

Rancang Bangun Aplikasi Permainan *Escape* Menggunakan *Logika Fuzzy* Dan *Algoritma* *Floyd Warshall*

Adrianus Prabowo, Siska Devella, Yohannes

STMIK GI MDP, Jalan Rajawali No. 14 Palembang, 0711-37640
Program Studi Teknik Informatika, STMIK GI MDP, Palembang
e-mail: adrianus@mhs.mdp.ac.id, siskadevella@mdp.ac.id, yohannesmasterous@mdp.ac.id

Abstrak

Aplikasi permainan ESCAPE merupakan permainan yang mengandalkan player untuk keluar dari labirin tersebut. Penelitian ini menggunakan Logika Fuzzy untuk membuat perilaku komputer menjadi susah ditebak dan Floyd Warshall untuk membuat item jebakan menghalangi player saat bermain. Aplikasi permainan ini dibangun dan dirancang dengan menggunakan Unity 3D dan menggunakan metodologi prototype. Hasil uji dari data sampel menunjukkan bahwa logika fuzzy berhasil diterapkan dalam menentukan perilaku NPC. Hasil uji dari data sampel yang dilakukan menunjukkan bahwa kemunculan item jebakan berhasil diterapkan pada aplikasi permainan ESCAPE.

Kata kunci: *Logika Fuzzy, Floyd Warshall, Prototype, Unity 3D*

Abstract

Application game ESCAPE is a game that relies on players to get out of the maze. This research use Fuzzy Logic to make the computer behaviour became unpredictable during play, and Floyd Warshall algorithm to make trap item appear and disturb player. Game application is build and designed using Unity 3D and using prototype methodology. The test results of sample data showed that fuzzy logic appear successful in changing the behaviour of the NPC. The test result from sample data show that Floyd Warshall make the trap item appear successfully at game application Escape.

Keywords: *Fuzzy Logic, Floyd Warshall, Unity 3D, Prototype*

1. PENDAHULUAN

Permainan komputer (*computer game*) adalah aplikasi hiburan dengan sejumlah standar aturan untuk mengatur perkembangan sesi, yang memiliki nilai kuantitatif tentang kesuksesan dan kegagalan saat pemain memainkannya. Terdapat 4 elemen utama dalam permainan, yaitu representasi, interaksi, konflik dan keamanan [1].

Kecerdasan buatan merupakan salah satu cabang ilmu komputer untuk memberikan suatu pengetahuan pada komputer agar mampu menyelesaikan tugas – tugas layaknya seperti manusia. Saat ini banyak bidang yang memanfaatkan kecerdasan buatan sebagai alat bantu dalam melakukan pekerjaan contohnya bidang kesehatan, industri, penerbangan, militer, game, dan lain – lain [2].

Pada tahun 2019, Oktavia dan Maulidi merancang Aplikasi Game Edukasi Aku Bisa. Aplikasi ini menggunakan logika fuzzy untuk mengatur *reward* yang akan diberikan kepada

pemain. Permainan edukasi yang dibuat berhasil dalam menerapkan logika fuzzy sugeno pada perangkat mobile [3].

Pada tahun 2016, Hermawan dan Putri merancang Aplikasi Game Helitap. Aplikasi ini menggunakan fuzzy mamdani untuk mengatur penilaian *score* pada game. Sistem *score* yang dibuat berhasil membuat pemain menjadi lebih bermotivasi dalam menyelesaikan permainan [4].

Pada tahun 2017, Febrita dan Mahmudy merancang Aplikasi Game Maze Treasure. Aplikasi ini menggunakan finite state machine untuk membuat perilaku monster susah ditebak dan floyd warshall untuk mencari jalur terpendek menuju player. Permainan yang dibuat berhasil dalam menerapkan finite state machine pada perilaku monster dan floyd warshall untuk pencarian jalur terpendek [5].

Pada permainan ini logika fuzzy digunakan untuk mengatur perilaku musuh, secara singkatnya jika jarak NPC dan Player masih jauh maka NPC akan mengejar Player sampai jangkauan NPC.

