

## IMPLEMENTASI ALGORITMA FUZZY MAMDANI DAN FISHER-YATES SHUFFLE PADA GAME CITY STREET RUN

### *Implementation of Fuzzy Mamdani and Fisher-Yates Shuffle Algorithms in City Street Run Game*

Kelvin Chandra, [s32190041@student.ubm.ac.id](mailto:s32190041@student.ubm.ac.id)<sup>1)</sup>, Evasaria Magdalena Sipayung, [11874@lecturer.ubm.ac.id](mailto:11874@lecturer.ubm.ac.id)<sup>2)\*</sup>

<sup>1)</sup>Program Studi Informatika, Fakultas Teknik dan Desain, Universitas Bunda Mulia

Diterima 11 Agustus 2024 / Disetujui 20 September 2024

#### ABSTRACT

*This City Street Run game is made by taking a city street background and characters running to collect coins and pass obstacles to get the highest score. The character will run through object obstacles in the form of orange cones and red barriers while collecting coins which will affect the player's score. The reason the author makes this game is that it is expected to improve thinking power, improve concentration, solve problems and improve brain memory. This study uses the Mamdani Fuzzy Algorithm because this algorithm can make scores affect the character's running speed. Then this research also uses the Fisher-Yates Shuffle Algorithm so that coins can be spawned randomly in the City Street Run game. As well as implementing the Multimedia Development Life Cycle Method for the City Street Run game because the City Street Run game combines text, images and audio. Based on the test results with Black Box Testing, it can be concluded that all the features contained in the City Street Run game have run as expected and based on the test results with User Acceptance Testing through a questionnaire, with a percentage value of 85.5%, so it can be concluded that the City Street Run game is classified as very good and can be accepted and played by users. Based on these results, it can be concluded that the design of the City Street Run game was successfully implemented using the Multimedia Development Life Cycle method, the fuzzy mamdani algorithm and the Fisher-Yates Shuffle algorithm.*

**Keywords:** *Game, Fuzzy Mamdani, Fisher-Yates Shuffle, Multimedia Development Life Cycle*

#### ABSTRAK

Game City Street Run ini dibuat dengan mengambil latar jalanan di perkotaan dan karakter yang berlari mengumpulkan koin dan melewati rintangan untuk mendapatkan score tertinggi. Karakter akan berlari melewati rintangan objek berupa cone orange dan red barrier sembari mengumpulkan koin yang akan mempengaruhi score pemain. Alasan penulis membuat game ini diharapkan dapat meningkatkan daya pikir, meningkatkan konsentrasi, memecahkan masalah dan meningkatkan daya ingat otak. Penelitian ini menggunakan Algoritma Fuzzy Mamdani karena algoritma ini bisa membuat score dapat mempengaruhi kecepatan berlari karakter. Lalu penelitian ini juga menggunakan Algoritma Fisher-Yates Shuffle supaya koin dapat terspawn secara random pada game City Street Run. Serta mengimplementasikan Metode Multimedia Development Life Cycle terhadap game City Street Run karena game City Street Run menggabungkan teks, gambar, dan audio. Berdasarkan hasil pengujian dengan Black Box Testing, dapat disimpulkan bahwa semua fitur yang terdapat pada game City Street Run telah berjalan sesuai harapan dan berdasarkan hasil pengujian dengan User Acceptance Testing melalui kuisioner, dengan nilai presentase sebesar 85,5%, sehingga dapat disimpulkan bahwa game City Street Run tergolong sangat baik serta dapat diterima dan dimainkan oleh pengguna. Berdasarkan hasil tersebut maka dapat disimpulkan bahwa perancangan game City Street Run berhasil diterapkan menggunakan metode Multimedia Development Life Cycle, Algoritma fuzzy mamdani dan Algoritma Fisher-Yates Shuffle.

