

Rancang Bangun Game Eternal Knight dengan Implementasi Finite State Machine untuk menciptakan Pola Perilaku Npc

Fahreza Azi Prayodha¹, Andi Iwan Nurhidayat²

Program Studi Manajemen Informatika, Universitas Negeri Surabaya

Jl. Ketintang, Kec. Gayungan, Kota Surabaya

¹fahreza.22042@mhs.unesa.ac.id

²andy134k5@unesa.ac.id

Abstrak - Industri game global terus mengalami pertumbuhan pesat baik dari segi teknologi maupun jumlah pemain, dengan nilai pasar mencapai lebih dari 184 miliar USD dan 3,3 miliar pemain di seluruh dunia. Salah satu elemen penting dalam pengembangan game adalah Non-Playable Character (NPC) yang meningkatkan interaksi dan tantangan permainan. Penelitian ini merancang dan membangun game Eternal Knight dengan implementasi Finite State Machine (FSM) untuk mengatur pola perilaku NPC menggunakan metodologi GDLC. Pengujian dilakukan melalui Black Box Testing dan Beta Testing. Hasil penelitian menunjukkan seluruh fitur dan transisi state NPC berjalan sesuai perancangan. Mayoritas responden menilai perilaku musuh konsisten, kontrol responsif, serta tingkat kepuasan pemain berada pada kategori baik
Kata kunci: Game Development, Finite State Machine, Non-Playable Character, GDLC, Black Box

Abstract - The global game industry continues to experience rapid growth in both technological advancement and number of players, with the market value exceeding 184 billion USD and reaching 3.3 billion players worldwide. One of the essential elements in game development is the Non-Playable Character (NPC), which enhances interaction and gameplay challenges. This research designs and develops the game Eternal Knight by implementing a Finite State Machine (FSM) to regulate NPC behavior patterns using the Game Development Life Cycle (GDLC) methodology. Testing was conducted through Black Box Testing and beta testing. The results indicate that all features and NPC state transitions functioned according to the design. The majority of respondents assessed enemy behavior as consistent, controls as responsive, and overall player satisfaction as good.

Keywords: Game Development, Finite State Machine, Non-Playable Character, GDLC, Black Box.

I. PENDAHULUAN

Perkembangan teknologi digital telah mendorong industri game mengalami pertumbuhan yang sangat pesat, khususnya pada pengembangan game berbasis interaksi dan kecerdasan buatan. Game tidak lagi hanya berfungsi sebagai media hiburan, tetapi juga menjadi sarana interaktif yang melibatkan strategi, pengambilan keputusan, serta kemampuan problem solving pemain dalam menyelesaikan tantangan di dalam permainan (Ghani et al., 2024). Salah

satu genre yang banyak diminati adalah game action-platformer 2D karena mampu menghadirkan kombinasi eksplorasi, pertarungan, dan tantangan mekanik permainan yang dinamis.

Pada game action-platformer, kualitas pengalaman bermain sangat dipengaruhi oleh perilaku musuh atau Non-Playable Character (NPC). NPC yang memiliki pola perilaku monoton dapat membuat permainan terasa repetitif dan kurang menantang bagi pemain. Oleh karena itu, pengembangan perilaku NPC yang dinamis menjadi salah satu aspek penting dalam meningkatkan kualitas gameplay (Faris et al., 2024). Selain memberikan tantangan, perilaku NPC juga dapat mendorong pemain untuk melakukan pattern recognition atau pengenalan pola terhadap pergerakan dan serangan musuh selama permainan berlangsung.

Pattern recognition merupakan bagian dari computational thinking yang menuntut pemain untuk mengidentifikasi dan memprediksi pola perilaku dalam lingkungan permainan. Kemampuan ini memungkinkan pemain menyusun strategi berdasarkan pola serangan, jeda animasi, maupun perubahan kondisi NPC yang diamati secara berulang (Agustini et al., 2024). Dengan demikian, implementasi pola perilaku NPC yang terstruktur tidak hanya meningkatkan tantangan permainan, tetapi juga memberikan pengalaman bermain yang lebih interaktif dan melatih kemampuan analitis pemain.

Dalam pengembangan perilaku NPC, penerapan Artificial Intelligence (AI) menjadi solusi yang umum digunakan untuk menciptakan respons karakter yang lebih realistis dan adaptif. AI pada game memungkinkan NPC merespons tindakan pemain secara otomatis sesuai kondisi permainan yang sedang berlangsung (Aksal et al., 2025). Salah satu metode AI yang banyak diterapkan dalam pengembangan game adalah Finite State Machine (FSM), karena metode ini mampu mengatur transisi perilaku NPC berdasarkan state tertentu secara terstruktur dan efisien.

