

IMPLEMENTASI FINITE STATE MACHINE (FSM) DAN LINE OF SIGHT (LoS) DALAM PERANCANGAN GAME 2D PETUALANGAN RADEN PANJI INU KERTAPATI

Latief Khanan Alqutbhi, Agung Panji Sasmito, Deddy Rudhistiar

Teknik Informatika, Institut Teknologi Nasional Malang

Jalan Raya Karanglo km 2 Malang, Indonesia

1818065@scholar.itn.ac.id

ABSTRAK

Pelestarian budaya lokal melalui media digital interaktif seperti *game* menjadi semakin penting di era modern, terutama untuk mengenalkan cerita rakyat kepada generasi muda agar mereka lebih tertarik mempelajari warisan budaya. Namun, pengembangan *game* yang menarik membutuhkan implementasi kecerdasan buatan (AI) yang efektif, khususnya untuk mengatur perilaku *NonPlayer Character* (NPC). Penelitian ini bertujuan mengembangkan *game* 2D berbasis Android berjudul "Petualangan Raden Panji Inu Kertapati" dengan mengimplementasikan metode *Finite State Machine* (FSM) dan *Line of Sight* (LoS) untuk memberikan kecerdasan buatan pada NPC agar menciptakan mekanisme permainan yang dinamis dan efisien. FSM digunakan untuk mengatur *state* dan transisi perilaku NPC, sementara LoS mengatur kemampuan NPC dalam mendeteksi keberadaan pemain berdasarkan garis pandang. Berdasarkan hasil pengujian fungsional didapat tingkat keberhasilan pemain berdasarkan garis pandang. Berdasarkan hasil pengujian fungsional didapat tingkat keberhasilan 100% sesuai. Pengujian metode FSM dan LoS didapat hasil 100% sesuai dan pengujian yang dilakukan kepada 15 *user* didapatkan hasil 84,44% pemain menyatakan *game* berjalan dengan baik. Jadi secara keseluruhan, sebagian besar pemain menyatakan bahwa *game* "Petualangan Raden Panji Inu Kertapati" berjalan dengan baik.

Kata kunci : *Game, Kecerdasan Buatan, Finite State Machine, Line of Sight, Android*

1. PENDAHULUAN

Game adalah media interaktif berbasis elektronik yang dirancang dengan tujuan menghibur dan memberikan tantangan kepada pemainnya. *Platformer* merupakan kategori *game* yang fokus pada pergerakan karakter utama dalam menjelajahi area permainan, di mana pemain harus memandu karakternya melintasi berbagai rintangan dengan aksi berlari dan melompat untuk sampai ke tujuan [1]. Kecerdasan buatan atau AI telah menjadi komponen vital dalam *game* masa kini, memungkinkan pemain untuk berinteraksi dengan sistem komputer yang mampu memberikan tantangan seperti saat bermain melawan manusia lainnya.

Dalam pengembangan *game*, teknologi kecerdasan buatan (AI) berperan vital dalam meningkatkan kualitas *gameplay*. Implementasi AI paling signifikan terlihat pada *Non-Player Characters* (NPC), yang mampu menampilkan perilaku kompleks dan adaptif. NPC yang diperkuat dengan sistem AI dapat menyajikan tantangan yang tepat, memberikan respon yang relevan terhadap tindakan pemain, dan menciptakan pengalaman bermain yang lebih mengasikkan. Oleh karena itu, penting untuk memilih metode AI yang mampu menghadirkan perilaku dinamis sekaligus efisien, terutama pada *platform* dengan keterbatasan *resources* seperti Android.

Finite State Machine (FSM) adalah model matematis yang digunakan untuk merepresentasikan sistem yang memiliki jumlah terbatas dari status (*state*) yang dapat diambil pada suatu waktu tertentu [2]. Dari penelitian yang dilakukan oleh [3] menunjukkan bahwa FSM memegang peranan penting dalam sistem permainan digital, terutama dalam mengelola perilaku karakter non-pemain (NPC) yang

berperan sebagai musuh. Implementasi FSM ini berkontribusi dalam menciptakan pengalaman bermain yang lebih menantang dan dinamis bagi pemain tunggal. Implementasi FSM oleh [4] didapat hasil bahwa FSM sudah terbukti bisa diterapkan dengan efektif, sehingga karakter musuh mempunyai tingkahlaku berdasarkan respons pemain.

Meskipun FSM memiliki keunggulan dalam kesederhanaan dan efisiensi, metode lain seperti *Behavior Tree* (BT) juga sering digunakan untuk memodelkan perilaku NPC dalam *game*. BT menyediakan struktur hierarkis untuk mengatur perilaku NPC dengan fleksibilitas tinggi. Namun, implementasinya cukup kompleks dan membutuhkan *resources* lebih besar, seperti dalam penelitian yang dilakukan oleh [5] menunjukkan bahwa meskipun BT mampu menciptakan NPC yang adaptif, namun waktu eksekusi kode meningkat seiring bertambahnya jumlah NPC. Dalam konteks platform Android yang memiliki keterbatasan *resources*, FSM menjadi pilihan yang lebih efisien karena strukturnya sederhana, mudah di-debug, dan hemat sumber daya.

