PERANCANGAN DAN IMPLEMENTASI METODE FSM (FINITE STATE MACHINE) PADA GAME MILITARY DEFENCE 2D BERBASIS ANDROID

Muhammad Hasan Syu'aibi, Ali Mahmudi, Karina Auliasari
Teknik Informatika, Institut Teknologi Nasional Malang
Jalan Raya Karanglo km 2 Malang, Indonesia
muhammadhasansyuaibi@gmail.com

ABSTRAK

Game *Tower Defense* adalah game strategi yang bertujuan untuk mencegah musuh melewati suatu wilayah atau area dengan menempatkan susunan *tower* seperti bangunan dan senjata. Game *tower defence* yang ada saat ini masih belum ada tindakan yang dilakukan musuh ketika terkena serangan atau *damage* dari *tower* dan belum adanya misi permainan pada game tersebut sehingga tidak memberikan pengalaman bermain yang memuaskan bagi pemain yang mencari tantangan. Game ini dirancang menggunakan game engine unity 2D dengan mengimplementasikan metode FSM (*Finite State Machine*) untuk pengambilan keputusan atau tindakan prilaku pada NPC (*Non Player Character*) karakter musuh atau *Monster* dengan menggunakan tiga hal yaitu State (Keadaan), Event (kejadian) dan action (aksi). Pengujian kecerdasan buatan yang diimplementasikan akan diuji dengan melakukan pengujian blackbox digunakan untuk mengamati hasil fungsional dari aplikasi dan mengamati apakah *input* dan *output* berjalan dengan baik. Dari pengujian yang telah dilakukan tersebut didapatkan hasil implementasi metode *Finite State Machine* pada NPC (*Non Player Character*) *Monster* atau *enemy* sudah bisa menujukkan prilaku cerdas berjalan dengan baik. Dari pengujian 18 orang responden, didapatkan hasil yang menunjukkan bahwa 25% (total 49) menyatakan sangat setuju, 70% (total 138) menyatakan setuju, dan 5% (total 11) menyatakan tidak setuju

Kata kunci : *Android, Finite State machine (FSM), Game, Military Defence, 2D*

1. PENDAHULUAN

Game *Tower Defense* adalah game strategi yang bertujuan untuk mencegah musuh melewati suatu wilayah atau area dengan menempatkan susunan *tower* seperti bangunan dan senjata. Musuh akan datang secara bergelombang dengan jumlah yang banyak dan *tower* akan menembak musuh dalam jangkauannya, biasanya *tower* memiliki berbagai jenis seperti tingkat kemampuan, biaya pembelian, dan biaya upgrade berbeda. Pemain akan mendapatkan *coin* setelah musuh dikalahkan, jumlah *coin* yang didapat berdasarkan jenis musuh, dan *coin* yang tersedia digunakan untuk meningkatkan *tower*. Namun, terdapat masalah yang saya temukan dari game *tower defence* yang ada saat ini yaitu tidak ada tindakan yang dilakukan musuh ketika terkena serangan dari *tower* dan belum adanya misi permainan yang ada pada game tersebut sehingga tidak memberikan pengalaman bermain yang memuaskan bagi pemain yang mencari tantangan.

Metode *Finite State Machine* (FSM) dipilih untuk menyelesaikan masalah diatas, karena metode FSM dapat membuat sebuah pengambilan keputusan atau tindakan, yaitu perancangan sistem kontrol yang menggambarkan perilaku atau prinsip kerja sistem dalam tiga hal yaitu *State* (Keadaan), *Event* (peristiwa) dan *Action* (aksi) [1]. Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada dalam *state* yang aktif. Sistem dapat beralih atau bertransisi ke *state* yang berbeda ketika menerima masukkan atau *event* tertentu, baik dari perangkat eksternal atau dari komponen sistem itu sendiri. Game engine yang digunakan untuk mengimplementasikan metode *Finite State Machine* (FSM) diantaranya yaitu Unity.

Unity merupakan game engine untuk mengembangkan game multi platform 2D dan 3D yang artinya game yang dibuat mendukung berbagai format file dan dapat berjalan di berbagai perangkat dan sistem operasi seperti Windows, mcOS, iOS, android. Grafik pada Unity juga menggunakan grafik tingkat lanjut seperti OpenGL dan DirectX. Di Unity juga terdapat banyak tools yang dapat memudahkan pengembangan game [2].

