

## **PENERAPAN ALGORITMA *FINITE STATE MACHINE* PADA *NON-PLAYABLE CHARACTER* DALAM GAME “SANGKURIANG: SIMPLE PLATFORM GAME”**

**Erico Septian Widodo<sup>1\*</sup>, Dewi Kusumaningsih<sup>2</sup>**

<sup>1,2</sup> Teknik Informatika, Teknologi Informasi, Universitas Budi Luhur, Jakarta Selatan, Indonesia

Email: <sup>1\*</sup>1911500013@student.budiluhur.ac.id, <sup>2</sup>dewi.kusumaningsih@budiluhur.ac.id

(\* : corresponding author)

**Abstrak-** Cerita rakyat merupakan bagian dari kekayaan budaya yang umumnya menceritakan tentang suatu peristiwa yang terjadi di suatu tempat. Pada Zaman sekarang bukan hanya anak kecil saja yang suka bermain *game* namun orang dewasa juga banyak yang suka bermain *game*, namun banyak anak muda yang tidak mengenal dan tidak minat untuk mendengar cerita rakyat Indonesia seperti contohnya cerita rakyat Sangkuriang, mereka perlu diajarkan agar dapat memahami kekayaan budaya Indonesia namun dengan cara yang menarik, oleh karena itu untuk mengenalkan cerita rakyat Sangkuriang dibuat *game 2D action platformer* dengan *game engine* Unity, *game platformer* merupakan jenis *video game* di mana tokoh utama memiliki kemampuan untuk melompati rintangan, berpindah dari satu lokasi ke lokasi lainnya, hingga mencapai tujuan tertentu serta *Finite State Machine* sebagai metode untuk mengatur perilaku dari *non playable character* yang berada di dalam *game* agar pergerakannya tidak monoton, serta dapat menarik minat pemain terutama anak muda yang berumur 12 – 18 tahun, Adapun metode pengembangan yang di terapkan pada penelitian ini yaitu *Gaming Development Life Cycle* (GDLC) dengan 6 unsur atau tahapan yaitu tahap inisialisasi, tahap pra-produksi, tahap produksi, tahap pengujian, tahap beta, dan tahap rilis, berdasarkan presentase responde pada pengujian fungsional sebesar 84,1 % dan usability 77,6%. serta *finite state machine* berhasil dengan baik dan sesuai keinginan serta cerita dalam *game* dapat disampaikan dengan baik.

**Kata Kunci:** *Finite State Machine* (FSM), *Gaming Development Life Cycle* (GDLC), Cerita Rakyat

## **APPLICATION OF *FINITE STATE MACHINE ALGORITHM* ON *NON-PLAYABLE CHARACTER* IN GAME “SANGKURIANG : A SIMPLE PLATFORM GAME”**

**Abstract-** *Folklore is a part of cultural wealth that generally tells about an event that occurred somewhere. Nowadays not only children like to play games but also many adults like to play games, but many young people do not know and have no interest in hearing Indonesian folklore such as the Sangkuriang folklore, they need to be taught in order to understand the richness of Indonesian culture but in an interesting way, therefore to introduce the Sangkuriang folklore, a 2D action platformer game was made with the Unity game engine, platformer games are a type of video game where the main character has the ability to jump over obstacles, move from one location to another, Finite State Machine as a method for regulating the behavior of non-playable characters in the game so that the movement is not monotonous, and can attract the interest of players, especially young people aged 12 - 18 years, The development method applied in this study is the Gaming Development Life Cycle (GDLC) with 6 elements or stages, namely the initialization stage, pre-production stage, production stage, testing stage, beta stage, and release stage, based on the percentage of respondents in functional testing of 84.1% and usability of 77.6%. In addition, the finite state machine works well and as desired and the story in the game can be conveyed well.*

