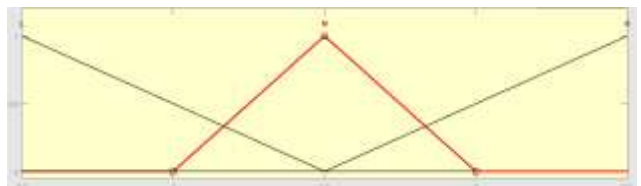


Gambar 5. Input Health Membership Function NPC Fuzzy Controller, Kelas Mage

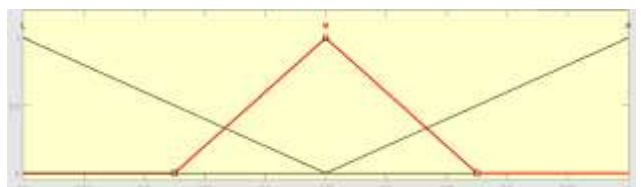


Gambar 6. Input Stamina Membership Function NPC Fuzzy Controller, Kelas Mage

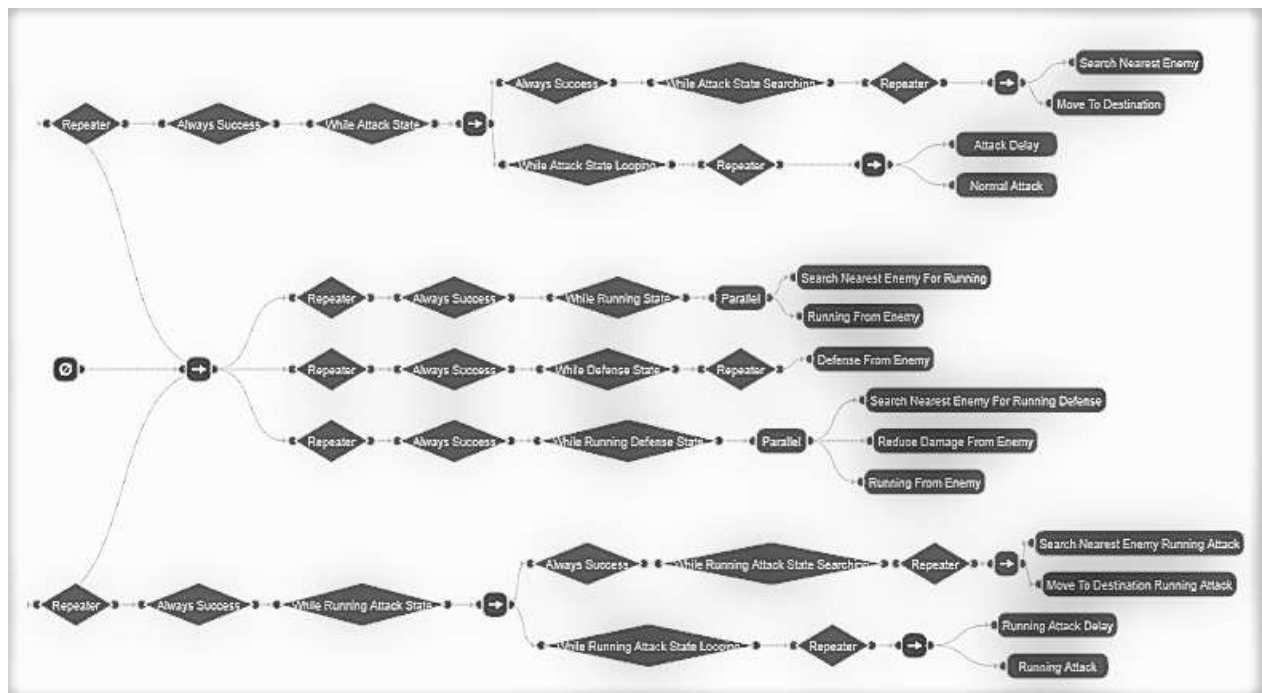
Pada fuzzy controller untuk agility dan aggressive memiliki membership function yang sama pada semua kelas. Untuk atribut agility terdapat 3 kategori fuzzy membership agility yaitu low (L), medium (M), dan high (H). Berikut Gambar 7 yang menjelaskan membership function L, M, dan H untuk atribut agility dan Berikut Gambar 8 yang menjelaskan membership function L, M, dan H untuk atribut agility.