Berdasarkan uraian diatas, maka penulis ingin membuat sebuah game strategi "*Military Defence 2D*" berbasis android dengan mengimplementasikan FSM (*Finite State Machine*) untuk pengambilan keputusan atau tindakan NPC (*non-player character*) pada karakter *enemy* dan membuat misi atau tantangan permainan yang ditentukan oleh pengembang serta memberikan kecerdasan buatan pada *turret*. Game *military defence* 2D ini menceritakan sebuah pulau yang dihuni para tentara, manusia, dan hewan. Pada suatu hari pulau mendapat informasi bahwa pulau diserang sekumpulan *Monster*, kemudian seketika itu manusia mengabarkan kepada tentara untuk membantu melindungi semua makluk hidup dari serangan *Monster*.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Menurut [3] dalam penelitiannya yang berjudul "Penerapan Metode *Finite State Machine* Pada Game Adventure Trapped Miners" yang bertujuan untuk menerapkan metode *Finite State Machine* untuk menghasilkan suatu kecerdasan buatan pada game tersebut. Berdasarkan hasil penelitian penerapan metode FSM (*Finite State Machine*) dengan indikasi musuh dapat mengejar dan menyerang player dengan kondisi tertentu, pengujian AI menggunakan FSM (*Finite State Machine*) didapat hasil musuh dapat bergerak dengan otomatis sesuai dengan kecerdasan buatan yang diterapkan.

Menurut [4] dalam penelitiannya yang berjudul "Penerapan Metode *Finite State Machine* pada Game Santri on the Road" yang bertujuan untuk mengimplementasikan metode FSM (*Finite State Machine*) pada game Endles Runing Santri on the road dengan menggunakan *platform* Unity. Berdasarkan hasil penelitian implementasi *Finite State Machine* pada game berjalan dengan bagus, dan dapat menjadikan alternatif bagi anak-anak usia dini untuk belajar akan pentingnya dalam mencari ilmu dengan sungguh.

Menurut [5] dalam penelitiannya yang berjudul "Pengembangan Game The Galaxy Menggunakan Metode Fsm (*Finite State Machine*)" yang bertujuan untuk merancang dan mengembangkan game the galaxy dengan mengimplementasikan FSM (*Finite State Machine*) sebagai metode untuk menentukan reaksi NPC. Berdasarkan hasil penelitian pengujian dari 10 user adalah yang menyatakan 71.8% menarik, 19.09% cukup menarik, dan 9.09% kurang menarik, sehingga mayoritas user menilai game the galaxy menarik.

Menurut [6] dalam penelitiannya yang berjudul "Penerapan Metode *Finite State Machine* (FSM) Pada Game Agent Legenda Anak Borneo" yang bertujuan untuk menerapkan FSM hanya pada NPC *enemy* dari game tersebut. Berdasarkan hasil pengujian NPC yang dapat memberi respon atau memiliki tingkah laku sesuai dengan keadaan yang terjadi pada pemain atau NPC lainnya. Adapun kekurangan yang masih ditemukan dalam penelitian ini adalah belum ada cara atau aturan dari permainan, tampilan UI yang perlu ditingkatkan lagi menjadi lebih menarik, serta tidak ada informasi mengenai nyawa dari karakter pemain dan karakter musuh.

Menurut [7] dalam penelitiannya yang berjudul "Penerapan Metode Finite State Machine Pada Game The Relationship" yang bertujuan untuk menerapkan metode FSM (*Finite State Machine*) pada game The Relationship yang dapat memberikan gameplay yang menarik peluang yang dihasilkan memaksa pemain untuk mengambil keputusan di setiap interaksi. Berdasarkan hasil penerapan metode FSM (*Finite State Machine*) menghasilkan respon NPC yang berbeda menyesuaikan apa yang telah dimainkan oleh pemain. Penelitian ini masih memiliki kekurangan yaitu interaksi yang dilakukan NPC masih belum kompleks dan karakter yang masih terbatas.

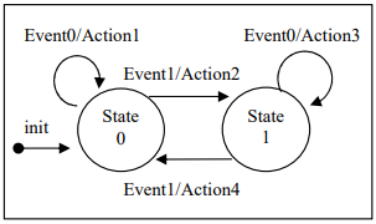
2.2. Kecerdasan Buatan

Kecerdasan buatan (*artificial intelligence*) merupakan simulasi dari kecerdasan yang dimiliki oleh manusia yang dimodelkan di dalam mesin dan diprogram agar bisa berpikir seperti halnya manusia.

Menurut [8], AI adalah untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan ditambah pengalaman. Penalaran (bagaimana membuat keputusan dan mengambil tindakan), moral yang baik. manusia cerdas dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki tentu lebih mampu menyelesaikan permasalahan. Tapi bekal pengetahuan tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil keputusan berdasarkan pengetahuan dan pengalaman yang dimiliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan pengalaman yang memadai manusia tidak bisa menyelesaikan masalah dengan baik [9].

