

DESAIN DAN IMPLEMENTASI NON PLAYABLE CHARACTER PADA PERMAINAN EDUKASI RANTAI MAKANAN UNTUK KONSUMEN TINGKAT 3 (ULAR) BERTEKNOLOGI MOBILE AUGMENTED REALITY

Agus Komarudin¹, ** Rezki Yuniarti²

(1) Universitas Jenderal Achmad Yani, (Contact : 022-6631302, adinmuflih@yahoo.co.id)

(2) Universitas Jenderal Achmad Yani, (Contact : 022-6631302, rezkiy@gmail.com)

Abstrak

Permainan komputer memiliki fungsi salah satunya sebagai sarana hiburan, namun belakangan ini permainan komputer juga digunakan untuk pembelajaran, disebut edugame. Edugame cukup efektif untuk menyampaikan suatu materi atau bahan ajar, serta memberikan user experience berbeda daripada proses belajar secara konvensional. Tahun 2015, octagon studio membuat aplikasi pembelajaran menggunakan teknologi mobile augmented reality. Pengguna dapat melihat visualisasi objek hewan 3 dimensi tanpa terdapat perilaku/kondisi didalamnya. Bermaksud mengembangkan user experience dari aplikasi tersebut, maka akan dibuat suatu edugame bertema rantai makanan di area sawah dimana dibutuhkan karakter hewan yang dapat berperilaku dan berinteraksi dengan perubahan variabel sekitarnya. Dengan diterapkannya kondisi-kondisi pada tokoh hewan tersebut akan menyebabkan seolah-olah hewan dapat berinteraksi dan hidup. Visualisasi ini akan mendukung proses belajar pada tema rantai makanan area sawah. Tokoh yang dibuat pada penelitian ini adalah ular selanjutnya disebut sebagai non playable character (NPC). Keberadaan NPC yang dapat bertindak laku karena pengaruh lingkungannya pada suatu permainan komputer sangatlah penting agar terdapat kesan nyata. Proses pengambilan keputusan NPC ular dipengaruhi oleh variabel sekitarnya berupa variabel yang dapat dikontrol oleh pemain (Playable Character). Untuk proses perhitungan beberapa variabel mengalami proses fuzzifikasi yang kemudian nilai-nilai tersebut akan menjadi masukan pada blok pengambilan keputusan yang digambarkan menggunakan Finite State Machine. Variabel yang berhubungan dengan jarak antara NPC dengan PC dihitung dengan menggunakan metode Line of Sight (LOS) yang kemudian akan dijadikan variabel masukan juga beserta variabel lainnya. Hasil dari penelitian adalah, tokoh NPC dapat berperilaku sesuai dengan rancangan pada FSM. Tokoh dapat menghindari dan mengejar Playable Character. Proses menghindar dan mengejar tetap menggunakan Line of Sight. Game tidak mempertimbangkan rintangan (obstacle), dengan asumsi ruangan terbuka tanpa kubangan air ataupun bukit.

Key word : edugame, rantai makanan, NPC, FSM, LOS

1. Pendahuluan

Perangkat komputer dapat digunakan sebagai alat untuk membantu dan memudahkan berbagai macam pekerjaan. Selain itu pula komputer dapat digunakan sebagai media untuk hiburan yaitu bermain game. Permainan komputer (*computer game*) memiliki fungsi salah satunya sebagai sarana hiburan, namun belakangan ini permainan komputer juga digunakan untuk pembelajaran, disebut edugame. Edugame cukup efektif untuk menyampaikan suatu materi atau bahan ajar [1], serta memberikan juga *user experience* berbeda daripada ketika seseorang belajar secara konvensional.

