

PENERAPAN FINITE STATE MACHINE PADA GAME “TIMUN MAS” BERBASIS ANDROID

Rangga Pradana^{1*}, Dewi Kusumaningsih²

^{1,2}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta Selatan, Indonesia

Email: ^{1*}pradanarangga11@gmail.com, ²dewi.kusumaningsih@budiluhur.ac.id
(* : corresponding author)

Abstrak-Penelitian ini bertujuan untuk membantu mengenalkan cerita rakyat “Timun Mas” kepada anak muda zaman sekarang di rentang usia 15 sampai 25 tahun dengan cara yang menyenangkan, yaitu mengenalkan cerita tersebut kedalam bentuk permainan ringan berbasis android. Permainan Timun Mas ini dibangun dengan unity engine menggunakan Algoritme *Finite State Machine* (FSM). Algoritma FSM ini diterapkan kepada musuh dari Timun Mas agar musuh dapat bertindak sesuai tingkah laku Pemain. Hasil dari penelitian ini menunjukkan bahwa Algoritma FSM yang diberikan kepada musuh dan juga bos, dapat mendeteksi posisi Pemain dan memberikan *output* yang sesuai berupa mengikuti Pemain dan menyerang Pemain saat Pemain berada di jarak yang telah ditentukan. Cerita dan mekanisme dalam game dapat berjalan sesuai keinginan dengan hasil pengujian pada penelitian ini mendapatkan nilai sebesar 84.12% dengan skala sangat baik.

Kata Kunci: Cerita Rakyat, Game, Finite State Machine, Adventure, Unity

IMPLEMENTATION OF FINITE STATE MACHINE IN THE ANDROID-BASED “TIMUN MAS” GAME

Abstract-This research aims to help introduce the folklore “Timun Mas” to today's youth in the age range of 15 to 25 years in a fun way, namely introducing the story in the form of light games based on Android. This Timun Mas game was built with the unity engine using the finite state machine (FSM) algorithm. This FSM algorithm is applied to enemies from Timun Mas so that enemies can act according to Pemin behavior. The results of this study indicate that the FSM algorithm given to enemies and also bosses can detect the Pemin's position and provide the appropriate output in the form of following the Pemin and attacking the Pemin when the Pemin is at a predetermined distance. The story and mechanism in the game can run as desired with the test results in this study getting a score of 84.12% with a very good scale.

Keywords: Folklore, Games, Finite State Machine, Adventure, Unity

1. PENDAHULUAN

Game merupakan salah satu industri yang besar saat ini, game dapat menjadi alternatif mengisi waktu luang dan menjadi sarana hiburan bagi anak-anak maupun orang dewasa. *Game* umumnya dibagi menjadi 2 jenis berdasarkan banyaknya pemain atau player, yaitu *single-player* adalah jenis *game* yang hanya dapat dimainkan oleh satu pemain saja, dan *multi-player* adalah *game* yang dapat dimainkan oleh dua orang atau lebih pemain[1]. Pada saat ini game sudah tersedia di berbagai macam platform, platform yang bisa memainkan game antara lain yaitu PC (*Personal Computer*), *Console*, dan *Mobile* [2]. *Game* juga memiliki beberapa *genre*, seperti *genre Adventure*, *Fighting*, Simulasi, *Real-Time Strategy* (RTS), *Role-Playing Game* (RPG), *First Person Shooter* (FPS) dan masih banyak lagi jenis yang lainnya[3]. Namun, sangat disayangkan masih banyak pengguna *game* di Indonesia yang menggunakan *game* buatan asing, sehingga banyak anak muda di Indonesia yang melupakan budaya lokal termasuk cerita rakyat.

Tetapi, dengan adanya aplikasi canggih, anak muda Indonesia juga dapat membuat *game* yang dapat dinikmati oleh banyak orang dan membanggakan anak-anak Indonesia, terlebih lagi membuat *game* sebagai media dalam pembelajaran dan memperkenalkan cerita rakyat Indonesia. Cerita rakyat adalah cerita yang berkembang dikalangan masyarakat. Itu merupakan karakteristik dari semua negara dengan berbagai ragam budaya yang terdiri dari banyaknya budaya dan sejarah dari asal negara yang ada[4].