Menurut Afan Galih Salman (2012) “Sistem fuzzy memiliki keuntungan bila dibandingkan dengan sistem tradisional lainnya, misalkan pada jumlah aturan yang digunakan” [6]. Pemrosesan awal nilai yang banyak menjadi sebuah derajat keanggotaan pada sistem fuzzy mengurangi jumlah nilai yang harus digunakan untuk membuat keputusan. Keuntungan lainnya adalah sistem fuzzy memiliki kemampuan berpikir yang hampir sama dengan kemampuan berpikir yang digunakan oleh manusia. Hal ini dikarenakan sistem fuzzy memiliki kemampuan untuk memberikan tanggapan berdasarkan informasi yang bersifat kualitatif, tidak akurat dan tidak pasti.

Teknik *pathfinding* merupakan teknik yang sering digunakan dalam *game* untuk penentuan jalur terpendek antara titik awal dan titik akhir tanpa melewati hambatan. Algoritma *Floyd Warshall* merupakan metode yang melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang terkait. Artinya solusi – solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi yang dibentuk lebih dari satu [7].

Penelitian – penelitian yang menggunakan algoritma *Floyd Warshall* telah banyak digunakan. Telah dilakukan penelitian membahas mengenai pembangunan sistem untuk menemukan jalur terpendek dengan menyertakan faktor kecepatan dan waktu tempuh perjalanan pada jalan raya wilayah blok M dan kota. Dari penelitian (Kriswanto, Bendi dan Aliyanto, 2014), didapatkan bahwa algoritma *dijkstra* mampu memberikan solusi untuk menemukan lintasan tercepat dan terpendek [8].

Berdasarkan uraian di atas, maka akan dikembangkan sebuah *game*, yaitu *Escape* dengan menggunakan logika fuzzy untuk menentukan perilaku NPC dan *floyd warshall* untuk memunculkan *item* jebakan pada stage.

2. METODE PENELITIAN

Metodologi yang digunakan dalam mengembangkan aplikasi ini adalah metode *prototyping*. *Prototype* merupakan bagian dari produk yang mengekspresikan logika maupun fisik antarmuka eksternal yang ditampilkan. Konsumen potensial menggunakan *prototyping* dan menyediakan masukan untuk tim pengembang sebelum pengembangan skala besar dimulai [9]. Ada beberapa tahap yang dilakukan dalam metodologi *prototyping* sebagai berikut:

2.1 Pengumpulan Kebutuhan

Pada tahap ini dilakukan desain sistem game dengan membuat aturan permainan, menganalisis jumlah level yang akan digunakan, lokasi penempatan *item* dan *enemy* yang berbeda – beda, serta pengenalan logika fuzzy dan floyd warshall pada game *Escape*.

1. Aturan Permainan.

Pada permainan ini ada aturan dan tujuan yang harus dicapai oleh player agar dapat menyelesaikan permainan. Player memiliki health bar dalam setiap permainan yang akan berkurang bila diserang oleh enemy dan menabrak item jebakan. Jadi player harus melarikan diri dari enemy dan lari menuju goal untuk menyelesaikan permainan.

2. Level dan Lokasi *Enemy* serta *Item* pada Permainan.

Permainan memiliki 3 level untuk dapat dimainkan oleh player. Setiap level memiliki item yang penempatan lokasi berbeda – beda dan 1 enemy.

3. Pengenalan Algoritma Logika Fuzzy dan *floyd warshall*

Aplikasi permainan menggunakan algoritma logika fuzzy sebagai penentuan perilaku enemy seperti mengejar player, menyerang player dan menuju goal. Algoritma floyd warshall digunakan untuk memunculkan item jebakan yang terdapat pada stage permainan.

4. Pembelajaran penelitian terdahulu.

Dalam membuat aplikasi ini diperlukan beberapa penelitian sebelumnya mengenai algoritma yang berhubungan dengan pembuatan aplikasi permainan Escape.

