

Perancangan game Attahdiid: Bertahan hidup dalam pondok dengan metode Finite State Machine

Muchammad Alif Zaidan^{1*}, Fresy Nugroho², Ahmad Fahmi Karami³, Reza Putra Pradana⁴

^{1,2,3,4} Program Studi Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang

e-mail: ^{*}210605110151@student.uin-malang.ac.id

Kata Kunci:

algoritma maze generation; desain karakter dan item; finite state machine; horror first-person survival; game.

Keywords:

maze generation algorithm; character and item design; finite state machine; survival horror first-person; game.

ABSTRAK

Penelitian ini membahas perancangan dan pengembangan game "Attahdiid: Bertahan Hidup dalam Pondok," sebuah permainan horror first-person survival yang memanfaatkan metode Finite State Machine (FSM) dan algoritma Maze Generation. FSM digunakan untuk mengontrol perilaku karakter, mengatur alur cerita, dan menyusun tantangan dalam permainan. Algoritma Maze Generation memberikan efisiensi dalam pembuatan level permainan secara otomatis, menciptakan pengalaman bermain yang dinamis dan bervariasi. Game ini menghadirkan suasana pondok pesantren yang autentik melalui desain karakter dan item yang dipertimbangkan dengan detail. Penerapan struktur menu yang sederhana mempermudah pengaturan permainan, meningkatkan aksesibilitas pemain, dan memberikan fokus

pada pengalaman bermain. Selain memberikan hiburan, "Attahdiid" menggabungkan nilai agama dan tantangan kompleks, menghadirkan pemain pada sebuah pengalaman yang lebih mendalam. Dengan menyatukan kreativitas dalam desain permainan dan penerapan teknologi, "Attahdiid" bukan hanya sebuah permainan yang menarik, tetapi juga mencerminkan pendekatan holistik terhadap pengembangan game yang dapat menjadi inspirasi bagi pengembang dan peneliti di masa depan.

ABSTRACT

This research discusses the design and development of the game "Attahdiid: Survival in the Hut," a first-person horror survival game that utilizes the Finite State Machine (FSM) method and Maze Generation algorithm. FSM is employed to control character behavior, organize the storyline, and create challenges within the game. The Maze Generation algorithm enhances efficiency in automatically creating game levels, resulting in a dynamic and varied gaming experience. The game creates an authentic atmosphere of a pesantren hut through meticulously considered character and item designs. The implementation of a simple menu structure facilitates game settings, improves player accessibility, and emphasizes the gaming experience. Beyond providing entertainment, "Attahdiid" integrates religious values and complex challenges, immersing players in a deeper experience. By combining creativity in game design with technological implementation, "Attahdiid" is not only an engaging game but also reflects a holistic approach to game development that can serve as inspiration for developers and researchers in the future.



This is an open access article under the [CC BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/) license.

Copyright © 2023 by Author. Published by Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pendahuluan

Perkembangan industri game telah menciptakan peluang yang tak terbatas dalam pengembangan permainan yang menarik dan inovatif. Game yang baik tidak hanya menawarkan hiburan, tetapi juga dapat menjadi alat edukatif yang efektif, sarana kreatif untuk menyampaikan cerita, serta tempat bagi pemain untuk mengembangkan keterampilan dan strategi (Kurniawati Mahardika et al., 2023). Dalam konteks ini, pengembangan game menjadi semakin menarik, dengan berbagai teknologi dan metode yang digunakan untuk menciptakan pengalaman bermain yang menarik.