Finite State Machine merupakan metode yang memodelkan perilaku sistem menggunakan kumpulan state, event, dan action sehingga NPC dapat berpindah dari satu kondisi ke kondisi lainnya berdasarkan input tertentu (Asrianda & Zulfadli, 2022). Penggunaan FSM pada game

dinilai efektif karena mampu meningkatkan responsivitas NPC tanpa membebani performa sistem secara berlebihan. Selain itu, implementasi FSM juga membuat perilaku NPC menjadi lebih alami dan interaktif sehingga pengalaman bermain menjadi lebih menarik (Nugraha et al., 2025).

Beberapa penelitian sebelumnya menunjukkan bahwa penerapan FSM mampu meningkatkan kualitas perilaku NPC dalam game. Penelitian oleh Naharu et al. (2021) menerapkan Hierarchical Finite State Machine untuk mengatasi kompleksitas perilaku NPC dalam jumlah besar pada game hack and slash. Hasil penelitian tersebut menunjukkan bahwa implementasi HFSM mampu menjaga performa permainan tetap stabil sekaligus memastikan transisi perilaku NPC berjalan sesuai rancangan. Penelitian lain yang dilakukan oleh Adeniyi et al. (2024) juga membuktikan bahwa penggunaan FSM pada game engine 2D berbasis JavaScript dan WebGL mampu mempertahankan frame rate tetap stabil dibandingkan pendekatan konvensional.

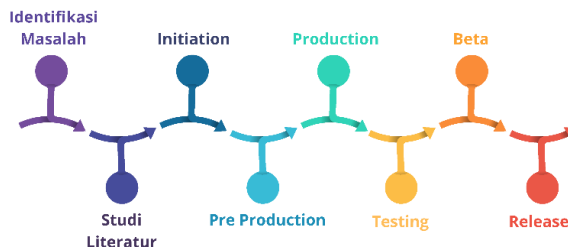
Selain itu, penelitian oleh Pratama et al. (2023) menunjukkan bahwa pengembangan game 2D menggunakan pendekatan terstruktur dapat menghasilkan gameplay yang berjalan dengan baik berdasarkan hasil pengujian pengguna. Sementara itu, penelitian Nguyen (2024) menjelaskan bahwa penggunaan Godot Engine dan GDScript mampu mempermudah implementasi mekanik permainan serta mendukung pengembangan game 2D secara fleksibel dan efisien. Godot sendiri merupakan game engine open source yang mendukung pengembangan lintas platform dan semakin populer di kalangan pengembang indie karena ringan serta mudah digunakan (Holfeld, 2024).

Dalam proses pengembangan aset visual game 2D, penggunaan software khusus pixel art juga menjadi faktor penting untuk mendukung kualitas tampilan permainan. Aseprite merupakan salah satu software yang dirancang khusus untuk pembuatan sprite dan animasi pixel art sehingga mempermudah pengembangan karakter serta animasi berbasis frame-by-frame pada game 2D (Aleefa, 2025). Penggunaan aset visual yang konsisten dapat mendukung pengalaman bermain menjadi lebih menarik dan imersif.

Berdasarkan permasalahan dan penelitian terdahulu tersebut, penelitian ini berfokus pada rancang bangun game action-platformer 2D berjudul *Eternal Knight* dengan implementasi Finite State Machine untuk menciptakan pola perilaku NPC yang dinamis dan terstruktur. Pengembangan game dilakukan menggunakan metode Game Development Life Cycle (GDLC) yang terdiri dari tahap initiation, pre-production, production, testing, beta, dan release sehingga proses pengembangan dapat berjalan lebih sistematis (Mustofa et al., 2021). Melalui penelitian ini diharapkan implementasi FSM mampu menghasilkan perilaku NPC yang konsisten, responsif, serta meningkatkan pengalaman bermain pemain dalam melakukan pattern recognition selama permainan berlangsung. Dokumen ini adalah *template, electronic copy* dapat diunduh dari *website* perpustakaan PNC. Isi pendahuluan sebaiknya mengandung latar belakang, tujuan, identifikasi masalah dan metoda penelitian, yang dipaparkan secara tersirat (*implicit*).

II. METODE PENELITIAN

Berikut merupakan tahapan penelitian dalam pengembangan game *Eternal Knight* Dengan Implementasi Finite State Machine Untuk Menciptakan Pola Perilaku NPC.



