

PENERAPAN METODE FINITE STATE MACHINE PADA GAME 2D “ADVENTURE OF RUVY FOX” BERBASIS ANDROID

Iva Roudhotul Rohmah, Febriana Santi Wahyuni, Renaldi Prismaswara Prasetya

Teknik Informatika, Institut Teknologi Nasional Malang

Jalan Raya Karanglo km 2 Malang, Indonesia

2018113@scholar.itn.ac.id

ABSTRAK

Perkembangan teknologi komputer yang pesat telah mengubah industri permainan komputer, menawarkan pengalaman game yang lebih interaktif seperti *Pets Adventure*. Meskipun game ini sudah diunduh lebih dari 5000 kali, masih terdapat tantangan dalam hal respons musuh terhadap pemain dan variasi misi. Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan metode *Finite State Machine* (FSM) pada perancangan *Adventure of Ruvy Fox*, sebuah game *side-scrolling* yang menampilkan petualangan rubah melawan monster untuk menjaga keseimbangan alam. Berdasarkan perancangan dan pembuatan game *Adventure of Ruvy Fox* sukses mengimplementasikan metode *Finite State Machine* (FSM) pada karakter pemain dan musuh. Hasil pengujian menyatakan tingkat keberhasilan 100% untuk seluruh aspek permainan, termasuk pemain, musuh seperti jamur, tikus, dan landak, serta kondisi musuh Dino. Fungsi FSM pada pemain dan musuh beroperasi sesuai dengan yang diinginkan dalam berbagai situasi. Uji kegunaan dengan 27 responden menunjukkan skor *System Usability Scale* (SUS) sebesar 68,79 dengan kategori “Ok” atau C, menandakan bahwa game ini dapat dianggap dapat digunakan dengan baik oleh pemain. Pengujian *black box* juga menunjukkan keberhasilan 100% untuk semua menu yang ada. Pengembangan yang disarankan adalah menambah fitur *multiplayer*, animasi pada layar *cutscene*, dan pemanfaatan koin dalam permainan menjadi potensi tambahan yang dapat meningkatkan daya tarik dan pengalaman bermain game secara keseluruhan.

Kata kunci : *finite state machine, game pets adventure, adventure of ruvy fox, usability scale, kecerdasan buatan game*

1. PENDAHULUAN

Perkembangan teknologi yang pesat, terutama di bidang komputasi, memberikan dampak signifikan pada industri permainan komputer. Permainan tidak hanya berkembang dalam *genre*, tapi juga berubah dari permainan yang dulu ke permainan yang menggunakan teknologi. Hal ini terjadi karena sekarang orang banyak yang biasa menggunakan *gadget* [1].

Game *Pets Adventure 2D Platformer* adalah permainan *side scrolling* di mana pemain memandu seekor kucing untuk mengumpulkan tiga bintang. Terdapat bendera sebagai titik penyelamatan. Jika pemain mati, dia akan *respawn* di lokasi terakhir kali dia membuka bendera. Walaupun game *Pets Adventure 2D Platformer* berhasil mendapatkan rating ketiga di *Google Play Store* dan telah diunduh lebih dari 5000 kali, masih terdapat beberapa masalah. Salah satunya adalah musuh tidak melakukan tindakan apa pun saat pemain menyerangnya. Untuk meningkatkan pengalaman bermain game, pemain diharuskan berperilaku seperti manusia, seperti melompat, jatuh, berlari, dan mati.

Finite State Machine (FSM) dipilih untuk mengatasi masalah ini karena kemampuannya dalam mendukung pengambilan keputusan atau operasi melalui perancangan sistem kendali. Sistem ini menggambarkan perilaku atau prinsip operasi dengan menggunakan tiga elemen dasar yaitu keadaan, peristiwa dan tindakan [2].

Berdasarkan penjelasan di atas, penulis berencana untuk mengembangkan sebuah game petualangan berjudul “*Adventure of Ruvy Fox*”, untuk platform *Android* yang mengimplementasikan *Finite State Machine* (FSM) yang bertindak terhadap pemain dan karakter musuh serta memberikan kecerdasan buatan kepada pemain menggunakan *Unity 2D*. Game “*Adventure of Ruvy Fox*” menceritakan petualangan seekor rubah yang berusaha menjaga keseimbangan alam dengan melawan monster-monster perusak lingkungan

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Menurut [3] yang merancang “Desain *Serious Game* Sosialisasi Bencana Berbasis Model Teori Aktivitas”, yang bertujuan untuk merancang game dengan tujuan sosialisasi bencana menggunakan *Finite State Machine* pada NPC. Hasil pengujian menunjukkan bahwa metode *Finite State Machine* untuk karakter musuh sudah sesuai dengan model yang dibutuhkan. Kekurangan dalam penelitian ini yaitu pengembangan lebih lanjut adalah game multi pemain dan dikembangkan untuk platform *Android*.

Berdasarkan penelitian [4], “Rancang Bangun *Game the Farmer Feed Animals* Menggunakan Metode *Finite State Machine*”, dibuatlah sebuah game dengan metode FSM untuk mempercepat strategi pencarian makanan bagi hewan. Pengujian menunjukkan bahwa penggunaan metode *Finite State Machine* tidak berhasil karena musuh tidak mengejar

pemain. Kelemahan penelitian ini yang perlu dikembangkan lebih lanjut adalah game dapat dikembangkan berbasis *mobile*, game dapat dimainkan secara *multiplayer*.

