

Harap mengisi tabel ini, Tabel ini digunakan untuk keperluan komunikasi administrasi saja, saat publish akan dihapus oleh team editor.	
Nama author ke 1	Muhammad Farrel Arrazzaq
Nomor WA	081239386986
Prodi/Jurusan	Teknik Informatika S-1
Perguruan Tinggi	Institut Teknologi Nasional Malang

PERANCANGAN GAME 2D PLATFORMER “ADVENTURE QUEST” DENGAN METODE FINITE STATE MACHINE BERBASIS ANDROID

Muhammad Farrel Arrazzaq, Agung Panji Sasmito, Hani Zulfia Zahro'

Program Studi Teknik Informatika S1, Fakultas Teknologi Industri
Institut Teknologi Nasional Malang, Jalan Raya Karanglo km 2 Malang, Indonesia
Email: _1918102@scholar.itn.ac.id

ABSTRAK

Game adalah permainan yang dimainkan untuk menghilangkan stres dengan aturan tertentu untuk menentukan siapa yang menang dan kalah. Dalam beberapa tahun terakhir, *smartphone* dan *tablet* telah menjadi perangkat utama untuk bermain game karena pengguna perangkat *mobile* memainkan *game* pada waktu dan lokasi yang mereka inginkan, bahkan saat sedang dalam perjalanan. Salah satu *genre game* yang populer di *platform mobile* yaitu *Game 2D Platformer*. *Game* dirancang menggunakan kecerdasan buatan *Finite State Machine* sehingga *enemy* dapat patroli dalam jarak tertentu, mengejar dan menyerang *player* ketika berada dalam jangkauan, dan kembali patroli jika berada diluar jangkauan. Berdasarkan pengujian fungsional diketahui semua fitur sudah berjalan dengan baik. Pengujian *Device* menghasilkan bahwa *game* dapat berjalan dengan minimal versi *Android 6*. Pada pengujian *user* didapatkan hasil kuesioner presentase bahwa 18,63% menyatakan Sangat Setuju, 37,27% menyatakan Setuju, 37,72% menyatakan Cukup, 10,90% menyatakan Tidak Setuju, dan 0,45% menyatakan Sangat Tidak Setuju, sehingga dapat disimpulkan bahwa *game* yang dirancang sudah baik.

Kata kunci : *Android, 2D Platformer, Finite State Machine, Game*

1. PENDAHULUAN

Perkembangan teknologi saat ini membuat hidup lebih mudah bagi semua orang. Kemajuan dalam bidang informasi dan komunikasi membuat informasi semakin mudah untuk didapatkan. Industri hiburan, seperti bermain *game online* atau *offline* juga memanfaatkan kemajuan ini [1].

Game adalah permainan yang dimainkan untuk menghilangkan stres dengan aturan tertentu untuk menentukan siapa yang menang dan kalah [2]. *Game* tidak hanya menyenangkan, tetapi juga dapat memiliki manfaat edukatif dan kognitif. *Game* memiliki berbagai macam jenis dan kategori, dari *game aksi* hingga *game strategi*.

Dalam beberapa tahun terakhir, *smartphone* dan *tablet* telah menjadi perangkat utama untuk bermain *game* bagi jutaan orang di seluruh dunia. Karena aksesibilitas yang tinggi, Pengguna perangkat *mobile* dapat memainkan *game* pada waktu dan lokasi yang mereka inginkan, bahkan saat sedang dalam perjalanan. Selain itu, penggunaan teknologi *touchscreen* dan sensor gerak dalam perangkat *mobile* telah membuka jalan untuk jenis *gameplay* baru dan lebih interaktif.

Game 2D platformer adalah salah satu *genre game* yang populer di *platform Android*. *Game* ini biasanya melibatkan karakter utama yang berlari, melompat, dan bergerak melalui serangkaian *platform*

dan rintangan. Meskipun *game 2D platformer* mungkin tampak sederhana, banyak dari mereka memiliki mekanik *gameplay* yang menantang dan visual yang menarik, serta cerita yang menarik dan atmosfer yang memikat. Contoh *game 2D platformer* yang dikenal sebagai *game* klasik dan menjadi ikon budaya, yaitu Super Mario Bros dan Sonic the Hedgehog.

Metode yang bisa diterapkan untuk mengembangkan *genre game* ini adalah melalui penggunaan *metode Finite State Machine*. Pendekatan Metode *Finite State Machine* memanfaatkan beragam kondisi, peristiwa, dan tindakan yang saling terhubung [3]. Pada *game Adventure Quest* Metode FSM digunakan untuk mengatur cara karakter berperilaku dalam *game* sehingga memungkinkan karakter untuk mengubah perilaku sesuai dengan kondisi *game* serta dapat berinteraksi dengan lingkup sekitarnya dengan jenis *game mobile* karena akan dikembangkan untuk *platform Android* menggunakan *Unity* sebagai *Game Engine*.

Berdasarkan permasalahan yang ada, maka penulis melakukan penelitian dengan judul “Perancangan Game 2D Platformer “Adventure Quest” Dengan Metode Finite State Machine Berbasis Android”. *Game* tersebut akan menceritakan tentang Seorang Petualang yang akan menjalankan sebuah misi untuk menjelajah di daerah pegunungan dengan

membasmi monster - monster yang ada didaerah tersebut.

2. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam Penelitiannya yang berjudul “Rancang Bangun Game 2d Shooter Platformer Menggunakan Metode Finite State Machine” Karakter NPC *Enemy* dan *Boss* pada *game* berhasil menggunakan Metode *Finite State Machine*. Berdasarkan pengujian, fungsi dari setiap *state* berjalan dengan baik, dengan tingkat persetujuan dari responden sekitar 60% [4].

Dalam Penelitiannya yang berjudul “Penerapan Metode Finite State Machine Game 2d Adventure Kebokicak Dan Surontanu Berbasis Android” yang bertujuan untuk menciptakan *game Android* berbasis 2D yang menarik bagi generasi muda, dengan memfokuskan pada edukasi cerita rakyat Indonesia. Melalui implementasi dan pengujian, *game* ini berhasil mendapatkan tingkat persetujuan sebesar 82,24% dari para responden [1].

Dalam Penelitiannya yang berjudul “Penerapan Finite State Machine Pada Game “Pendekar Cisadane” Berbasis Android” yang bertujuan untuk memperkenalkan karakter Pendekar Cisadane kepada generasi muda sambil memberikan hiburan yang menyenangkan. Berdasarkan pengujian fungsional, aplikasi *game* ini berhasil berjalan sesuai harapan. Dari 20 responden yang diuji, sebanyak 61,5% berhasil mencapai penilaian yang baik, menunjukkan bahwa desain *game* Pendekar Cisadane sesuai dengan harapan yang diinginkan [5].

2.2 Game

Game adalah salah satu bidang ilmu komputer yang sangat berkembang. Umumnya, *Game* dilakukan dengan aturan tertentu yang menentukan pemenang dan kalah, dan bertujuan untuk menyegarkan pikiran. *Game* juga berfungsi sebagai jenis aplikasi edukatif, yang berarti dapat digunakan sebagai sarana pembelajaran di mana pendekatannya dapat menggabungkan konsep belajar sambil bermain, terutama untuk pendidikan anak-anak [6].

2.3 Kecerdasan Buatan

Kecerdasan buatan merupakan bagian dari ilmu komputer yang fokus pada konsepsi, pembuatan model, dan penyimpanan kemampuan berpikir manusia dalam bentuk teknologi informasi, yang selanjutnya dapat digunakan untuk proses pengambilan keputusan [7].

Game adalah salah satu bidang yang menggunakan kecerdasan buatan, Kecerdasan buatan pada *game* digunakan untuk NPC sehingga memiliki perilaku yang lebih kompleks dan realistis, menyesuaikan tingkat kesulitan *game*, mengambil keputusan dinamis sehingga menciptakan keputusan berdasarkan tindakan pemain, menciptakan konten dinamis agar memberikan variasi yang tak terbatas pada pemain, mengoptimasi grafis sehingga lebih

efisien, berperan sebagai lawan AI, dan meningkatkan keseluruhan pengalaman bermain. Kecerdasan buatan pada *game* memiliki beberapa jenis yaitu :

1. Pathfinding

Pathfinding adalah Metode yang digunakan untuk menemukan jalan terbaik dari satu titik kepada tujuan lain. Pemanfaatan metode *pathfinding* dalam *game* bertujuan untuk memilih jalur terbaik dan optimal bagi karakter atau objek saat bergerak dari tempat awal menuju tempat lain di lingkungan *game* [8].

2. Rule-Based Reasoning (CBS)

Rule-Based Systems (RBS) adalah aplikasi yang menggunakan teknologi Kecerdasan Buatan (AI) untuk mewakili pengetahuan dan mampu menjalankan tugas seperti pengambilan keputusan yang dilakukan oleh seorang pakar [9].

3. Case-Based System (RBS)

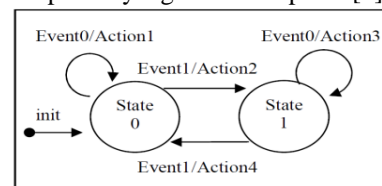
Case-Based Reasoning (CBR) merupakan pendekatan yang dimanfaatkan guna menyelesaikan berbagai permasalahan dengan merujuk pada solusi yang ditemukan dari permasalahan sebelumnya. Dalam pendekatan ini, masalah baru diselesaikan dengan mencari kasus yang mirip dari pengalaman masa lalu dan menerapkan solusi yang relevan pada situasi baru tersebut [10].

4. Decision Tree

Algoritma *decision tree* merupakan salah satu metode klasifikasi yang membagi masalah dengan metode *divide-and-conquer*, memilah atribut dari atas ke bawah ke dalam kelas yang sesuai, dan menghasilkan pohon keputusan berupa aturan klasifikasi [11].

2.4 Finite State Machine

Metode *Finite State Machine* digunakan untuk merancang perilaku *Artificial Intelligence* di dalam *game* dengan keunggulan dalam komputasi yang sederhana dan kemudahan pemahaman serta implementasinya. Metode ini menggunakan diagram *state* berdasarkan jenis perilaku yang telah ditentukan, dan sistem akan berada dalam satu *state* yang aktif selama jangka waktu yang cukup lama. Jika sistem menerima masukan atau peristiwa tertentu dari perangkat luar atau komponen internal, sistem dapat berpindah ke *state* lain. Transisi keadaan ini biasanya diikuti oleh tindakan yang dilakukan sistem sebagai respons terhadap masukan yang diterima. Mereka dapat berupa tindakan sederhana atau termasuk serangkaian proses yang lebih kompleks [3]



Gambar 1. Blok Diagram *Finite State Machine*

Diagram tersebut menjelaskan *Finite State Machine* dengan dengan dua *state*, dua input, dan empat tindakan yang berbeda. Seperti yang terlihat dalam ilustrasi, ketika sistem dinyalakan, sistem akan beralih ke State0. Pada kondisi ini, jika ada input Event0, sistem akan memicu Action1. Di sisi lain, jika input Event1 terdeteksi, sistem akan melakukan Action2. Setelah itu, sistem akan beralih ke State1, dan sebaliknya.