Gambar 1 Alur Penelitian

A. Identifikasi Masalah

Tahap pertama dalam penelitian ini adalah identifikasi masalah. Pada tahap ini peneliti menganalisis permasalahan yang sering muncul pada game action-platformer, khususnya terkait perilaku Non-Playable Character (NPC). Permasalahan utama yang ditemukan adalah perilaku NPC yang monoton, tidak responsif, serta rentan mengalami kesalahan transisi state ketika menghadapi aksi pemain yang tidak terduga. Kondisi tersebut dapat menyebabkan pengalaman bermain menjadi kurang menarik dan mengurangi kemampuan pemain dalam melakukan pattern recognition terhadap perilaku musuh. Oleh karena itu, penelitian ini berfokus pada implementasi Finite State Machine (FSM) untuk menciptakan pola perilaku NPC yang lebih dinamis, terstruktur, dan konsisten pada game *Eternal Knight*.

B. Studi Literatur

Tahap studi literatur dilakukan dengan mengumpulkan dan mempelajari berbagai referensi yang berkaitan dengan penelitian. Sumber literatur diperoleh melalui jurnal ilmiah, artikel penelitian, serta dokumentasi resmi yang membahas pengembangan game, Artificial Intelligence (AI), Finite State Machine (FSM), pattern recognition, Game Development Life Cycle (GDLC), serta penggunaan Godot Engine dan GDScript dalam pengembangan game 2D. Studi literatur bertujuan untuk memperkuat landasan teori sekaligus menjadi acuan dalam perancangan sistem perilaku NPC pada game *Eternal Knight*.

C. Initiation

Tahap initiation merupakan tahap awal dalam metode Game Development Life Cycle (GDLC). Pada tahap ini dilakukan perencanaan konsep dasar permainan yang akan dikembangkan. Peneliti menentukan genre game yaitu action-platformer 2D, target pemain, tujuan permainan, serta konsep utama gameplay. Selain itu, dilakukan juga analisis kebutuhan fungsional dan non-fungsional seperti mechanic karakter, perilaku NPC, antarmuka pengguna, desain level, dan kebutuhan aset visual maupun audio yang akan digunakan selama proses pengembangan game.

D. Pre Production

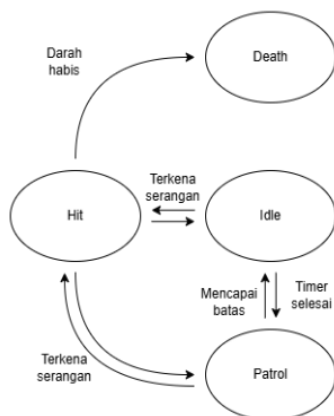
Tahap pre-production merupakan tahap perancangan konsep, struktur, serta kebutuhan teknis permainan sebelum proses implementasi dilakukan. Pada tahap ini peneliti menyusun berbagai rancangan yang akan digunakan sebagai pedoman utama dalam proses pengembangan game *Eternal Knight*. Seluruh perancangan dilakukan secara rinci dan terstruktur agar proses produksi dapat berjalan lebih terarah, sistematis, dan sesuai dengan tujuan penelitian yang telah ditentukan sebelumnya. Selain itu, tahap ini juga bertujuan untuk meminimalisir terjadinya kesalahan logika, bug, maupun ketidaksesuaian sistem selama proses pengembangan game berlangsung. Pada tahap pre-production ini dilakukan beberapa proses perancangan, meliputi perancangan konsep permainan, rancangan Finite State Machine (FSM), skenario game, rancangan user interface (UI), serta rancangan karakter yang akan digunakan di dalam permainan.

1) Konsep Game

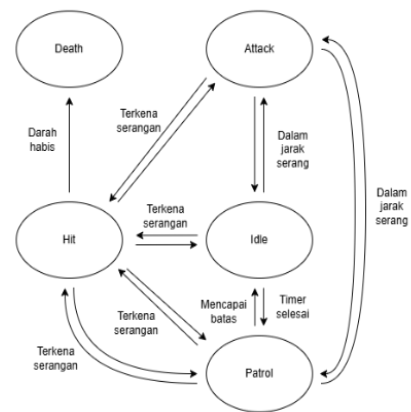
Game *Eternal Knight* mengusung genre action-platformer 2D di mana pemain diharuskan mengeksplorasi area permainan, melawan berbagai jenis musuh, serta menyelesaikan tantangan untuk melanjutkan permainan. Setiap NPC musuh memiliki pola perilaku yang berbeda sehingga pemain dituntut untuk melakukan pattern recognition terhadap pola serangan dan pergerakan musuh selama permainan berlangsung.

2) Rancangan Finite State Machine (FSM)

Pada tahap ini dilakukan perancangan Finite State Machine (FSM) sebagai sistem utama yang mengatur perilaku NPC di dalam game. FSM dirancang dengan menentukan state, event, dan transisi perilaku NPC seperti idle, patrol, chase, attack, hit, dan death. Setiap NPC memiliki rancangan FSM yang berbeda sesuai dengan karakteristik perilakunya sehingga musuh dapat bertindak secara dinamis dan responsif terhadap tindakan pemain.