Salah satu aspek penting dalam perancangan game adalah implementasi perilaku karakter dan logika permainan. Dalam game "Attahdiid: Bertahan Hidup dalam Pondok," mengimplementasikan metode Finite State Machine (FSM), dimana metode ini memainkan peran penting dalam mengendalikan perilaku karakter utama, menyusun tantangan, dan mengarahkan alur cerita secara koheren. FSM menjadi pondasi dalam memastikan karakter utama berinteraksi dengan makhluk halus, dan mampu mengikuti perkembangan cerita sesuai dengan perkembangan permainan (Haryanto, n.d.). Dengan memadukan aspek kreatif dalam pengembangan game dan pendekatan berbasis teknologi seperti FSM, penulis mencoba membuktikan bagaimana game "Attahdiid" menggabungkan hiburan dengan nilai agama yakni ketakwaan seorang hamba kepada Tuhan serta menantang pemain melalui mekanisme permainan yang kompleks. Penulis akan membahas bagaimana FSM dapat digunakan untuk menciptakan permainan bertahan hidup yang menantang, dengan penekanan pada elemen-elemen desain yang mempengaruhi pengalaman pemain.

Dalam proses perancangan game "Attahdiid: Bertahan Hidup dalam Pondok," penulis juga mengintegrasikan kecerdasan buatan (AI) melalui algoritma Maze Generation. Algoritma ini berperan sentral dalam pembuatan level-level permainan, karena seluruh desain level dikelola secara otomatis oleh perhitungan matematis. Dengan demikian, algoritma Procedural Maze & Dungeon Generation memberikan efisiensi yang signifikan dalam pengembangan game, mempercepat proses penggalian konten dan menciptakan pengalaman bermain yang dinamis dan beragam untuk pemain.

Berdasarkan uraian di atas, maka dapat disimpulkan rumusan masalah yang akan diangkat oleh penulis yaitu Bagaimana implementasi metode Finite State Machine dan proses perancangan game Attahdiid: Bertahan Hidup dalam Pondok.

Melalui pembahasan ini, penulis berharap untuk memberikan wawasan yang lebih mendalam tentang penggunaan metode Finite State Machine dalam perancangan game "Attahdiid" dan dapat mempengaruhi kualitas gameplay serta pengalaman pemain. Dengan pemahaman yang lebih baik tentang penggunaan FSM dalam pengembangan game, penulis berharap para pengembang dan peneliti dapat menghasilkan permainan yang lebih menarik dan menantang di masa depan.

Metode Penelitian

Game "Attahdiid: Bertahan Hidup dalam Pondok" adalah genre permainan horror first person survival. Pada awal permainan, pengguna disajikan dengan kisah seorang santri yang tidak betah hidup di pondok. Santri ini kemudian memutuskan untuk melarikan diri dari pondoknya dalam upaya mencari kebebasan. Di awal permainan, karakter utama akan menghadapi berbagai rintangan, termasuk pertemuan dengan makhluk halus dan juga ustadz dari pondok tersebut. Kan tetapi karakter utama (pemain) memiliki senjata unik, yaitu berdoa, untuk melawan makhluk halus. Jika karakter utama bertemu dengan ustad maka karakter utama ini akan kembali ke state awal.

Dalam perancangan game "Attahdiid: Bertahan Hidup dalam Pondok," penulis ingin menciptakan suatu pengalaman yang membenamkan pemain dalam suasana pondok pesantren yang menarik dan autentik. Dengan memperhatikan detail dan suasana pondok, penulis ingin menghadirkan dunia dalam permainan yang memadukan kesenangan dengan ketakwaan seorang hamba kepada Tuhan. Dengan tampilan yang menarik dan kreatif, pemain akan dapat merasakan keunikan dan tantangan dalam menjalani kehidupan sehari-hari di dalam pondok pesantren, menciptakan pengalaman bermain yang tak berkesan.

Finite State Machine

Finite State Machine (FSM) adalah sebuah model matematis yang digunakan secara luas dalam pengembangan perangkat lunak dan permainan untuk menggambarkan dan mengatur sistem dengan sejumlah keadaan terbatas (Sifaulloh et al., 2021). Model FSM ini terdiri dari sejumlah keadaan (states) yang merepresentasikan kondisi atau situasi yang berbeda dalam sistem. Sistem ini dapat berpindah dari satu keadaan ke keadaan lainnya berdasarkan serangkaian masukan, peristiwa, atau transisi yang telah didefinisikan sebelumnya.

