

RANCANG BANGUN RESPON DUA AGEN OTONOM DALAM AUGMENTED REALITY MENGUNAKAN METODE LOGIKA FUZZY

Dani Widinugroho¹⁾ Moch. Hariadi²⁾

^{1,2}Jurusan Teknik Elektro ITS, Surabaya 60111

.Email: daniwidinugroho@yahoo.com, mochar@ee.its.ac.id

ABSTRAK

Perkembangan penelitian kecerdasan buatan dapat dimanfaatkan untuk melakukan penelitian mengenai kecerdasan dalam permainan termasuk dalam augmented reality (AR). Respon dua agen otonom akan disimulasikan dengan menggunakan pendekatan metode logika fuzzy. Penelitian ini tentang bagaimana membangun respon dua agen otonom sehingga mampu merespon gerakan satu dengan lainnya menggunakan logika fuzzy. Dalam penelitian ini akan ditetapkan dua perilaku agen otonom. Pertama adalah perilaku mengejar yang diterapkan pada agen pertama. Dan kedua adalah perilaku menghindar yang akan diterapkan pada agen kedua. Untuk membangun respon kedua agen otonom menggunakan logika fuzzy metode Mamdani. Agen pertama mempunyai dua variabel input yaitu jarak dan kekuatan serta satu output berupa ketangkasan agen pertama. Didapat 9 aturan fuzzy untuk agen pertama dengan harapan keberhasilan 0,9 %. Agen kedua mempunyai dua variabel input yaitu ketangkasan agen pertama dan kewaspadaan agen kedua, serta satu output berupa ketangkasan agen kedua ketika menghindari agen pertama. Didapat 6 aturan fuzzy untuk agen kedua dengan kemungkinan keberhasilan 0,6%.

Kata kunci : agen otonom, respon agen, logika fuzzy, augmented reality (AR).

1. PENDAHULUAN

Dunia virtual yang dulu belum dirasakan nyata oleh manusia, kini hadir semakin nyata dalam kehidupan manusia. Dunia virtual tersebut semakin dibutuhkan di berbagai bidang. Dalam menghadirkan dunia virtual ke dalam dunia nyata, diperlukan beberapa faktor. Menurut [4], tantangan komputer grafis untuk membuat dunia virtual tampak nyata adalah, suara nyata, bergerak dan merespon interaksi secara *real time*, dan bahkan dirasa nyata oleh indera penglihatan manusia. Biasanya di dalam dunia virtual terdapat agen/*Non Playable Character* (NPC) berupa personifikasi penggambaran mirip manusia, hewan atau tumbuhan dengan sekaligus karakteristik atau sifat-sifatnya. Oleh karena itu selain grafis, kecerdasan buatan diperlukan untuk ditanamkan kepada agen/NPC sehingga di dalam dunia virtual tersebut ada aktifitas seperti dalam dunia nyata. Tidak hanya itu, agen/NPC menjadi seperti hidup dan berperilaku layaknya sebagai makhluk hidup.

Kecerdasan buatan mempunyai tujuan utama mengembangkan suatu agen otonom dengan memasukkan sifat *intelligent* dimana agen otonom tersebut memiliki karakteristik yang ada pada tingkah laku manusia sehingga memungkinkan agen otonom tersebut untuk bergerak sealam mungkin sesuai dengan keadaan dunia nyata [2]. Pergerakan otonom ini bisa dibangun dengan menggunakan logika *fuzzy*. Logika *fuzzy* telah banyak dipergunakan untuk mengembangkan berbagai penelitian tentang pergerakan agen otonom [6, 10].

Augmented reality (AR) merupakan teknologi yang menghadirkan citra sintesis ke dalam dunia nyata manusia. Manusia melihat dunia nyata yang ditambahkan ke dalamnya berupa grafis yang dibangkitkan oleh komputer [3]. AR dapat dipergunakan untuk memvisualkan suatu virtual agen/NPC dengan dunia nyata dalam proporsi yang berbeda, karena AR menghadirkan suatu level imersif dimana sampai saat ini belum ada satu peralatan virtual yang bisa melakukannya. Perkembangan penelitian terhadap AR sangat meningkat tajam pada dasawarsa terakhir ini [14].

Tantangan yang paling besar dari AR adalah bagaimana memadukan obyek virtual tersebut ke dalam kondisi dunia nyata (*real world*) secara fisik dalam suatu tingkatan interaksi yang mendekati serealistis mungkin. Untuk mampu beradaptasi serealistis mungkin dengan kondisi dunia nyata, obyek virtual/agen/NPC tersebut harus mempunyai suatu perilaku (*behavior*) yang khusus. Untuk itulah

diperlukan kecerdasan buatan agar obyek virtual/agen/NPC dapat berinteraksi dengan kondisi di sekitar dan dengan obyek virtual/agen/NPC lain secara realistis.

Penelitian kali ini bertujuan untuk membangun respon dua agen otonom yang masing-masing mampu merespon gerakan satu dengan lainnya. Respon masing-masing agen otonom dibangkitkan dengan menggunakan logika *fuzzy*.

2. Teori Penunjang

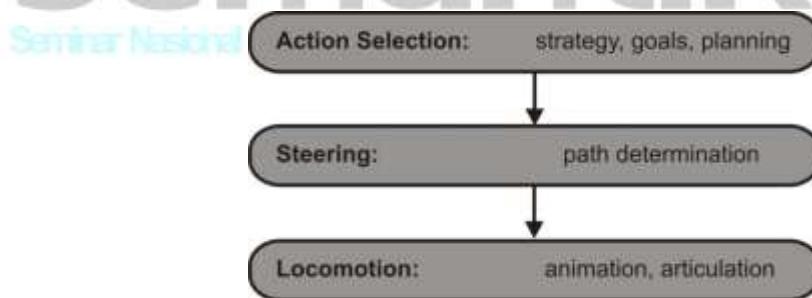
Untuk memberikan gambaran secara umum, pada bab 2 ini akan dibahas secara singkat mengenai agen otonom, *Finite State Machine* (FSM), *augmented reality* (AR), logika *fuzzy*, dan teori penunjang lain yang berkaitan dengan penelitian ini.