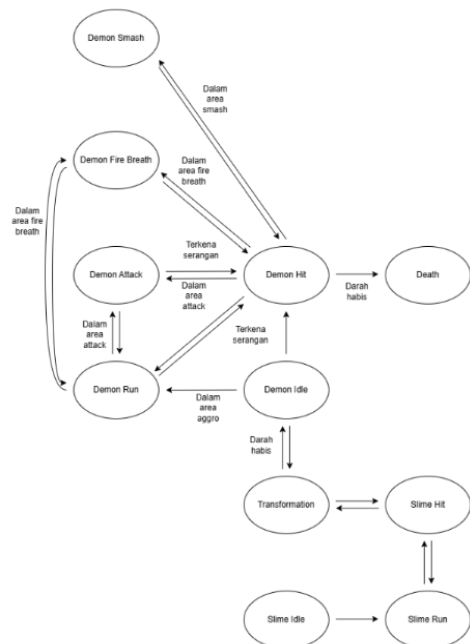
Gambar 2 FSM Skeleton



Gambar 3 FSM Octopus



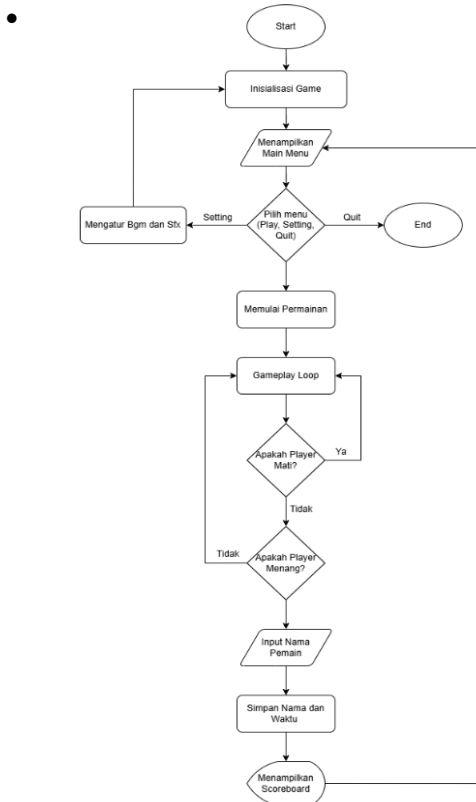
Gambar 4 FSM Goblin



Gambar 5 FSM Demon Slime

3) Skenario Game

Skenario game dirancang untuk menggambarkan alur utama permainan mulai dari pemain membuka game hingga menyelesaikan permainan. Pada tahap ini dirancang alur interaksi pemain dengan sistem seperti menu utama, eksplorasi level, pertarungan melawan musuh, sistem kemenangan dan kekalahan, hingga tampilan scoreboard. Perancangan skenario dilakukan agar alur permainan dapat berjalan secara runtut dan terstruktur.



Gambar 6 Skenario Game

4) Rancangan User Interface

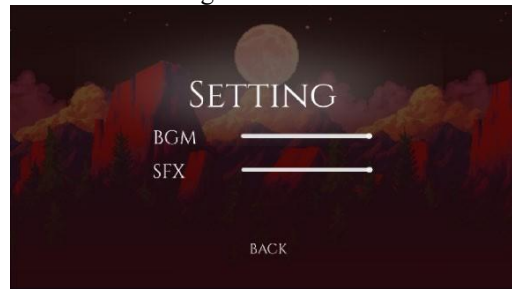
Rancangan user interface dilakukan untuk menentukan tampilan antarmuka yang digunakan dalam game *Eternal Knight*. Perancangan UI meliputi tampilan main menu, setting, scoreboard, dan elemen antarmuka lainnya yang mendukung interaksi pemain dengan sistem permainan. Desain antarmuka dibuat menggunakan wireframe untuk mempermudah proses implementasi pada tahap produksi.

- Wireframe Main Menu



Gambar 7 Wireframe Main Menu

Wireframe Setting



Gambar 8 Wireframe Setting

5) Rancangan Karakter

Pada tahap ini dilakukan perancangan karakter utama dan NPC yang digunakan dalam permainan. Seluruh karakter dirancang menggunakan gaya visual pixel art dengan software Aseprite. Karakter yang dirancang meliputi karakter utama *Doom Margrave* serta beberapa NPC musuh seperti *Skeleton*, *Octopus*, *Goblin*, dan *Demon Slime*. Setiap karakter memiliki desain visual dan animasi yang berbeda sesuai dengan perannya di dalam permainan.