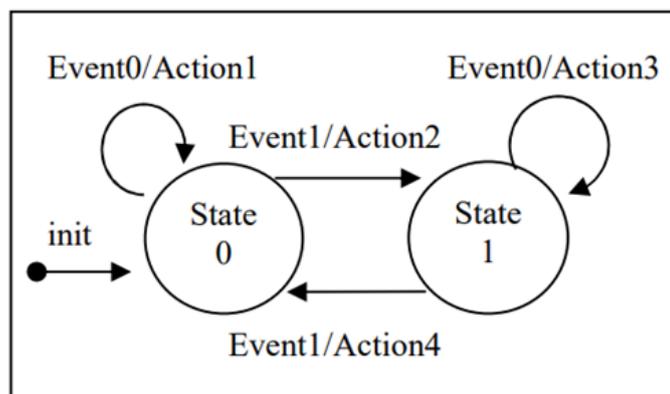
Pada Penelitian sebelumnya dalam mengenalkan cerita rakyat melalui *game* dengan metode yang sama adalah *game* implementasi *Finite State Machine* (FSM) pada game “Pendekar Cisadane”. Dengan dibuatnya *game* “Pendekar Cisadane” dapat menjadi media untuk lebih mengenal cerita rakyat kepada anak muda di Indonesia. Metode FSM yang pada game ini digunakan untuk menjadi penggerak dari *Non Playable Character* (NPC)[5].

Game dengan implementasi FSM selanjutnya adalah game “Permainan Tradisional Setatak”, game ini menjadi media pembelajaran kepada anak Indonesia dalam mengenal permainan tradisional Setatak dalam bentuk modern dan menyenangkan. Dalam game ini FSM diterapkan untuk mengatur perilaku sistem yaitu perpindahan kaki antara petak satu dengan petak lainnya selama game berlangsung[6].

Penelitian yang akan dibuat kali ini adalah membuat game 2D platformer bergenre Adventure sebagai media pembelajaran cerita rakyat “Timun Mas” dalam memperkenalkan tokoh Timun Mas dengan sedikit perubahan pada jalan ceritanya. Game ini dibuat menggunakan unity sebagai game enginnya dan terdapat tokoh-tokoh utama dalam cerita Timun Mas, seperti Timun Mas sebagai karakter utama yang dapat dikontrol oleh pemain dan Buto Ijo sebagai bos utama dalam game ini yang harus dikalahkan. Metode yang digunakan pada game ini ialah metode FSM, metode ini digunakan kepada Non Playable Character (NPC) yaitu musuh-musuh yang ada pada game ini agar NPC atau musuh dapat bertindak berdasarkan aksi yang diberikan pemain. Pada game ini pemain akan dituntut untuk mencari sebuah kunci agar dapat membuka pintu untuk menyelesaikan permainan, sedangkan fungsi FSM pada musuh di game ini untuk menghalau pemain agar tidak bisa menyelesaikan permainan.

ISO 9126 adalah salah satu bagian dari standar ISO/IEC 9126 yang khusus mendefinisikan tentang kualitas penggunaan (quality in use) dari suatu perangkat lunak. Dalam ISO 9126 ini terdapat 4 model kualitas/karakteristik yang masing-masing memiliki metrik sendiri untuk melakukan evaluasi[7]

Finite State Machine (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan perilaku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: State (keadaan), Event (Kejadian) dan Action (aksi). Sebagai metodologi perancangan sistem kontrol, penerapan FSM telah banyak diimplementasikan pada perangkat lunak, khususnya pada game[8].



Gambar 1. Finite State Machine

Gambar 1. memperlihatkan Finite State Machine(FSM) dengan dua buah state dan dua buah input serta empat buah output yang berbeda seperti terlihat pada gambar, ketika sistem mulai dinyalakan, sistem akan berpindah menuju State0, pada keadaan ini sistem akan menghasilkan Action1 jika terjadi masukan Event0, sedangkan jika terjadi Event1 maka Action2 akan dieksekusi kemudian sistem selanjutnya berpindah ke keadaan State1 dan seterusnya.

2. METODE PENELITIAN

2.1 Desain Karakter

Tabel 1. merupakan desain karakter pada game Timun Mas, di tabel tersebut diperkenalkan keterangan dari masing-masing karakter pada game ini.

