

Penerapan Algoritme Logika Fuzzy Untuk *Dynamic Difficulty Scaling* Pada Game Labirin

Ilham Akbar Ahmadi¹, Eriq Muhammad Adams Jonemaro², Muhammad Aminul Akbar³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹ilhamakbarahmadi10@gmail.com, ²eriq.adams@ub.ac.id, ³muhammad.aminul@ub.ac.id

Abstrak

Mayoritas pemain baru dalam sebuah game cenderung asal dalam memilih level. Hal ini menyebabkan terjadinya imbalance pada permainan, yang membuat game menjadi terlalu mudah atau bahkan terlalu susah. Untuk mengetahui tingkat kemampuan dan level, maka diperlukan suatu *input* dari pemain sebagai parameter yang digunakan untuk proses menentukan tingkat kesulitan pada game. Dalam menangani masalah tersebut, maka dalam penelitian ini akan dilakukan penerapan implementasi dengan metode *fuzzy* dalam membuat *Dynamic Difficulty Scaling* untuk memberikan *balance level* dalam game labirin. Metode pengujian yang dipakai meliputi pengujian validasi *behavior* dan performa dari permainan. Pengujian validasi *behavior* dilakukan dengan cara mencari 5 orang relawan dalam menguji permainan untuk melihat hasil keluaran *next level* labirin yang muncul serta menganalisis hasil dari *balance level* yang terjadi. Hasil dari analisis yang dilakukan yaitu metode *fuzzy* ini berhasil untuk menyetarakan *softskill* yang dimiliki pemain dengan *level* yang dimainkan. Sedangkan pengujian performa dilakukan dengan cara menguji besarnya *frames per second* (fps) yang dihasilkan selama permainan. Hasil menunjukkan bahwa performa terbaik terdapat pada map dengan ukuran 5x5 sebesar 107,22 FPS, serta performa terburuk terjadi pada labirin dengan ukuran 50x50 yang memiliki 87,65 FPS. Dari hasil pengujian performa tersebut dapat diketahui bahwa permainan ini layak untuk dimainkan mengingat batas minimum kelayakan FPS untuk permainan dikategorikan layak untuk dimainkan adalah 30 FPS.

Kata kunci: fuzzy, game, labirin, dynamic difficulty scaling

Abstract

A majority of new players in a game tend to pick levels at random. This causes an imbalance in the game, which makes the game too easy or too hard. To find out their abilities and levels, input is required from players as a parameter used in the process of determining the difficulty level of a game. In order to address this issue, in this research an implementation of the fuzzy method was utilized for Dynamic Difficulty Scaling to provide a level balance in a labyrinth game. The utilized testing methods were testing of behavior validation and gameplay performance. Behavior validation testing was conducted by requesting 5 volunteers to test the game to see the next level labyrinth output and to analyze the level balance that occurred. The analysis results show that the fuzzy method succeeded in equalizing the soft skills possessed by the players with the level being played. Meanwhile, performance testing was conducted by calculating the number of frames per second (FPS) that are outputted during gameplay. The results show that the best performance was obtained for a map of 5x5 size at 107.22 FPS, while the worst performance was for a labyrinth of 50x50 size at 87.65 FPS. From the results of this performance testing, it can be concluded that the game can be reasonably played, considering that the lower limit for the FPS of a game that can be reasonably played is 30 FPS.

Keywords: fuzzy, game, maze, dynamic difficulty scaling

1. PENDAHULUAN

Game adalah suatu program *virtual* yang dimainkan dengan menyelesaikan

konflik buatan sebagai aturan permainan (Dogan, 2014). *Game* yang kita kenal saat ini pada umumnya memiliki beberapa tingkat kesulitan, dimulai dari kesulitan tingkat mudah (*easy*), sedang (*medium*), hingga sulit

(*hard*). Namun tak sedikit pula game yang memberi kategori lebih untuk tingkat kesulitan, seperti pemain baru (*beginner*), sangat sulit (*very hard*), dan ahli (*master*). Perubahan tingkat kesulitan atau yang biasa disebut dengan istilah level dalam game memiliki fungsi untuk merubah tingkat kesulitan dalam permainan, semakin tinggi level yang diambil, maka tingkat kesulitan untuk menyelesaikan permainan juga akan semakin sulit.

Mayoritas pemain baru dalam sebuah game cenderung asal dalam memilih level. Hal ini menyebabkan terjadinya imbalance pada permainan, yang membuat game menjadi terlalu mudah atau bahkan terlalu susah. Kasus sering terjadi dikarenakan pemain tidak atau belum mengetahui kemampuan atau *softskill* mereka dalam bermain game yang baru dimainkan. *Softskill* merupakan istilah sosiologis yang berkaitan dengan kecerdasan emosional, sifat kepribadian, keterampilan, komunikasi, Bahasa, kebiasaan, keramahan, dan optimisme yang mencerminkan kemampuan seseorang (Sri Yuliani, 2015). Hal itu akan berdampak pada pemain, dimana pemain kurang mendapatkan *feedback* yang diharapkan saat pertama kali memainkan game yang pertama kali dimainkan.

