

## PERANCANGAN GAME 2D “BARBOSSA THE SAVIOR” MENGGUNAKAN FINITE STATE MACHINE

**Rizki Aditya Prasetyo, Agung Panji Sasmito, Karina Auliasari**  
Program Studi Teknik Informatika S1, Fakultas Teknologi Industri  
Institut Teknologi Nasional Malang, Jalan Raya Karanglo km 2 Malang, Indonesia  
1918124@scholar.itn.ac.id

### ABSTRAK

Saat ini perkembangan industri *game* dalam beberapa tahun terakhir menghasilkan berbagai *genre* permainan menarik, termasuk permainan 2D yang terus populer dan menghibur pemain dari berbagai kalangan. Perkembangan *game Android* yang semakin pesat memberikan peluang kepada generasi muda untuk membuat *game* sendiri dengan menggunakan *software* dan teknologi canggih, seperti *Unity* yang merupakan sebuah *game engine* lintas *platform* yang banyak digunakan pengguna *game* di *Google Play*. Meskipun *game* saat ini menawarkan pengalaman yang menghibur dan menarik, terdapat kekurangan seperti kurangnya keterlibatan pemain yang berkelanjutan, kurangnya variasi dalam *gameplay*, dan kurangnya inovasi dan kreativitas dalam desain permainan. Tujuan dari pembuatan *game 2D Barbossa The Savior* ini agar dapat membangun, merancang serta mempelajari penerapan metode *Finite State Machine* untuk kecerdasan *game* pada *Non Player Character* menggunakan *unity engine* yang di terapkan pada semua level. Hasil implementasi pengujian beberapa *device Android* berjalan sesuai dengan tingkat keberhasilan yang diharapkan dan pengujian *user* yang dilakukan oleh 30 responden didapatkan hasil persentase yakni 12,59% menyatakan Sangat Setuju, 34,44% menyatakan Setuju, 31,48% menyatakan Cukup, 18,44% menyatakan Tidak Setuju dan 3,33% menyatakan Sangat Tidak Setuju. Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa *game* yang dirancang sudah baik dan mayoritas pengguna merasa puas dengan *game* yang telah dirancang.

**Kata kunci** : *Android, Finite State Machine, Game*

### 1. PENDAHULUAN

Saat ini perkembangan industri *game* telah mengalami perkembangan pesat dalam beberapa tahun terakhir. Hal ini menghasilkan beragam *genre* permainan yang menarik dan menghibur bagi pemain dari berbagai kalangan. Salah satu jenis permainan yang terus meraih popularitas adalah *game 2D*. *Game* menjadi pilihan hiburan favorit banyak orang, membantu dalam melatih fokus, meningkatkan konsentrasi, melatih kesabaran, dan mengisi waktu luang. Sebelumnya, *game* dimainkan dengan cara konvensional sebelum berkembang pada berbagai *platform*. Namun, kini *game* dikembangkan dan dirancang untuk berjalan di berbagai *platform* menggunakan teknologi modern. Semakin banyak permainan baru yang bermunculan dalam berbagai *genre*. Dengan demikian, terbentuklah klasifikasi permainan yang berbeda-beda yang disesuaikan dengan minat para pemain. [1]

Perkembangan *game android* yang saat ini berkembang pesat, memberikan peluang kepada generasi muda dalam membuat *game* sendiri dengan menggunakan perangkat lunak dan teknologi yang canggih. Saat ini, terdapat banyak perangkat lunak *game* yang memiliki fitur-fitur yang mudah dipahami untuk menciptakan animasi yang dapat diintegrasikan ke dalam permainan. Salah satu contohnya adalah *platform Unity*. *Unity* sendiri merupakan *game engine* lintas *platform* yang banyak digunakan pengguna *game* di *Google Play*. *Unity* dapat membantu dalam

memproduksi dan *me-render game 2D* atau *3D* berbasis desktop maupun *android* [2]

Meskipun *game* saat ini menawarkan pengalaman yang menghibur dan menarik, ada beberapa kekurangan dan kesulitan yang dapat diidentifikasi dalam banyak permainan. Salah satu kekurangan umum adalah kurangnya tingkat keterlibatan pemain yang berkelanjutan. Beberapa *game* mungkin memberikan tantangan awal yang menarik, tetapi setelah beberapa waktu, pemain mungkin merasa repetitif dan kehilangan minat. Ini dapat disebabkan oleh kurangnya variasi dalam *gameplay* untuk menyesuaikan tingkat kesulitan dengan kemampuan individu pemain. Selain itu, kurangnya inovasi dan kreativitas dalam desain permainan juga dapat menjadi masalah. Hal ini dapat menyebabkan kejenuhan dan kehilangan minat pemain dalam jangka waktu yang lebih lama

Berdasarkan permasalahan yang ada, penulis menjalankan sebuah penelitian dengan judul "Perancangan Permainan 2D Berjudul 'Barbossa The Savior' Menggunakan Pendekatan Metode Finite State Machine". Permainan ini merupakan sebuah permainan petualangan berjenis cerita rakyat dengan *genre* aksi, yang mengisahkan karakter utama yang terlibat dalam sebuah misi penyelamatan ikan mas dengan cara mengatasi musuh-musuh yang terdapat pada setiap level permainan.