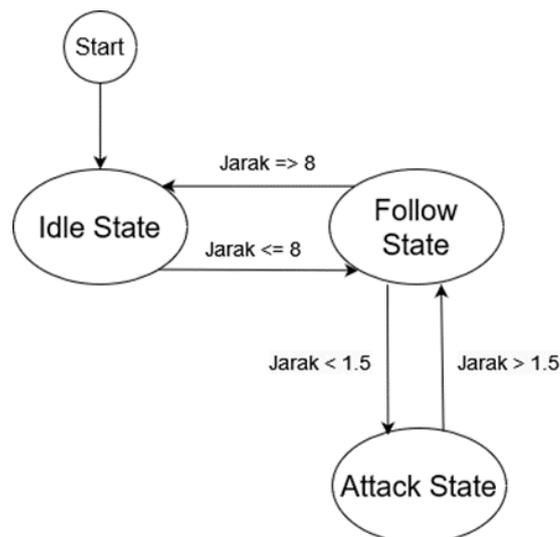
Table 1. Desain Karakter Pada *Game*

No	Karakter	Keterangan
1		Timun Mas merupakan karakter utama dalam <i>game</i> ini. Dia dapat dikontrol <i>user</i> seperti bergerak, menyerang dan melompat. Timun Mas dapat melemparkan jarum kearah musuh.
2		Buto Ijo merupakan penjahat utama dalam <i>game</i> ini. Dia berperan sebagai <i>bos</i> di level akhir. Buto Ijo dapat menyerang pemain dari jarak dekat.
3		Kurcaci Hijau merupakan anak buah dari Buto Ijo, Kurcaci Hijau dapat ditemukan pada level 1 dan dapat menyerang Pemain dari jarak dekat.
4		Monster Mata Satu merupakan anak buah Buto Ijo yang dapat menyerang dari jarak jauh. Musuh ini dapat ditemukan di level 2

2.2 Penerapan Metode

Pada penerapan metode kali ini menggunakan metode *Finite State Machine* (FSM), yaitu pada awal permainan sistem akan menempati satu *state* (keadaan), kemudian sistem akan berpindah ke *state* lainnya jika mendapatkan masukan *event* (Kejadian). perpindahan keadaan ini umumnya juga disertai oleh *action* (aksi) yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa *action* (aksi) sederhana yang melibatkan rangkaian proses yang relatif rumit[9]. Berikut merupakan penerapan FSM pada musuh dalam permainan ini.

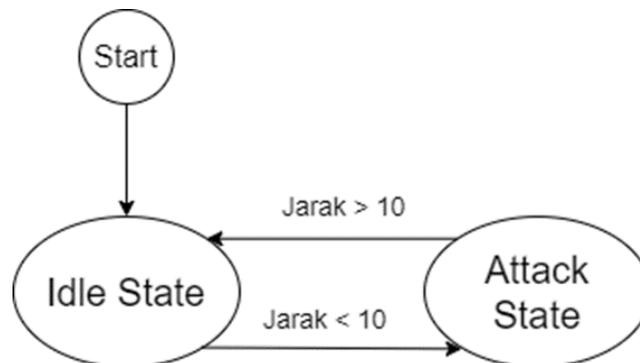
Pada Gambar 2. merupakan penerapan *Finite State Machine* pada Kurcaci Hijau. Kurcaci Hijau memiliki 3 *state* yaitu *state idle*, *state follow* dan *state attack*. Pada saat *game* dimulai Kurcaci Hijau akan berada pada *state idle*, jika pemain mendekati Kurcaci Hijau dengan jarak ≤ 8 maka Kurcaci Hijau akan memasuki *state follow* dan akan mengikut pemain, lalu jika jarak *Pemain* < 1.5 maka Kurcaci Hijau akan memasuki *state attack* dan akan menyerang pemain dengan *damage -10 health point*. Kurcaci Hijau akan kembali ke *state follow* jika jarak pemain > 1.5 dan memasuki *state idle* jika jarak *Pemain* $\Rightarrow 8$.



Gambar 2. Penerapan *Finite State Machine* Pada Kurcaci Hijau

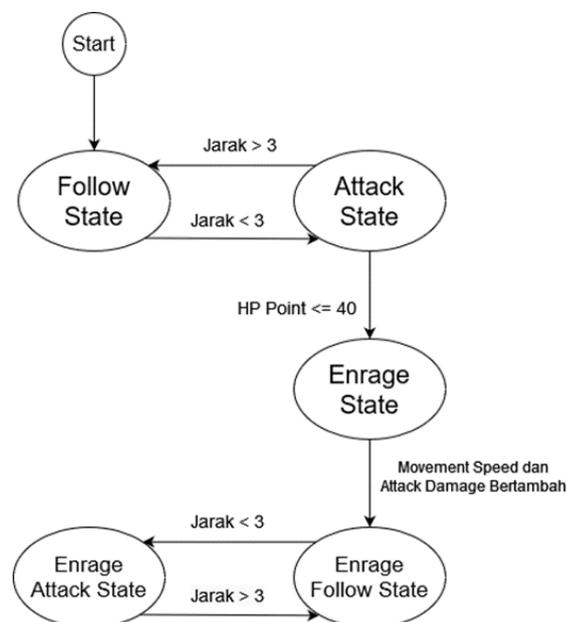
Pada Gambar 3. merupakan penerapan *Finite State Machine* pada Monster Mata Satu. Monster Mata Satu memiliki 2 *state* yaitu *state idle* dan *state attack*. Pada saat *game* dimulai Monster Mata Satu akan memasuki *state*

