

PERANCANGAN GAME 2D TOP-DOWN SHOOTER "OUTBREAK25" MENGUNAKAN METODE FINITE STATE MACHINE (FSM) DAN PROCEDURAL GENERATION

Rohibun Rahman, Agung Panji Sasmito, Deddy Rudhistiar

Teknik Informatika, Institut Teknologi Nasional Malang

Jalan Raya Karanglo km 2 Malang, Indonesia

1818072@scholar.itn.ac.id

ABSTRAK

Banyak game 2D *top-down shooter* masih menggunakan *level* statis dan musuh dengan perilaku sederhana sehingga cepat membosankan. Penelitian ini merancang *Outbreak25*, game 2D berbasis *desktop* dengan penerapan *Finite State Machine* (FSM) untuk perilaku musuh serta *procedural generation* untuk menciptakan *level* dinamis. Game dikembangkan menggunakan *Unity Engine* melalui tahapan perancangan, implementasi, dan pengujian. Pengujian dilakukan dengan metode *black box*, uji FSM, uji *procedural generation*, uji kontrol pemain, uji kompatibilitas perangkat, dan uji pengguna. Hasilnya menunjukkan seluruh fitur berjalan sesuai fungsinya, FSM mampu mengatur transisi perilaku musuh secara adaptif, *procedural generation* menghasilkan *level* berbeda di setiap sesi, kontrol pemain responsif, game kompatibel di berbagai perangkat, serta 70,51% pengguna memberikan penilaian positif. Dengan demikian, *Outbreak25* dinilai berhasil memberikan pengalaman bermain yang menarik dan layak dikembangkan lebih lanjut.

Kata kunci : *Finite State Machine, Game 2D, Kecerdasan Buatan, Procedural Generation, Top-Down Shooter.*

1. PENDAHULUAN

Game adalah aktivitas interaktif yang bertujuan memberikan hiburan, melatih fokus, serta strategi, dan diminati oleh berbagai kalangan usia. Salah satu elemen penting dalam game adalah kecerdasan buatan (AI), yang memungkinkan karakter non-pemain (NPC) berperilaku layaknya manusia. Penerapan AI, seperti *Finite State Machine* (FSM) membantu mengelola tingkah laku NPC dengan cara yang terorganisir dalam kondisi seperti “*Idle*”, “*Chase*”, atau “*Attack*”, membuat mereka lebih *realistis* dan efisien dalam komputasi [4].

Pemilihan objek penelitian game 2D *top-down shooter Outbreak25* didasarkan pada kebutuhan menghadirkan permainan yang lebih dinamis dan menantang. Banyak game sejenis masih menggunakan *level* statis dan perilaku musuh sederhana, sehingga cepat membosankan. *Outbreak25* dirancang dengan menggabungkan metode *Finite State Machine* (FSM) untuk perilaku musuh dan *procedural generation* untuk *level* permainan, sehingga mampu memberikan pengalaman bermain yang variatif serta layak dijadikan objek penelitian.

Game *Outbreak25* perlu dirancang karena banyak game 2D *top-down shooter* masih menggunakan *level* statis dan musuh dengan perilaku sederhana, sehingga cepat membuat pemain bosan. Masalah penelitian terletak pada bagaimana menghadirkan permainan yang lebih menarik dan adaptif dengan menerapkan *Finite State Machine* (FSM) untuk perilaku musuh yang *realistis* serta *procedural generation* untuk membentuk *level* yang selalu berbeda.

Metode *Finite State Machine* (FSM) dipilih karena sederhana, efisien, dan mampu mengatur perilaku musuh secara dinamis, sedangkan *procedural*

generation digunakan untuk menciptakan *level* permainan yang selalu berbeda sehingga meningkatkan variasi dan *replayability*. Metode lain seperti *behavior tree* atau *desain level manual* memang ada, tetapi dinilai lebih kompleks, memakan waktu, atau kurang mendukung variasi dinamis. Oleh karena itu, FSM dan *procedural generation* dianggap paling sesuai dengan tujuan penelitian ini.

