

PEMBUATAN GAME 2D BERBASIS ANDROID “JAWA WARRIOR” DENGAN METODE FINITE STATE MACHINE DAN FUZZY LOGIC

Syahdan Izzur Rosuli, Suryo Adi Wibowo, Eko Heri Susanto

Teknik Informatika, Institut Teknologi Nasional Malang

Jalan Raya Karanglo km 2 Malang, Indonesia

izzursyahdan@gmail.com

ABSTRAK

Industri game Indonesia tumbuh pesat dengan nilai US\$1,47 miliar (Rp25 triliun), namun kontribusi lokal hanya 0,49%, sehingga dominasi asing berpotensi merugikan hingga Rp30 triliun. Untuk mendukung game lokal, dikembangkan Jawa Warrior, game aksi 2D berbasis level dengan *combat* santai namun menantang. Sistem AI menggunakan FSM untuk transisi perilaku musuh dan logika fuzzy Mamdani untuk menentukan jenis serangan boss berdasarkan *health point* dan jarak pemain. Hasil dari pengujian meliputi Pengujian AI dan kontrol mencapai 100% keberhasilan dari total 7 pengujian yang dilakukan, uji kompatibilitas pada 9 perangkat menunjukkan 77,78% berjalan lancar, dan penilaian pengguna menempatkan game pada kategori cukup (56,67%), menandakan kualitas yang baik secara keseluruhan.

Kata kunci : Game, Action, 2D, FSM, Fuzzy, Mamdani,

1. PENDAHULUAN

Game merupakan produk digital interaktif yang memadukan elemen suara, gambar, teks, dan animasi dengan aturan serta tujuan yang jelas. Industri game di Indonesia saat ini berkembang pesat, dengan nilai pertumbuhan mencapai US\$1,47 miliar (sekitar Rp25 triliun) dan menempati posisi ke-16 dunia menurut DataNewszoo. Namun, kontribusi game lokal masih sangat rendah, hanya sekitar 0,49% dari total pasar, sehingga dominasi produk asing diperkirakan menimbulkan kerugian hingga Rp30 triliun per tahun. Kondisi ini menunjukkan perlunya peningkatan produksi game lokal yang kompetitif agar mampu menarik minat pemain dalam negeri.[1]

Salah satu elemen penting dalam game adalah *Non-Player Character* (NPC), yang berperan memberikan interaksi, membangun suasana, dan menambah tantangan bagi pemain. Pada banyak game 2D bergenre aksi, sistem AI musuh masih menggunakan pola serangan sederhana dan repetitif, sehingga kurang memberikan variasi perilaku. Hal ini dapat membuat pemain cepat bosan. Oleh karena itu, dibutuhkan pengembangan AI yang mampu menghasilkan perilaku NPC lebih dinamis dan adaptif, sehingga pengalaman bermain menjadi lebih menarik.[2]

Genre action menjadi salah satu jenis permainan yang digemari oleh berbagai usia, mulai dari anak-anak hingga orang dewasa. Dalam genre ini, kecerdasan buatan (AI) yang diterapkan pada NPC memiliki peran yang sangat penting. Misalnya, musuh dalam game dapat diprogram untuk mengejar pemain ketika berada dalam jarak patroli tertentu, dan akan mulai menyerang saat pemain masuk ke dalam jangkauan serangnya.[3]

Berdasarkan permasalahan tersebut, penelitian ini mengembangkan Jawa Warrior, game aksi-petualangan 2D berbasis level yang ditujukan untuk anak-anak hingga remaja, sekaligus mengenalkan

budaya Jawa melalui visual, karakter, dan latar permainan. Metode *Finite State Machine* (FSM) digunakan untuk mengatur transisi perilaku NPC musuh, sedangkan logika fuzzy Mamdani digunakan untuk menentukan tipe serangan boss berdasarkan jarak dan *health point*. Pemilihan format 2D dilakukan karena pengembangan game 2D relatif lebih efisien dalam hal sumber daya, mudah diakses pada perangkat Android berspesifikasi rendah hingga menengah, dan tetap mampu memberikan pengalaman bermain yang menyenangkan jika didukung dengan desain grafis dan AI yang baik. Dengan pendekatan ini, diharapkan Jawa Warrior mampu memberikan pengalaman bermain yang menyenangkan sekaligus memperkenalkan budaya lokal kepada pemain.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Mengacu pada penelitian oleh Suryo berjudul "Implementasi Fuzzy Logic Sebagai Penentu Jumlah Konsumsi Kalori Penderita Diabetes Melitus", prinsip dasar Fuzzy Logic yang bersifat fleksibel dan adaptif ternyata juga dapat diterapkan dalam konteks yang berbeda, termasuk dalam pengembangan game. Dalam penelitian ini, metode Fuzzy Logic Mamdani dimanfaatkan untuk membentuk perilaku musuh (*enemy AI*) dalam game, sehingga musuh dapat merespons kondisi permainan dengan lebih dinamis dan realistis—mirip dengan cara kerja sistem fuzzy di bidang kesehatan yang menyesuaikan respons berdasarkan parameter tubuh manusia. [4].