idle, jika pemain mendekati Monster Mata Satu < 10 , maka Monster Mata Satu akan memasuki *state attack* dan mulai menyerang *Pemain*. Monster Mata Satu akan kembali memasuki *state idle* jika *Pemain* berjarak > 10 .



Gambar 3. Penerapan *Finite State Machine* Pada Monster Mata Satu

Pada Gambar 4. merupakan penerapan *Finite State Machine* pada Buto Ijo. Buto Ijo sebagai *bos* di *game* ini memiliki 5 *state* yaitu *state follow*, *state attack*, *state enrage*, *state enrage follow*, dan *enrage attack*. Pada saat *game* dimulai Buto Ijo akan memasuki *state follow* dan akan langsung mengikuti pemain kemanapun pemain pergi, jika Buto Ijo atau pemain mendekati dengan jarak < 3 maka Buto Ijo akan memasuki *state attack* dan akan menyerang *Pemain* dengan *damage* -10 *health point*, lalu jika pemain menjauh dengan jarak > 3 maka Buto Ijo kembali memasuki *state follow*. Pada saat *health* Buto Ijo mencapai ≤ 40 maka Buto Ijo memasuki *state enrage* dan langsung beralih dari *state follow* ke *state enrage follow*, dalam mode *state enrage follow* kecepatan dan serangan yang diberikan Buto Ijo meningkat, pada saat pemain atau Buto Ijo mencapai jarak < 3 maka Buto Ijo akan memasuki *state enrage attack* dan akan menyerang pemain dengan *damage* -40 *health point*. lalu jika pemain menjauh dengan jarak > 3 maka Buto Ijo Kembali memasuki *state enrage follow*.



Gambar 4. Penerapan *Finite State Machine* Pada Buto Ijo

2.3 Desain Level

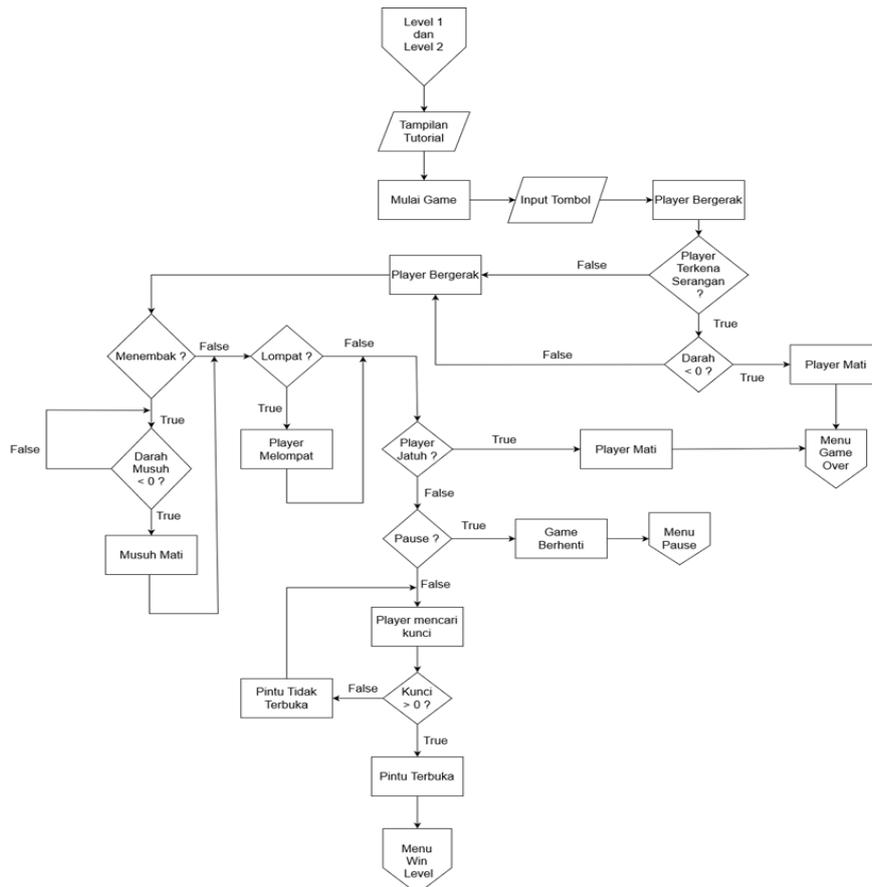
Dalam *game* ini terbagi menjadi 3 *level* utama, disetiap *level* memiliki rintangan yang berbeda, Berikut merupakan desain *level* dari *game* “Timun Mas”.