## 2. TINJAUAN PUSTAKA

### 2.1. Penelitian Terdahulu

Dalam penelitian yang berjudul “Rancang Bangun *Game* 2D Shooter Menggunakan Metode Finite State Machine” yang bertujuan untuk menerapkan pendekatan *Finite State Machine* pada karakter *Non-Playable Character* (NPC), yakni karakter seperti musuh (*Enemy*) dan bos (*Boss*) dalam permainan. Hasil pengujian implementasi *Finite State Machine* pada permainan tembak-tembakan 2D ini menunjukkan bahwa sebagian besar responden, sekitar 32%, memberikan penilaian "Sangat Bagus," 41% memberikan penilaian "Bagus," 24% memberikan penilaian "Biasa Saja," dan hanya 3% memberikan penilaian "Buruk." [3]

Dalam penelitian yang berjudul “Pembuatan *Game* 2D “Escape Plan” Dengan Metode Finite State Machine” bertujuan untuk melengkapi kecerdasan buatan, dimana dengan bantuan kecerdasan buatan, pemain dan karakter musuh dapat berinteraksi satu sama lain dan lebih realistis, apalagi menggunakan metode FSM (*Finite State machine*) yang dapat mendeteksi keberadaan karakter. Dari pengujian yang dilakukan oleh karakter-karakter yang ada di dalam game mampu mendeteksi musuh dan menentukan tindakan yang akan dilakukan sesuai metode FSM yang diterapkan, dan hasil pengujiannya mencapai 6100\$ terhadap karakter musuh. Namun penelitian ini tetap mencakup penambahan metode berbeda pada karakter musuh dan juga dapat dikembangkan berdasarkan platform Android atau iOS. [4]

Dalam penelitian yang berjudul “Penerapan A\* Pathfinding Dan FSM (Finite State Machine) Pada *Game* “Lost Civilization” Berbasis Android” bertujuan untuk menerapkan metode pathfinding A\* untuk mencari posisi pemain dengan menjumlahkan jarak minimum pergerakan setiap node dan menggunakan metode FSM (*Finite State Machine*) untuk menghasilkan perilaku musuh. Dari hasil pengujian yang diperoleh, penerapan metode algoritma A\* dengan akurasi 100% untuk mencari posisi pemain untuk menentukan rute tercepat dan FSM pada musuh dapat bertindak berdasarkan perilaku pemain. [5]

### 2.2. Game

*Game* adalah suatu bentuk permainan yang dibuat untuk komputer dengan memanfaatkan teknik dan metode animasi. Untuk mendalami penerapan animasi, penting untuk memiliki pemahaman dalam pembuatan *game*. Bila tujuan Anda adalah menciptakan permainan, maka akan sangat diperlukan pemahaman yang kuat mengenai teknik dan metode animasi, karena keduanya saling memiliki hubungan yang erat [6]

*Game* berasal dari Bahasa Inggris yang berarti permainan. Setiap permainan memiliki aturan unik yang beragam untuk mengawali aktivitasnya, menghasilkan variasi yang kaya dalam jenis permainan. Karena peran permainan untuk meredakan

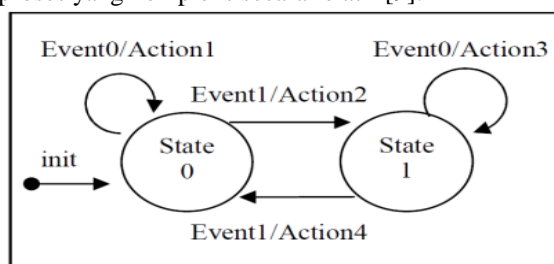
stres dan kebosanan. Hampir semua kalangan termasuk anak-anak, remaja, dan orang dewasa menikmati bermain permainan, mungkin dengan preferensi jenis permainan yang berbeda-beda [7]

### 2.3. Kecerdasan Buatan

Kecerdasan buatan, atau yang dikenal sebagai *Artificial Intelligence* (AI), merupakan bagian dari disiplin ilmu komputer yang memungkinkan mesin (komputer) untuk melaksanakan tugas yang serupa dengan yang dilakukan oleh manusia. Pada awalnya, komputer digunakan terutama untuk perhitungan, tetapi seiring berjalannya waktu, peran komputer semakin mendominasi dalam kehidupan masyarakat. Komputer tidak lagi terbatas pada peran sebagai alat perhitungan semata, melainkan juga diharapkan dapat menjalankan berbagai tugas yang pada dasarnya dilakukan oleh manusia [8]

### 2.4. Finite State Machine

*Finite State Machine* adalah sebuah metode dalam perancangan sistem kontrol yang menggambarkan perilaku atau prinsip operasionalnya dengan menggunakan tiga elemen utama, yakni *State* (keadaan), *Event* (peristiwa), dan *Action* (aksi). Dalam jangka waktu yang cukup besar, sistem akan berada pada salah satu *state* yang aktif. Sistem memiliki kapasitas untuk beralih ke keadaan lain ketika menerima *input* atau peristiwa tertentu, baik itu berasal dari luar atau komponen internal sistem itu sendiri. Perubahan keadaan ini umumnya disertai dengan pelaksanaan tindakan yang merupakan respons terhadap input yang diterima, yang bisa berupa tindakan sederhana atau melibatkan serangkaian proses yang kompleks secara relatif [9].



