IMPLEMENTASI FSM (*FINITE STATE MACHINE*) PADA GAME *MUSLIMS EXPLORER'S FAITHFUL MAZE ADVENTURE*

Najah Muchsin Sanin [1*], Muhammad Sulthon. K [2], Furkon Aziz Aini [3], Moh Dawud [4], Faishal Reza Ubaidillah[5], Ahmad Fahmi Karami[6], Fresy Nugroho[7]
Universitas Islam Negeri Maulana Malik Ibrahim Malang

e-mail: 210605110015@student.uin-malang.ac.id, 210605110033@student.uin-malang.ac.id, 210605110128@student-uin-malang.ac.id, 210605110131@student.uin-malang.ac.id, 210605110132@student.uin-malang.ac.id, afkarami@ti.uin-malang.ac.id, fresy@ti.uin-malang.ac.id

Abstrak −Perkembangan industri game telah mengalami transformasi yang pesat dalam beberapa dekade terakhir. Dalam konteks ini, penekanan pengembangan game semakin ditujukan pada pengalaman player yang lebih mendalam dan kompleks. Sebagai tanggapan terhadap tuntutan tersebut, Finite State Machine menjadi salah satu pendekatan yang popular dan efektif dalam menciptakan alur cerita yang terorganisir dan mengelola karakter dengan lebih baik. Dengan menerapkan FSM pada game "*Muslims Explorer's Faithful Maze Adventure"* dapat memberikan pengalaman permainan yang lebih dinamis dan terstruktur. Game ini dirancang sebagai sarana dalam edukasi dan hiburan. Adapun, artikel ini akan mengulas konsep-konsep dasar FSM, menggambarkan implementasinya dalam game, dan menyoroti penggunaan FSM dalam meningkatkan kualitas pengalaman bermain.

Kata kunci: *Finite State Machine, Game Development*

PENDAHULUAN

Industri game telah menjadi salah satu ranah hiburan yang mengalami perkembangan pesat dalam beberapa dekade terakhir. Transformasi ini tidak hanya mencakup segi teknologi, tetapi juga aspek kreatif dalam merancang pengalaman bermain yang semakin mendalam dan kompleks. Dalam menghadapi tuntutan player yang semakin tinggi, pengembang game mencari pendekatan yang dapat memberikan alur cerita yang terorganisir dan mengelola karakter dengan lebih efisien. Salah satu solusi yang muncul sebagai jawaban atas tantangan tersebut adalah *Finite State Machine* (FSM).

Pendekatan FSM bukanlah hal yang baru, namun, peranannya semakin vital dalam menghadapi ekspektasi player yang semakin tinggi. Dengan menerapkan konsep-konsep dasar FSM, pengembang dapat menciptakan alur cerita yang terstruktur, karakter yang responsif, dan pengalaman bermain yang lebih dinamis. Artikel ini akan membahas implementasi FSM dalam konteks spesifik pada game yang dirancang untuk edukasi dan hiburan, yaitu "*Muslims Explorer's Faithful Maze Adventure*."

Sebelum memasuki diskusi mengenai FSM, penting untuk memahami latar belakang perkembangan industri game. Sejak era awal permainan video, kita telah menyaksikan perubahan signifikan dalam hal grafis, mekanika permainan, dan narasi (Toha, 2021). Pengembangan teknologi seperti mesin grafis yang canggih, kecerdasan buatan, dan platform game yang semakin kuat telah membuka pintu untuk pengalaman bermain yang lebih imersif.

Namun, semakin tingginya harapan player menuntut lebih dari sekadar visual yang mengagumkan. Mereka menginginkan cerita yang mendalam, karakter yang kompleks, dan pengambilan keputusan yang dapat memengaruhi alur cerita. Inilah titik di mana FSM muncul sebagai solusi yang dapat mengatasi kompleksitas tersebut FSM adalah

model matematis yang digunakan untuk merancang sistem yang memiliki sejumlah keadaan diskrit (Akbar Setiawan, 2023). Dalam konteks game, keadaan ini dapat merujuk pada kondisi karakter, tahapan cerita, atau respons terhadap input player. FSM terdiri dari sejumlah keadaan (*states*) yang bertransisi satu sama lain berdasarkan aturan tertentu.