- a. Peraturan yang ada didalam *game*
 - 1) Pemain mempunyai *Health point* berbentuk *Health bar*.
 - 2) Pemain akan mati jika *Health point* < 0 .
 - 3) Pemain memiliki 3 aksi utama bergerak, melompat, dan menyerang.

- 4) Pemain harus mencari pintu untuk keluar dari kastil.
 - 5) Pemain diharuskan mengambil kunci untuk membuka pintu.
 - 6) Pemain dapat menembakan sebuah peluru jarum kepada musuh, jarum memberikan 10 *damage* kepada musuh
 - 7) Peluru pemain akan bertambah 3 jika mengambil *item* sekantong jarum.
 - 8) Pemain akan kebal terhadap *damage* serangan apapun jika pemain mengambil *item* timun.
 - 9) Pemain harus mengalahkan musuh dan melewati rintangan pada *level* 1 sampai *level* 3.
 - 10) Pemain diharuskan mengalahkan Buto Ijo sebagai bos pada *level* 3 atau *level* terakhir.
 - 11) Terdapat 2 rintangan dalam *game* ini yaitu :
 - a) Api merupakan rintangan yang bergerak dalam bentuk patrol, rintangan ini dapat memberikan *damage* 10 kepada Pemain jika mengenyainya. Api ini tidak dapat di bunuh, pemain diharuskan menghindari api ini
 - b) Duri merupakan rintangan yang diam ditempat, rintangan ini dapat memberikan *damage* 5 kepada Pemain jika mengenyainya, pemain diharuskan menghindari duri ini
 - 12) Terdapat 3 musuh utama dalam *game* ini yang memiliki keunikan berbeda yaitu :
 - a) Musuh 1 sebagai Kurcaci Hijau, dia memiliki *health point* 20 dan dapat menyerang pemain jika pemain terlalu dekat dengannya
 - b) Musuh 2 sebagai mata satu, dia memiliki *health point* 10 dan dapat menyerang pemain dari jarak jauh jika Pemain memasuki jarak tembaknya
 - c) Musuh 3 sebagai Buto Ijo atau bos dalam cerita ini, dia memiliki *health point* 100 dan dapat memasuki mode marah jika *health point* nya ≤ 40 .
 - 13) Apabila pemain mati maka harus mengulang *level* yang ada.
- b. Kondisi yang terdapat di berbagai *level* sebagai berikut :
- 1) *Level* 1
Pada *level* 1 pemain akan berhadapan dengan berbagai rintangan seperti api dan duri, di sini pemain juga akan bertemu dengan musuh 1 yaitu Kurcaci Hijau, player diharuskan mencari kunci untuk bisa membuka pintu dan melangkah ke *level* selanjutnya.
 - 2) *Level* 2
Pada *level* 2 hampir mirip seperti *level* 1, pemain harus menghindari rintangan dan musuh, serta mencari kunci untuk membuka pintu, yang membedakan hanyalah di *level* ini jenis musuh bertambah, yaitu pemain akan bertemu Monster Mata Satu yang dapat menembak dari jarak jauh.
 - 3) *Level* 3
Pada *level* 3 merupakan *level* akhir yang di mana pemain akan berhadapan langsung dengan Buto Ijo sebagai bos *level* ini, pemain harus mengalahkan Buto Ijo agar dapat menyelesaikan *level* ini.
- c. *Winner* dan *Game Over*
- 1) Pemain dinyatakan berhasil ketika dapat mencapai pintu keluar pada *level* 1 dan 2, sedangkan pada *level* 3 pemain perlu mengalahkan Buto Ijo untuk menyelesaikan *level*.
 - 2) Pemain dinyatakan gagal ketika *health bar* pemain habis atau mencapai 0, dan pemain dinyatakan gagal jika terjatuh saat melewati platform.
- d. *Health Bar*
Health bar merupakan darah pemain atau nyawa pemain saat berada didalam permainan, berikut merupakan hal-hal yang bisa berdampak pada *health bar*:
- 1) Pemain akan kehilangan *health point* -10 saat terkena musuh 1 (Kurcaci Hijau).
 - 2) Pemain akan kehilangan *health point* -10 saat terkena rintangan api.
 - 3) Pemain akan kehilangan *health point* -5 saat terkena rintangan duri.
 - 4) Pemain akan kehilangan *health point* -20 saat terkena peluru dari musuh 2 (Mata Satu) dan rintangan duri
 - 5) Pemain akan kehilangan *health point* -10 saat terkena serangan bos dan berkurang -40 saat terkena serangan bos yang memasuki mode *enrage*
 - 6) Pemain akan bertambah *health point* +20 saat mengambil *item* *hearth*
 - 7) Pemain akan kebal terhadap serangan jika mengambil *item* timun
- e. Peluru
Peluru didalam *game* merupakan sebuah jarum, yang di mana jarum tersebut akan berkurang -1 saat pemain menembak dan bertambah +3 saat pemain mengambil *item* sekantong jarum. Peluru ini dapat memberikan *damage* -10 kepada musuh-musuh yang ada di dalam *game*.