Namun, FSM sendiri tidak cukup untuk menciptakan perilaku NPC yang realistis dan menantang. Dalam hal ini, *Line of Sight* (LoS) dapat menjadi metode pelengkap yang ideal. LoS memberikan kemampuan pada NPC untuk mendeteksi keberadaan pemain berdasarkan garis pandang, mirip dengan cara manusia melihat dalam dunia nyata. Penambahan LoS membuat perilaku NPC menjadi lebih masuk akal, karena NPC hanya akan bereaksi terhadap pemain ketika berada dalam jangkauan pandangannya.

Di tengah maraknya *game* modern yang mengadopsi cerita-cerita barat, pelestarian warisan budaya lokal melalui media interaktif menjadi semakin penting. Cerita Raden Panji Inu Kertapati dipilih secara khusus karena struktur naratifnya yang kaya akan petualangan heroik dan rintangan, sangat cocok untuk implementasi mekanisme FSM dan LoS. Karakter-karakter antagonis dalam cerita ini ideal untuk diimplementasikan sebagai NPC. Adaptasi cerita ini ke dalam bentuk *game* tidak hanya berkontribusi pada pelestarian budaya, tetapi juga memberikan cara yang menarik bagi generasi muda untuk mengenal dan mengapresiasi warisan budaya lokal melalui medium yang dekat dengan keseharian mereka.

Berdasarkan permasalahan diatas, muncul gagasan untuk mengembangkan sebuah *game* 2D berbasis Android yang mengimplementasikan kombinasi metode FSM dan LoS untuk menciptakan perilaku NPC yang lebih kompleks dan realistis. *Game* ini akan mengadaptasi cerita Raden Panji Inu Kertapati sebagai alur cerita, di mana pemain akan berinteraksi dengan berbagai NPC yang memiliki perilaku berbeda-beda. Implementasi FSM akan mengatur *state* dan transisi perilaku NPC, sementara LoS akan mengatur kemampuan NPC dalam mendeteksi keberadaan pemain berdasarkan garis pandang.

Penggunaan kedua metode ini diharapkan dapat menciptakan *gameplay* yang lebih menantang dengan adanya NPC yang dapat bereaksi secara dinamis terhadap tindakan pemain. FSM akan mengatur perilaku dasar NPC seperti patroli, mengejar, atau menyerang, sementara LoS akan menentukan kapan NPC dapat mendeteksi pemain dan mengubah *state*-nya. Kombinasi kedua metode ini memungkinkan terciptanya sistem AI yang lebih kompleks namun tetap efisien dalam implementasinya pada platform Android.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Dalam penelitian dengan judul Penerapan Line of Sight dan Finite State Machine Pada Game Platformer "RUN!" diterapkan metode *Line of Sight* saat membuat *game* sehingga karakter nonpemain tidak dapat menemukan pemain jika jarak pandang terhalang suatu objek dan metode *Finite State Machine* diterapkan agar karakter nonpemain bisa bergerak sendiri dan dapat saling berinteraksi dengan pemain. Dari hasil penelitian yang didapat adalah kedua metode tersebut dapat diterapkan dengan baik, dimana nonpemain yang diterapkan metode *Finite State Machine* (FSM) dapat berganti *state* dari *idle* ke *Attack* atau sebaliknya dengan *output* dari metode *Line of Sight* (LoS) [6].

Dalam penelitian dengan judul Rancang Bangun *Game* Berbasis Android Bertemakan Cerita Rakyat Betawi Si Pitung. Permainan ini mengisahkan perlawanan Si Pitung terhadap penguasa kolonial Belanda dengan cara merampok harta mereka demi

membantu kehidupan kaum miskin. Dibuat dalam bentuk visual 2D menggunakan *software Unity* yang dimana target pemain adalah anak dibawah 15 tahun. Dari hasil penelitian yang dilakukan dapat berfungsi dengan baik 100% berdasarkan uji fungsionalitas, uji usabilitas, dan uji portabilitas yang dilakukan [7]

Dalam penelitian dengan judul Perancangan *Game* 2d "Barbossa The Savior" Menggunakan Finite State Machine. Penelitian ini memiliki tujuan untuk mengimplementasikan kecerdasan buatan dalam *video game* dengan metode FSM pada karakter-karakter *game* sehingga *game* lebih menarik dimainkan dan memperkaya *gameplay*. Hasil penelitian dari penerapan metode FSM pada karakter di dapat hasil semuanya berhasil dan berdasarkan survei *user*, diperoleh data bahwa 12,59% responden memberikan respon Sangat Setuju, 34,44% menyatakan Setuju, dan 31,48% memberikan penilaian Cukup [8].

Dalam penelitian dengan judul Perancangan *Game* Alien Warfare 2d Menggunakan Metode Finite State Machine (FSM) Berbasis Android dimana *game* ini dirancang menggunakan *Unity Engine* dan pada karakter nonpemain atau musuh diterapkannya AI dengan menggunakan metode FSM. Metode ini diterapkan untuk memberikan kecerdasan buatan agar karakter musuh dapat bergerak mendekati pemain dan menyerang. Hasil pengujian fungsionalitas pada setiap fitur, serta pengujian untuk *movement player* juga sepenuhnya berhasil. Selain itu, dari pengujian pengguna, diperoleh persentase sebesar 86% dari 15 *user* yang mengindikasikan bahwa kinerja *game* dinilai baik. Sementara itu hasil pengujian metode FSM untuk karakter musuh semuanya dinilai berhasil,[9].