2.2 Mendesain Prototype

Pada tahap ini dilakukan perancangan aplikasi permainan yang akan dibuat mulai dari perancangan aplikasi. Berikut ini adalah rancangan aplikasi yang akan dibuat:

1. Menu Awal Permainan

Rancangan tampilan menu awal dapat dilihat pada Gambar 1



Gambar 1. Rancangan Tampilan Menu Pembuka

2. Menu How to Play

Pada aplikasi ini terdapat menu yang akan memberikan penjelasan cara bermain yang dapat dilihat pada Gambar 2



Gambar 2. Rancangan Tampilan Cara Bermain

3. Menu Pembuat

Dibuat sebuah halaman tentang pembuat aplikasi yang dapat dilihat pada Gambar 3



Gambar 3. Rancangan Tampilan Tentang Pembuat

4. Permainan

Tombol Mulai Permainan akan mengarahkan pada scene Permainan yang dapat dilihat pada Gambar 4



Gambar 4. Rancangan Tampilan Permainan

2. 3 Membangun Sistem

Pada tahap ini, rancangan yang telah selesai akan mulai dibangun. Berikut sistem yang telah dibangun:

1. Player pada permainan

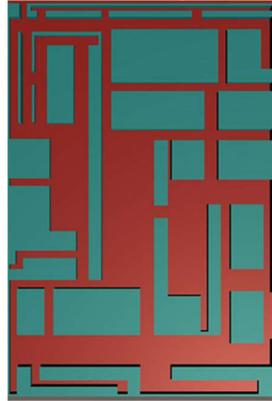
Objek sebagai player didapat melalui Asset Store Unity. Rancangan objek sebagai player dapat dilihat pada Gambar 5



Gambar 5. Karakter Player

2. Labirin Stage 1

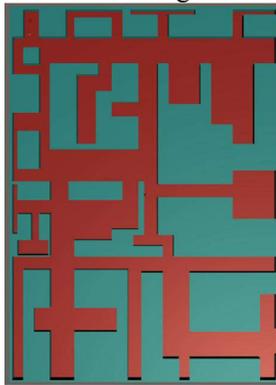
Gambar 6 menunjukkan tampilan akhir dari stage 1



Gambar 6. Desain Stage 1

3. Labirin Stage 2

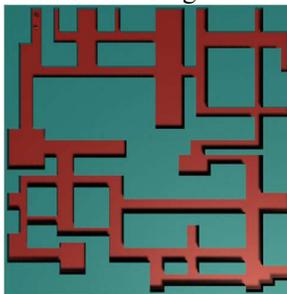
Gambar 7 menunjukkan tampilan akhir dari stage 2



Gambar 7. Desain Stage 2

4. Labirin Stage 3

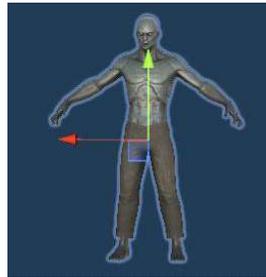
Gambar 8 menunjukkan tampilan akhir dari stage 3



Gambar 8. Desain Stage 3

5. Enemy pada permainan

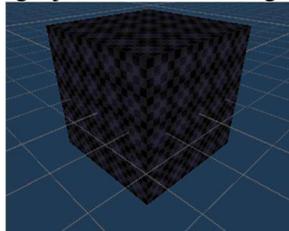
Objek enemy yang digunakan dalam aplikasi dapat dilihat pada Gambar 9



Gambar 9. Karakter Enemy

6. Item pada permainan

Objek item yang berfungsi sebagai jebakan untuk mengurangi HP player. Gambar 10



Gambar 10. Item pada Game

2.4 Evaluasi Prototype

Pada tahap ini dilakukan evaluasi terhadap rancangan game. Akan dilihat apakah aplikasi yang sudah dibuat sudah sesuai dengan yang diharapkan. Jika tidak, maka *prototyping* direvisi dengan mengulang langkah sebelumnya.

2.5 Perubahan Rancangan dari Prototype

Jika ada permasalahan atau tidak sesuai harapan pada pembuatan sistem aplikasi, maka dilakukan perubahan desain dengan kembali ke langkah 2 yaitu mendesain *prototype*. Jika tidak ada perubahan maka dilanjutkan ke tahapan berikutnya.