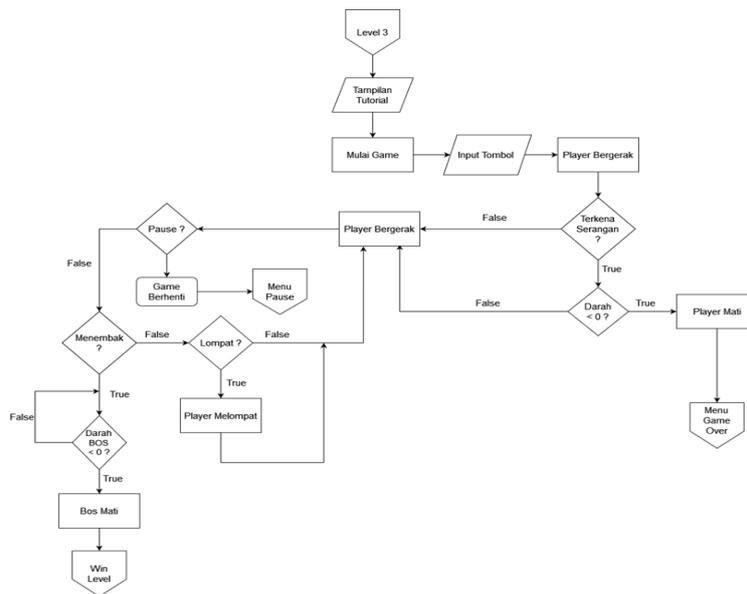
2.4 Flowchart

Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem[10]. Pada Gambar 5. terdapat *flowchart* level 1 dan level 2 dari alur permainan ini. Di dalam *flowchart* tersebut menjelaskan bagaimana pemain harus menyelesaikan level permainan ini. Di alur tersebut dijelaskan bahwa *player* perlu untuk mencari kunci supaya pintu yang terdapat pada level itu terbuka. Di dalam *game* pemain dapat bergerak, melompat dan menyerang, jika *player* terkena serangan dan darah < 0 maka pemain akan mati, lalu *pemain* akan langsung mati apabila jatuh dari *platform*.



Gambar 5. Flowchart Level 1 dan Level 2

Pada Gambar 6 terdapat *flowchart bos level*. Flowchart ini hampir mirip dengan *flowchart* level 1 dan 2, yang membedakan hanyalah keadaan menang *player*. Jika pada *flowchart* level 1 dan 2 *player* diharuskan mencari kunci dan pintu untuk menang, maka pada *flowchart bos level* *player* hanya bisa menang jika sudah mengalahkan *bos level* yang ada.