Untuk bermain game sekarang tidak hanya menggunakan media komputer, melainkan dapat menggunakan media *smartphone*. Permainan pada *smartphone* semakin banyak, dan tidak hanya permainan namun edugame juga dapat diunduh secara gratis atau berbayar. Pengguna

smartphone tidak hanya untuk dewasa, namun remaja dan anak-anak menggunakan *smartphone* untuk bermain dan belajar. Pada tahun 2015, octagon studio membuat aplikasi pembelajaran berplatform android [2]. Pengguna dapat melihat objek hewan 3 dimensi dengan menggunakan teknologi *augmented reality*. Objek yang dihasilkan adalah visualisasi dari hewan, dan belum terdapat *artificial life* didalamnya. Dengan memberikan perilaku pada karakter hewan tersebut tentu akan memberikan *user experience* yang berbeda. Pembuatan karakter tak termainkan, selanjutnya disebut NPC (*Non Playable Character*) tidaklah harus sangat cerdas, namun NPC dapat bertindak sesuai dengan keinginannya. Pada beberapa permainan komputer, tokoh yang sangat pintar justru dihindari agar tidak memberikan tantangan yang tidak dapat diselesaikan oleh pemain. Penentuan sifat karakter sangat dipengaruhi oleh genre permainan [3].

Tema Rantai makanan merupakan tema yang pembelajaran yang cukup sering dibahas di mata

pelajaran IPA ataupun Biologi. Menariknya, selain membahas rantai makanan dimana terdapat proses memakan dan dimakan, berburu dan diburu, juga melibatkan banyak karakter hewan yang dapat divisualisasikan disini. Sehingga dibuat suatu edugame bertema rantai makanan dengan daerah sawah. Maka dibutuhkan karakter hewan yang dapat berperilaku dan menunjang scenario rantai makanan.

Kecerdasan buatan seringkali digunakan untuk pembuatan karakter, agar tercakup suatu *behaviour* (sifat) pada karakter ataupun digunakan untuk pengambilan keputusan pada NPC [4] [5]. Keberadaan NPC yang dapat bertingkah karena pengaruh lingkungannya pada suatu permainan komputer sangatlah penting agar terdapat kesan nyata. Untuk membentuk sifat atau perilaku NPC dapat menggunakan Fuzzy Logic dan Finite State Machine (FSM). Tokoh yang akan dibangun adalah NPC ular, dimana ular berada pada konsumen tingkat 3 dapat berburu dan diburu. Namun secara keseluruhan rantai makanan memiliki beberapa macam hewan yang belum akan dibahas di sini.

Tujuan dari penelitian ini adalah membangun model dan implementasi salah satu karakter (NPC) konsumen tingkat satu yaitu ular yang merupakan bagian dari edugame rantai makanan. Sehingga akan dibuat pemodelan status aksi ular menggunakan *Finite State Machine* dan *Fuzzy Tsukamoto* untuk keputusan kecepatan gerak dalam hal mengejar mangsa atau dikejar predator (*chasing atau evading*). Algoritma Line Of Sight digunakan untuk menghitung jarak antara NPC dan predator/mangsa.