Dalam penelitiannya [5], game petualangan “Castle of Illusion” mencoba menerapkan *Finite State Machine* pada karakter NPC, termasuk musuh dan bos. Hasil pengujian metode FSM dengan karakter NPC dapat mengidentifikasi dan menyerang pemain. Kelemahan pada penelitian ini yang perlu dikembangkan lebih lanjut antara lain penambahan fitur-fitur menarik, kemampuan bermain di berbagai platform dan pengembangan, sistem skor *leadboard*, penyimpanan game dan kemampuan *multiplayer*.

Penelitian [6] dengan judul “Pembuatan Game 2D Escape Plan Dengan Metode *Finite State Machine*” yang memiliki tujuan untuk membuat game simulasi *escape* dengan menggunakan *Finite State Machine* pada musuh. Hasil pengujian level 1, 2, dan 3 menunjukkan penerapan metode FSM pada karakter musuh berhasil. Namun penelitian ini mengungkapkan beberapa kekurangan seperti, perlunya pengembangan game berbasis *Android*, kurangnya audio dalam game, dan informasi yang tidak jelas tentang nyawa pemain.

2.2. Game

Game adalah salah satu bentuk hiburan yang sedang berkembang dengan cepat untuk saat ini, dan bermain game adalah cara yang populer untuk mengisi waktu senggang. Inovasi dalam pengembangan game terus meningkat, termasuk pengembangan karakter non-pemain (NPC) seperti *Finite State Machine* dalam menghadapi perilaku karakter musuh [7].

2.3. Jenis-Jenis Game

Menurut [8] jenis permainan dapat dibagi menjadi beberapa kategori berikut:

- Game aksi atau *action* menekankan pada reaksi cepat dan keterampilan refleks pemainnya.
- Petualangan merupakan *genre* permainan yang dibangun berdasarkan cerita. Biasanya ketika suatu peristiwa dimainkan, pemain tidak memiliki kemampuan untuk kembali ke peristiwa sebelumnya.
- RPG (*Role Playing Game*) merupakan salah satu jenis permainan yang berfokus pada pengembangan karakter yang dimainkan oleh pema
- Game simulasi berfokus pada simulasi dunia nyata.
- Side Scrolling* adalah genre permainan di mana pemain bergerak ke satu arah selama bermain game untuk menyelesaikan suatu tugas.

2.4. Kecerdasan Buatan

Kecerdasan buatan adalah studi dan pemodelan bagaimana proses berpikir manusia dan dibuat desain mesin untuk meniru perilaku dan cara berpikir manusia [9]. Ada beberapa jenis kecerdasan buatan yang digunakan dalam pengembangan game, yaitu:

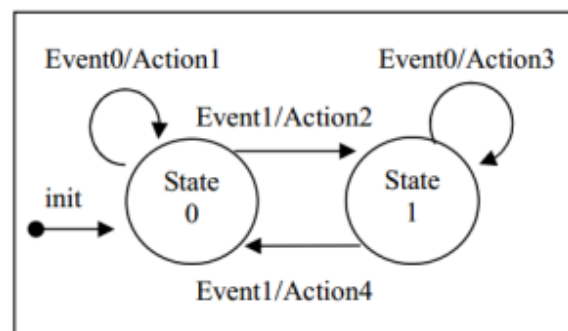
2.4.1. Decision Making

1. Fuzzy Logic

Pada logika *fuzzy* memperbolehkan nilai anggota antara 0 dan 1. Proses *fuzzy* terdiri dari *fuzzifikasi* masukan, operasi logika *fuzzy*, implikasi, agregasi, dan *defuzzifikasi*.

2. Finite State Machine

Finite State Machine (FSM) adalah suatu desain sistem yang memperlihatkan bagaimana suatu sistem berperilaku atau beroperasi dengan memanfaatkan tiga elemen utama, yaitu keadaan, peristiwa, dan tindakan [2]. Contoh diagram FSM ditunjukkan pada Gambar 1.



Gambar 1. Diagram fsm [10]

Pada diagram *Finite State Machine* menunjukkan 2 keadaan dan 2 masukan yang menghasilkan 4 tindakan keluaran berbeda. Pada saat sistem dijalankan, sistem menuju ke keadaan 0, dimana pada keadaan ini, sistem melakukan tindakan 1, jika terjadi masukan kejadian 0, kemudian ketika terjadi peristiwa, melakukan tindakan 2, maka sistem berubah ke keadaan 1.

2.4.2. Pathfinding

1. Algoritma A*

A* merupakan salah satu algoritma pencarian jalur terbaik untuk mencari jalur terpendek dari titik awal hingga titik akhir, sehingga total biaya jalur tersebut dapat diminimalkan [11].

2. Breadth-First Search

BFS atau *Breadth-First Search* adalah algoritma untuk mencari grafik yang mendasar. Algoritma ini bermanfaat dalam menemukan rute terpendek dalam grafik yang tidak memiliki nilai bobot atau biaya [12].