Selain itu, *Procedural Generation* memungkinkan pembuatan konten game seperti *level* dan item secara otomatis, menghemat waktu dan meningkatkan variasi permainan. Berdasarkan hal tersebut, penelitian ini mengusulkan pengembangan game *Outbreak25* dengan mengimplementasikan metode FSM untuk perilaku musuh dan *Procedural Generation* untuk membentuk lingkungan permainan yang dinamis [5].

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Dalam penelitian yang berjudul “Perancangan Game 2d “*Barbossa The Savior*” Menggunakan *Finite State Machine*” yang bertujuan menerapkan metode *Finite State Machine* untuk mengelola perilaku NPC (Non-Playable Character) dalam game. FSM digunakan untuk mengatur transisi antar keadaan, seperti patroli, mengejar, menyerang, dan mati, sehingga AI dalam game menjadi lebih *realistis* dan responsif terhadap interaksi pemain [6].

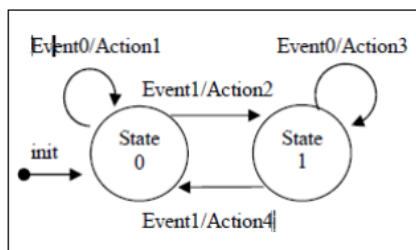
Procedural Content Generation (PCG) untuk memuat konten game secara otomatis. PCG digunakan untuk menghasilkan elemen permainan selain karakter utama, seperti rintangan, latar belakang, dan item yang muncul selama gameplay. Metode ini memastikan variasi dalam permainan tanpa harus membuat setiap elemen secara manual, sehingga meningkatkan efisiensi pengembangan dan pengalaman bermain

yang lebih dinamis. Hasil penelitian menunjukkan bahwa PCG berhasil diterapkan dalam *game*, memungkinkan konten berubah secara otomatis sesuai progres pemain, yang sesuai dengan konsep *Endless Runner* dalam *game*. [11].

Dalam jurnal “Procedural Content Generation Pada *Game World Exploration Sandbox* Menggunakan Algoritma Perlin Noise” membahas penerapan *Procedural Content Generation* dengan *Binary Tree Algorithm* untuk menghasilkan *level* labirin secara otomatis. Metode ini memastikan setiap *level* bervariasi, valid, dan dapat dimainkan, sekaligus meningkatkan efisiensi waktu serta mengurangi beban pengembang dalam mendesain *level* secara manual. [5].

2.2. Finite State Machine

Finite State Machine (FSM) adalah metode pengendalian sistem yang mengatur transisi antar keadaan berdasarkan peristiwa tertentu, memungkinkan NPC dalam *game* berpindah dari satu aksi ke aksi lainnya, seperti berjalan, menyerang, atau mati, sehingga menciptakan perilaku yang lebih dinamis dan realistis. [10].



Gambar 1. Diagram Finite State Machine

2.3. Procedural Generation

Procedural Content Generation (PCG) adalah metode otomatisasi dalam pembuatan konten menggunakan algoritma untuk menghasilkan data dengan ukuran penyimpanan kecil tetapi bervariasi dalam jumlah besar. Dalam *game*, PCG digunakan untuk menciptakan elemen seperti *level*, karakter, dan musuh secara dinamis, meningkatkan efisiensi pengembangan serta nilai *replayability* agar pengalaman bermain selalu unik setiap kali dimainkan [5].

2.4. Unity Engine

Unity adalah sebuah aplikasi yang digunakan untuk mengembangkan *game* multi-platform dan dirancang agar mudah digunakan oleh pengembang. *Editor* pada *Unity* memiliki antarmuka pengguna yang sederhana dan intuitif, sehingga memudahkan pengguna dalam merancang dan mengembangkan *game*. Proses pengembangan *editor* ini telah melalui ribuan jam kerja, dengan tujuan untuk menyempurnakan fitur-fitur yang ada dan meningkatkan pengalaman pengguna. Berkat upaya tersebut, *Unity* berhasil menjadi salah satu *editor game* terbaik dan menduduki peringkat teratas dalam kategori pengembangan *game* [12].