Dalam konteks pengembangan permainan, FSM adalah alat yang sangat berguna untuk mengatur perilaku berbagai elemen dalam permainan. Misalnya, karakter permainan, musuh, atau objek dalam permainan dapat memiliki berbagai keadaan, seperti "berjalan," "berlari," "menyerang," atau "mati." FSM membantu mengorganisir dan mengontrol bagaimana entitas tersebut berpindah antara keadaan-keadaan ini berdasarkan interaksi dengan pemain, lingkungan, atau kondisi lainnya dalam permainan.

Dalam Unity, penggunaan FSM adalah umum, dan sering digunakan dalam kombinasi dengan Bahasa Pemrograman C#. Pengembang menggunakan C# untuk mengimplementasikan logika FSM dalam skrip-skrip permainan. Ini memungkinkan pengendalian perilaku game objects dengan cara yang terstruktur dan mudah dimengerti. FSM memainkan peran penting dalam mengatur kecerdasan buatan karakter permainan, mengatur alur permainan, dan mengendalikan berbagai aspek permainan lainnya. Dengan bantuan FSM, pengembang dapat menciptakan permainan yang lebih kompleks, realistis, dan menarik.

Bahasa Pemrograman C#

Bahasa Pemrograman C# ("C sharp") adalah bahasa pemrograman yang dikembangkan oleh Microsoft. Ini adalah bahasa pemrograman yang kuat dan sering digunakan dalam pengembangan perangkat lunak, terutama dalam pengembangan permainan dengan Unity (Wahyudi et al., 2022). C# adalah bahasa pemrograman berbasis objek yang memiliki sintaks yang mudah dipahami dan kaya fitur. Dalam konteks Unity, C# digunakan untuk mengembangkan skrip yang mengontrol perilaku game objects, mengatur logika permainan, dan berinteraksi dengan berbagai komponen dan sistem dalam permainan.

Design Level

Dalam game "Attahdiid: Bertahan Hidup dalam Pondok," pemain akan menghadapi tantangan yang sangat berkaitan dengan keberhasilan dalam mengalahkan musuh-musuhnya. Keberhasilan dalam mengatasi dan mengalahkan musuh akan menjadi faktor penentu untuk melanjutkan ke tingkat level berikutnya dan menentukan tingkat kesulitan yang dihadapi oleh pemain (Wibowo & Hardiwinata, 2023). Beberapa aturan permainan yang ada di dalam game ini yaitu:

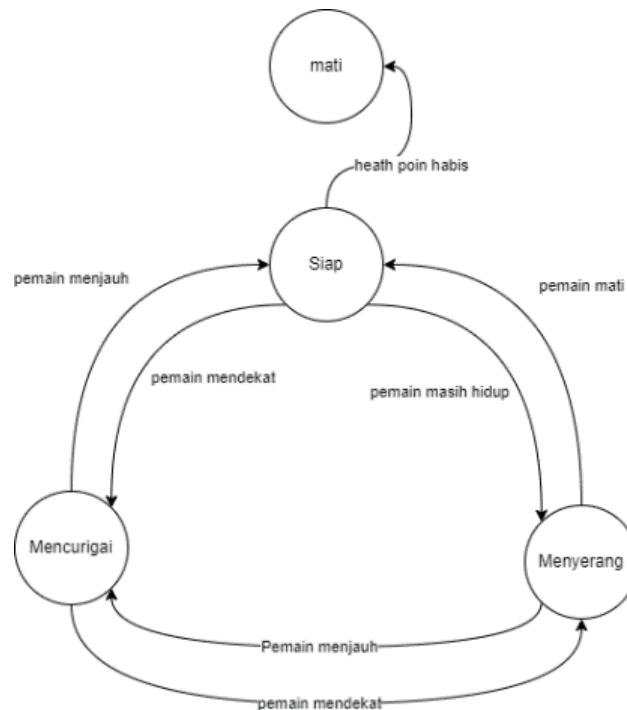
1. Player memiliki aksi utama seperti lari (running), melompat (jump), berjalan (walk), dan berdoa (pray).
2. Player mampu mengalahkan musuh, kecuali ustadz dengan berdoa.
3. Jika player bertemu dengan ustadz maka player akan kembali ke state awal.
4. Jika player mampu mengalahkan musuh maka player akan menghadapi tantangan yang lebih berat