3. Dijkstra

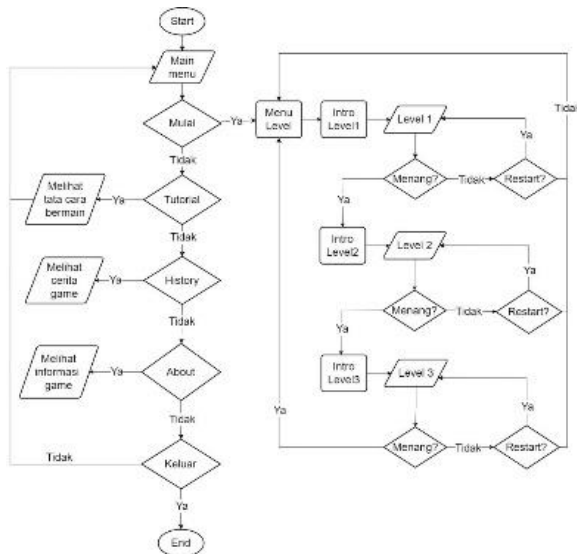
Algoritma ini mengkomputasi jarak selama proses traversal pada saat mencapai suatu *node* dan membentuk jalur terpendek dari *node* awal ke *node* tujuan dengan melalui proses eksplorasi yang disebut relaksasi *edge* [12].

2.5. System Usability Scale

System Usability Scale (SUS) merupakan suatu metode pengujian antarmuka yang melibatkan langsung partisipasi pengguna akhir. Keunggulan SUS terletak pada lebih menekankan pada pengalaman pengguna saat pengujian, sehingga hasil evaluasi lebih mencerminkan kondisi dunia nyata. Sebagai alat ukur, tes SUS terdiri dari 10 pernyataan [13].

3. METODE PENELITIAN

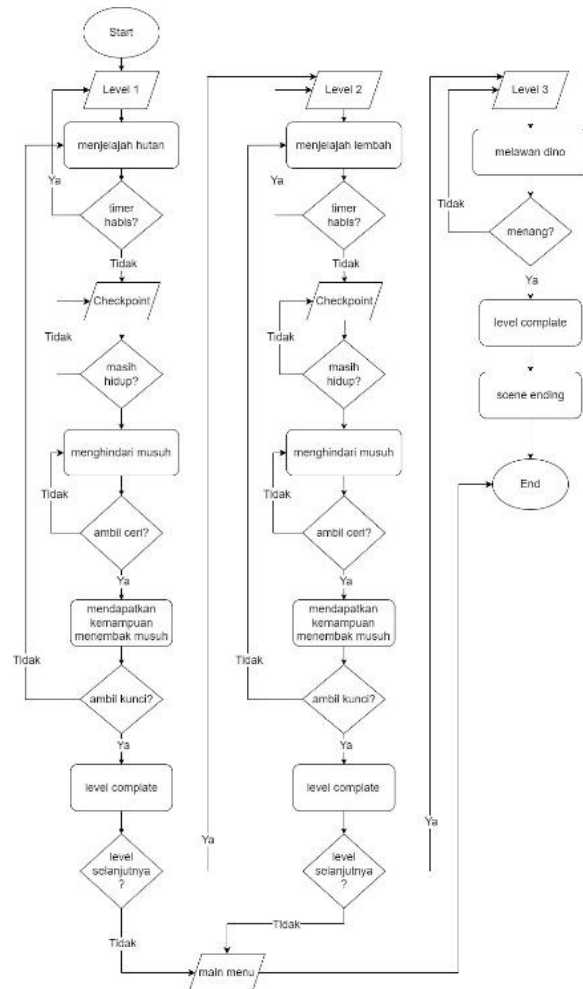
3.1. Flowchart Menu



Gambar 2. Flowchart menu

Proses permainan dimulai dari menu utama yang meliputi kondisi *start*, *tutorial*, *history*, *about* dan *exit*. Dengan memilih *start*, pemain akan memulai menu level, lalu pilih level 1, setelah memilih level 1 maka akan muncul adegan *intro* level 1, lalu lanjut ke level 1, jika level 1 sudah selesai, maka ke level berikutnya, jika sudah gagal, pemain harus mengulang level atau kembali ke menu level. Jika pemain memilih *tutorial*, prosedur yang terkait dengan permainan akan muncul. Jika pemain memilih *history*, pemain dapat melihat sejarah permainan. Apabila pemain memilih opsi "*about*", data mengenai pembuat game akan muncul. Sebaliknya, jika pemain memilih untuk keluar, aplikasi akan mengarahkannya kembali ke beranda *Android*, sebagaimana yang terlihat pada Gambar 2.

3.2. Flowchart Sistem Game



Gambar 3. Flowchart sistem game

Gambar 3 menunjukkan *flowchart* sistem permainan yang dimulai dari level 1. Di Level 1, pemain menjelajahi hutan. Saat menjelajah, pemain diberikan waktu untuk menyelesaikan level. Ketika waktu berakhir, level dimulai dari awal. Untuk membantu, pemain diberikan lebih banyak pos pemeriksaan untuk *respawn* ketika nyawa mereka hampir habis. Selama permainan, pemain mendapatkan kekuatan berupa buah ceri, sehingga pemain bisa membunuh musuh. Untuk menyelesaikan level, pemain perlu mendapatkan kunci untuk membuka gerbang ke pintu *level complete*. Setelah menyelesaikan level, pemain diberikan pilihan untuk memutar ulang level, melanjutkan level, atau memilih level. Jika pemain memilih level selanjutnya maka akan lanjut ke level selanjutnya yaitu level 2. Pada

level ini alurnya sama seperti pada level 1. Kemudian pada level terakhir pemain harus mengalahkan boss tersebut untuk bisa menang.