3. METODE PENELITIAN

3.1 Analisis Target Pengguna

Analisis Target Pengguna digunakan untuk mengetahui spesifikasi pengguna agar dapat memainkan *game* “Adventure Quest”, diantaranya :

1. Pengguna diprioritaskan dalam rentang usia 18 hingga 30 tahun.
2. Pengguna memiliki *Android*.
3. Peminat *game platformer* dengan *genre adventure*.

3.2 Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan berbagai analisis yang berhubungan dengan *game* Adventure Quest, diantaranya:

1. Fitur *Play* di *Main Menu* dapat menampilkan total 3 Level yang terdiri dari Level 1, Level 2, dan Level 3.
2. Karakter dapat digerakkan oleh pemain, tapi musuh tidak dapat digerakkan.
3. Kontrol untuk menggerakkan karakter terdapat analog, jump, dan menyerang.
4. Fitur *Setting* di *Main Menu* dapat menampilkan dan mengubah volume suara dari *game*.
5. Dalam *ingame* terdapat Fitur *Pause* digunakan untuk menjeda permainan.
6. Terdapat Fitur untuk mengetahui *Lifepoint* dengan menampilkan Logo Hati.
7. Fitur *Coin* untuk menampilkan total *coin* yang telah didapatkan.
8. Fitur Level menampilkan level berapa yang sedang dimainkan.

3.3 Kebutuhan Non Fungsional

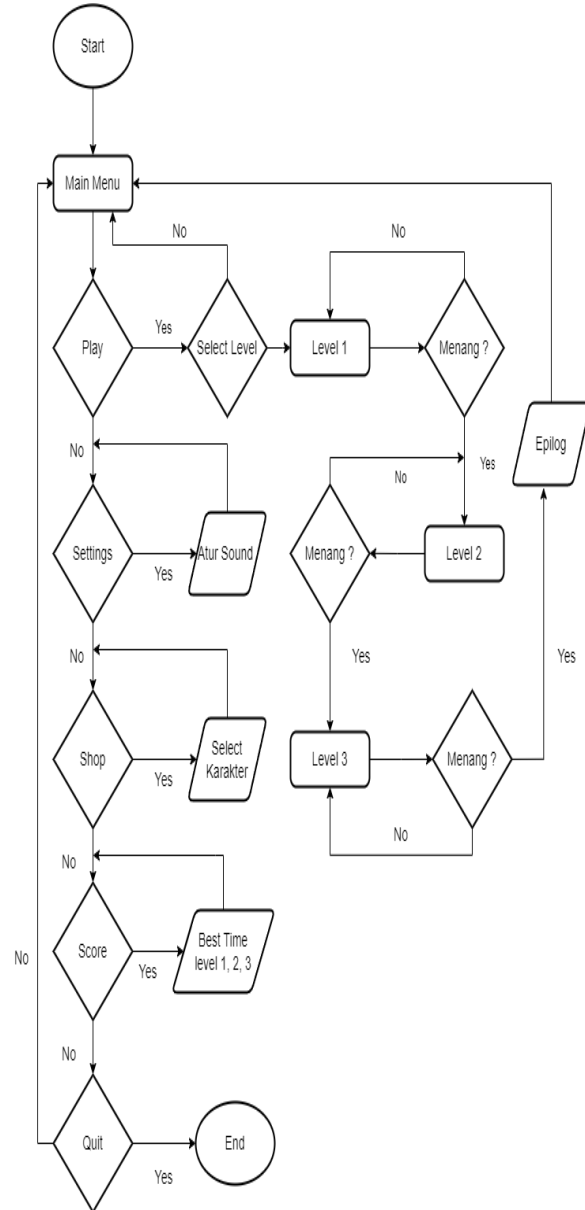
Spesifikasi yang diperlukan agar *game* Adventure Quest dapat dijalankan tergantung pada kebutuhan non fungsional. Faktor-faktor ini memiliki dampak yang signifikan pada seluruh proses perancangan *game*, dari awal hingga akhir tahapan perancangan.

1. *Game* dapat dimainkan melalui *platform Android*.
2. *Game* memiliki tampilan grafis yang menarik.
3. *Game* memiliki kontrol yang mudah untuk digunakan.
4. *Game* berukuran ringan sehingga dapat diunduh dengan cepat dan tidak menghabiskan terlalu

banyak ruang penyimpanan pada perangkat pemain.

5. *Game* memiliki kualitas suara dan musik yang baik sehingga dapat didengar dengan jelas.
6. *Game* dapat dijalankan di berbagai jenis perangkat yang berbeda.