3. METODE PENELITIAN

Untuk mencapai tujuan dalam perancangan *Game Outbreak25* menggunakan Metode *Procedural Generation* dan *Finite State Machine* (FSM) berbasis Desktop, langkah-langkah berikut harus diambil:

- Studi Literatur: Penelitian literatur dilakukan dengan mencari dan menganalisis berbagai sumber, seperti buku, *e-book*, dan jurnal ilmiah di internet yang relevan dengan pengembangan *game top-down shooter*, metode FSM untuk kecerdasan buatan musuh, serta penerapan *Procedural Generation* dalam pembuatan lingkungan permainan.
- Pengumpulan Data: Pada tahap ini dilakukan pengumpulan data yang mendukung pengembangan *game*, seperti referensi desain karakter, struktur *level*, pola pergerakan AI musuh, serta teknik *Procedural Generation*. Selain itu, dilakukan analisis dari *game* sejenis sebagai bahan pembandingan dalam menyusun fitur dan mekanik *game Outbreak25*.
- Perancangan Sistem: Tahap ini meliputi perancangan flowchart *gameplay*, diagram *Finite State Machine* (FSM) untuk perilaku musuh zombie, flowchart untuk *procedural generation* map, struktur *menu*, *storyboard*, serta desain aset karakter dan lingkungan. Perancangan dilakukan dengan mempertimbangkan modularitas dan fleksibilitas sistem agar dapat diimplementasikan secara efisien dalam *Unity*.
- Implementasi: Pada tahap ini dilakukan pembangunan *game Outbreak25* di dalam *Unity*, mulai dari penerapan aset karakter, pembuatan logika FSM untuk AI zombie, hingga pengaturan algoritma *Procedural Generation* untuk membentuk *level* permainan yang berbeda-beda setiap sesi. Implementasi dilakukan secara bertahap sesuai struktur rancangan yang telah dibuat.
- Pengujian Sistem: Pengujian Sistem: Proses pengujian dilakukan setelah semua komponen selesai. Pengujian fungsional dan pengujian kinerja dilakukan untuk mengevaluasi keberhasilan *game* yang dikembangkan.

3.1. Pengujian Black Box

Pengujian *Black Box* pada tujuh fitur utama *game Outbreak25* menunjukkan hasil 100% keberhasilan, yang berarti seluruh fungsi berjalan sesuai harapan tanpa ditemukan *error*. Metode ini menguji sistem berdasarkan *respons* terhadap *input*, seperti *menu* utama, tombol suara, dan transisi *level*. Hasil uji menunjukkan setiap *input* mendapat *output* yang tepat, misalnya *menu* “Mulai *Game*” berhasil menampilkan permainan, tombol “*Sound*” berfungsi mematikan dan menyalakan suara, serta tombol *level* membawa pemain ke lingkungan permainan sesuai spesifikasi. Secara keseluruhan, struktur *menu* dan kontrol dasar *game* telah terimplementasi dengan baik, stabil, dan nyaman digunakan oleh pemain.

3.2. Pengujian *Finite State Machine*

Pengujian FSM pada musuh dalam *game Outbreak25* menunjukkan hasil sesuai rancangan, di mana musuh berpindah dari kondisi *Patrol* saat pemain berada di luar radius, ke *Chase* ketika masuk radius ± 7 unit, dan ke *Attack* saat jarak kurang dari 2 unit. Jika pemain menjauh, musuh kembali ke *Patrol* atau tetap *Chase* sesuai jarak terakhir, sedangkan ketika HP musuh habis, FSM membawa musuh ke kondisi *Death* dan menghilang. Seluruh transisi antar state berjalan lancar tanpa konflik logika maupun kesalahan animasi, sehingga perilaku musuh tampil dinamis, realistis, dan menambah tantangan permainan.

3.3. Pengujian *Procedural Generation*

Pengujian *Procedural Generation* pada *game Outbreak25* menunjukkan bahwa setiap sesi permainan menghasilkan peta dengan tata letak, jalur, penempatan item, musuh, dan posisi antivirus yang berbeda namun tetap logis dan dapat diakses. Seluruh ruangan saling terhubung tanpa area tertutup, sehingga validasi peta berjalan baik. Variasi ini membuktikan bahwa sistem *Procedural Generation* berhasil menciptakan lingkungan permainan yang unik, menantang, serta mendukung elemen kejutan dan *replayability*, sehingga pemain tidak cepat bosan karena tidak bisa menghafal jalur permainan.