2.1. Agen Otonom

Agen otonom yang biasanya dipakai untuk penggunaan komputer animasi dan media interaktif seperti *games* dan *virtual reality*, sering disebut dengan istilah *autonomous character* [12]. Agen ini mewakili tokoh dalam cerita atau permainan dan memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan dari karakter dalam sebuah animasi, yang tindakannya ditulis di muka, dan untuk “avatar” dalam sebuah permainan atau *virtual reality*, tindakan yang diarahkan secara *realtime* oleh pemain. Agen virtual mendiami dunia yang dinamis dan tidak bisa diprediksi. Agar menjadi otonom, agen harus mampu merasakan lingkungannya dan menentukan reaksi apa yang harus dilakukan untuk mencapai tujuan sesuai dengan perilaku yang diberikan kepadanya [11]. Untuk itu desain dari perilaku agen harus ditentukan seperti memberikan persepsi, seleksi, aksi, kontrol pergerakan yang membuat perilakunya bisa diterima oleh akal manusia.

Perilaku agen mempunyai arti yang luas, salah satunya perilaku dapat dinyatakan sebagai sekumpulan aksi dari manusia atau hewan yang didasari oleh kemauannya atau insting [12]. Jadi, untuk menjadi otonom, agen tersebut harus bisa merasa lingkungannya dan menentukan sendiri apa yang harus dilakukan untuk mencapai tujuan yang telah ditentukan sebelumnya lewat perilaku [15].

Dari Gambar 1, hirarki teratas menunjukkan saat agen memilih aksi yang akan dilakukan. Pada bagian ini agen mempunyai strategi, tujuan serta perencanaan yang akan dilakukan (*action*). Lalu pada hirarki lapis ke dua, agen dalam bergerak akan menentukan arahnya (*steering*). Dan pada hirarki lapis ke tiga, agen mulai bergerak. Dalam hal ini karena di dunia virtual, jadi pergerakan ditunjukkan dengan beranimasi.

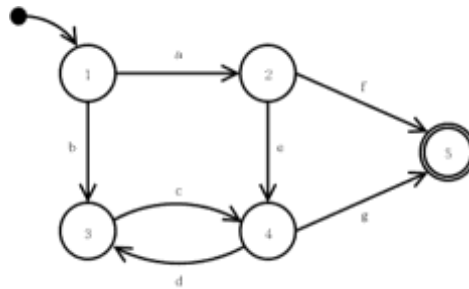


Gambar 1: Hirarki perilaku pergerakan agen [12]

2.2. *Finite State Machine* (FSM)

FSM adalah salah satu metode yang paling dikenal untuk memodelkan perilaku agen/NPC dalam sebuah game. Hal ini dikarenakan kesederhanaan dan kemudahan FSM untuk diimplementasikan [4].

Struktur FSM terdiri dari dua komponen utama yaitu keadaan (*state*) dan transisi (*transition*). *State* merupakan keadaan objek saat ini, sedangkan transisi adalah hal yang dilakukan untuk dapat berpindah dari satu *state* ke *state* yang lain (Gambar 2).



Gambar 2: Diagram FSM

Simbol *State* digambarkan dengan lingkaran, sedangkan transisi disimbolkan dengan anak panah dengan arah tertentu. Pada Gambar 2, angka 1-5 adalah *state* sedangkan huruf a-g adalah transisi. Kelebihan dari FSM adalah sederhana dan mudah diimplementasikan. Sedangkan kekurangannya, dalam sistem yang besar FSM akan sulit dipelihara.

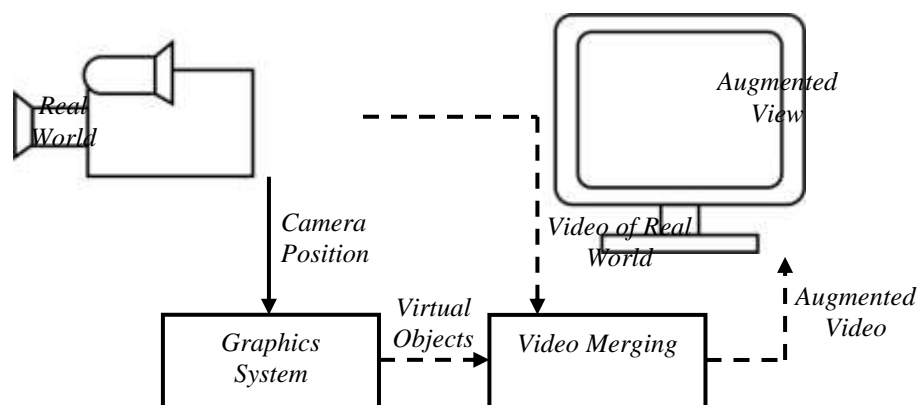
Implementasikan FSM dalam bahasa pemrograman dapat dilakukan dengan berbagai cara, yaitu: cara tradisional (menggunakan *switch-case*), *look-up* tabel (menggunakan matriks untuk menyimpan *state*), dan dengan paradigma *Object Oriented*. FSM berkembang menjadi beberapa variasi bila digabungkan dengan metode lain, antara lain: *Fuzzy State Machine (FuSM)*, *Probabilistic FSM (PFSM)*, *Hierarchical FSM (HFMS)*, dan lainnya.