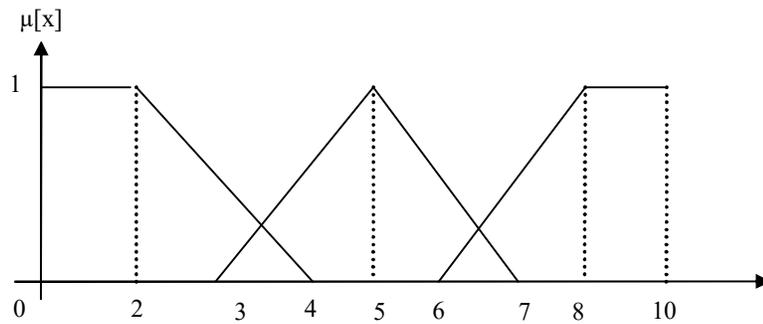
2.6 Pembangunan Sistem

Sistem yang sudah dibangun dikembangkan dan digunakan oleh *user* sesuai dengan kebutuhan dan manfaat dari game.

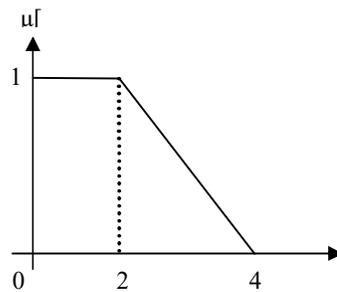
3. HASIL DAN PEMBAHASAN

3.1 Implementasi Logika Fuzzy Terhadap Aplikasi Permainan “Escape”

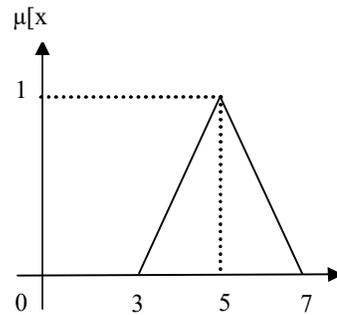
Logika fuzzy diterapkan dalam pengambilan keputusan gerakan *Enemy*. Terdapat dua parameter input dan satu parameter output yang digunakan pada logika fuzzy. Perancangan akan dijelaskan melalui tahapan berikut:



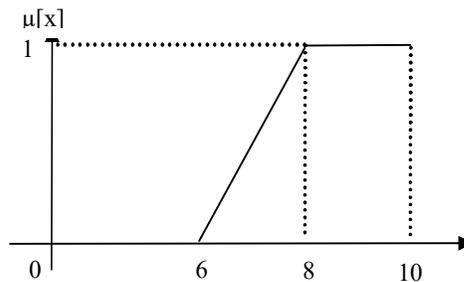
Gambar 11. Fungsi Keanggotaan *Distance to Base* dan *Distance to Player*



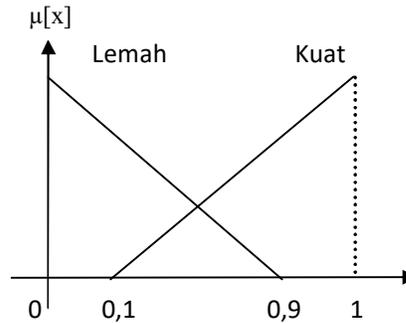
Gambar 12. Fungsi Keanggotaan *Distance to Base* dan *Distance to Player Dekat*



Gambar 13. Fungsi Keanggotaan *Distance to Base* dan *Distance to Player Sedang*



Gambar 14. Fungsi Keanggotaan *Distance to Base* dan *Distance to Player Jauh*



Gambar 15. Fungsi Keanggotaan Variabel Output

Rule – rule yang digunakan dalam proses fuzzy:

Tabel 1. Rule Perilaku NPC pada C#

No Rule	Rules
R1	IF DistanceToBase is Dekat AND DistanceToPlayer is Dekat THEN KondisiMengejar Kuat.
R2	IF DistanceToBase is Dekat AND DistanceToPlayer is Sedang THEN KondisiMengejar Kuat.
R3	IF DistanceToBase is Dekat AND DistanceToPlayer is Jauh THEN KondisiMengejar Lemah.
R4	IF DistanceToBase is Sedang AND DistanceToPlayer is Dekat THEN KondisiMengejar Kuat.
R5	IF DistanceToBase is Sedang AND DistanceToPlayer is Sedang THEN KondisiMengejar Kuat.
R6	IF DistanceToBase is Sedang AND DistanceToPlayer is Jauh THEN KondisiMengejar Lemah.
R7	IF DistanceToBase is Jauh AND DistanceToPlayer is Dekat THEN KondisiMengejar Kuat.
R8	IF DistanceToBase is Jauh AND DistanceToPlayer is Sedang THEN KondisiMengejar Lemah.
R9	IF DistanceToBase is Jauh AND DistanceToPlayer is Jauh THEN KondisiMengejar Lemah.

3.2 Implementasi Floyd Warshall Terhadap Aplikasi Permainan “Escape”

```

public void FloydWarshall(float[,] graph, int verticesCount)
{
    float[,] distance = new float[verticesCount, verticesCount];

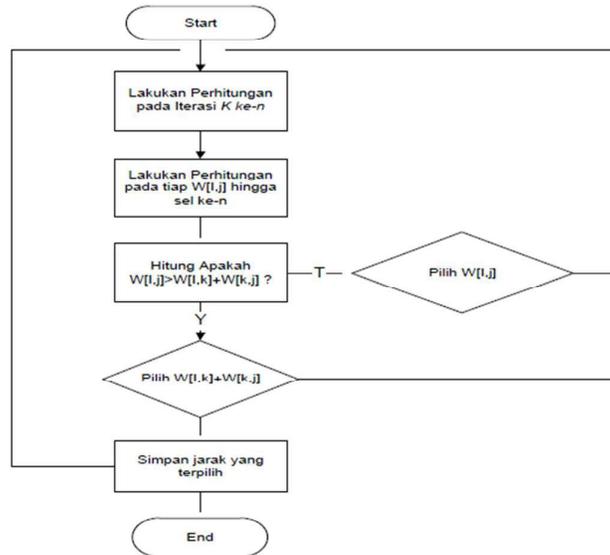
    for (int i = 0; i < verticesCount; ++i)
    {
        for (int j = 0; j < verticesCount; ++j)
        {
            distance[i, j] = graph[i, j];
        }
    }

    for(int k = 0; k < verticesCount; ++k)
    {
        for(int i = 0; i < verticesCount; ++i)
        {
            for(int j = 0; j < verticesCount; ++j)
            {
                if (distance[i, k] + distance[k, j] < distance[i, j])
                {
                    distance[i, j] = distance[i, k] + distance[k, j];
                }
            }
        }
    }
}
    
```

Gambar 16. Perhitungan Algoritma Floyd Warshall

Tahapan pada Gambar merupakan proses perhitungan dari nilai yang di simpan pada bahasa pemrograman C#.

Berikut merupakan alur flowchart dari algoritma Floyd Warshall yang dapat dilihat pada gambar



Gambar 17. Flowchart Algoritma *Floyd Warshall*

Proses penentuan nilai minimum algoritma *Floyd Warshall* dapat dituliskan sebagai berikut:

1. Pada iterasi ke – 1, setiap sel matrik dilakukan pengecekan apakah jarak antar titik awal lebih besar daripada penjumlahan antara titik asal ke titik tujuan (titik tujuan = iterasi ke – 1) dengan jarak titik asal (titik asal = iterasi ke – 1) ke titik tujuan. Dengan kata lain apakah $W[I,j] > W[I,k] + W[k,j]$.
 2. Jika iya, maka jarak antar dua titik awal diganti dengan penjumlahan antar jarak titik asal (titik asal = titik iterasike – 1) ke titik tujuan ($W[I,k] + W[k,j]$).
 3. Jika tidak, maka yang digunakan yaitu jarak antar dua titik awal ($W[I,j]$).
- Proses iterasi dilakukan hingga pada iterasi terakhir (jumlah iterasi = jumlah total titik).

3.3 Implementasi Terhadap Aplikasi Permainan “*Escape*”

Berikut akan dibahas secara sederhana penerapan logika fuzzy dan *floyd warshall* pada permainan *Escape*:

1. Logika Fuzzy

Dengan adanya perubahan pada jarak, maka *state* karakter Enemy akan berubah antara menyerang player, mengejar player dan menuju goal. Pengujian dilakukan oleh satu player dan satu NPC.