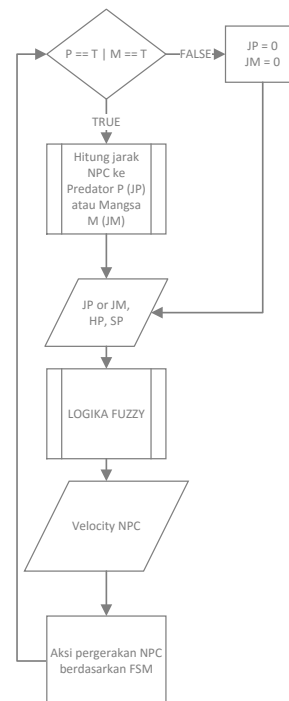
2. Metodologi

Pemodelan NPC dapat menggunakan banyak metode sesuai kebutuhannya. Pada penelitian sebelumnya, perilaku NPC berperang berbasis giliran (*turn based strategy*) dimodelkan dengan menggunakan Logika Fuzzy dan Naïve Bayes Classifier, tanpa mempertimbangkan jarak antara musuh dan NPC. Hasil yang didapat adalah dengan menggunakan 15 data sampel secara acak, menghasilkan aksi penyerangan NPC yang sesuai [6]. Pemodelan bagi autonomous NPC juga dapat dicapai dengan menggunakan FSM, namun ketika terdapat *multiple* NPC, pemodelan jenis ini akan menghilangkan kesan koordinasi antar NPC [7]. Terdapat pula game yang mempertimbangkan jarak serta menggunakan logika Fuzzy untuk proses pengecekan status kesehatan dari NPC tersebut, namun variable jarak digunakan untuk pencarian jalur terdekat dan tidak menjadi pertimbangan untuk dimasukkan ke dalam blok logika Fuzzy, logika Fuzzy digunakan untuk menentukan keputusan perilaku perang [8].

Pemodelan karakter ular (NPC) pada permainan rantai makanan memiliki beberapa tahapan :

1. NPC memeriksa keberadaan tokoh lainnya (mangsa ataupun predator) kemudian memeriksa jarak antara NPC dengan tokoh lainnya menggunakan perhitungan *Line of Sight* [5].
2. NPC melakukan proses fuzzifikasi beberapa variable untuk menghasilkan nilai kecepatan NPC. Adapun arah gerak NPC masih menggunakan algoritma *Basic Chasing and Evading* [5]
3. NPC mengambil keputusan atas aksi yang dilakukan berdasarkan status yang dimodelkan menggunakan FSM

Diagram proses tersebut digambarkan seperti pada Gambar 1 berikut ini:



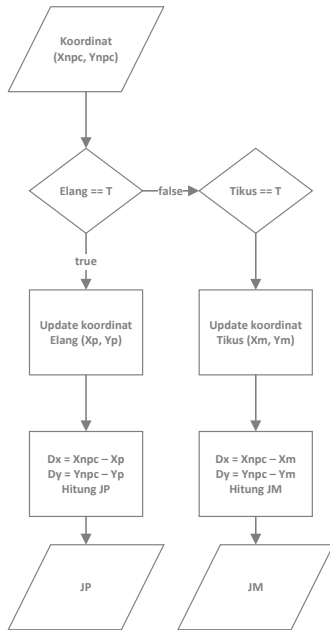
Gambar 1.Proses Perhitungan Velocity NPC secara umum

Masukan yang melalui proses Fuzzifikasi adalah nilai-nilai *variable* sebagai berikut ini:

1. JP dan JM, adalah nilai jarak antara NPC dan karakter predator (JP) atau antara NPC dan karakter mangsa (JM). Nilai ini selalu dicari seandainya terdapat mangsa atau predator. Jika tidak terdapat keduanya, maka akan diinisialisasi dengan nilai nol. Nilai ini menjadi penentu kecepatan gerak (*Velocity*) NPC.
2. HP (*Healthy Point*) dan SP (*Stamina Point*) adalah nilai yang telah diinisialisasi sebelumnya pada NPC. Nilai ini berubah sesuai dengan keadaan

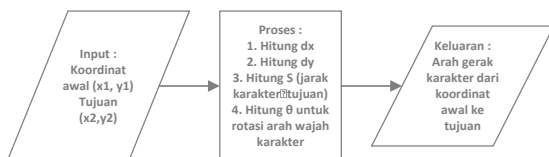
(states) yang dikerjakan oleh NPC. Nilai bersifat dinamis, sehingga proses perhitungan pada Gambar 1 akan terus berulang.

Proses perhitungan pencarian nilai JP dan JM ditunjukkan seperti pada Gambar 2.



Gambar 2. Proses Perhitungan jarak antara NPC dan karakter lainnya

Lingkungan game tidak membagi lantai menjadi sekumpulan *grid/tile* besar yang memiliki *terrain*. Namun tetap saja pada pemilihan jalur harus mempertimbangkan *grid* tersebut, baik menggunakan *pels* atau *pels* yang diperbesar menjadi *grid*. Pemilihan jalur oleh karakter ular menggunakan metoda sederhana seperti *Line of sight chasing* pada lingkungan *grid* karena belum membahas *terrain* ataupun *obstacle*. *LOS chasing* tidak membahas arah muka tokoh/karakter. Agar terlihat nyata, arah muka karakter harus dirotasi sebanyak θ derajat. Gambaran proses ini dapat dilihat pada Gambar 3.