**Kata Kunci:** *Game, Fuzzy Mamdani, Fisher-Yates Shuffle, Multimedia Development Life Cycle*

\*Korespondensi Penulis:  
E-mail: [11874@lecturer.ubm.ac.id](mailto:11874@lecturer.ubm.ac.id)

## PENDAHULUAN

Pada saat ini, game tidaklah asing lagi didengar oleh semua orang, sifatnya yang dinamis yaitu selalu berkembang mengikuti perkembangan teknologi. Oleh karena itu game dianggap sebagai media mengekspresikan emosi seseorang, bisa juga sebagai sarana hiburan di waktu senggang. Tidak hanya itu ada pula yang menjadikan game sebagai salah satu mata pencaharian untuk mendapatkan adsense atau uang. Melalui game, penggunaanya juga diasah untuk memiliki sikap pantang menyerah, patriotisme dan kemanusiaan pada saat memainkannya [1]. Game merupakan sarana hiburan terpopuler oleh khalayak ramai, termasuk oleh anak-anak.

Game juga menjadi salah satu alternatif media hiburan dalam aktivitas harian anak-anak jadi bisa dipergunakan menjadi media pembelajaran sejarah yang kreatif, edukasi matematika, mengasikan, dan efektif sehingga game menjadi lebih menarik dan interaktif [2][3][4]. Game juga membantu seseorang memiliki motivasi dalam segala hal. Seperti halnya game yang dibuat dengan harapan dapat meningkatkan kemampuan berpikir, menyelesaikan masalah dan memaksa otak untuk terus berlatih berfikir agar tidak mudah lupa.

Saat ini game semakin variatif, berkembangnya varian game membuatnya tidak pernah ada habisnya terutama pada bagian Artificial Intelligence (AI). Setiap game mempunyai AI atau kecerdasan bukannya sendiri-sendiri sehingga banyak variasi yang dihasilkan dan membuat sebuah game menjadi mempunyai banyak ciri khas tersendiri [5]. Game mempunyai berbagai macam variasi pada genrenya seperti Action, Adventure, Endless Runner, Fantasy, Horror, Mystery, Racing, Simulation, Puzzle, Role-Playing (RPG), Strategy, Sports, dan masih banyak lagi yang dikemas dengan karakter tokoh dan grafik yang menarik.

AI atau kecerdasan buatan sendiri adalah sebuah penerapan ilmu komputer yang dibuat untuk memberikan perintah kepada mesin atau komputer sehingga dapat melakukan dan menyelesaikan pekerjaan dengan hasil yang sesuai perintah seperti yang dilakukan oleh kecerdasan otak manusia. Penelitian menggunakan algoritma Fuzzy Mamdani [6], Logika Fuzzy ialah hasil pengembangan logika Boolean yang memperkenalkan konsep setengah kebenaran. Yang mana logika klasik menyampaikan bahwa semua dapat dinyatakan sebagai binary (hitam atau putih, ya atau tidak, 0 atau 1), logika fuzzy mengganti kebenaran boolean menjadi tingkat kebenaran.

Dalam penelitian ini game 3D yaitu game yang memiliki 3 sisi (X, Y, Z) dan bisa dipandang dari segala arah [7]. Hal ini disebabkan oleh grafis 3D yang lebih baik dalam penggambaran karakter dan suasana sehingga game yang dihasilkan akan terlihat lebih real dan menarik pada saat dimainkan.

Penerapan Algoritma Fisher-Yates Shuffle yang digunakan untuk mengacak koin yang akan dimunculkan dalam permainan, objek puzzle sehingga user tidak bisa menghafal posisi objek [8]. Algoritma Fisher-Yates Shuffle juga biasa digunakan dalam pembuatan game teka-teki. Algoritma ini melakukan pengacakan random pada soal, dengan begitu soal yang ditampilkan menjadi lebih bervariasi dan tidak berulang-ulang [9].

Penggunaan salah satu metode Artificial Intelligence yaitu Fuzzy Mamdani dalam perancangan game City Street Run diharapkan dapat melatih ketangkasan dan konsentrasi pemainnya dan dapat dimainkan oleh berbagai kalangan usia mengisi waktu luang.

### **MDLC (Multimedia Development Life Cycle)**

Metode MDLC (*Multimedia Development Life Cycle*), yang merupakan sebuah metode yang cocok untuk membuat dan menginovasikan aplikasi media yang menyatukan suara, gambar, animasi, video dan media lainnya. Metodologi MDLC lebih mengutamakan fase pengembangan perangkat lunak [10]. Tahapan dari metode MDLC [11]:

1. *Concept*

Pada tahapan ini merupakan tahap dalam menetapkan target pengguna dan tujuan. Yang diharapkan pada tahap ini adalah program dapat berjalan sesuai identitas multimedia yang sudah di program sehingga bisa sampai ke *audience* dengan tepat

2. *Design*

Pada tahapan ini penulis mulai merancang pembentukan spesifikasi terkait kerangka program, tampilan, gaya, dan material yang diperlukan program. Desain dirancang memakai desain *interface* dari tampilan menu aplikasi.