**Keywords:** *Finite State Machine* (FSM), *Gaming Development Life Cycle* (GDLC), *Folklore*

---

### **1. PENDAHULUAN**

Cerita rakyat adalah bagian yang amat berharga dari kekayaan budaya dan sejarah Indonesia. Kisah-kisah rakyat ini seringkali mengisahkan peristiwa yang berlangsung di suatu tempat atau menggambarkan keadaan tempat tersebut. Kisah-kisah ini memiliki nilai yang tidak bisa diukur dengan harta materi[1]. Setiap daerah memiliki ciri khas yang menunjukkan harta budaya dan sejarah masing – masing daerah. Walaupun sekarang sangat mudah untuk belajar mengenai kebudayaan, banyak anak muda zaman sekarang yang tidak begitu mengetahui tentang kebudayaan dan nilai-nilainya maka dari itu mereka harus diajarkan tetapi tidak melewatkan hal-hal yang menarik para anak muda yaitu melewati media digital seperti Game[2].

Game merupakan suatu bentuk permainan multimedia yang dirancang dengan sebaik mungkin untuk menarik minat pemain agar mendapatkan kepuasan batin. Zaman sekarang bukan hanya anak kecil saja yang suka bermain *game* namun orang dewasa juga banyak yang suka bermain *game* [3]. Ide cerita pada *game* dapat diambil dari sejarah, olah raga, kehidupan nyata, atau cerita rakyat. Salah satu cerita rakyat yang dapat diangkat ke dalam *game* adalah kisah Sangkuriang atau Tangkuban Perahu yang menceritakan kegagalan Sangkuriang

dalam memining Dayang Sumbi. untuk mengenalkan kebudayaan dari cerita rakyat sangkuriang maka dibuatlah sebuah *game* dengan genre 2D platformer berbasis desktop, *game* platformer merupakan jenis *video game* di mana tokoh utama memiliki kemampuan untuk melompati rintangan, berpindah dari satu lokasi ke lokasi lainnya, hingga mencapai tujuan tertentu[4].

Unity menjadi *game engine* yang digunakan untuk membuat *game* ini. Pemain akan menggunakan karakter utama sangkuriang untuk melewati 3 level utama dalam Game ini, dengan mengumpulkan beberapa item dan mengalahkan musuh yang menghadang. Setiap levelnya memiliki tingkat kesulitan yang berbeda dan terdapat item pendukung untuk membantu pemain mengalahkan musuh. Dalam perancangan *game* ini, digunakan metode *Finite State Machine* (FSM), yang berperan dalam mengatur kondisi-kondisi tertentu pada objek yang ada dalam *game*, seperti karakter musuh.

## 1.1 Game

*Game* memiliki peran penting dalam kehidupan sehari-hari. Biasanya, *game* memberikan tantangan kepada pemain yang harus diatasi untuk menyelesaikan permainan. Selain digunakan untuk interaksi sosial, *game* menyediakan cara yang menyenangkan bagi anak-anak dan orang dewasa untuk belajar dan mengembangkan kemampuan pemecahan masalah mereka[5].

*Game* merupakan permainan dengan menggunakan aturan tertentu sehingga terdapat elemen menang dan kalah, Biasanya dimainkan dalam konteks yang tidak serius dengan tujuan untuk menyegarkan diri[6].

## 1.2 Jenis – Jenis Game

Pada saat ini, permainan dapat diakses dan dimainkan melalui berbagai *platform*, berikut jenis *game* beserta penjelasannya[7].

- Arcade Game**, jenis *game* yang umumnya memerlukan perangkat keras khusus yang dirancang untuk jenis *video game* tertentu, serta didukung oleh aksesoris seperti sensor gerak dan setir kemudi untuk simulasi mobil.
- PC Game**, jenis *game* yang dimainkan pada komputer pribadi memiliki keunggulan dari antarmuka *input* dan *output* yang lebih responsif.
- Console Game**, jenis *game* yang dimainkan di konsol tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii, digunakan untuk menikmati berbagai *video game*.
- Handheld Game**, jenis *game* yang dimainkan pada platform portabel seperti Sony PSP Vita dan Nintendo Switch, memungkinkan pemain untuk membawanya ke mana-mana.
- Mobile Game**, jenis *game* yang dirancang untuk dimainkan di ponsel (*mobile*) dan tablet.