3.4. Pengujian *Control Player*

Pengujian kontrol pemain menunjukkan bahwa karakter bergerak mulus sesuai *input* tombol W, A, S, dan D, berhenti tepat saat tombol dilepas, serta mampu menembak menggunakan klik kiri *mouse* dengan arah peluru mengikuti kursor secara *real-time*. Sistem deteksi tabrakan berfungsi baik karena karakter berhenti saat menyentuh objek tanpa menembusnya. Hasil ini membuktikan bahwa kontrol telah responsif, presisi, dan optimal, sehingga mendukung kenyamanan serta strategi pemain dalam genre shooter.

3.5. Pengujian *Device*

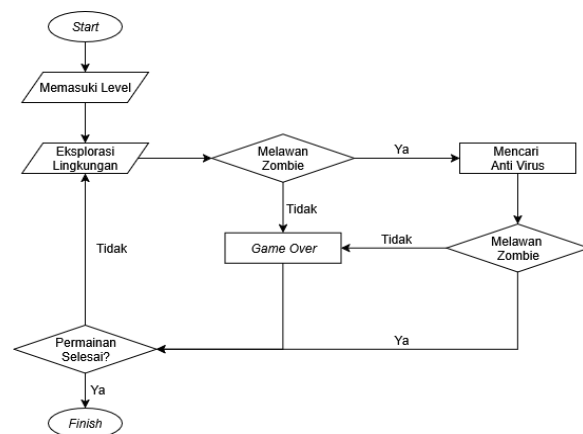
Pengujian perangkat menunjukkan bahwa *game Outbreak25* dapat dijalankan di berbagai spesifikasi laptop/komputer berbasis Windows. Pada perangkat berspesifikasi rendah (Intel Core i3, RAM 4 GB, GPU onboard), game tetap berjalan meski terjadi penurunan frame pada kondisi intensif. Pada spesifikasi menengah (Ryzen 5, RAM 8 GB, GPU Radeon Vega 8), performa sangat baik dengan kontrol responsif tanpa kendala. Sedangkan pada spesifikasi tinggi (Core i7, RAM 16 GB, GPU GTX), game berjalan stabil dengan grafis maksimal dan tanpa lag. Dengan demikian, *Outbreak25* kompatibel di semua perangkat, namun untuk pengalaman optimal disarankan minimal prosesor setara Core i5/Ryzen 3 dengan RAM 8 GB.

3.6. Pengujian *User*

Pengujian melibatkan 13 pengguna dengan 6 pernyataan, sehingga total faktor pembagi adalah 78. Hasilnya, sebanyak 55 jawaban “Iya” (70,51%) menunjukkan penilaian positif terhadap fitur, desain, peta, cerita, suara, dan tema *game Outbreak25*, sedangkan 23 jawaban “Tidak” (29,48%) menandakan masih ada aspek yang perlu diperbaiki, terutama pada kualitas suara dan keseimbangan *gameplay*.

3.7. Flowchart

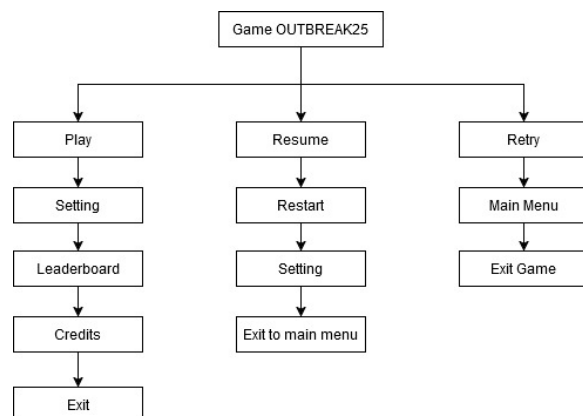
Perancangan flowchart *game* berfungsi untuk mengetahui alur proses dari alur dimulai dari start hingga end *game* seperti pada Gambar 2.