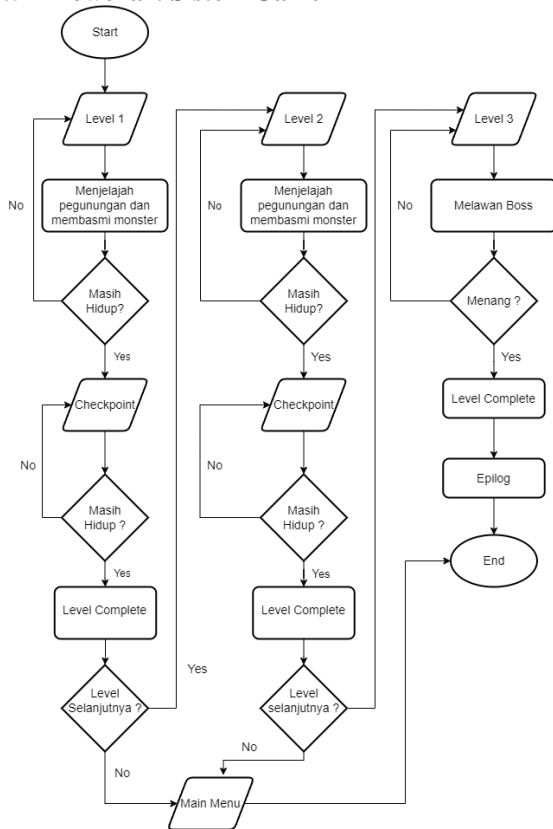
3.4 Flowchart Menu



Gambar 2. Flowchart Menu

Berdasarkan *flowchart* pada gambar 2 di atas, *main menu* akan menjadi *menu* utama *game* ketika kita telah memasuki *game* tersebut. Pemain bisa memilih *menu play* untuk memilih level dari *game* yang akan dimainkan, jika tidak pemain bisa kembali ke *main menu* dengan memilih *menu setting* untuk pengaturan suara pada *game*, *menu shop* untuk membeli atau mengganti karakter, *menu score* untuk menampilkan *best time* di setiap level, dan *quit* jika pemain memilih untuk keluar dari *Game*.

3.5 Flowchart Sistem Game



Gambar 3. Flowchart Sistem Game

Berdasarkan flowchart pada gambar 3 diatas, game dimulai pemain memasuki level 1 dengan tampilan karakter utama yang memasuki area di level 1. Kemudian diharuskan melanjutkan perjalanan hingga ke tempat akhir dengan membasmi musuh yang menghadang dengan selamat, terdapat checkpoint yang berada di tengah perjalanan dan collectible item coin di sepanjang perjalanan, jika pemain terbunuh sebelum checkpoint maka pemain akan respawn di tempat awal. Setelah mendapat checkpoint, jika selamat sampai tujuan maka level 1 berakhir dan dapat melanjutkannya ke level selanjutnya atau kembali ke Main Menu, jika terbunuh akan respawn kembali di tempat checkpoint tersebut.

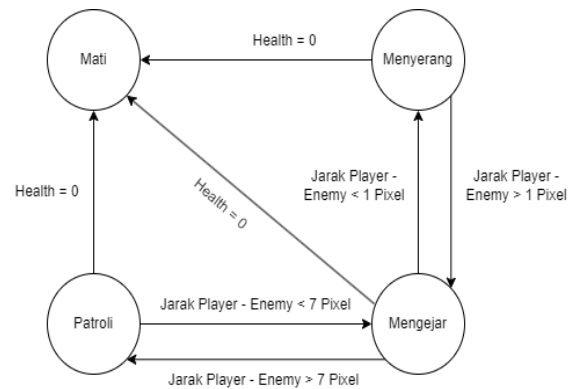
Pada level 2 memiliki konsep yang sama, dengan level 1, yaitu melanjutkan perjalanan hingga ke tempat akhir dengan melawan musuh yang ada dengan selamat, dengan checkpoint yang berada di tengah perjalanan dan collectible item coin di sepanjang perjalanan, namun memiliki perbedaan di desain level map dan tempat musuh berada.

Pada Level 3, Pemain diharuskan membunuh Kedua boss yang ada jika terbunuh makan pemain akan respawn ditempat awal, jika berhasil maka akan menampilkan epilog dari game.

3.6 Diagram Finite State Machine

Game Adventure Quest menggunakan metode *Finite State Machine*, dengan diagram yang digunakan sebagai berikut :

