

PENERAPAN ALGORITMA DIJKSTRA DAN FUZZY LOGIC SUGENO PADA GAME ZOMBIE SHOOTER

Andryano Pratama¹, Fadli Delta Rizky², Daniel Udjulawa³

³STMIK GI MDP; JL. Rajawali No.14 Palembang, 0711-376400

^{1,2}Jurusan Teknik Informatika, STMIK GI MDP, Palembang

e-mail: and.rhy17@mhs.mdp.ac.id, fadli.delta@mhs.mdp.ac.id, daniel@mdp.ac.id

Abstrak

Algoritma Dijkstra adalah algoritma yang digunakan untuk menentukan jalur terpendek dan logika fuzzy sugeno untuk mempersentasikan aturan dari yang dibuat kedalam aplikasi. Tujuan penelitian dari skripsi ini adalah menerapkan algoritma dijkstra dan logika fuzzy sugeno kedalam aplikasi. Metode penelitian yang digunakan dalam penerapan algoritma kedalam game zombie shooter yaitu metode penelitian dengan studi literatur, membandingkan dengan penelitian sejenis. Hasil yang didapatkan dari penilitan ini adalah penerapan dijkstra sesuai yaitu zombie dapat mengejar pemain dengan jalur terpendek dan logika fuzzy sugeno didapatkan tingkat keberhasilan 80%.

Kata kunci— *Zombie, Algoritma Dijkstra, Fuzzy Logic Sugeno, Unity*

Abstract

Algorithm is an algorithm used to determine the shortest path and Sugeno fuzzy logic to present the rules of which were made into the application. The research objective of this thesis is to apply Sugeno fuzzy algorithms and logic into the application method used in the application of the algorithm into a zombie shooter game which is the method the researcher with the study of literature, compared with similar research. The results obtained from research is the application of appropriate dijkstra that zombies can pursue the player with the shortest path and fuzzy logic sugeno 80% success rate.

Keywords— *Zombie, Algorithm Dijkstra, Fuzzy Logic Sugeno, Unity*

1. PENDAHULUAN

Algoritma *Dijkstra* ditemukan oleh ilmuwan komputer, Edsger Dijkstra adalah sebuah pengembangan dari algoritma *greedy* yang dipakai dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed graph*) dengan bobot-bobot sisi (*edge weights*) yang melambangkan jarak antar suatu tempat sehingga dapat digunakan untuk menemukan jarak terpendek antara dua tempat. Input algoritma ini adalah sebuah graf berarah yang berbobot (*weighted directed graph*) G & sebuah sumber *vertex* s dalam G & V adalah himpunan semua *vertices* dalam *graph* G . Setiap sisi dari graf ini adalah pasangan *vertices* (u,v) yang melambangkan hubungan dari *vertex* u ke *vertex* v dan semua himpunan tepi disebut $E[1]$.

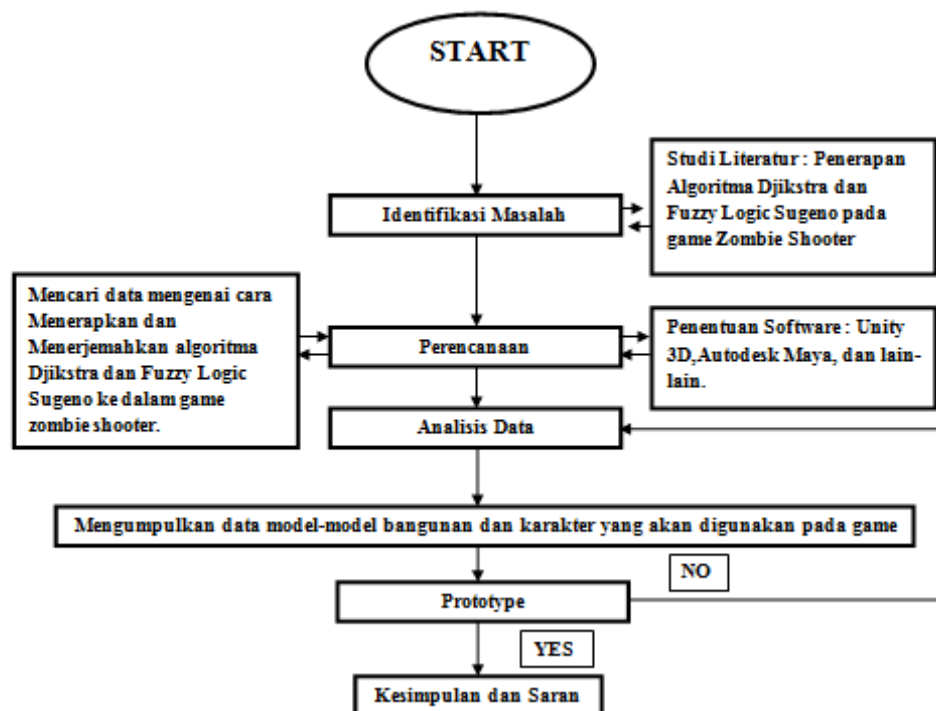
Logika *fuzzy* adalah salah satu komponen pembentuk *soft computing*. Logika *fuzzy* pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. *Fuzzy logic* banyak digunakan karena mirip dengan cara berpikir manusia. Sistem *fuzzy logic* dapat merepresentasikan