Gambar 2. Flowchart

3.8. Struktur *menu*

Dalam permainan *Top-down shooter Breakout25* terdapat beberapa *menu*, yaitu *menu* utama, *menu level*, *menu* tentang, *menu* pengaturan, dan *menu* keluar. Struktur diagram *menu* terlihat seperti yang ditunjukkan pada Gambar 3



Gambar 3. Struktur *menu*

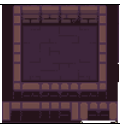
3.9. *Desain Karakter*

Perancangan karakter dan *item* adalah proses di mana dalam *game* terdapat karakter-karakter baik yang dapat dikendalikan pemain maupun yang tidak, seperti musuh. *Desain* karakter dan *item* dalam permainan *Outbreak25* ditampilkan pada Tabel 1.

Tabel 1. Desain karakter

Karakter	Keterangan
	Karakter utama dari <i>game</i> .
	Yaitu karakter musuh yang dapat memberikan serangan Ketika karakter utama mendekat.
	Yaitu karakter musuh yang lebih kuat dari Zombie 1 dimana dapat memberikan serangan Ketika karakter utama mendekat.

Tabel 2. Desain Enviroment

Karakter	Keterangan
	Sebagai properti pijakan untuk karakter utama dan musuh. Juga untuk sebagai pembatas map.
	Tempat mengambil amunisi dan heal saat pertarungan.

4. HASIL DAN PEMBAHASAN

4.1. Perancangan Game

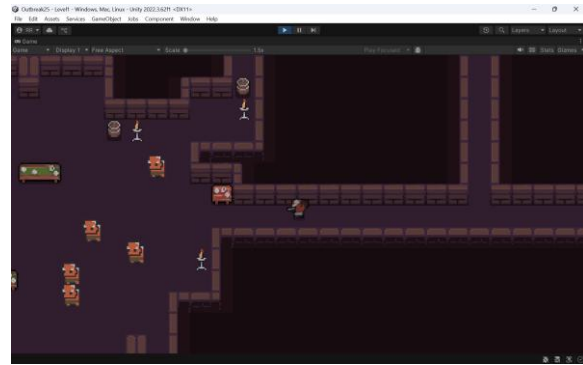
Perancangan game *Outbreak25* yang telah dilakukan diimplementasikan dan menghasilkan Desain visual permainan mulai dari tampilan awal (*menu* utama), *level* 1 sampai *level* 3, dengan perbedaan di setiap *level*. yang ditetapkan secara prosedural, baik dari segi layout ruangan, posisi musuh, item, maupun tingkat kesulitan.

4.2. Implementasi FSM

Dalam permainan *Outbreak25*, perilaku zombie sebagai musuh diatur dengan metode *Finite State Machine* (FSM) yang diawali dari kondisi awal (*initial state*) yaitu “idle” atau “patrol”, di mana zombie bergerak secara acak tanpa target spesifik. Ketika pemain memasuki radius deteksi tertentu, FSM akan bertransisi ke keadaan “chasing”, yaitu kondisi di mana zombie mulai mengejar pemain secara aktif.

Jika pemain berada dalam jangkauan serangan, FSM akan berpindah ke keadaan “attacking”. Pada tahap ini, zombie akan menjalankan animasi serangan, dan sistem akan memeriksa apakah serangan tersebut mengenai pemain. Jika serangan berhasil, maka sistem akan mengurangi *HP* pemain sesuai dengan kerusakan (*damage*) yang ditentukan. Sebaliknya, jika pemain berhasil menghindari atau menjauh, FSM akan kembali ke status sebelumnya, seperti “chasing” atau bahkan “patrol”, tergantung jarak pemain terhadap zombie.