Dalam penelitian yang dilakukan oleh Maldy berjudul "Rancang Bangun Game *Lost In The Zombie Apocalypse*", diterapkan metode FSM untuk mengatur perilaku musuh dalam game tersebut. Penelitian ini bertujuan menguji bagaimana FSM dapat mengendalikan gerakan dan respons musuh secara terstruktur. Hasil pengujian menunjukkan bahwa musuh mampu bertindak sesuai dengan skenario yang

dirancang, dan seluruh fitur yang menggunakan metode FSM dapat berfungsi dengan baik sesuai harapan. [5].

Dalam penelitian yang dilakukan oleh Rizki berjudul "Perancangan Game 2D 'Barbossa The Savior' Menggunakan *Finite State Machine*", menerangkan perihal metode FSM sangat sesuai digunakan pada NPC dalam game bergenre action maupun strategi. Metode ini tidak hanya meningkatkan tingkat interaksi dalam permainan, tetapi juga mendorong pemain untuk berpikir strategis, karena mereka harus merespons berbagai situasi yang berubah-ubah sesuai dengan logika FSM yang diterapkan pada karakter musuh.[6]

2.2. Game Action

Game bergenre action merupakan salah satu jenis permainan digital yang paling digemari oleh berbagai kelompok usia. Genre ini mengandalkan elemen kecepatan, ketangkasan, koordinasi mata dan tangan, serta kemampuan mengambil keputusan secara cepat. Umumnya, game action menghadirkan tantangan berupa musuh, rintangan, atau misi dimana player diharuskan menamatkan dalam jangka waktu sesuai aturan pada game atau dengan tingkat kesulitan yang terus meningkat. Dalam proses pengembangannya, genre ini memberikan peluang besar untuk diterapkannya FSM sebagai otak dari musuh atau logika fuzzy, yang memungkinkan NPC merespons strategi pemain secara langsung.[7]

2.3. Fuzzy Logic

Secara etimologis, istilah *fuzzy* berarti kabur atau tidak jelas. Dalam konteks logika, hal ini menunjukkan bahwa suatu nilai bisa berada di antara benar dan salah secara bersamaan, tidak mutlak. Logika fuzzy berbeda dari logika konvensional (logika tegas) dimana terdapat 2 nilai pasti, yaitu 0 (salah/tidak) dan 1 (benar/ya). Sebaliknya, logika fuzzy menggunakan *range* nilai yang terdapat dari 0 hingga 1, di mana tingkat kebenaran suatu nilai ditentukan berdasarkan tingkat keanggotaannya dalam suatu himpunan. Untuk menggambarkan tingkat keanggotaan tersebut, digunakan fungsi keanggotaan (*membership function*) yang menunjukkan bagaimana suatu data input dipetakan ke dalam derajat keanggotaan dengan nilai antara 0 sampai 1. Nilai derajat ini diperoleh melalui pendekatan tertentu, tergantung bentuk fungsi yang digunakan.[8]

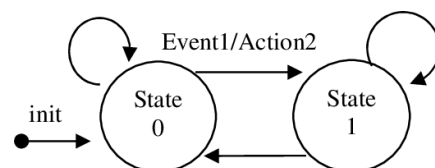
2.4. Fuzzy Mamdani

Salah satu pendekatan logika fuzzy yang paling banyak digunakan adalah metode Mamdani, yang pertama kali dipertontonkan oleh Ebrahim Mamdani pada tahun 1975. Fuzzy ini menerapkan aturan berbasis bahasa alami dalam bentuk *IF-THEN*, mirip dengan cara manusia membuat keputusan secara logis namun tidak kaku. Fuzzy Mamdani sangat cocok diterapkan pada sistem yang membutuhkan penilaian deskriptif dan bersifat evaluatif, seperti dalam sistem kontrol atau perilaku AI dalam game. Pendekatan ini

memungkinkan sistem menjadi lebih adaptif, serta mampu mengelola data yang bersifat tidak pasti atau ambigu secara lebih efektif.[9].

2.5. Finite State Machine

FSM ialah cara untuk mengendalikan bagaimana suatu sistem berperilaku atau bergerak melalui berbagai langkah.. Metode FSM terdiri dari 3 komponen utama, yaitu *state* (keadaan), *event* (kejadian), dan *action* (aksi). Sistem dapat berpindah dari satu keadaan ke keadaan lain melalui proses yang disebut transisi. Setiap perubahan umumnya dipicu oleh suatu peristiwa dan diikuti oleh tindakan tertentu sebagai reaksi sistem menanggapi *inputan* yang masuk. Tindakan ini terdiri dari langkah-langkah yang *simple* maupun yang termasuk kategori rumit, tergantung pada kebutuhan sistem.[10].



Gambar 1. Diagram FSM [10]

2.6. Unity

Unity ialah *game engine* dengan developernya adalah Unity Technologies dan dibuat sejak tahun 2005. Platform ini banyak digunakan dalam pembuatan game 2D dan 3D, serta aplikasi berbasis VR, dengan dukungan lintas *platform* seperti Android, iOS, Windows, dan lainnya. *Unity* dikenal dengan antarmukanya yang ramah pengguna dan mendukung bahasa pemrograman C#. *Engine* ini menggunakan arsitektur berbasis komponen, di mana setiap elemen dalam game dibangun dari *GameObject* yang dilengkapi berbagai komponen seperti transform, fisika, dan skrip. Didukung oleh ekosistem yang lengkap seperti *Unity Hub*, *Asset Store*, serta berbagai teknologi *rendering* seperti URP dan HDRP, *Unity* menjadi salah satu alat pengembangan paling populer dan fleksibel di dunia pengembangan game sekarang[11].