## 1.3 Gaming Development Life Cycle

*Gaming Development Life Cycle* (GDLC) adalah kerangka kerja yang digunakan untuk mengarahkan proses pembuatan aplikasi *game*. Proses ini melibatkan tiga kegiatan inti: Desain dan Pembuatan Prototipe, Produksi, dan Pengujian. Desain dan Pembuatan Prototipe melibatkan pembuatan konsep desain awal dari aplikasi *game*, yang kemudian diwujudkan dalam bentuk prototipe. Fase Produksi fokus pada pembuatan kode sumber dan aset, serta mengintegrasikannya dengan lancar. Setelah fase produksi, aplikasi dites secara menyeluruh dalam beberapa iterasi hingga mencapai tahap beta. Selama tahap beta, aplikasi dievaluasi oleh penguji pihak ketiga sebelum siap untuk dirilis. GDLC terdiri dari total 6 tahap pengembangan [8], yaitu :

- Inisialisasi
- Pra – Produksi
- Produksi
- Pengujian
- Beta
- Rilis

## 2. METODE PENELITIAN

### 2.1 Penerapan Metode

Pada penelitian ini digunakan *Finite State Machine* sebagai algoritma pembuatannya. *Finite State Machine* adalah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut : Keadaan (*state*), kejadian (*event*) dan aksi (*action*). Sebagai sebuah metodologi perancangan sistem kontrol, penerapan FSM telah banyak diterapkan pada perangkat lunak, khususnya pada *game*[9].

Pada satu saat dalam periode waktu yang cukup signifikan, system akan berada pada salah satu *state* yang aktif. System dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau event tertentu,

baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri[10] , ada beberapa penerapan *Finite State Machine* pada penelitian ini, diantaranya :

### 2.1.1 Penerapan FSM Pada Musuh Orc

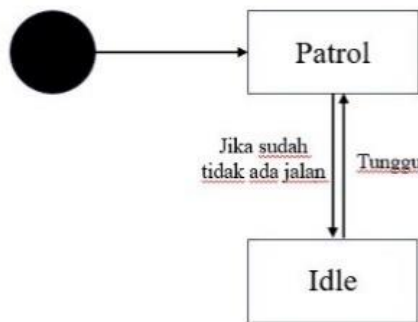


Gambar 1. Penerapan FSM Pada Musuh Orc

*Finite State Machine* yang diterapkan pada musuh Orc terdiri dari tiga state yaitu *idle* , *chase* dan *attack*. Berikut adalah seluruh deskripsi dari masing – masing *state* :

- State* : *Idle*  
Deskripsi : *State Idle* merupakan keadaan awal dari FSM, pada keadaan ini musuh orc akan berdiam di tempat dan melakukan sebuah gerakan tertentu.
- State* : *Chase*  
Deskripsi : Musuh akan mulai mengejar pemain apabila pemain berada di jarak < 7 pixel, jika pemain sudah berada di luar jarak maka musuh akan kembali ke *Idle State* .
- State* : *Attack*  
Deskripsi : Musuh akan menyerang pemain jika berada di jarak < 1.5 pixel, jika pemain berada di jarak > 2.5 pixel maka musuh akan kembali ke *Chase State*.

### 2.1.2 Penerapan FSM Pada Musuh Snake



Gambar 2. Penerapan FSM Pada Musuh

*Finite State Machine* yang diterapkan pada musuh *Snake* terdiri dari dua state yaitu *Patrol* dan *Idle*. Berikut adalah seluruh deskripsi dari masing – masing *state*.

- State* : *Patrol*  
Deskripsi : *State patrol* merupakan state awal pada FSM ini. Pada state ini musuh *Snake* berjalan ke kanan, jika sudah tidak ada platform untuk di pijak maka akan menjalankan *state idle*.
- State* : *Idle*  
Deskripsi : Musuh *Snake* akan menunggu 1 detik lalu akan berputar arah dan kembali menjalankan *state patrol*.

