

Pengembangan *Game Escape From Basement* dengan AI Berbasis *Finite State Machine* di Godot Engine Menggunakan Metode GDLC

Bagus Tri Sasognko¹

¹Universitas Muhammadiyah Surabaya

E-mail: bagus.tri.sasongko-2022@ft.um-surabaya.ac.id¹

Abstrak. *Escape From Basement* adalah *game survival horror* dengan AI berbasis *Finite State Machine* (FSM) dan *Waypoint Navigation* untuk mengontrol perilaku serta pergerakan karakter AI. Penelitian ini bertujuan mengembangkan AI adaptif dan mengevaluasi efektivitasnya dalam meningkatkan tantangan dan imersi pemain. Metode yang digunakan adalah *Research and Development* (R&D) dengan pendekatan *Game Development Life Cycle* (GDLC), mencakup *initiation*, *production*, *testing*, dan *release*. FSM mengatur perilaku AI, sementara *Waypoint Navigation* memastikan pergerakan yang dinamis. Hasil penelitian menunjukkan kombinasi FSM dan *Waypoint Navigation* menciptakan AI yang responsif, meningkatkan tingkat kesulitan, serta keterlibatan pemain. Penelitian ini menjadi referensi bagi pengembang *game indie* dalam mengimplementasikan AI berbasis FSM di Godot Engine. Studi selanjutnya dapat mengeksplorasi *Behavior Trees*, pembelajaran mesin, atau fitur *multiplayer* untuk pengalaman yang lebih dinamis.

Kata kunci: GDLC; FSM; *Waypoint Navigation*; AI Monster; Godot Engine.

Abstract. *Escape From Basement* is a survival horror game with AI using Finite State Machine (FSM) and Waypoint Navigation to control character behavior and movement. This study aims to develop adaptive AI and evaluate its effectiveness in enhancing challenge and immersion. The method used is Research and Development (R&D) with the Game Development Life Cycle (GDLC) approach, including initiation, production, testing, and release. FSM manages AI behavior, while Waypoint Navigation ensures dynamic movement. Results show that combining FSM and Waypoint Navigation creates responsive AI, increases difficulty, and enhances player engagement. This research serves as a reference for indie developers using FSM-based AI in Godot Engine. Future studies may explore Behavior Trees, machine learning, or multiplayer features for a more dynamic experience.

Keywords: GDLC; FSM; *Waypoint Navigation*; AI Monster; Godot Engine.

1. Pendahuluan

Perkembangan teknologi saat ini semakin pesat[1], salah satu perkembangan teknologi yang sangat pesat dan banyak diminati oleh masyarakat adalah proses pengembangan *game*[2]. Terutama dalam genre *survival horror* yang menekankan pengalaman bermain yang imersif dan menegangkan[1]. Banyak masyarakat yang menghabiskan waktunya hanya untuk bermain *game* dibanding dengan aktivitas yang lainnya dimana menurut laporan dari *Decesion Lab* dan *Mobile Marketing Association* (MMA) menyebutkan jumlah *gamer* di Indonesia mencapai 60 juta yang di perkirakan akan mencapai

angka 100 juta pada tahun 2020[3]. Salah satu elemen kunci dalam genre ini adalah kecerdasan buatan *Artificial Intelligence* (AI) pada musuh atau monster yang adaptif, mampu memberikan tantangan nyata bagi pemain[1]. Namun, beberapa *developer game* menghadapi kendala penerapan dalam mengembangkan AI yang responsif dan realistis, khususnya dalam aspek pergerakan dan perilaku musuh[4].

Finite State Machine (FSM) dan navigasi *waypoint* adalah dua metode yang umum digunakan untuk mengatasi masalah ini. FSM memungkinkan AI untuk berpindah antar berbagai status, seperti diam(*idle*), patroli, mengejar, dan menyerang, berdasarkan kondisi tertentu[3]. Penerapan metode *Finite State Machine* pada *game* di terapkan untuk menentukan berbagai macam respon *Non Player Character* berdasarkan interaksi yang dilakukan oleh pemain[5]. *Navigasi waypoint*, di sisi lain, membantu AI dalam menentukan jalur pergerakan yang efisien dalam lingkungan permainan. Teknik ini menggunakan informasi berupa kumpulan koordinat gerak dari satu posisi ke posisi lain dengan mengasumsikan setiap posisi dalam proses pergerakannya menjadi suatu titik dalam sistem koordinat tertentu misalnya menunjukkan garis lintang, bujur, kecepatan, dan sudut rute[6]. Implementasi kedua metode ini dapat meningkatkan realisme dan dinamika perilaku musuh dalam *game*.

Metode FSM dalam permainan petualangan "*Trapped Miners*" memungkinkan karakter *NPC* untuk mendeteksi keberadaan pemain dan menentukan tindakan serangan musuh tanpa memerlukan intervensi dari pemain. Hasil pengujian ini mencapai tingkat keberhasilan sebesar 100%[7]. Pada *game Lastri And The Last Tree* adalah *Finite State Machine* (FSM) yang berfungsi untuk mengontrol *NPC* (*Non Playable Character*) yaitu karakter yang digunakan sebagai musuh dan digerakkan oleh kecerdasan buatan[8]. Meningkatkan kinerja teknis dan pengalaman pemain secara keseluruhan [9]. FSM menghubungkan keadaan-keadaan ini melalui transisi, memandu perilaku karakter berdasarkan masukan pemain dan aturan permainan, dan secara efektif berfungsi sebagai peta jalan bagi tindakan karakter [10].