Gambar 1. Diagram FSM

Pada Gambar 1, terdapat representasi dari *Finite State Machine* yang menggambarkan dua keadaan, dua input, dan empat aksi yang berbeda. Ketika sistem diinisialisasi, proses transisi akan mengarahkan sistem ke *State0*. Pada fase ini, sistem akan menjalankan *Action1* jika *Event0* terjadi, sementara jika *Event1* terdeteksi, *Action2* akan dijalankan. Setelah itu, seiring berjalannya waktu, sistem akan berpindah ke *State1*, dan proses ini juga bisa terjadi sebaliknya [10].

### 2.5. Unity

Unity adalah sebuah aplikasi yang berguna untuk menciptakan *game* yang dapat berjalan di berbagai platform, dengan fokus pada kemudahan penggunaan. Unity juga dikenal memiliki kualitas yang tinggi dan

dapat terintegrasi dengan perangkat lunak profesional lainnya. Antarmuka Unity dirancang dengan tampilan yang sederhana namun telah melibatkan ribuan jam kerja untuk memastikan kualitasnya yang unggul di dunia *editor game*. Aspek visual dalam Unity dibuat dengan tingkat kualitas grafis yang tinggi, kompatibel dengan OpenGL dan DirectX. Selain itu, Unity juga mendukung berbagai format file yang umumnya digunakan dalam seni visual. Kelebihan lainnya adalah kompatibilitasnya dengan sistem 64-bit, dapat digunakan pada platform Mac OS X dan Windows, serta mampu menghasilkan game untuk berbagai platform seperti Mac, Windows, Wii, iPhone, iPad, dan Android [11].

**2.6. Android**

*Android* merupakan sebuah sistem operasi yang dirancang oleh Google berdasarkan kernel Linux. Sistem ini bertujuan untuk mendukung perangkat elektronik yang memiliki layar sentuh, seperti *smartphone* atau *tablet*. Sebagian besar orang dewasa saat ini memiliki dan menggunakan ponsel pintar *Android* untuk beragam keperluan, termasuk komunikasi, hiburan, belanja, transportasi, dan bahkan sebagai alat bantu untuk menemukan lokasi berdasarkan koordinat awal dan tujuan. *Android* bersifat *open source*, yang berarti dapat digunakan, disesuaikan, diperbaiki, dan disebarluaskan oleh para pembuat dan pengembang perangkat lunak secara bebas. Karakteristik *open source* ini memungkinkan perusahaan teknologi untuk memanfaatkannya dalam perangkat mereka tanpa perlu membayar lisensi. [12]

**3. METODE PENELITIAN**

**3.1. Target User**

Analisis pengguna digunakan untuk menghasilkan spesifikasi pengguna yang memiliki kemampuan untuk bermain game "Barbossa the Savior". Beberapa di antaranya meliputi:

1. Target pengguna *game* ini minimal berusia 15 tahun, karena *game* tersebut berisi banyak adegan pertarungan.
2. Pengguna memiliki gadget *android* dan dapat mengoperasikannya.

**3.2. Kebutuhan Fungsional**

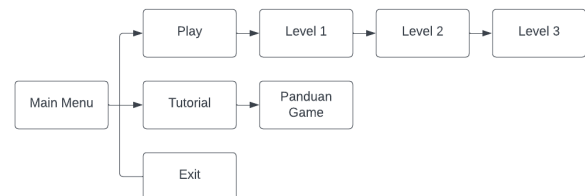
Kebutuhan fungsional berisi proses-proses apa saja yang nantinya yang dilakukan oleh sistem. Selain itu, kebutuhan fungsional juga memuat berbagai informasi yang perlu dihasilkan oleh sistem.

Berikut merupakan berbagai fitur yang dimiliki oleh *game* Barbossa the Savior, meliputi :

1. Fitur *Play* menampilkan total 3 level yang terdiri dari level 1, level 2 dan level 3.
2. Karakter digerakkan oleh pengguna menggunakan kontrol yang terdiri *analog* , *jump* dan *attack*.
3. Fitur koin digunakan untuk menampilkan skor permainan.

4. Fitur *pause* memungkinkan pemain untuk menghentikan sementara permainan yang sedang berlangsung.
5. Sistem dapat memberikan *sound effect* saat melawan musuh.
6. Fitur *health bar* untuk menggambarkan tingkat kesehatan pada karakter.
7. Fitur *life point* digunakan menggambarkan tingkat kehidupan atau kelangsungan hidup karakter atau pemain dalam permainan.
8. Pada tampilan *main menu* terdapat fitur mulai, panduan dan keluar
  - a) *Play*, berfungsi untuk menampilkan pilihan level permainan untuk memulai *game*.
  - b) Tutorial, berfungsi untuk menampilkan panduan *game*.
  - c) *Exit*, berfungsi untuk konfirmasi keluar *game*.