3.3. Perancangan Karakter

Pada Tabel 1. menunjukkan gambar karakter pemain yang digunakan dalam *Adventure of Ruvy Fox*.

Tabel 1. Rancangan karakter

No	Pemain	Keterangan
1		Ruvy sebagai karakter pemain memiliki 5 nyawa pada setiap levelnya.

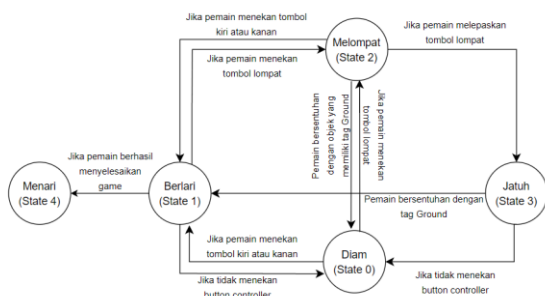
3.4. Perancangan Karakter Enemy

Tabel 2 menunjukkan desain musuh yang digunakan dalam *Adventure of Ruvy Fox*.

Tabel 2. Rancangan musuh

No	Enemy	Keterangan
1		Jamur - Kerusakan serangan 0.25 - Total nyawa 4
2		Tikus - Kerusakan serangan 0.5 - Total nyawa 5
3		Landak - Kerusakan serangan 0.5 - Total Nyawa 7
4		Dino - Kerusakan serangan 1 - Total nyawa 10

3.5. Rancangan FSM Pada Player

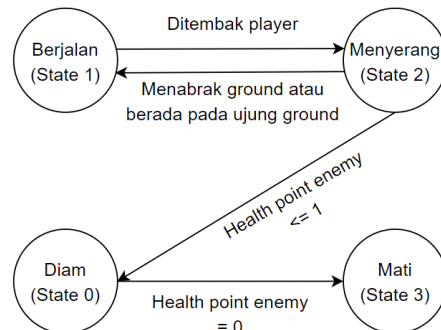


Gambar 4. Diagram finite state machine pada player

Diagram FSM pada Gambar 4 diterapkan pada karakter pemain. Awalnya, pemain secara *default* berada pada kondisi diam. Ketika pemain menekan tombol kanan atau kiri, pemain beralih ke kondisi berjalan. Saat tombol lompat ditekan, pemain akan masuk ke keadaan lompat, ketika ke bawah akan dalam keadaan jatuh, dan jika pemain tidak menekan tombol melompat maka animasi akan berubah ke keadaan jatuh. Pemain akan kembali ke animasi diam. Ketika pemain berhasil menyelesaikan permainan

dengan memperoleh *Harmony Key*, maka pemain memasuki keadaan menari, dimana tandanya mereka telah berhasil menyelesaikan permainan.

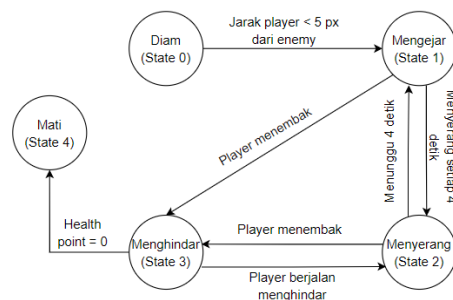
3.6. Rancangan FSM Pada Musuh Jamur, Tikus, Landak



Gambar 5. Finite state machine musuh jamur, tikus, landak

Diagram FSM pada Gambar 5. dapat dilihat penggambaran perilaku untuk musuh jamur, tikus, dan landak. Pada awalnya musuh berada pada *state* berjalan dengan jarak *move distance* yang di atur pada *inspector unity*. Ketika pemain menembak musuh, musuh akan masuk ke *state* menyerang. Jika musuh menyentuh ujung tanah, dia akan kembali ke *state* berjalan. Jika *health* musuh kurang dari atau sama dengan 1 maka musuh akan masuk *state* diam. Dan jika *health* musuh mencapai 0 maka akan masuk ke *state* mati.

3.7. Rancangan FSM Pada Musuh Dino



Gambar 6. Finite state machine pada musuh dino

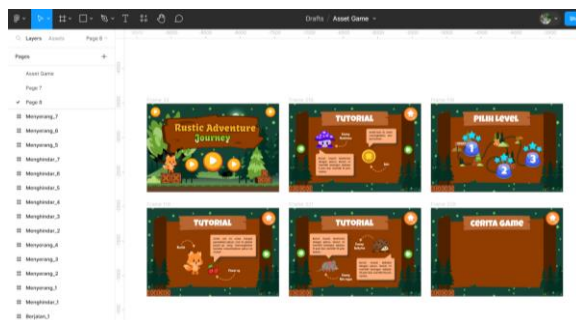
Diagram FSM pada Gambar 6. digunakan untuk musuh Dino. Secara *default*, musuh dalam keadaan diam. Kemudian ketika pemain mendekati musuh dalam jarak 5px, musuh akan mengejar pemain tersebut dan musuh akan terus mengejar pemain tersebut. Dino kemudian akan masuk ke kondisi menyerang setiap 4 detik dan kembali ke posisi mengejar sambil menunggu serangan berikutnya. Kemudian dino akan masuk ke keadaan menghindari jika pemain menembak dino, namun jika tidak ada dino yang tertembak maka dino akan kembali ke keadaan menyerang. Ketika musuh kehabisan nyawa, dino tersebut masuk ke dalam keadaan mati.