Dalam konteks pengembangan *game indie*, keterbatasan sumber daya dan teknologi sering menjadi hambatan dalam menciptakan AI yang kompleks untuk mencapai kepentingan tersebut memerlukan tim pengembang yang besar dengan pengorbanan yang juga besar dalam segi tenaga dan waktu[11]. Penggunaan AI dalam video *game* sering dijumpai di berbagai jenis *game* seperti *game* menembak, strategi, dan balapan. Fitur AI biasanya digunakan untuk mengontrol karakter non-pemain. Salah satu keahlian kecerdasan buatan yang digunakan yaitu algoritma *Finite-state machine* (FSM) [12].

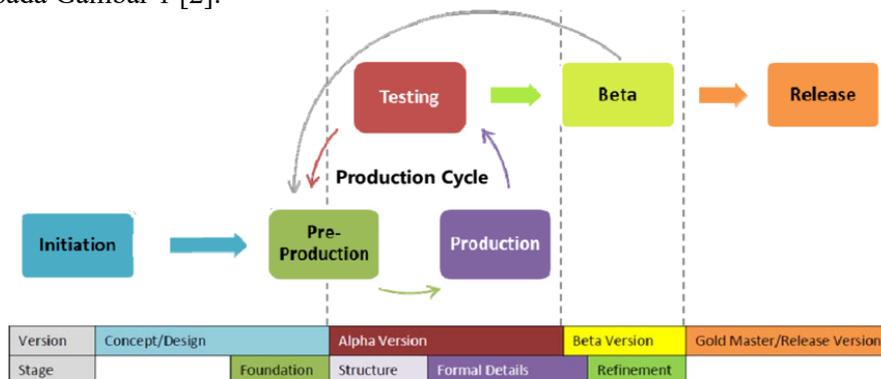
Godot Engine, sebagai salah satu *game engine* yang populer di kalangan pengembang *indie*, menyediakan fitur untuk implementasi FSM dan navigasi *waypoint*. Namun, pemanfaatannya dalam genre *survival horror* masih belum optimal. Penelitian sebelumnya telah membahas penerapan FSM dalam pengembangan perilaku karakter pemain menggunakan Godot Engine, namun fokus pada implementasi AI musuh dalam *game survival horror* masih terbatas. Godot Engine, yang dikenal dengan antarmuka yang ramah pengguna, dukungan lintas platform, pemrograman visual yang kuat, fleksibilitas grafis, mesin fisika yang handal, dan komunitas pengembang yang aktif[13].

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan *game "Escape from Basement 3D"*, sebuah *game survival* horor berbasis *puzzle* dengan implementasi AI monster menggunakan FSM dan navigasi *waypoint*. *Game* ini diharapkan tidak hanya menantang pemain dalam menyelesaikan teka-teki, tetapi juga menghadirkan pengalaman yang lebih menegangkan melalui AI monster yang adaptif. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan AI pada *game indie* serta menjadi referensi bagi pengembang *game* dalam implementasi FSM dan navigasi AI yang lebih efektif.

2. Metode

Penelitian ini menggunakan metode *Game Development Life Cycle* (GDLC), yang mengatur fase-fase penting seperti perencanaan, desain, pengembangan, pengujian, dan peluncuran[13]. *Game Development Life Cycle* merupakan pendekatan sistematis dalam pengembangan *game* yang menggambarkan tahapan utama dari konseptualisasi hingga peluncuran *game*[13]. GDLC dipilih dalam penelitian ini karena telah banyak digunakan oleh peneliti sebelumnya dan terbukti merupakan metode pengembangan *game* yang tepat jika diimplementasikan dalam desain *game*. Metode ini memiliki

langkah-langkah yang sederhana dan sesuai dengan kondisi dalam penelitian ini seperti yang ditunjukkan pada Gambar 1 [2].



Gambar 1. *Game Development Life Cycle (GLDC)*

Pengembangan *game* dimulai dari tahap pertama, yaitu inisiasi, di mana konsep awal *game* edukasi dikembangkan. Selanjutnya, pada tahap praproduksi, dilakukan perancangan *game* yang mencakup *gameplay* dan pembuatan mini prototipe. Tahap produksi berfokus pada pengembangan desain *game*, termasuk penyempurnaan *gameplay* serta pembuatan mini prototipe untuk menentukan mekanisme permainan yang akan diimplementasikan. Setelah itu, dilakukan tahap pengujian, di mana fitur-fitur *game* dievaluasi untuk memastikan kesesuaiannya dengan rancangan awal. Pada tahap pengujian beta, *game* diuji secara lebih luas dengan melibatkan pengguna untuk mendeteksi serta memperbaiki kesalahan atau *bug* yang mungkin masih ada. Terakhir, tahap rilis merupakan tahap akhir dalam pengembangan *game*, di mana aplikasi yang telah lolos pengujian beta akan diluncurkan dan dapat digunakan oleh pengguna. [14].

Metode GDLC yang dikemukakan oleh *Heather Chandler* dianggap sebagai pendekatan yang paling tepat dalam penelitian ini, karena langkah-langkahnya yang sederhana dan sesuai dengan kondisi serta tujuan penelitian[15]. Dalam perancangan video game terdapat beberapa GDLC yang cukup populer, diantaranya yang disusun oleh *Blitz Games Studios*, *Arnold Hendrick*, *Doppler Interactive*, *Penny de Byl* dan *Heather Chandler*. Setiap GDLC memiliki karakteristik yang berbeda sesuai dengan kebutuhan dan kondisi developer[16].