3. *Material Collecting*

Tahap ini merupakan saat dimana proses penggabungan bahan yang diperlukan program oleh penulis. Semua bahan tersebut meliputi karakter, suara, *background*, *3D object* dan lainnya sesuai plan yang sudah dirancang.

4. *Assembly*

Pada tahapan ini merupakan tahap dimana penulis mulai menggabungkan semua detail atau bahan multimedia untuk dapat dijalankan sesuai *planning*. Perancangan aplikasi berdasarkan tahapan desain, seperti *usecase*, *flowchart* dan *storyboard*. Pembuatannya sendiri memerlukan *software Unity 3D*.

5. *Testing*

Tahapan ini dilaksanakan sesudah penulis menyelesaikan tahap *assembly* (pembuatan) dengan cara mengoperasikan aplikasi/program, sehingga bisa diteliti jika terjadi *error* atau tidak. Pengujiannya sendiri bisa memakai teknik *black box testing* untuk menguji program menurut fungsinya. Tujuan teknik *black box testing* ialah mendapatkan kegagalan peranan dari program.

6. *Distribution*

*Distribution* adalah tahapan terakhir pada metode MDLC, yang mana tahapan ini merupakan hasil dari perancangan atau pembuatan aplikasi sudah berjalan dengan maksimal lalu sudah layak disebarakan kepada pengguna.

### Algoritma Fisher-Yates Shuffle

Algoritma *fisher-yates shuffle* biasanya dipakai dalam *game* teka-teki. Algoritma ini melakukan pengacakan *random* pada soal, jadi soal yang dimunculkan lebih bervariasi serta tidak berulang Kembali [9]. Pada penelitian ini digunakan untuk melakukan pengacakan terhadap item yang akan dimunculkan dalam permainan. Dengan menggunakan algoritma *fisher-yates shuffle* diharapkan pemain tidak dapat menebak letak koin yang akan dikumpulkan untuk mendapatkan *score*.

## METODOLOGI PENELITIAN

Algoritma *Fuzzy Mamdani* biasa digunakan sebagai parameter *score* [12], penerapan Algoritma *Fuzzy Mamdani* dalam perancangan *game* City Street Run digunakan untuk mengatur kecepatan pergerakan *player* dengan membandingkan parameter *score* dan parameter waktu. Terdapat 4 tahapan dalam proses menginput [13], terdiri dari:

1. *Fuzzification* merupakan langkah awal mengubah input yang nilai kebenarannya pasti atau akurat ke bentuk *fuzzy* input, menjadi nilai himpunan yang keberadaannya ditetapkan dari fungsi keanggotaan tertentu.
2. *Rule Evaluation* merupakan langkah mengambil nilai input yang telah difuzzyfikasikan lalu mengimplementasikannya kedalam *antecedents* terhadap aturan *fuzzy* yang berlaku. Fungsi implikasi yang biasa dipakai ialah *mi*

$$\mu_{A \cap B}(x) = \min(\mu_A[x], \mu_B[x])$$

Keterangan :

$\mu$  = *fuzzy set*

$\mu_A[x]$  = kumpulan *fuzzy* set A dari x

$\mu_B[x]$  = kumpulan *fuzzy* set B dari x

3. *Rule Aggregation* merupakan cara memproses penyatuan nilai keluaran dari seluruh aturan. Ditahap ini yang dipakai adalah metode *max*, yang mana solusi himpunan *fuzzy* didapatkan dengan teknik mengambil nilai *maksimum*, aturan yang dipakai untuk mengubah daerah *fuzzy*.

$$\mu_{sf} [x_i] = \max(\mu_{sf} [x_i], \mu_{kf} [x_i])$$

Keterangan :

$\mu_{sf} [x_i]$  = nilai keanggotaan solusi *fuzzy*

$\mu_{kf} [x_i]$  = nilai keanggotaan konsekuen *fuzzy*

4. *Defuzzification* merupakan proses terakhir dari metode *fuzzy* yaitu untuk mengubah nilai *fuzzy* hasil dari agregasi aturan ke dalam angka atau bilangan, metode yang paling umum dipakai untuk metode inferensi *fuzzy mamdani* ialah metode centroid (*Centre Of Gravity*) atau COG.

$$Z^* = \frac{\int \mu(z)zdz}{\int \mu(z)dz}$$

## HASIL DAN PEMBAHASAN

Implementasi algoritma Fuzzy Mamdani dan Fisher-Yates Shuffle pada Game City Street Run dengan tampilan *main menu* ditunjukkan pada Gambar 1.