Gambar 18. Kondisi Enemy Saat Menuju *Goal/Exit*

Misalkan $DistanceToBase = 5.86$ dan $DistanceToPlayer = 2.80$

$$DistanceToBase\ Sedang = \frac{7-5.864627}{7-5} = \frac{1.135373}{2} = 0.5676865 \quad (1)$$

$$DistanceToPlayer\ Dekat = \frac{4-2.806418}{4-2} = \frac{1.193582}{2} = 0.596791 \quad (2)$$

Metode yang digunakan untuk mencari nilai a-predikat adalah metode minimum.

Berikut adalah aturan yang memiliki nilai a-predikat:

Aturan ke - 4 [R4]

[R4] IF Distance to Base is Sedang AND Distance to Player is Dekat THEN

KondisiMengejar Kuat.

$$\alpha - predikat_4 = \min(\mu_{DistanceToBase}[5.864627], \mu_{DistanceToPlayer}[2.806418]) = 0.5676865 \quad (3)$$

Lalu dicari nilai output dengan rumus :

$$\mu[x]Lemah = \begin{cases} \frac{0,9-z}{0,9} & , 0 \leq z \leq 0,9 \\ 0 & , z \geq 0,9 \end{cases} \quad (4) \quad \mu[x]Kuat = \begin{cases} 0 & , z \leq 0,1 \\ \frac{z-0,1}{0,9} & , 0,1 \leq z \leq 1 \end{cases} \quad (5)$$

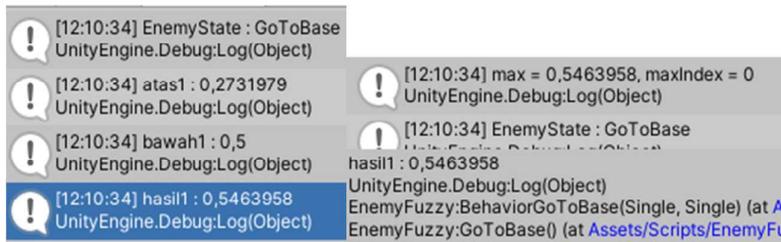
$$\mu[R4]Kuat = \frac{z_4-0.1}{0.9} = 0.5676865$$

$$z_4 - 0.1 = 0.51091785$$

$$z_4 = 0.51091785 + 0.1 = 0.61091785 \quad (6)$$

Lalu di lakukan Defuzzifikasi :

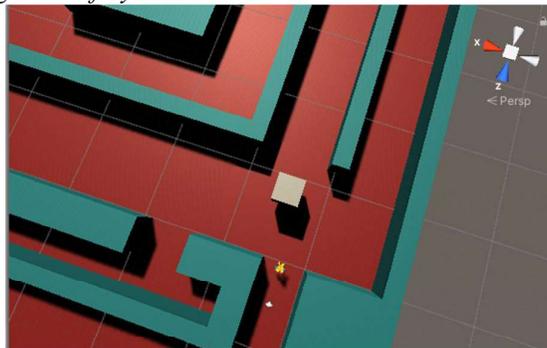
$$z^* = \frac{\alpha_4 * z_4}{\alpha_4} = \frac{0.5676865 * 0.61091785}{0.5676865} = 0.61091785 \quad (7)$$



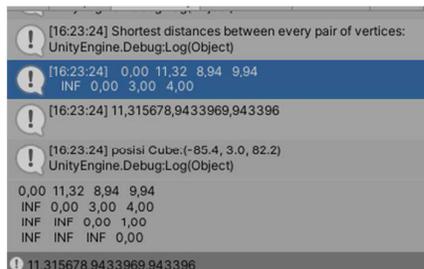
Gambar 19. Log Saat Enemy State GoToBase

2. Floyd Warshall

Dengan menggunakan floyd warshall item akan muncul dan menghalangi player.