2.1. Initiation

Tahap *initiation* merupakan proses perancangan konsep awal [17], dari *game Escape From Basement* yang akan dikembangkan. Pada tahap ini, terdapat beberapa aspek penting yang perlu diperhatikan dalam pembuatan *game* ini, yaitu:

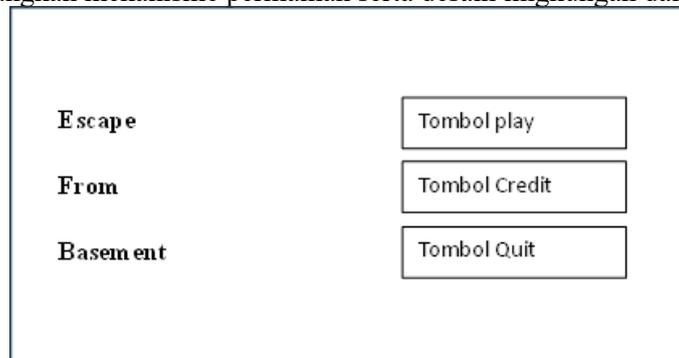
1. *Game* yang dikembangkan merupakan *game adventure horror* dengan visualisasi 3D.
2. *Game* ini berbasis *single-player* dan dapat dimainkan secara *offline*.
3. *Game* ini bertujuan untuk memberikan pengalaman bermain yang menegangkan dengan implementasi Monster AI berbasis *Finite State Machine (FSM)* guna meningkatkan tantangan dalam permainan.
4. *Game* ini dirancang untuk dimainkan pada perangkat desktop/laptop.
5. Pengembangan *game* menggunakan Godot Engine dengan metode *Game Development Life Cycle (GDLC)*.

2.2. Pre-Production

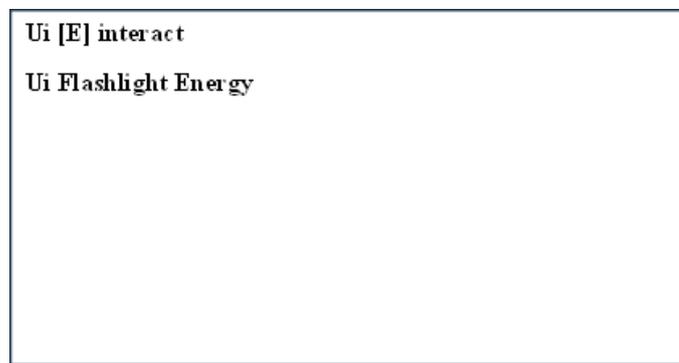
Tahap *pre-production* merupakan tahap perancangan dan perencanaan awal [17], dari *game Escape From Basement*. Pada tahap ini, dilakukan berbagai persiapan penting sebelum memasuki tahap pengembangan utama, termasuk pembuatan *storyboard*, *flowchart*, serta *prototyping*.

2.2.1 Game Design

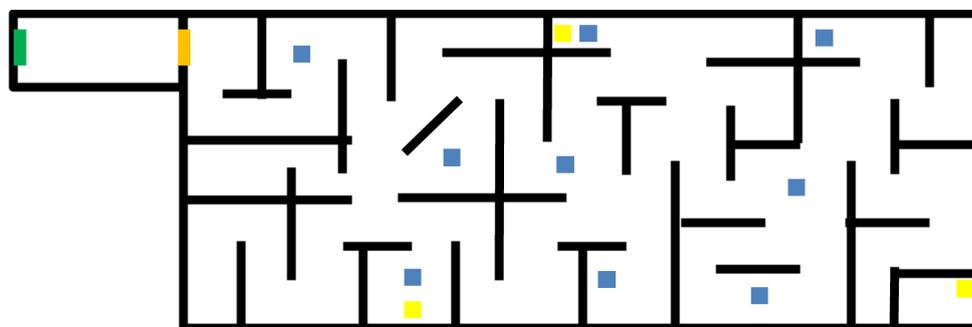
Dalam membangun *game Escape From Basement*, dilakukan perancangan *storyboard* dan desain antarmuka *game* untuk menggambarkan alur permainan secara visual. *Storyboard* ini digunakan sebagai acuan dalam mengembangkan mekanisme permainan serta desain lingkungan dalam *game*.



Gambar 2. Tampilan Main Menu



Gambar 3. Tampilan Saat *Gameplay*



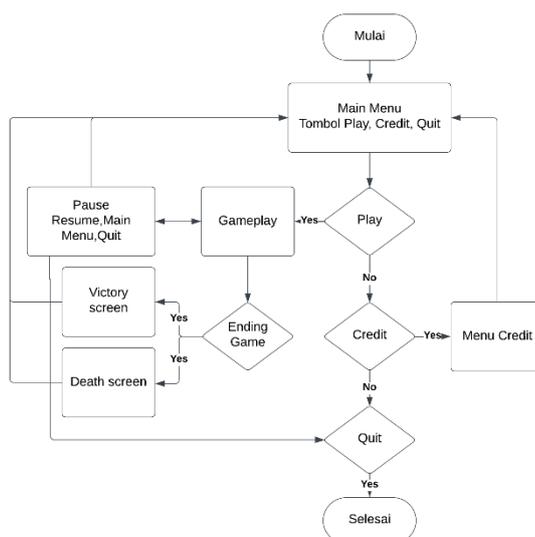
Gambar 4. Peta *Game Escape From Basement*

Table 1. Deskripsi Warna

Warna	Deskripsi
	Waypoint Navigasi titik – titik untuk monster berpatroli
	Pedang untuk membuka pintu besar untuk keluar <i>basement</i> , perlu mengumpulkan 3 item pedang
	Gerbang Kayu untuk ke arah cahaya putih
	Cahaya putih menggambarkan karakter sudah keluar untuk <i>game</i> selesai