3. METODE PENELITIAN

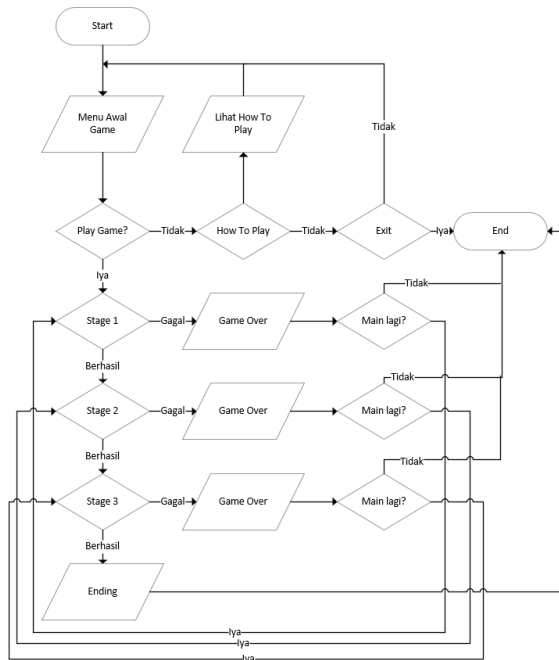
3.1. Alur Game

Pada saat game pertama dimulai, akan menampilkan scene menu yang berisi opsi untuk *play game*, *How To Play* dan *exit*. Ketika player memilih *play game*, maka scene akan berganti untuk mulai bermain level 1. Jika berhasil menyelesaikan level 1 dan mengalahkan boss level 1, player akan memasuki level 2 dan jika gagal, akan muncul death screen yang dimana terdapat pilihan untuk restart level atau quit.

Jika player memilih restart level, maka player akan memulai level 1 dari awal dan jika memilih quit, maka player akan kembali ke main menu. Hal ini juga berlaku hingga level 3. Ketika player telah menyelesaikan level 3, maka akan muncul button untuk memasuki scene ending. Setelah ending selesai

dimainkan, maka game selesai dan akan kembali ke main menu.

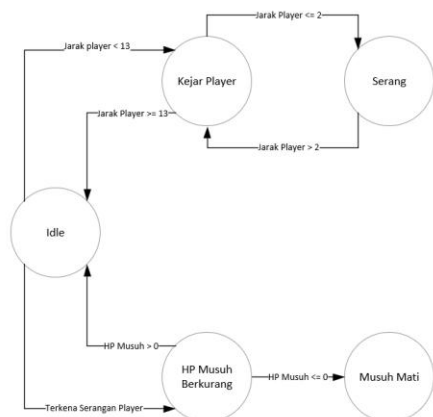
Jika player memilih opsi *how to play* pada main menu, maka akan diarahkan ke layar ke layar yang berisi tentang tujuan dalam game, fungsi tombol yang ada dalam game, dan penjelasan tentang cara kerja healthpoint dan mana pada game.



Gambar 2. Flowchart Alur Game Jawa Warrior

3.2. Perancangan Metode Finite State Machine

Penerapan FSM terdapat pada NPC musuh dimana setiap musuh memiliki aktivitas AI yang berbeda beda. Berikut merupakan aktivitas pada musuh.



Gambar 3. Finite State Machine Pada NPC enemy

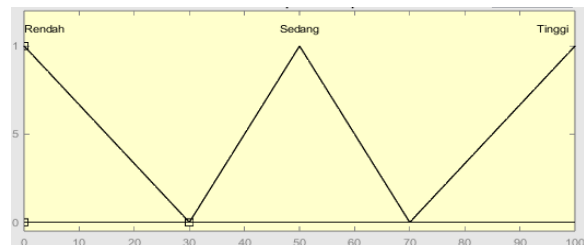
3.3. Fuzzy Mamdani Pada AI Enemy

Pada game “Jawa Warrior”, fuzzy digunakan untuk menentukan aksi selanjutnya yang akan dilakukan oleh boss. Terdapat 2 masukan yaitu *Healthpoint* dan *distance* dan terdapat 4 output pada fuzzy yaitu *Kejar Player*, *Serang Dekat*, *Serang Jauh*, dan *Special Attack*.

Tabel 1. Keanggotaan *Healthpoint*

Fungsi Keanggotaan	Domain	Parameter
Rendah	0-30	[0,15,30]
Sedang	30-70	[30,50,70]
Tinggi	70-100	[70,100,100]

Dari tabel fungsi keanggotaan *healthpoint* didapat grafik seperti pada gambar 4.

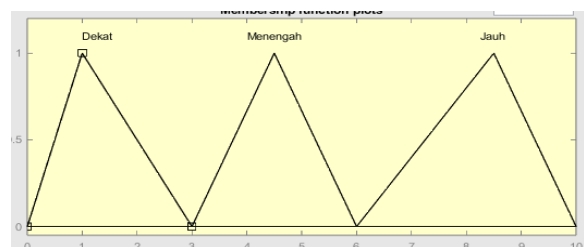


Gambar 4. Grafik Keanggotaan *Healthpoint*

Tabel 2. Keanggotaan *Distance*

Fungsi Keanggotaan	Domain	Parameter
Dekat	0-3	[0, 1.5, 3]
Menengah	3-6	[3, 4.5, 6]
Jauh	6-10	[6,8.5, 10]

Dari tabel fungsi keanggotaan *distance* akan didapat hasil pada gambar 5.