Gambar 3. Proses Perhitungan jarak dan arah muka NPC

Proses perhitungan untuk mendapatkan nilai pada Gambar 2 dan Gambar 3 adalah sebagai berikut :

1. *Line of Sight*, perhitungan nilai dx dan dy dapat dihitung dengan menggunakan persamaan (1) :

$$dx = x_2 - x_1 \quad (1)$$

$$dy = y_2 - y_1$$

dimana,

dx = selisih jarak x antara NPC dengan predator atau mangsa,

dy = selisih jarak y antara NPC dengan predator atau mangsa.

Kemudian menghitung jarak tempuh dari koordinat karakter menuju tujuannya dapat menggunakan teorema *phytagoras* menggunakan persamaan (2)

$$JPatauJp = \sqrt{dx^2 + dy^2} \quad (2)$$

2. Menghitung sudut rotasi untuk arah muka NPC terhadap mangsa. Langkah ini bukan termasuk *Los Chasing*.

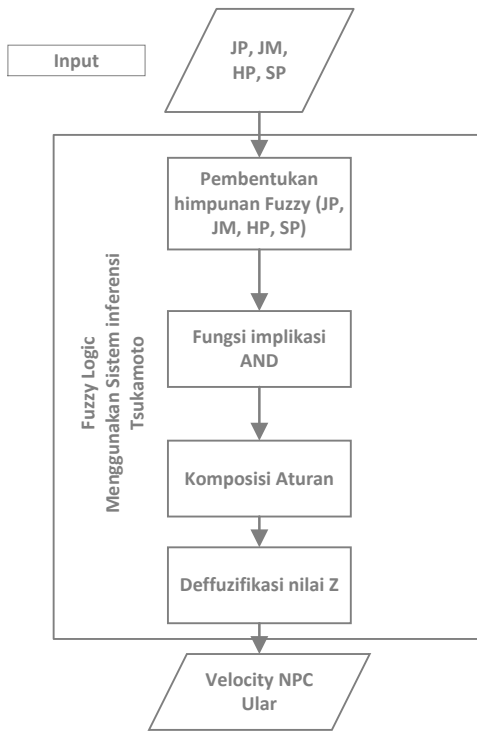
Agar karakter memutar arah muka menuju mangsa, maka harus ditentukan arah sudut putar/rotasi karakter. Sehingga arah putar dapat ditentukan dengan cara mencari nilai sudut menggunakan persamaan (3)

$$\theta = \arctan\left(\frac{dy}{dx}\right) \quad (3)$$

Sehingga jika karakter yang ditemui adalah predator, maka arah muka haruslah sebaliknya, menggunakan persamaan (4)

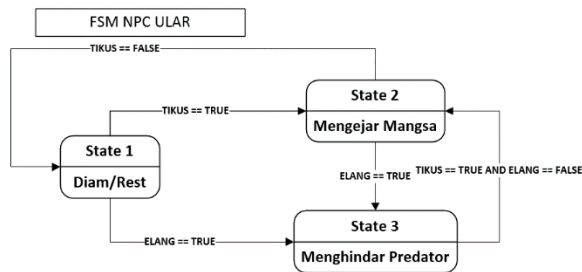
$$\theta = 180 + \left(\arctan\left(\frac{dy}{dx}\right)\right) \quad (4)$$

Selanjutnya dengan menggunakan 4 masukan, selanjutnya variable akan melalui proses fuzzyfikasi. Seperti pada Gambar 4. Logika fuzzy pernah diterapkan untuk mengoptimasi perilaku NPC dalam hal ini si musuh pada permainan RPG dengan menggunakan tiga fungsi keanggotaan tanpa melibatkan variable lainnya lagi. Hasilnya Penerapan fuzzy dalam permainan ini telah berjalan dengan baik, dimana musuh tipe penyerang menjadi cukup agresif (28% perilaku menyerang, 17% menyerang brutal), tipe pemanah agresif jika berada pada jarak jauh (49% perilaku menyerang), dan musuh boss sangat agresif (20% perilaku menyerang, 69% perilaku menyerang brutal) [9].