Game Control

Game ini dapat dimainkan dengan menggunakan control mapping yang intuitif. Pemain dapat menggerakkan karakter dengan tombol WASD, tombol Shift digunakan untuk berlari, dan tombol Spasi untuk melompat. Selain itu, pemain juga dapat mengendalikan kamera dengan mouse. Pemain dapat mengambil item dengan tombol F, menggunakan item dengan tombol E, berinteraksi dengan objek atau karakter lain dengan tombol I, dan melakukan perlawanan dengan berdoa melalui tombol klik kiri mouse.

Diagram Finite State Machine

Gambar 1.1 FSM NPC Musuh



Gambar 1.1 Diagram Finite State Machine NPC Musuh

Tabel 1. Penerapan Metode dan State NPC Musuh.

Status	Event	Action
Status	Event	Action
Siap/idle	Posisi awal Ketika game dimulai	NPC berada pada posisi idel di tempat
Mencurigai	Ketika player mendekati NPC	NPC berjalan mendekati player
Menyerang	Ketika player berada di jarak serang	NPC menyerang player hingga player mati

Berdasarkan tabel diatas NPC memiliki sejumlah status, event, dan action yang mengontrol perilaku mereka. Mereka memulai dalam status "Siap/Idle" ketika permainan dimulai, tetapi akan beralih ke status "Mencurigai" ketika pemain mendekati. Jika pemain berada dalam jarak serangan, NPC akan menjadi "Menyerang," dan jika pemain berhasil mengalahkan NPC, mereka akan menjadi "Mati." Semua ini membentuk

kerangka kerja penting dalam mengatur bagaimana NPC berinteraksi dengan pemain dalam permainan.

Dalam permainan, karakter NPC memiliki status awal "Siap/Idle," di mana mereka berada di tempat dan tidak aktif. Ketika pemain mendekati NPC, mereka beralih ke status "Memberi Petunjuk," yang mengizinkan NPC memberikan petunjuk dan informasi kepada pemain untuk membantu mereka dalam permainan (Arif et al., n.d.). Ini memungkinkan NPC untuk berfungsi sebagai sumber informasi yang merespons interaksi pemain dengan memberikan panduan yang relevan.

Flowchart

Flowchart adalah representasi visual yang digunakan untuk menjelaskan urutan langkah atau alur tindakan dalam suatu proses atau peristiwa (Saputra & Rahmadhani, 2021). Flowchart menggunakan simbol-simbol grafis dan panah-panah yang menghubungkannya untuk menggambarkan setiap langkah atau keputusan dalam suatu alur kerja atau permainan. Dalam flowchart game "Attahdiid: Bertahan Hidup dalam Pondok," alur dimulai dengan "Start," menunjukkan titik awal dari proses atau permainan. Kemudian, langkah pertama adalah "Mendapatkan Item," yang mengindikasikan bahwa pemain atau karakter dalam permainan mendapatkan item tertentu. Setelah itu, langkah kritis adalah "Menghindari Rintangan," dimana pemain harus memutuskan apakah mereka akan menghindari rintangan atau tidak. Jika berhasil menghindari, langkah selanjutnya adalah "Keluar dari Pondok," yang merupakan pencapaian penting dalam permainan. Akhirnya, "Kebebasan" menunjukkan bahwa pemain atau karakter telah mencapai tujuan mereka, dan "End" adalah titik akhir dari alur.