2.3. Augmented Reality (AR)

AR merupakan percabangan ilmu dari *computer vision*. Definisi *computer vision* secara umum adalah merupakan ilmu dan teknologi bagaimana suatu sistem/machine melihat sesuatu [9]. Input untuk suatu sistem yang berbasis *computer vision* adalah citra atau *image* yang berupa *video sequent*, citra dari kamera, dan lain-lain. AR juga merupakan variasi dari *Virtual Environments (VE)*. Dalam VE, pengguna sepenuhnya berada dalam lingkungan buatan dari komputer, dan pengguna tidak dapat menyaksikan lingkungan nyata (*real world*) di sekitarnya secara langsung. Berbeda dengan VE, AR memberikan pengguna melihat lingkungan nyata sambil menyaksikan objek 3D virtual yang ditambahkan ke lingkungan nyata tersebut [14].

Walaupun AR merupakan teknologi yang mampu menempatkan suatu *image virtual* dari grafis komputer ke dunia nyata, elemen AR sendiri tidak dapat dilihat dengan mata telanjang. Untuk bisa melihat elemen virtual AR, diperlukan *display device* tertentu. Tahun 1968, Ivan Sutherland menciptakan sebuah alat yang menjadi sebuah *prototype* dari sistem AR yang pertama. Dalam makalahnya berjudul “*A Head-Mounted Three-Dimensional Display*”, sistem yang dibangunnya mengharuskan pengguna memakai alat seperti kacamata untuk melihat citra grafis 3D sederhana. Kacamata ini dalam perkembangannya sampai sekarang disebut *Head-Mounted Display (HMD)* [14]. Tetapi, setelah mengalami perkembangan dalam penelitian yang banyak dilakukan, dengan menggunakan monitor komputer atau televisi, elemen virtual AR dapat dinikmati dan dirasakan kehadirannya oleh pengguna.

Untuk membangun AR secara sederhana, tidak diperlukan peralatan yang sulit dicari dan mahal. Peralatan yang dibutuhkan hanya kamera/*web cam*, komputer dan monitor. Kamera/*web cam* digunakan untuk menangkap marker yang diperlukan sistem untuk menempatkan dan menampilkan elemen virtual dalam AR. Komputer dibantu aplikasi tertentu melakukan proses identifikasi *marker* sampai proses render model. Sedangkan monitor digunakan untuk menampilkan hasil akhir dari keseluruhan proses AR (Gambar 3).



Gambar 3: Perangkat pendukung teknologi AR [13]

2.3.1. Penanda dalam AR

Dalam sistem yang terdapat dalam AR, diperlukan suatu penanda dimana sebenarnya elemen virtual AR akan ditampilkan. Oleh karena itu, AR membutuhkan apa yang disebut *marker* AR. Berbagai macam *marker* sudah diujicoba dan dikembangkan, termasuk menggunakan LED (*light-emitting diodes*) pada tangan manusia. Pada perkembangannya, *marker* AR bisa dibuat dengan pola unik yang sederhana dengan syarat, *marker* dapat tertangkap dengan baik oleh kamera serta dapat diidentifikasi oleh sistem AR yang dibangun.

Pada Gambar 4 diperlihatkan ikan merupakan elemen objek virtual 3D, dan di bawahnya merupakan pola-pola berupa *marker*. Tugas *marker* tersebut untuk menentukan *viewpoint* dari dunia nyata sehingga virtual model 3D ikan dapat di render dan diletakkan dengan tepat.



Gambar 4: Virtual model 3D ikan tampil di atas *marker* [3]

2.3.2. ARToolKit

ARToolKit merupakan sebuah *library* tambahan untuk pemrograman dalam bahasa C dan C++ yang dikembangkan oleh HIT Lab dari *University of Washington*, digunakan untuk membuat aplikasi AR. Permasalahan yang dihadapi dalam membangun AR adalah menghitung sudut pandang pengguna secara *real time* dan akurat sehingga objek virtual dapat ditampilkan tepat pada lingkungan nyata yang diinginkan. ARToolKit menggunakan teknik pencitraan komputer untuk menghitung posisi dan orientasi kamera relatif terhadap *tracking device*. Sehingga programmer dapat menempatkan dengan tepat objek virtual yang dibuatnya pada *tracking device* tersebut.

Terdapat lima langkah proses kerja yang terdapat pada ARToolKit dalam menampilkan objek virtual dalam dunia nyata [7], seperti yang ditunjukkan pada Gambar 5.



Gambar 5: Proses kerja ARToolKit [7]

2.4. Logika Fuzzy

Logika *fuzzy* diperkenalkan pertama kali oleh Prof. Lotfi A. Zadeh pada tahun 1965. Logika *fuzzy* merupakan salah satu komponen pembentuk *soft computing* dan sebuah metode yang sering digunakan untuk mengatasi masalah-masalah yang memiliki jawaban lebih dari 1 serta digunakan untuk mengatasi masalah-masalah ketidakpastian. Yang dimaksud dengan ketidakpastian yaitu suatu masalah yang mengandung keraguan, ketidaktepatan, kurang lengkapnya informasi, dan nilai kebenarannya bersifat sebagian. Dalam banyak hal, logika *fuzzy* digunakan sebagai suatu cara untuk memetakan permasalahan dari input menuju ke output yang diharapkan [8]. Ada banyak metode aturan fuzzy yang bisa dipergunakan dalam proses pengambilan keputusan, dan dua metode aturan yang sering digunakan adalah:

1. Metode Mamdani

Metode ini sering dikenal juga dengan metode Max-Min, diperkenalkan oleh Ebrahim Mamdani tahun 1975. Baik variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan fuzzy [8]. Bentuk aturan yang digunakan pada metode Mamdani adalah:

$$\text{IF } x_1 \text{ is } A_1 \text{ AND } \dots \text{ AND } x_n \text{ is } A_n \text{ THEN } y \text{ is } B \quad (1)$$

Dimana A_1, \dots, A_n, B adalah nilai-nilai linguistik, sedangkan x_1 is A_1 menyatakan bahwa nilai dari variabel x_1 merupakan anggota himpunan *fuzzy* A .