2.2. Video Game

Video game merupakan bentuk hiburan interaktif yang dapat diakses melalui beragam perangkat elektronik, seperti perangkat khusus *gaming* seperti konsol, komputer, hingga perangkat mobile seperti smartphone. Di era modern ini, *video game* telah menjadi pilihan populer untuk mengisi waktu luang dan melepas penat, terutama dalam format mobile yang praktis dibawa kemanapun. Berbagai jenis *game* kini tersedia dan dapat diakses dengan mudah melalui toko aplikasi di smartphone, menjadikannya pilihan hiburan yang praktis dan mudah dijangkau [4].

2.3. Kecerdasan Buatan

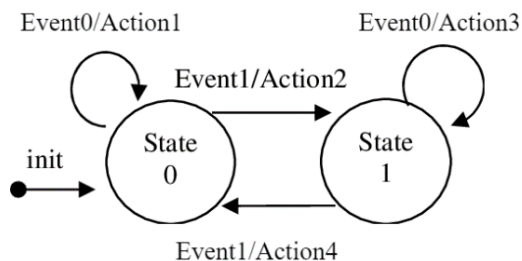
Artificial Intelligence (AI) Kecerdasan buatan dalam bahasa Indonesia adalah bagian ilmu komputer yang berfokus pada cara mengendalikan komputer agar mampu membantu atau bahkan menggantikan tugas yang biasanya dilakukan oleh manusia [10].

Kecerdasan Buatan dalam *game* memainkan peran penting dalam meningkatkan pengalaman pemain. Saat ini, AI kian marak digunakan dan diterapkan di beragam bidang kehidupan, termasuk dalam pembuatan *video game*. Dalam konteks *game*, AI digunakan untuk mengendalikan berbagai elemen

dalam permainan, memungkinkan komputer untuk mengambil keputusan secara cerdas ketika menghadapi situasi dengan berbagai opsi sehingga memiliki *output* yang berbeda-beda. Hal ini dapat menghasilkan tingkahlaku yang cocok dan optimal didalam *game*, sehingga meningkatkan interaksi dan tantangan bagi pemain.

2.4. Metode Finite State Machine

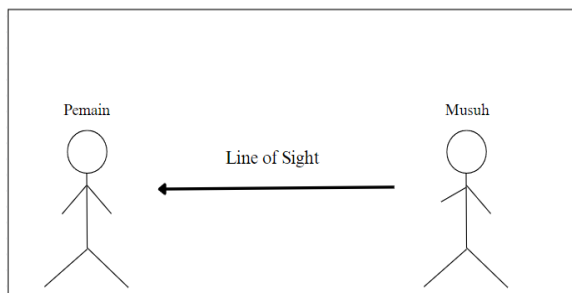
Finite State Machine (FSM) merupakan sebuah model untuk merancang sebuah sistem yang dapat mendeskripsikan tingkahlaku sistem menggunakan tiga komponen utama: *State* (kondisi sistem), *Event* (pemicu perubahan), dan *Action* (tindakan yang dilakukan). Sistem ini bekerja layaknya sebuah mesin dengan beberapa kondisi yang berbeda, dimana pada suatu waktu tertentu hanya satu kondisi yang dapat aktif. Untuk melakukan aksi yang berbeda, mesin tersebut perlu melakukan transisi dari satu kondisi ke kondisi lainnya [4].



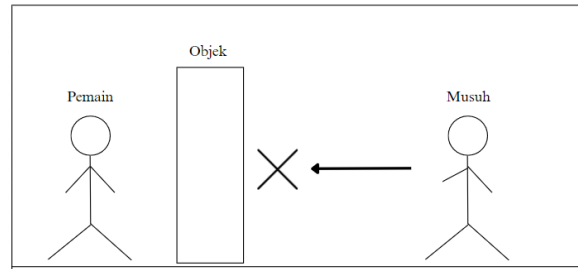
Gambar 1. Diagram *finite state machine*

2.5. Metode Line of Sight

Line Of Sight (LoS) adalah konsep yang menggambarkan kemampuan karakter atau objek untuk melihat atau mendeteksi objek lain dalam jarak pandangnya. Algoritma LoS bekerja dengan menarik garis lurus dari NPC (*Non-Player Character*) ke arah pemain. Jika garis ini mencapai pemain tanpa terhalang oleh objek seperti dinding atau penghalang lainnya, maka NPC dapat melihat pemain. Sebaliknya, jika garis terhalang oleh objek, NPC tidak dapat melihat pemain. Konsep ini melibatkan garis yang harus bebas dari halangan fisik untuk memastikan bahwa karakter atau objek dapat mendeteksi target yang diinginkan [6].