- Doom Margrave



Gambar 9 Doom Margrave

- Skeleton



Gambar 10 Skeleton

- Octopus



Gambar 11 Octopus

- Goblin

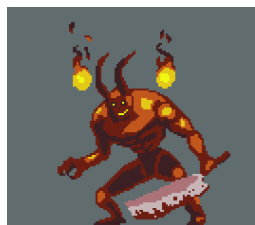


Gambar 12 Goblin

- Demon Slime



Gambar 13 Slime



Gambar 14 Demon

E. Production

Tahap production merupakan tahap implementasi seluruh rancangan yang telah dibuat pada tahap sebelumnya ke dalam game engine. Pada penelitian ini, pengembangan game dilakukan menggunakan Godot Engine dengan bahasa pemrograman GDScript. Proses produksi meliputi pembuatan level permainan, implementasi mechanic karakter utama, penerapan sistem Finite State Machine pada NPC, integrasi aset visual dan audio, serta pengembangan antarmuka pengguna. Seluruh fitur dan sistem dikembangkan secara bertahap dan dilakukan pengujian internal secara berkala untuk memastikan tidak terjadi kesalahan logika maupun bug selama proses implementasi.

F. Testing

Tahap testing dilakukan untuk menguji fungsionalitas sistem dan memastikan seluruh fitur game berjalan sesuai rancangan. Metode pengujian yang digunakan adalah Black Box Testing, yaitu pengujian yang berfokus pada fungsi sistem tanpa melihat struktur kode program secara langsung. Pengujian dilakukan pada mechanic karakter, perilaku NPC, sistem transisi state FSM, antarmuka pengguna, serta fitur permainan lainnya. Hasil pengujian digunakan untuk mengetahui apakah seluruh fungsi dalam game telah berjalan sesuai dengan tujuan penelitian dan

bebas dari error yang dapat mengganggu pengalaman bermain.

G. Beta

Tahap beta merupakan tahap pengujian game secara langsung oleh pengguna. Pada tahap ini game *Eternal Knight* didistribusikan kepada beberapa responden untuk dimainkan dan dievaluasi. Setelah memainkan game, responden diminta mengisi angket atau kuesioner mengenai pengalaman bermain, kenyamanan kontrol, perilaku NPC, tingkat kesulitan permainan, serta stabilitas sistem game. Data hasil beta testing digunakan sebagai bahan evaluasi untuk mengetahui tingkat keberhasilan implementasi FSM dalam menciptakan perilaku NPC yang konsisten dan interaktif.

H. Release

Tahap release merupakan tahap akhir dalam penelitian, yaitu proses penyelesaian dan publikasi game *Eternal Knight*. Setelah seluruh proses pengujian selesai dan perbaikan telah dilakukan berdasarkan hasil testing dan beta testing, game dinyatakan siap digunakan. Pada tahap ini dilakukan finalisasi fitur, optimalisasi performa, serta penyempurnaan sistem agar game dapat berjalan dengan baik. Hasil akhir penelitian berupa prototype game action-platformer 2D dengan implementasi Finite State Machine pada NPC yang mampu menciptakan perilaku musuh lebih dinamis dan responsif.

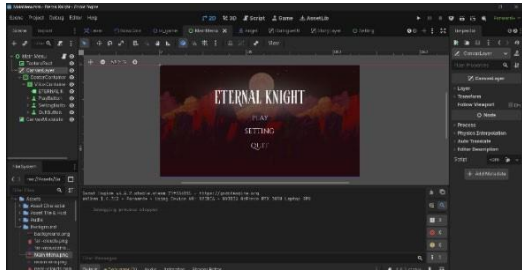
III. HASIL DAN PEMBAHASAN

Tahap hasil dan pembahasan merupakan tahap yang berisi implementasi seluruh rancangan yang telah dibuat pada tahap sebelumnya ke dalam game *Eternal Knight*. Pada tahap ini peneliti membahas hasil pengembangan game mulai dari implementasi antarmuka pengguna, implementasi karakter, hingga hasil pengujian sistem yang telah dilakukan. Selain itu, pada bagian ini juga dilakukan analisis terhadap keberhasilan implementasi sistem dalam menciptakan gameplay yang interaktif serta perilaku NPC yang berjalan sesuai dengan rancangan..

A. Implementasi User Interface(UI)

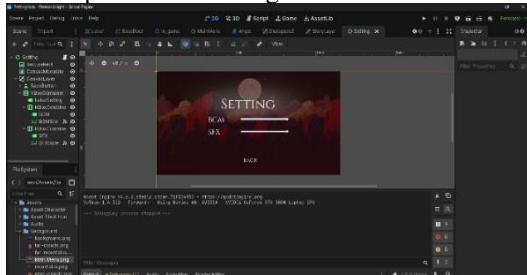
Pada tahap ini dilakukan implementasi rancangan antarmuka pengguna ke dalam game *Eternal Knight*. Antarmuka dirancang agar dapat mempermudah pemain dalam berinteraksi dengan sistem permainan serta memberikan pengalaman bermain yang lebih nyaman. Implementasi UI meliputi tampilan main menu, setting, HUD, dan scoreboard yang telah disesuaikan dengan konsep visual game.