2. Metode Sugeno

Diperkenalkan oleh Takagi-Sugeno Kang tahun 1985. Penalaran metode Sugeno hampir menyerupai penalaran metode Mamdani, hanya ouputnya tidak berupa himpunan *fuzzy*, tetapi berupa konstanta atau persamaan linier [8].

Type equation here. $\text{IF } x_1 \text{ is } A_1 \text{ AND } \dots \text{ AND } x_n \text{ is } A_n \text{ THEN } y = f(x_1, \dots, x_n) \quad (2)$

Dimana f bisa berupa sembarang fungsi dari variabel-variabel masukan yang nilainya berada dalam interval variabel keluaran.

Kedua metode tersebut memiliki karakteristik yang berbeda. Metode Sugeno sering digunakan untuk membangun sistem yang menuntut hasil yang cepat. Karena perhitungannya yang sangat sederhana, maka metode ini sering digunakan untuk sistem kontrol. Kelemahan metode ini adalah hasil yang diperoleh kurang manusiawi atau masih terkesan kaku. Sedangkan metode Mamdani membutuhkan proses perhitungan yang lebih lama, tetapi hasil yang didapat lebih manusiawi.

3. Metode Penelitian

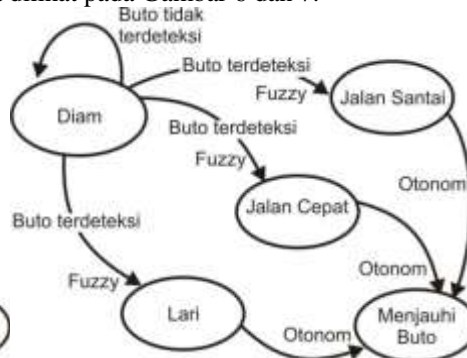
3.1. Skenario Respon Agen Otonom

Pada penelitian ini dibuat sebuah skenario yang inti cerita diambil berdasarkan dongeng atau cerita rakyat “Timun Emas”. Karakter agen otonom pertama adalah Buto, dan karakter agen otonom kedua adalah Timun Emas.

Bagian cerita Buto mengejar Timun Emas yang akan diangkat dalam penelitian ini, dengan mengesampingkan berbagai properti yang dibawa oleh Timun Emas dalam cerita sebenarnya. Ditetapkan respon perilaku mengejar yang diterapkan pada agen pertama, dan perilaku menghindari yang akan diterapkan pada agen kedua. Gambaran skenario dapat dilihat pada Gambar 6 dan 7.



Gambar 6: FSM agen pertama



Gambar 7: FSM agen kedua

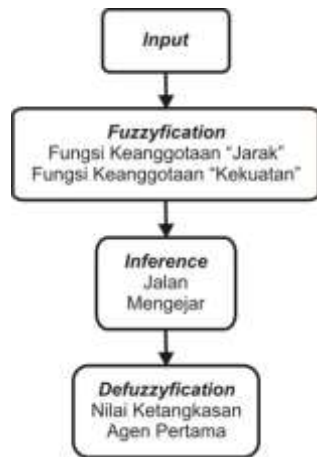
3.2. Desain Fuzzy

Logika *fuzzy* dalam penelitian ini digunakan untuk menentukan variasi respon perilaku yang dilakukan oleh dua agen otonom. Dengan adanya logika *fuzzy* tersebut masing-masing agen otonom dapat

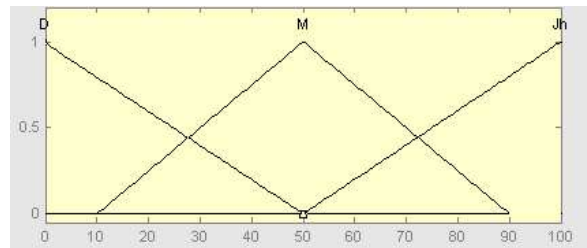
merespon perubahan variabel input menjadi perilaku yang sudah di desain sebelumnya. Metode *fuzzy* yang dipergunakan adalah metode Mamdani, *Mean of Maximum (MOM) defuzzycation*, karena metode ini menghasilkan output yang lebih manusiawi.

3.2.1. Desain *Fuzzy* Agen Pertama (Buto)

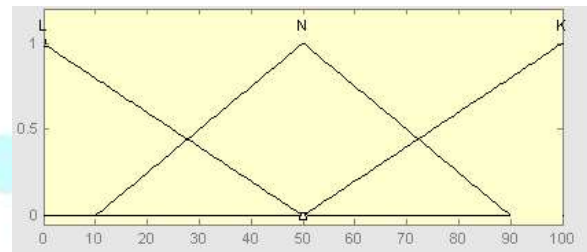
Untuk menghasilkan respon perilaku pada agen pertama ada 2 variabel yang digunakan, yaitu jarak (dekat, medium, jauh) dan kekuatan (lemah, normal, kuat). Dengan pemberian 2 variabel atribut tersebut diharapkan keluar nilai ketangkasan agen pertama yang variatif. Untuk menjadi otonom maka dipergunakan beberapa aturan sebab akibat antara ketangkasan agen pertama dengan atribut variabel yang menempel pada agen pertama. Misalnya ketika jaraknya jauh dan kekuatan lemah, maka agen pertama cenderung berjalan sesuai dengan hasil *defuzzifikasi*.