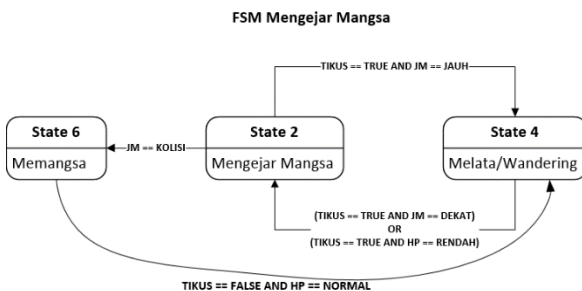
Gambar 4. Proses Perhitungan Velocity NPC ular

Kemudian setelah didapatkan Velocity, NPC mengambil keputusan atas aksi yang dilakukan berdasarkan status yang dimodelkan menggunakan FSM pada Gambar 5. Status mengejar dan dikejar akan menyebabkan NPC ular bergerak dengan menggunakan nilai Velocity hasil defuzzifikasi pada proses Fuzzy Logic.



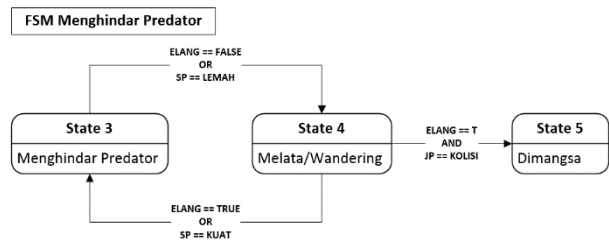
Gambar 5. FSM perilaku NPC Ular

State mengejar mangsa seperti pada Gambar 6.

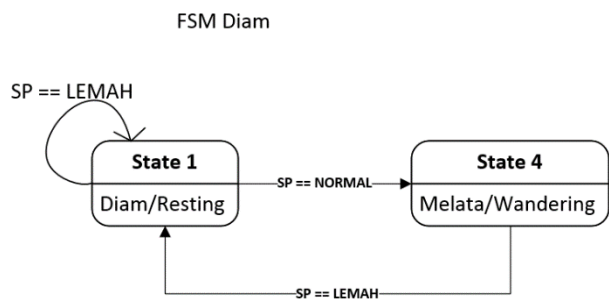


Gambar 6. FSM mengejar mangsa

Penjabaran dari setiap state terlihat pada Gambar 7 dan Gambar 8.



Gambar 7. FSM NPC ular menghindari Elang



Gambar 7. FSM NPC ular diam dan melata

3. Pembahasan

Pada sub bahasan ini akan dibahas perancangan perhitungan logika fuzzy, yang akan mengikuti langkah seperti pada Gambar 4. Dalam system logika fuzzy terdapat beberapa tahapan, yaitu :

1. Perancangan himpunan Domain/variabel yang akan dibahas adalah:
 - a) JP (jarak predator),
 - b) JM (jarak mangsa),
 - c) HP (healthy point mempengaruhi perasaan lapar dan kesehatan. Nilai ini akan mentrigger mengejar mangsa),
 - d) SP (stamina point mempengaruhi kekuatan gerak dari stamina NPC)

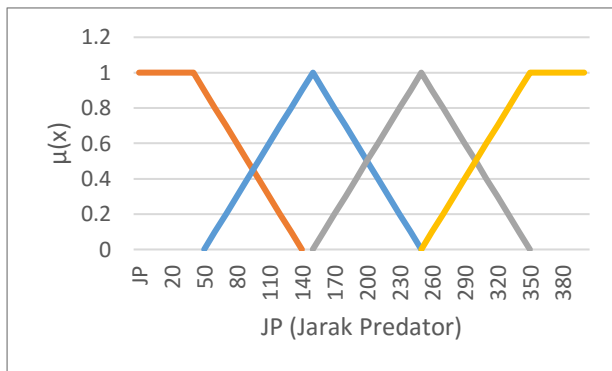
Keempat domain tersebut akan mempengaruhi nilai konsekuen yang dalam hal ini adalah nilai velocity NPC. Operator Representasi Kurva Segitiga. Rancangan himpunan untuk setiap variable terlihat seperti pada Tabel 1