pengetahuan manusia dalam bentuk matematis, seperti mengolah nilai yang tidak pasti berupa batasan, seperti “sangat”, “sedikit”, dan “kurang lebih.”[1].

Game Zombie Shooter yang ada saat ini kebanyakan masih berbasis *desktop* untuk memainkannya membutuhkan perangkat komputer. Sedangkan *game zombie shooter* berbasis *android* sehingga dapat dimainkan dengan *smartphone android*. Selain itu tema atau konsep *game Zombie Shooter* mengangkat tema seputar melawan *zombie* yang terinfeksi *virus* dan pada karakter *player* menggambarkan tokoh tentara. Tentara sendiri adalah suatu profesi yang mempunyai nilai dedikasi yang tinggi terhadap bangsa dan negaranya. Sehingga pemain dapat merasakan perjuangan tentara dalam memusnahkan *zombie*. Pemain bertugas menyelamatkan dunia dari serangan *zombie*, meskipun pemain bermain melawan komputer sebagai musuh, bukan berarti mudah untuk mengalahkannya, karena lawan main atau komputer dalam *game zombie shooter* telah diterapkan algoritma *Dijkstra* pada *zombie* untuk menemukan jalur terpendek dalam mencari *player* dan *fuzzy* untuk mengatur perilaku NPC (Non-Player Character) musuh dengan kriteria yang telah ditentukan. Penerapan algoritma pada *game* ini sangat dibutuhkan untuk membuat permainan lebih menarik dan menantang untuk dimainkan.

2. METODE PENELITIAN

Metode adalah aspek yang sangat penting dan besar pengaruhnya terhadap berhasil tidaknya suatu penelitian, terutama untuk mengumpulkan data. Sebab data yang diperoleh dalam suatu penelitian merupakan gambaran dari objek penelitian. Metodologi penelitian dapat dilihat pada gambar 1.1.



Gambar 1 Diagram Metodologi Penelitian

2.1 Identifikasi Masalah

Pada tahap ini yang dilakukan adalah mencari judul yang baik dan cocok untuk diangkat menjadi sebuah skripsi. Dalam tahap ini juga memulai untuk mencari dan membandingkan beberapa jurnal yang terkait.

2.2 Perencanaan

Tahap perencanaan adalah tahap yang digunakan dalam penelitian ini untuk membuat konsep pembuatan game, mulai dari menentukan software dan peralatan yang dibutuhkan untuk mendukung pembuatan game, mencari data – data tentang game yang akan dibuat meliputi gambar model – model bangunan dan karakter. Mencari data mengenai cara – cara dalam menerapkan algoritma dijkstra dan fuzzy logic kedalam game zombie shooter.

2.3 Analisis Data

Pada tahap ini, mengumpulkan data-data apa saja yang dibutuhkan dalam pembuatan game. Baik pengumpulan data berupa data tentang software yang akan dipakai, data mengenai model-model bangunan, pohon dan karakter yang akan dipergunakan untuk objek didalam game

2.4 Prototype

Pada tahap ini, penulis melakukan enam tahapan :

2.4.1 Analisis Kebutuhan

Pada tahap ini dilakukan analisis terhadap data yang akan digunakan untuk game seperti gambar model bangunan dan karakter.

2.4.2 Mendesain Prototype

Pada tahap ini dilakukan perancangan game seperti merancang skenario game, perancangan peta, tempat, karakter player dan zombie.

2.4.3 Membangun game

Membangun game merupakan tahap yang digunakan untuk menyatuhkan rancangan game sebelumnya menjadi satu dan membuat source code algoritma dijkstra dan fuzzy sugeno untuk digunakan didalam game.