Penentuan level permainan sebenarnya sudah banyak diterapkan dalam pembuatan game, namun dalam penerapannya masih cenderung hanya memiliki satu arah. Contohnya game "Piano Tiles" dimana tempo rythme dari permainan akan semakin cepat seiring lamanya permainan berlangsung, dan game "Neo fm" dimana setiap pemain melakukan "x" kali kesalahan, maka level permainan akan diturunkan ke 1 tingkat dibawah *current level*. Hal ini dilakukan demi menciptakan *balance* pada game untuk membuat pemain tertarik untuk menyelesaikan permainan. *Game Balancing* telah diakui oleh komunitas pengembang game sebagai salah satu faktor kunci dari kesuksesan permainan komputer (Falstein 2004). Maka dari itu diambillah sebuah metode untuk memberikan sebuah desain sistem yang dapat membaca kemampuan pemain serta menjadikannya sebagai parameter penentu pada level permainan yang dimainkan, yaitu dengan membuat *Dynamic Difficulty Scaling* yang difokuskan dalam

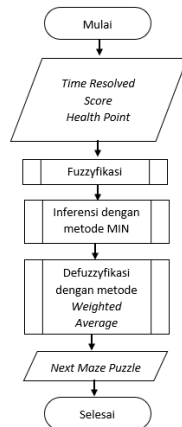
pengkategorian kemampuan pemain dalam *scale* khusus untuk pengkategorian kemampuan pemain dan tingkat kesulitan sistem.

Untuk mengetahui tingkat kemampuan dan level, maka diperlukan suatu input dari pemain yang kemudian dijadikan sebagai parameter yang kemudian diproses untuk menentukan tingkat kesulitan pada game. Pada tahap ini algoritme *Fuzzy* merupakan salah satu algoritme yang sesuai untuk membuat pengkategorian level berdasarkan tingkat kemampuan dari pemain. Logika *Fuzzy* sendiri merupakan suatu metode pendekatan sederhana yang dilakukan secara satuan untuk memindahkan aspek linguistic kedalam model matematis yang kemudian dapat digunakan untuk memberifikasi validasi dari penjelasan verbal. Logika *Fuzzy* merupakan salah satu komponen pembentuk *Soft Computing* dimana *membership function* menjadi ciri utama dari penalaran logika *fuzzy* (Kusumadewi S, Purnomo J, 2010). Algoritme *Fuzzy* dikatakan cocok untuk penerapan level karena keunggulannya dalam pemodelan aspek kualitatif dari pengetahuan dan kemampuan pemain, serta keputusan yang dibuat oleh pemain dengan menerapkan aturan dasar.

Tujuan dari penelitian ini adalah mengimplementasi algoritme *Fuzzy* yang digunakan sebagai pembuatan model untuk menentukan tingkat kesulitan otomatis berdasarkan kemampuan pemain saat dalam permainan, dimana tingkat kesulitan bisa bertambah, bisa berkurang, dan bisa bertahan pada tingkat kesulitan tertentu jika tingkat kesulitan itu dikategorikan sesuai dengan kemampuan pemain saat ini.

2. Perancangan Fuzzy

Pada tahap ini akan dijelaskan mengenai alur kinerja dari aplikasi dalam menentukan *output difficulty* dari permainan labirin dengan menggunakan metode *Fuzzy Inference System* Sugeno. Diagram alir kinerja fuzzy ditunjukkan pada gambar 1.



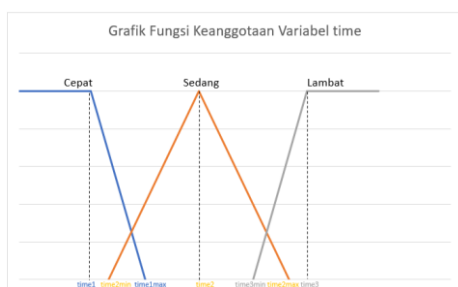
Gambar 1 Diagram alir fuzzy

Berdasarkan gambar 1 proses perhitungan menggunakan metode *Fuzzy Inference System* Sugeno dengan masukan variabel *Time Resolved*, *Score*, dan *Health Point*, kemudian dimulailah proses fuzzyfikasi, inferensi, dan defuzzyfikasi yang menghasilkan nilai dari *Next Maze Puzzle*.

2.1 Fungsi Keanggotaan

2.1.1 Time Resolved

Time Resolved diinisialisasi sebagai variabel “time” dan diklasifikasikan ke dalam tiga kategori yaitu lambat, sedang, dan cepat. Grafik fungsi keanggotaan variabel *time* ditunjukkan pada gambar 2.



Gambar 2 Grafik fungsi keanggotaan variabel *time*

Tabel himpunan fuzzy dari variabel *time* ditunjukkan pada tabel 1.