Gambar 8: Logika *fuzzy* untuk menghasilkan ketangkasan agen pertama.



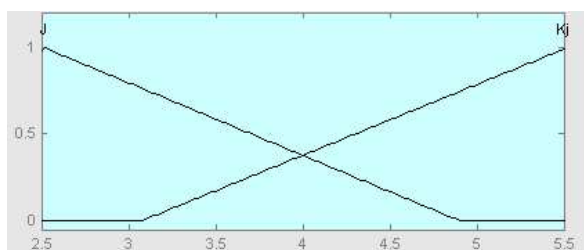
(a)



(b)

Gambar 9: Derajat keanggotaan variabel input jarak (a), dan kekuatan (b) agen pertama.

Gambar 8 menunjukkan bahwa dalam penelitian ini, dua atribut input yang diberikan, yaitu jarak dan kekuatan dalam logika *fuzzy* masing-masing menggunakan fungsi keanggotaan segitiga. Jarak diberikan tiga buah parameter masukan, yaitu: dekat (D), medium (M), dan jauh (Jh), Gambar 9 (a). Untuk kekuatan diberikan tiga buah parameter masukan, yaitu: lemah (L), normal (N), dan kuat (K), Gambar 9 (b). Hasil keluaran ketangkasan agen pertama diberikan dua buah parameter, yaitu: jalan (J) dan mengejar (Kj), Gambar 10. Variabel linguistik pada output ketangkasan agen pertama ini diambil berdasar pada penelitian yang sudah ada mengenai kecepatan jalan manusia yang rata-rata 2.5 mil/jam untuk jalan santai sampai 5.5 mil/jam untuk lari [1]. Dari 3 buah parameter jarak dan 3 buah parameter kekuatan, didapat 9 aturan *fuzzy* untuk respon ketangkasan agen pertama seperti terlihat pada Gambar 11.



Gambar 10: Output ketangkasan agen pertama

Tabel 1: Aturan *fuzzy* menghasilkan ketangkasan agen pertama

| Kekuatan | Jarak | | |
|----------|----------|----|----|
| | Variabel | D | M |
| L | Kj | J | J |
| N | Kj | Kj | J |
| K | Kj | Kj | Kj |

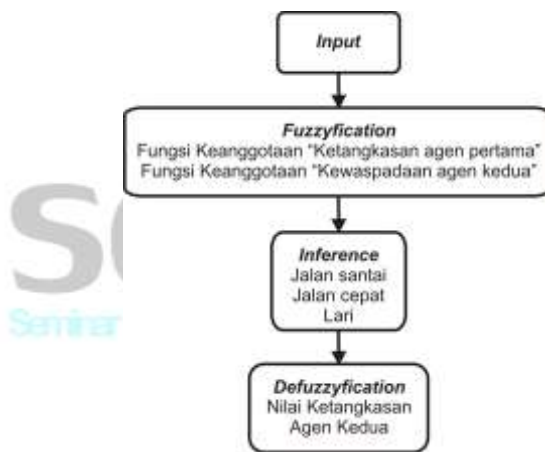
1. If (jarak is D) and (kekuatan is L) then (ketangkasanbuto is Kj) (1)
2. If (jarak is D) and (kekuatan is N) then (ketangkasanbuto is Kj) (1)
3. If (jarak is D) and (kekuatan is K) then (ketangkasanbuto is Kj) (1)
4. If (jarak is M) and (kekuatan is L) then (ketangkasanbuto is J) (1)
5. If (jarak is M) and (kekuatan is N) then (ketangkasanbuto is Kj) (1)
6. If (jarak is M) and (kekuatan is K) then (ketangkasanbuto is Kj) (1)
7. If (jarak is Jh) and (kekuatan is L) then (ketangkasanbuto is J) (1)
8. If (jarak is Jh) and (kekuatan is N) then (ketangkasanbuto is J) (1)
9. If (jarak is Jh) and (kekuatan is K) then (ketangkasanbuto is Kj) (1)

Gambar 11: Aturan *fuzzy* untuk respon ketangkasan agen pertama

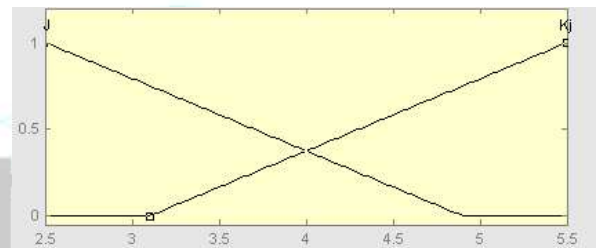
Dari 9 aturan *fuzzy* respon perilaku agen pertama yang telah dihasilkan, peluang prosentase gerakan agen diharapkan 0.9 % dari keseluruhan aturan dapat berhasil diaplikasikan menjadi respon perilaku agen pertama dalam AR.