Seluruh transisi antar status ini berlangsung secara dinamis dan real-time, menciptakan musuh yang adaptif dan memberikan tantangan strategis kepada pemain



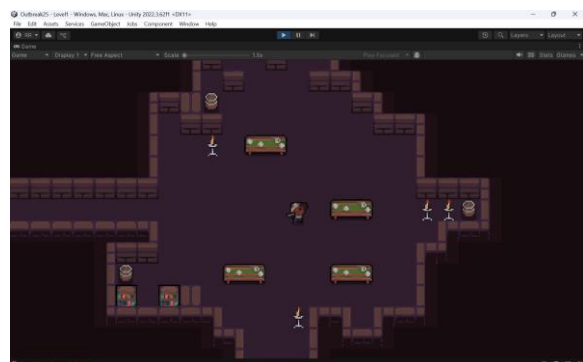
Gambar 3. Implementasi FSM

4.3. Implementasi Procedural Generation

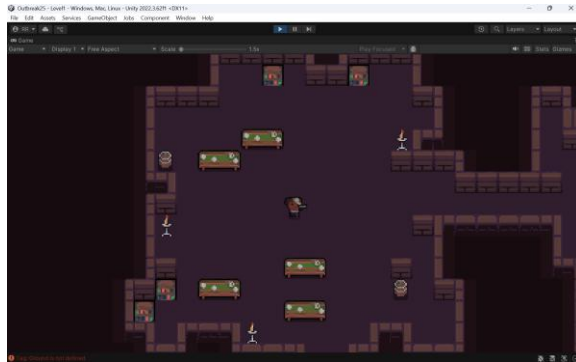
Implementasi *Procedural Generation* dalam game *Outbreak25* bertujuan untuk menciptakan lingkungan permainan yang selalu berbeda setiap kali dimainkan, sehingga meningkatkan variasi dan nilai replayability. Proses ini dimulai dengan pembangkitan tata letak peta menggunakan algoritma acak, di mana ruangan-ruangan, koridor, serta posisi elemen penting seperti item, musuh, dan antivirus ditentukan secara dinamis.

Lingkungan permainan dipetakan dalam bentuk grid yang diisi oleh tile, seperti lantai, dinding, dan objek interaktif. Setiap kali pemain memulai *level*, sistem akan menyusun ulang kombinasi tile untuk menciptakan struktur labirin yang baru. Tata letak ini tidak hanya memengaruhi navigasi pemain, tetapi juga memengaruhi jalur gerak musuh dan strategi yang harus diterapkan oleh pemain.

Dengan pendekatan ini, pemain tidak dapat mengandalkan hafalan pola *level*, karena setiap *level* memiliki susunan ruangan dan posisi rintangan yang unik. Selain itu, penempatan musuh dan item juga dilakukan secara acak namun tetap dalam batasan tertentu untuk menjaga keseimbangan gameplay. Implementasi ini memungkinkan *Outbreak25* menawarkan pengalaman bermain yang segar dan menantang pada setiap sesi permainan.



Gambar 4. Implementasi Procedural Generation 1

Gambar 5. Implementasi *Procedural Generation 2*

4.4. Hasil Pengujian

Outbreak25 menerapkan berbagai ujian seperti pengujian blackbox, pengendali, alat, dan konsumen.

Uji coba *black box* dilakukan untuk melihat bagaimana aplikasi berfungsi dan memastikan bahwa *input* dan *output* berjalan dengan baik di *Desktop*. Tabel 3 menunjukkan hasil dari pengujian ini

Tabel 3. Pengujian *Black Box*

No	Fitur	Aksi	Output	Hasil
1.	Tombol Play	Klik “Play” pada <i>menu</i> utama	Muncul pilihan <i>Level 1 – Level 3</i> dan tombol “Back” ke <i>menu</i> utama	Berhasil
2	Tombol <i>Level 1–3</i>	Klik salah satu <i>level</i>	Masuk ke peta permainan <i>level</i> yang dipilih	Berhasil
3	Tombol Back	Klik tombol “Back”	Kembali ke <i>menu</i> utama	Berhasil
4	Tombol Setting	Klik “Setting”	Muncul pengaturan <i>FX</i> , <i>Sound</i> , dan tombol “Back” ke <i>menu</i> utama	Berhasil
5	Tombol pengaturan <i>FX</i> dan <i>Sound</i>	Atur Nilai <i>FX</i> dan <i>Sound</i>	Volume suara dan efek berubah sesuai slider	Berhasil
6	Tombol Back (dari <i>menu</i> Setting)	Klik tombol “Back”	Kembali ke <i>menu</i> utama	Berhasil
7	Tombol Exit	Klik “Exit”	Game keluar/ditutup dengan aman	Berhasil

Berdasarkan Tabel 3 Pengujian *Black Box* dilakukan 5 pengujian *input* sehingga di dapatkan hasil 5 *input* berhasil.