Tabel 1 Tabel himpunan fuzzy variabel *time*

No	Himpunan Fuzzy Time Resolved	Range
1	Cepat	$x \leq \text{time1max}$
2	Sedang	$\text{time2min} \leq x \leq \text{time2max}$
3	Lambat	$\text{time3min} \leq x$

Nilai persamaan :

$$\mu_{\text{cepat}}(X) = \begin{cases} 1, 0 \leq x \leq \text{time1} \\ \frac{\text{time1max} - x}{\text{time1max} - \text{time1}}, \text{time1} < x < \text{time1max} \dots (\text{persamaan 1}) \\ 0, x \geq \text{time1max} \end{cases}$$

$$\mu_{\text{sedang}}(X) = \begin{cases} 0, x \leq \text{time2min} \text{ || } x \geq \text{time2max} \\ \frac{x - \text{time2min}}{\text{time2} - \text{time2min}}, \text{time2min} < x \leq \text{time2} \dots (\text{persamaan 2}) \\ \frac{\text{time2max} - x}{\text{time2max} - \text{time2}}, \text{time2} < x < \text{time2max} \end{cases}$$

$$\mu_{\text{lambat}}(X) = \begin{cases} 1, x \geq \text{time3} \\ \frac{x - \text{time2min}}{\text{time3} - \text{time3min}}, \text{time3min} < x < \text{time3} \dots (\text{persamaan 3}) \\ 0, x \leq \text{time3min} \end{cases}$$

Keterangan nilai variabel:

$$\text{time1} = 2 * \text{pathCount} * 1 / 4;$$

$$\text{time2} = 2 * \text{pathCount} * 2 / 4;$$

$$\text{time3} = 2 * \text{pathCount} * 3 / 4;$$

$$\text{time1max} = \text{time1} + ((\text{time2} - \text{time1}) * 3 / 4);$$

$$\text{time2min} = \text{time1} + ((\text{time2} - \text{time1}) * 1 / 4);$$

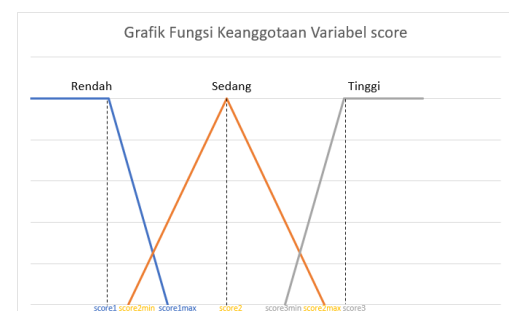
$$\text{time2max} = \text{time2} + ((\text{time3} - \text{time2}) * 3 / 4);$$

$$\text{time3min} = \text{time2} + ((\text{time3} - \text{time2}) * 1 / 4);$$

variabel *pathCount* merupakan total dari langkah yang dapat dilewati hingga titik *finish*. variabel *x* merupakan nilai dari waktu penyelesaian *maze* hingga titik *finish*.

2.1.2 Score

Score diinisialisasi sebagai variabel “score” dan diklasifikasikan ke dalam tiga kategori yaitu Sedikit, sedang, dan Banyak. Grafik fungsi keanggotaan variabel *score* ditunjukkan pada gambar 3.



Gambar 3 Grafik fungsi keanggotaan variabel *score*

Tabel himpunan fuzzy dari variabel *score* ditunjukkan pada tabel 2.

Tabel 2 Tabel Rule Base variabel *score*

No	Himpunan Fuzzy score	Range
1	Rendah	$x \leq \text{score1max}$
2	Sedang	$\text{score2min} \leq x \leq \text{score2max}$
3	Tinggi	$\text{score3min} \leq x$

Nilai persamaan :

$$\mu_{Rendah}(X) = \begin{cases} 1, 0 \leq x \leq score1 \\ \frac{score1max-x}{score1max-score1}, score1 < x < score1max \dots(\text{persamaan 4}) \\ 0, x \geq score1max \end{cases}$$

$$\mu_{Sedang}(X) = \begin{cases} 0, x \leq score2min \parallel x \geq score2max \\ \frac{x-score2min}{score2-score2min}, score2min < x \leq score2 \dots(\text{persamaan 5}) \\ \frac{score2max-x}{score2max-score2}, score2 < x < score2max \end{cases}$$

$$\mu_{Tinggi}(X) = \begin{cases} 1, x \geq score3 \\ \frac{x-score3min}{score3-score3min}, score3min < x < score3 \dots(\text{persamaan 6}) \\ 0, x \leq score3min \end{cases}$$

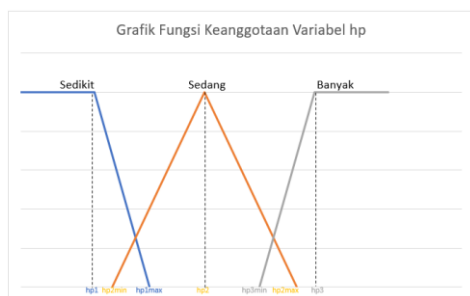
Keterangan nilai variabel:

- score1 = totCoin * 1;
- score2 = totCoin * 2;
- score3 = totCoin * 3;
- score1max = score1 + ((score2 - score1) * 3 / 4);
- score2min = score1 + ((score2 - score1) * 1 / 4);
- score2max = score2 + ((score3 - score2) * 3 / 4);
- score3min = score2 + ((score3 - score2) * 1 / 4);

variabel totCoin merupakan nilai dari total coin yang ada dalam permainan (hanya coin bernilai plus). Variabel x merupakan hasil perhitungan dari (coinPlus yang diperoleh*4) – (trapCoin yang diperoleh*2).