2.4.4 Implementasi Game

Menerapkan source code algoritma yang telah dibuat sebelumnya kedalam game yang telah selesai didesain sebelumnya.

2.4.5 Evaluasi Prototype

Pada tahap ini dilakukan evaluasi terhadap rancangan dan algoritma telah berjalan sesuai yang diharapkan atau tidak. Jika telah sesuai maka akan masuk ketahap selanjutnya dan jika belum sesuai maka akan mengulang ketahap analisis kebutuhan sampai dengan selesai.

2.4.6 Pengujian Prototype

Tahapan terakhir dalam metodologi pengembangan ini adalah melakukan pengujian terhadap permainan yang telah selesai dibangun berdasarkan rancangan yang telah dibuat.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengujian Algoritma *Dijkstra* dan *Fuzzy Sugeno*

3.1.1 Pengujian Algoritma *Dijkstra*

Pengujian pada algoritma *Dijkstra* menggunakan teknik pengujian *black-box*. Teknik pengujian ini hanya menguji fungsionalitas algoritma pada game. Hasil pengujian algoritma *dijkstra* pada *zombie* dalam game dapat dilihat pada tabel 3. :

Tabel 3 Hasil Pengujian Algoritma *Dijkstra*

No	Uji Coba	Hasil yang Diharapkan	Hasil
1	<i>Zombie</i> dapat mendeteksi dan mengejar <i>player</i>	Ketika <i>Player</i> mendekati <i>zombie</i> maka <i>zombie</i> dapat mendeteksi dimana posisi <i>player</i> dan akan mengejar melalui jalur terpendek tanpa	Berhasil

		melewati hambatan..	
2	<i>Zombie</i> dapat mengejar <i>player</i> secara terus menerus.	<i>Zombie</i> akan mengejar <i>player</i> secara terus menerus melalui jalur terpendek tanpa melewati hambatan.	Berhasil

3.1.2 Pengujian Algoritma *Fuzzy* Sugeno

Pengujian logika *fuzzy* dilakukan dengan cara membandingkan hasil sistem *game* yang menggunakan *fuzzy* dan hasil sistem yang tidak menggunakan *fuzzy* kemudian dibandingkan dengan *rule fuzzy* yang telah dibuat. Hasil pengujian dapat dilihat pada tabel 4

Tabel 4 Hasil Pengujian *Fuzzy* Sugeno

No	Input		Output Aplikasi Menggunakan <i>fuzzy</i>	Aturan Manual	Keterangan
	Jenis Peluru	Jarak Tembak			
1	11	7	lemah	lemah	Sesuai
2	11	12	lemah	lemah	Sesuai
3	11	20	lemah	lemah	Sesuai
4	30	7	sedang	sedang	Sesuai
5	30	12	lemah	lemah	Sesuai
6	30	20	lemah	lemah	Sesuai
7	46	7	kuat	sedang	Tidak sesuai
8	46	12	sedang	sedang	Sesuai
9	46	20	lemah	sedang	Tidak sesuai
10	68	7	kuat	kuat	Sesuai
11	68	12	kuat	kuat	Sesuai
12	68	20	sedang	kuat	Tidak sesuai
13	80	7	kuat	kuat	Sesuai
14	80	12	kuat	kuat	Sesuai
15	80	20	kuat	kuat	Sesuai

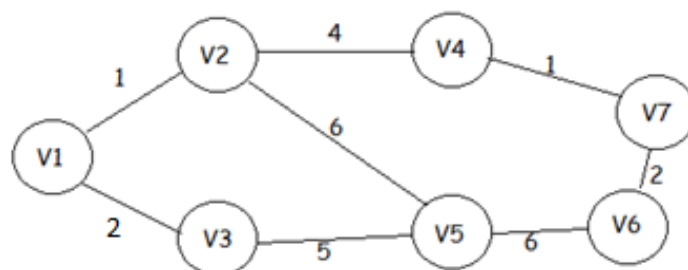
3.2 Pembahasan Algoritma *Dijkstra* dan *Fuzzy* Sugeno

3.2.1 Implementasi Algoritma *Dijkstra*

Tahap selanjutnya adalah dengan menerapkan algoritma *dijkstra* pada *zombie* untuk mencari jalur terpendek terhadap *player*. Adapun langkah-langkah untuk menerapkan algoritma *dijkstra* sebagai berikut :