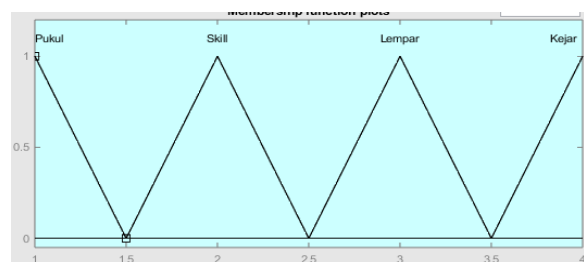
Gambar 5. Grafik Keanggotaan *Distance*

Selanjutnya ialah domain dari output yaitu aksi yang dilakukan boss pada serangan berikutnya.

Tabel 3. Keanggotaan *Output*

Fungsi Keanggotaan	Domain	Parameter
Pukul	0.5-1.5	[0.5, 1, 1.5]
Skill	1.5-2.5	[1.5, 2, 2.5]
Lempar	2.5-3.5	[2.5, 3, 3.5]
Kejar	3.5-4.5	[3.5, 4, 4.5]

Dari tabel fungsi keanggotaan aksi diatas menghasilkan grafik pada gambar 6.



Gambar 6. Grafik Keanggotaan Aksi

Dan berikut merupakan pendeklarasian fuzzy rule

- IF Health Rendah AND Distance Rendah THEN Aksi = Pukul
- IF Health Rendah AND Distance Menengah THEN Aksi = Skill
- IF Health Rendah AND Distance Jauh THEN Aksi = Lempar
- IF Health Sedang AND Distance Rendah THEN Aksi = Pukul
- IF Health Sedang AND Distance Menengah THEN Aksi = Skill
- IF Health Rendah AND Distance Jauh THEN Aksi = Kejar
- IF Health Tinggi AND Distance Rendah THEN Aksi = Pukul
- IF Health Tinggi AND Distance Menengah THEN Aksi = Kejar
- IF Health Tinggi AND Distance Jauh THEN Aksi = Kejar

Contoh perhitungan Defuzzifikasi ketika *enemy* boss memiliki inputan berupa *healthpoint* = 40% dan jarak boss dari *player* adalah 5:
Fuzzifikasi

Variabel Distance

- Dekat ($\mu_{rendah}(5)$)
 - Domain: [0, 0, 3]
 - Karena $5 > 3$, maka: $\mu_{dekat}(5) = 0$
- Sedang ($\mu_{sedang}(5)$)
 - Domain: [3, 4.5, 6]
 - Menggunakan rumus *triangular*:

$$\mu_{sedang}(x) = \frac{6-x}{6-4.5}, \text{ untuk } 3 \leq x \leq 5$$

$$\mu_{sedang}(12) = \frac{6-5}{6-4.5} = \frac{1}{1.5} = 0.6$$
- Jauh ($\mu_{jauh}(5)$)
 - Domain: [6, 8, 10]
 - Karena $5 < 6$, maka: $\mu_{jauh}(5) = 0$

Hasil fuzzifikasi distance:

- Dekat: 0
- Sedang: 0.6
- Jauh: 0

Variabel Darah

- Rendah ($\mu_{rendah}(40)$)
 - Domain: [0, 0, 30]
 - Karena $40 > 30$, maka: $\mu_{rendah}(40) = 0$
- Sedang ($\mu_{sedang}(40)$)
 - Domain: [30, 50, 70]
 - Menggunakan rumus *triangular*:

$$\mu_{sedang}(y) = \frac{y-30}{50-30}, \text{ untuk } 30 \leq y \leq 50$$

$$\mu_{sedang}(y) = \frac{40-30}{50-30} = \frac{10}{20} = 0.5$$
- Tinggi ($\mu_{tinggi}(40)$)
 - Domain: [70, 100, 100]
 - Karena $40 < 70$, maka: $\mu_{tinggi}(40) = 0$

Hasil fuzzifikasi darah:

- Rendah: 0
- Sedang: 0.5
- Tinggi: 0

Berdasarkan derajat keanggotaan yang telah dihitung, kita akan menerapkan aturan fuzzy yang baru:

- Rule 1 = $\text{Min}(\mu_{HealthRendah}, \mu_{DistanceDekat})$
 $\mu_{SerangDekat} = \text{Min}(0, 0) = 0$
- Rule 2 = $\text{Min}(\mu_{HealthRendah}, \mu_{DistanceSedang})$
 $\mu_{SerangJauh} = \text{Min}(0, 0.6) = 0$
- Rule 3 = $\text{Min}(\mu_{HealthRendah}, \mu_{DistanceJauh})$
 $\mu_{SpecialAttack} = \text{Min}(0, 0) = 0$
- Rule 4 = $\text{Min}(\mu_{HealthSedang}, \mu_{DistanceDekat})$
 $\mu_{SerangDekat} = \text{Min}(0.5, 0) = 0$
- Rule 5 = $\text{Min}(\mu_{HealthSedang}, \mu_{DistanceSedang})$
 $\mu_{SerangJauh} = \text{Min}(0.5, 0.6) = 0.5$
- Rule 6 = $\text{Min}(\mu_{HealthSedang}, \mu_{DistanceJauh})$
 $\mu_{KejarPlayer} = \text{Min}(0.5, 0) = 0$
- Rule 7 = $\text{Min}(\mu_{HealthTinggi}, \mu_{DistanceDekat})$
 $\mu_{SerangDekat} = \text{Min}(0, 0) = 0$
- Rule 8 = $\text{Min}(\mu_{HealthTinggi}, \mu_{DistanceSedang})$
 $\mu_{KejarPlayer} = \text{Min}(0, 0.6) = 0$
- Rule 9 = $\text{Min}(\mu_{HealthTinggi}, \mu_{DistanceJauh})$
 $\mu_{KejarPlayer} = \text{Min}(0, 0) = 0$