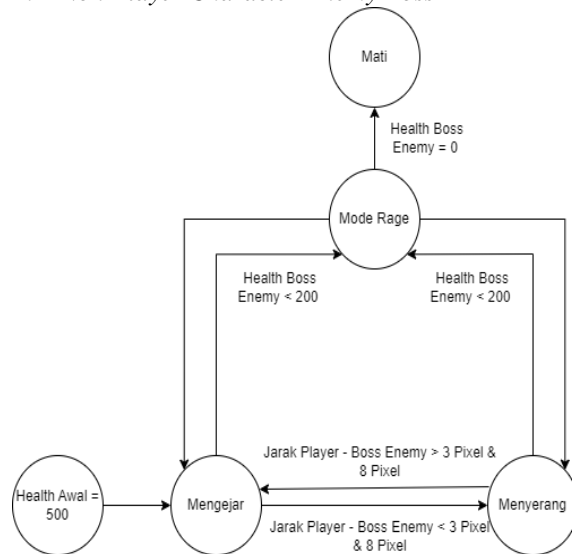
1. Non Player Character Enemy



Gambar 4. Diagram *Finite State Machine Enemy*

Diagram *Finite State Machine* pada gambar 4 diatas digunakan untuk semua karakter *Enemy* biasa. *Non Player Character Enemy* memiliki sifat *Patroli* sebagai sifat *default* nya, lalu jika pemain mendekati dan hingga jarak dari musuh menjadi < 7 pixel maka *Enemy* akan mengejar, apabila pemain menjauh dari *enemy* hingga berjarak > 7 pixel *enemy* akan kembali patroli. Jika mendekati lagi hingga berjarak < 1 pixel maka *Enemy* akan kembali mengejar, apabila pemain menjauh dari *Enemy* hingga berjarak > 1 pixel *Enemy* akan kembali mengejar dan ketika *Health Enemy* = 0 ketika dalam keadaan patroli, mengejar, maupun menyerang *Enemy* akan mati.

2. Non Player Character Enemy Boss

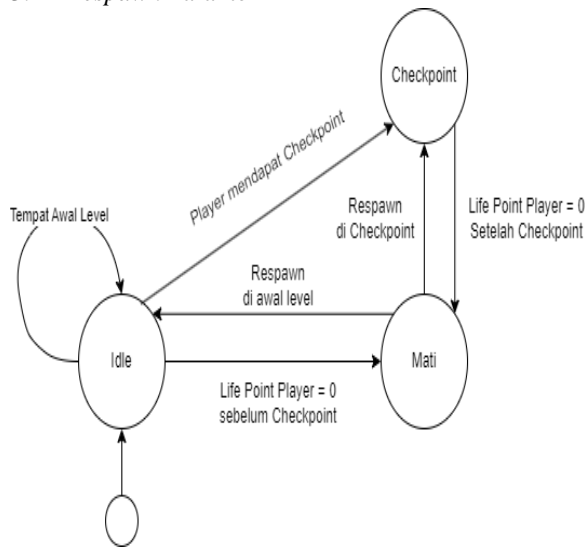


Gambar 5. Diagram *Finite State Machine Enemy Boss*

Diagram *Finite State Machine* pada Gambar 5 Diatas digunakan untuk karakter *Enemy Boss* 1 & 2. *Non Player Character Enemy Boss* 1 dan 2 memiliki *health* awal 500, dengan sifat *default* akan langsung mengejar, Untuk *Enemy Boss* 1 pada saat pemain dan *Enemy Boss* berjarak < 3 pixel *Enemy Boss* akan langsung menyerang, apabila pemain menjauh dari *Enemy Boss* hingga berjarak > 3 pixel *enemy boss* akan kembali Mengejar. Sedangkan untuk *Boss Enemy* 2 akan menyerang ketika pemain dan *enemy boss*

berjarak < 8 pixel, apabila pemain menjauh dari *Enemy Boss* hingga berjarak > 8 pixel *enemy boss* akan kembali Mengejar. Ketika *Health Enemy Boss* 1 dan 2 < 200 maka *Enemy Boss* akan memasuki mode *Rage* dan menjadi agresif, apabila *Health Enemy Boss* 1 dan 2 = 0 maka *Enemy Boss* 1 dan 2 akan mati.

3. *Respawn* karakter



Gambar 6. Diagram *Finite State Machine Respawn* Karakter

Diagram *Finite State Machine* pada Gambar 6 diatas digunakan untuk karakter utama, yaitu sebagai *Respawn* karakter. Ketika *Lifepoint* karakter = 0 karena serangan atau bersentuhan dengan *Enemy* dan *trap* sebelum mendapat *checkpoint* maka karakter akan *Respawn* kembali di tempat awal level. Ketika karakter mendapat *checkpoint* dan *Lifepoint* karakter nya = 0 maka karakter akan *Respawn* di tempat *checkpoint* yang telah didapatkan.

3.7 Desain Karakter

Pada Tabel 1 dibawah menampilkan desain karakter utama dan NPC musuh pada *Game Adventure Quest*.

Tabel 1. Desain Karakter

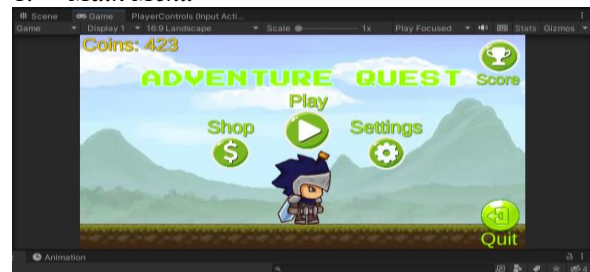
Gambar	Keterangan
	Hiro Karakter Utama
	Adventure Girl Karakter Utama 2
	Swordsman Hiro Karakter Utama 3

Gambar	Keterangan
	Soldier Karakter Utama 4
	Bandit Karakter Utama 5
	Yillo Karakter <i>Enemy</i> yang ada di setiap level
	Blu Karakter <i>Enemy</i> yang ada di setiap level
	Knight Karakter <i>Enemy Boss</i> Pertama di Level 3
	Burny Karakter <i>Enemy Boss</i> Kedua di Level 3

HASIL DAN PEMBAHASAN

4.1 Pembuatan Main Menu

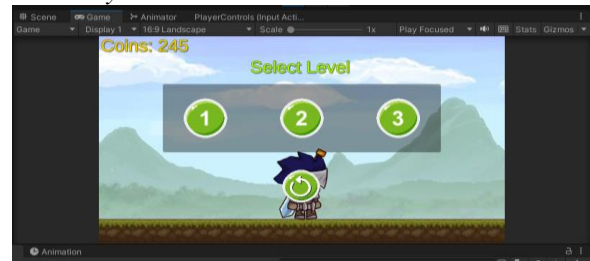
1. *Main Menu*



Gambar 7. *Main Menu*

Pada Gambar 7 merupakan tampilan *Main Menu*, terdapat *backgorund* pegunungan dengan judul *Game*, *Button Shop*, *Button Play*, Jumlah *Coin*, *Button Settings*, *Button Score*, dan *Button Quit*.