2.3. Finite State Machine

Finite State Machine (FSM) adalah metodologi perancangan sistem kontrol yang menggambarkan perilaku atau prinsip kerja sistem dalam tiga hal berikut: *State* (Keadaan), *Event* (peristiwa) dan *Action* (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada dalam *state* yang aktif. Sistem dapat beralih atau bertransisi ke *state* yang berbeda ketika menerima masukkan atau *event* tertentu, baik dari perangkat eksternal atau dari komponen sistem itu sendiri. Transisi keadaan ini juga disertai dengan fungsi sistem yang merespons *input* yang terjadi. Aksi dapat berupa prosedur sederhana atau dapat melibatkan serangkaian proses yang relatif kompleks [3]. Contoh diagram *Finite State Machine* sederhana ditunjukkan pada Gambar 1.



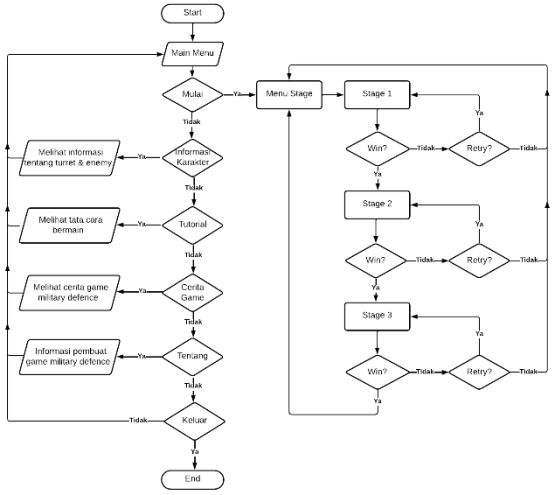
Gambar 1. Diagram finite state machine [10]

Alur diagram *Finite State Machine* pada Gambar 1 yaitu ada dua buah *state* dan 2 buah *input* serta empat buah aksi output yang berbeda : seperti terlihat pada gambar, ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *state*0, pada keadaan ini sistem akan menghasilkan *Action*1 jika terjadi masukan *Event*0, sedangkan jika terjadi *Event*1 maka *Action*2

akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State*1 dan seterusnya.

3. METODE PENELITIAN

3.1. Flowchart Game

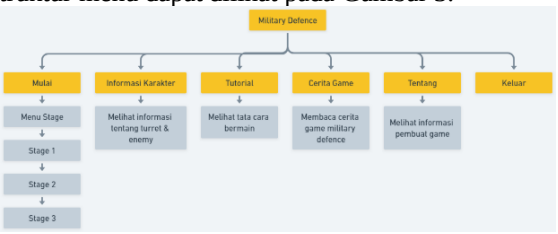


Gambar 2. Flowchart game

Pada proses pertama dari *flowchart* alur game tersebut adalah berada pada main menu yang berisikan mulai, informasi karakter, tutorial, cerita game, tentang, dan keluar. Pemain memilih mulai kemudian akan memuat menu *stage*, kemudian pemain memilih *stage* 1 karena *stage* 2 dan 3 masih terkunci dan menjalankan game tersebut, apakah game *stage* 1 menang, jika gagal maka pemain bisa memilih untuk *restart* atau mengulang *state* tersebut, namun apabila tidak ingin mengulang permainan maka diarahkan kembali ke halaman menu *state*, jika game *stage* 1 menang maka akan lanjut ke stage 2 dengan tingkat kesulitan yang ditingkatkan pada *enemy* atau musuh, sampai game benar-benar berakhir dan kembali ke halaman menu *stage*. Pada main menu jika pemain memilih informasi karakter akan menampilkan informasi tentang *turret* dan *enemy*, jika pemain memilih tutorial akan menampilkan tata cara bermain, jika pemain memilih cerita maka akan menampilkan cerita game tersebut, jika pemain memilih tentang maka akan menampilkan informasi tentang pembuat game, jika pemain memilih keluar maka pemain akan dikembalikan menuju *home* android, seperti yang ditunjukkan pada Gambar 2.

3.2. Blok Diagram Sistem

Pada game "*Military Defence*" ini terdiri dari beberapa menu yaitu mulai, informasi karakter, tutorial, cerita game, tentang, dan keluar. Diagram struktur menu dapat dilihat pada Gambar 3.



Gambar 3. Struktur Menu

Menu utama terdiri dari 6 menu yaitu mulai, informasi karakter, tutorial, cerita game, tentang, dan keluar. Sebelum memulai permainan pemain sebaikanya melihat tata cara bermain game tersebut dengan cara memilih menu tutorial, apabila ingin mengerti cerita dari game tersebut bisa memilih menu cerita game, jika ingin mengetahui informasi tentang *turret* dan *enemy* pilih menu Informasi karakter dan apabila ingin mengetahui informasi pembuat game pilih menu tentang, setelah itu pemain bisa memulai permainan tersebut dengan memilih menu mulai, kemudian pada menu *stage* pemain memilih *stage*, namun ketika pemain memilih menu keluar artinya pemain ingin mengakhiri permainan dan menuju *home screen* android.