Gambar 7. Input Agility Membership Function NPC Fuzzy Controller Semua Kelas



Gambar 8. Input Aggressive Membership Function NPC Fuzzy Controller Semua Kelas



Gambar 9. Behavior Tree Dari Pasukan

**B. Behavior Tree NPC Controller**

NPC Fuzzy Controller akan menjalankan 1 bagian kecil dari keseluruhan behavior tree dan keseluruhan behavior tree akan dibagi menjadi 5 bagian behavior yaitu **Attack (A)**, **Running Attack (RA)**, **Defense (D)**, **Running (RD)**, **Running (R)** yang akan diproses dan dijadikan sebagai action atau behavior dari NPC Controller. Gambar 9 adalah behavior tree yang memiliki kelima bagian behavior yang dimiliki oleh semua NPC.

Cara menjalankan bagian kecil dari behavior adalah dengan menggunakan decorator while yang memiliki kondisi tertentu. Sebagai contoh jika NPC akan menjalankan behavior attack, maka Attack State pada pada “While Attack Sate” akan diubah nilainya menjadi true dan semua state kecuali attack state akan diubah nilainya menjadi false.

Dengan menggunakan metode model Fuzzy Zero-Order Sugeno, output (dalam bentuk behavior) yang dihasilkan dari NPC Fuzzy Controller yang berupa rule dijabarkan pada tabel 2. Nilai kosong pada salah satu cell pada tabel 2 merupakan cara untuk meringkas rule yang lebih sederhana. Sebagai contoh rule nomor 1 yang

memiliki value dari atribut health yaitu weak, rule ini akan menghasilkan action (behavior) running apapun value dari atribut stamina, agility, dan aggressive.

TABEL II. Rule Pada NPC Controller

No.	Player 1										Action									
	Health			Stamina			Agility		Aggressive		Running Attack	Attack	Defense	Running	Running Defense					
	Weak	Medium	Strong	Weak	Medium	Strong	Low	Medium	High	Low						Medium	High			
1	✓																			
2			✓			✓								✓						
3			✓												✓					
4	✓					✓								✓						
5	✓							✓								✓				
6			✓	✓											✓					
7									✓				✓							
8	✓		✓						✓								✓			
9						✓			✓											✓
10	✓												✓							
11					✓									✓						
12							✓						✓							
13			✓										✓							✓
14			✓		✓									✓						
15				✓					✓											✓
16									✓				✓							
17	✓			✓											✓					
18	✓		✓										✓							

#### 4. PERCOBAAN DAN HASIL

##### A. Analisa persiapan percobaan

Untuk bagian uji coba pada penelitian ini, akan diterapkan 2 skenario pada permainan. Setiap skenario akan dilakukan 50 kali pertempuran, dengan atribut dan posisi setiap pasukan akan diacak. Setiap prajurit memiliki nilai minimal untuk atribut health dan stamina sebesar 10. Permainan berakhir ketika jumlah pasukan pada salah satu kelompok mencapai angka 0. Setiap behavior dari pasukan akan dicatat, dan direpresentasikan dalam bentuk frekuensi kemunculan setiap behavior.

##### B. Analisa pada akhir percobaan

Terdapat 2 skenario percobaan yang setiap skenario akan dijalankan 50 kali percobaan perang.