2.1.3 Health Point

Health Point diinisialisasi sebagai variabel “hp” dan diklasifikasikan ke dalam tiga kategori yaitu Sedikit, sedang, dan Banyak. Grafik fungsi keanggotaan variabel hp ditunjukkan pada gambar 3.



Gambar 4 Grafik fungsi keanggotaan variabel hp

Tabel himpunan fuzzy dari variabel time ditunjukkan pada tabel 3.

Tabel 3 Tabel Rule Base variabel hp

No	Himpunan Fuzzy score	Range
1	Sedikit	x ≤ hp1max
2	Sedang	hp2min ≤ x ≤ hp2max
3	Banyak	hp3min ≤ x

Nilai persamaan :

$$\mu_{Sedikit}(X) = \begin{cases} 1, 0 \leq x \leq hp1 \\ \frac{hp1max-x}{hp1max-hp1}, hp1 < x < hp1max \dots\dots\dots(\text{persamaan 7}) \\ 0, x \geq hp1max \end{cases}$$

$$\mu_{Sedang}(X) = \begin{cases} 0, x \leq hp2min \parallel x \geq hp2max \\ \frac{x-hp2min}{hp2-hp2min}, hp2min < x \leq hp2 \dots\dots(\text{persamaan 8}) \\ \frac{hp2max-x}{hp2max-hp2}, hp2 < x < hp2max \end{cases}$$

$$\mu_{Banyak}(X) = \begin{cases} 1, x \geq hp3 \\ \frac{x-hp3min}{hp3-hp3min}, hp3min < x < hp3 \dots\dots\dots(\text{persamaan 9}) \\ 0, x \leq hp3min \end{cases}$$

Keterangan nilai variabel:

- hp1 = radius * 1 / 4;
- hp2 = radius * 2 / 4;
- hp3 = radius * 3 / 4;
- hp1max = hp1 + ((hp2 - hp1) * 3 / 4);
- hp2min = hp1 + ((hp2 - hp1) * 1 / 4);
- hp2max = hp2 + ((hp3 - hp2) * 3 / 4);
- hp3min = hp2 + ((hp3 - hp2) * 1 / 4);

variabel radius merupakan nilai range dari game maze atau merupakan nilai dari level game saat ini. Variabel x adalah nilai dari perhitungan variabel maksimal health point – total jebakan yang diinjak oleh pemain.

2.2 Knowledge Base

Berisi tentang aturan-aturan yang digunakan, aturan yang dipakai diperoleh dari pengetahuan penulis yang kemudian diterapkan dalam sebuah rule base. Rule base ini yang nantinya digunakan sebagai aturan dasar dari logika fuzzy.

Tabel 4 Tabel Rule Base fuzzy

No	Time Resolved	Score	Health Point	Next Maze Puzzle
1	Cepat	Tinggi	Banyak	Naik Level
2	Cepat	Tinggi	Sedang	Naik Level
3	Cepat	Tinggi	Sedikit	Naik Level
4	Cepat	Sedang	Banyak	Naik Level
5	Cepat	Sedang	Sedang	Naik Level
6	Cepat	Sedang	Sedikit	Level Tetap
7	Cepat	Rendah	Banyak	Level Tetap
8	Cepat	Rendah	Sedang	Level Tetap
9	Cepat	Rendah	Sedikit	Level Tetap
10	Sedang	Tinggi	Banyak	Naik Level

11	Sedang	Tinggi	Sedang	Level Tetap
12	Sedang	Tinggi	Sedikit	Level Tetap
13	Sedang	Sedang	Banyak	Level Tetap
14	Sedang	Sedang	Sedang	Level Tetap
15	Sedang	Sedang	Sedikit	Level Tetap
16	Sedang	Rendah	Banyak	Level Tetap
17	Sedang	Rendah	Sedang	Level Tetap
18	Sedang	Rendah	Sedikit	Turun Level
19	Lambat	Tinggi	Banyak	Level Tetap
20	Lambat	Tinggi	Sedang	Level Tetap
21	Lambat	Tinggi	Sedikit	Level Tetap
22	Lambat	Sedang	Banyak	Level Tetap
23	Lambat	Sedang	Sedang	Turun Level
24	Lambat	Sedang	Sedikit	Turun Level
25	Lambat	Rendah	Banyak	Turun Level
26	Lambat	Rendah	Sedang	Turun Level
27	Lambat	Rendah	Sedikit	Turun Level