Gambar 2 Diagram Blok Algoritma *Dijkstra*
 Inisialisasi algoritma *dijkstra* yang digunakan dapat dilihat sebagai berikut :



Gambar 3 Graf Algoritma *Dijkstra*

$V(G)$ = $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$. titik atau *node*
 L = Himpunan titik-titik $\in V(G)$ yang sudah terpilih (titik permanen) dalam jalur path terpendek.
 $D(j)$ = Jumlah bobot path terkecil v_1 ke v_j .
 $W(i,j)$ = Bobot garis dari titik v_i ke titik v_j .
 $W^*(1,j)$ = Jumlah bobot path terkecil dari v_1 ke v_j .
 Misalkan :
 $Node = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$

$$W = \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \begin{matrix} V1 \\ V2 \\ V3 \\ V4 \\ V5 \\ V6 \\ V7 \end{matrix} & \begin{pmatrix} \infty & 1 & 2 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 6 & \infty \\ \infty & \infty & \infty & \infty & 5 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 1 \\ \infty & \infty & \infty & \infty & \infty & 6 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix} \end{matrix}$$

Tabel 1 Hasil dari pencarian jalur terpendek

i	D(1)	D(2)	D(3)	D(4)	D(5)	D(6)	D(7)
1	0	$\text{Min}(\infty, 0+1)$ 1	$\text{Min}(\infty, 0+2)$ 2	∞	∞	∞	∞
2	0	1	2	$\text{Min}(\infty, 2+4)$ 6	∞	$\text{Min}(\infty, 2+6)$ 8	∞
3	0	1	2	6	∞	8	$\text{Min}(\infty, 6+1)$ 7
4	0	1	2	6	∞	8	7

Hasil yang didapatkan dari langkah-langkah yang dilakukan sebelumnya, jalur terpendek dari titik V1 ke titik V7 adalah V1 – V2 – V4 – V7 dengan nilai bobot 7.

3.2.2 Penerapan Fuzzy Sugeno

Pada penelitian ini logika *fuzzy* sugeno digunakan untuk mengatur kekuatan serangan *player* terhadap *zombie*. Pada penerapan *fuzzy* sugeno ada empat tahapan yang digunakan. Tahapan tersebut bertujuan untuk mendapatkan nilai *output* yang berupa konstanta atau persamaan linier. Berikut tahapan dalam logika *fuzzy* sugeno :

Gambar 4 Diagram Blok *Fuzzy* Sugeno

- Parameter *Input* : Jenis peluru dan jarak tembak.
- Parameter *Output* : Kekuatan tembak.

- c. Himpunan :
- Jenis Peluru :
 - Sangat Lemah = 0 – 15
 - Lemah = 10 - 35
 - Sedang = 36 - 55
 - Kuat = 56 - 70
 - Sangat Kuat = 71 – 100
 - Jarak Tembak() :
 - Jauh = 18 - 24
 - Sedang = 9 - 17
 - Dekat = 4 – 8
 - Kekuatan Tembak :
 - Lemah : 20
 - Sedang : 35
 - Kuat : 50

3.2.2.1 Pembentukan Aturan Dasar

Setelah mencari himpunan pada masing – masing variabel *input*, maka akan dilakukan pembentukan aturan dasar untuk menyatakan relasi antara *input* dan *output*. Pembentukan aturan dasar menggunakan bentuk *Fuzzy* metode Sugeno orde nol, dan operator yang digunakan untuk menghubungkan antara input adalah operator AND. Aturan dapat dilihat pada tabel 2

Tabel 2 Aturan *Fuzzy*

R[n]	α_n	Aturan Fuzzy
R[1]	α_1	IF Jenis Peluru sangat lemah AND Jarak Tembak dekat THEN Kekuatan Tembak lemah
R[2]	α_2	IF Jenis Peluru sangat lemah AND Jarak Tembak sedang THEN Kekuatan Tembak lemah
R[3]	α_3	IF Jenis Peluru sangat lemah AND Jarak Tembak jauh THEN Kekuatan Tembak lemah
R[4]	α_4	IF Jenis Peluru lemah AND Jarak Tembak dekat THEN Kekuatan Tembak sedang
R[5]	α_5	IF Jenis Peluru lemah AND Jarak Tembak sedang THEN Kekuatan Tembak lemah
R[6]	α_6	IF Jenis Peluru lemah AND Jarak Tembak jauh THEN Kekuatan Tembak lemah
R[7]	α_7	IF Jenis Peluru sedang AND Jarak Tembak dekat THEN Kekuatan kuat
R[8]	α_8	IF Jenis Peluru sedang AND Jarak Tembak sedang THEN Kekuatan Tembak sedang
R[9]	α_9	IF Jenis Peluru sedang AND Jarak Tembak jauh THEN Kekuatan Tembak lemah
R[10]	α_{10}	IF Jenis Peluru kuat AND Jarak Tembak dekat THEN