### 2.1.3 Penerapan FSM Pada Musuh Ranged Enemy

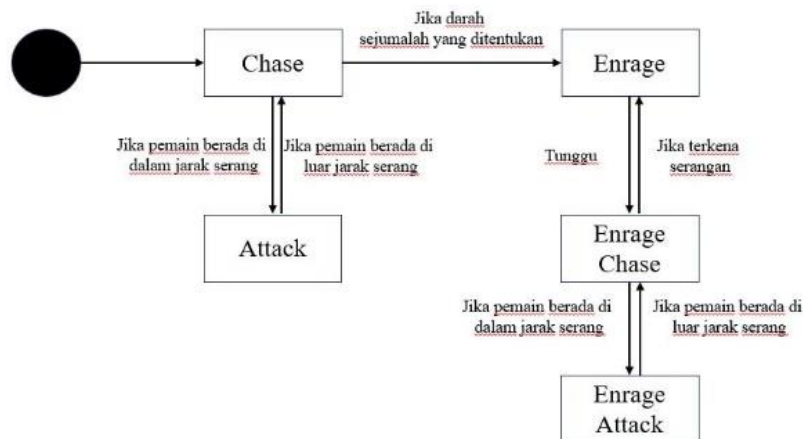


Gambar 3. Penerapan FSM Pada Musuh Ranged Enemy

Finite State Machine yang diterapkan pada musuh Orc terdiri dari tiga state yaitu *idle*, *patrol* dan *attack*. Berikut adalah seluruh deskripsi dari masing – masing *state*.

- State : *Patrol*  
Deskripsi : *State patrol* merupakan keadaan awal dari FSM ini, pada keadaan ini musuh *ranged enemy* akan berjalan ke titik yang sudah ditentukan, jika sudah mencapai titik yang sudah ditentukan maka akan menjalankan *state idle*.
- State : *Idle*  
Deskripsi : Musuh *ranged enemy* akan menunggu 1 detik lalu akan berputar arah dan kembali menjalankan *state patrol*, jika pemain berada dalam jarak tembakan maka akan menjalankan *state attack*.
- State : *Attack*  
Deskripsi : Musuh *ranged enemy* akan menyerang pemain dari jarak jauh jika pemain berada di jarak < 6 pixel, jika pemain diluar jarak serang maka akan menjalankan *state patrol*.

### 2.1.4 Penerapan FSM Pada Musuh Boss



Gambar 3. Penerapan FSM Pada Musuh Boss

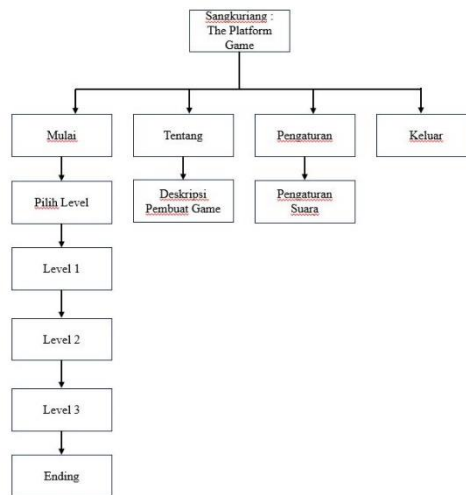
Finite State Machine yang diterapkan pada musuh Orc terdiri dari lima state yaitu *chase*, *attack*, *enrage*, *enrage chase*, dan *enrage attack*. Berikut adalah seluruh deskripsi dari masing – masing *state*.

