

MEMBANDINGKAN ALGORITMA DJIKSTRA DENGAN ALGORITMA A*STAR MENGGUNAKAN METODE LOGIKA FUZZY DI UNITY 3D

COMPARING THE DJIKSTRA ALGORITHM WITH THE A*STAR ALGORITHM USING THE FUZZY LOGIC METHOD IN UNITY 3D

Mohammad Yasrid Angkat, Andrew Brian Osmond², Anton Siswo Raharjo ansori³

¹Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom

²Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom

³Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom

¹abosmond@student.telkomuniversity.ac.id, ²raharjo@telkomuniversity.co.id,

³yasridangkat@student.telkomuniversity.ac.id

Abstrak

Perkembangan dunia *Game* dari masa ke masa terus mengalami perkembangan, hingga kini bermunculan beberapa Genre *Game* dari text based hingga yang memiliki animasi 3D. Sebuah *Game* tidak hanya berkembang pada Interface saja, kemampuan *Game* untuk menjadi agen learning menjadikan sebuah *Game* itu lebih hidup. Agen learning dalam *Game* dikenal sebagai *Artificial intelegent* (AI). AI tersebut di kembangkan untuk merancang perilaku *Non-Player Character* (NPC) dari *Game* atau simulasi yang seakan-akan NPC tersebut mempunyai kecerdasan dan pergerakan sealami mungkin. Penelitian ini membahas untuk membandingkan 2 metode yaitu algoritma djikstra dengan A*star menggunakan metode logika fuzzy, sehingga kita dapat membandingkan 2 alogiritma dari pergerakan suatu NPC di dalam *Game* tersebut. *Game* dibuat menggunakan *Game engine* Unity 3D. Hasil uji coba implementasi pada NPC disesuaikan dengan perilaku yang normal, yaitu bergerak kearah yang ditentukan atau jalan secara tidak menentu..

Kata Kunci : Simulasi, AI, NPC, Logic Fuzzy

Abstract

The development of the Game world from time to time continues to develop, until now there are several Game genres emerging from text based to those that have 3D animation. A Game not only develops on the interface, the Game's ability to become an agent of learning makes a Game more alive. In-Game learning agents are known as Artificial intelegent (AI). The AI was developed to design the behavior of Non-Player Character (NPC) from Games or simulations as if the NPC had intelligence and movement as natural as possible. This study discusses making the movement of Non-Player Character NPC in the Game using the Fuzzy algorithm method, so that we can determine the direction of the movement of an NPC in the Game. The Game is made using the Unity 3D Game engine, and determines the direction of its movement using FuSM. The results of the FuSM implementation test on the NPC are adjusted to normal behavior, which is moving towards the specified direction or the road is uncertain.

Key Word : Dijkstra Algorithm, Game, NPC..

1. Pendahuluan

Game telah menjadi satu hal yang ada di dalam keseharian kita. Perkembangan *game* disetiap masa semakin menunjukkan keeksistensiannya. Pada abad sekarang industri *game* semakin pesat perkembangannya. *Game* menjadi sebuah kebutuhan yang cukup esensial bagi beberapa kalangan,

terutama bagi kalangan anak – anak, remaja dan pemuda. Dahulu, *game* hanya dijadikan sarana hiburan semata namun sekarang *game* telah menjadi luas fungsinya, misalnya *game* dapat dijadikan sarana pembelajaran, lahan bisnis, dan dipertandingkan sebagai salah satu dari cabang olahraga oleh para profesional. Didalam *game* sendiri terdapat beberapa karakter yang berfungsi untuk membantu atau meramaikan suasana didalam *game* tersebut yaitu adalah *Non-Player Character* (NPC).

Non-Player Character (NPC) berinteraksi dengan para pemain, dan memainkan peran yang penting dalam berbagai jenis permainan. Ketika *Non-Player Character* (NPC) merespons reaksi irasional atau menduplikasi pola perilaku tetap, atau tingkat kesulitan dalam permainan terlalu sederhana atau sulit, akan menghilangkan ketidakpuasan umum dari pemain *Game* dan mengurangi nilai hiburan yang ada di dalam *Game* tersebut [1].