		Kekuatan Tembak kuat
R[11]	α_{11}	IF Jenis Peluru kuat AND Jarak Tembak sedang THEN Kekuatan Tembak kuat
R[12]	α_{12}	IF Jenis Peluru kuat AND Jarak Tembak jauh THEN Kekuatan Tembak sedang
R[13]	α_{13}	IF Jenis Peluru sangat kuat AND Jarak Tembak dekat THEN Kekuatan Tembak kuat
R[14]	α_{14}	IF Jenis Peluru sangat kuat AND Jarak Tembak sedang THEN Kekuatan Tembak kuat
R[n15]	α_{15}	IF Jenis Peluru sangat kuat AND Jarak Tembak jauh THEN Kekuatan Tembak kuat

3.2.2.1 Komposisi Aturan

Tahap ini setelah dibentuk aturan dasar, maka akan dilakukan komposisi aturan untuk mendapatkan bobot dari aturan dasar menggunakan fungsi implikasi MIN.

$$\alpha - \text{predikat 1} = \min (\mu_{\text{Jenis Peluru}} \cap \mu_{\text{Jarak Tembak}})$$

contoh :

$$\begin{aligned} \alpha - \text{predikat 1} &= \min (\mu_{\text{Sangatlemah}} \cap \mu_{\text{Dekat}}) \\ &= \min(\mu_{\text{Sangatlemah}}[11] \cap \mu_{\text{Dekat}}[15]) \\ &= \min(0.8;0.75) = 0.75 \end{aligned}$$

3.2.2.2 Defuzzyfikasi

Setelah melakukan komposisi aturan maka akan dilakukan defuzzyfikasi untuk mendapatkan output berupa nilai tegas. Nilai tegas ini kemudian akan ditransformasikan menjadi kekuatan tembak yang diperoleh dari tembakan player terhadap zombie. Defuzzyfikasi dilakukan dengan mencari nilai rata – ratanya dari predikat atau nilai bobot aturan yang digunakan.

$$Z = \frac{\alpha_{\text{predikat 1}} * z_1 + \alpha_{\text{predikat 2}} * z_2 + \alpha_{\text{predikat 3}} * z_3 + \alpha_{\text{predikatn}} * z_n}{\alpha_{\text{predikat 1}} + \alpha_{\text{predikat 2}} + \alpha_{\text{predikat 3}} + \alpha_{\text{predikatn}}}$$

Ket :

z_1, z_2, z_3, z_n = nilai output

$\alpha_{\text{predikat1}}, \alpha_{\text{predikat2}}, \alpha_{\text{predikat3}}, \alpha_{\text{predikatn}}$ = nilai predikat sesuai dengan himpunan

4. KESIMPULAN

Pengambilan keputusan untuk pencarian jalur terpendek dengan algoritma *dijkstra* dan logika *fuzzy* sugeno yang telah digunakan pada *game Zombie Shooter* dapat disimpulkan bahwa :

1. Penerapan logika *fuzzy* metode Sugeno, nilai output atau kekuatan tembak sesuai berdasarkan kondisi – kondisi yang sudah diberikan.
2. Penerapan algoritma *Dijkstra* dapat berjalan sesuai dengan harapan, dimana *zombie* dapat menentukan jalur terpendek yang digunakan dalam mencari dan mengejar *player* pada *game zombie shooter*.

5. SARAN

Saran yang dapat direkomendasikan penulis dalam penelitian kedepannya adalah menambah variasi *zombie* dengan kekuatan serang *zombie* yang berbeda dan arena bermain yang lebih luas.

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung, juga kepada pihak-pihak yang telah memberikan bimbingan, pengarahan, maupun ide-ide untuk penelitian.

DAFTAR PUSTAKA

- [1]Siang, J. J, 2014, *Riset Operasi dengan Pendekatan Algoritma*, Andi Offset, Yogyakarta.
- [2] Budiharto, W. 2014. *Artificial Intelligence Konsep dan Penerapannya*, Andi, Yogyakarta.