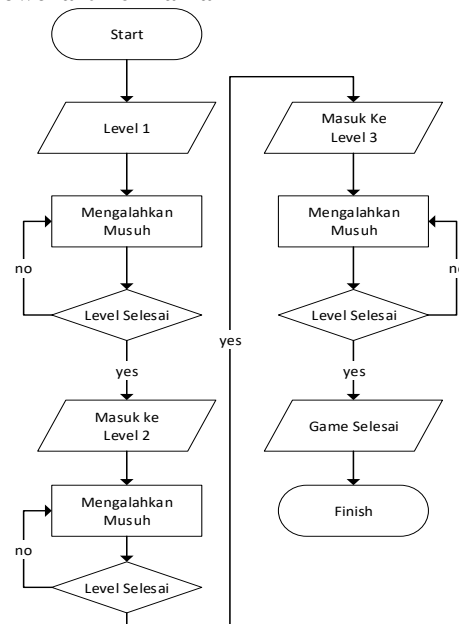
**3.3. Struktur Menu**



Gambar 2. Flowchart Struktur Menu

Pada Gambar 2 merupakan struktur menu. *Main menu* menjadi halaman *default* saat *game* pertama kali dibuka yang terdiri dari *play*, tutorial dan *exit*. Untuk memainkan *game* terdapat menu *play* yang terdiri dari 3 level yang harus dimainkan, menu tutorial digunakan untuk membantu pemain dalam menggunakan panduan dari *game* dan menu *exit* untuk menyudahi permainan.

**3.4. Flowchart Permainan**



Gambar 3. Flowchart Permainan

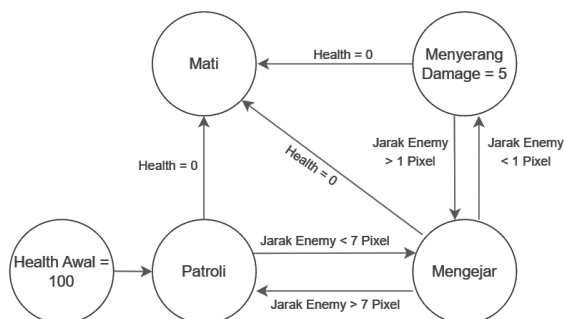
Pada gambar 3 merupakan *flowchart game*. *Game* akan dimulai dari tampilan main menu kemudian player akan masuk ke tampilan *game* pada level 1, Kemudian *player* akan berperang melawan musuh dengan darah 100% dan bos dengan darah 150% yang tersedia pada level tersebut. Setelah berhasil mengalahkan musuh maka *player* akan lanjut ke tahap berikutnya. Pada tampilan *game* level 2, *player* juga harus berperang melawan musuh dengan darah 100% dan bos dengan darah 200% yang tersedia pada level tersebut. Setelah berhasil mengalahkan musuh maka *player* akan lanjut ke tahap berikutnya. Pada level 3, *player* juga harus berperang melawan musuh dengan darah 100% dan bos dengan darah 250% yang dimana *player* harus mengalahkannya untuk memenangkan pertandingan dan juga mengakhiri permainan. Setelah berhasil mengalahkan musuh pada level 3 maka *player* akan kembali ke tampilan main menu.

### 3.5. Diagram FSM

Struktur game 2D "Barbossa The" memerlukan pertimbangan yang baik karena merupakan bagian dari cerita dan sistem permainan. Ada juga diagram FSM yang digunakan untuk menggambarkan bentuk keadaan dan kondisi transisi setiap NPC (*Non Player Character*) atau musuh dari game Barbossa The savior, mulai dari tipe mini garon, boss garon, ace, mini megalith, boss megalith, helium, mini oracle, boss oracle dan lithium.

Tiap karakter akan mempunyai *state* Patroli dan Die. *State* Patroli digunakan untuk membuat karakter NPC berpatroli pada tempat yang telah ditentukan dan *state* Die merupakan *state* ketika *Health Point* karakter NPC mencapai 0. Pada semua karakter NPC terdapat *state* yang sama yaitu *state* Mengejar dimana karakter NPC akan mulai mengejar pemain jika pemain mulai mendekati pemain. *State* Serang dimiliki oleh seluruh NPC untuk menyerang pemain jika pemain mencapai jarak serang NPC. Berikut adalah diagram FSM dari masing-masing tipe musuh di tiap levelnya:

#### 1. Diagram FSM Mini Garon

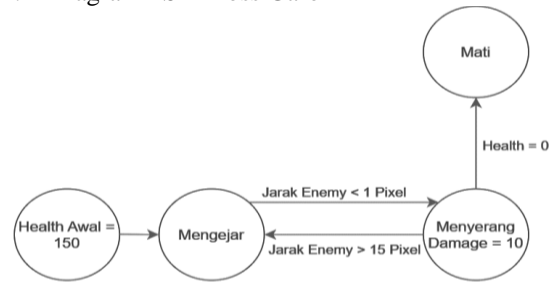


Gambar 4. Diagram FSM Karakter Mini Garon

Pada Gambar 4 merupakan diagram *Finite State Machine* dari tingkah laku atau pergerakan *Non Player Character Enemy* biasa yang berada di level 1 dengan *health* awal *enemy* yaitu 100. *Non Player Character Enemy* secara *default* berada pada *state* patroli. jika

pemain mendekati dan hingga jarak dari musuh menjadi  $< 7 \text{ pixel}$  maka *Enemy* akan mengejar, apabila pemain menjauh dari *enemy* hingga berjarak  $> 7 \text{ pixel}$  *enemy* akan kembali patroli. Jika *enemy* sudah berada di jangkauan serang dengan *player* maka *enemy* dapat memberikan *damage* sebesar 5 yang dapat mengurangi *health bar* pemain. Apabila *Non Player Character* berhasil menyerang *player* hingga  $\text{Health} \leq 0$  maka *player* akan mati dan sebaliknya jika *player* dapat menyerang *enemy* dengan  $\text{Health} \leq 0$  maka *Enemy* akan mati.

#### 2. Diagram FSM Boss Garon



Gambar 5. Diagram FSM Karakter Boss Garon






Pada Gambar 5 merupakan diagram *Finite State Machine* dari tingkah laku atau pergerakan *Non Player Character Enemy boss* yang berada di level 1 dengan *health* awal *enemy* yaitu 150. *Non Player Character Enemy Boss* Garon memiliki sifat hanya mengejar dan tidak berpatroli, Jika *enemy* sudah berada di jangkauan serang dengan *player* maka *enemy* dapat memberikan *damage* sebesar 10 yang dapat mengurangi *health bar* pemain. Apabila *Non Player Character* berhasil menyerang *player* hingga  $\text{Health} \leq 0$  maka *player* akan mati dan sebaliknya jika *player* dapat menyerang *enemy* dengan  $\text{Health} \leq 0$  maka *Enemy* akan mati.

### 3.6. Desain Karakter

Pada tabel 1 merupakan desain karakter dan musuh yang terdapat pada *Game* Barbossa the savior.

Tabel 1. Desain Karakter

No	Gambar	Keterangan
1		Barbossa Karakter Utama dengan <i>Health Bar</i> 100, <i>Life Point</i> 3 dan <i>Damage</i> 30
2		Mini Garon Musuh level 1 dengan <i>Health Bar</i> 100 dan <i>Damage</i> 5.
3		Boss Garon Musuh level 1 dengan <i>Health Bar</i> 150 dan <i>Damage</i> 10.
4		Ace Musuh level 1 dengan <i>Health Bar</i> 100 dan <i>Damage</i> 8.
5		Mini Oracle Musuh level 2 dengan <i>Health Bar</i> 100 dan <i>Damage</i> 5

No	Gambar	Keterangan
6		Boss Oracle Musuh level 2 dengan <i>Health Bar</i> 200 dan <i>Damage</i> 15.
7		Helium Musuh level 2 dengan <i>Health Bar</i> 100 dan <i>Damage</i> 10.
8		Mini Megaluth Musuh level 3 dengan <i>Health Bar</i> 200 dan <i>Damage</i> 9.
9		Boss Megaluth Musuh level 3 dengan <i>Health Bar</i> 100 dan <i>Damage</i> 20
10		Lithium Musuh level 3 dengan <i>Health Bar</i> 100 dan <i>Damage</i> 12

Pada Tabel 1 merupakan tampilan dari karakter utama dan musuh pada game. Karakter utama diperankan oleh barbossa dengan *health bar* 100%, *life point* 3 dan *damage* sebesar 30. Pada level 1 terdapat 3 musuh yaitu mini garon dengan *damage* sebesar 5, ace dengan *damage* sebesar 8, dan boss garon dengan *damage* 10 mempunyai *health bar* sebesar 150%. Kemudian pada level 2 terdapat 3 musuh yaitu mini megaluth dengan *damage* sebesar 7, helium dengan *damage* sebesar 10, dan boss oracle dengan *damage* 15 mempunyai *health bar* sebesar 200%. Setelah itu pada level terakhir terdapat 3 musuh yaitu mini oracle dengan *damage* sebesar 9, lithium dengan *damage* sebesar 12, dan boss megaluth dengan *damage* sebesar 20 mempunyai *health bar* sebesar 250%.

#### 4. HASIL DAN PEMBAHASAN

##### 4.1. Pembuatan Main Menu

###### 1. Main Menu

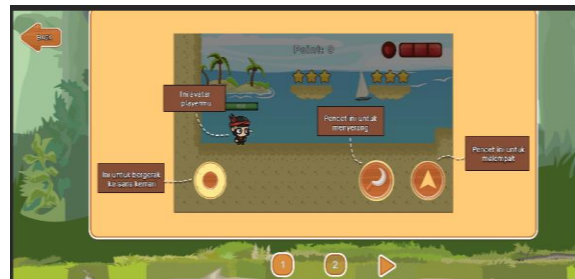


Gambar 6. Tampilan Main Menu

Pada Gambar 6 menampilkan antarmuka utama permainan. Latar belakangnya menggambarkan pantai dengan perpaduan warna biru dan coklat. Di tengahnya, terdapat judul permainan, sementara tombol-tombol lainnya juga terlihat. Tombol "Play" berfungsi untuk memulai permainan "Barbossa the Savior". Terdapat pula opsi "Tutorial" yang dapat diakses untuk panduan bermain, tombol "Unlock Level" yang memungkinkan akses ke semua level,

tombol "Suara" untuk mengatur suara permainan, tombol "Music" untuk mengontrol musik latar, dan tombol "Exit" untuk keluar dari permainan.