Sehingga menurut perhitungan pada *rule* yang telah dilakukan, didapatkan hasil akhir seperti berikut:

- Pukul : 0
- Skill : 0.5
- Lempar : 0
- Kejar : 0

Berikutnya merupakan defuzzifikasi menggunakan metode *centroid* untuk skill : [1.5, 2, 2.5]

- Titik kiri potong (z1): $\mu = 0.5$ pada garis naik

$$\frac{z1 - 1.5}{2 - 1.5} = 0.5$$

$$\frac{z1 - 1.5}{0.5} = 0.5$$

$$z1 - 1.5 = 0.5 \times 0.5$$

$$z1 - 1.5 = 0.25$$

$$z1 = 0.25 + 1.5 = 1.75$$
- Titik kanan potong (z2): $\mu = 0.5$ pada garis turun

$$\frac{2.5 - z2}{2.5 - 2} = 0.5$$

$$\frac{2.5 - z2}{0.5} = 0.5$$

$$2.5 - z2 = 0.5 \times 0.5$$

$$2.5 - z2 = 0.25$$

$$-z2 = 0.25 - 2.5$$

$$-z2 = -2.25$$

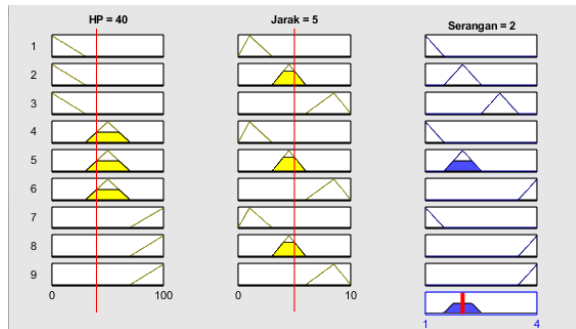
$$z2 = 2.25$$

Jadi, potongan fuzzy-nya berada pada trapesium dengan area di antara 1.75 dan 2.25, tinggi 0.5, dan puncaknya di 2.

Langkah terakhir ialah menghitung *centroid* dari trapesium dengan rumus :

$$z = \frac{z1 + z2}{2} = \frac{1.75 + 2.25}{2} = \frac{4}{2} = 2$$

Jadi, *output* defuzzifikasinya ialah 2 yang sesuai dengan label *skill*. Pada pengujian matlab juga menghasilkan *output* yang sama yaitu 2.

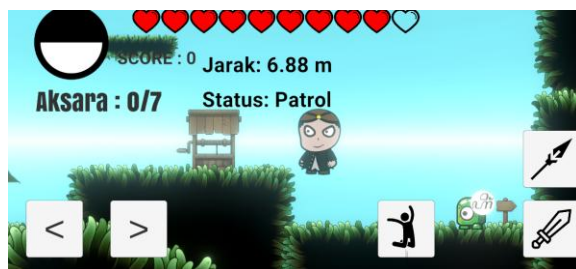


Gambar 7. Hasil Perhitungan Matlab

4. HASIL DAN PEMBAHASAN

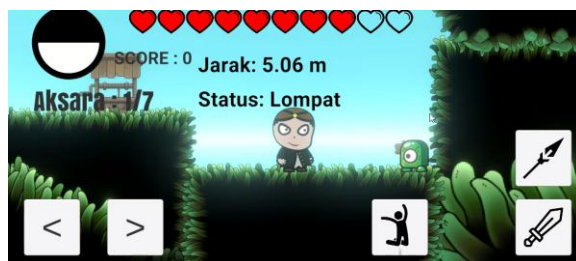
4.1. Pengujian FSM

Pada pengujian FSM dilakukan pada salah satu musuh yaitu charger yang akan mengejar player ketika player berada dalam jarak <5 disertai dengan peningkatan *movement*. Gambar 9 merupakan state pada saat patrol.



Gambar 8. Kondisi Patrol

Ketika player berada pada jarak >5 maka charger akan melompat, penanda bahwa dia akan segera memasuki state kejar seperti pada gambar 10.



Gambar 9. State Lompat

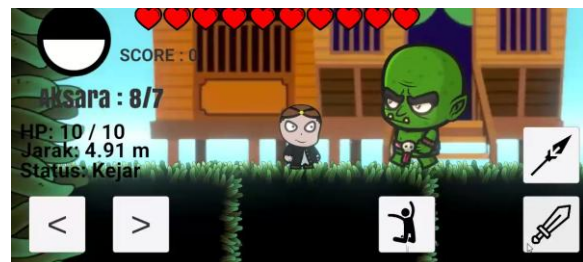
Setelah memasuki state lompat, *crawler* akan memasuki state untuk mengejar dan meningkatkan kecepatannya.