Mengapa FSM menjadi begitu relevan dalam industri game? Karena FSM memberikan kerangka kerja yang jelas dan terstruktur untuk mengelola kompleksitas dalam pengembangan game. Pengembang dapat mendefinisikan keadaan-keadaan yang berbeda, mengidentifikasi perubahan keadaan, dan mengaitkan respons yang sesuai dengan setiap transisi. Dalam menciptakan pengalaman bermain yang mendalam, pengembang perlu memperhitungkan berbagai faktor, seperti bagaimana karakter berinteraksi dengan lingkungan, bagaimana cerita berkembang, dan bagaimana player dapat memengaruhi narasi. FSM menyediakan struktur yang memungkinkan pengembang untuk mengatasi kompleksitas ini dengan lebih efektif. Pertama-tama, FSM membantu mengorganisir logika permainan ke dalam keadaan- keadaan terdefinisi dengan jelas (Widodo, 2023). Misalnya, dalam game "*Muslims Explorer's Faithful Maze Adventure*," karakter utama dapat berada dalam keadaan "mengeksplorasi," dan "melawan enemy." Setiap keadaan ini memiliki respons unik terhadap input player atau peristiwa dalam game.

Selain itu, FSM mempermudah pemeliharaan dan perluasan game. Dengan memiliki struktur yang terorganisir, pengembang dapat lebih mudah menambahkan konten baru, memperbaiki *bug*, atau mengubah alur cerita tanpa mengganggu seluruh sistem. Hal ini sangat penting dalam menghadapi tuntutan player yang menginginkan konten tambahan dan pembaruan secara teratur.

Dalam konteks game ini, implementasi FSM dapat merangkum beberapa aspek kunci. Pertama-tama, FSM dapat digunakan untuk mengontrol gerakan karakter utama berdasarkan kondisi tertentu. Misalnya, saat karakter berada dalam keadaan "menghadapi enemy," FSM dapat mengatur respons karakter untuk menghindari atau mengatasi enemy tersebut.

Selain itu, FSM dapat digunakan untuk mengelola tahapan cerita. Game ini memiliki keadaan-keadaan seperti "memasuki labirin," "menemukan petunjuk," atau "melawan enemy". Setiap keadaan ini akan memicu perkembangan cerita yang sesuai dan menciptakan pengalaman bermain yang lebih terstruktur.

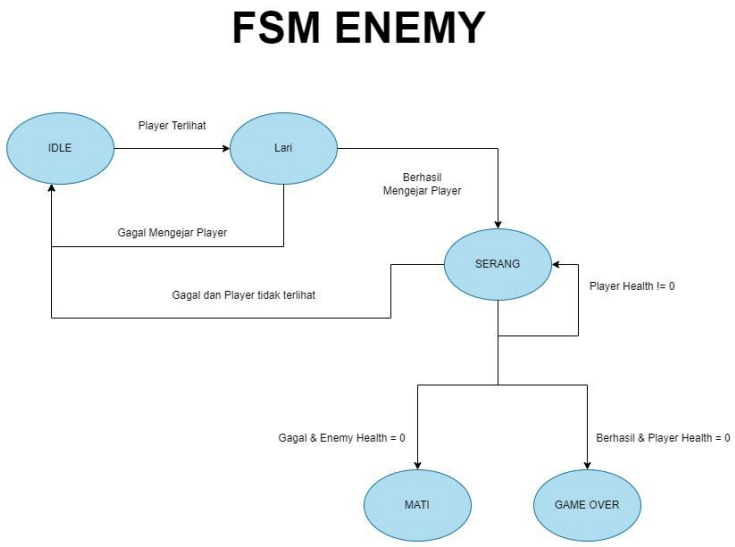
Penerapan FSM dalam "*Muslims Explorer's Faithful Maze Adventure*" tidak hanya memberikan struktur yang terorganisir, tetapi juga menghadirkan beberapa keunggulan khusus diantaranya, meningkatkan responsivitas karakter terhadap input player. Ketika karakter berada dalam keadaan "bertemu karakter *non-player*," dan dapat memastikan bahwa karakter tersebut merespons dengan dialog atau aksi yang sesuai.