Dalam setiap aturan digunakan operator *AND* untuk mendapatkan perdidat dengan mencari nilai terkecil dari setiap aturan yang ada. Agar lebih mudah dalam memproses implikasi, komposisi, dan defuzzyfikasi, dapat diasumsikan dengan memberikan bobot pada setiap komposisi dari setiap variabel (Mahzar, 2013). Didapatkan hasil sebagai berikut :

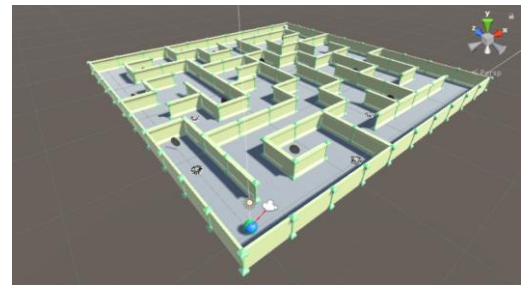
- Time Resolved* (time) diberikan bobot 3 kali lipat dari masing masing himpunan, dengan himpunan cepat berbobot 3, sedang berbobot 2, dan lambat berbobot 1.
- Score* (score) diberikan bobot 2 kali lipat dari masing masing himpunan, dengan himpunan rendah berbobot 1, sedang berbobot 2, dan tinggi berbobot 3.
- Health Point* (hp) diberikan bobot sesuai dengan masing masing himpunan, dengan himpunan sedikit berbobot 1, sedang berbobot 2, dan banyak berbobot 3.

Dari pernyataan diatas, untuk mendapatkan nilai minimum 0 dan maksimum 1 diperoleh rumus sebagai berikut :

$$z = ((3 * time) + (2 * score) + hp - 6) / 12$$

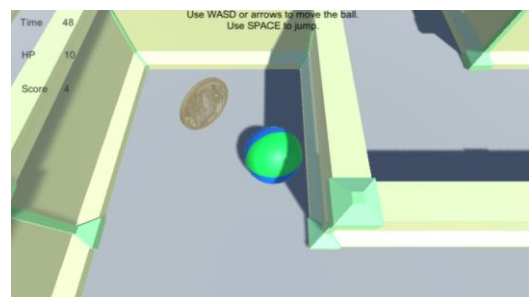
3. Implementasi

Adapun tampilan yang menunjukkan hasil dari implementasi metode fuzzy pada *game* labirin yang ditunjukkan pada gambar 7 dan 8.



Gambar 5 Gambar *game* labirin dari perspektif *unity scene*

Gambar 5 menunjukkan pandangan *game* labirin dari perspektif *scene* dari Unity untuk melihat seluruh ruangan labirin serta objek didalamnya telah berhasil tergenerate dan terbentuk.



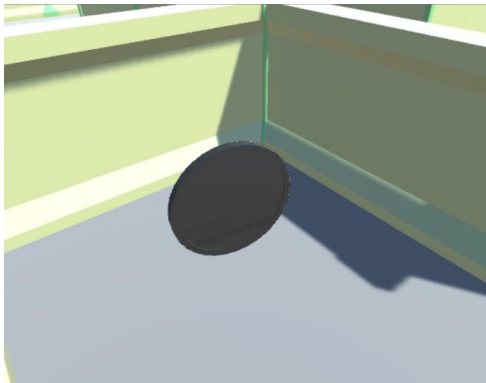
Gambar 6 Gambar *game* labirin dari perspektif pemain

Pada gambar 6 menunjukkan pandangan *game* labirin dari sudut pandang pemain yang bermain *game* labirin. Terdapat 4 macam text, yaitu *Time* untuk menunjukkan lamanya permainan berlangsung dalam detik, *HP* yang menunjukkan *health point* yang dimiliki saat ini, *Score* yang menunjukkan peroleh *score* yang diperoleh saat ini, serta papan pemberitahuan yang ada di tengah untuk memberikan panduan *basic* kepada pemain.



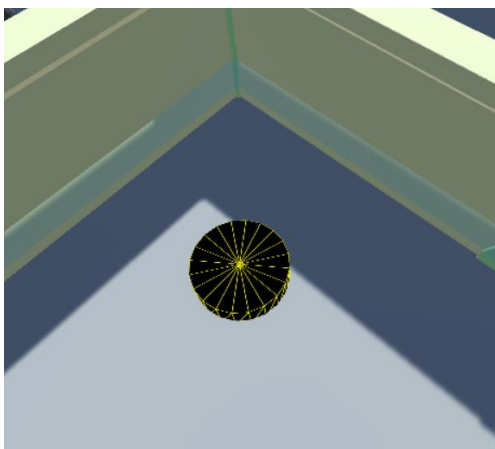
Gambar 7 Gambar coin

Gambar 7 menunjukkan gambar coin yang dipakai dalam menambah nilai plus pada score saat diambil, setiap coin yang didapat bernilai *plus 4 point*.