Dalam pembuatan sebuah *game* sering dibutuhkan suatu kecerdasan buatan untuk dapat membuat *game* tersebut lebih menarik. Kecerdasan buatan ini sering digunakan terhadap *Non-Player Character* (NPC) untuk dapat membuat suatu perilaku yang cerdas dan dapat mensimulasikan kecerdasan manusia, contohnya seperti NPC yang dapat mengikuti player kemana saja dan dapat menghindari suatu rintangan yang ada secara otomatis, dan juga NPC tersebut dapat mengikuti kecepatan player seperti pada saat NPC berada jauh pada player, maka NPC akan berlari dengan cepat menuju player dan jika NPC tersebut berada didekat player, maka NPC akan berjalan sesuai dengan kecepatan player. Di dalam *game* yang akan dibangun ini fokus kecerdasan buatan akan dilakukan untuk membuat sistem kecepatan dari NPC dan metode yang akan dipakai adalah metode logika *fuzzy sugeno*.

2. Dasar Teori

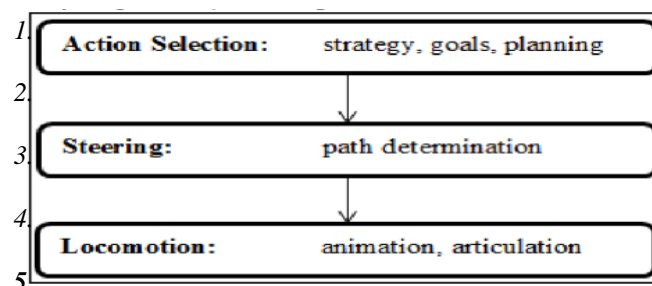
2.1 Non Player Character (NPC)

NPC (*Non-player character*) merupakan jenis *autonomus* yang di tujukan untuk penggunaan komputer animasi dan media interaktif seperti *game* dan *virtual reality*. Agen ini mewakili tokoh dalam cerita atau permainan dan memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah sebuah kebalikan dari tokoh animasi dari sebuah film animasi yang tindakannya di tulis dimuka, dan untuk “*avatar*” dalam sebuah permainan atau *virtual reality*, tindakan yang di arahkan pemain secara *real time* oleh pemain. Dalam permainan, karakter otonom biasanya disebut NPC (*Non-player character*)[2]. Perilaku *Non-player character* dibuat semirip mungkin dengan realistik. Semakin banyak sifat yang dapat di lakukan karakter maka semakin banyak pula gerakan yang dapat dilakukan karakter. [3].

Untuk memprediksi model dalam peta akan digunakan *self-organizing map*. Kombinasi algoritma *fuzzy sugeno* akan mempengaruhi peta selanjutnya dan memungkinkan untuk menemukan jalur terpendek antara dua posisi di peta berdasarkan nilai berat hambatan di area peta. Raynold (1999) membagi perilaku NPC menjadi tiga lapisan yaitu :

- Seleksi Tindakan (*action selection*)
- Kendali (*steering*)
- Penggerak (*locomotion*)

Sesuai seperti yang ada di Gambar 2. 1 berikut:



Gambar 1. 1 Hirarki Gerak Perilaku

Non-player character dari *game* simulasi yang seakan – akan NPC tersebut mempunyai kecerdasan dan pergerakan sealam mungkin. Keberadaan NPC sendiri dalam suatu *game* merupakan salah satu faktor dan komponen penting dalam permainan komputer modern yang dapat menentukan permainan tersebut menjadi menarik atau tidak. Konsep agen cerdas merupakan salah satu model yang digunakan dalam membuat NPC. Sifat otonom dari agen cerdas merupakan keunggulan dalam memodelkan satu NPC *game*. Salah satu model atau kecerdasan buatan yang dapat digunakan dalam menentukan kecepatan pergerakan NPC yaitu algoritma *fuzzy sugeno*. *Non Playable Character* (NPC) mempunyai 4 jenis yang menjadi acuan, yaitu : NPC *Partner*, NPC *Enemy*, NPC *Quest*, NPC Pendukung Cerita. Keempat NPC tersebut dapat berada dalam satu permainan tergantung jenis dan genre *game* yang dimainkan.