Selain itu, FSM dapat membantu menciptakan variasi dalam pengalaman bermain. Dengan adanya keadaan-keadaan yang berbeda, player dapat mengalami berbagai situasi dan tantangan, menjadikan game lebih menarik dan dinamis (Bimantoro & Haryanto, 2016). Hal tersebut khususnya penting dalam konteks game edukasi, di mana variasi dapat meningkatkan efektivitas dalam penyampaian informasi.

METODE

Dalam pengembangan game "*Muslims Explorer's Faithful Maze Adventure*," penerapan *Finite State Machine* (FSM) menjadi kunci untuk menciptakan pengalaman

bermain yang terstruktur dan dinamis. Bagian ini akan membahas secara rinci metode yang digunakan dalam mengimplementasikan FSM dalam kerangka pengembangan game ini. Pada gambar 1 merupakan Implementasi *Finite State Machine* (FSM) pada NPC Enemy.



Gambar 1. Implementasi *Finite State Machine* pada Enemy

Berdasarkan gambar 1 terdapat 5 keadaan dalam game dengan pada karakter *enemy*. Setiap keadaan memiliki deskripsi yang menggambarkan kondisi serta respons yang diharapkan (Asyraq et al., 2020). Keadaan "*Idle*" mewakili kondisi dasar ketika *enemy* sedang tidak melakukan tindakan atau bergerak secara aktif. Ini bisa terjadi ketika tidak ada input dari *player* atau *enemy* berada dalam keadaan yang tidak memerlukan respons khusus (Gansala, 2016). Misalnya, *enemy* mungkin berada dalam keadaan diam atau menjaga suatu area. Keadaan "Lari" terjadi ketika *enemy* mendeteksi keberadaan *player* di sekitarnya. Dalam keadaan ini, *enemy* akan beralih ke mode mengejar dan berusaha mendekati *player*. Ini menciptakan respons dinamis terhadap keberadaan *player* dan menambah tingkat ketegangan dalam permainan.

Selanjutnya, keadaan "Serang" mewakili situasi ketika *enemy* telah berhasil mengejar *player*. Pada titik ini, *enemy* akan melancarkan serangan terhadap player dengan tujuan mengurangi kesehatan *player* hingga mencapai nilai nol. Hal tersebut dapat menciptakan ketegangan dan meningkatkan tingkat kesulitan permainan. Keadaan ini memiliki dua transisi, yaitu keadaan "Mati" dan keadaan "*Game Over*". Keadaan "Mati" terjadi ketika *enemy* gagal menyerang *player* dan sebaliknya diserang oleh *player* hingga kesehatan *enemy* mencapai nol. Pada saat ini, *enemy* dianggap telah dikalahkan atau mati. Hal tersebut menciptakan dinamika di mana *player* dapat mengalahkan *enemy* dan melanjutkan ke tahapan berikutnya dalam permainan.

Sedangkan keadaan "*Game Over*" merupakan kondisi terakhir yang terjadi ketika *enemy* berhasil menyerang pemain hingga kesehatan *player* mencapai nol. Pada saat ini, permainan akan berakhir, dan *player* akan dihadapkan pada *panel game over*. Hal tersebut memberikan hasil akhir dari interaksi antara *player* dan *enemy*, menciptakan kesempatan untuk mengulang atau mengevaluasi strategi permainan.