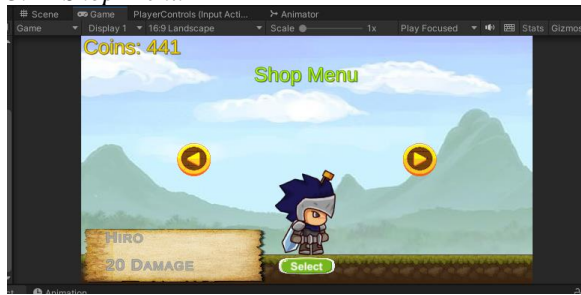
2. *Play Menu*



Gambar 8. *Play Menu*

Pada Gambar 8 merupakan tampilan *Play Menu* yang digunakan untuk memilih level yang akan dimainkan, dibagian tengah terdapat *button* Level 1, 2, dan 3.

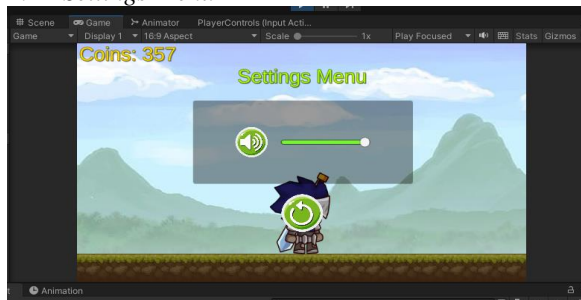
3. Shop Menu



Gambar 9. Shop Menu

Pada Gambar 9 merupakan tampilan dari Shop Menu yang digunakan untuk memilih atau membeli karakter dengan coin yang telah dikumpulkan

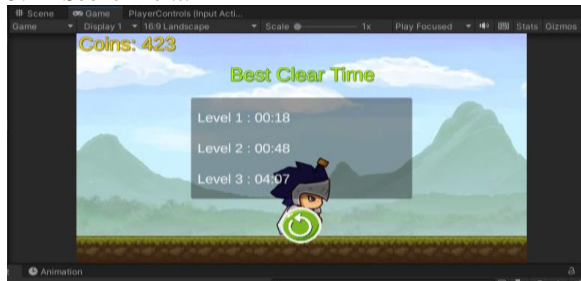
4. Settings Menu



Gambar 10. Settings Menu

Pada Gambar 10 merupakan tampilan dari Settings Menu yang digunakan untuk mengatur suara dari game.

5. Score Menu

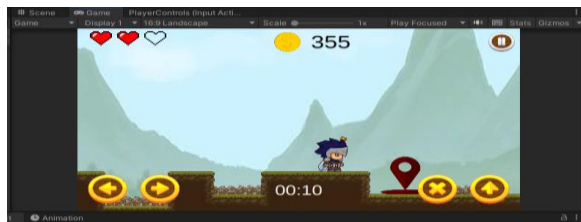


Gambar 11. Score Menu

Pada Gambar 11 merupakan tampilan dari Score Menu yang digunakan untuk menampilkan best time atau waktu tercepat menyelesaikan level 1, level 2, dan level 3. Jika ada pemain yang dapat melewati rekor sebelumnya maka waktu tercepat pun ikut berubah.

4.2 Pembuatan Level

1. Level 1



Gambar 12. Tampilan Level 1

Pada Gambar 12 merupakan tampilan level 1, Map Level 1 memiliki lebih sedikit trap, enemy, dan coin dari level 2.

2. Level 2



Gambar 13. Tampilan Level 2

Pada Gambar 13 merupakan tampilan level 2, jumlah enemy dan coin pada level 2 yaitu 2 kali dari jumlah enemy dan coin pada level 1.

3. Level 3



Gambar 14. Tampilan Level 3

Pada Gambar 14 merupakan tampilan level 3, pada level 3 terdapat 2 boss, sedikit coin dan tidak ada trap. Ketika boss pertama mati, boss kedua akan muncul, dan ketika boss kedua mati akan menampilkan epilog game.

4.3 Pengujian Metode Diagram FSM

Pengujian Metode digunakan untuk mengetahui fungsi Finite State Machine pada game Adventure Quest seperti pada Tabel 2.