4. HASIL DAN PEMBAHASAN

4.1. Pembuatan Menu Asset

Pembuatan karakter dan aset dalam game menggunakan aplikasi Figma dengan menggambar tangan di aplikasi menggunakan alat bentuk aplikasi, seperti *pen tool*, *rectangle tool*, *line tool*, *ellipse tool*, dan alat lainnya.

Item menu dibuat dengan ukuran bingkai 1920x1280 piksel. Item menu ini dibuat dengan *rectangle tool* untuk objek persegi atau persegi panjang, *Ellipse tool* untuk objek melingkar, dan *pen tool* untuk objek berbentuk khusus, seperti objek melengkung. Item menu dapat dilihat pada Gambar 7.



Gambar 7. Pembuatan asset menu

4.2. Perancangan Main Menu

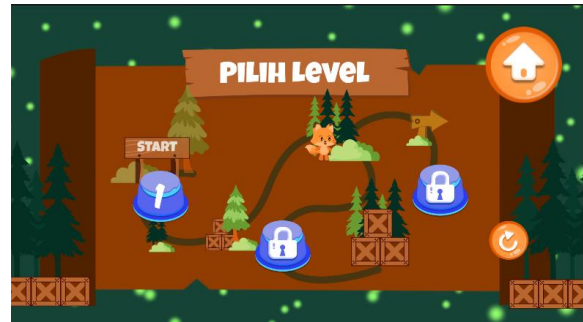
Terdapat lima tombol pada menu utama seperti yang ada pada Gambar 8. yaitu tombol *play*, *tutorial*, *history*, *about*, dan *exit*. Tombol ini memudahkan untuk mengakses fitur-fitur utama seperti meluncurkan game, mempelajari game melalui tutorial, menjelajahi sejarah game, mendapatkan informasi tentang pembuat game, atau keluar dari aplikasi.



Gambar 8. Implementasi main menu

4.3. Perancangan Menu Pilih Level

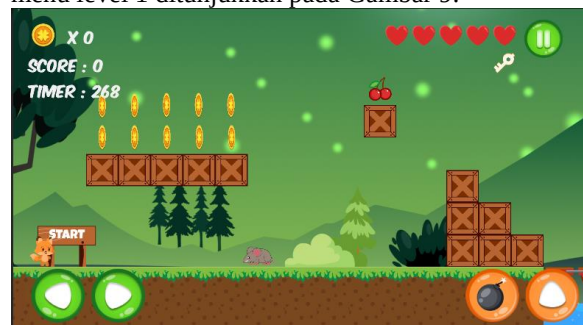
Tata letak pemilihan level ditunjukkan pada antarmuka pengguna pemilihan level yang ditunjukkan pada Gambar 9. Pada menu pemilihan level, pemain harus berhasil menyelesaikan setiap level satu demi satu, jika pemain belum menyelesaikan level awal, mereka tidak dapat melanjutkan ke level tingkat berikutnya.



Gambar 9. Implementasi menu pilih level

4.4. Perancangan Menu Level 1

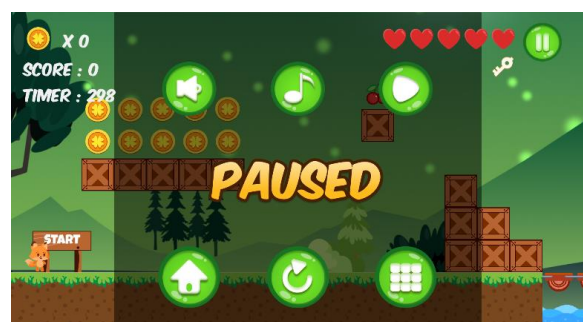
Di level awal permainan, pemain menghadapi berbagai rintangan seperti tanah, air, jamur, dan tikus. Mereka harus menghindari jebakan api dan ikan sambil mengumpulkan kunci untuk membuka level berikutnya. Terdapat tombol jeda untuk menghentikan permainan sementara. Pemain memiliki 5 kesehatan. Musuh terdiri dari 8 jamur dengan *damage* serangan 0,25 dan 5 tikus dengan *damage* serangan 0,5. Pemain perlu mengumpulkan 50 koin, mengambil kekuatan peluru untuk melawan musuh dan kunci untuk membuka pintu untuk menyelesaikan level. Desain menu level 1 ditunjukkan pada Gambar 9.



Gambar 10. Implementasi menu level 1

4.5. Perancangan Menu Pause

Tujuan dari tombol jeda adalah untuk memungkinkan pemain menghentikan sementara permainan. Saat tombol jeda ditekan, permainan akan dijeda dan *popup* jeda akan muncul, menawarkan berbagai opsi tombol seperti pada Gambar 11.

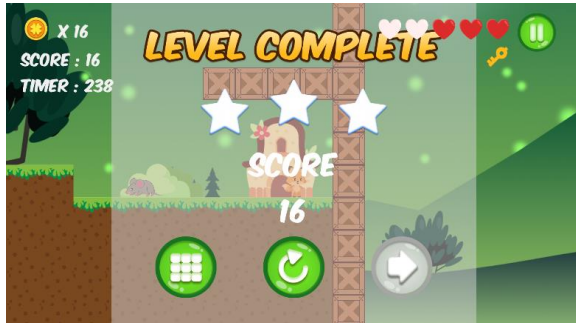


Gambar 11. Implementasi menu pause

4.6. Perancangan Menu Level Complete

Setelah menyelesaikan level, pengguna akan dibawa ke menu skor. Tombol pilih level

memungkinkan pemain untuk kembali ke menu level dan menjelajahi opsi *gameplay* lainnya. Tombol *restart* memungkinkan pemain untuk memulai kembali permainan dari awal, memberi mereka pilihan untuk meningkatkan skor atau mencoba strategi baru. Tombol *next* memungkinkan pemain untuk ke level berikutnya. Desain *level complete* dapat dilihat pada Gambar 12.