Main menu mempunyai 3 tombol yaitu *START*, *ABOUT*, dan *QUIT GAME*. Tombol *START* berfungsi untuk menjalankan *game*, dan tombol *ABOUT* berfungsi untuk menampilkan *about* dan tombol *QUIT GAME* berfungsi untuk menutup *game*. Gambar 2 merupakan tampilan *Interface About* dari *game* City Street Run, dimana terdapat deskripsi *game* dan tombol *BACK* yang berfungsi untuk mengembalikan tampilan *player* ke *main menu*.



Gambar 1 Tampilan *Interface Main Menu*



**Gambar 2** Tampilan *Interface About*

Saat *player* menekan *start* pada *main menu* maka akan muncul *loading screen* ditunjukkan pada Gambar 3.



**Gambar 3** Tampilan *Interface Loading Screen*

*Score* yang didapatkan oleh *player* akan ditampilkan waktu yang terus berjalan seiring *player* bermain dan tombol *pause* yang berfungsi untuk menghentikan *game*. *Interface In-Game* dari *game City Street Run* ditunjukkan pada Gambar 4.



**Gambar 4** Tampilan *Interface In-Game*

Pada *Interface Game Paused* dari *game City Street Run*, terdapat 3 tombol yaitu *RESUME*, *BACK TO MENU*, dan *QUIT GAME*. Tombol *RESUME* berfungsi untuk menjalankan *game* kembali setelah di *pause*, tombol *BACK TO MENU* berfungsi untuk mengembalikan tampilan

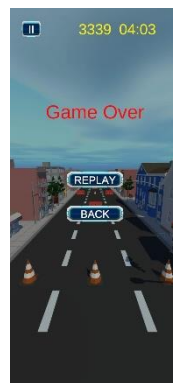
player ke main menu dan tombol *QUIT GAME* yang berfungsi untuk menutup *game*. *Interface Game Paused* dari *game City Street Run* ditunjukkan pada Gambar 5.



A.

**Gambar 5** Tampilan *Interface Game Paused*

Pada *Interface Game Over* dari *game City Street Run* terdapat 2 tombol yaitu *REPLAY* dan *BACK*. Tombol *REPLAY* berfungsi untuk mengulang *game* dan tombol *BACK* yang berfungsi untuk mengembalikan tampilan *player* ke *main menu*. *Interface Game Over* dari *game* ditunjukkan pada Gambar 6.



**Gambar 6** Tampilan *Interface Game Over*

### Contoh Pengujian Metode *Fuzzy Mamdani*

Jika *player* memiliki *score* 1500 dan waktu 120, kategori kecepatan lari apa yang digunakan?

Berikut langkah-langkah yang diperlukan untuk mendapatkan hasil adalah sebagai berikut :

#### 1. *Fuzzification*

Variabel *score* 1500 termasuk kedalam himpunan **sedikit** dan **sedang**.

$$\mu[\text{sedikit}] = \begin{cases} 0; x \geq 2500 \\ \frac{2500 - x}{2500 - 1000}; 1000 \leq x \leq 2500 \\ 1; x \leq 1000 \end{cases}$$

$$\mu[\text{sedang}] = \begin{cases} 0; x \leq 1000 \text{ atau } x \geq 3500 \\ \frac{x - 1000}{2500 - 1000}; 1000 \leq x \leq 2500 \\ \frac{3500 - x}{3500 - 2500}; 2500 \leq x \leq 3500 \end{cases}$$

Sehingga diperoleh :

$$\begin{aligned}\mu_{Sedikit}(1500) &= \frac{2500 - 1500}{2500 - 1000} = \frac{1000}{1500} = 0,67 \\ \mu_{Sedang}(1500) &= \frac{1500 - 1000}{2500 - 1000} = \frac{500}{1500} = 0,33 \\ \mu_{Banyak}(1500) &= 0\end{aligned}$$