###### 2. Menu Tutorial



Gambar 7. Tampilan Menu Tutorial

Pada Gambar 7 merupakan tampilan dari tutorial halaman pertama. Digunakan untuk memberikan petunjuk tentang cara memainkan game tersebut. Halaman tutorial biasanya berisi informasi tentang kontrol permainan, mekanisme gameplay untuk membantu pemain memahami dasar-dasar game.

##### 4.2. Pembuatan Game

###### 1. Prolog



Gambar 8. Prolog Game

Pada Gambar 8 merupakan prolog dalam game yang dirancang untuk memperkenalkan pemain pada cerita, dunia, atau konteks permainan sebelum mereka memulai petualangan utama. Prolog biasanya berfungsi sebagai pengantar untuk membangun suasana, memperkenalkan karakter utama, dan menetapkan latar belakang cerita

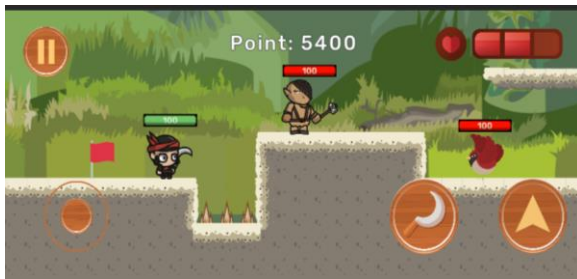
###### 2. Level 1



Gambar 9. Tampilan Map Level 1

Pada Gambar 9 merupakan map level 1 lebih kecil daripada level 2 dan konsep di pantai. Map ini disusun dengan berlatar belakang pantai, dan terdiri dari musuh yang disesuaikan pada tiap level tersebut. beberapa trap bulu babi sebagai rintangan yang dapat mengurangi *health point player*.

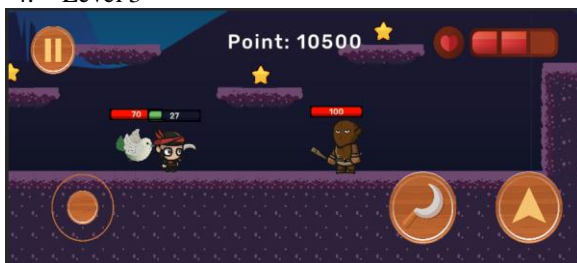
3. Level 2



Gambar 10. Tampilan Map Level 2

Pada Gambar 10 merupakan map level 2 lebih luas daripada level 1 dan konsep di hutan. Map ini disusun dengan berlatar belakang hutan, dan terdiri dari musuh yang disesuaikan pada tiap level tersebut. beberapa *trap* kayu sebagai rintangan yang dapat mengurangi health point player.

4. Level 3



Gambar 11. Tampilan Map Level 3

Pada Gambar 11 merupakan Map level 3 lebih luas daripada level 1 dan 2 dan konsep di pegunungan. Map disusun dengan berlatar belakang gunung, dan terdiri dari musuh yang disesuaikan pada tiap level tersebut. beberapa *trap* duri sebagai rintangan yang dapat mengurangi health point

5. Epilog



Gambar 12. Tampilan Epilog

Pada Gambar 12 merupakan epilog dalam *game* dan mengacu pada bagian akhir permainan yang menyajikan penyelesaian cerita, penutupan naratif, dan pemenuhan tujuan utama permainan. Epilog memberikan pemain pemahaman yang lebih lengkap tentang akhir cerita dan nasib karakter utama.

4.3. Pengujian Metode FSM

Pengujian metode *Finite State Machine* (FSM) dalam konteks *game* Barbossa the Savior memastikan bahwa implementasi FSM di dalam *game* berfungsi sebagaimana mestinya dan memberikan perilaku yang diharapkan. Hasil dari pengujian metode FSM bisa dilihat pada Tabel 2.

Tabel 2. Pengujian Metode FSM

No.	Tujuan	Hasil Sistem	Hasil
1	Enemy mendekati Player, jika jarak Player dengan enemy < 7 pixel dan < 4 pixel	Enemy bergerak mendekat ke arah Player	Diterima
2	Enemy kembali patroli jika jarak Player dengan enemy > 7 pixel dan > 4 pixel	Enemy bergerak patroli	Diterima
3	Semua Enemy menyerang Player ketika jarak Player dengan enemy < 1 pixel	Enemy menyerang dan memberikan damage	Diterima
4	Health Enemy = 0, maka Enemy mati	Enemy akan menghilang dari game	Diterima
5	Semua Enemy Boss mengejar player jika jarak Player dengan Enemy Boss < 15 pixel	Enemy Boss mengejar Player	Diterima
7	Semua Enemy Boss akan menyerang jika jarak Player dengan Enemy Boss < 1 pixel	Enemy Boss menyerang player	Diterima
8	Semua Enemy Boss akan mati ketika Health = 0	Enemy Boss Mati	Diterima
9	Health Bar Player akan berkurang ketika terkena Damage Enemy	Player mati dan respawn ke tempat checkpoint	Diterima

Dari pengujian Tabel 4.1 menunjukkan bahwa metode *Finite State Machine* yang digunakan pada *game* memberikan hasil sesuai yang diharapkan.