- Implementasi Main Menu



Gambar 15 Implementasi Main Menu

- Implementasi Setting

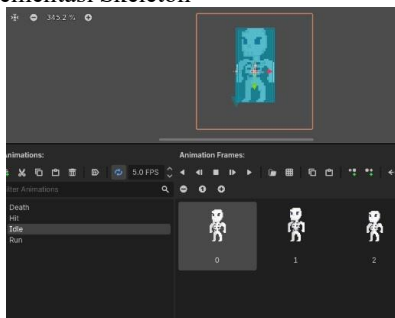


Gambar 16 Implementasi Setting

B. Implementasi Karakter

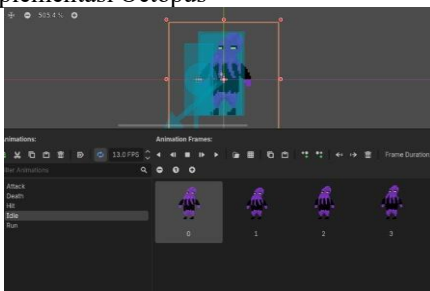
Tahap implementasi karakter dilakukan dengan mengintegrasikan seluruh aset karakter ke dalam game menggunakan gaya visual pixel art. Karakter yang diimplementasikan meliputi karakter utama dan beberapa NPC musuh yang memiliki desain serta animasi berbeda sesuai dengan perannya masing-masing di dalam permainan. Selain implementasi visual, pada tahap ini juga dilakukan penerapan animasi dan perilaku karakter agar dapat berjalan sesuai dengan mekanik permainan yang telah dirancang

- Implementasi Skeleton



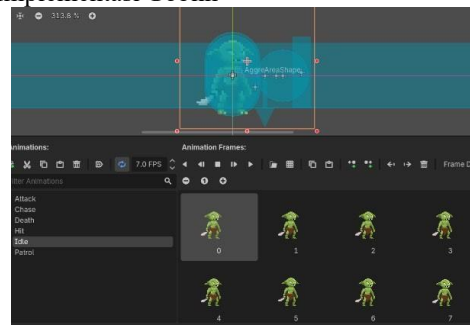
Gambar 17 Implementasi Skeleton

- Implementasi Octopus



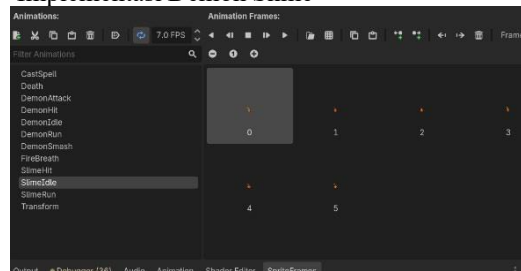
Gambar 18 Implementasi Octopus

- Implementasi Goblin

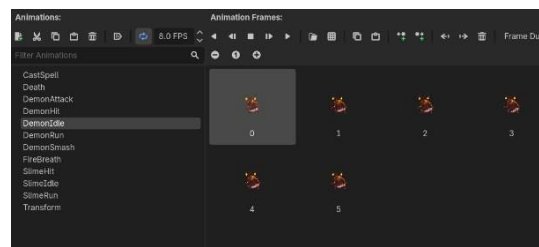


Gambar 19 Implementasi Goblin

- Implementasi Demon Slime



Gambar 20 Implementasi Slime



Gambar 21 Implementasi Demon

C. Hasil Pengujian

Tahap hasil pengujian dilakukan untuk mengetahui apakah seluruh fitur dan sistem pada game Eternal Knight telah berjalan sesuai dengan rancangan. Pengujian dilakukan menggunakan metode Black Box Testing dan Beta Testing untuk mengevaluasi fungsionalitas game, perilaku NPC, kenyamanan gameplay, serta pengalaman bermain pengguna. Hasil pengujian kemudian dianalisis untuk mengetahui tingkat keberhasilan implementasi sistem pada game.