Gambar 6. Flowchart Level Bos

2.5 Rancangan Pengujian

Pengujian merupakan tahapan yang sangat penting dalam pengembangan aplikasi *game*, hal ini berguna untuk mengetahui apakah *game* dapat berjalan sesuai yang diharapkan. Pengujian dilakukan menggunakan pengisian kuesioner setelah memainkan *game* ini. Pengujian kuesioner dilakukan dengan ISO 9126 sebagai standar yang diterapkan. Para responden akan diberikan 9 pertanyaan pilihan ganda serta 5 jawaban mulai dari sangat setuju hingga sangat tidak setuju. Diharapkan dengan adanya pengujian ini penulis dapat menemukan bug dan error yang terdapat pada *game* yang dikembangkan.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi *Finite State Machine* (FSM)

Pada tabel di bawah merupakan implementasi dari metode *Finite State Machine* (FSM). Di tabel tersebut dijelaskan cara kerja musuh untuk memasuki suatu state. Tabel 2. merupakan implementasi FSM untuk Kurcaci Hijau di sana terdapat 3 state dan cara kerja state tersebut untuk pindah ke state lain.

Tabel 2. Implementasi FSM Pada Kurcaci Hijau

Keadaan	Kejadian	Tindakan	Hasil
Idle	Jarak player < 10	Musuh memasuki state follow	Berhasil
Idle	Jarak player < 3	Musuh memasuki state attack	Berhasil
Follow	Jarak player < 3	Musuh memasuki state attack	Berhasil
Follow	Jarak player > 10	Musuh memasuki state idle	Berhasil
Attack	Jarak player > 3	Musuh memasuki state follow	Berhasil
Attack	Jarak player > 10	Musuh memasuki state idle	Berhasil

Pada Tabel 3. merupakan implementasi FSM untuk Monster Mata Satu. Di dalam tabel tersebut menjelaskan cara kerja dari 2 state yang ada.

Tabel 3. Implementasi FSM Pada Monster Mata Satu

Keadaan	Kejadian	Tindakan	Hasil
Idle	Jarak player < 10	Musuh memasuki state attack	Berhasil
Attack	Jarak player > 10	Musuh memasuki state idle	Berhasil

Pada Tabel 4. merupakan implementasi FSM untuk Buto Ijo, di tabel tersebut menjelaskan alur dari ke 5 *state* yang ada, diawali pada *state follow* dan akan berpindah ke *state enrage* jika darah bos ≤ 40 . Pada *state enrage*

bos tidak akan kembali ke *state follow* dan *state attack*, melainkan langsung berpindah ke *state follow enrage* dan *attack enrage*.

Tabel 4. Implementasi FSM Pada Buto Ijo

Keadaan	Kejadian	Tindakan	Hasil
Follow	Jarak player < 3	Musuh memasuki state Attack	Berhasil
Follow	Darah musuh <= 40	Musuh memasuki state Enrage	Berhasil
Attack	Jarak player > 3	Musuh memasuki state Follow	Berhasil
Attack	Darah musuh <= 40	Musuh memasuki state Enrage	Berhasil
Enrage	Darah musuh <= 40	Musuh memasuki state enrage follow	Berhasil
Enrage	Darah musuh <= 40	Musuh memasuki state enrage attack	Berhasil
Enrage Follow	Jarak player < 3	Musuh memasuki state enrage attack	Berhasil
Enrage attack	Jarak player > 3	Musuh memasuki state enrage follow	Berhasil

3.1.1 Hasil Permainan

Pada Gambar 7. terdapat tampilan dari menu utama. Tampilan ini akan muncul pertama kali saat game dimulai dan di tampilan tersebut terdapat tombol mulai, tentang dan keluar.



Gambar 7. Tampilan Menu Utama

Pada Gambar 8. merupakan tampilan dari level 1 dan level 2 pada game ini. Pada tampilan tersebut terdapat beberapa tombol seperti 4 tombol *movement* dan 1 tombol *pause* yang akan menghentikan game sementara dan menampilkan menu pause saat di tekan. Didalam gameplay juga terdapat beberapa indikator seperti indikator health bar, indikator jumlah peluru dan indikator kunci. Di level 1 dan 2 pemain akan menemui beberapa musuh dan rintangan yang ada, lalu pemain juga dapat mengambil beberapa item untuk membantu menyelesaikan level.



Gambar 8. Tampilan Level 1 dan Level 2

Pada Gambar 9. merupakan tampilan level bos, sama seperti tampilan gameplay sebelumnya yang membedakan hanyalah terdapat indikator *health* bos untuk melihat darah dari bos tersebut. Pada level ini berbeda dengan level-level biasanya, karena di level ini tidak ada jalan keluarnya dan satu-satunya jalan keluar adalah

mengalahkan buto ijo sebagai bos ruangan tersebut. Di level ini pemain dapat mengambil beberapa item seperti biasa untuk membantu mengalahkan bos.