Gambar 2. Pemain terlihat oleh musuh



Gambar 3. Pemain tidak terlihat oleh musuh

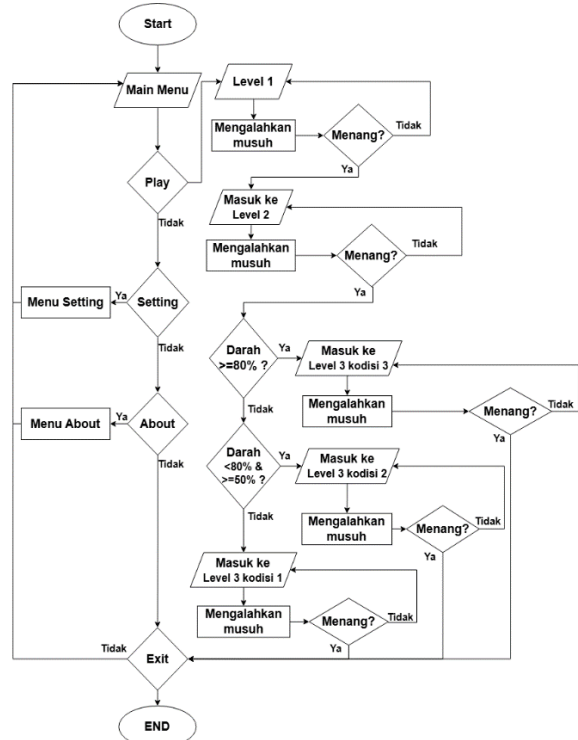
2.6. Unity Engine

Unity merupakan platform pengembangan *game* yang menyediakan dukungan untuk pembuatan *game* 2D maupun 3D. Dengan fitur lintas platform yang dimiliki, Unity memungkinkan pengembang untuk membuat *game* yang dapat dijalankan di beragam sistem operasi, mulai dari *Windows* hingga perangkat *mobile* seperti *Android*. Unity menggunakan C# sebagai bahasa pemrograman utama, meskipun juga mendukung JavaScript. Unity menggunakan teknologi grafis mutakhir untuk OpenGL dan DirectX, kompatibel dengan berbagai format *file* dari aplikasi grafis, dan berfungsi pada sistem 64-bit di macOS dan Windows. Platform ini juga dapat digunakan untuk menghasilkan *game* untuk berbagai perangkat [11].

3. METODE PENELITIAN

Pada bagian ini berisikan metode-metode yang digunakan dalam pembuatan permainan ini.

3.1. Flowchart Game



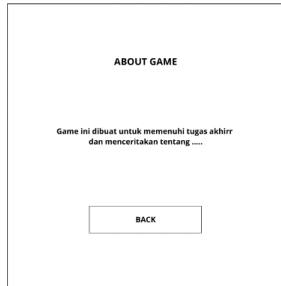
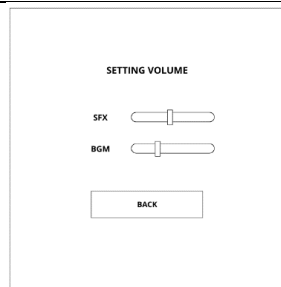
Gambar 4. Flowchart game

Pada Gambar 4 atas, proses pertama dimulai dari main menu yang berisi opsi *Play*, *Setting*, *About*, dan

Exit. Jika *player* memilih *Play*, permainan akan menampilkan prolog *scene* yang berisi cerita awal permainan. Setelah itu, *player* masuk ke *map level 1* dan mulai bermain. Jika semua musuh berhasil dikalahkan, *player* akan melanjutkan ke *map level 2*. Namun, jika gagal, *player* harus mengulang kembali dari awal *map level 1*. Di *map level 2*, mekanisme permainan serupa dengan *map level 1*, tetapi musuh menjadi lebih kuat. Saat memasuki *map level 3*, kondisi permainan akan menyesuaikan dengan jumlah darah karakter yang tersisa dari *map level 12*. Semakin banyak darah yang tersisa, semakin sulit musuh untuk dikalahkan di *map level 3*. Di *Main Menu*, jika *player* memilih *Setting*, maka akan muncul pengaturan suara. Jika memilih *About*, halaman informasi tentang pengembangan permainan akan ditampilkan. Sedangkan jika memilih *Exit*, *player* akan keluar dari permainan dan kembali ke *Home Android*.

3.2. Storyboard

Table 1 Storyboard

No	Desain
1	 Menu utama dari game "Petualangan Raden Panji Inu Kertapati". Pada <i>Main Menu</i> ada 4 menu, yaitu <i>Play</i>, <i>About</i>, <i>Setting</i> dan <i>Exit</i>.
2	 Jika <i>player</i> memilih <i>button About</i> maka diarahkan pada menu <i>about</i>, yang terdapat isi tentang game dan ada <i>button back</i> untuk kembali ke menu utama
3	 Menu untuk mengatur suara pada game dengan menggeser <i>slider</i> yang ada dan ada tombol <i>back</i> untuk kembali ke <i>main menu</i>.

No	Desain
4	 Gambar diatas merupakan tampilan permainan terdapat kembali untuk bergerak, lompat, dan menyerang.