3.3. Perancangan Karakter Turret

Pada Tabel 1 menampilkan desain karakter *turret* yang digunakan dalam game "*Military Defence*", berikut desain karakter yang digunakan.

Tabel 1. Perancangan Karakter *Turret*

No	Karakter Turret	Keterangan
1		One Missile *Turret* - *Damage* (5, 10, 15) - *Attack Delay* (sedang, cukup cepat, cepat) - *Attack Range* (pendek, sedang, panjang)
2		Dual Machine *Turret* - *Damage* per 1 misslie (2, 4, 6) - *Attack Delay* (sedang, cukup cepat, cepat) - *Attack Range* (pendek, sedang, panjang)
3		Tank *Turret* - *Damage* (15, 30, 45) - *Attack Delay* (sedang, cukup cepat, cepat) - *Attack Range* (pendek, sedang, panjang)
4		Ultra Tank *Turret* - *Damage* (25, 50, 75) - *Attack Delay* (sedang, cukup cepat, cepat) - *Attack Range* (pendek, sedang, panjang)

3.4. Perancangan Karakter Enemy

Pada Tabel 2 menampilkan desain karakter *enemy* yang digunakan dalam game "*Military Defence*", berikut desain karakter yang digunakan.

Tabel 2. Perancangan Karakter *Enemy* (Craftpix.net)

No	Karakter *Enemy*	Keterangan
1		Moster Red - Total nyawa 40 - Berlari normal

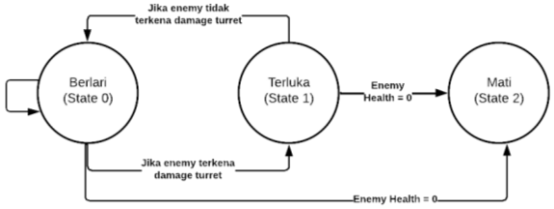
No	Karakter *Enemy*	Keterangan
2		Moster Purple - Total nyawa 60 - Berlari normal
3		Moster Green - Total nyawa 80 - Berlari normal - Namun jika nyawa tersisa <=15% maka berlari 2 kali lebih cepat
4		Moster Sword - Total nyawa 110 - Berlari normal - Namun jika nyawa tersisa <=50% maka berlari 2 kali lebih cepat
5		Moster Ogre - Total nyawa 220 - Berlari lambat

3.5. Rancangan Implementasi *Finite State Machine* Pada *Enemy*

Pengimplementasian metode FSM dalam membangun game *military defence* terletak pada relasi antara NPC *enemy* yang dilakukan dengan adanya trigger atau pemicu dari NPC *turret*.

1. Alur FSM *Enemy* Moster Red, Moster Purple, dan Moster Ogre

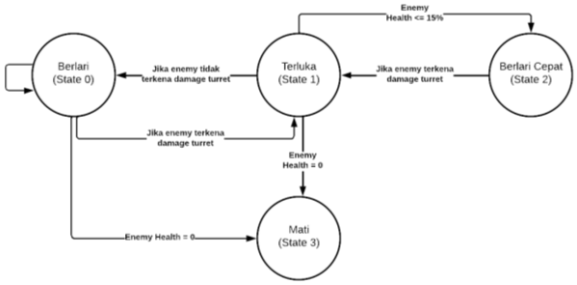
Penerapan alur FSM pada *enemy Monster* Red, *Monster* Purple, dan *Monster* Ogre yaitu pada saat *enemy spawn* dan mendapatkan jalur *waypoint*, secara default *enemy* masuk *state* berlari mengikuti jalur *waypoint*, sampai ketika jarak *enemy* <= max jangkauan serang *turret* lalu *enemy* terkena *trigger damage turret* maka *enemy* keadaan terluka dan nyawa *enemy* berkurang, jika jarak *enemy* > jangkauan serang *turret* dan tidak terkena *trigger damage turret* maka *enemy* akan kembali *state* berlari selama nyawa *enemy* > 0, namun ketika nyawa *enemy* <= 0 maka *enemy* masuk *state* mati, dapat ditunjukkan pada Gambar 4.