Tabel 1. Perancangan himpunan untuk setiap domain

Variabel	Himpunan	Domain himpunan fuzzy
JP (Jarak Predator)	KOLISI	[0 150]
	DEKAT	[50 250]
	SEDANG	[150 350]
	JAUH	[250 +∞]

JM (Jarak Mangsa)	KOLISI	[0 150]
	DEKAT	[50 250]
	SEDANG	[150 350]
	JAUH	[250 +∞]
HP (Healthy Point)	RENDAH	[0 50]
	NORMAL	[25 75]
	TINGGI	[50 100]
SP (Stamina Point)	LEMAH	[0 50]
	NORMAL	[25 75]
	KUAT	[50 100]

Fungsi keanggotaan berdasarkan Tabel 1, untuk JP dan JM menggunakan direpresentasi menggunakan Kurva Segitiga dan representasi Kurva Bentuk Bahu dapat dilihat pada Gambar 8.



Gambar 8. Representasi himpunan fuzzy untuk JP

Fungsi keanggotaan untuk setiap masing-masing himpunan pada domain JP adalah :

Fungsi keanggotaan Kolisi :

$$\mu[x] = \begin{cases} 1; & x \leq 50 \\ \frac{150-x}{150-50}; & 50 < x \leq 150 \end{cases} \quad (5)$$

Fungsi keanggotaan Dekat :

$$\mu[x] = \begin{cases} 0; & x \leq 50 \\ \frac{x-50}{150-50}; & 50 < x \leq 150 \\ \frac{250-x}{250-150}; & 150 < x \leq 250 \end{cases} \quad (6)$$

Fungsi keanggotaan Sedang :

$$\mu[x] = \begin{cases} 0; & x \leq 150 \\ \frac{x-150}{250-150}; & 150 < x \leq 250 \\ \frac{350-x}{350-250}; & 250 < x \leq 350 \end{cases} \quad (7)$$

Fungsi keanggotaan Jauh :

$$\mu[x] = \begin{cases} 0; & x \leq 150 \\ \frac{x-250}{350-250}; & 250 < x \leq 350 \\ 1; & 350 \leq x \end{cases} \quad (8)$$

Hal tersebut dapat diterapkan juga untuk semua variable JM, HP dan SP.

- Menggunakan Metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-Then harus direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan yang monoton. Keluaran hasil inferensi dari tiap-tiap aturan diberikan secara tegas berdasarkan $\alpha_{predikat}$ (fire strength). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

Berikutnya menentukan aturan, misalnya :

- [R1] IF jp KOLISI THEN VELOCITY DIAM;
- [R2] IF jp DEKAT AND hp RENDAH AND sp LEMAH THEN velocity DIAM;

Selanjutnya aturan Fuzzy yang digunakan dalam permainan ini dijelaskan di dalam Tabel 2

Tabel 2. Aturan menggunakan operator AND

Rule	JP	HP	SP	VELOCITY
[R1]	Kolisi			Diam
[R2]	Dekat	Rendah	Lemah	Lambat
[R3]	Dekat	Rendah	Normal	Lambat
[R4]	Dekat	Rendah	Kuat	Normal
[R5]	Dekat	Normal	Lemah	Lambat
[R6]	Dekat	Normal	Normal	Normal
[R7]	Dekat	Normal	Kuat	Cepat
[R8]	Dekat	Tinggi	Lemah	Cepat
[R9]	Dekat	Tinggi	Normal	Cepat
[R10]	Dekat	Tinggi	Kuat	Cepat
[R11]	Sedang	Rendah	Lemah	Lambat
[R12]	Sedang	Rendah	Normal	Lambat
[R13]	Sedang	Rendah	Kuat	Normal
[R14]	Sedang	Normal	Lemah	Lambat
[R15]	Sedang	Normal	Normal	Normal
[R16]	Sedang	Normal	Kuat	Cepat
[R17]	Sedang	Tinggi	Lemah	Cepat
[R18]	Sedang	Tinggi	Normal	Cepat
[R19]	Sedang	Tinggi	Kuat	Cepat