3.3. Gameplay

a. Tujuan

Tujuan utama dari game Petualangan Raden Panji Inu Kertapati adalah untuk mengalahkan boss pada *map* terakhir dengan cara mengalahkan semua antek boss pada *map-map* sebelumnya.

b. Start

Pemain akan memasuki *main menu* sebelum game dimulai, pemain bisa memilih menu lain yang tersedia, yaitu menu *play*, menu *about*, menu *setting*, dan menu *exit*.

c. Middle

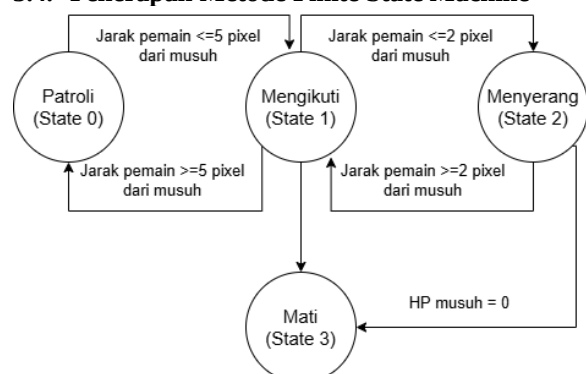
Pada bagian *middle* ini akan ditunjukkan proses *gameplay* dari game Petualangan Raden Panji Inu Kertapati pada *level ke level*.

- Pemain melawan *enemy* yang tersebar pada tiap *map* setelah mengalahkan semua *enemy* pemain pergi ke *map* selanjutnya.
- Serangan pemain yang mengenai *enemy* akan mengurangi jumlah nyawa *enemy* dan akan dikalahkan jika nyawa menjadi 0.
- Serangan musuh yang mengenai pemain akan mengurangi jumlah nyawa dan pemain akan dikalahkan apabila nyawanya menjadi 0.

d. Ending

Setelah menyelesaikan *level* terakhir pemain akan mendapat harta pusaka sesuai dengan kondisi saat menyelesaikan *level* ke 3 dan pemain dapat mengulangi permainan.

3.4. Penerapan Metode Finite State Machine

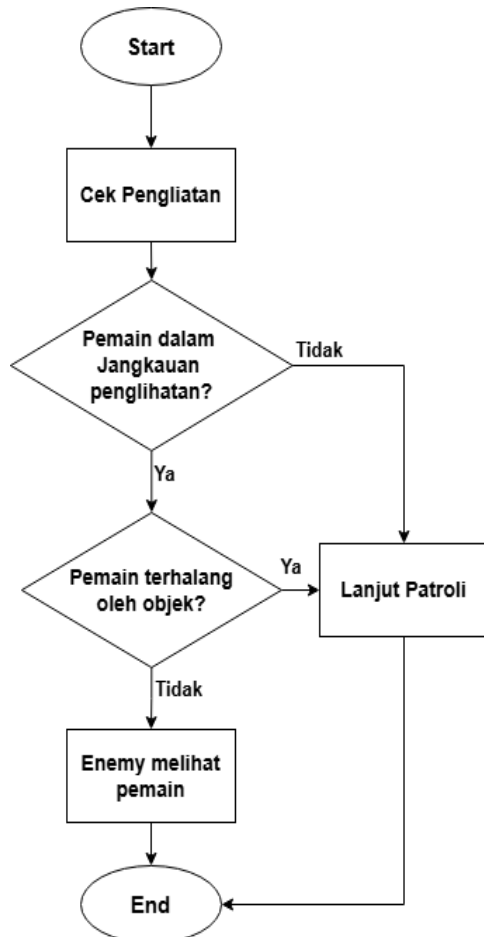


Gambar 5. Flowchart finite state machine

Penerapan alur FSM pada karakter musuh memungkinkan musuh merespons pergerakan pemain secara dinamis. Jika pemain mendekat hingga berada dalam jarak ≤ 5 pixel, musuh akan mulai mengejar.

Ketika jarak semakin kecil hingga $\leq 2 \text{ pixel}$, musuh akan langsung menyerang. Namun, jika pemain menjauh dan berada $\geq 5 \text{ pixel}$, musuh akan berhenti mengejar dan kembali ke mode patroli.

3.5. Penerapan Metode Line of Sight



Gambar 6. Flowchart line of sight

Pada Gambar 6. diatas proses dimulai saat NPC melakukan pemeriksaan penglihatan untuk mendeteksi keberadaan pemain. NPC kemudian memeriksa apakah pemain berada dalam jangkauan penglihatannya. Jika pemain tidak berada dalam jangkauan, NPC akan melanjutkan aktivitas patrolinya seperti biasa. Namun jika pemain terdeteksi dalam jangkauan, NPC akan melakukan pengecekan lanjutan untuk menentukan apakah ada objek yang menghalangi pandangannya ke arah pemain. Ketika ada halangan, NPC tidak dapat melihat pemain meskipun dalam jangkauan, sehingga NPC melanjutkan aktivitas patrolinya. Sebaliknya, jika tidak ada halangan, NPC akan berhasil melihat pemain. Proses ini berakhir setelah NPC mengambil keputusan berdasarkan kondisi yang terdeteksi

3.6. Desain Karakter

Table 2. Desain karakter

Karakter	Keterangan
	Raden Panji sebagai karakter utama permainan ini dan digerakkan oleh <i>player</i>
	Antek Petapa Jahat sebagai karakter musuh paling lemah dapat menyerang ketika <i>player</i> mendekat.
	Antek Petapa Jahat Elit sebagai karakter musuh yang dapat menyerang ketika <i>player</i> mendekat ketika <i>player</i> mendekat..
	Petapa Jahat sebagai karakter musuh yang paling kuat atau <i>Boss</i> dimana dapat menyerang ketika <i>player</i> mendekat ketika <i>player</i> mendekat

Table 3. Desain Environment & UI

Asset	Keterangan
	<i>Environment</i> batu untuk jadi <i>obstacle</i> pada game ini.
	<i>Environment</i> tanah pada game ini
	<i>UI Attack Button</i> untuk serangan <i>player</i>
	<i>UI Arrow Button</i> untuk 3546embali ke kanan <i>player</i>
	<i>UI Jump Button</i> untuk gerakan ke kanan <i>player</i> .