2.2.2 Flowchart

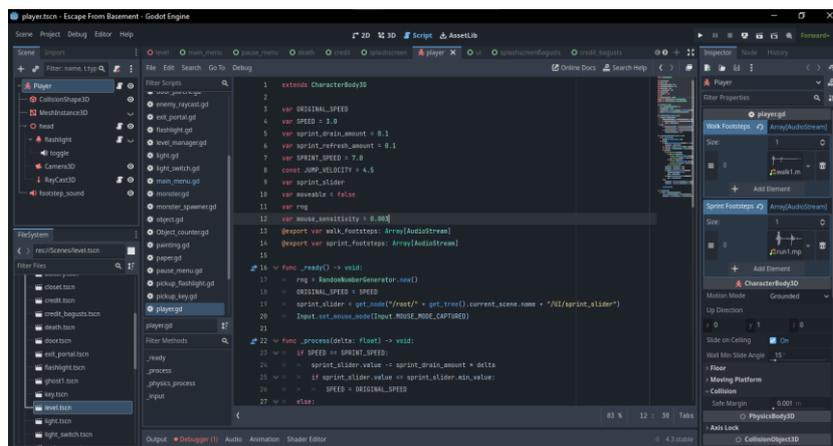
Pada tahap pembuatan *flowchart* menjelaskan langkah-langkah yang menggambarkan bagan alur atau urutan proses dari *game*[17].



Gambar 5. Flowchart Game

2.2.3 Production

Production merupakan tahap menyempurnakan hasil dari perancangan *game design* dan *prototype* yang telah dibuat. Pada proses produksi *game* ini terdiri dari pembuatan aset *game*, pemrograman, hingga mengintegrasikan hasil aset dan pemrograman pada *software Godot Engine*. Tahap produksi pada *game Escape From Basement* dipaparkan seperti pada gambar berikut.



Gambar 6. Proses pembuatan *game* menggunakan Godot Engine 4.3

2.2.4 Beta Testing

Tahap beta testing dilakukan dengan menguji *game Escape From Basement* secara langsung kepada pengguna akhir, yaitu teman-teman di kampus dan beberapa teman di luar kampus. Tujuan dari pengujian ini adalah untuk mengidentifikasi dan memperbaiki kekurangan yang mungkin masih terdapat dalam *game* berdasarkan umpan balik dari pemain.

Pengujian ini dilakukan menggunakan metode *Usability Testing* guna mengukur tingkat kemudahan pengguna dalam memainkan *game*. Metode *Usability Testing* dapat mencakup pengamatan terhadap interaksi pengguna dengan teknologi dan/atau pengguna yang menyelesaikan kuesioner setelah berinteraksi dengan teknologi [18]. Responden akan mencoba memainkan *gamennya* lalu mengisi kuesioner berdasarkan indikator pada *Usability Testing*, seperti:

1. *Learnability* – Kemudahan dalam memahami mekanisme permainan.
2. *Memorability* – Kemampuan pemain dalam mengingat cara bermain.
3. *Efficiency* – Efisiensi *gameplay* dan kompatibilitas pada perangkat.

4. *Errors* – Evaluasi terhadap kesalahan dalam *game*, termasuk respons tombol dan instruksi.
5. *Satisfaction* – Kepuasan pemain setelah memainkan *game*.

Tabel 2. Tabel Kuesioner *Usability Testing*

Indikator	Kode	Pertanyaan
<i>Learnability</i>	L1	Saya dapat memahami cara bermain <i>game</i> ini dengan mudah.
	L2	Instruksi dalam <i>game</i> jelas dan mudah diikuti.
	L3	Tujuan permainan mudah dipahami sejak awal.
<i>Memorability</i>	M1	Saya dapat mengingat cara bermain <i>game</i> ini dengan mudah setelah beberapa kali bermain.
	M2	Kontrol dalam <i>game</i> mudah diingat dan tidak membingungkan.
<i>Efficiency</i>	E1	<i>Game</i> berjalan dengan lancar tanpa gangguan teknis yang berarti.
	E2	Waktu <i>loading game</i> tidak terlalu lama.
	E3	<i>Game</i> ini dapat dimainkan dengan baik di perangkat yang saya gunakan.
<i>Errors</i>	ER1	Saya tidak menemukan banyak <i>bug</i> atau kesalahan saat bermain.
	ER2	Tombol dan kontrol dalam <i>game</i> berfungsi sesuai dengan harapan.
	ER3	Petunjuk dalam <i>game</i> sesuai dengan mekanisme permainan.
<i>Satisfaction</i>	S1	Saya merasa terhibur saat memainkan <i>game</i> ini.
	S2	Atmosfer horor dalam <i>game</i> berhasil memberikan pengalaman yang menegangkan.
	S3	Saya tertarik untuk memainkan <i>game</i> ini kembali di lain waktu.
	S4	Secara keseluruhan, saya puas dengan <i>game Escape From Basement</i> .

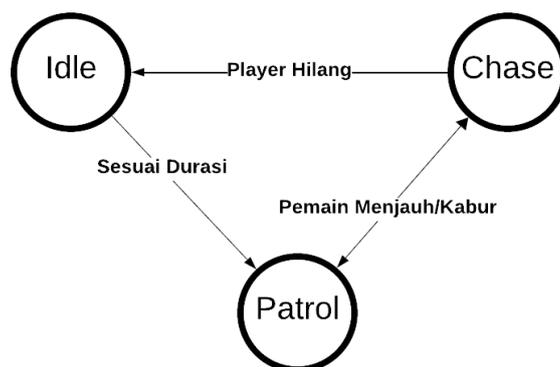
Penilaian setiap pertanyaan dalam kuesioner menggunakan skala *Likert* dengan lima tingkat, yaitu Sangat Tidak Setuju (1), Tidak Setuju (2), Netral (3), Setuju (4), dan Sangat Setuju (5). Hasil kuesioner yang telah diisi akan dianalisis lebih lanjut menggunakan perhitungan skala *Likert*.