- 1) Blacbox Testing

Tabel 1 Blackbox Testing Main Menu

No	Skema Pengujian	Hasil Yang Diharapkan	Hasil
1	Menekan tombol play	Beralih ke scene game	Sesuai
2	Menekan tombol setting	Beralih ke scene setting	Sesuai
3	Menekan tombol quit	Keluar dari permainan	Sesuai

Tabel 2 Blackbox Testing Setting

No	Skema Pengujian	Hasil Yang Diharapkan	Hasil
1	Mengecilkan atau membesarkan bgm	Bgm berhasil dikecilkan atau dibesarkan	Sesuai
2	Mengecilkan atau membesarkan sfx	Sfx berhasil dikecilkan atau dibesarkan	Sesuai
3	Menekan tombol back	Beralih ke scene	Sesuai

Tabel 3 Blackbox Testing Pause

No	Skema Pengujian	Hasil Yang Diharapkan	Hasil
1	Menekan tombol resume	Melanjutkan permainan	Sesuai
2	Menekan tombol restart	Memulai ulang permainan	Sesuai
3	Menekan tombol exit	Beralih ke scene main menu	Sesuai

Tabel 4 Blackbox Testing Ingame

No	Skema Pengujian	Hasil Yang Diharapkan	Hasil
1	Menekan tombol Esc	Bgm berhasil dikecilkan atau dibesarkan	Sesuai
2	Pemain mati	Sfx berhasil dikecilkan atau dibesarkan	Sesuai
3	Menekan tombol submit	Beralih ke scene	Sesuai
4	Pemain menang	Menampilkan victory panel	Sesuai
5	Menekan tombol submit	Menampilkan leaderboard	Sesuai
6	Menekan tombol back	Beralih ke scene main menu	Sesuai

Tabel 5 Blackbox Testing Karakter Utama

No	Skema Pengujian	Hasil Yang Diharapkan	Hasil
1	Menekan tombol panah	Karakter bergerak sesuai arah	Sesuai

2	Menekan tombol lompat	Karakter melompat	Sesuai
3	Menekan tombol J	Karakter menampilkan Attack	Sesuai
4	Menekan panah + J	Karakter Menampilkan Attack Movement	Sesuai
5	Terkena serangan musuh	Menjalankan animasi Hit dan mengurangi darah pemain	Sesuai
6	Darah habis	Menjalankan animasi Death dan bangkit dari titik awal atau checkpoint	Sesuai

Tabel 6 Blackbox Testing Skeleton

No	Skema Pengujian	Hasil Yang Diharapkan	Hasil
1	Diam di tempat	Mengeluarkan animasi Idle	Sesuai
2	Berpatroli	Skeleton berpatroli	Sesuai
3	Terkena serangan pemain	Darah berkurang dan menjalankan animasi Hit	Sesuai
4	Darah habis	Menjalankan animasi Death dan menghilang	Sesuai

Tabel 7 Blackbox Testing Octopus

No	Skema Pengujian	Hasil Yang Diharapkan	Hasil
1	Diam di tempat	Mengeluarkan animasi Idle	Sesuai
2	Berpatroli	Octopus berpatroli	Sesuai
3	Pemain mendekati Octopus	Octopus menyerang pemain	Sesuai
4	Terkena serangan pemain	Darah berkurang dan menjalankan animasi Hit	Sesuai
5	Darah habis	Menjalankan animasi Death dan menghilang	Sesuai

Tabel 8 Blackbox Testing Goblin

No	Skema Pengujian	Hasil Yang Diharapkan	Hasil
1	Diam di tempat	Mengeluarkan animasi Idle	Sesuai
2	Berpatroli	Goblin berpatroli	Sesuai
3	Pemain memasuki AggroArea	Goblin mengejar pemain	Sesuai
4	Pemain mendekati Goblin	Goblin menyerang pemain	Sesuai
5	Terkena serangan pemain	Darah berkurang dan menjalankan animasi Hit	Sesuai
6	Darah habis	Menjalankan animasi Death dan menghilang	Sesuai

Tabel 9 Blackbox Testing Demon Slime

No	Skema Pengujian	Hasil Yang Diharapkan	Hasil
1	Diam di tempat	Menjalankan animasi SlimeIdle	Sesuai
2	Pemain mendekati Slime	Slime menjauh	Sesuai
3	Terkena serangan pemain	Menjalankan animasi SlimeHit	Sesuai
4	Darah slime habis	Transformasi	Sesuai
5	Pemain menjauh dari Demon	Menjalankan animasi DemonIdle	Sesuai
6	Pemain mendekat	Demon mengejar pemain	Sesuai
7	Pemain memasuki MeleeTrigger	Menjalankan animasi DemonAttack	
8	Pemain memasuki FireBreathTrigger	Menjalankan animasi DemonFireBreath	
9	Pemain memasuki SmashTrigger	Menjalankan animasi DemonSmash	
10	Pemain menyerang Demon	Menjalankan animasi DemonHit	