3.2.2. Desain *Fuzzy* Agen Kedua (Timun Emas)

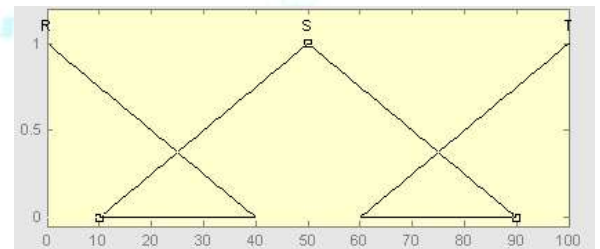
Untuk menghasilkan respon perilaku pada agen kedua ada 2 variabel yang digunakan, yaitu ketangkasan agen pertama (jalan, mengejar) dan kewaspadaan agen kedua (rendah, sedang, tinggi). Dengan pemberian 2 variabel atribut tersebut diharapkan keluar nilai ketangkasan agen kedua yang variatif. Untuk menjadi otonom maka dipergunakan beberapa aturan sebab akibat antara ketangkasan agen kedua dengan atribut variabel yang menempel pada agen kedua. Misalnya ketika ketangkasan agen pertama mengejar, dan kewaspadaan agen kedua rendah, maka agen kedua cenderung berjalan santai sesuai dengan hasil *defuzzyfikasi*.



Gambar 12: Logika *fuzzy* untuk menghasilkan ketangkasan agen pertama.



(a)

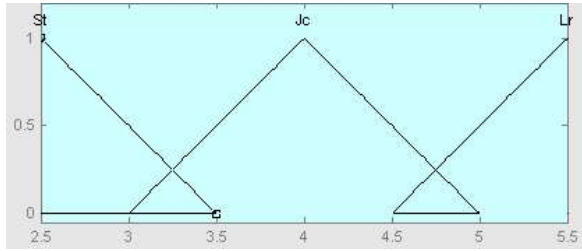


(b)

Gambar 13: Derajat keanggotaan variabel input ketangkasan agen pertama (a), dan kewaspadaan agen kedua (b).

Gambar 12 menunjukkan bahwa dalam penelitian ini, dua atribut input yang diberikan, yaitu ketangkasan agen pertama dan kewaspadaan agen kedua dalam logika *fuzzy* masing-masing menggunakan fungsi keanggotaan segitiga. Atribut ketangkasan agen pertama diberikan dua buah parameter masukan, yaitu: jalan (J), dan mengejar (Kj) Gambar 13 (a). Untuk atribut kewaspadaan agen kedua diberikan tiga buah parameter masukan, yaitu: rendah (R), sedang (S), dan tinggi (T), Gambar 12 (b). Hasil keluaran ketangkasan agen kedua diberikan tiga buah parameter, yaitu: jalan santai (St), jalan cepat (Jc), dan Lari (Lr), Gambar 14. Variabel linguistik pada output ketangkasan agen kedua ini diambil berdasar pada penelitian yang sudah ada mengenai kecepatan jalan manusia yang rata-rata 2.5 mil/jam untuk jalan santai sampai 5.5 mil/jam untuk lari [1]. Dari 2 buah parameter ketangkasan agen pertama dan 3 buah parameter kewaspadaan agen kedua, didapat 6 aturan *fuzzy* untuk ketangkasan agen kedua seperti terlihat pada Gambar 15.

Tabel 2: Aturan *fuzzy* menghasilkan ketangkasan agen kedua



Gambar 14: Output ketangkasannya agen kedua

| | | Ketangkasannya Agen I | | |
|------------------------|---|-----------------------|----|----|
| | | Variabel | J | Kj |
| Kewaspadaannya Agen II | R | St | Jc | |
| | S | Jc | Lr | |
| | T | Lr | Lr | |
| | | | | |

1. If (ketangkasannya buto is J) and (kewaspadaannya timun is R) then (ketangkasannya timun is St) (1)
2. If (ketangkasannya buto is J) and (kewaspadaannya timun is S) then (ketangkasannya timun is Jc) (1)
3. If (ketangkasannya buto is J) and (kewaspadaannya timun is T) then (ketangkasannya timun is Lr) (1)
4. If (ketangkasannya buto is Kj) and (kewaspadaannya timun is R) then (ketangkasannya timun is Jc) (1)
5. If (ketangkasannya buto is Kj) and (kewaspadaannya timun is S) then (ketangkasannya timun is Lr) (1)
6. If (ketangkasannya buto is Kj) and (kewaspadaannya timun is T) then (ketangkasannya timun is Lr) (1)

Gambar 15: Aturan fuzzy untuk respon ketangkasannya agen kedua

Dari 6 aturan fuzzy respon perilaku agen kedua yang telah dihasilkan, peluang prosentase gerakan agen adalah 0.6 % dari keseluruhan aturan dapat berhasil diaplikasikan menjadi respon perilaku agen kedua dalam AR.

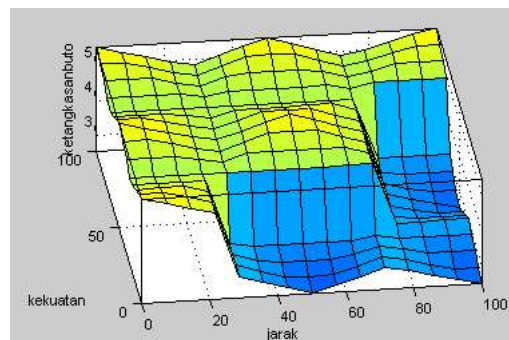
3.3. Pengujian Sistem Fuzzy

3.3.1. Pengujian Fuzzy Ketangkasannya Agen Pertama

Untuk mendapatkan variasi ketangkasannya agen pertama, diberikan pengujian parameter masukan yang berbeda-beda dari dua variabel yang telah ditetapkan.