Dengan mengidentifikasi keadaan-keadaan khusus dan mendefinisikan respons yang sesuai, kita dapat menciptakan pengalaman bermain yang dinamis dan menarik

(Azahari et al., 2020). Implementasi FSM pada keadaan-keadaan ini membantu mencapai tujuan untuk memberikan tingkat kesulitan dan ketegangan yang sesuai dengan konsep permainan yang diinginkan (Tamyiz & Muni, 2023). Ringkasan mengenai keadaan-keadaan dalam *Finite State Machine* pada karakter *enemy* dapat dirangkum dalam tabel 1.

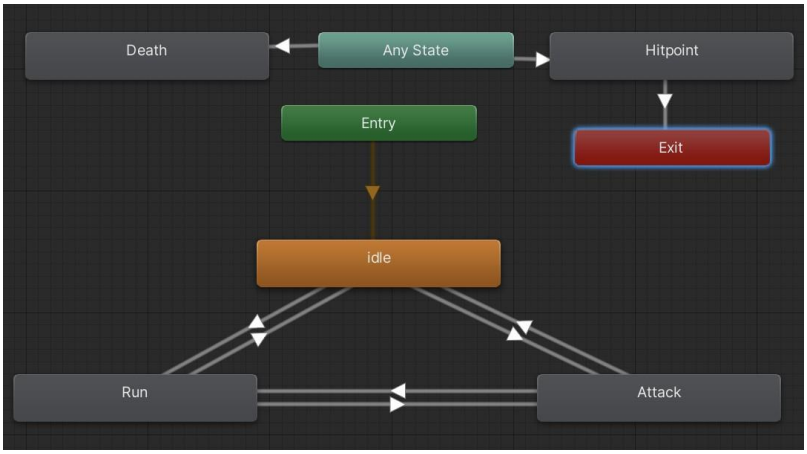
Tabel 1. Keadaan dalam *Finite State Machine* pada karakter *enemy*

No	*State*	Deskripsi
1.	*Idle*	Keadaan dasar yang mewakili kondisi ketika *enemy* sedang tidak melakukan tindakan atau bergerak secara aktif. Dengan kata lain, tidak ada input *enemy* yang diterima atau ketika *enemy* sedang berada dalam keadaan yang tidak memerlukan respons khusus.
2.	Lari	Keadaan ketika *enemy* dalam *game* mendeteksi keberadaan *player* di sekitarnya, maka *enemy* akan mengejar *player*.
3.	Serang	Keaadaan ketika *enemy* dalam *game* berhasil mengejar *player*, maka *enemy* akan menyerang *player* hingga *player health* = 0.
4.	Mati	Keadaan ketika *enemy* gagal menyerang *player* dan diserang oleh *player* hingga *enemy health* = 0 maka *enemy* akan mati.
5.	*Game Over*	Keadaan ketika *enemy* berhasil menyerang *player* hingga *player health* = 0, maka *game* akan berakhir.

HASIL KARYA UTAMA DAN PEMBAHASAN

Muslim Explorer's Faithful Maze Adventure merupakan *game action adventure* yang bersifat *solo player*, dimana ada 1 *player* dan beberapa NPC. Tujuannya adalah menyelesaikan *quest* dari NPC yang berupa pencarian item di dalam *dungeon* / labirin untuk mendapatkan reward, jika *quest* gagal dilakukan maka akan mendapat *punishment*/hukuman. Sebagai contoh: NPC memberi *quest* untuk mencari huruf tajwid dari bacaan idhar halqi di labirin A. Setelah mencari dan menemukan serta bersentuhan dengan huruf dari bacaan idhar halqi di labirin A, *quest*-nya selesai. Player tidak bisa meminta misi baru jika misi sebelumnya belum selesai.

Pada *game* ini menggunakan metode FSM pada NPC (*Non Player Character*) *Enemy* sebagai proses otonom dan adaptif sehingga karakter dapat bergerak sendiri secara otonom (Angkat et al., 2020). Konsep FSM pada NPC *Enemy* adalah NPC *Enemy* berada pada keadaan diam apabila *Player* tidak memasuki daerah jangkauan NPC *Enemy* (Nafianto, 2017). Apabila *player* memasuki daerah jangkauan NPC *Enemy* maka *NPC Enemy* akan mengejar *player* dan apabila *player* terjangkau maka NPC *Enemy* akan menyerang *player*. Penerapan alur FSM pada animator dapat dilihat pada gambar 2 dan penerapan pada kode program dapat dilihat pada gambar 3. Karakter NPC *Enemy* dapat dilihat pada gambar 4.