2.3 Implementasi *Finite State Machine (FSM)* untuk Monster

Finite State Machine (FSM) digunakan dalam *game Escape From Basement* untuk mengontrol perilaku monster berdasarkan *Waypoint Navigation* tertentu dalam permainan. *FSM* bekerja dengan mendefinisikan serangkaian *state* (keadaan) dan transisi antar *state* yang bergantung pada aksi pemain dan kondisi lingkungan.

2.3.1 State pada Monster

1. *Idle* (Diam) → Monster diam di tempat sebelum mulai bergerak atau setelah kehilangan target.
2. *Patrol* (Berpatroli) → Monster bergerak dari satu titik *waypoint* navigasi (biru) ke titik lainnya dalam pola yang telah ditentukan.
3. *Chase* (Mengejar) → Jika monster mendeteksi pemain dalam jangkauan penglihatannya, ia berhenti berpatroli dan mulai mengejar pemain.



Gambar 7. Diagram *Finite State Machine (FSM)* untuk monster

Deskripsi Diagram FSM:

1. *Idle* → Monster diam di tempat sebelum memulai patroli.
2. *Patrol* → Monster bergerak di antara titik *waypoint* navigasi warna biru, pada Gambar 4. Peta *Game Escape From Basement* dan secara acak monsternya ke *waypoint* navigasi.
3. *Chase* → Monster mengejar pemain jika mendeteksi keberadaannya.
4. Transisi antar *state* terjadi berdasarkan deteksi pemain dan kehilangan jejak pemain.

2.3.1 Transisi Antar State

1. Dari *Idle* → *Patrol* → Monster mulai bergerak mengikuti titik *waypoint* navigasi (biru) setelah durasi tertentu atau saat permainan dimulai.
2. Dari *Patrol* → *Chase* → Monster mendeteksi pemain dalam jangkauan penglihatannya, sehingga ia langsung berhenti berpatroli dan mulai mengejar pemain.
3. Dari *Chase* → *Idle*/*Patrol* → Jika monster kehilangan jejak pemain (misalnya, pemain keluar dari jangkauan penglihatan monster), monster bisa:
 1. Kembali ke *Idle* sebelum melanjutkan patroli.
 2. Langsung kembali ke *Patrol*, melanjutkan perjalanannya di titik *waypoint* terdekat.

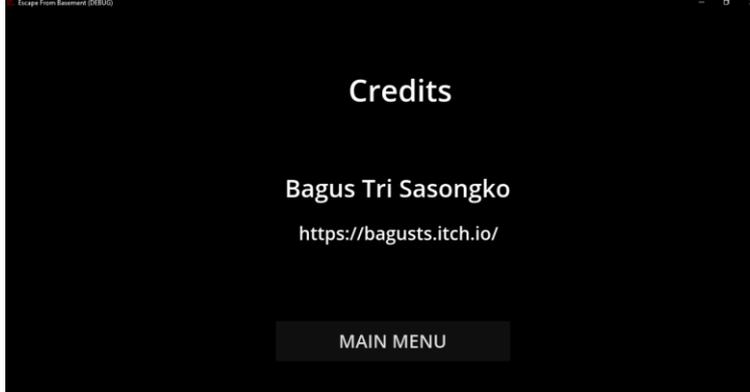
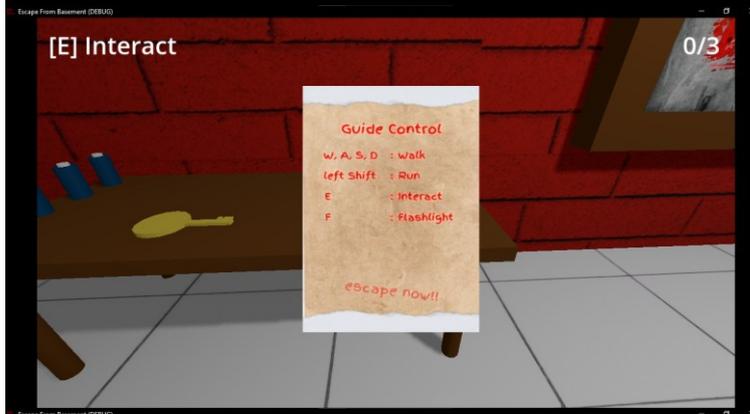
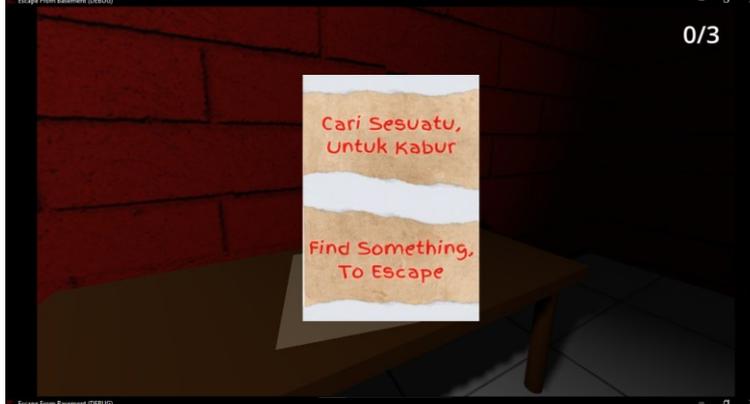
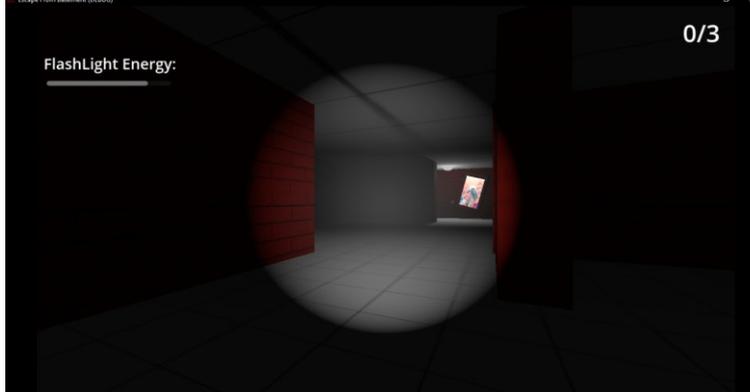
FSM ini memastikan monster berperilaku secara terstruktur, bergerak di sekitar lingkungan permainan dengan patroli yang terarah, dan berubah perilaku secara dinamis ketika melihat pemain. Hal ini menciptakan pengalaman permainan yang lebih menantang dan realistis.