4.4. Pengujian Main Menu

Pengujian *main menu* dalam *game* Barbossa the Savior sangat penting karena menu ini merupakan titik awal interaksi pengguna dengan *game*. *Main menu* haruslah intuitif, mudah digunakan, dan memberikan informasi yang cukup kepada pemain sebelum mereka memulai permainan

Tabel 3 Pengujian Main Menu

No	Tujuan	Hasil Yang Diharapkan	Hasil System	Hasil Uji
1	Mengklik menu play	Sistem akan memasuki halaman pemilihan stage level	Tampilan Stage Level	Diterima
2	Mengklik menu stage level	Sistem akan memulai permainan	Tampilan Gameplay	Diterima

No	Tujuan	Hasil Yang Diharapkan	Hasil System	Hasil Uji
3	Mengklik menu tutorial	Sistem akan memasuki halaman tutorial yang terdiri dua halaman.	Tampilan menu tutorial	Diterima
4	Mengklik menu <i>exit</i>	Permainan akan diakhiri oleh sistem.	Keluar dari permainan	Diterima
5	Mengklik menu <i>sound</i>	Sistem akan mematikan suara <i>game</i>	Suara <i>game</i> mati	Diterima

Dari hasil pengujian yang tertera pada Tabel 3 di atas, dapat dilihat bahwa tombol-tombol pada menu utama, termasuk menu play, pemilihan stage, sound, tutorial, dan exit, memberikan hasil yang sesuai dengan yang diharapkan.

#### 4.5. Pengujian Control Player

Pengujian kontrol pemain melibatkan pemeriksaan setiap fungsi dari tombol dan pengendali analog yang telah diimplementasikan untuk menggerakkan karakter barbossa dalam permainan.

Tabel 4. Pengujian Control Player

No	Tombol	Fungsi	Hasil
1	Analog Kanan	Player berjalan ke arah kanan	Diterima
2	Analog Kiri	Player berjalan ke arah kiri	Diterima
3	Attack	Player menyerang	Diterima
4	Jump	Player Melompat	Diterima

Dari pengujian Tabel 4 menunjukkan bahwa tombol *Analog*, *Attack*, *Jump* dan *Pause* pada *control player* barbossa the savior memberikan hasil sesuai yang diharapkan.

#### 4.6. Pengujian Device

Pengujian device ini merupakan proses mengidentifikasi, menguji, dan memastikan bahwa *game* Barbossa the Savior berfungsi dengan baik, bebas dari bug, dan memberikan pengalaman bermain yang baik bagi pengguna.

Tabel 5. Pengujian Device

No	Nama Device	RAM	Versi Android	Hasil
1	Knox Emulator	3 GB	5	Gagal Dijalankan
2	Oppo A57	3 GB	6	Berhasil Dijalankan
3	Samsung S8+	4 GB	7	Berhasil Dijalankan
4	Vivo 1820	2 GB	8	Berhasil Dijalankan
5	Samsung S9+	4 GB	9	Berhasil Dijalankan
6	Realme 5	4 GB	10	Berhasil Dijalankan
8	Poco M3	4 GB	11	Berhasil Dijalankan
9	Redmi Note 8Pro	6 GB	12	Berhasil Dijalankan

No	Nama Device	RAM	Versi Android	Hasil
10	Samsung S23 Ultra	12 GB	13	Berhasil Dijalankan

Dari pengujian Tabel 5 menunjukkan bahwa *game* barbossa the savior dapat berhasil dijalankan menggunakan perangkat dengan RAM 2 GB keatas dan versi *Android* 6 hingga versi *Android* 13.

#### 4.7. Pengujian User

Pengujian user *game* akan dilakukan dengan menggunakan penyebaran kuisioner. Pengujian dilakukan untuk memberikan informasi bahwa *game* berfungsi dengan baik . Hasil dari pengujian user bisa dilihat pada Tabel 6.