- State : *Chase*  
Deskripsi : *State chase* merupakan keadaan awal dari FSM ini, musuh boss mengejar pemain berapapun jarak antara musuh boss dan pemain, jika jarak pemain < 2 maka akan menjalankan *state attack*, jika darah musuh boss < 5 maka akan menjalankan *state enrage*.
- State : *Attack*

Deskripsi	: Musuh boss akan menyerang pemain jika berada di jarak < 2 pixel pixel, jika pemain berada di jarak > 2 pixel maka musuh akan kembali ke <i>state chase</i> .
State	: <i>Enrage</i>
Deskripsi	: Pada <i>state</i> ini musuh boss akan kebal selama 1 detik dan mendapatkan kecepatan tambahan setelah itu langsung menjalankan <i>state enrage chase</i> .
State	: <i>Enrage Chase</i>
Deskripsi	: Dalam <i>state enrage chase</i> musuh boss mengejar pemain berapapun jarak antara musuh boss dan pemain, jika jarak pemain < 2 pixel maka akan menjalankan <i>state enrage attack</i> jika terkena serangan maka akan menjalankan <i>state enrage</i> .
State	: <i>Enrage Attack</i>
Deskripsi	: Musuh boss akan menyerang pemain 2 kali jika berada di jarak < 2 pixel, jika pemain berada di jarak > 2 pixel maka musuh akan kembali ke <i>state enrage chase</i> .

## 2.2 Rancangan Menu

Pada rancangan menu terdapat proses yang dilalui user, bagaimana perpindahan yang terjadi pada setiap scene yang ada pada game sangkuriang : simple platform game. Terdapat 4 *menu* utama serta menu level, menu utama terdiri dari *menu* mulai, *menu* tentang, *menu* pengaturan, dan *menu* keluar, masing – masing menu memiliki fungsinya tersendiri, berikut adalah struktur dari rancangan *menu*.



**Gambar 4.** Rancangan Menu

Berikut deskripsi dari tiap bagan pada rancangan menu :

- a. Mulai  
Mulai adalah sebuah tombol yang terdapat pada halaman utama, jika pemain mengklik tombol mulai maka akan masuk ke menu pilih level.
- b. Tentang  
Tentang adalah sebuah tombol yang terdapat pada halaman utama, jika pemain mengklik tombol tentang maka akan masuk ke menu deskripsi pembuat game.
- c. Pengaturan  
Pengaturan adalah sebuah tombol yang terdapat pada halaman utama, jika pemain mengklik tombol pengaturan maka akan masuk ke menu pengaturan untuk merubah suara.
- d. Keluar  
Menu untuk keluar dari *game*.
- e. Pilih Level  
Menu untuk memilih level 1 – 3.
- f. Level 1 - 3  
Pemain akan menjalankan permainan, jika berhasil maka akan lanjut ke level berikutnya.
- g. Ending  
Scene yang akan muncul jika pemain berhasil menyelesaikan level 3.

## 2.3 Rancangan Pengujian

Pada penelitian ini dilakukan pengujian melalui kuesioner, dengan melibatkan beberapa responden dari berbagai kalangan usia, dengan memberi pertanyaan setelah para responden memainkan game yang dikembangkan, beberapa tahapan yang akan dilakukan pada pengujian adalah sebagai berikut,

- a. Pengumpulan Responden  
Pada tahap ini responden diminta pendapat tentang aplikasi *game* yang telah dibuat.
- b. Pengumpulan Aplikasi  
Pada tahap ini responden mencoba aplikasi *game* yang telah dibuat untuk mengetahui tanggapan tentang game tersebut.
- c. Kuesioner  
Kuesioner ini merupakan kuesioner penelitian, dan merupakan hasil dari tanggapan para responden setelah mencoba aplikasi *game* ini. Kuesioner ini meliputi pertanyaan tentang aspek fungsional, kegunaan dan kompatibilitas, yang ada pada aplikasi permainan ini.

## 3. HASIL DAN PEMBAHASAN

Pada bagian ini berisi analisis, hasil implementasi ataupun pengujian serta pembahasan dari topik penelitian, yang bisa dibuat terlebih dahulu metodologi penelitian. Bagian ini juga merepresentasikan penjelasan yang berupa penjelasan, gambar, tabel dan lainnya.

### 3.1 Implementasi Metode

Berdasarkan penerapan metode yang diusulkan pada pembahasan sebelumnya berikut adalah hasil implementasi pada game yang dibuat.