Gambar 1.2 Flowchart



Gambar 1.2 Diagram Alur Karakter

Desain Karakter Dan Item

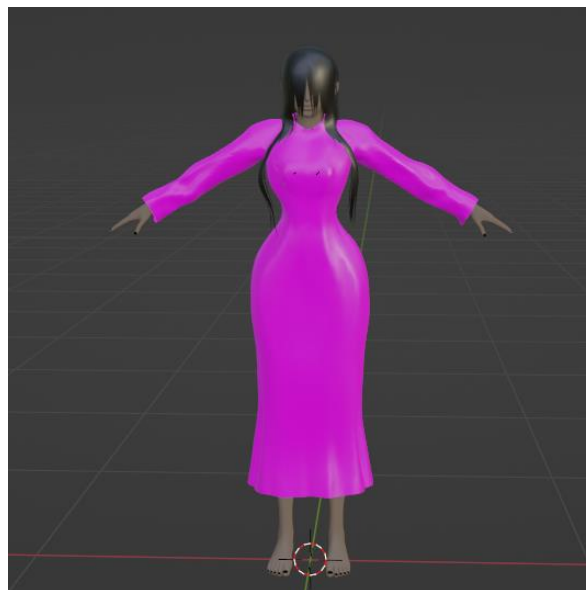
Karakter pada game "Attahdiid: Bertahan Hidup dalam Pondok" terdiri dari Udin sebagai karakter utama, hantu Kuntilanak sebagai musuh dan pondok pesantren sebagai latar cerita game, untuk membuat latar pondok pesantren ini, hanya diperlukan 1 tembok yang nantinya akan dimasukkan ke dalam Maze Generation yang akan membentuk sebuah koridor atau lorong. Adapun untuk detailnya sebagai berikut ini.

Gambar 1.3 Udin

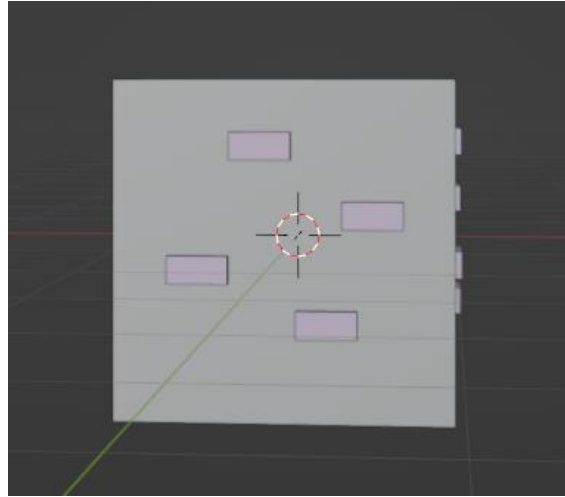


Gambar 1.3 Desain Karakter Udin

Gambar 1.4 Kuntilanak



Gambar 1.4 Desain Karakter Kuntilanak

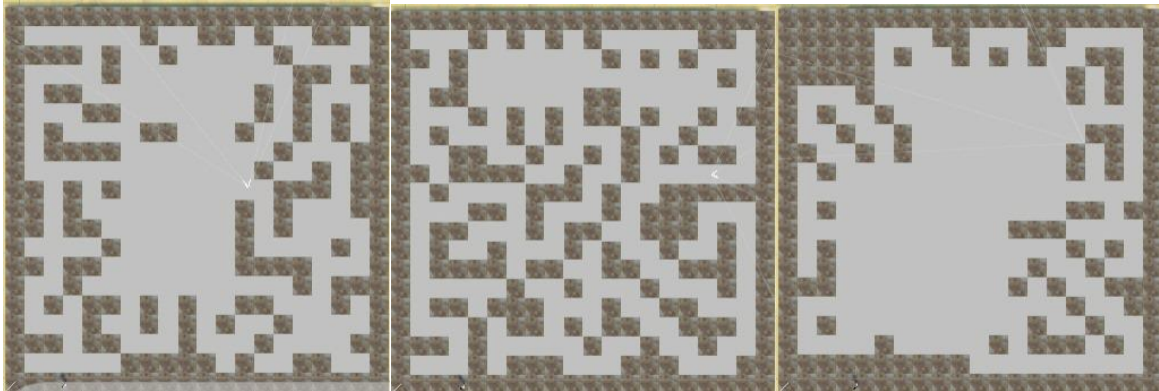
Gambar 1.5 Tembok**Gambar 1.5 Desain Tembok**