Tabel 2. Pengujian Metode Diagram FSM

No	Fungsi	Tampilan	Hasil
1	Enemy mengejar player	enemy mengejar Player	Diterima
2	Enemy patroli	Enemy Patroli	Diterima
3	Enemy kembali ke posisi awal	Enemy patroli kembali ke posisi awal	Diterima
4	Enemy menyerang Player	Enemy menyerang player dan memberikan damage	Diterima
5	Enemy mati	Enemy mati dan objek enemy menghilang dari game	Diterima
6	Enemy Boss 1 dan 2 mengejar Player	Boss 1 dan 2 mengejar Player	Diterima
7	Enemy Boss 1 dan 2 menyerang player	Boss 1 dan 2 menyerang player	Diterima

No	Fungsi	Tampilan	Hasil
8	Enemy Boss 1 dan 2 berubah ke mode Rage	Boss 1 dan 2 berubah ke mode rage	Diterima
9	Enemy 1 dan 2 Boss mati	Boss 1 dan 2 Mati	Diterima
10	Enemy Boss 2 muncul ketika boss 1 mati	Boss 2 muncul setelah boss 1 mati	Diterima
11	Life Point Player = 0 sebelum checkpoint	Player mati dan respawn ke tempat awal	Diterima
12	Life Point Player = 0 sesudah checkpoint	Player mati dan respawn ke tempat checkpoint	Diterima

Berdasarkan Pengujian Metode Diagram FSM pada Tabel 2 didapatkan hasil dapat Diterima dan Tampilan sesuai dengan Fungsi FSM.

4.4 Pengujian Device

Pengujian device dilakukan untuk mengidentifikasi spesifikasi perangkat Android yang mampu menjalankan game Adventure Quest seperti pada Tabel 3.

Tabel 3. Pengujian Device

No	Nama Device	Resolusi & RAM	Android	Hasil
1	Oppo A37F	720 x 1280 piksel & 2GB RAM	5	Gagal
2	Oppo A57	720 x 1280 piksel & 3GB RAM	6	Berhasil
3	Samsung Galaxy S8+	1440 x 2960 piksel & 4GB RAM	7	Berhasil
4	Redmi A4	1280 x 720 piksel (HD+) & 2GB RAM	8	Berhasil
5	Honor 9 Lite	2160 x 1080 piksel & 3GB RAM	9	Berhasil
6	Oppo F7	1080 x 2280 piksel & 4GB RAM	10	Berhasil
7	Redmi Note 9 Pro	1080 x 2400 piksel (FHD+) & 6GB RAM	11	Berhasil
8	Redmi Note 8 Pro	1080 x 2340 piksel (FHD+) & 6GB RAM	12	Berhasil
9	Xiaomi 12	2400 x 1080 piksel (FHD+) & 8GB RAM	13	Berhasil

Dari pengujian Tabel 3 menunjukkan bahwa game dapat dijalankan dari perangkat dengan resolusi 720 hingga 1080 piksel, RAM 2GB hingga 8GB, dan versi Android 6 hingga versi Android 13.

4.5 Pengujian Control Player

Pengujian ini dilakukan untuk mengetahui kontrol gerakan Player pada game Adventure Quest seperti pada Tabel 4.

Tabel 4. Pengujian Control Player

No	Tombol	Fungsi	Hasil
1	Kiri	Player bergerak ke kiri	Diterima
2	Kanan	Player bergerak ke kanan	Diterima
3	Atas	Player melompat	Diterima
4	X	Player menyerang	Diterima

Dari pengujian Tabel 4 menunjukkan bahwa tombol kiri, kanan, atas, dan X memberikan hasil sesuai yang diharapkan.

4.6 Pengujian User

Pengujian user game akan dilakukan dengan menggunakan penyebaran kuisioner google form. Pengujian dilakukan untuk memberikan informasi bahwa game berfungsi dengan baik dan hasil pengujian dapat dilihat pada Tabel 5 di bawah

Tabel 5. Pengujian User

No	Pernyataan	Jawaban				
		SS	S	C	TS	STS
1	Pendapat anda tentang desain karakter dalam permainan sudah menarik	4	10	7	1	0
2	Tampilan didalam game sudah menarik	2	12	8	0	0
3	Terdapat bug atau kesalahan dalam permainan	4	3	10	4	1
4	Kesulitan level game dalam bermain sesuai dengan tingkatan level yang ada	6	11	4	1	0
5	Penggunaan kontrol pada permainan mudah dioperasikan dan responsif	5	10	6	1	0
6	Terdapat bagian dalam permainan yang tidak masuk akal atau tidak masuk logika	4	6	8	4	0
7	Kualitas suara dan efek suara pada permainan sesuai dengan suasana dalam menghidupkan pengalaman bermain	5	9	6	2	0
8	Anda merasa bosan ketika bermain game ini	4	4	7	7	0

No	Pernyataan	Jawaban				
		SS	S	C	TS	STS
9	Menemukan fitur-fitur yang kurang dalam permainan	3	5	11	3	0
10	Permainan mudah untuk diselesaikan	4	12	5	1	0
Total		41	82	72	24	1

SS = Sangat Setuju TS = Tidak Setuju
 S = Setuju STS = Sangat Tidak Setuju
 C = Cukup
 Jumlah Pertanyaan : 10
 Jumlah Pengguna : 22
 Faktor Pembagi : $10 * 22 = 220$

Dari pengujian *User* pada Tabel 5 terdapat 10 pertanyaan dan 22 pengguna yang menjawab kuesioner lalu didapatkan hasil faktor pembagi 220, dengan total nilai Sangat Setuju 41, Setuju 82, Cukup 72, Tidak Setuju 24, dan Sangat Tidak Setuju 1.