11	Darah habis	Menjalankan animasi Death dan menghilang	
----	-------------	--	--

2) Beta Testing

Tabel 10 Instrumen Beta Testing

No	Pertanyaan	Jenis Skala
1	Apakah Anda mengalami kebingungan saat awal bermain?	Skala 1 dan 2
2	Seberapa mudah anda memahami kontrol?	Likert 1-5
3	Bagaimana respon kontrol karakter?	Likert 1-5
4	Apakah perilaku musuh konsisten?	Likert 1-5
5	Menurut Anda, apakah pola perilaku musuh terasa menantang?	Likert 1-5
6	Apakah terdapat bug pada saat bermain?	Skala 1 dan 2
7	Seberapa sering bug mengganggu permainan?	Likert 1-5
8	Tingkat kesulitan game secara keseluruhan?	Likert 1-5
9	Seberapa puas Anda dengan game Eternal Knight?	Likert 1-5
10	Apakah Anda tertarik memainkan versi final game ini?	Skala 1-3

Tabel 11 Hasil Pengujian Skala 1-3

Skala	Keterangan	Responden	Persentase
1	Ya	8	66,7%
2	Mungkin	4	33,3%
3	Tidak	0	0%

Tabel 12 Hasil Pengujian Skala 1 dan 2

Pertanyaan	Jawaban Dominan	Persentase
Apakah Anda mengalami kebingungan saat awal bermain?	Ya	66,7%
Apakah terdapat bug pada saat bermain?	Ya	66,7%

Tabel 13 Hasil Pengujian Skala Likert

No	Pertanyaan	Rata-rata skor	Kategori
1.	Seberapa mudah anda memahami kontrol?	3,58	Baik
2.	Bagaimana respon kontrol karakter?	3,33	Cukup
3.	Apakah perilaku musuh konsisten?	3,83	Baik
4.	Menurut Anda, apakah pola perilaku musuh terasa menantang?	2,00	Kurang
5.	Seberapa sering bug mengganggu permainan?	4,00	Baik
6.	Tingkat kesulitan game secara keseluruhan?	2,08	Mudah
7.	Seberapa puas Anda dengan game Eternal Knight?	3,75	Baik

Peneliti berhasil mengimplementasikan FSM pada NPC dan memperbaiki bug yang ada pada saat Beta testing secara menyeluruh.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil yang diperoleh dalam penelitian ini, kesimpulan yang dapat diambil adalah sebagai berikut:

- 1) Berdasarkan hasil perancangan, implementasi, serta pengujian yang telah dilakukan pada pengembangan game Eternal Knight, dapat disimpulkan bahwa model Finite State Machine atau fsm berhasil dirancang untuk menciptakan pola perilaku NPC yang adaptif. Model FSM mampu mengatur perubahan perilaku NPC melalui beberapa kondisi utama seperti patrol, chase, attack, hit, dan death, sehingga NPC dapat merespons keberadaan pemain maupun kondisi lingkungan permainan secara otomatis.
- 2) Implementasi algoritma Finite State Machine ke dalam game engine menggunakan Godot Engine juga telah berjalan dengan baik, di mana setiap NPC memiliki mekanisme transisi state yang

terstruktur dan sesuai dengan rancangan gameplay yang telah dibuat. Berdasarkan hasil pengujian Blackbox Testing, seluruh fungsi permainan dapat berjalan sesuai dengan yang diharapkan, sedangkan hasil Beta Testing menunjukkan bahwa sistem kontrol, perilaku musuh, serta pengalaman bermain secara keseluruhan dapat diterima dengan baik oleh pemain. Dengan demikian, penerapan metode Finite State Machine terbukti efektif dalam mengendalikan perilaku NPC dan mendukung terciptanya gameplay yang lebih interaktif pada game Eternal Knight.

B. Saran

Meskipun implementasi Finite State Machine pada proyek ini sudah berjalan dengan baik, terdapat beberapa saran untuk pengembangan lebih lanjut guna meningkatkan kecerdasan buatan dalam game. Peneliti menyarankan adanya integrasi sistem Pathfinding (seperti A* atau Navigation2D) untuk mendukung kondisi chase yang lebih kompleks, terutama pada tata letak level yang memiliki banyak rintangan. Selain itu, penambahan variasi kondisi pada FSM, seperti mekanisme retreat atau mundur saat darah NPC rendah atau sistem koordinasi antar musuh, dapat dilakukan untuk menciptakan tantangan yang lebih tinggi bagi pemain. Terakhir, pengujian lebih mendalam mengenai pengaruh jumlah NPC yang menggunakan FSM terhadap performa frame rate pada perangkat dengan spesifikasi rendah perlu dilakukan untuk memastikan optimasi kode tetap terjaga.

REFERENSI

- [1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
- [2] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
- [3] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.
- [4] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.
- [5] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.
- [6] (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [7] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/>
- [8] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.
- [9] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
- [10] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
- [11] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
- [12] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.