Tabel 6 Pengujian User

No	Pernyataan	Jawaban				
		SS	S	C	TS	STS
1	Desain karakter dalam permainan , apakah sudah menarik	9	17	4	0	0
2	Desain dari <i>prolog</i> dan <i>epilog game</i> terlihat menarik	6	7	16	1	0
3	Anda menemukan <i>bug</i> atau kesalahan dalam permainan	1	6	8	14	1
4	Kesulitan level <i>game</i> dalam bermain sesuai dengan tingkatan level yang ada	3	14	12	1	0
5	Penggunaan kontrol pada permainan mudah dioperasikan dan responsif	4	18	7	1	0
6	Ada bagian dalam permainan yang tidak masuk akal	1	3	15	10	1
7	Anda menemukan fitur yang kurang dalam permainan	1	9	12	7	1
8	Anda merasa bosan ketika bermain	1	4	6	13	6
9	Kualitas dan efek suara pada permainan sesuai dengan suasana	8	15	5	2	0
Total		34	93	85	49	9

- SS = Sangat Setuju
- S = Setuju
- C = Cukup
- TS = Tidak Setuju
- STS = Sangat Tidak Setuju

Tabel 7. Persentase Responden Pada Pengujian User

No	Jawaban	Persentase
1	Sangat Setuju	$(34 / 270 * 100\%) = 12,59 \%$
2	Setuju	$(93 / 270 * 100\%) = 34,44 \%$
3	Cukup	$(85 / 270 * 100\%) = 31,48 \%$
4	Tidak Setuju	$(49 / 270 * 100\%) = 18,14 \%$
5	Sangat Tidak Setuju	$(9 / 270 * 100\%) = 3,33 \%$

Dari pengujian yang dilakukan oleh 30 responden pada table 7 ,didapatkan hasil persentase yakni 12,59% menyatakan Sangat Setuju, 34,44% menyatakan Setuju, 31,48% menyatakan Cukup, 18,44% menyatakan Tidak Setuju dan 3,33% menyatakan Sangat Tidak Setuju. Berdasarkan hasil pengujian *user*, dapat disimpulkan bahwa *game* yang dirancang sudah baik

### 5. KESIMPULAN DAN SARAN

Berdasarkan hasil perancangan dan implementasi dalam permainan "Barbossa the Savior", dapat ditarik kesimpulan penting. Pertama, hasil pengujian implementasi pada berbagai perangkat *Android* menunjukkan tingkat keberhasilan sesuai harapan. Kedua, pengujian *blackbox* yang melibatkan menu, level, dan sistem *gameplay* juga menunjukkan tingkat keberhasilan yang sesuai dengan yang diinginkan. Ketiga, hasil pengujian implementasi kontrol pemain juga sesuai dengan harapan yang telah ditetapkan. Kesimpulan ini mengindikasikan bahwa perancangan dan implementasi *game* telah berhasil mencapai tujuan pengujian dengan baik. Sebagai arahan untuk penelitian dan pengembangan selanjutnya, disarankan agar diperkenalkan fitur penggunaan koin dalam permainan untuk membeli item-item tertentu, sehingga dapat meningkatkan daya tarik *game*. Selain itu, disarankan juga untuk melakukan pengujian menggunakan metode *white box* guna memastikan kualitas *script program* dalam *game* tetap optimal.

### DAFTAR PUSTAKA

[1] 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, doi: 10.35957/jatisi.v8i2.580.

[2] R. D. Kartika, "PEMBUATAN GAME ADVENTURE KUMACHI NO SHIMA DENGAN MENGGUNAKAN METODE FSM (Finite State Machine)," *J. Mhs. Tek. Inform.*, vol. 1, no. 1, pp. 492–497, 2017.

[3] F. A. Asyraq, D. S. Hormansyah, and M. Astiningrum, "Implementasi Fsm (Finite State Machine) Pada Game Surabaya Membara," *J.*

*Inform. Polinema*, vol. 6, no. 2, pp. 11–17, 2020, doi: 10.33795/jip.v6i2.278.

[4] A. Muhamad Rumakey, J. Dedi Irawan, and A. Wahid, "Pembuatan Game 2D 'Escape Plan' Dengan Metode Finite State Machine," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 4, no. 2, pp. 65–72, 2020, doi: 10.36040/jati.v4i2.2712.

[5] I. Satrio, F. Santi Wahyuni, and D. Rudhistiar, "Penerapan a\* Pathfinding Dan Fsm (Finite State Machine) Pada Game 'Lost Civilization' Berbasis Android," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 6, no. 2, pp. 1192–1199, 2023, doi: 10.36040/jati.v6i2.5402.

[6] J. Petrus, "Rancang Bangun Aplikasi Game Edukasi Puzzle Pengenalan Tokoh Sejarah Berbasis Android Dengan Metode Linear Congruential Generator ( LCG)," pp. 1–9, 2014.

[7] M. Ridoi, "Cara Mudah Membuat Game Edukasi dengan Construct 2: Tutorial sederhana Construct 2," p. 124, 2018.

[8] H. Jaya *et al.*, *Kecerdasan Buatan*, vol. 53, no. 9, 2018.

[9] J. Rahmadi, "Pembuatan Game Dangerous Escape dengan Menggunakan Metode FSM (Finite State Machine)," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 2, no. 1, pp. 67–74, 2018.

[10] 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, doi: 10.46984/sebatik.v23i1.453.

[11] D. Horachek, *Creating e-Learning games with Unity*. Packt Publishing Ltd, 2014.

[12] D. Yusuf and F. N. Afandi, "Aplikasi Absensi Berbasis Android Menggunakan Validasi Koordinat Lokasi Dan Nomor Handpone Guna Menghindari Penularan Virus Covid 19," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 10, no. 1, pp. 16–22, 2020, doi: 10.36448/jmsit.v10i1.1492.