Selanjutnya, variabel waktu 120 termasuk kedalam himpunan **lambat** dan **sedang**.

$$\begin{aligned}\mu[lambat] &= \begin{cases} 0; x \geq 150 \\ \frac{150 - x}{150 - 50}; 50 \leq x \leq 150 \\ 1; x \leq 50 \end{cases} \\ \mu[sedang] &= \begin{cases} 0; x \leq 50 \text{ atau } x \geq 200 \\ \frac{x - 50}{150 - 50}; 50 \leq x \leq 150 \\ \frac{200 - x}{200 - 150}; 150 \leq x \leq 200 \end{cases}\end{aligned}$$

Sehingga diperoleh :

$$\begin{aligned}\mu_{Lambat}(120) &= \frac{150 - 120}{150 - 50} = \frac{30}{100} = 0,3 \\ \mu_{Sedang}(120) &= \frac{120 - 50}{150 - 50} = \frac{70}{100} = 0,7 \\ \mu_{Cepat}(120) &= 0\end{aligned}$$

#### Rule Evaluation

Perbandingan antara setiap variabel berdasarkan aturan yang telah dibuat dengan nilai *input score* 1500 (termasuk dalam himpunan sedang) dan waktu 120 (termasuk dalam himpunan sedang). Aturan yang memenuhi kondisi tersebut adalah :

[R1] if *score sedikit* dan waktu **lambat** then kecepatan **lambat**

$$\begin{aligned}\alpha - predikat_1 &= \mu_{Sedikit}(x) \cap \mu_{Lambat}(x) \\ &= 0,3\end{aligned}$$

[R2] if *score sedikit* dan waktu **sedang** then kecepatan **lambat**

$$\begin{aligned}\alpha - predikat_2 &= \mu_{Sedikit}(x) \cap \mu_{Sedang}(x) \\ &= 0,67\end{aligned}$$

[R3] if *score sedikit* dan waktu **banyak** then kecepatan **lambat**

$$\begin{aligned}\alpha - predikat_3 &= \mu_{Sedikit}(x) \cap \mu_{Banyak}(x) \\ &= 0\end{aligned}$$

[R4] if *score sedang* dan waktu **lambat** then kecepatan **lambat**

$$\begin{aligned}\alpha - predikat_4 &= \mu_{Sedang}(x) \cap \mu_{Lambat}(x) \\ &= 0,3\end{aligned}$$

[R5] if *score sedang* dan waktu **sedang** then kecepatan **cepat**

$$= 0,33$$

[R6] if *score sedang* dan waktu **cepat** then kecepatan **cepat**

$$\begin{aligned}\alpha - predikat_6 &= \mu_{Sedang}(x) \cap \mu_{Cepat}(x) \\ &= 0\end{aligned}$$

[R7] if *score banyak* dan waktu **lambat** then kecepatan **cepat**

$$\begin{aligned}\alpha - predikat_7 &= \mu_{Banyak}(x) \cap \mu_{Lambat}(x) \\ &= 0\end{aligned}$$

[R8] if *score banyak* dan waktu **sedang** then kecepatan **cepat**

$$\alpha - predikat_8 = \mu_{Banyak}(x) \cap \mu_{Sedang}(x)$$

$$\begin{aligned}
 &= 0 \\
 \text{[R9] if score banyak dan waktu cepat then kecepatan cepat} \\
 \alpha - \text{predikat}_8 &= \mu_{\text{Banyak}}(x) \cap \mu_{\text{Cepat}}(x) \\
 &= 0
 \end{aligned}$$

#### Rule Aggregation

Pada tahap ini, akan memakai fungsi MAX untuk mengambil nilai derajat keanggotaan maksimum dari seluruh konsekuen dalam aplikasi fungsi implikasi. Selanjutnya akan digabungkan seluruh kesimpulan dari setiap aturan sehingga diperoleh solusi fuzzy sebagai berikut:

Variabel Keputusan Lambat

$$[R1] = 0,3$$

$$[R2] = 0,67$$

$$[R3] = 0$$

$$[R4] = 0,3$$

Setelah menggunakan fungsi MAX maka didapatkan nilai tertinggi terletak pada [R2] sebesar 0,67.