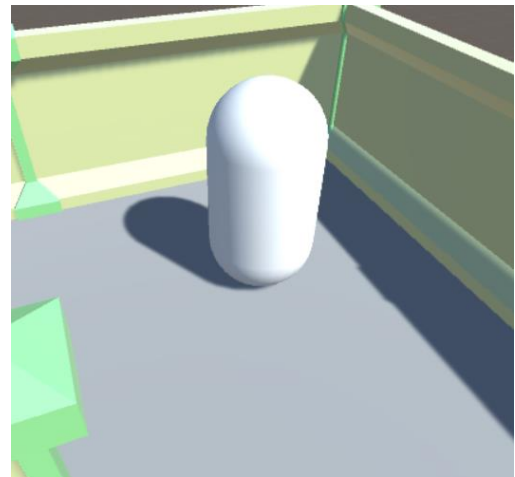
Gambar 8 Gambar trap coin

Gambar 8 menunjukkan gambar *trap coin* yang dipakai dalam mengurangi score saat tertabrak, setiap *trap coin* yang didapat bernilai *minus 2 point*. Berbeda dari coin yang bernilai *plus*, coin bernilai *minus* berwarna hitam kelam dan tidak memiliki shadow.



Gambar 9 Gambar jebakan

Gambar 9 menunjukkan gambar jebakan yang terdapat pada beberapa jalan yang harus dihindari oleh pemain, jika terkena jebakan maka *health point* pemain akan berkurang.



Gambar 10 Gambar goal

Gambar 10 menunjukkan gambar *goal* atau *finish* dari *game* labirin ini. Setelah pemain mencapai titik ini, maka proses fuzzy akan dilakukan untuk menentukan tingkat keberhasilan pemain serta mengenerate labirin yang akan muncul selanjutnya.

4. Pengujian

Pengujian Kinerja yang dipakai meliputi pengujian validasi behavior dan performa dari logika *fuzzy*. Pengujian validasi behavior dilakukan dengan cara menyelesaikan level dari map yang ditentukan serta melihat hasil keluaran *next level maze* yang muncul setelah *level maze* tersebut selesai. Sedangkan pengujian performa dilakukan dengan cara menguji besarnya *frames per second* (fps) yang dihasilkan selama permainan.

Pengujian *behavior* dilakukan dengan mengujikan *game* labirin ini sebanyak 10 kali terhadap 5 orang random untuk mendapatkan hasil grafik dari permainan pemain untuk melihat apakah balance level tersebut sudah cocok untuk diterapkan pada *game* labirin ini. Adapun hasil dari pengujian *behavior* yang ditunjukkan pada tabel 5.

Tabel 5 Tabel Hasil Pengujian Behavior

Player No.	Map	Range	Coin	Time Resolved	Score	Health Point	Output fuzzy	Output Game
1	1	10x10	10	318	18	4	0.225	Turun level
	2	9x9	9	221	26	6	0.4901	Level tetap
	3	9x9	8	157	20	5	0.6667	Level tetap
	4	9x9	9	163	26	7	0.75	Naik level
	5	10x10	9	285	22	8	0.3981	Level tetap
	6	10x10	10	303	30	6	0.4416	Level tetap
	7	10x10	10	73	2	7	0.6667	Level tetap
	8	10x10	8	259	28	8	0.5734	Level tetap
	9	10x10	9	169	24	7	0.7554	Level naik
	10	11x11	10	240	24	6	0.55	Level tetap
2	1	10x10	10	237	26	7	0.684	Level tetap
	2	10x10	9	201	30	9	0.75	Naik level
	3	11x11	8	193	22	9	0.8241	Naik level
	4	12x12	11	230	28	9	0.7688	Naik level
	5	13x13	13	248	30	8	0.7033	Level tetap
	6	13x13	15	265	26	10	0.6638	Level tetap
	7	13x13	12	301	32	11	0.7222	Level tetap
	8	13x13	13	273	36	11	0.8149	Naik level
	9	14x14	12	512	26	6	0.3125	Level tetap
	10	14x14	12	493	30	7	0.4526	Level tetap
3	1	10x10	9	201	26	7	0.75	Naik level
	2	11x11	11	265	32	8	0.75	Naik level
	3	12x12	10	347	28	8	0.6348	Level tetap
	4	12x12	11	352	30	9	0.6326	Level tetap
	5	12x12	11	312	26	7	0.57	Level tetap
	6	12x12	11	320	34	9	0.75	Naik level
	7	13x13	12	470	24	6	0.25	Turun level
	8	12x12	10	317	30	7	0.6805	Level tetap
	9	12x12	11	291	32	10	0.7499	Level tetap
	10	12x12	12	339	38	10	0.6938	Level tetap
4	1	10x10	10	259	20	6	0.3737	Level tetap
	2	10x10	9	263	24	5	0.4462	Level tetap
	3	10x10	9	231	24	7	0.6541	Level tetap
	4	10x10	10	222	26	6	0.6354	Level tetap
	5	10x10	8	199	22	7	0.75	Naik level
	6	11x11	11	287	28	7	0.5411	Level tetap
	7	11x11	10	319	30	7	0.5364	Level tetap
	8	11x11	11	295	28	8	0.6509	Level tetap
	9	11x11	10	270	30	8	0.75	Naik level