Gambar 4. Alur FSM pada *Enemy* Moster Red, Purple, dan Ogre

2. Alur FSM *Enemy* Moster Green

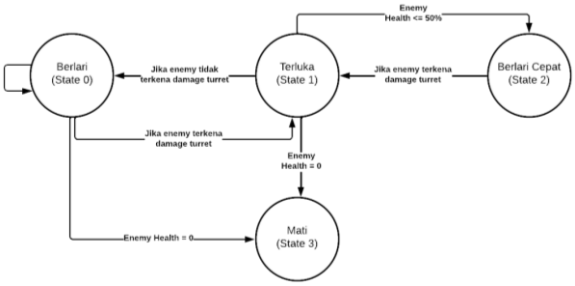
Penerapan alur FSM pada *enemy Monster* Green yaitu pada saat *enemy spawn* dan mendapatkan jalur *waypoint*, secara default *enemy* masuk *state* berlari mengikuti jalur *waypoint*, sampai ketika jarak *enemy* <= max jangkauan serang *turret* lalu *enemy* terkena *trigger damage turret* maka *enemy* keadaan terluka dan nyawa *enemy* berkurang, jika jarak *enemy* > jangkauan serang *turret* dan tidak terkena *trigger damage turret* maka *enemy* akan kembali *state* berlari selama nyawa *enemy* > 0, apabila pada *state* terluka nyawa *enemy* tersisa <= 15% maka *enemy* masuk ke kondisi berlari cepat, namun ketika nyawa *enemy* <= 0 maka *enemy* masuk *state* mati, dapat ditunjukkan pada Gambar 5.



Gambar 5. Alur FSM pada *Enemy* Moster Green

3. Alur FSM *Enemy* Moster Sword

Penerapan alur FSM pada *enemy Monster* Sword yaitu pada saat *enemy spawn* dan mendapatkan jalur *waypoint*, secara default *enemy* masuk *state* berlari mengikuti jalur *waypoint*, sampai ketika jarak *enemy* <= max jangkauan serang *turret* lalu *enemy* terkena *trigger damage turret* maka *enemy* keadaan terluka dan nyawa *enemy* berkurang, jika jarak *enemy* > jangkauan serang *turret* dan tidak terkena *trigger damage turret* maka *enemy* akan kembali *state* berlari selama nyawa *enemy* > 0, apabila pada *state* terluka nyawa *enemy* tersisa <= 50% maka *enemy* masuk ke kondisi berlari cepat, namun ketika nyawa *enemy* <= 0 maka *enemy* masuk *state* mati, dapat ditunjukkan pada Gambar 6.



Gambar 6. Alur FSM pada Enemy Moster Sword

4. HASIL DAN PEMBAHASAN

4.1. Tampilan Map Stage

Tampilan desain Map Stage dibuat dalam software coreldraw. *Stage* didesain dengan suasana didalam hutan. Pada Gambar 7 merupakan map stage pertama kali ketika pemain memulai permainan, jalur *enemy* cukup pendek berwarna coklat yang dibuat dengan *pen tool*, disamping jalur terdapat node yang berjumlah 7 dan terdapat pohon serta *environment* lainnya yaitu batu-batu, pohon yang sudah ditebang, tenda, burung, anjing, dan api unggun.

Gambar 7. Tampilan Map Stage Pertama

Tampilan desain map stage kedua, terdapat pohon serta *environment* lainnya yaitu batu-batu, pohon, ranting pohon, tenda, burung, kelinci, dan api unggun, jalur *enemy* berwarna coklat yang dibuat dengan *pen tool*, *enemy* yang muncul ada 3 jenis, disamping jalur terdapat node berjumlah 7 dapat dilihat pada Gambar 8 dibawah ini.

Gambar 8. Tampilan Map Stage Kedua

Tampilan desain map stage ketiga, terdapat pohon serta *environment* lainnya yaitu batu-batu, pohon yang sudah ditebang, ranting pohon, tenda, burung, dan api unggun, jalur *enemy* berwarna coklat yang cukup panjang dibuat dengan *pen tool*, *enemy* yang muncul ada 5 jenis, disamping jalur terdapat node yang berjumlah 8 dapat dilihat pada Gambar 9 dibawah ini.

Gambar 9. Tampilan Map Stage Ketiga

4.2. Implementasi Main Menu

Pada implementasi main menu dalam game "*Military Defence*" terdapat 6 tombol, yaitu tombol mulai untuk memilih stage dan memulai permainan, tombol tutorial berisikan cara bermain game, tombol cerita game untuk melihat kisah cerita game, tombol tentang berisikan informasi pembuat game, tombol icon buku berisikan informasi dari masing-masing karakter dalam game, dan tombol icon keluar untuk keluar dari game. Seperti yang ditunjukkan pada Gambar 10.

Gambar 10. Implementasi Main Menu

4.3. Implementasi Stage 1

Pada stage 1 dalam game "*Military Defence*" merupakan *stage* yang pertama kali dimainkan, pada *stage* 1 *enemy* yang muncul ada 2 jenis, jalur jalan untuk *enemy* cukup pendek, *turret* yang tersedia hanya ada 2, nyawa pemain berjumlah 20, *coin* awal yang diberikan berjumlah 265, dan total *wave* atau gelombang yang harus diselesaikan sebanyak 3. Seperti yang ditunjukkan pada Gambar 11.

Gambar 11. Implementasi Stage 1 Game Military Defence

4.4. Implementasi Pencapaian

Pada implementasi pencapaian game digunakan untuk melihat misi yang sudah dan belum diselesaikan, misi yang sudah selesai akan ditandai dengan gambar *mission complete*. Seperti yang ditunjukkan pada Gambar 12.