Variabel Keputusan Cepat

$$[R5] = 0,33$$

$$[R6] = 0$$

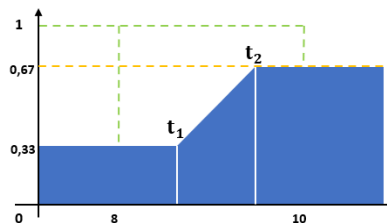
$$[R7] = 0$$

$$[R8] = 0$$

$$[R9] = 0$$

Setelah menggunakan fungsi MAX maka didapatkan nilai tertinggi terletak pada [R5] sebesar 0,33.

Nilai yang telah diperoleh yaitu 0,67 dan 0,33, akan menjadi hasil dari komposisi aturan yang dilihat pada gambar 8.



**Gambar 8** Kurva Hasil Komposisi Aturan

Sumber : Kelvin (2023)

Untuk mendapatkan nilai  $z$ , kita harus terlebih dahulu menentukan nilai  $t_1$  dan  $t_2$  yang merupakan nilai dari titik perpotongan, dengan menggunakan perhitungan sebagai berikut :

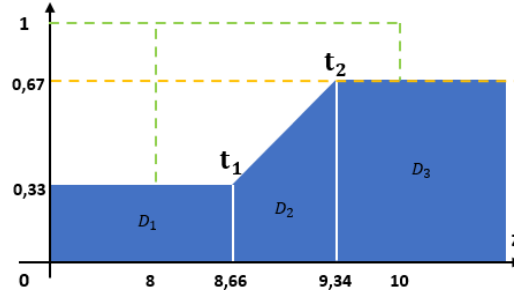
$$\begin{aligned}
 \text{Nilai } t_1 &= \frac{t_1 - 8}{10 - 8} = 0,33 \\
 t_1 &= (0,33 * 2) + 8 \\
 t_1 &= (0,33 * 2) + 8 \\
 t_1 &= 8,66
 \end{aligned}$$

$$\begin{aligned}
 \text{Nilai } t_2 &= \frac{t_2 - 8}{10 - 8} = 0,67 \\
 t_2 &= (0,67 * 2) + 8 \\
 t_2 &= 1,34 + 8
 \end{aligned}$$



$$t_2 = 9,34$$

Berdasarkan hasil perhitungan diatas, didapatkan nilai  $t_1$  sebesar **8,66** dan nilai  $t_2$  sebesar **9,34**. Setelah nilai  $t_1$  dan  $t_2$  didapatkan, langkah selanjutnya adalah menentukan fungsi keanggotaan baru berdasarkan kurva yang dihasilkan, yang dapat ditemukan pada gambar 9.



**Gambar 9** Grafik Kurva Baru  
 Sumber : Kelvin (2023)

Pada gambar 9 menunjukkan wilayah D1, D2, dan D3 pada kurva yang akan digunakan dalam proses *defuzzifikasi*. Langkah untuk menentukan fungsi keanggotaan baru berdasarkan gambar 4.11 adalah sebagai berikut :

$$\mu(z) = \begin{cases} 0,33; & x \leq 8,66 \\ \frac{z-8}{10-8}; & 8,66 \leq x \leq 9,34 \\ 0,67; & x \geq 9,34 \end{cases}$$

Titik potong  $t_1$  dan  $t_2$  akan terbagi menjadi tiga daerah, yaitu D1, D2, dan D3, yang memiliki luas masing-masing  $A_1$ ,  $A_2$ , dan  $A_3$ , serta momen  $M_1$ ,  $M_2$ , dan  $M_3$ , yang kemudian akan digunakan dalam tahap *defuzzifikasi*.

### Defuzzifikasi

Langkah-langkah perhitungan *defuzzifikasi* dengan menggunakan metode *Centroid* terdiri dari:

$$Z^* = \frac{\int \mu(z)zdz}{\int \mu(z)dz}$$

Menghitung **Momen (M)** :

$$M_1 = \int_0^{8,66} 0,33z dz = \left[ 0,33 * \frac{1}{2} z^2 \right]_0^{8,66}$$

$$M_1 = \left( 0,33 * \frac{1}{2} * (8,66)^2 \right) - \left( 0,33 * \frac{1}{2} * (0)^2 \right)$$