dst:	:	:	:	:
[R28]	Jauh	Tinggi	Kuat	Lambat

Rule yang melibatkan perhitungan variable JM juga dibuat dengan cara mirip seperti Tabel 2, namun nilai velocity digunakan untuk mengejar mangsa, bukan menghindari mangsa.

4. Kesimpulan

Berdasarkan hasil penelitian dapat disimpulkan bahwa kondisi NPC ular dapat memiliki perilaku cerdas yang didasari dari 4 inputan *fuzzy* dan *FSM* dengan nilai akurasi sebesar 90%.

Dengan demikian dari hasil tersebut dapat disimpulkan bahwa metode logika *fuzzy* dengan *FSM* dalam penelitian ini dapat diterapkan dengan baik pada NPC ular sesuai dengan analisis, perancangan dan pengujian sistem yang telah dilakukan.

Daftar Pustaka

- [1] R. Yuniarti and A. Komarudin, "Rancang Bangun Interaksi Pembelajaran Drill Menggunakan Pendekatan User Centerd Design (UCD) Untuk Game Peningat Resep Masakan," in *Seminar Nasional Informatika dan Aplikasinya*, Cimahi, 2013.
- [2] O. Studio, "Animal 4D+ Augmented Reality Apps & Flashcards for Kids," Octagon, 28 Desember 2015. [Online]. Available: <https://www.octagonstudio.com/1/animal-4d-cards>. [Accessed 15 Februari 2016].
- [3] J. A. Glasser and L.-K. Soh, "AI in Computer Games: From the Player's Goal to AI's Role," UNL, Lincoln, 2004.
- [4] K. Donald, "Designing Artificial Intelligence for Games (Part 1)," Intel Developer Zone, 20 Juni 2009. [Online]. Available: <https://software.intel.com/en-us/articles/designing-artificial-intelligence-for-games-part-1>. [Accessed 1 April 2015].
- [5] D. M. Bourg and G. Seeman, *AI for Game Developers*, Sebastopol, California: O'Reilly, 2004.
- [6] D. Fathurochman, W. Witanti and R. Yuniarti, "Perancangan Game Turn Based Strategy Menggunakan Logika Fuzzy dan Naive Bayes Classifier," in *Seminar Nasional Informatika Semnasif*, Yogyakarta, 2014.
- [7] E. Sanjaya and G. S. Budhi, "Aplikasi Game yang Menerapkan Metode Autonomous NPC untuk Mengkoordinasi AI dari Enemy," *Jurnal Infra*, vol. 2, no. 1, pp. 120-124, 2014.
- [8] W. Muhammad, V. Suhartono and M. A. Soeleman, "Penentuan Jalur Pergerakan dan Perilaku Perang Karakter Bukan Pemain Menggunakan Algoritma A* dan Metode Logika Fuzzy pada Game Simulator Tank," *Jurnal Teknologi Informasi*, vol. 10, no. 2, pp. 128-144, Oktober 2014.
- [9] K. R. Purba, R. N. Hasanah and M. A. Muslim, "Implementasi Logika Fuzzy Untuk Mengatur Perilaku Musuh dalam Game Bertipe Action-RPG," *Jurnal EECCIS*, vol. 7, no. 1, pp. 15-20, 2013.
- [10] F. Feisal, "Modeling Multi-Agent in Futsal Sport Using Finite State Machine," Departemen Ilmu Komputer FMIPA IPB, Bogor, 2009.