Gambar 10. State Kejar

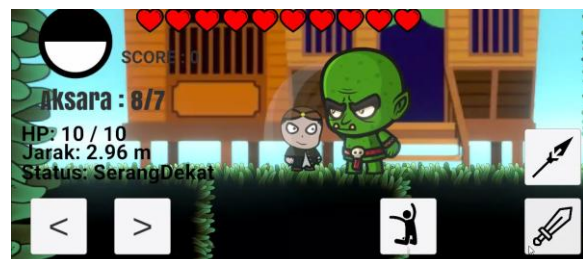
4.2. Pengujian Fuzzy

Pada boss terdapat 4 state yaitu kejar, pukul, skill dan lempar. Gambar 11 menunjukkan ketika dalam jarak 5.64 dan HP masih full yaitu 100%, maka boss akan memasuki state kejar.



Gambar 11. State Kejar Boss

Pada gambar 12 merupakan state dimana boss memasuki state serangan pertama dengan jarak yang kurang dari 3 dan *healthpoint full*.



Gambar 12. State Serang Dekat Boss

Berikutnya pada gambar 13 merupakan implementasi saat boss menggunakan skill dimana inputnya ialah *healthpoint* 60% dan jarak 5.64.



Gambar 13. State Skill Boss

State terakhir merupakan state saat boss mengeluarkan serangan melempar ketika *healthpoint* mencapai kurang dari 30% dan jaraknya lebih dari 6.



Gambar 14. State Lempar Boss

Pada tabel 4 terdapat hasil dari pengujian pada AI yang dilakukan.

Tabel 4. Pengujian AI

No.	Tujuan	Hasil Sistem	Hasil
1	musuh mengejar pemain ketika player masuk jarak patrolnya	Musuh bergerak kearah player	Sukses
2	Enemy dalam status idle atau patrol ketika player keluar dari jarak patrolnya	Enemy berhenti bergerak mengikuti player	Sukses
3	Player terkena damage ketika collider player dan enemy bertabrakan	HP Player berkurang	Sukses
4	Boss menyerang jarak dekat ketika jarak Player dengan enemy < 3	Boss Menyerang jarak dekat	Sukses
5	Boss menyerang jarak menengah ketika jarak Player dengan enemy > 3 dan < 6 dengan hp boss < 70%	Boss melakukan serangan dash kearah player	Sukses
6	Boss menyerang jarak jauh ketika jarak Player dengan enemy < 6 dan < 10 dengan hp boss < 30%	Boss melempar projektil kearah player	Sukses
7	Health enemy = 0, maka enemy mati	Enemy mati	Diterima

Dari hasil pengujian diatas didapat metode fuzzy dan fsm berjalan dengan semestinya yang dimana maupun AI dari NPC dan boss akan memiliki output seperti yang diharapkan ketika mencapai input yang sesuai dengan kondisinya. Seperti pada pengujian pertama dimana enemy akan mengejar player ketika player berada di jarak kejarnya. Kemudian pengujian kedua musuh akan kembali ke state awal yaitu idle/patrol ketika player telah keluar dari jarak patrolnya.

Nomor 3 merupakan pengujian pengurangan healthpoint player apabila collider antara player dan enemy bertabrakan. Pada pengujian nomor 4 hingga 6

merupakan pengujian sesuai dengan output fuzzy yang merupakan tipe serangan dari boss. Dan didapat hasilnya semua tipe serangan akan memberi output yang sesuai dengan inputan yang ada. Kemudian pengujian terakhir merupakan pengujian apabila healthpoint musuh habis, maka objek musuh akan di destroy yang dimana memiliki hasil yang sesuai. Dari 7 pengujian pada AI Musuh, didapat 7 yang berhasil dan dimana dapat disimpulkan bahwa persentase keberhasilannya ialah 100%.

4.3. Pengujian Kontrol Player

Merupakan pengujian button kontrol pada player yang berfungsi apakah seluruh button berfungsi dengan semestinya. Pada game jawa warrior terdapat 7 button yang dapat digunakan oleh playernya dengan keterangan dan detail button seperti pada tabel 5

Tabel 5. Pengujian Kontrol Player

No	Tombol	Hasil Sistem	Hasil
1	Arah kiri	SiJawa bergerak menuju kiri	Sukses
2	Arah kanan	SiJawa bergerak menuju kanan	Sukses
3	Attack Pedang	SiJawa menyerang jarak dekat	Sukses
4	Attack Tombak	SiJawa menyerang jarak menengah	Sukses
5	Attack Spell	SiJawa menyerang jarak jauh	Sukses
6	Heal	SiJawa memulihkan healthpoint	Sukses
7	Jump	SiJawa melompat	Sukses

Semua button bekerja dengan semestinya mulai dari button untuk movement yang terdiri dari arah kiri untuk bergerak dengan posisi -x, button arah kanan untuk bergerak dengan posisi x. sedangkan untuk attack pedang merupakan serangan jarak dekat, dan attack tombak merupakan serangan jarak menengah, kemudian attack spell merupakan tipe serangan jarak jauh.