$$M_1 = \left( 0,33 * \frac{1}{2} * 75 \right) - 0$$

$$M_1 = 0,165 * 75$$

$$M_1 = \mathbf{12,38}$$

$$M_2 = \int_{8,66}^{9,34} \frac{z-8}{10-8} z dz = \frac{1}{2} \int_{8,66}^{9,34} (z^2 - 8z) dz$$

$$M_2 = \frac{1}{2} \left[ \frac{z^3}{3} - \frac{8z^2}{2} \right]_{8,66}^{9,34}$$

$$M_2 = \frac{1}{2} \left[ \left[ \frac{9,34^3}{3} - \frac{8 * (9,34)^2}{2} \right] - \left[ \frac{8,66^3}{3} - \frac{8 * (8,66)^2}{2} \right] \right]$$

$$M_2 = \frac{1}{2} [[271,59 - 348,94] - [216,49 - 299,98]]$$

$$M_2 = \frac{1}{2} [(-77,35) - (-83,50)]$$

$$M_2 = \frac{1}{2} (6,15)$$

$$M_2 = 3,07$$

$$M_3 = \int_{9,34}^{10} 0,67z \, dz = \left[ 0,67 * \frac{1}{2} z^2 \right]_{9,34}^{10}$$

$$M_3 = \left( 0,67 * \frac{1}{2} * (10)^2 \right) - \left( 0,67 * \frac{1}{2} * (9,34)^2 \right)$$

$$M_3 = \left( 0,67 * \frac{1}{2} * 100 \right) - \left( 0,67 * \frac{1}{2} * 87,23 \right)$$

$$M_3 = (0,67 * 50) - (0,335 * 87,23)$$

$$M_3 = 33,5 - 29,22$$

$$M_3 = 4,28$$

Langkah berikutnya adalah menghitung nilai **luas (A)** :

$$A_1 = \int_0^{8,66} 0,33 \, dz = [0,33 z]_0^{8,66}$$

$$A_1 = (0,33 * 8,66) - 0$$

$$A_1 = 2,86$$

$$A_2 = \int_{8,66}^{9,34} \frac{z - 8}{10 - 8} \, dz = \frac{1}{2} \int_{8,66}^{9,34} (z - 8) \, dz$$

$$A_2 = \frac{1}{2} \left[ \frac{z^2}{2} - 8z \right]_{8,66}^{9,34}$$

$$A_2 = \frac{1}{2} \left[ \left[ \frac{(9,34)^2}{2} - 8 * (9,34) \right] - \left[ \frac{(8,66)^2}{2} - 8 * (8,66) \right] \right]$$

$$A_2 = \frac{1}{2} [[43,62 - 74,72] - [37,50 - 69,28]]$$

$$A_2 = \frac{1}{2} [(-31,10 - (-31,78))]$$

$$A_2 = \frac{1}{2} (0,68)$$

$$A_2 = 0,34$$

$$A_3 = \int_{9,34}^{10} 0,67 \, dz = [0,67 z]_{9,34}^{10}$$

$$\begin{aligned}A_3 &= (0,67 * 10) - (0,67 * 9,34) \\A_3 &= 6,7 - 6,26 \\A_3 &= 0,44\end{aligned}$$

Melakukan perhitungan nilai  $Z$  dengan menggunakan metode *Centroid* :

$$\begin{aligned}Z^* &= \frac{\int \mu(z)zdz}{\int \mu(z)dz} = \frac{M_1 + M_2 + M_3}{A_1 + A_2 + A_3} \\Z^* &= \frac{12,38 + 3,07 + 4,28}{2,86 + 0,34 + 0,44} \\Z^* &= \frac{19,73}{3,64} \\Z^* &= 5,42\end{aligned}$$

Jika jumlah *score* yang diperoleh sebesar **1500** dan waktu yang ditempuh sebesar **120**, maka kecepatan lari *player* akan bernilai **5,42** yang dapat didekati menjadi **5**. Sesuai Tabel 4.3, nilai **5** termasuk dalam kategori **Lambat**.