5	10	12x12	11	354	24	9	0.4082	Level tetap
	1	10x10	9	165	22	6	0.6885	Level tetap
	2	10x10	10	147	30	8	0.8863	Naik level
	3	11x11	10	175	32	9	0.8988	Naik level
	4	12x12	12	294	36	12	0.75	Naik level
	5	13x13	11	268	34	11	0.8297	Naik level
	6	14x14	13	340	40	11	0.755	Naik level
	7	15x15	15	421	40	12	0.7222	Level tetap
	8	15x15	15	380	44	13	0.7786	Naik level
	9	16x16	14	501	36	13	0.6904	Level tetap
10	16x16	16	529	40	14	0.6666	Level tetap	

Berdasarkan tabel 5 ditunjukkan table hasil pengujian yang dilakukan terhadap 5 pemain acak. Dalam table diberikan pengetahuan tentang range/radius dari game labirin serta jumlah coin hasil dari generate labirin, pemain diharuskan untuk menyelesaikan permainan dan mendapatkan nilai dari Time Resolved, Score, dan Health Point. Kemudian pada kolom terakhir ditunjukkan hasil dari perhitungan dengan metode fuzzy dalam bentuk output. Sebagai contoh akan diambil 1 sample dari pengujian untuk di analisis



Gambar 11 Grafik hasil pengujian player 1

Berdasarkan Gambar 11 ditunjukkan hasil dari pengujian yang dilakukan oleh pemain pertama. Titik pada gambar tersebut menjelaskan tentang besarnya nilai fuzzy yang didapatkan oleh pemain saat berada pada level tersebut. Diketahui bahwa titik awal pemain diberikan level 10 karena dengan anggapan pemain belum mengetahui kemampuan bermain dalam game labirin ini. Terlihat pada table 1, pada pengujian ke 1, pemain mendapatkan nilai output yang berada dibawah rata rata yaitu 0.225, mengakibatkan dirinya harus turun satu level dibawah level saat ini. Seiring berjalannya permainan, pemain mulai mengerti dan belajar tentang strategi permainan yang membuatnya dapat naik

level pada pengujian ke 4. Pada pengujian ke 5 hingga ke 8, pemain masih berusaha untuk meningkatkan strategi bermainnya hingga pada pengujian ke 9 pemain berhasil naik level ke level selanjutnya.

Tabel 6 Tabel Hasil Pengujian Performa

No	Frames per second (fps)				
	5x5	10x10	20x20	30x30	50x50
1	97.02	95.12	94.25	95.93	96.24
2	95.66	99.79	99.43	93.48	94.85
3	102.26	101.91	97.12	89.43	91.01
4	96.91	93.25	92.99	97.23	98.23
5	107.22	95.76	94.66	95.33	87.94
Rata-rata	99.814	97.166	95.69	94.28	93.65

Dari tabel 2 ditunjukkan bahwa hasil pengujian performa dari *game* labirin setelah ditambahkan implementasi *fuzzy* untuk *dynamic difficulty scaling* memiliki nilai performa terbaik pada saat pengujian ke 5 dengan labirin berukuran 5x5 dengan nilai 107,22, serta performa terburuk berada pada saat pengujian ke 4 dengan labirin berukuran 50x50 dengan nilai 87,65.

Pengimplementasian metode *fuzzy* untuk *dynamic difficulty scaling* pada *game* labirin yang diuji dalam 5 kali percobaan dengan sample random mendapatkan nilai yang cukup balance bagi pemain dengan kemampuan bermain yang berbeda beda. kemampuan pemain dapat diasah dengan pengalaman bermain, sehingga level permainan akan merespon kemampuan dari pemain. Jika pemain terlalu berada dibawah rata rata maka level permainan akan diturunkan, jika pemain berada didalam rata rata maka level permainan akan tetap pada level tersebut, dan jika pemain berada diatas rata rata maka level permainan akan dinaikan.

5. Kesimpulan

Berdasarkan hasil dari pengujian yang telah dilakukan, dapat disimpulkan beberapa poin sebagai berikut.

1. Penerapan algoritme *fuzzy* dilakukan dengan cara mengambil data input dari pemain yang telah menyelesaikan suatu level dari *game* labirin, yang di ambil sebagai input yaitu nilai dari waktu penyelesaian permainan (*time resolved*), total *score* yang di dapat (*score*), dan sisa *health point* dari pemain (*hp*).