Pengujian ini berisi skenario pengujian untuk memastikan perilaku musuh (*enemy*) dalam sebuah

game sesuai dengan yang diharapkan. Musuh menyerang ketika pemain berada dalam jangkauan pandangan musuh, musuh diharapkan mati ketika nyawa (HP) musuh mencapai nol.

Tabel 4. Pengujian *FSM*

No	Aksi	Hasil Yang Diharapkan	Hasil Pengujian
1	Musuh mendeteksi pemain dalam radius tertentu	Musuh berpindah dari <i>Patrol</i> ke <i>Chase</i>	Sesuai
2	Musuh berada sangat dekat dengan pemain	Musuh berpindah dari <i>Chase</i> ke <i>Attack</i>	Sesuai
3	Pemain menjauh dari musuh	Musuh kembali ke <i>Patrol</i> atau tetap <i>Chase</i>	Sesuai
4	Musuh diserang hingga HP = 0	Musuh berpindah ke <i>state Death</i> dan menghilang	Sesuai

Berdasarkan Tabel 4 Pengujian *FSM* dilakukan terhadap NPC, sehingga di dapatkan 2 hasil pengujian sesuai.

Pengujian ini bertujuan untuk memastikan bahwa Map dapat berubah menggunakan sistem *Procedural*

Generation dengan benar untuk map ketika berganti *level*. Map seharusnya bisa berubah atau berbeda bentuk untuk setiap *level*.

Tabel 5. Pengujian *Procedural Generation*

No	Aksi	Hasil Yang Diharapkan	Hasil Pengujian
1	Game memulai <i>level</i> baru	Tata letak ruangan, lorong, dan posisi item serta musuh dihasilkan secara acak	Sesuai
2	Menjalankan ulang permainan	Layout berbeda dari sesi sebelumnya	Sesuai
3	Pemain menjelajahi peta	Lingkungan berbeda dari sesi permainan sebelumnya (ruangan dan jalur tidak dapat dihafal)	Sesuai
4	Item dan musuh muncul di area baru	Posisi spawn item (amunisi, heal) dan musuh ditentukan secara prosedural dan tetap seimbang	Sesuai
5	Jalur antar ruangan	Semua ruangan terhubung dan dapat dijelajahi	Sesuai

Berdasarkan Tabel 5 Pengujian *Procedural Generation* dilakukan terhadap NPC, sehingga di dapatkan 3 hasil pengujian sesuai.

Pengujian kontrol pemain meliputi menilai setiap fungsi yang sudah diterapkan pada tombol-tombol

untuk mengendalikan karakter pemain dalam permainan itu. Uji coba pada Tabel 6 Pengujian Kontrol Pemain.

Tabel 6. Pengujian *Controll player*

No	Aksi	Input	Hasil Yang Diharapkan	Hasil
1	Bergerak ke empat arah	Tombol W, A, S, D	Karakter bergerak sesuai arah	Sesuai
2	Menembak ke arah kursor	Klik kiri mouse	Karakter menembak ke arah yang ditunjuk	Sesuai
3	Berhenti saat tombol dilepas	Tombol W/A/S/D dilepas	Karakter berhenti bergerak	Sesuai
4	Arah tembakan mengikuti kursor	Gerakkan mouse ke sudut lain	Tembakan mengikuti arah kursor	Sesuai
5	Menabrak dinding atau objek	Menyentuh obstacle	Gerakan berhenti, tidak bisa menembus objek	Sesuai

Berdasarkan Tabel 6 Pengujian *Controll player* dilakukan terhadap player, sehingga di dapatkan 5 hasil pengujian sesuai.