Gambar 12. Implementasi level complete

4.7. Pengujian Metode Finite State Machine

Pengujian metode yang digunakan untuk mengatur tingkah laku karakter pemain dan musuh diuji, dan data hasilnya dicatat dalam tabel berikut.

4.7.1. Pengujian FSM Player

Tabel 3. menunjukkan hasil pengujian metode *Finite State Machine* pada *player*.

Tabel 3. Pengujian finite state machine player

No	State	Kondisi	Hasil
1	Player Diam	Player akan menjalankan kondisi diam ketika tidak diaktifkan dengan tombol gerakan.	Sesuai
2	Player Berlari	Player akan berlari sesuai tombol yang digunakan untuk berjalan ke kanan dan ke kiri.	Sesuai
3	Player Melompat	Player akan melompat jika tombol untuk melompat ditekan.	Sesuai
4	Player Jatuh	Player akan menampilkan animasi terjatuh saat selesai melakukan sebuah lompatan	Sesuai
5	Player Menari	Player akan menampilkan animasi menari ketika bertemu dengan manusia.	Sesuai

4.7.2. Pengujian FSM Enemy Jamur, Tikus, dan Landak

Tabel 4. menunjukkan hasil pengujian metode FSM pada jamur, tikus dan landak. Saat musuh berjalan, sistem bekerja dengan baik sehingga musuh bergerak sesuai jarak yang ditentukan. Saat musuh menyerang, musuh bereaksi terhadap tembakan pemain. Musuh akan diam ketika nyawa mereka kurang dari atau sama dengan satu. Ketika musuh mati,

sistem berhasil membunuh musuh ketika nyawa mencapai nol.

Tabel 4. Pengujian *fsm* musuh jamur, tikus, landak

No	State	Kondisi	Hasil
1	Enemy Berjalan	Enemy akan berjalan sesuai dengan jarak pergerakan yang diatur pada <i>inspector unity</i> .	Sesuai
2	Enemy Menyerang	Enemy akan menyerang jika terkena <i>power bullet</i> yang dikeluarkan oleh <i>player</i> .	Sesuai
3	Enemy Diam	Enemy akan diam jika nyawa <i>enemy</i> ≤ 1 .	Sesuai
4	Enemy Mati	Enemy akan mati jika nyawa <i>enemy</i> = 0.	Sesuai

4.7.3. Pengujian FSM Dino

Tabel 5. memberikan gambaran pengujian FSM pada Dino. Ketika musuh diam, sistem berhasil membunkam musuh, sesuai dengan bagaimana musuh tetap diam jika belum mendekati pemain. Sistem akan membuat musuh akan bergerak mendekati pemain jika pemain berada dalam jarak 5 piksel. Jika terjadi serangan musuh, sistem dapat melakukan serangan dengan tembakan setiap 4 detik, sesuai kondisi yang ditentukan. Selain itu, dalam keadaan menghindari musuh, sistem berhasil menghindari tembakan pemain setiap kali pemain menembak. Musuh mati ketika nyawanya mencapai nol.

Tabel 5. Pengujian *finite state machine* musuh dino

No	State	Kondisi	Hasil
1	Enemy Diam	Enemy akan diam pada state awal apabila belum didekati player	Sesuai
2	Enemy Berjalan	Enemy akan berjalan mendekati player saat player mendekati dino dalam jarak kurang dari 5px.	Sesuai
3	Enemy Menyerang	Enemy akan menyerang setiap 4 detik sekali dengan mengeluarkan tembakan.	Sesuai
4	Enemy Menghindar	Enemy akan menghindari tembakan player setiap kali player menembak.	Sesuai
5	Enemy Mati	Enemy akan mati jika nyawa <i>enemy</i> = 0.	Sesuai

4.8. Pengujian Usability

Aplikasi ini diuji menggunakan metode *System Usability Scale* (SUS) dengan cara menyebarkan Google Form dan melakukan pengujian langsung kepada pengguna. Responden diminta memberikan pendapat subjektifnya melalui 10 pertanyaan standar dengan menggunakan skala Likert 5 poin, seperti disajikan pada Tabel 6.

Tabel 6. Daftar pertanyaan kuesioner

No	Pertanyaan	Kode
1	Sepertinya saya sering memainkan game ini	K1
2	Menurut saya game ini sulit dimengerti	K2
3	Game ini sepertinya mudah dimainkan	K3
4	Jika ada masalah, saya mungkin akan meminta bantuan	K4
5	Menurut saya game ini berfungsi dengan baik	K5
6	Menurut saya ada beberapa hal di game ini yang tidak sama	K6
7	Saya yakin orang lain akan cepat memahami cara memainkan game ini	K7
8	Saya sedikit bingung cara memainkan game ini	K8

9	Saya berpikir ketika saya memainkan ini permainan permainan tidak ada kendala yang berarti	K9
10	Saya mungkin harus mengajari cara memainkan permainan ini lebih awal	K10

Data kuesioner yang didapatkan akan dihitung menggunakan skala kegunaan sistem. Untuk setiap pertanyaan yang bernomor ganjil (1, 3, 5, 7 dan 9), rentang jawaban responden akan dikurangi sebesar 1. Sedangkan untuk pertanyaan yang bernomor genap (2, 4, 6, 8 dan 10), nilai 5 akan dikurangi dengan rentang jawaban responden. Hasil skor SUS disajikan pada Tabel 7.