Tabel 3: Hasil pengujian ketangkasannya agen pertama dengan menggunakan beberapa parameter masukan yang berbeda.

| | | Jarak | | | | | | | | | | |
|----------|-----|-------|------|------|------|------|------|------|------|------|------|------|
| | | 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| Kekuatan | 0 | 5.5 | 5.26 | 5.02 | 3.1 | 2.8 | 2.53 | 2.8 | 3.1 | 2.98 | 2.74 | 2.53 |
| | 10 | 5.28 | 5.26 | 5.02 | 3.1 | 2.8 | 2.74 | 2.8 | 3.1 | 2.98 | 2.74 | 2.74 |
| | 20 | 5.04 | 5.04 | 5.02 | 3.1 | 2.98 | 2.98 | 2.98 | 3.1 | 2.98 | 2.98 | 2.98 |
| | 30 | 4.92 | 4.92 | 4.92 | 4.92 | 4.92 | 4.92 | 4.92 | 4.92 | 3.1 | 3.1 | 3.1 |
| | 40 | 5.21 | 5.21 | 5.02 | 4.92 | 5.21 | 5.21 | 5.21 | 4.92 | 2.98 | 2.8 | 2.8 |
| | 50 | 5.5 | 5.26 | 5.02 | 4.92 | 5.21 | 5.5 | 5.21 | 4.92 | 2.98 | 2.74 | 2.53 |
| | 60 | 5.21 | 5.21 | 5.02 | 4.92 | 5.21 | 5.21 | 5.21 | 4.92 | 2.98 | 2.8 | 2.8 |
| | 70 | 4.92 | 4.92 | 4.92 | 4.92 | 4.92 | 4.92 | 4.92 | 4.92 | 3.1 | 3.1 | 3.1 |
| | 80 | 5.04 | 5.04 | 5.02 | 4.92 | 5.04 | 5.04 | 5.04 | 4.92 | 5.04 | 5.04 | 5.04 |
| | 90 | 5.28 | 5.26 | 5.02 | 4.92 | 5.21 | 5.28 | 5.21 | 4.92 | 5.04 | 5.28 | 5.28 |
| 100 | 5.5 | 5.26 | 5.02 | 4.92 | 5.21 | 5.5 | 5.21 | 4.92 | 5.04 | 5.28 | 5.5 | |



Gambar 16: Respon *fuzzy* ketangkasan agen pertama dalam grafik permukaan

Dari hasil pengujian Tabel 3 di atas, dapat diketahui:

- Jika kekuatan minimum (0) dan jarak jauh (100), maka ketangkasan agen pertama akan melakukan respon untuk berjalan, terbukti dengan kecepatan ditunjukkan dengan angka 2.53 mil/jam.
- Jika kekuatan maksimum (100) dan jarak dekat (0), maka ketangkasan agen pertama akan melakukan respon untuk lari mengejar, terbukti dengan kecepatan ditunjukkan dengan angka 5.5 mil/jam.
- Jika kekuatan normal (50) dan jarak medium (50), maka ketangkasan agen pertama akan melakukan respon lari mengejar, terbukti dengan kecepatan ditunjukkan dengan angka 5.5 mil/jam.
- Diperoleh 2 kelompok model respon perilaku agen pertama yaitu jalan dan lari mengejar.

Respon keluaran *fuzzy* perilaku agen pertama terhadap variasi masing-masing input direpresentasikan oleh grafik permukaan pada Gambar 16.

3.3.2. Pengujian Fuzzy Ketangkasan Agen Kedua

Untuk mendapatkan variasi ketangkasan agen kedua, diberikan pengujian parameter masukan yang berbeda-beda dari dua variabel yang telah ditetapkan.

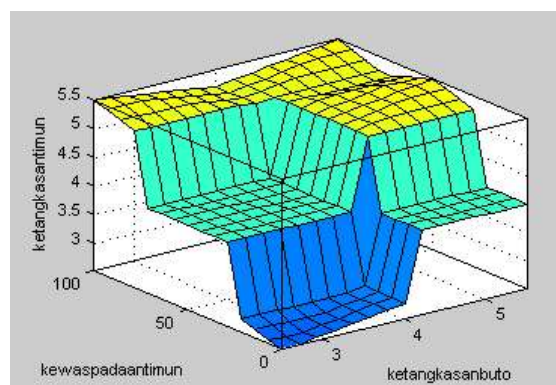
Dari hasil pengujian Tabel 4 dapat diketahui:

Diperoleh 3 kelompok model respon perilaku agen kedua yaitu jalan santai direpresentasikan dalam kelompok dengan kecepatan rata-rata 2 mil/jam, jalan cepat direpresentasikan dalam kelompok dengan kecepatan 4 mil/jam, dan lari menjauh direpresentasikan dalam kelompok dengan kecepatan rata-rata 5 mil/jam.

Respon keluaran *fuzzy* perilaku agen kedua terhadap variasi masing-masing input direpresentasikan oleh grafik permukaan pada Gambar 17.

Tabel 4: Hasil pengujian ketangkasan agen kedua dengan menggunakan beberapa parameter masukan yang berbeda.

| | | Ketangkasan Agen I | | | | | | |
|---------------------|-----|--------------------|------|------|------|------|------|------|
| | | 2.5 | 3 | 3.5 | 4 | 4.5 | 5 | 5.5 |
| Kewaspadaan Agen II | 0 | 2.52 | 2.59 | 2.69 | 2.8 | 4 | 4 | 4 |
| | 10 | 2.62 | 2.62 | 2.69 | 2.8 | 4 | 4 | 4 |
| | 20 | 2.74 | 2.74 | 2.74 | 2.8 | 4 | 4 | 4 |
| | 30 | 4 | 4 | 4 | 4 | 5.26 | 5.26 | 5.26 |
| | 40 | 4 | 4 | 4 | 4 | 5.3 | 5.38 | 5.38 |
| | 50 | 4 | 4 | 4 | 4 | 5.3 | 5.41 | 5.5 |
| | 60 | 4 | 4 | 4 | 4 | 5.3 | 5.38 | 5.38 |
| | 70 | 4 | 4 | 4 | 4 | 5.26 | 5.26 | 5.26 |
| | 80 | 5.26 | 5.26 | 5.26 | 5.26 | 5.26 | 5.26 | 5.26 |
| | 90 | 5.38 | 5.38 | 5.3 | 5.2 | 5.3 | 5.38 | 5.38 |
| | 100 | 5.48 | 5.41 | 5.3 | 5.2 | 5.3 | 5.41 | 5.5 |