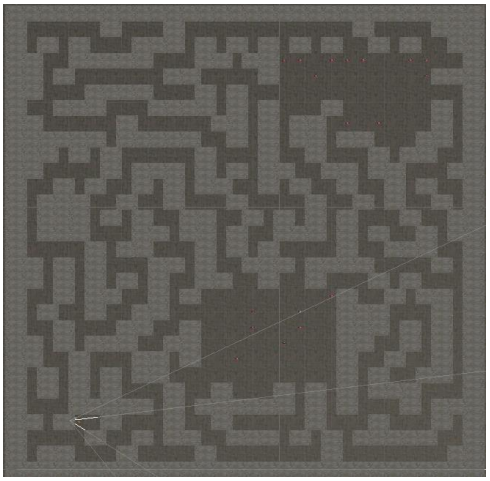


Gambar 3. *Animator* NPC (*Non Player Character) Enemy*

Gambar 4. NPC (*Non Player Character) Enemy*

Game ini akan menampilkan beberapa peti berdasarkan *quest* yang diberikan. Kemudian *player* akan menjelajahi labirin untuk menemukan peti tersubut. Lokasi peti akan dijaga oleh NPC *Enemy*. Labirin akan di-*generate* menggunakan menggunakan metode *Artifial Intelligence* yaitu, *Recursive* DFS. Tampilan menu awal sebelum *game* dimulai dapat dilihat pada gambar 5. Sedangkan, salah satu contoh hasil *generate* labirin setelah *game* dimulai dapat dilihat pada gambar 6.

Gambar 5. *Interface* Menu Utama



Gambar 6. Labirin

Setelah labirin di-*generate* maka *player* akan menjelajahinya untuk menemukan peti dalam rangka menyelesaikan *quest* yang telah diberikan. Adapun, setiap peti dijaga oleh NPC *Enemy*, sehingga apabila *player* ingin menyentuh peti maka harus melakukan perlawanan terhadap NPC *Enemy*. Ketika *player* memasukin daerah jangkauan NPC *Enemy* maka NPC *Enemy* akan mengejar *player* yang dapat dilihat pada gambar 7.

Gambar 7. NPC *Enemy* mengejar *player*

Sedangkan, *player* dapat melakukan perlawanan terhadap NPC *Enemy*, yang mana jika *player* berhasil menyerang NPC *Enemy* hingga *health*-nya sama denga 0, maka NPC *Enemy* akan mati, seperti pada gambar 8.

Gambar 8. (*Non Player Character) Enemy Death*

Namun, apabila *player* gagal melakukan penyerangan sehingga *health*-nya menjadi 0 akibat serangan balik dari NPC *Enemy*, maka *player* akan mati seperti pada gambar 9. Selanjutnya akan muncul *panel* "*Game Over*" seperti pada gambar 10.

Gambar 9. *Player Death*

Gambar 10. Panel *Game Over*

KESIMPULAN

Industri *game* mengalami perkembangan pesat dalam beberapa dekade terakhir, baik dari segi teknologi maupun aspek kreatif. Dalam menghadapi tuntutan *player* yang semakin tinggi, pengembang game mencari solusi untuk memberikan pengalaman bermain yang terorganisir dan responsif. Salah satu solusi yang muncul adalah penerapan *Finite State Machine* (FSM) dalam pengembangan game.

FSM, meskipun bukan konsep baru, semakin vital dalam menghadapi ekspektasi *player* yang menginginkan cerita yang mendalam, karakter yang kompleks, dan pengalaman bermain yang dinamis. FSM memberikan kerangka kerja terstruktur untuk mengelola kompleksitas dalam pengembangan game dengan mendefinisikan keadaan-keadaan yang berbeda, mengidentifikasi perubahan keadaan, dan mengaitkan respons yang sesuai.