Pembahasan

Dalam hasil perancangan game, penulis akan membahas hasil konkrit dari perancangan game yang telah dilakukan. Penulis akan menampilkan hasil dari proses maze generation, memperlihatkan struktur menu yang telah dirancang, dan membahas bagaimana interaksi player dengan musuh. Dengan komponen-komponen kunci ini, kita dapat memberikan gambaran yang holistik tentang bagaimana desain permainan ini berkembang dan bagaimana elemen-elemen tersebut berkontribusi pada pengalaman bermain yang diinginkan.

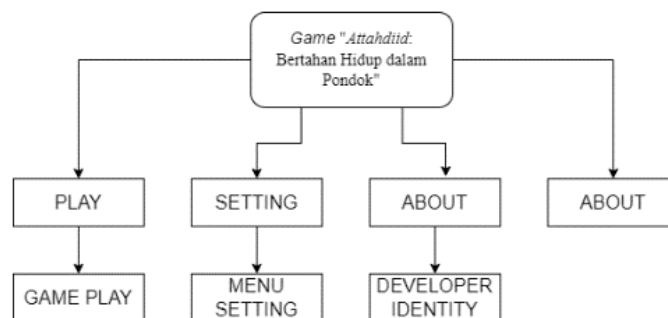
Maze Generation

Maze generation dalam Unity dapat diimplementasikan melalui berbagai algoritma, dua di antaranya adalah Recursive Backtracking dan Depth-First Search (DFS). Algoritma Recursive Backtracking bekerja dengan memilih sel acak sebagai titik awal, kemudian secara acak memilih tetangga yang belum dikunjungi dan menandainya sebagai bagian dari labirin. Proses ini berlanjut dengan backtracking ketika tidak mungkin melanjutkan (SAAT Lab in Computer Engineering Department, Ankara University et al., 2016). Implementasinya dalam Unity menggunakan C# melibatkan rekursi dan pemilihan tetangga secara acak. Algoritma DFS, sementara itu, juga mengambil pendekatan serupa dengan menandai sel dan mengeksplorasi tetangga yang belum dikunjungi, tetapi menggunakan pendekatan depth-first dengan mundur sejauh mungkin sebelum mencoba jalur lain (SAAT Lab in Computer Engineering Department, Ankara University et al., 2016). Keduanya dapat memberikan hasil labirin yang dinamis dan menarik untuk permainan Unity. Penting untuk diingat bahwa implementasi ini hanya mencakup pembuatan struktur labirin dan dapat diintegrasikan dengan logika pemain atau tujuan permainan sesuai kebutuhan pengembang.

Gambar 2.1 Maze Generation DFS**Gambar 2.1** Contoh Bentuk Generate 3 Maze

Struktur Menu

Struktur menu pada game ini mempunyai tiga menu yang terdiri dari menu Play, Setting, About, Exit. Jika user memilih menu play maka akan diarahkan pada Game Play, jika memilih Setting maka user akan diarahkan pada menu setting game, jika memilih Exit maka user akan keluar dari permainan. Berikut ini adalah diagram setruktur menu.

Gambar 2.1 Struktur Menu**Gambar 2.1** Struktur Menu Awal Game

Berikut ini adalah tampilan dari menu game "Attahdiid: Bertahan Hidup dalam Pondok" yang terdiri dari Play, Setting, Exit

Gambar 2.2 UI Struktur Menu**Gambar 2.2** Tampilan Antarmuka Struktur Menu Awal Game