#### 3.1.1 Implementasi FSM Pada Musuh Orc

Pada tabel 1 ini adalah hasil dari Implementasi *Finite State Machine* (FSM) pada musuh orc.

**Tabel 1.** Tabel Implementasi FSM Musuh Orc

State	Event	Action	Hasil
<i>Idle</i>	Jarak pemain < 7 pixel	Musuh akan memasuki <i>chase state</i>	Berhasil
<i>Chase</i>	Jarak < 1.5 pixel	Musuh akan memasuki <i>attack state</i>	Berhasil
<i>Chase</i>	Jarak > 7 pixel	Musuh akan memasuki <i>idle state</i>	Berhasil
<i>Attack</i>	Jarak > 2.5 pixel	Musuh akan memasuki <i>chase state</i>	Berhasil

#### 3.1.2 Implementasi FSM Pada Musuh Snake

Pada tabel 2 ini adalah hasil dari Implementasi *Finite State Machine* pada musuh snake.

**Tabel 2.** Tabel Implementasi FSM Musuh Snake

State	Event	Action	Hasil
<i>Patrol</i>	Jika sudah tidak ada platform untuk dipijak	Musuh akan memasuki <i>idle state</i>	Berhasil
<i>Idle</i>	Setelah 1 detik lalu berputar	Musuh akan memasuki <i>patrol state</i>	Berhasil

#### 3.1.3 Implementasi FSM Pada Musuh Ranged Enemy

Pada tabel 3 ini adalah hasil dari Implementasi *Finite State Machine*(FSM) pada musuh *Ranged Enemy*.

**Tabel 3.** Tabel Implementasi FSM Musuh Ranged Enemy

State	Event	Action	Hasil
<i>Patrol</i>	Jika sudah berjalan sampai titik yang ditentukan	Musuh akan memasuki <i>idle state</i>	Berhasil
<i>Patrol</i>	Jika jarak pemain < 6 pixel	Musuh akan memasuki <i>attack state</i>	Berhasil
<i>Idle</i>	Setelah 1 detik lalu berputar	Musuh akan memasuki <i>patrol state</i>	Berhasil
<i>Idle</i>	Jika jarak pemain < 6	Musuh akan memasuki <i>attack state</i>	Berhasil
<i>Attack</i>	Jika jarak pemain > 6 pixel	Musuh akan memasuki <i>patrol state</i>	Berhasil

#### 3.1.4 Implementasi FSM Pada Musuh Boss

Pada tabel 4 ini adalah hasil dari Implementasi *Finite State Machine* (FSM) pada musuh Boss

**Tabel 4.** Tabel Implementasi FSM Musuh Boss

State	Event	Action	Hasil
-------	-------	--------	-------

<i>Chase</i>	Jika jarak pemain < 2 pixel	Musuh akan memasuki <i>attack state</i>	Berhasil
<i>Attack</i>	Jika jarak pemain > 2 pixel	Musuh akan memasuki <i>chase state</i>	Berhasil
<i>Chase</i>	Jika darah musuh < 5	Musuh akan memasuki <i>enrage state</i>	Berhasil
<i>Enrage</i>	Setelah animasi selesai	Musuh akan memasuki <i>enrage chase state</i>	Berhasil
<i>Enrage Chase</i>	Jika jarak pemain < 2 pixel	Musuh akan memasuki <i>enrage attack state</i>	Berhasil
<i>Enrage Chase</i>	Jika musuh terkena damage	Musuh akan memasuki <i>enrage state</i>	Berhasil
<i>Enrage Attack</i>	Jika jarak pemain < 2 pixel	Musuh akan memasuki <i>enrage chase state</i>	Berhasil