Gambar 12. Implementasi Pencapaian

4.5. Implementasi Menu Tutorial

Pada implementasi menu tutorial dalam game "*Military Defence*" berisikan tata cara dan alur bermain game *military defence*, terdapat button *next* dan *prev* untuk melihat informasi berikutnya dan sebelumnya, serta tombol main menu untuk kembali ke halaman main menu. Seperti yang ditunjukkan pada Gambar 13.

Gambar 13. Implementasi Menu Tutorial

4.6. Implementasi Menu Informasi Karakter

Pada implementasi menu cerita dalam game "*Military Defence*" berisikan informasi dari masing-masing karakter *turret* dan *Monster* seperti nyawa, *damage*, prilaku, dan jangkauan serang. Tombol main menu untuk kembali ke halaman main menu. Seperti yang ditunjukkan pada Gambar 14.

Gambar 14. Implementasi Menu Informasi Karakter

4.7. Implementasi Kecerdasan Buatan *Finite State Machine*

Metodologi FSM (*Finite State Machine*) pada game *Military Defence* digunakan untuk tingkah laku npc atau *non player character* pada *enemy*, seperti berlari, terluka dan mati. Pada Gambar 15 enemy akan masuk *state* berlari ketika tidak terkena trigger, namun jika terkena trigger yaitu jarak *enemy* <= max jangkauan *turret* dan terkena *damage turret* maka *enemy* dalam *state* terluka, kalau nyawa *enemy* <= 0 selama dalam *state* berlari atau terluka maka otomatis masuk *state* mati. *Capsule collider* berguna sebagai trigger ketika mendapati suatu *event* untuk masuk kedalam *state*.


Gambar 15. Implementasi FSM

4.8. Pengujian Metode *Finite State Machine*

Pengujian metode *Finite State Machine* yang diterapkan untuk pengendali animasi dan perilaku karakter *Monster* atau musuh dilakukan pada saat permainan dijalankan menghasilkan data pada tabel berikut.

1. Pengujian FSM *Enemy* Moster Red, Moster Purple, dan Moster Ogre

Pada Tabel 3 dapat diketahui bahwa *enemy* akan *spawn* dan berlari sampai *waypoint* akhir selama jarak *enemy* > jangkauan turret dan tidak terkena damage turret maka *enemy* masuk ke dalam *state* berlari, namun apabila jarak *enemy* <= max jangkauan serang atau area *circle collider* turret dan terkena damage turret maka *enemy* masuk *state* terluka. Dan ketika nyawa *enemy* <= 0 maka *enemy* mati.

Tabel 3. Pengujian FSM *Enemy* Moster Red, Moster Purple, dan Moster Ogre

No	State	kondisi	Hasil
1	*Enemy* Berlari	*Enemy* akan berlari sampai menuju *waypoint* akhir selama jarak *enemy* > jangkauan serang turret dan tidak terkena damage turret.	Sesuai
2	*Enemy* Terluka	*Enemy* akan terluka jika jarak *enemy* <= max jangkauan serang turret dan terkena damage turret.	Sesuai
3	*Enemy* Mati	*Enemy* akan mati apabila nyawa *Enemy* <= 0.	Sesuai

2. Pengujian FSM *Enemy* Moster Green

Pada Tabel 4 dapat diketahui bahwa *enemy* akan *spawn* dan berlari sampai *waypoint* akhir selama jarak *enemy* > jangkauan turret dan tidak terkena damage turret maka *enemy* masuk ke dalam *state* berlari, namun apabila jarak *enemy* <= max jangkauan serang atau area *circle collider* turret dan terkena damage turret maka *enemy* masuk *state* terluka, apabila pada *state* terluka nyawa *enemy* tersisa <= 15% maka masuk ke dalam *state* berlari cepat. Dan ketika nyawa *enemy* <= 0 maka *enemy* mati.

Tabel 4. Pengujian FSM *Enemy* Moster Green

No	State	kondisi	Hasil
1	*Enemy* Berlari	*Enemy* akan berlari sampai menuju *waypoint* akhir selama jarak *enemy* > jangkauan serang turret dan tidak terkena damage turret.	Sesuai
2	*Enemy* Terluka	*Enemy* akan terluka jika jarak *enemy* <= max jangkauan serang turret dan terkena damage turret.	Sesuai
3	*Enemy* Berlari Cepat	*Enemy* akan berlari cepat jika nyawa *enemy* tersisa <= 15%	Sesuai
4	*Enemy* Mati	*Enemy* akan mati apabila nyawa *Enemy* <= 0.	Sesuai

3. Pengujian FSM *Enemy* Moster Sword

Pada Tabel 5 dapat diketahui bahwa *enemy* akan *spawn* dan berlari sampai *waypoint* akhir selama jarak *enemy* > jangkauan turret dan tidak terkena damage turret maka *enemy* masuk ke dalam *state* berlari, namun apabila jarak *enemy* <= max jangkauan serang atau area *circle collider* turret dan terkena damage turret maka *enemy* masuk *state* terluka, apabila pada *state* terluka nyawa *enemy* tersisa <= 50% maka masuk ke dalam *state* berlari cepat. Dan ketika nyawa *enemy* <= 0 maka *enemy* mati.