4. HASIL DAN PENGUJIAN

Pada bagian ini memuat hasil implementasi dan pengujian.

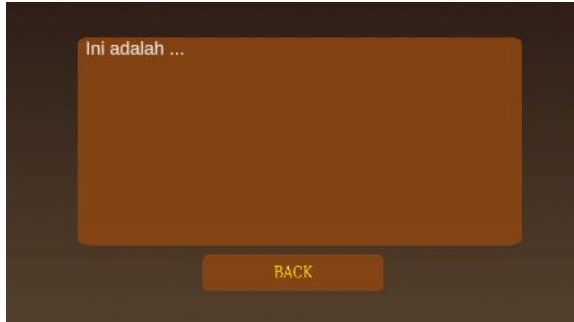
4.1. Menu Utama



Gambar 7. Tampilan main menu game

Pada Gambar 7 diatas yaitu menu utama dari *game* dimana terdapat *button Play*, *About*, *Setting*, dan *Exit*. *Button Play* digunakan untuk memulai permainan, *Button About* berisi penjelasan mengapa *game* ini dibuat, *Button Setting* berisi pengaturan suara *game*, dan *Button Exit* untuk keluar dari permainan.

4.2. Menu About



Gambar 8. Tampilan menu *about*

Pada Gambar 8 diatas yaitu menu *About* berisi teks penjelasan mengapa *game* ini dibuat dan terdapat *button Back* untuk kembali ke *Main Menu*.

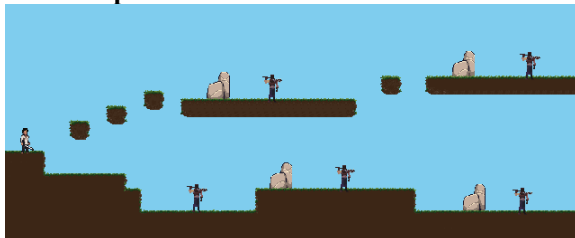
4.3. Menu Settings



Gambar 9. Tampilan menu *setting*

Pada Gambar 9 diatas yaitu menu *Setting* berisi *slider* untuk mengatur suara *bgm* dan *sfx* dan terdapat *button Back* untuk kembali ke *Main Menu*.

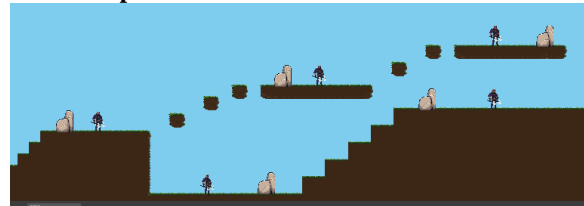
4.4. Tampilan Level 1



Gambar 10. *Map level 1*

Pada Gambar 10 diatas yaitu tampilan *map level 1* dimana terdapat 5 musuh yang harus dikalahkan untuk melanjutkan ke *map* selanjutnya.

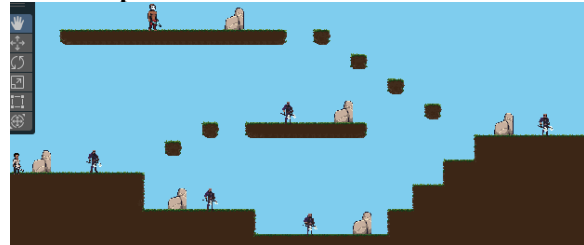
4.5. Tampilan Level 2



Gambar 11. *Map level 2*

Pada Gambar 11 diatas yaitu tampilan *map level 2* dimana terdapat 5 musuh *elit* yang lebih kuat dari *map* sebelumnya dan harus dikalahkan untuk melanjutkan ke *map* selanjutnya.

4.6. Tampilan Level 3



Gambar 12. *Map level 3*

Pada Gambar 12 diatas yaitu tampilan *map permainan level 3* dimana terdapat 5 musuh *elit* dan 1 *Boss* yang harus dikalahkan untuk menyelesaikan *game*.

4.7. Tampilan Ending



Gambar 13. *Ending game*

Pada Gambar 13 diatas yaitu tampilan *ending game* dimana berisi *ending* dan ucapan *terimakasih* telah memainkan *game*.

4.8. Pengujian Main Menu

Table 4. Pengujian menu utama

No	Fitur Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1.	Tombol "PLAY" pada	Menekan tombol " PLAY "	Game menampilkan <i>StroryProlog</i>	Sesuai
2.	Tombol "ABOUT"	Menekan tombol " ABOUT "	Game menampilkan menu about	Sesuai
3.	Tombol " SETTING "	Menekan tombol " SETTING "	Game menampilkan menu <i>setting</i>	Sesuai
4.	Tombol " EXIT "	Menekan tombol " EXIT"	Keluar dari <i>game</i>	Sesuai

Dari pengujian pada Tabel 4 diatas, diuji 4 fitur dan didapatkan hasil 100% sesuai yang diharapkan dimana tombol *play*, *about*, *setting*, dan *exit* berjalan dengan semestinya.