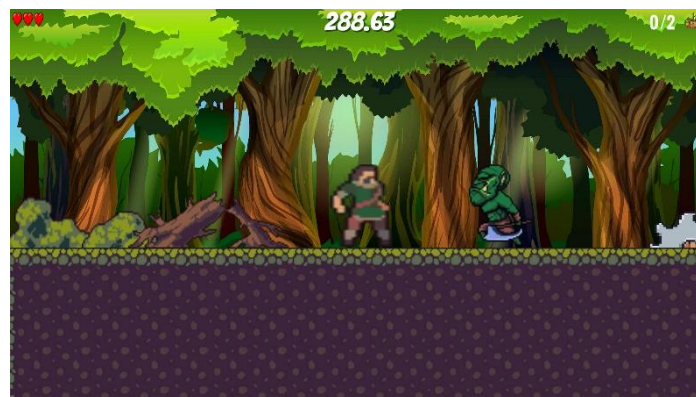
### 3.2 Tampilan Game

Berikut merupakan tampilan pada game “Sangkuriang : *Simple Platform Game*”, Pada gambar 6 terdapat 4 tombol yang memiliki fungsi yang berbeda, yaitu tombol mulai, tombol tentang, tombol pengaturan, dan tombol keluar.



Gambar 5. Tampilan Layar Utama

Pada gambar 8 menunjukkan level 1 – 2 dimana pemain diharuskan mengumpulkan item sebanyak yang telah ditentukan, di level 1 - 2 terdapat beberapa musuh dan *trap* yang harus dilalui pemain untuk bisa mengumpulkan semua item, ada juga waktu yang akan membuat pemain kalah jika waktu habis.



Gambar 7. Tampilan Layar Level 1 dan 2

Pada gambar 9 menunjukkan level 3 dimana pemain diharuskan mengalahkan boss untuk memenangkan *game*.



Gambar 8. Tampilan Layar Level 3

### 3.3 Hasil Pengujian

Pengujian kuesioner dilakukan menggunakan ISO 9126 untuk menjadi tolak ukur kualitas, mengambil 2 poin penilaian berupa *functionality* dan *usability*, tabel 5 menunjukkan nilai dari setiap jawaban responden.

**Table 5.** Tabel Nilai Jawaban Responden

Jawaban	Nilai
Sangat Setuju	5
Setuju	4
Cukup Setuju	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Tabel 6 menunjukkan presentase dari hasil *functionality*

**Table 6.** Nilai *Functionality*

No	Pertanyaan	ss	s	cs	ts	sts	presentase
1	Apakah karakter sangkuriang dapat berjalan dengan baik ?	40	40	15	0	0	82,6 %
2	Apakah musuh di dalam game ini berjalan dengan lancar ?	75	28	3	0	0	92,1 %
3	Apakah tombol di dalam game dapat digunakan sesuai fungsinya ?	40	36	18	0	0	81,7 %
4	Apakah tertarik untuk bermain lagi ?	35	44	9	4	0	80 %

Perhitungan nilai rata – rata : 
$$\text{Skor} = \frac{336,4 \times 100}{400} = 84,1\% \quad (1)$$

Tabel 7 menunjukkan presentase dari hasil *usability*

**Table 7.** Nilai *Usability*

No	Pertanyaan	ss	s	cs	ts	sts	Presentase
1	Sebelumnya apakah anda mengetahui cerita Sangkuriang ?	20	44	15	6	1	74,7 %
2	Apakah anda suka bermain game ?	45	32	9	5	0	77 %
3	Apakah tampilan dari game ini menarik ?	20	48	15	0	0	72 %
4	Apakah tantangan dalam game ini menarik ?	30	28	21	6	0	73,9 %
5	Apakah tertarik untuk bermain lagi ?	35	44	9	4	0	80 %
6	Apakah cerita sangkuriang tersampaikan dengan baik ?	70	24	6	2	0	88 %

Perhitungan nilai rata – rata : 
$$\text{Skor} = \frac{465,6 \times 100}{600} = 77,6\% \quad (2)$$

## 4. KESIMPULAN

Berdasarkan permasalahan dan hasil pengujian dari penelitian ini, dapat disimpulkan FSM diterapkan pada *Non Playable Character* (NPC) telah diuji dan hasilnya berjalan sesuai dengan yang diharapkan, hasil dari penelitian ini adalah 84,1 % untuk pengujian *functionality* dan 77,6% untuk pengujian *usability* yang didapat dari



jawaban kuesioner. Setelah melakukan proses perancangan dan pembuatan game maka dapat disimpulkan sebagai berikut :