Pedoman yang dipakai dalam menentukan kriteria fungsi keanggotaan adalah nilai dari total jalan yang dapat diambil (*pathCount*), total dari coin yang bernilai positif (*coinPlus*), dan *range area* dari level labirin (*range*). Dari ketiga inputan tersebut dilakukan proses fuzzyfikasi untuk menggolongkan fungsi keanggotaan dari setiap variabel. Dari setiap fungsi keanggotaan yang diperoleh kemudian dimasukkan pada setiap *rule base* dari *knowledge base* untuk proses implikasi dengan menggunakan metode MIN, hasil tersebut kemudian dilakukan proses defuzzyfikasi untuk mendapatkan nilai keluaran dari perhitungan fuzzy, hasil dari *fuzzy* tersebut yang dijadikan sebagai tolok ukur sebagai penentu level pada level selanjutnya dari *game* labirin.

2. Dari pengujian performa yang dilakukan, nilai dari fps permainan setelah penambahan *dynamic difficulty scaling* tergolong dalam kategori stabil dalam artian baik, yang artinya pengimplementasian logika *fuzzy* ini tidak mengganggu atau memberatkan permainan. Berdasarkan pengujian performa fps yang dilakukan, ditunjukkan hasil performa dari *game* hanya terpengaruh sedikit oleh ukuran dari labirin yang di *generate*. Nilai dari pengujian fps pun tergolong dalam kategori layak untuk dimainkan mengingat batas minimal standart fps adalah 30fps, sedangkan saat pengujian nilai terkecil adalah 87,65.

6. Daftar Pustaka

Arif, Y. M., Kurniawan F., Nugroho, F., 2011. *Desain Perubahan Perilaku pada NPC Game Menggunakan Logika Fuzzy*. Seminar on Electrical, Informatics and Education, p. A2.110 -A2.111, 2011.

Ahsani, N. H., Jonemao, E. M. A., Candra, D., 2014. *Penerapan Algoritma Carlo Tree Search Pada Permainan Komputer Maze Treasure*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Anggiawan, S. D., 2011. *Pembuatan Game*

- Logic Puzzle untuk Edukasi Logika Anak*. Surabaya: Teknologi Multimedia Broadcasting, Institut Teknologi Sepuluh November.
- Dogan, A., 2014. *Prevention and Treatments of Games Addiction: NonPharmacological Approaches for Game Addiction*. In S. Gunuc, ed. *Epidemiology of Game Addiction*. California: OMICS Group.
- Handriyantini, Eva, 2009. *Permainan Edukatif (Educational Games) Berbasis Komputer untuk Siswa Sekolah Dasar*. Malang: Sekolah Tinggi Informasi & Komputer Indonesia.
- Kusumadewi, S., 2010, *Aplikasi Logika Fuzzy untuk pendukung keputusan*, Edisi 2, Graha Ilmu, Yogyakarta.
- Kusumadewi, S., 2003, *Artificial Intelligence (Teknik dan Aplikasinya)*, Yogyakarta : Graha Ilmu.
- Millington, I., Funge, J., 2009. *Artificial Intelligence For Games Second Edition*. Tersedia di <http://lecturer.ukdw.ac.id/~mahas/dossier/gameng_AIFG.pdf> [Diakses 27 Oktober 2017].
- Mutiah, Diana, 2010. *Psikologi Bermain Anak Usia Dini*. Jakarta: Kencana Prenada Media Group.
- Moreira, R. V., Regan, L., Mandryk, Gutwin, C., 2015, *Now You Can Compete With Anyone : Balancing Players of Different Skill Levels in a First-Person Shooter Game*, Chi, 2015, 2255–64.
- Nur, H., 2015. *Pergerakan dan Perilaku Non Player Characters pada Permainan First Person Shooter dengan Algoritma Boids dan Logika Fuzzy*. Banjarbaru.
- Purba, K. Radion, Hasanah, Nur R., Muslim, Azis, M., 2013. *Implementasi Logika Fuzzy Untuk Mengatur Perilaku Musuh dalam Game Bertipe Action-RPG*, Jurnal EECCIS Vol. 7, No. 1, Juni 2013.
- Roedavan, R. (2014). *Unity Tutorial Game Engine*. Bandung: Informatika.
- Septian, R. F., 2014. *Implementasi Fuzzy Logic Metode Mamdani Untuk Pengembangan Intelligent Non-Player Character Pada Game Strategy*. Universitas Pendidikan Indonesia.
- Suyanto, 2007. *Artificial Intelligence Searching, Reasoning, Planning and Learning*, Penerbit Informatika, Bandung.
- Widiastuti, N. I., 2013. *Model Perilaku Berjalan Agen-Agen Menggunakan Fuzzy Logic*, Jurnal Ilmiah Komputer dan Informatika, Vol. 1, No.1, 2013.
- Yuliani, S., 2015. *Apa Itu Softskill?. Tersedia di* <<http://scdc.binus.ac.id/bslc/2017/08/apa-itu-soft-skills/>> [Diakses 18 Oktober 2017].