Tabel 5. Pengujian FSM *Enemy* Moster Sword

No	State	kondisi	Hasil
1	*Enemy* Berlari	*Enemy* akan berlari sampai menuju *waypoint* akhir selama jarak *enemy* > jangkauan serang turret dan tidak terkena damage turret.	Sesuai
2	*Enemy* Terluka	*Enemy* akan terluka jika jarak *enemy* <= max jangkauan serang turret dan terkena damage turret.	Sesuai
3	*Enemy* Berlari Cepat	*Enemy* akan berlari cepat jika nyawa *enemy* tersisa <= 50%	Sesuai
4	*Enemy* Mati	*Enemy* akan mati apabila nyawa *Enemy* <= 0.	Sesuai

4.9. Pengujian Perangkat

Pengujian perangkat dilakukan agar dapat mengetahui spesifikasi perangkat android apa saja yang berhasil menjalankan game *military defence*, hasil dapat dilihat dalam Tabel 6 dibawah.

Tabel 6. Pengujian Perangkat

No	Nama Perangkat, Chipset dan GPU	Versi OS Android	RAM	Hasil
1	Samsung J1 Ace Spreadtrum SC9830 Mali-400MP2	5.1	1	Gagal
2	Galaxy S6 Edge+ Exynos 7420 Okta (14 nm) Mali-T760MP8	6.1	4	Berhasil
3	Oppo A71 Qualcomm SDM450 Snapdragon 450 (14 nm) Adreno 506	7.1	3	Berhasil
4	Vivo Y95 Snapdragon 439 Adreno 505	8.1	4	Berhasil
5	Oppo A3s Qualcom SDM450 Snapdragon 450 Adreno 506	8.1	3	Berhasil
6	Honor 9 Lite Hisilicon Kirin 659 Mali-T830 MP2	9	3	Berhasil
7	Samsung A30s Exynos 7904 (14 nm) Mali-G71 MP2	9	4	Berhasil
8	Samsung Galaxy S10 Snapdragon 845 Mali-G76 MP12	10	12	Berhasil
9	Redmi 9C Mediatek MT6765G Helio G35 (12 nm) PowerVR GE8320	10	4	Berhasil
10	Xiaomi Redmi Note 8 Snapdragon 665 (11 nm) Adreno 610	11	3	Berhasil
11	Oppo A92 Qualcomm snapdragon 665 Adreno 610	11	8	Berhasil
12	Xiaomi Redmi 10 Mediatek Helio G88 Mali-G52	11	6	Berhasil
13	Realme Narzo 50 A Prime Unisoc Tiger T612 Mali-G57	12	4	Berhasil
14	Xiaomi 12T Mediatek Dimensity 8100-Ultra Mali-G610 MC6	12	8	Berhasil
15	Samsung Galaxy A53 5g Exynos 1280 Mali-G68	13	8	Berhasil
16	Poco F4 Snapdragon 870 Andreno 650	13	8	Berhasil
17	Xiaomi 12 Qualcomm SM8450 Snapdragon 8 Gen 1 Adreno 730	13	8	Berhasil
18	Samsung Galaxy A52s 5g Qualcomm SM7325 Snapdragon 778G 5G Adreno 642L	13	6	Berhasil

Berdasarkan Tabel 6 menunjukkan bahwa hasil pengujian game *military defence* dibeberapa perangkat diatas berhasil dijalankan, namun terdapat satu gagal ketika dijalankan pada android versi 5.1, RAM 1 GB, Chipset Spreadtrum SC9830, dan GPU Mali-400MP2 yaitu pink screen dan hanya audio musik game. Sehingga dapat disimpulkan dalam menjalankan game *military defence* perlu dibutuhkan minimal spesifikasi android versi 6.1 Marshmallow, RAM 1 GB, chipset Snapdragon 439 atau setara, dan GPU Adreno 506 atau setara.

4.10. Pengujian Responden

Pengujian responden dilakukan untuk mengetahui kepuasan dan kemudahan pengguna dalam menjalankan aplikasi game *military defence*. Pengujian dilakukan kepada 18 orang tester dan memiliki total sepeti pada Tabel 7 dibawah.