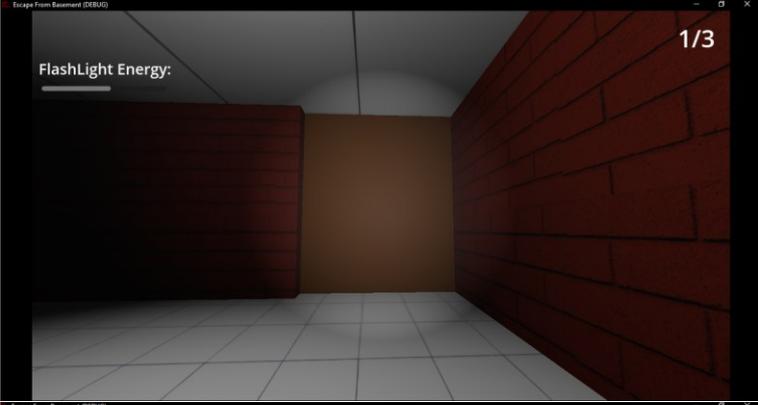
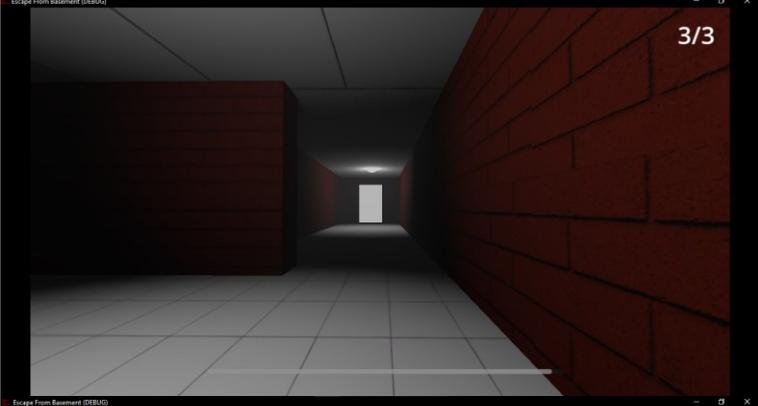
3. Hasil dan Pembahasan

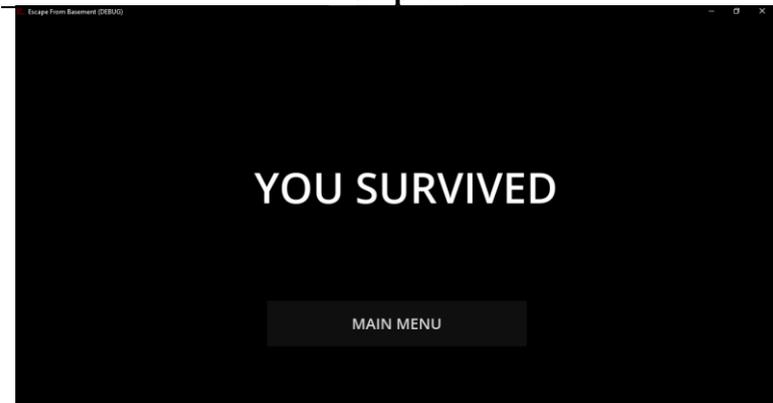
3.1 Hasil Tampilan *Escape From Basement*

Tabel 3. Deskripsi Tampilan

Tampilan	Deskripsi
	<p><i>Main Menu</i> : Berisi pilihan tombol play, ke menu credit, keluar game.</p>

Tampilan	Deskripsi
	<p><i>Menu Credit</i> : Berisi nama pembuat dan <i>link</i> akun itch io untuk <i>platform game web</i>.</p>
	<p><i>Menu Gameplay</i> : Menampilkan <i>Guide Control</i> saat dimainkan.</p>
	<p><i>Menu Gameplay</i> : Menampilkan <i>Guide</i> untuk memberikan <i>clue</i> tujuan <i>gamenya</i> harus bagaimana.</p>
	<p><i>Menu Gameplay</i> : Kondisi keluar ruangan <i>spawn</i> dan kondisi pencahayaan nya dan penggunaan <i>Flashlight</i> pada saat dimainkan.</p>

Tampilan	Deskripsi
	<p><i>Menu Gameplay</i> : Item “Pedang” untuk membuka gerbang kayu.</p>
	<p><i>Menu Gameplay</i> : Gerbang Kayu untuk ke arah cahaya putih.</p>
	<p><i>Menu Gameplay</i> : Cahaya putih untuk <i>ending game</i> selesai (memvisualisasikan keluar dari <i>basement</i>).</p>
	<p><i>Menu Death</i> : Menampilkan <i>Death Scene</i> ketika player karakter terkena monster.</p>

Tampilan	Deskripsi
	<p><i>Menu Survived</i> : Menampilkan Survived Scene ketika player karakter menyentuh cahaya putih.</p>

3.2 Beta Testing

Berdasarkan hasil pengujian beta testing, diperoleh data dari 25 responden yang telah memainkan *game Escape From Basement* dan mengisi kuesioner *Usability Testing*. Data yang dikumpulkan dianalisis menggunakan skala *Likert* untuk mengukur aspek *learnability*, *memorability*, *efficiency*, *errors*, dan *satisfaction* terhadap *game*.