## SIMPULAN

Game City Street Run dikembangkan dengan metode Multimedia Development Life Cycle serta menerapkan algoritma Fuzzy Mamdani yang digunakan untuk mengatur kecepatan pergerakan player dengan membandingkan parameter score dan parameter waktu dan algoritma Fisher-Yates Shuffle yang digunakan untuk mengacak posisi dalam meletakkan objek. Dari hasil pengujian dengan Black Box Testing, dapat disimpulkan bahwa semua fitur yang terdapat pada game City Street Run telah berjalan sesuai harapan dan berdasarkan hasil pengujian dengan User Acceptance Testing melalui kuesioner, dengan nilai persentase sebesar 85,5%, sehingga dapat disimpulkan bahwa game City Street Run tergolong sangat baik serta dapat diterima dan dimainkan oleh pengguna.

## DAFTAR PUSTAKA

- [1] Mekel, W. J., Sompie A, S. R. U., & Sugiarto, B. A. , "Rancang Bangun Game 3D Pertahanan Kerajaan Bowontehu. Jurnal Teknik Informatika", 14(4), 455–464. 2019.
- [2] Handayani, E., Fatirul, A. N., & Rusmawati, R. D., "Pengaruh metode praktik langsung dengan variasi game terhadap motivasi dan prestrasi belajar teknologi perkantoran", Jurnal Inovasi Teknologi Pendidikan, 7(2), 188–195, 2021.
- [3] Lucky Himawan, I Gusti Ngurah Suryantara, "PENERAPAN GAME EDUKASI TERHADAP BIMBINGAN BELAJAR MATEMATIKA UNTUK KELAS 2 SD", Jurnal Algoritma, Logika dan Komputasi Vol 6, No 2, 2023.
- [4] Eri Kristian, Ignatius Adrian Mastan, "Penerapan Algoritma Finite State Machine Pada Game “Knight And King” Dengan Metode Game Development Life Cycle", JBASE - Journal of Business and Audit Information Systems, Vol 6, No 2, 2023.
- [5] Fahrullazi, M., & Riwinoto, R., “Implementasi Penerapan MDA pada Game Endless Runner 2D CAVE RACER Berbasis Android”, Journal of Applied Multimedia and Networking, 3(2), 19–29, 2019.

<http://jurnal.polibatam.ac.id/index.php/JAMN%0Ahttps://jurnal.polibatam.ac.id/index.php/JAMN/article/view/1636>

- [6] Setia, B., & Ramadan, A., “Penerapan Logika Fuzzy pada Sistem Cerdas”, *Jurnal Sistem Cerdas*, 02, 61–66, 2019.
- [7] Robot, I. J., Tulenan, V., & Paturusi, S. D. E., “Pengembangan Game 3D First Person Shooter Peristiwa Kemerdekaan 14 Februari 1946 Di Sulawesi Utara”, *Jurnal Teknik Informatika*, 15(2), 109–118, 2020.
- [8] Maria Virginia, Jusia Amanda Ginting, "Game Edukasi Match Puzzle Menggunakan Algoritma Fishier-Yates Shuffle Berbasis Android", *Jurnal Algoritma, Logika dan Komputasi* Vol. VI (No.1 ) : 531 - 542. Th. 2023, DOI: <http://dx.doi.org/10.30813/j-alu.v2i2.3530>
- [9] Putro, B. A., Sari, K. A., & Wahid, A., “Penerapan Metode Finite State Machine Pada Game “Escape From Punk Hazard.”, *Jurnal Mahasiswa Teknik Informatika*, 5(1), 71–78, 2021.
- [10] Kristianto, B. A., “APLIKASI AUGMENTED REALITY SEDERHANA BERBASIS MOBILE DENGAN MENGGUNAKAN UNITY”, *Creating a Simple Profile Application Using Android Studio*. November, 0–8. (2021).
- [11] Putra, R. D. E., “Rancang Bangun Aplikasi Augmented Reality Sederhana Berbasis Android Rancang Bangun Aplikasi Augmented Reality Sederhana Berbasis Android Menggunakan Metode Pengembangan MDLC”, December 2022.
- [12] Pratama, W., “PERHITUNGAN SCORE PADA GAME PEMBELAJARAN BAHASA ARAB MENGGUNAKAN METODE FUZZY TYPE-2”, thesis, Universitas Islam Negeri Maulana Malik Ibrahim, 2022.
- [13] Hafizh, F., Maulita, Y., & Khair, H., “Penerapan Logika Fuzzy Logic Pada Enemy Ai Game Horor 3D the Gate of Nightmare Menggunakan Aplikasi Unity3D”. *Jikstra*, 02(02), 86–94, 2020.