- a. Dengan dibuatnya game “Sangkuriang : Simple Platform Game” maka dapat membantu mengenalkan cerita rakyat sangkuriang kepada para pemain dengan menarik dan kreatif.
- b. *Finite State Machine* diterapkan pada NPC, sehingga memiliki perilaku sendiri.
- c. Dengan adanya game “Sangkuriang : Simple Platform Game” dapat menghibur pemain.

## DAFTAR PUSTAKA

- [1] K. Nursa, J. Bahasa Jawa, F. Bahasa dan Seni, and U. Negeri Semarang, “Sutasoma: Journal of Javanese Literature INVENTARISASI CERITA RAKYAT DI KABUPATEN BANJARNEGARA,” vol. 3, no. 1, 2014, [Online]. Available: <http://journal.unnes.ac.id/sju/index.php/sutasoma>
- [2] M. Yoga Altoofa, T. Fatimah, D. Kusumaningsih, and W. Pramusinto, “IMPLEMENTASI FINITE STATE MACHINE PADA GAME ‘MALIN KUNDANG: SIMPLE PLATFORM GAME’ DENGAN UNITY GAME ENGINE,” vol. 2, no. 1, 2023.
- [3] R. Supardi and T. D. Putra, “RANCANG BANGUN GAME SUITCAKE BERBASIS ANDRODI DENGAN METODE ALGORITMA LINEAR CONGRUENT,” *Jl. Meranti Raya No. 32 Kota Bengkulu*, vol. 4, no. 1, p. 341139, 2020.
- [4] M. Meison, H. Setiawan, and N. Suhandi, “DESAIN GAME PLATFORMER PENGENALAN NAMA MATA UANG ASIA 2 (DUA) DIMENSI,” *Jurnal Informatika Global*, vol. 9, no. 1, 2018.
- [5] B. Fernando, D. A. Haris, and V. Christanti, “‘LEGATO LETS GO TO MUSIC’ PADA PLATFORM PC,” *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 6, no. 2, 2018, Accessed: Jul. 27, 2023. [Online]. Available: <https://journal.untar.ac.id/index.php/jiksi/article/view/2622>
- [6] R. A. Rahman and D. Tresnawati, “PENGEMBANGAN GAME EDUKASI PENGENALAN NAMA HEWAN DAN HABITATNYA DALAM 3 BAHASA SEBAGAI MEDIA PEMBELAJARAN BERBASIS MULTIMEDIA,” *urnal Algoritma Sekolah Tinggi Teknologi Garut*, vol. 13, no. 1, pp. 184–190, 2016, [Online]. Available: <http://jurnal.sttgarut.ac.id>
- [7] K. Teguh, “PENGEMBANGAN GAME DENGAN MENGGUNAKAN GAME ENGINE GAME MAKER,” *Jurnal Sistem Komputer*, vol. 5, no. 1, pp. 23–30, 2015.
- [8] I. W. Fauzy *et al.*, “GAME SOSIALISASI SEJARAH BANDUNG LAUTAN API DENGAN METODE MNEMONIK,” *NARATIF(Jurnal Ilmiah Nasional Riset Aplikasi dan Teknik Informatika)*, vol. 02, no. 2, 2020.
- [9] Z. Cahya, A. Wahana, S. A. Wibowo, and A. Wahid, “GAME ADVENTURE HORROR ‘LET’S ESCAPE’ DENGAN UNITY ENGINE BERBASIS DESKTOP MENGGUNAKAN METODE FINITE STATE MACHINE,” *Jurnal Mahasiswa Teknik Informatika*, vol. 4, no. 2, 2020.
- [10] D. Putra Pradana, “PENERAPAN METODE FINITE STATE MACHINE PADA GAME ‘ADVENTURE IN DARK TERRITORY,’” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 3, no. 2, 2019.