2.2 Kecerdasan Buatan

Kecerdasan Buatan merupakan salah satu 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. Untuk mengetahui dan memodelkan proses–proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan ditambah pengalaman, penalaran (bagaimana membuat keputusan dan mengambil tindakan), moral yang baik[5].

Manusia cerdas (pandai) dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki tentu akan lebih mampu menyelesaikan permasalahan. Tapi bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang dimiliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik[6].

2.3 Logika Fuzzy

Logika *fuzzy* adalah seperangkat teknik matematika yang dirancang untuk mengatasi area abu-abu. Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multichannel* atau *workstation* berbasis akuisisi data dan sistem kontrol[7]. Ada beberapa alasan mengapa orang menggunakan logika *fuzzy*, antara lain

- (1) Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
- (2) Logika *fuzzy* sangat *fleksibel*.
- (3) Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
- (4) Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.
- (5) Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
- (6) Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendal secara konvensional.
- (7) Logika *fuzzy* didasarkan pada bahasa alam[8].

2.4 Metode Fuzzy sugeno

Penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja output (konsekuen) sistem tidak berupa himpunan *Fuzzy*, melainkan berupa konstanta atau persamaan linear. Michio Sugeno mengusulkan penggunaan *singleton* sebagai fungsi keanggotaan dari konsekuen. *Singleton* adalah sebuah himpunan *Fuzzy* dengan fungsi keanggotaan yang pada titik tertentu mempunyai sebuah nilai dan 0 di luar titik tersebut. Ada 2 model *Fuzzy* metode Sugeno yaitu sebagai berikut:

a. Model *Fuzzy* Sugeno Orde-Nol

Secara umum bentuk model *Fuzzy* Sugeno Orde Nol adalah: IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $z = k$ dengan A_i adalah himpunan *Fuzzy* ke- i sebagai antesenden, dan k adalah suatu konstanta sebagai konsekuen.

b. Model *Fuzzy* Sugeno Orde-Satu

Secara umum bentuk model *Fuzzy* Sugeno Orde-Satu adalah: IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $z = p_1 * x_1 + \dots + p_N * x_N + q$ dengan A_i adalah himpunan *Fuzzy* ke- i sebagai antesenden, dan p_i adalah suatu konstanta ke- i dan q juga merupakan konstanta dalam konsekuen.

Berdasarkan model *Fuzzy* tersebut, ada tahapan-tahapan yang harus dilakukan dalam implementasi metode Sugeno yaitu sebagai berikut:

1) Pembentukan *Fuzzy*

Pada tahapan ini variabel input dari system *Fuzzy* ditransfer ke dalam himpunan *Fuzzy* untuk dapat digunakan dalam perhitungan nilai kebenaran dari premis pada setiap aturan dalam basis pengetahuan. Dengan demikian tahap ini mengambil nilai-nilai tegas dan menentukan derajat di mana nilai-nilai tersebut menjadi anggota dari setiap himpunan *Fuzzy* yang sesuai.

2) Aplikasi fungsi implikasi

Tiap-tiap aturan (proposisi) pada basis pengetahuan *Fuzzy* akan berhubungan dengan suatu relasi *Fuzzy*. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah sebagai berikut: IF $x \text{ is } A$ THEN $y \text{ is } B$ dengan x dan y adalah skalar, dan A dan B adalah himpunan *Fuzzy*. Proposisi yang mengikuti IF disebut sebagai antesenden sedangkan proposisi yang mengikuti THEN disebut konsekuen. Proposisi ini dapat diperluas dengan menggunakan operator *Fuzzy* seperti, IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $y \text{ is } B$ dengan \circ adalah operator (misal: OR atau AND). Secara umum fungsi implikasi yang dapat digunakan yaitu sebagai berikut:

- Min (*minimum*) Fungsi ini akan memotong output himpunan *fuzzy*.
- Dot (*product*) Fungsi ini akan menskala output himpunan *fuzzy*.

Pada metode Sugeno ini, fungsi implikasi yang digunakan adalah fungsi min.

c. Defuzzifikasi

Defuzzifikasi (*Defuzzification*) *Input* dari proses defuzzifikasi adalah himpunan *fuzzy* yang dihasilkan dari proses komposisi dan *output* adalah sebuah nilai. Untuk aturan IF-THEN *fuzzy* dalam persamaan.