4.9. Pengujian Menu About

Table 5. Pengujian menu *about*

No.	Fitur Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1.	Tampilan teks penjelasan	menu about menampilkan teks penjelasan	teks penjelasan ditampilkan	Sesuai
2.	Tombol " BACK "	Menekan tombol " BACK "	Game kembali ke main menu	Sesuai

Dari pengujian pada Tabel 5 diatas, diuji 2 fitur dan didapatkan hasil 100% sesuai yang diharapkan dimana informasi tentang *game* bisa didapatkan dan tombol *back* berfungsi dengan baik.

4.10. Pengujian Menu Setting

Table 6. Pengujian menu *setting*

No.	Fitur Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1.	<i>Slider</i> pada <i>soundfx</i> .	Menggeser <i>slider</i> kekiri dan kekanan	Kekiri suara efek semakin kecil dan sebaliknya	Sesuai
2.	<i>Slider</i> pada bgm	Menggeser <i>slider</i> kekiri dan kekanan	Kekiri suara bgm semakin kecil dan sebaliknya	Sesuai
3.	Tombol " BACK "	Menekan tombol " BACK "	Game Kembali ke <i>main s</i>	Sesuai

Dari pengujian pada Tabel 6 diatas, diuji 3 fitur dan didapatkan hasil 100% sesuai yang diharapkan dimana *player* dapat mengatur *volume sound fx* dan bgm dengan cara menggeser *slider*.

4.11. Pengujian Control Player

Table 7. Pengujian *Control Player*

No.	Tombol	Fungsi	Hasil
1.		Jalan ke arah kiri.	Sesuai
2.		Jalan ke arah kanan.	Sesuai
3.		Pemain melompat.	Sesuai
4.		Pemain menyerang.	Sesuai

Dari pengujian pada Tabel 7 diatas, diuji 4 fitur didapatkan hasil 100% sesuai yang diharapkan dimana semua kontrol *player* bisa dijalankan dengan baik.

4.12. Pengujian Finite State Machine

Table 8. Pengujian *Finite State Machine*

No	State	Hasil Yang Diharapkan	Hasil
1	<i>Enemy</i> melakukan patroli	<i>Enemy</i> patroli sesuai radius yang ditentukan.	Sesuai
2	<i>Enemy</i> masuk <i>state chase</i>	Saat <i>player</i> mendekat ke <i>enemy</i> dalam radius <i>state chase enemy</i> masuk ke <i>state chase</i> .	Sesuai
3	<i>Enemy</i> keluar <i>state chase</i>	Saat <i>player</i> menjauh dari <i>enemy</i> dalam radius <i>state chase enemy</i> kembali ke <i>state patroli</i>	Sesuai
4	<i>Enemy</i> masuk <i>state attack</i> .	Saat <i>player</i> mendekat ke <i>enemy</i> dalam radius <i>state attack enemy</i> masuk ke <i>state attack</i> .	Sesuai
5	<i>Enemy</i> keluar <i>state attack</i>	Saat <i>player</i> menjauh dari <i>enemy</i> dalam radius <i>state attack enemy</i> masuk ke <i>state chase</i>	Sesuai
6	<i>Enemy</i> masuk <i>state death</i> .	Saat nyawa <i>enemy</i> dimiliki 0.	Sesuai

Dari pengujian pada Tabel 8 diatas diuji 4 *state* dengan 6 kondisi pada *enemy* dimana didapatkan hasil 100% sesuai yang diharapkan sehingga implementasi

metode *Finite State Machine* pada *enemy* untuk memberikan AI berhasil diimplementasikan.

4.13. Pengujian Line of Sight

Table 9. Pengujian *Line of Sight*

No	Posisi Player	Dalam pandangan	Obstacle	Hasil
1	Didepan <i>Enemy</i>	Tidak	Tidak	<i>Enemy</i> tidak melihat <i>player</i> .
2	Didepan <i>Enemy</i>	Tidak	Ya	<i>Enemy</i> dapat melihat <i>player</i> .
3	Didepan <i>Enemy</i>	Ya	Tidak	<i>Enemy</i> tidak melihat <i>player</i> .
4	Didepan <i>Enemy</i>	Ya	Ya	<i>Enemy</i> tidak melihat <i>player</i> .
5	Dibelakang <i>Enemy</i>	Tidak	Tidak	<i>Enemy</i> tidak melihat <i>player</i> .
6	Dibelakang <i>Enemy</i>	Tidak	Ya	<i>Enemy</i> tidak melihat <i>player</i> .
7	Dibelakang <i>Enemy</i>	Ya	Tidak	<i>Enemy</i> tidak melihat <i>player</i> .
8	Dibelakang <i>Enemy</i>	Ya	Ya	<i>Enemy</i> tidak melihat <i>player</i> .

Dari pengujian pada Tabel 9 diatas, diatas diuji dengan berbagai kondisi untuk melihat apakah *line of sight* sudah diterapkan dengan baik dan didapatkan

hasil pengujian 100% sesuai dengan yang diharapkan sehingga penerapan metode *Line of Sight* pada *enemy* berhasil diimplementasikan dengan baik.