Gambar 20. Kondisi Saat Kemunculan Item1



Gambar 21. Log yang Menunjukkan Kemunculan Item

4. KESIMPULAN

Berdasarkan hasil implementasi dan uji coba yang dilakukan kesimpulan dari pembuatan game ini, yaitu:

- Logika Fuzzy sangat cocok untuk menentukan perilaku karakter.
- Berdasarkan hasil pengujian yang dilakukan, *Fuzzy Logic* dan *Floyd Warshall* berhasil diterapkan pada aplikasi permainan *Escape*.

5. SARAN

Berdasarkan hasil pembuatan game ini, saran yang dapat direkomendasikan adalah:

- Variabel input fuzzy baiknya selain jarak dan lebih banyak dalam variasinya.
- Perbaikan pada *asset* game yang lebih bagus baik dari bagian 2D dan 3D.
- Penambahan pilihan stage akan membuat game lebih menarik dan tidak membuat pemain menjadi mudah bosan.
- Perbaikan pada menu sangat diperlukan.
- Game dapat dikembangkan untuk digunakan pada sistem operasi lain, contohnya Android.
- Penambahan kunci item untuk membuka pintu keluar akan membuat game lebih menantang bagi player.

DAFTAR PUSTAKA

- [1] Purba, K., Hasanah, R. Dan Muslim, M., 2013, *Implementasi Logika Fuzzy Untuk Mengatur Perilaku Musuh Dalam Game Bertipe Action – RPG*, Jurnal *EECCIS*, No.1, Vol.7, 15 – 20, <https://jurnaleeccis.ub.ac.id/index.php/eccis/article/view/196/169.pdf>.
- [2] M. Yusuf Febryan, M. Zainul Umri Muttaqin, Iis Pradesan, Y., 2017, *Penerapan Algoritma Floyd Warshall dan Metode Finite State Machine pada Aplikasi Permainan “Maze Treasure”*, Fakultas Teknik Informatika, STMIK MDP, Palembang.
- [3] Oktavia, C. A. dan Maulidi, R., 2019, *Penerapan Logika Fuzzy Sugeno Untuk Penentuan Reward pada Game Edukasi Aku Bisa*, *JUTI: Jurnal Ilmiah Teknologi Informasi*, No.2, Vol.17, 117 – 124, https://www.researchgate.net/profile/Rakhmad_Maulidi/publication/335421117_PENERAPAN_LOGIKA_FUZZY_SUGENO_UNTUK_PENENTUAN_REWARD_PADA_GAME_EDUKASI_AKU_BISA/links/5d64cb5b299bf1f70b0ec2a7/PENERAPAN-

LOGIKA-FUZZY-SUGENO-UNTUK-PENENTUAN-REWARD-PADA-GAME-
EDUKASI-AKU-BISA.Pdf

- [4] Hermawan, L. dan Putri, A. N., 2016, *Penerapan Algoritma Fuzzy Mamdani Untuk Mengatur Game Scoring pada Game Helitap*, Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2014(SEMANTIK 2014), Semarang, 15 November,
- [5] Febrita R. E. dan Mahmudy, W. F., 2017, *Modified Genetic Algorithm for High School Time – Table Scheduling With Fuzzy Time Window*, *International Conference on Sustainable Information Engineering and Technology*, 88 – 92.
- [6] Afan Galih Salman, 2012, *Pemodelan Dasar Sistem Fuzzy*, <https://socs.binus.ac.id/2012/03/02/pemodelan-dasar-sistem-fuzzy/>, Diakses Tanggal 1 Oktober 2020.
- [7] M. Zani Wibowo, Lutfi Pratama, I. P., 2016, *Rancang Bangun Aplikasi Permainan “Help Your Mom” Menggunakan Algoritma Floyd – Warshall*, Fakultas Teknik Informatika, STMIK MDP, Palembang.
- [8] Kriswanto, Y. R., Bendi, R. K. dan Aliyanto, A., 2014, *Penentuan Jarak Terpendek Rute Transmisi Dengan Algoritma Floyd – Warshall*, *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2014(SEMANTIK 2014)*, Semarang, 15 November.
- [9] Simarmata, Janner 2010, *Rekayasa Perangkat Lunak*, Andi, Yogyakarta.