Tabel 7. Hasil perhitungan skor system usability scale

R	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	Skor
R1	4-1	5-3	4-1	5-2	4-1	5-2	5-1	5-3	4-1	5 - 4	27
R2	4-1	5-3	4-1	5-3	4-1	5-3	5-1	5-3	4-1	5 - 3	26
R3	4-1	5-2	4-1	5-2	4-1	5-2	3-1	5-2	4-1	5 - 3	28
R4	2-1	5-4	4-1	5-4	1-1	5-4	4-1	5-4	4-1	5 - 4	15
R5	2-1	5-4	2-1	5-4	4-1	5-4	2-1	5-4	4-1	5 - 4	14
R6	2-1	5-2	2-1	5-4	4-1	5-4	2-1	5-5	2-1	5 - 4	13
R7	2-1	5-4	2-1	5-4	4-1	5-4	4-1	5-2	4-1	5 - 2	20
R8	4-1	5-2	4-1	5-3	4-1	5-3	5-1	5-3	4-1	5 - 2	28
R9	2-1	5-2	2-1	5-4	4-1	5-2	2-1	5-2	3-1	5 - 4	19
R10	4-1	5-2	4-1	5-2	4-1	5-3	4-1	5-2	4-1	5 - 4	27
R11	4-1	5-4	4-1	5-3	4-1	5-2	4-1	5-2	4-1	5 - 2	27
R12	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5 - 1	40
R13	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5 - 1	40
R14	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5 - 3	38
R15	3-1	5-1	5-1	5-2	3-1	5-1	5-1	5-1	4-1	5 - 1	34
R16	1-1	5-3	3-1	5-1	3-1	5-1	4-1	5-2	3-1	5 - 3	24
R17	3-1	5-3	4-1	5-5	4-1	5-1	4-1	5-2	3-1	5 - 4	23
R18	4-1	5-1	5-1	5-4	5-1	5-1	4-1	5-1	4-1	5 - 4	31
R19	5-1	5-1	5-1	5-1	5-1	5-5	4-1	5-1	5-1	5 - 4	32
R20	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5-1	5 - 1	40
R21	3-1	5-1	5-1	5-1	4-1	5-2	5-1	5-1	4-1	5 - 1	35
R22	4-1	5-2	4-1	5-1	4-1	5-2	4-1	5-2	3-1	5 - 2	30
R23	4-1	5-2	4-1	5-2	4-1	5-2	4-1	5-2	4-1	5 - 4	28
R24	3-1	5-2	4-1	5-2	5-1	5-3	4-1	5-2	4-1	5 - 4	27
R25	4-1	5-3	2-1	5-3	4-1	5-4	4-1	5-2	3-1	5 - 5	20
R26	3-1	5-4	5-1	5-2	5-1	5-1	5-1	5-2	5-1	5 - 3	31
R27	3-1	5-2	4-1	5-2	5-1	5-2	5-1	5-2	2-1	5 - 5	26

Langkah selanjutnya adalah skor SUS dari Tabel 7. dikalikan dengan 2.5. Kemudian proses selanjutnya adalah menjumlahkan hasil perkaliannya. Pada langkah terakhir, hasil perkalian dijumlahkan dan dihitung nilai rata-ratanya [14]. Pada Tabel 8. Dapat dilihat hasil perkalian skor *System Usability Scale* (SUS).

Tabel 8. Hasil perkalian system usability scale

Responden	Skor SUS x 2.5	Total
R1	27x2.5	67,5
R2	26x2.5	65
R3	28x2.5	70
R4	15x2.5	37,5
R5	14x2.5	35
R6	13x2.5	32,5

Responden	Skor SUS x 2.5	Total
R7	20x2.5	50
R8	28x2.5	70
R9	19x2.5	47,5
R10	27x2.5	67,5
R11	27x2.5	67,5
R12	40x2.5	100
R13	40x2.5	100
R14	38x2.5	95
R15	34x2.5	85
R16	24x2.5	60
R17	23x2.5	57,5
R18	31x2.5	77,5
R19	32x2.5	80
R20	40x2.5	100
R21	35x2.5	87,5
R22	30x2.5	75

Responden	Skor SUS x 2.5	Total
R23	28x2.5	70
R24	27x2.5	67,5
R25	20x2.5	50
R26	31x2.5	77,5
R27	26x2,5	65
Jumlah		1857.5

Langkah selanjutnya menghitung rata-rata skor SUS responden dengan cara menjumlahkan skor SUS seluruh responden dan membagi hasilnya dengan jumlah responden. Dengan cara ini diperoleh skor rata-rata seperti dijelaskan pada Persamaan (1).

$$\bar{x} = \frac{\sum x}{n} \quad (1)$$

Keterangan:

$\bar{x}$ = skor rata-rata SUS

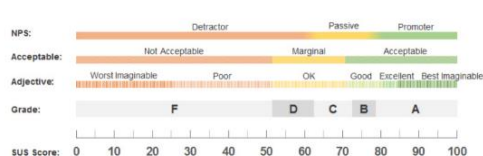
$\sum x$ = jumlah skor SUS

n = jumlah responden

Total skor SUS pada pengujian ini adalah 1857,5 seperti terlihat pada Tabel 8 dengan jumlah responden 27 orang. Berdasarkan Persamaan (1), rata-rata skor SUS dihitung sebagai berikut:

$$\bar{x} = \frac{\sum x}{n} = \frac{1857,5}{27} = 68,79$$

Dengan demikian hasil perhitungan menunjukkan skor rata-rata sebesar 68,79, maka dapat disimpulkan bahwa kegunaan game *Adventure of Ruvy Fox* termasuk dalam kategori Ok, dimana aplikasi masih cukup baik. Dari rata-rata skor yang diperoleh, hasilnya dimasukkan ke dalam skala C dan diperoleh kelompok Pasif dalam penilaian NPS seperti terlihat pada Gambar 13. Berdasarkan hasil tersebut, kualitas aplikasi dari segi kegunaan mendapat penilaian cukup baik.



Gambar 13. Penentuan hasil system usability scale [13]

5. KESIMPULAN DAN SARAN

Berdasarkan perancangan dan pembuatan game *Adventure of Ruvy Fox* sukses mengimplementasikan metode *Finite State Machine* (FSM) pada karakter pemain dan musuh, menciptakan perilaku cerdas yang baik. Hasil pengujian menyatakan tingkat keberhasilan 100% untuk seluruh aspek permainan, termasuk pemain, musuh seperti jamur, tikus, dan landak, serta kondisi musuh Dino. Fungsi FSM pada pemain dan musuh beroperasi sesuai dengan yang diinginkan dalam berbagai situasi. Uji kegunaan dengan 27 responden menunjukkan skor *System Usability Scale* (SUS) sebesar 68,79 dengan kategori “Ok” atau C, menandakan bahwa game ini dapat dianggap dapat digunakan dengan baik oleh pemain. Pengujian *black box* juga menunjukkan keberhasilan 100% untuk semua menu yang ada, menambah kehandalan keseluruhan pengalaman bermain game.

Pengembangan yang disarankan adalah menambah fitur *multiplayer*, animasi pada layar *cutscene*, dan pemanfaatan koin dalam permainan menjadi potensi tambahan yang dapat meningkatkan daya tarik dan pengalaman bermain game secara keseluruhan.

DAFTAR PUSTAKA

- [1] D. S. Hormansyah, M. Astiningrum dan F. A. Asyraq, “Implementasi FSM (Finite State Machine) Pada Game Surabaya Membara,” *JIP (Jurnal Informatika Polinema)*, vol. 6, no. 2, pp. 11-17, 2020.
- [2] 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.
- [3] F. Nugroho, E. M. Yuniarno dan M. Hariadi, “Desain Serious Game Sosialisasi Bencana Berbasis Model Teori Aktifitas,” *Jurnal Mnemonic*, vol. 2, no. 1, pp. 59-66, 2019.
- [4] Y. W. Ramadan, “Rancang Bangun Game The Farmer Feed Animals Menggunakan Metode Finite State Machine,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 3, no. 2, pp. 120-125, 2019.
- [5] A. Badruddin, “Perancangan Dan Implementasi Finite State Machine Pada Game Castle Of Illusion,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 3, no. 1, pp. 345-350, 2019.
- [6] A. M. Rumakey, J. D. Irawan dan A. Wahid, “Pembuatan Game 2d Escape Plan Dengan Metode Finite State Machine,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 4, no. 2, pp. 65-72, 2020.
- [7] D. Herumurti, I. Kuswardayan, A. A. Yunanto, G. A. Senna dan S. Arifiani, “Development of First-Person Shooter Game with Survival Maze Based on Virtual Reality,” *Information Technology International Seminar (ITIS)*, vol. 14, no. 16, pp. 81-86, 2020.
- [8] J. B. Cakra, *Pembuatan Game Bergenre Side Scroller 2.5d Bertemakan Cerita Rakyat Sawunggaling Berjudul The Legend Of Sawunggaling*, Surabaya: Institut Bisnis Dan Informatika STIKOM, 2018.
- [9] McCarthy, *Father of Artificial Intelligence, biography, LISP, arti-ficial intelligence, commonsense knowledge*, Institute of ScienceBangalore.
- [10] I. Setiawan, *Perancangan Software Embedded System Berbasis FSM*, Semarang: Universitas Diponegoro, 2006.
- [11] I. Satrio, F. S. Wahyuni dan 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.
- [12] N. Sugianti, A. Mardhiyah dan N. R. Fadlilah, “Komparasi Kinerja Algoritma BFS, Dijkstra, Greedy BFS, dan A* dalam Melakukan Pathfinding,” (*JISKa Jurnal Informatika Sunan Kalijaga*), vol. 5, no. 3, p. 194 – 205, 2020.
- [13] G. Ramadhan, Evaluasi Usability Pada Aplikasi Talent Menggunakan System Usability Scale (SUS) Di Astra Credit Companies Jambi, Jambi: Universitas Jambi, 2023.
- [14] U. Ependi, A. Putra dan F. Panjaitan, “Evaluasi tingkat kebergunaan aplikasi Administrasi Penduduk menggunakan teknik System Usability Scale,” *Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 5, no. 1, pp. 63-76, 2019.