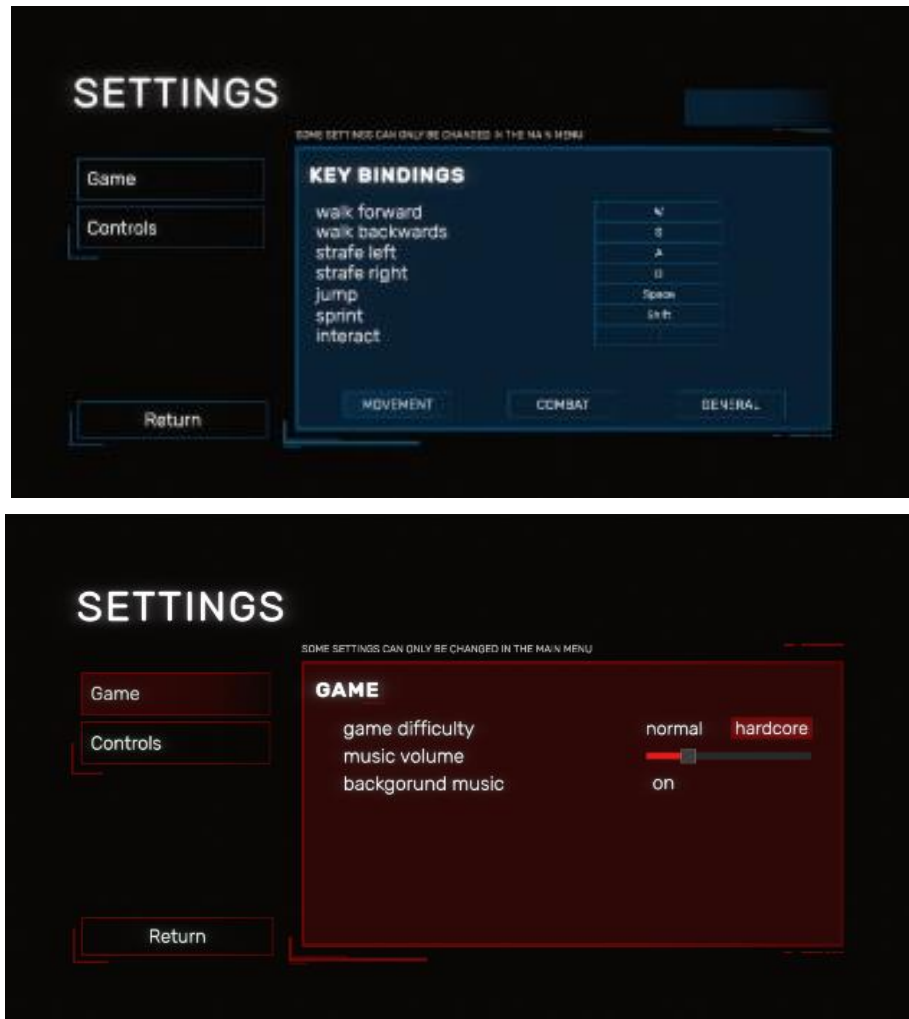
Ketika user memilih menu play maka akan diarahkan pada Game Play dan akan memunculkan tampilan berikut ini

Gambar 2.3 Play Mode**Gambar 2.3** Tampilan Antarmuka didalam Game

Jika user memilih menu Setting yang berisi pengaturan Key Bindings seperti user dapat melakukan aksi walk forward dengan menekan tombol W, Walk Backwards dengan menekan tombol S, Strafe Left dengan menekan tombol A, Strafe Right dengan menekan tombol D, Jump dengan menekan tombol Space, Interact dengan menggunakan tombol I, dan dilengkapi dengan 3 tombol button yang terdiri dari Movement, Combat, General(Winarno, 2014).

Selain itu di dalam menu Setting ini juga user dapat mengatur Difficulty Game user dapat memilih apakah ingin bermain game dengan level normal apa hardcore. Selain itu user juga dapat mengatur pengaturan volume dan background music.

Gambar 2.4 UI Setting



Gambar 2.4 Tampilan Antarmuka Halaman Setting

Jika user memilih menu about maka akan menampilkan identitas dari developer seperti gambar di bawah ini.