$RU(k) = \text{IF } x_1 \text{ is } A_{1k} \text{ and } \dots \text{ and } x_n \text{ is } A_{nk} \text{ THEN } y \text{ is } B_k$, dimana A_{1k} dan B_k berturut-turut adalah himpunan *Fuzzy* dalam $U_i R$ (U dan V adalah domain fisik), $i = 1, 2, \dots, n$ dan $x = (x_1, x_2, \dots, x_n)$ U dan y berturut-turut adalah variabel input dan output (linguistik) dari sistem *fuzzy*. Defuzzifier pada persamaan di atas didefinisikan sebagai suatu pemetaan dari himpunan *fuzzy* B ke dalam $V R$ (yang merupakan output dari inferensi *fuzzy*) ke titik tegas $y \in V$. [2]. Pada metode Sugeno *defuzzification* dilakukan dengan perhitungan *Weight Average* (WA):

$$WA = \frac{a_1 z_1 + a_2 z_2 + a_3 z_3 + \dots + a_n z_n}{a_1 + a_2 + a_3 + \dots + a_n}$$

Keterangan:

WA = Nilai rata-rata, a_n = nilai predikat aturan ke- n , dan z_n = indeks nilai output (konstanta) ke- n .

2.5 Fungsi Keanggotaan

Fungsi keanggotaan adalah grafik yang mewakili besar dari derajat keanggotaan masing-masing variabel input yang berada dalam interval antara 0 dan 1. Derajat keanggotaan sebuah

variabel x dilambangkan dengan simbol $\mu(x)$. Aturan-aturan (*Rules*) menggunakan nilai keanggotaan sebagai faktor bobot untuk menentukan pengaruhnya pada saat melakukan inferensi dalam menarik kesimpulan. Ada beberapa fungsi yang bisa digunakan namun dalam penelitian ini peneliti memakai fungsi keanggotaan kurva bahu dan kurva segitiga.

✓ Representasi Kurva Bahu

Fungsi keanggotaan yang merepresentasikan kurva bahu kiri:

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

Fungsi keanggotaan yang merepresentasikan kurva bahu kanan:

$$\mu(x) = \begin{cases} 0, & x \leq b; \\ \frac{x-b}{c-b}, & b \leq x \leq c; \\ 1, & x \geq c; \end{cases}$$

✓ Representasi kurva segitiga

Fungsi keanggotaan yang merepresentasikan kurva segitiga adalah :

$$\mu(x) = \begin{cases} 0, & x \leq a \text{ atau } x \geq c; \\ \frac{x-a}{b-a}, & a < x < b; \\ \frac{c-x}{c-b}, & b < x < c; \end{cases}$$

Keterangan :

a = nilai domain terkecil yang mempunyai derajat keanggotaan nol.

b = nilai domain yang mempunyai derajat keanggotaan satu.

c = nilai domain terbesar yang mempunyai derajat keanggotaan nol.

x = nilai input atau output yang akan diubah ke dalam bilangan *Fuzzy*.

Fungsi untuk memetakan kembali nilai *Fuzzy* menjadi nilai *crisp* yang menjadi output/nilai solusi permasalahan.

3. Implementasi Sistem

Dalam bab ini membahas mengenai implementasi metode terhadap aplikasi yang sudah dibuat dan juga pengujian metode yang di terapkan. Serta melakukan uji coba pada aplikasi yang telah dibangun, apakah telah sesuai dengan perancangan dan hasil yang diharapkan.

4. Implementasi Algoritma *fuzzy* pada Perilaku NPC

Proses implementasi adalah proses penerapan komponen system utama yang dibangun berdasarkan rancangan yang telah dibuat dan diajukan sebelumnya. Implementasi kecerdasan buatan pada penelitian ini diterapkan pada pengaturan kecepatan pada NPC dengan menggunakan metode algoritma *fuzzy* sugeno. Metode *fuzzy* sugeno ini digunakan untuk dapat mengetahui jarak dan kecepatan NPC. Posisi NPC dijadikan sebagai starting point dan karakter prajurit diinisiasi sebagai tujuan.