Pada 2 tombol lainnya juga berjalan sesuai dengan semestinya yang dimana ketika menekan tombol jump maka player akan bergerak kearah y dan ketika player menekan tombol heal pada saat healthpointnya berkurang maka player akan menuju state healing untuk memulihkan healthpointnya ketika mana mencukupi. Dari ke-7 button yang diuji, ketujuh button bekerja sesuai fungsinya yang dimana dapat disimpulkan tingkat keberhasilan pada pengujian kontrol player adalah 100%.

4.4. Pengujian Perangkat

Adapun pengujian perangkat yang dilakukan untuk menguji apakah aplikasi berjalan dengan lancar pada berbagai macam tipe handphone. Hasil dari pengujian perangkat dapat dilihat pada tabel 6.

Tabel 6. Pengujian Perangkat

No	Nama Device	RAM	Versi Android	Hasil
1	Oppo A57	3 GB	6	Lag
2	Oppo F7	4 GB	8	Sukses
3	Vivo 1820	2 GB	8	Lag
4	Samsung S9+	4 GB	9	Sukses
5	Realme 5	4GB	10	Sukses
6	Poco M3	4 GB	11	Sukses
7	Redmi Note 8 Pro	6 GB	12	Sukses
8	Samsung S23	8 GB	13	Sukses
9	Poco X3 Pro	8GB	11	Sukses

Pada pengujian device ini, dari 9 device yang diuji terdapat 7 device yang dapat memainkan game tanpa lag dan 2 diantaranya yang dimana memiliki ram dibawah 4 GB mengalami lag sehingga dapat disimpulkan persentase keberhasilan dari pengujian device ini ialah 77.78%..

4.5. Pengujian User

Tabel 7. Pengujian User

No	Pernyataan	Jawaban		
		Baik	Cukup	Kurang
1	Penjelasan how to play mudah dimengerti	7	12	1
2	Game berjalan dengan lancar	11	7	2
3	Desain karakter dalam game	2	16	2
4	Animasi dan efek pada game	10	8	2
5	Kecocokan suara dengan game	12	5	3
6	Gameplay yang menarik	6	13	1
7	Kemudahan kontrol pada player	9	10	1
8	Apakah tingkat kesulitan bertambah disetiap level	4	13	3
9	Interaksi tokoh dalam game	3	8	9
Total		64	102	24

Jumlah Responden : 20

Jumlah Pertanyaan : 9

Faktor Pembagi : $20 \times 9 = 180$

Responden memilih baik $(64/180 \times 100) = 35.55\%$

Responden memilih cukup $(102/180 \times 100) = 56.67\%$

Responden memilih kurang $(24/180 \times 100) = 13.33\%$

Berdasarkan hasil evaluasi dari 20 responden terhadap 9 pertanyaan, diperoleh total 160 jawaban. Dari jumlah tersebut, sebanyak 35.55% responden memberikan penilaian Baik, 56.67% memberikan penilaian Cukup, dan hanya 13.33% yang memberikan penilaian Kurang. Persentase tertinggi pada kategori Cukup menunjukkan berdasarkan hasil pengujian user, game yang dirancang sudah cukup baik.

5. KESIMPULAN DAN SARAN

Kesimpulan penelitian ini menunjukkan bahwa game 2D Action berhasil dikembangkan menggunakan Unity dengan penerapan metode Fuzzy dan FSM sebagai sistem kecerdasan buatan musuh. Implementasi FSM mampu mengatur transisi antar state seperti idle, chase, dan attack berdasarkan input jarak antara pemain dan musuh. Selain itu, AI berbasis fuzzy pada karakter boss juga berhasil diimplementasikan, memungkinkan boss melakukan tiga jenis serangan yang ditentukan oleh health point dan jarak terhadap pemain. Dari sisi gameplay, pemain memiliki tiga jenis serangan—jarak jauh, menengah, dan dekat—serta dilengkapi skill heal untuk memulihkan health point. Pengujian AI musuh menunjukkan tingkat keberhasilan 100% dari 7 kali pengujian, di mana AI berperilaku sesuai logika yang diinginkan. Pada pengujian kontrol, seluruh tujuh tombol berfungsi dengan baik dengan tingkat keberhasilan 100%. Uji kompatibilitas pada sembilan perangkat menunjukkan bahwa tujuh perangkat dapat menjalankan game dengan lancar, sedangkan dua perangkat mengalami lag, menghasilkan tingkat keberhasilan sebesar 77,78%. Berdasarkan penilaian pengguna dengan kategori baik, cukup, dan kurang, mayoritas responden memilih kategori cukup sebesar 56,67%, yang menandakan bahwa game yang dirancang sudah cukup baik secara keseluruhan.