4.14. Pengujian User

Table 4.10 Pengujian Responden

No	Pertanyaan	Penilaian	
		Ya	Tidak
1	Apakah story game tersampaikan dengan baik?	13	2
2	Apakah kontrol game mudah dipahami ?	15	0
3	Apakah desain karakter dapat diterima ?	11	4
4	Apakah kualitas musik dan sound effect game dapat diterima ?	10	5
5	Apakah Tingkat kesulitan game sudah sesuai?	14	1
6	Apakah game berjalan dengan lancar pada perangkat anda ?	13	2
Total		76	14
Persentase (%)		84,44	15,56

Pengujian *game* ini melibatkan 15 responden dengan usia 13 tahun keatas. Responden terdiri dari kalangan pelajar dan umum yang tertarik bermain *game* 2D. Semua responden memiliki perangkat Android dengan spesifikasi minimal RAM 2GB dan OS Android 8.0, yang memadai untuk menjalankan *game* yang diuji. Dari tabel 4.10 diatas 15 orang yang memainkan *game* “Petualangan Raden Panji Inu Kertapati” menunjukkan bahwa 84.44% menyatakan *game* berjalan dengan baik.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dari perancangan dan implementasi pada *game* “Petualangan Raden Panji Inu Kertapati” maka dapat diambil kesimpulan bahwa penerapan metode *Finite State Machine* dan *Line of Sight* pada karakter musuh berhasil diterapkan dimana didapat hasil pengujian dari berbagai aspek dengan tingkat keberhasilan 100% sesuai, sehingga 2 metode ini dinilai berhasil diimplementasikan ke dalam *game*, dari pengujian fungsional berbagai fitur di menu *game* serta pengujian *control player* juga didapat tingkat keberhasilan 100% sesuai, sehingga *game* dinilai sudah

berjalan dengan baik. Berdasarkan hasil kusioner dari 15 orang menunjukkan bahwa 84,44% menyatakan *game* berjalan dengan baik. Jadi secara keseluruhan, sebagian besar pemain menyatakan bahwa *game* "Petualangan Raden Panji Inu Kertapati" sudah berjalan dengan baik. Untuk pengembangan lebih lanjut, disarankan menambah jumlah *level* dan meningkatkan *enviroment* dalam *game*.

DAFTAR PUSTAKA

- [1] Senoaji, A. H., Haryanto, H., and Novianto, S. (2023). Implementasi behaviour Tree untuk mengatur perilaku NPC musuh pada game 2D platformer `Cyberun`. *Techno.Creative*, 1(2)
- [2] Ariobimo Wijaya D, Santi Wahyuni F, and Xaverius Ariwibisono F. Game 3D "Creature Tactic" Dengan Genre Tactical RPG Menggunakan Metode Algoritma A*. *JATI (Jurnal Mhs Tek Inform.* 2023;7(1):859-864
- [3] Abiyu, A. (2023). IMPLEMENTASI FSM (FINITE STATE MACHINE) PADA GAME MALIK LOOKS FOR THE HOLY BOOK. *JASTEN (Jurnal Aplikasi Sains Teknologi Nasional)*, 4(2), 61–67
- [4] Satrio, I., Wahyuni, F. S., and Rudhistiar, D. 2022. PENERAPAN A* PATHFINDING DAN FSM (FINITE STATE MACHINE) PADA GAME "LOST CIVILIZATION" BERBASIS ANDROID. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2).
- [5] Rifqi, A. (2020). Penerapan Metode Behaviour Tree Pada NPC Mu.uh Dalam Game 2090 Sebagai Pendidikan Moral Pancasila (Skripsi). Universitas Islam Negeri Maulana Malik Ibrahim Malang
- [6] Rangan, A. Y. 2023. Penerapan Line of Sight dan Finit State Machine pada *Game* Platformer "RUN!". *Digital Transformation Technology*, 3(1), 236-247
- [7] Hoesen, N. 2021. Rancang Bangun Game Berbasis Android Bertemakan Cerita Rakyat Betawi Si Pitung. *Jurnal Esensi Infokom Jurnal Esensi Sistem Informasi Dan Sistem Komputer*, 5(2), 32–37
- [8] Prasetyo, R. A., Sasmito, A. P., and Auliasari, K. (2023). PERANCANGAN GAME 2D "BARBOSSA THE SAVIOR" MENGGUNAKAN FINITE STATE MACHINE. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(5), 2999-3006
- [9] Wicaksono, B. A., Mahmudi, A., and Auliasari, K. 2023. PERANCANGAN GAME ALIEN WARFARE 2D MENGGUNAKAN METODE FINITE STATE MACHINE (FSM) BERBASIS ANDROID. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(5)
- [10] Syandana, M. A. A., Zahro, H. Z., and Achmadi, S. (2023). PERANCANGAN GAME MUTES DENGAN MENGGUNAKAN METODE FINITE STATE MACHINE BERBASIS ANDROID. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(4), 2167-2174
- [11] Uliontang, E. S., and Chandra, F. H. 2020. Pemanfaatan augmented reality pada media pembelajaran sejarah tentang benda-benda bersejarah peninggalan Kerajaan Majapahit di Trowulan Mojokerto. *Tek. Eng. Sains J*, 4(1), 19-26