5. Implementasi Kecepatan

Uji coba algoritma *fuzzy* ini dilakukan untuk mengetahui kecepatan pada jarak yang ditempuh oleh NPC dan ketepatan algoritma dalam menghitung nilai kecepatan yang dihasilkan pada saat perhitungan manual dengan yang ada di program, Proses ini melakukan uji coba dengan

memenggunakan distance awal NPC dan Kecepatan yang diterapkan oleh NPC. Data-data tersebut dapat dilihat pada Tabel 4.1 Data Sampel :

Distance Awal NPC : 71,45

Kecepatan Awal NPC : 10 f

Tabel 4.1 Hasil Uji Coba Algoritma *fuzzy* terhadap NPC

NO	Jarak	Kecepatan Awal	Output Kecepatan
1	71,45	10f	10
2	55,36	8f	7,6

- Perhitungan pada jarak ke-1

$$\mu_{Jauh} [71,45] = \frac{71,45}{71,45} = 1$$

$$z = \frac{\alpha_{Jauh} (10)}{\alpha_{Jauh}}$$

$$z = \frac{1 (10)}{1} = 10f$$

- Perhitungan pada jarak ke-2

$$\mu_{Jauh} [55,36] = \frac{55,36 - 50,015}{71,45 - 50,015} = \frac{5,345}{21,43} = 0.2$$

$$\mu_{Sedang} [55,36] = \frac{71,45 - 55,36}{71,45 - 50,015} = \frac{16,09}{21,43} = 0.8$$

$$z = \frac{\alpha_{Jauh} (10) + \alpha_{Sedang} (7)}{\alpha_{Jauh} + \alpha_{Sedang}}$$

$$z = \frac{\alpha 0.2(10) + \alpha 0.8 (7)}{0,4 + 0,6} = \frac{7,6}{1} = 7,6 f$$

Setelah didapatkan nilai Kecepatan pada perhitungan *fuzzy*, maka dilakukan pengujian pada program.

7. Kesimpulan

Berdasarkan pengujian *game*, dan *perhitungan manual*, *game* ini dapat bergerak dengan kecepatan yang diharapkan dan dapat menampilkan jarak dengan sesuai, sehingga tingkat keberhasilan yang didapat adalah sebesar 100%.

Daftar Pusaka :

- [1] Putra, Fahrul Pradhana, Ahmad Zainul Fanani and Moch Hariadi. 2015. „*Perilaku Otonomi dan adaptif Non Player Character Musuh pada Game 3 Dimensi Menggunakan Fuzzy State Machine dan Rule Based System.*“. Semarang. Seminar Nasional Teknologi dan Komunikasi Terapan (SEMANTIK).
- [2] Blair Purkiss, Imran Khaliq. 2015.„*A Study of Interaction in Idle Games & Perceptions on the Definition of a Game*“, *IEEE Games Entertainment Media Conference (GEM)*,
- [3] N. Lipkin. 2013.„*Examining Indie’s Independence: The Meaning of “Indie” Games, the Politics of Production, and Mainstream Co-optation*“, *The Journal of the Canadian Game Studies Association*.

- [4] B. N. Matahari , G. G. Samuel and G. S. R, "Pemetaan Perilaku *Non-player character* Pada Permainan Berbasis Role Playing Game Menggunakan Metode Finite State Machine," *Journal of Animation and Games Studies*, Vol. 1 No. 2, 2015.
- [5] I. K. W. N. K. Billy, "Implementasi Artificial Intelligence pada Game Defender of Metal City dengan Menggunakan Finite State Machine," *TEKNIK POMITS*, vol. 6, pp. A640-A642, 2017.
- [6] Dorteus Lodewyik Rahakbauw," PENERAPAN LOGIKA FUZZY METODE SUGENO UNTUK MENENTUKAN JUMLAH PRODUKSI ROTI BERDASARKAN DATA PERSEDIAAN DAN JUMLAH PERMINTAAN",. Maluku. Jurnal Ilmu Matematika dan Terapan, 2015.
- [7] Radim Farana, Bogdan Walek, Michal Janošek, Jaroslav Žápek. 2014.,” *Fuzzy-Logic Control in Fast Technological Processes*”,. IEEE.
- [8] Setyoningsih Wibowo. 2015. ,“PENERAPAN LOGIKA FUZZY DALAM PENJADWALAN WAKTU KULIAH”,. Jurnal Informatika UPGRIS.