Gambar 17: Respon *fuzzy* ketangkasan agen kedua dalam grafik permukaan

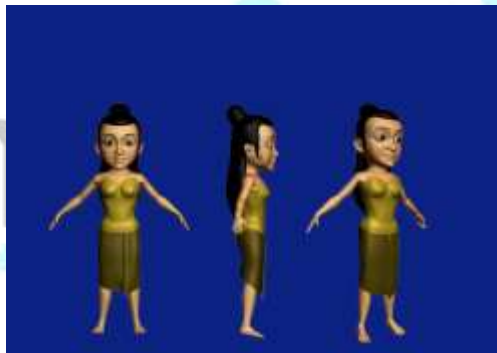
3.4. Karakter 3D Agen

3.4.1. Agen Pertama



Gambar 18: Karakter 3D agen pertama

3.4.2. Agen Kedua



Gambar 19: Karakter 3D agen kedua

4. Kesimpulan

Dari hasil pengujian untuk mendapatkan respon perilaku ketangkasan agen pertama dan agen kedua dalam penelitian ini dapat diperoleh beberapa kesimpulan antara lain:

1. Aturan *fuzzy* dapat dipakai untuk menghasilkan respon perilaku untuk masing-masing agen sesuai dengan variabel yang dimiliki. Terbukti dengan hasil pengujian pada masing-masing respon agen dengan menggunakan beberapa parameter yang berbeda menunjukkan pola sesuai dengan yang diharapkan. Ketangkasan agen pertama menghasilkan 2 pola respon, yaitu jalan dan lari mengejar. Sedangkan ketangkasan agen kedua menghasilkan 3 pola respon, yaitu jalan santai, jalan cepat dan lari menjauh.
2. Untuk mendapatkan hasil perhitungan *defuzzyfication* yang optimal pada penelitian ini digunakan model MOM *defuzzyfication* karena dengan memakai model yang lain nilai hasil *defuzzyfication* tidak seperti yang diharapkan.

Penelitian ini akan dilanjutkan dengan menggabungkan data hasil *defuzzyfication* ke dalam proses AR.

Daftar Pustaka

- [1] Aspelin, Karen, "Establishing Pedestrian Walking Speed", Karen Aspelin, P.E., P.T.O.E., ITE District 6 Technical Chair Parsons Brinckerhoff Albuquerque, New Mexico, Portland State University, 2005.
- [2] Buckland, M., "Programming Game AI by Example", Worldware Publishing, 2005.

- [3] Cawood Stephen, Fiala Mark, "Augmented Reality: A Practical Guide", The Pragmatic Programmers, LLC. <http://media.pragprog.com/titles/cfar>, 2008.
- [4] Ian Millington, "Artificial Intelligence for Games", Morgan Kaufmann Publisher Inc., 2006.
- [5] I.E. Sutherland, "The Ultimate Display," invited lecture, IFIP Congress 65. An abstract appears in Information Processing 1965: Proc. IFIP Congress 65, Vol. 2, W.A. Kalenich, ed., Spartan Books, Washington, D.C., and Macmillan, New York, pp. 506-508. A similar, but quite complete, paper is Sutherland's "Computer Displays," Scientific American, Vol. 222, No. 6, June, pp. 57-81, 1970.
- [6] Jaafar, Jafreezal and McKenzie, Eric., "Behavior Coordination of Virtual Agent Navigation using Fuzzy Logic", IEEE, Institute of Perception, Action and Behavior, School of Informatics, University of Edinburgh, UK, 2006.
- [7] Kato, Hirokazu, ARToolKit Version 2.33, <http://www.tinmith.net/lca2004/ARToolkit/ARToolKit2.33doc.pdf>, 2000.
- [8] Kusumadewi Sri, Purnomo Hari, "Aplikasi Logika Fuzzy untuk Pendukung Keputusan", Graha Ilmu, 2010.
- [9] Lindeberg Tony, "Scale-Space Theory in Computer Vision", Royal Institute of Technology Stockholm, Sweden, 1993.
- [10] Lori L. Delooze and Wesley R.Viner, "Fuzzy Q-Learning in a Nondeterministic Environment: Developing an Intelligent Ms.Pac-Man Agent", IEEE, United State Naval Academy, Annapolis, USA, 2009.
- [11] Maes, P., "Modeling Adaptive Autonomous Agents", dalam MIT Media-Laboratory 20 Ames Street Rm 305 Cambridge, MA 02139, 1994.
- [12] Reynolds, CW., "Steering Behaviors For Autonomous Characters", Sony Computer Entertainment America 919 East Hillsdale Boulevard Foster City, California 94404, 1999.
- [13] Shahzad, Malik, "Robust Registration of Virtual Objects for Real-Time Augmented Reality", Thesis, The Ottawa-Carleton Institute for Computer Science, Canada, 2002..
- [14] T. Azuma Ronald, "A Survey of Augmented Reality", Hughes Research Laboratories, Malibu, California.
- [15] Thalmann, D., Musse, S.R., Kallmann, M., "Virtual Humans Behaviour: Individuals, Groups and Crowds", dalam Proceeding of Digital Media Future, 1999.