DAFTAR PUSTAKA

- [1] Bayu Aji Wicaksono, Ali Mahmudi, Karina Auliasari, "Perancangan Game Alien Warfare 2d Menggunakan Metode Finite State Machine (FSM) Berbasis Android," JATI (Jurnal Mhs. Tek. Inform., Vol. 7, No. 5, Pp. 2945–2951, 2023, Doi: 10.36040/Jati.V7i5.7555.
- [2] Puja Angga Firmansyah, Karina Auliasari, Yosep Agus Pranoto, " Penerapan Metode Fuzzy Logic Pada Game 3d "Aliester" JATI (Jurnal Mhs. Tek. Inform., Vol. 7, No. 5, Pp. 3098 – 3105, 2023, Doi: 10.36040/Jati.V7i5.7607.
- [3] Rizki Aditya Prasetyo, Agung Panji Sasmito, Karina Auliasari, "Perancangan Game 2D "Barbossa The Savior" Menggunakan Finite State Machine" JATI (Jurnal Mhs. Tek. Inform., Vol. 7, No. 5, Pp. 2999–3006, 2024, Doi: 10.36040/Jati.V7i5.7584.
- [4] Yosep Agus Pranoto, Hani Zulfia Zahro, Suryo Adi Wibowo, " Implementasi Fuzzy Logic Sebagai Penentu Jumlah Konsumsi Kalori Penderita Diabetes Melitus" Jurnal Industri inovatif 7 (2). ISSN 2087-8869, 2017
- [5] Maldi Ramadhan, Renaldi Primaswara Prasetya, Hani Zulfia Zahro, "Rancang Bangun Game Lost In The Zombie Apocalypse Menggunakan Metode Finite State Machine Berbasis Dekstop" JATI (Jurnal Mhs. Tek. Inform., Vol. 7, No. 4, Pp. 2264–2271, 2023, Doi: 10.36040/Jati.V7i4.7468.
- [6] Nur Rahmat Hidayat Karismadi, Syarifah

- Nurhafni A, "Implementasi Logika Fuzzy Mamdani Dalam Prediksi Curah Hujan Di Kota Kendari" JATI (Jurnal Mhs. Tek. Inform., Vol.8 No.1, 2024 doi:10.36040/jati.v8i1.8926
- [7] Febrina Ade Susianti, Anik Vega Vitianingsih, Achmad Choiron, Dwi Cahyono, Anggit Wikaningrum "Aplikasi Game Survival Horror 3d Pencegahan Penculikan Anak Remaja Berbasis Dekstop" Vol. 7 No. 2 2024
- [8] Alaa A. Qaffas "An Operational Study of Video Games' Genres" International Journal of Interactive Mobile Technologies (IJIM), vol. 14, no. 15, 2020 pp 175 - 194, doi: 10.3991/ijim.v14i15.16691
- [9] Mochamad Yusuf, Febriana Santi Wahyuni, Ahmad Fahrudi S, "The Meet" Game 3d First Person Berbasis Windows " JATI (Jurnal Mhs. Tek. Inform., Vol. 6, No. 1, Pp. 213–220, 2022, Doi: 10.36040/Jati.V6i1.4511.
- [10] Daniel Eka Wijaya, Karina Auliasari, Hani Zulfia Zahro, " The Method Combination Between Finite State Machine And Game-Based Learning On "Escape From Cov-Madness" Game" JATI (Jurnal Mhs. Tek. Inform., Vol. 5, No. 1, Pp. 86 – 93, 2021, Doi: 10.36040/jati.v5i1.3220
- [11] Fikriadi Sakrani, Suryo Adi Wibowo, Febriana Santi Wahyuni "Implementasi Finite State Machine Sebagai Kontrol Untuk Non Player Character Pada Game "Lastri And The Last Tree" JATI (Jurnal Mhs. Tek. Inform., Vol.4 No.2, pp 128 – 135, 2021 doi:10.36040/jati.v4i2.270
- [12] D. Yusuf and F. N. Afandi, "Aplikasi Absensi Berbasis Android Menggunakan Validasi Kordinat Lokasi Dan Nomor Handpone Guna Menghindari Penularan Virus Covid 19," Expert J. Manaj. Sist. Inf. dan Teknol., vol. 10, no. 1, pp. 16–22, 2020, doi: 10.36448/jmsit.v10i1.1492.
- [13] Russell L Laczniak, Doug Walker "Parental Restrictive Mediation and Children's Violent Video Game Play: The Effectiveness of the Entertainment Software Rating Board (ESRB) Rating System", Journal of Public Policy & Marketing., 2016, doi: 10.1509/jppm.15.071
- [14] Michael Hendra Nata, Joseph Dedy Irawan, Hani Zulfia Zahro, " Perancangan Game Android Bring Back The Scroll Menggunakan Metode FSM (Finite State Machine)" JATI (Jurnal Mhs. Tek. Inform., Vol. 8, No. 2, Pp. 1236–1244, 2024, Doi: 10.36040/Jati.V8i2.9086.
- [15] Bima Armedianto Putro, Karina Aulia Sari, Abdul Wahid, "Penerapan Metode Finite State Machine Pada Game "Escape From Punk Hazard". JATI (Jurnal Mhs. Tek. Inform., Vol. 5, No. 1, Pp. 71–78, 2022, Doi: 10.36040/Jati.V5i1.3276.
- [16] Muhammad Farrel Arrazzaq, Agung Panji Sasmito, Hani Zulfia Zahro, " Perancangan Game 2d Platformer "Adventure Quest" Dengan Metode Finite State Machine Berbasis Android" JATI (Jurnal Mhs. Tek. Inform., Vol. 7, No. 5, Pp. 2419 – 2427, 2023, Doi: 10.36040/Jati.V7i4.7537.
- [17] Zefid Frenda Fathoni, Moh Ahsan, Muhammad Priyono Tri Sulistyanto "Penerapan Fuzzy Sugeno Pada Npc Rintangan Dalam Game "Ana Cerita Extreme" Menggunakan Unity" BIMASAKTI Vol. 5 No. 2 2023, pp 25 – 31, doi : 10.21067/bimasakti.v5i2.8944