Tabel 7. Pengujian Responden

No	Pernyataan	Penilaian		
		Sangat Setuju	Setuju	Tidak Setuju
1	tampilan grafis pada Hp anda sudah jelas?	4	13	1

2	Desain tampilan karakter, environment dan beberapa game assets menarik	4	13	1
3	Tombol atau button berfungsi semua dengan baik	3	14	1
4	Petunjuk atau tutorial game mudah dipahami	5	12	1
5	Tingkat kesulitan game meningkat saat berpindah ke stage selanjutnya	4	13	1
6	Sistem membangun, menjual, dan upgrade turret berjalan dengan baik	3	14	1
7	Turret menembak moster ketika berada dalam area jangkauannya	3	14	1
8	Muncul tanda mission complete ketika misi sudah selesai atau tercapai	6	11	1
9	Menyimpan dan menampilkan skor berupa icon bintang pada setiap stage	7	10	1
10	Nyawa moster berkurang jika terkena serangan dari turret	5	12	1
11	Coin pemain bertambah ketika membunuh moster	5	12	1
Total		**49**	**138**	**11**
Presentase (%)		**25%**	**70%**	**5%**

Berdasarkan Tabel 7 Dari orang yang memainkan game *military defence* menunjukkan bahwa 25% (total 49) menyatakan sangat setuju, 70% (total 138) menyatakan setuju, dan 5% (total 11) menyatakan tidak setuju yaitu karena salah satu perangkat user mengalami pink screen dan hanya keluar audio musik game ketika menjalankan game, oleh karena itu user tidak bisa melanjutkan pengujian game dengan baik.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dari perancangan dan implementasi pada game *Military Defence* 2D maka dapat diambil kesimpulan yaitu implementasi metode *Finite State Machine* pada NPC (*Non Player Character*) karakter *enemy* atau *Monster*, terbukti pada saat jarak *enemy* > jangkauan *turret* dan tidak terkena trigger *turret* akan masuk *state* berlari, namun jika jarak *enemy* <= max jangkauan *turret* dan terkena *damage turret* maka *enemy* masuk *state* terluka, kalau nyawa *enemy* <= 0 selama dalam *state* berlari atau terluka maka otomatis masuk *state* mati. hal ini menunjukkan perilaku cerdas npc sudah berjalan dengan baik. Pengujian responden sebanyak 18 pengguna, didapatkan hasil yang menunjukkan bahwa 25% (total 49) menyatakan sangat setuju, 70% (total 138) menyatakan setuju, dan 5% (total 11) menyatakan tidak setuju. Adapun saran sebagai acuan terhadap penelitian atau pengembangan selanjutnya, diantaranya menambahkan *enemy* dan *turret* dengan perilaku yang berbeda atau variatif tiap *stage*. *Turret* ketika di *upgrade* desain karakternya masih sama, kedepannya perlu menambahkan desain *turret* yang berbeda tiap level *upgrade*. Tidak ada waktu persiapan khusus untuk membangun *turret*, kedepannya perlu ditambahkan tombol siap tempur untuk memunculkan *enemy.*

DAFTAR PUSTAKA

[1] I. , F. J. Millington, "'Artificial Intelligence,'" 2009.

[2] I. Satrio, F. S. Wahyuni, and D. Rudhistiar, "PENERAPAN A* PATHFINDING DAN FSM (FINITE STATE MACHINE) PADA GAME 'LOST CIVILIZATION' BERBASIS ANDROID," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 2, pp. 1192–1199, 2022.

[3] M. Firdaus, "PENERAPAN METODE FINITE STATE MACHINE PADA GAME ADVENTURE 'TRAPPED MINERS,'" *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 3, no. 1, pp. 158–164, 2019.

[4] H. Sifaulloh, J. N. Fadila, and F. Nugroho, "Penerapan Metode Finite State Machine pada Game Santri on the Road," *Walisongo Journal of Information Technology*, vol. 3, no. 1, pp. 11–18, 2021.

[5] R. Bimantika, "Pengembangan Game the Galaxy Menggunakan Metode Fsm (Finite State Machine)," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 1, no. 1, pp. 180–187, 2017.

[6] E. Yulsilviana and H. Ekawati, "Penerapan metode finite state machine (FSM) pada game agent legenda anak borneo," *Sebatik*, vol. 23, no. 1, pp. 116–123, 2019.

[7] M. F. Rahadian, A. Suyatno, and S. Maharani, "Penerapan metode finite state machine pada game 'The Relationship,'" 2017.

[8] McCarthy, "Father of Artificial Intelligence, biography, LISP, arti-ficial intelligence, commonsense knowledge," *Institute of ScienceBangalore*.

[9] M. H. Alqorni, "Penerapan Metode Finite State Machine Untuk Non Player Character Pada Game Lost in Space," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 1, no. 2, pp. 111–119, 2017.

[10] I. Setiawan, "Perancangan Software Embedded System Berbasis FSM," *Semarang: Universitas Diponegoro*, 2006.