Tabel 6. Presentase Responden pada pengujian *User*

No	Persentase	Nilai
1	<i>User</i> memilih Sangat Setuju	$(40/210 * 100\%) = 18,63\%$
2	<i>User</i> memilih Setuju	$(78/210 * 100\%) = 37,27\%$
3	<i>User</i> memilih Cukup	$(68/210 * 100\%) = 37,72\%$
4	<i>User</i> memilih Tidak Setuju	$(23/210 * 100\%) = 10,90\%$
5	<i>User</i> memilih Sangat Tidak Setuju	$(1/210 * 100\%) = 0,45\%$

Dari perhitungan pada Tabel 6 yang ditujukan pada 22 pengguna didapatkan hasil persentase bahwa 18,63% menyatakan Sangat Setuju, 37,27% menyatakan Setuju, 37,72% menyatakan Cukup, 10,90% menyatakan Tidak Setuju, dan 0,45% menyatakan Sangat Tidak Setuju, sehingga dapat disimpulkan bahwa *game* yang dirancang sudah baik.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dari perancangan dan implementasi pada *game* Adventure Quest maka dapat diambil kesimpulan yaitu, Metode *Finite State Machine* dapat diimplementasikan pada *game* Adventure Quest dengan indikasi monster dapat mendekati *Player* dan dapat kembali patroli kembali jika tidak berada dalam jangkauan. Hasil implementasi pengujian pada perangkat menunjukkan *game* dapat dijalankan dari perangkat dengan resolusi 720 hingga 1080 piksel, RAM 2GB hingga 8GB, dan versi *Android* 6 hingga versi *Android* 13. Hasil implementasi pengujian *control player* berjalan sesuai dengan hasil yang diharapkan. Hasil pengujian *user* didapatkan *game* yang dirancang sudah baik dengan hasil presentase 18,63% menyatakan Sangat Setuju, 37,27% menyatakan Setuju, 37,72% menyatakan Cukup, 10,90% menyatakan Tidak Setuju, dan 0,45% menyatakan Sangat Tidak Setuju. Adapun saran sebagai acuan terhadap penelitian atau pengembangan selanjutnya yaitu, Penambahan Fitur nama pada *Highscore game* agar pemain mengetahui siapa yang

mendapat skor tertinggi di *game* tersebut, penambahan *item* untuk menambah *LifePoint* karakter agar karakter dalam *game* tidak mudah mati, membuat karakter daerah (Majapahit) untuk digunakan pada *game* yang dibuat agar *game* lebih menarik.

DAFTAR PUSTAKA

- [1] M. R. Febrianto and T. Fatimah, "Penerapan Metode Finite State Machine Game 2D Adventure Kebokicak Dan Surontanu Berbasis Android," *Semin. Nas. Mhs. Fak. Teknol. Inf. Jakarta-Indonesia*, no. September, pp. 867–874, 2022.
- [2] T. F. A. Esa, "Sistem Informasi Unit Kegiatan Mahasiswa Taekwondo Universitas Islam Indonesia Fakultas Teknologi Industri Universitas Islam Indonesia Yogyakarta," 2016.
- [3] I. Setiawan, "Perancangan Software Embedded System Berbasis FSM," *J. Tek. Elektro*, pp. 1–2, 2006.
- [4] G. Mau, "Rancang Bangun Game 2D Shooter Platformer Menggunakan Metode Finite State Machine," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 3, no. 1, pp. 117–122, 2019.
- [5] A. T. Ramadan and M. Hardjianto, "Penerapan Finite State Machine Pada Game 'Pendekar Cisadane' Berbasis Android," *Semin. Nas. Mhs. Fak. Teknol. Inf. Jakarta-Indonesia*, no. September, pp. 1065–1072, 2022, [Online]. Available: <https://senafiti.budiluhur.ac.id/index.php/senafiti/index>
- [6] A. Yusnita, A. Y. Rangan, and F. Setiawan, "Membangun Game Fun Animal Puzzle Menggunakan Algoritma Shuffle Random," *Sebatik*, vol. 15, no. 1, pp. 1–6, 2016, doi: 10.46984/sebatik.v15i1.68.
- [7] A. O. P. Dewi, "Kecerdasan Buatan sebagai Konsep Baru pada Perpustakaan," *Anuva J. Kaji. Budaya, Perpustakaan, dan Inf.*, vol. 4, no. 4, pp. 453–460, 2020, doi: 10.14710/anuva.4.4.453-460.
- [8] I. Maulana and J. C. Chandra, "Implementasi Primas™s Algorithm Dan A* Pathfinding Pada Game Roguelike Sederhana," *Garuda - Garba Rujukan Digit.*, vol. 1, no. 3, pp. 1018–1025, 2018, [Online]. Available: <https://garuda.ristekbrin.go.id/documents/detail/1838518>
- [9] N. Masri *et al.*, "Survey of Rule-Based Systems," vol. 3, no. 7, pp. 1–22, 2019.
- [10] Y. R. Septyapratama, "Impelementasi Case Based Reasoning Pada Kerusakan Yang Terjadi Di Vending," 2019.
- [11] P. Indriyanti, M. H. Fazalika, and M. Mujiono, "Penerapan Decision tree dalam pengambilan keputusan untuk pemain Texas Holdem Poker," *J. Ilm. FIFO*, vol. 12, no. 2, p. 167, 2021, doi: 10.22441/fifo.2020.v12i2.006.