Dalam konteks game "*Muslims Explorer's Faithful Maze Adventure*," penerapan FSM membawa beberapa keunggulan khusus. FSM membantu mengorganisir logika permainan ke dalam keadaan-keadaan terdefinisi dengan jelas, mempermudah pemeliharaan dan perluasan game, serta meningkatkan responsivitas karakter terhadap input player. Dengan adanya keadaan-keadaan yang berbeda, game ini dapat memberikan variasi dalam pengalaman bermain, yang penting dalam konteks game edukasi.

Metode implementasi FSM dalam game tersebut membawa hasil karya utama berupa "*Muslims Explorer's Faithful Maze Adventure*," sebuah game action adventure *solo player* dengan *quest* dari NPC. Penerapan FSM pada NPC *Enemy* memberikan karakteristik otonom dan adaptif pada musuh, meningkatkan dinamika permainan. *Game* ini menggabungkan konsep FSM dengan metode *Artificial Intelligence* untuk menghasilkan labirin yang menantang bagi player.

Dengan demikian, implementasi FSM dalam pengembangan game membuktikan keberhasilannya dalam menciptakan pengalaman bermain yang terstruktur, dinamis, dan responsif. *Game* "*Muslims Explorer's Faithful Maze Adventure*" menjadi contoh nyata bagaimana FSM dapat digunakan secara efektif dalam mencapai tujuan tersebut, memenuhi ekspektasi player modern yang semakin tinggi.

DAFTAR PUSTAKA

[1] Akbar Setiawan, F. (2023). *Perancangan Game Simulasi 3d Bahasa Inggris Metode Finite State Machine Berbasis Desktop*.

[2] Angkat, M. Y., Osmond, A. B., & Ansori, A. S. R. (2020). Membandingkan Algoritma Djikstra Dengan Algoritma A* star Menggunakan Metode Logika Fuzzy Di Unity 3d. *eProceedings of Engineering*, *7*(1).

[3] Asyraq, F. A., Hormansyah, D. S., & Astiningrum, M. (2020). Implementasi Fsm (Finite State Machine) Pada Game Surabaya Membara. *Jurnal Informatika Polinema*, *6*(2), 11–17.

[4] Azahari, A., Haristyawan, I., & Jariansyah, M. (2020). IMPLEMENTASI FSM SEBAGAI RESPON OTONOM DAN ADAPTIF NPC PADA GAME "AWANG MENJELAJAHI KOTA TENGGARONG." *Jurnal Ilmiah MATRIK*,*22*(2), 153–161.

[5] Bimantoro, T., & Haryanto, H. (2016). Pemodelan perilaku musuh menggunakan Finite State Machine (FSM) pada game pengenalan unsur kimia. *Journal of Applied Intelligent System*, *1*(3), 210–219.

[6] Gansala, F. (2016). *Perilaku non playable character (NPC) musuh pada game sepeda dengan menggunakan fuzzy state machine (FuSM)*.

[7] Nafianto, M. R. (2017). *Implementasi Algoritma Finite State Machine Sebagai Penentu Perubahan Perilaku Pada Non-Player Character Dalam Game Kiki Adventure*.

[8] Tamyiz, U. M. H., & Muni, L. S. A. (2023). Penerapan Finite State Machine Pada Squid game Berbasis Dekstop Menggunakan Unity 3D. *NUansa: Jurnal Penelitian, Pengabdian Dan Kajian Keislaman*, *2*(1), 10–21.

[9] Toha, I. F. (2021). *Representasi Sensualitas Pada Skin K/Da Popstar Dalam Game League Of Legends*.

[10] Widodo, B. (2023). PERANCANGAN GAME ARCADE 2D" SPACESHIP BATTLE" MENGGUNAKAN FLAME GAME ENGINE BERBASIS MOBILE. *Jurnal Ilmiah Betrik*, *14*(02 AGUSTUS), 402–413.