Hasil perhitungan rata-rata skor dari setiap indikator disajikan dalam Tabel 4, yang menunjukkan tingkat kepuasan pengguna terhadap pengalaman bermain *game Escape From Basement*.

Tabel 4. Hasil Penilaian Beta Testing

Indikator	Kode	Pertanyaan	Rata-rata Skor (%)	Kategori
<i>Learnability</i>	L1	Saya dapat memahami cara bermain <i>game</i> ini dengan mudah.	85%	Sangat Puas
	L2	Instruksi dalam <i>game</i> jelas dan mudah diikuti.	82%	Sangat Puas
	L3	Tujuan permainan mudah dipahami sejak awal.	80%	Sangat Puas
<i>Memorability</i>	M1	Saya dapat mengingat cara bermain <i>game</i> ini dengan mudah setelah beberapa kali bermain.	78%	Puas
	M2	Kontrol dalam <i>game</i> mudah diingat dan tidak membingungkan.	76%	Puas
<i>Efficiency</i>	E1	<i>Game</i> berjalan dengan lancar tanpa gangguan teknis yang berarti.	75%	Puas
	E2	Waktu <i>loading game</i> tidak terlalu lama.	82%	Sangat Puas
	E3	<i>Game</i> ini dapat dimainkan dengan baik di perangkat yang saya gunakan.	74%	Puas
<i>Errors</i>	ER1	Saya tidak menemukan banyak <i>bug</i> atau kesalahan saat bermain.	65%	Cukup Puas
	ER2	Tombol dan kontrol dalam <i>game</i> berfungsi sesuai dengan harapan.	70%	Puas
	ER3	Petunjuk dalam <i>game</i> sesuai dengan mekanisme permainan.	73%	Puas
<i>Satisfaction</i>	S1	Saya merasa terhibur saat memainkan <i>game</i> ini.	88%	Sangat Puas

Indikator	Kode	Pertanyaan	Rata-rata Skor (%)	Kategori
	S2	Atmosfer horor dalam <i>game</i> berhasil memberikan pengalaman yang menegangkan.	85%	Sangat Puas
	S3	Saya tertarik untuk memainkan <i>game</i> ini kembali di lain waktu.	83%	Sangat Puas
	S4	Secara keseluruhan, saya puas dengan <i>game Escape From Basement</i> .	86%	Sangat Puas

Dari hasil pengujian beta, sebagian besar indikator mendapatkan skor tinggi, khususnya pada *Learnability* dan *Satisfaction*, yang menunjukkan bahwa *game* ini mudah dipahami dan memberikan pengalaman bermain yang menyenangkan.

Namun, aspek *Errors* mendapatkan skor yang lebih rendah (65%), menunjukkan bahwa masih ada bug atau kesalahan teknis yang perlu diperbaiki sebelum *game* dirilis ke publik. Secara keseluruhan, *game Escape From Basement* mendapatkan umpan balik positif dari pemain dan dapat ditingkatkan lebih lanjut untuk pengalaman yang lebih optimal.

4. Kesimpulan

Pengujian beta pada *game Escape From Basement* telah dilakukan untuk mengevaluasi aspek *usability* berdasarkan lima indikator utama: *Learnability*, *Memorability*, *Efficiency*, *Errors*, dan *Satisfaction*. Hasil pengujian menunjukkan bahwa mayoritas responden merasa *game* ini mudah dipahami, mekanisme kontrolnya intuitif, dan pengalaman bermainnya memuaskan.

Dari segi efisiensi, *game* berjalan dengan baik pada sebagian besar perangkat tanpa kendala teknis yang signifikan. Namun, masih ditemukan beberapa *bug* minor yang perlu diperbaiki agar pengalaman bermain lebih optimal. Selain itu, aspek atmosfer horor dalam *game* berhasil memberikan pengalaman yang menegangkan bagi pemain, sesuai dengan tujuan pengembangannya.

Finite State Machine (FSM) untuk Monster

Monster dalam *game* ini menggunakan sistem *Finite State Machine* (FSM) dengan tiga kondisi utama:

1. *Idle* – Monster diam di tempat ketika tidak ada pemain dalam jangkauan atau setelah menyelesaikan patrolinya.
2. *Patrol* – Monster bergerak mengikuti titik *Waypoint Navigation* yang telah ditentukan.
3. *Chase* – Ketika monster mendeteksi keberadaan pemain, ia akan beralih ke mode pengejaran untuk mengejar pemain hingga keluar dari jangkauan atau pemain berhasil melarikan diri.
4. Sistem FSM ini dirancang untuk meningkatkan tantangan dalam permainan serta menciptakan pengalaman bermain yang lebih dinamis dan menegangkan. Namun, beberapa peningkatan masih dapat dilakukan, seperti memperbaiki sistem deteksi pemain agar lebih responsif serta menyesuaikan kecepatan pengejaran untuk keseimbangan *gameplay*.