- Skenario Pertama : Pada skenario uji coba ini setiap pasukan pada kelompok 2 (musuh) hanya menggunakan behavior attack (dikarenakan behavior attack adalah behavior yang bisa dilakukan untuk mencapai kemenangan). Seluruh pasukan pada kelompok 2 (musuh) tidak akan menggunakan NPC fuzzy controller, sedangkan seluruh pasukan kelompok 1 akan diterapkan NPC fuzzy controller.
- Skenario Kedua : Pada skenario uji coba ini setiap pasukan pada kelompok 2 (musuh) akan diterapkan NPC fuzzy controller dan behavior dari setiap pasukan akan dikontrol oleh NPC fuzzy controller. Pada seluruh pasukan kelompok 1 akan diterapkan NPC fuzzy controller.

TABEL III. HASIL DARI PERCOBAAN SKENARIO PERTAMA

Kelompok 1	Kelompok 2
45 Menang	5 Menang



Gambar 10. Gambar Frekuensi Kemunculan Setiap Behavior Pada Skenario Pertama Kelompok 1



Gambar 11. Gambar Frekuensi Kemunculan Setiap Behavior Pada Skenario Pertama Kelompok 2

TABEL IV. HASIL DARI PERCOBAAN SKENARIO KEDUA

Kelompok 1	Kelompok 2
23 Menang	27 Menang



Gambar 12. Gambar Frekuensi Kemunculan Setiap Behavior Pada Skenario Kedua Kelompok 1



Gambar 13. Gambar Frekuensi Kemunculan Setiap Behavior Pada Skenario Kedua Kelompok 2

Dari hasil percobaan pada tabel 3, menunjukkan bahwa skenario pertama kelompok 1 memiliki tingkat persentase kemenangan 90% (45 kemenangan), hal ini dikarenakan kelompok kedua hanya menggunakan behavior attack yang melawan kelompok pertama yang menggunakan NPC fuzzy controller dan dapat membuat behavior setiap pasukan kelompok 1 lebih baik.

Kemudian pada skenario kedua terlihat pada tabel 4, kelompok 1 memiliki tingkat persentase



kemenangan sebesar 46% (23 kemenangan) dan kelompok 2 memiliki tingkat persentase sebesar 54% (27 kemenangan). Peningkatan persentase kemenangan kelompok 2 meningkat dari 10% (skenario pertama) menuju 54%, dikarenakan kelompok 1 dan kelompok 2 pada skenario ini sama – sama menggunakan NPC fuzzy controller.

## 5. KESIMPULAN

Dari hasil percobaan pada penelitian ini, setiap pasukan yang memiliki NPC fuzzy controller, akan jauh lebih baik daripada pasukan yang tidak menggunakan NPC fuzzy controller. Hal ini dikarenakan NPC fuzzy controller akan membuat behavior setiap pasukan lebih baik berdasarkan parameter – parameter yang diberikan kepada NPC fuzzy controller. Tingkat persentase behavior yang paling sering muncul adalah behavior Running Attack dan Defense, behavior tersebut yang paling dominan dari kelima behavior yang lain.

## 6. DAFTAR PUSTAKA

- [1] Supeno Mardi Susiki Nugroho, Ika Widiastuti, Mochamad Hariadi, dan Mauridhi Hery Purnomo, 2013, “*Fuzzy Coordinator Based Intelligent Agents For Team Coordination Behavior In Close Combat Games*”, Journal of Theoretical and Applied Information Technology (Jatit)
- [2] Muhammad Aminul Akbar, Mochamad Hariadi, Supeno Mardi S.N, 2015, “*Smart Agent for Multi Behaviour NPC Coordination Using Fuzzy Coordinator*.”, Intelligent Technology and Its Applications (ISITIA)
- [3] Michele Colledanchise dan Petter O gren, 2014, “*How Behavior Trees Modularize Robustness and Safety in Hybrid Systems*”, IEEE
- [4] Alexander Shoulson, Francisco M. Garcia, Matthew Jones, Robert Mead, dan Norman I. Badler, 2011, “*Parameterizing Behavior Trees*”, Lecture Notes in Computer Science pp 144-145
- [5] Ian Millington dan John Funge, ”*Artificial Intelligence for Games (Second Edition)*”, Morgan Kaufmann, Agustus 2009, pp 309-330.