Gambar 2.5 UI About**Gambar 2.5** Tampilan Antarmuka Halaman About

Jika user sudah dapat bermain di Play Mode dan tidak berhasil mengalahkan musuh maka akan muncul tampilan Game Over yang nantinya user akan suguhkan dengan dua option yaitu kembali ke Home atau bermain game kembali seperti gambar di bawah ini.

Gambar 2.6 UI Game Over**Gambar 2.6** Tampilan Antarmuka Halaman Game Over

Kesimpulan dan Saran

Dalam perancangan game "Attahdiid: Bertahan Hidup dalam Pondok," pengembang berhasil mengintegrasikan metode Finite State Machine (FSM) untuk mengendalikan perilaku karakter utama, mengatur alur cerita, dan menyusun tantangan dalam permainan. FSM memberikan kerangka kerja yang efektif untuk menciptakan pengalaman bermain yang kompleks dan realistis. Selain itu, penggunaan algoritma Maze Generation meningkatkan efisiensi pengembangan game, memungkinkan

pembuatan level permainan secara otomatis, dan menciptakan pengalaman bermain yang dinamis. Desain karakter dan item dalam permainan berhasil menciptakan suasana pondok pesantren yang autentik, sementara struktur menu yang sederhana memberikan akses mudah untuk mengelola pengaturan permainan. Game ini tidak hanya menawarkan hiburan, tetapi juga menyampaikan nilai agama dan tantangan kompleks bagi pemain. Dengan demikian, "Attahdiid" menjadi contoh bagaimana kreativitas dalam desain permainan dapat diimbangi dengan penerapan teknologi untuk menghasilkan pengalaman bermain yang menarik dan bervariasi.

Daftar Pustaka

- Arif, Y. M., Wicaksono, A., & Kurniawan, F. (n.d.). Pergantian Senjata NPC pada Game FPS Menggunakan Fuzzy Sugeno.
- Haryanto, H. (n.d.). Reward Dinamis dalam Skenario Adaptif Menggunakan Metode Finite State Machine pada Game Edukasi. 2.
- Kurniawati Mahardika, E., Sevi Nurmanita, T., Anam, K., & Aditya Prasetyo, M. (2023). Strategi Literasi Budaya Anak Usia Dini melalui Pengembangan Game Edukatif. *Murhum : Jurnal Pendidikan Anak Usia Dini*, 4(2), 80–93. <https://doi.org/10.37985/murhum.v4i2.287>
- SAAT Lab in Computer Engineering Department, Ankara University, Safak, A. B., Bostanci, E., & Soylocicek, A. E. (2016). Automated Maze Generation for Ms. Pac-Man Using Genetic Algorithms. *International Journal of Machine Learning and Computing*, 6(4), 226–230. <https://doi.org/10.18178/ijmlc.2016.6.4.602>
- Saputra, R., & Rahmadhani, A. Y. (2021). Penerapan Finite State Machine Pada Game Role Playing. 1.
- Sifaulloh, H., Fadila, J. N., & Nugroho, F. (2021). Penerapan Metode Finite State Machine pada Game Santri on the Road. *Walisongo Journal of Information Technology*, 3(1), 11–18. <https://doi.org/10.21580/wjit.2021.3.1.7135>
- Wahyudi, I., Fadilah, J. N., & Nugroho, F. (2022). Perancangan Game Pair Matching untuk Pengenalan Huruf Hijaiyah Menggunakan Unity Game Engine. *Walisongo Journal of Information Technology*, 4(2), 139–146. <https://doi.org/10.21580/wjit.2022.4.2.7102>
- Wibowo, T., & Hardiwinata, W. H. (2023). Pengaruh Tingkat Kesulitan Tinggi Dalam Video Game Terhadap Motivasi Pemainnya. *INTECOMS: Journal of Information Technology and Computer Science*, 6(2), 1122–1132. <https://doi.org/10.31539/intecom.v6i2.7973>
- Winarno, B. H. (2014). Designing and Developing Interactive Learning Multimedia Using 3D Game Engine. *Jurnal Buana Informatika*, 5(2). <https://doi.org/10.24002/jbi.v5i2.356>