Gambar 9. Tampilan Level Bos

3.1.2 Hasil Pengujian

Perhitungan Persamaan :
$$\text{Skor} = \frac{1060}{1260} \times 100\% \quad (1)$$

$$\text{Skor} = 84,12\%$$

Pada Tabel 5. merupakan presentase keberhasilan. Berdasarkan hasil pengujian kualitas perangkat lunak yang dibangun berdasarkan karakteristik *usability* pada *game* mendapatkan hasil keberhasilan presentase yaitu sebesar 84,12%, nilai yang telah diperoleh tersebut selanjutnya dikonversi berdasarkan skala konversi pengujian. Dapat disimpulkan bahwa nilai presentase yang diperoleh menunjukkan kualitas perangkat lunak karakteristik *usability* mempunyai skala “Sangat Baik”.

Tabel 5. Presentase Keberhasilan

Presentase (%) Skor	Kriteria
0 - 20	Sangat Buruk
21 - 40	Buruk
41 - 60	Cukup Baik
61 - 80	Baik
81 - 100	Sangat Baik

4. KESIMPULAN

Berdasarkan hasil setelah melakukan proses perancangan, penerapan serta pengujian pada penelitian ini, maka didapatkan kesimpulan sebagai berikut:

- Penerapan metode *Finite State Machine* (FSM) dapat berjalan dengan baik,
- Game* ini dapat digunakan sebagai sarana untuk lebih mengenal cerita Timun Mas,
- Implementasi algoritme FSM pada *game* ini dapat menjadi dasar penelitian untuk mengembangkan kecerdasan buatan pada penelitian lainnya.

DAFTAR PUSTAKA

- [1] A. Enggar *et al.*, “Implementasi Finite State Machine Untuk Npc Pada Game 2d Side-Scroll Shooter Implementation Of Finite-State Machine For Npc In 2d Side Scroll Shooter,” vol. 9, no. 3, pp. 1080–1086, 2022.
- [2] F. Alamsyah, W. Diwa, and A. Yunus, “Implementasi Algoritma Collision Detection Dan Finite State Machine Untuk Karakter Musuh Pada Game Bertipe Metroidvania,” *RAINSTEK J. Terap. Sains Teknol.*, vol. 1, no. 2, pp. 8–13, 2019,
- [3] F. Sakrani, S. Adi Wibowo, and F. Santi wahyuni, “Implementasi Finite State Machine Sebagai Kontrol Untuk Non

- Player Character Pada Game ‘Lastri and the Last Tree,’” *JATI (Jurnal Mhs. Tek. Inform.,* vol. 4, no. 2, pp. 128–135, 2020,
- [4] G. S. Wibowo and M. Syafrullah, “Penerapan Metode Multimedia Development Life Cycle Untuk Aplikasi Game Permainan Timun Mas Berbasis Application of Multimedia Development Life Cycle,” no. September, pp. 982–990, 2022.
- [5] A. T. Ramadan and M. Hardjianto, “Penerapan Finite State Machine Pada Game ‘ Pendekar Cisdane ’ Berbasis Android Application of Finite State Machine on Android-Based " Pendekar Cisdane " Game,” no. September, pp. 1065–1072, 2022.
- [6] A. Agustin, A. Evel, S. Susanti, and R. Rahmaddeni, “Implementasi Metode Finite State Machine pada Permainan Tradisional Setatak Berbasis Android,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi),* vol. 8, no. 2, pp. 738–751, 2021,
- [7] V. A. K. Siren, N. Y. Setiawan, and R. I. Rokhmawati, “Evaluasi kualitas perangkat lunak menggunakan ISO / IEC 9126-4 quality in use (Studi kasus : FILKOM Apps) [Evaluation of software quality using ISO / IEC 9126-4 quality in use (case study: FILKOM Apps)],” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.,* vol. 3, no. 2, pp. 1625–1632, 2019,
- [8] C. A. Z. Wahana, S. Adi Wibowo, and A. Wahid, “Game Adventure Horror ‘Let’S Escape’ Dengan Unity Engine Berbasis Desktop Menggunakan Metode Finite State Machine,” *JATI (Jurnal Mhs. Tek. Inform.,* vol. 4, no. 2, pp. 306–314, 2020,
- [9] 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,
- [10] R. Rosaly and A. Prasetyo, “Pengertian Flowchart Beserta Fungsi dan Simbol-simbol Flowchart yang Paling Umum Digunakan,” *Https://Www.Nesabamedia.Com,* vol. 2, p. 2, 2019.