Pengembangan *Escape From Basement* mengikuti tahapan *Game Development Life Cycle* (GDLC) yang dimulai dari inisiasi, di mana konsep *game*, mekanisme *gameplay*, dan sistem AI monster ditentukan. Pada tahap *pre-production*, dilakukan perancangan *game*, dan implementasi FSM untuk perilaku monster. *Production* meliputi pengembangan fitur utama seperti navigasi *waypoint* dan sistem pengejaran AI. Selanjutnya, tahap testing dilakukan dengan pengujian internal dan beta testing untuk mengidentifikasi *bug* serta mengevaluasi aspek *usability*. Setelah melalui perbaikan, *game* memasuki tahap *release*, diikuti oleh *post-release* yang berfokus pada pemeliharaan dan pengembangan fitur tambahan agar pengalaman bermain lebih optimal.

5. Referensi

- [1] A. Habibi and M. I. Athoillah, "Pengembangan Game Survival Horor Unity 3D dengan Menerapkan AI pada NPC," *Jurnal Penelitian Rumpun Ilmu Teknik*, vol. 4, no. 1, pp. 189–200, doi: 10.55606/juprit.v4i1.3473.

- [2] A. Andi, J. Charles, O. Pribadi, C. Juliandy, and R. Robet, "Game Development 'Kill Corona Virus' For Education About Vaccination Using Finite State Machine and Collision Detection," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, Nov. 2022, doi: 10.22219/kinetik.v7i4.1470.
- [3] A. Pranselga, I. R. Setiawan, and W. Apriandari, "Implementasi Finite State Machine Pada Karakter NPC Musuh Dalam Game Adventure In Java".
- [4] F. S. F. Kusumah, H. Fajri, and M. Al Barake, "Game 3D 'Zombie Attack' dengan Menerapkan AI Pada NPC," *KREA-TIF*, vol. 7, no. 1, p. 20, May 2019, doi: 10.32832/kreatif.v7i1.2044.
- [5] A. Masyhudi, "PENGEMBANGAN GAME SURVIVAL HORROR 3D RAPTA MENGGUNAKAN METODE FINITE STATE MACHINE," 2017.
- [6] R. K. Fachri, M. Z. Romdlony, and M. R. Rosa, "SIMULASI MODEL NAVIGASI MOBILE ROBOT DENGAN PENERAPAN METODE CONTROL LYAPUNOV-BARRIER FUNCTION (CLBF) TERHADAP SISTEM NAVIGASI WAYPOINT," 2021.
- [7] D. Abdurrohman, Y. A. Pranoto, and R. Primaswara Prasetya, "PERANCANGAN GAME LOST FOREST 3D MENGGUNAKAN METODE FINITE STATE MACHINE BERBASIS DESKTOP," 2023.
- [8] F. Sakrani, S. Adi Wibowo, and F. Santi Wahyuni, "IMPLEMENTASI FINITE STATE MACHINE SEBAGAI KONTROL UNTUK NON PLAYER CHARACTER PADA GAME 'LASTRI AND THE LAST TREE,'" 2020.
- [9] M. B. Firdaus, A. Z. Waksito, A. Tejawati, M. Taruk, M. K. Anam, and A. Irsyad, "Finite state machine for retro arcade fighting game development," *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 14, no. 1, p. 102, Apr. 2025, doi: 10.11591/ijict.v14i1.pp102-110.
- [10] A. F. Pukeng, R. R. Fauzi, Lilyana, R. Andrea, E. Yulsilviana, and S. Mallala, "An intelligent agent of finite state machine in educational game 'flora the Explorer,'" in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Nov. 2019. doi: 10.1088/1742-6596/1341/4/042006.
- [11] M. Y. Febrianta *et al.*, "Analisis Ulasan Indie Video Game Lokal pada Steam Menggunakan Analisis Sentimen dan Pemodelan Topik Berbasis Latent Dirichlet Allocation."
- [12] F. Mandita and B. Kusumo Jati, "Application of Finite State Machine in the 3D Game 'Virus Hunter,'" *Jurnal Ilmu Komputer dan Desain Komunikasi Visual*, vol. 7, no. 2, 2022.
- [13] N. C. Alam and I. A. Mastan, "'Waddle' Game Application Using Godot Game Engine Technology," vol. 7, no. 1, pp. 37–48, doi: 10.30813/jbase.v7i1.5604.
- [14] B. Rahman, M. Naufal Shofy, and S. Andryana, "IMPLEMENTATION OF FISHER-YATES SHUFFLE ALGORITHM IN ANDROID-BASED JAVANESE BATIK CULTURE EDUCATION GAME," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 10, no. 1, pp. 42–52, Jul. 2024, doi: 10.33480/jitk.v10i1.5256.
- [15] L. Husniah, B. F. Pratama, and H. Wibowo, "Gamification And GDLC (Game Development Life Cycle) Application For Designing The Sumbawa Folklore Game "The Legend Of Tanjung Menangis (Crying Cape)"", *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pp. 351–358, Oct. 2018, doi: 10.22219/kinetik.v3i4.721.
- [16] "PENERAPAN BYL's GAME DEVELOPMENT LIFE CYCLE DALAM PERANCANGAN VIDEO GAME ORACLE FOR ANGEL Mustofa 1 ; Vadlya Ma'arif 2 ; Ragil Wijianto 3 ; Frengki Pernando 4", [Online]. Available: www.bsi.ac.id
- [17] R. Ardiansyah, Y. Putra, U. Savika Putri, G. S. Permadi, and D. A. Dermawan, "PADA ANAK USIA DINI," vol. 10, no. 2, pp. 281–290, 2023, doi: 10.25126/jtiik.2023105961.
- [18] M. Luctkar-Flude, J. Tyerman, E. Ziegler, S. Walker, and B. Carroll, "Usability testing of the sexual orientation and gender identity nursing education eLearning toolkit and virtual simulation games," *Teaching and Learning in Nursing*, vol. 16, no. 4, pp. 321–325, Oct. 2021, doi: 10.1016/j.teln.2021.06.015.