Pengujian ini dilakukan untuk mengetahui spesifikasi device yang dapat menjalankan game *Outbreak25*

Tabel 7. Pengujian *Device*

No	Device	Spesifikasi	Hasil
1.	HP Pavilion 14-ec0007AU	- Prosesor: Intel Core i3 generasi ke-12 - RAM: 4GB - Penyimpanan: SSD 512GB - Layar: 15,6 inci	Berhasil
2.	Advan WorkPlus	- Prosesor: AMD Ryzen 7 gen 7 - RAM: 8GB - Penyimpanan: SSD 512GB	Berhasil
3.	Advan AI Gen	- Prosesor: AMD Ryzen 8 gen 8 - RAM: 16GB - Penyimpanan: SSD 512GB	Berhasil
4.	MSI Modern 14	- Intel Core i5-10210U - RAM 8GB - UHD Graphics	Berhasil
5.	HP 14s-fq0591AU	- AMD Ryzen 5 5500U - RAM 8GB - Radeon Vega 7	Berhasil

Berdasarkan Tabel 7 Pengujian *Device* dilakukan 3 pengujian, sehingga di dapatkan 3 hasil pengujian berhasil.

Pengujian dengan tujuan untuk mengetahui apakah sistem game sudah berjalan dengan baik atau belum ini dilakukan terhadap 14 orang dan memiliki total perhitungan seperti pada Tabel 8 berikut:

Tabel 8. Pengujian *User*

No.	Pertanyaan	Jawaban	
		Iya	Tidak
1.	Apakah fitur kontrol dan interaksi dalam game <i>Outbreak25</i> berfungsi dengan baik?	9	4
2.	Apakah <i>Desain</i> karakter utama dan musuh zombie terlihat menarik dan jelas?	9	4
3.	Apakah <i>Desain</i> peta permainan mudah dipahami dan membuat gameplay lebih seru?	9	4
4.	Apakah cerita game <i>Outbreak25</i> mudah dipahami dan menarik untuk diikuti?	10	3
5.	Apakah efek suara dan musik latar mendukung suasana permainan?	8	5
6.	Apakah tema zombie survival yang diusung sesuai dengan konsep game <i>top-down shooter</i> ?	10	3
Total		55	23

Jumlah pertanyaan : 6
 Jumlah user : 13
 Faktor pembagi : $6 \times 13 = 78$

Tabel 9. Persentase Responden pada Pengujian *User*

No.	Persentase	Nilai
1.	Persentase user memilih Iya	$(55 \div 78) \times 100\% = 70,51\%$
2.	Persentase user memilih Tidak	$(23 \div 78) \times 100\% = 29,48\%$

Dari hasil pengujian user terhadap game *Outbreak25*, diperoleh bahwa 70,51% responden memilih "Iya", sedangkan 29,48% responden memilih "Tidak". Persentase ini menunjukkan bahwa mayoritas pengguna memberikan penilaian positif terhadap fitur,

Desain karakter, desain peta, cerita, suara, dan tema game yang diusung. Namun, terdapat sebagian responden (29,48%) yang memberikan jawaban "Tidak", yang menunjukkan adanya beberapa aspek yang masih perlu diperbaiki, terutama pada kualitas suara dan keseimbangan gameplay.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian yang dilakukan, seluruh fitur utama pada game *Outbreak25* dinyatakan berhasil berjalan dengan baik. Pengujian Black Box terhadap 7 fitur menunjukkan tingkat keberhasilan 100%, yang berarti semua *input* menghasilkan *output* sesuai fungsinya. Pengujian FSM membuktikan bahwa musuh dapat bertransisi antar state (*Patrol*, *Chase*, *Attack*, dan *Death*) secara dinamis tanpa konflik logika. Pengujian *Procedural Generation* menunjukkan peta permainan selalu berbeda setiap sesi, dengan ruangan saling terhubung, penempatan item dan musuh bervariasi, serta posisi antivirus tetap logis. Pada pengujian kontrol pemain, seluruh pergerakan dan mekanisme tembakan responsif serta sistem deteksi tabrakan berfungsi dengan baik. Pengujian kompatibilitas perangkat menunjukkan game dapat berjalan pada spesifikasi rendah hingga tinggi, meski pada perangkat berspesifikasi rendah terdapat penurunan performa saat kondisi intensif. Sementara itu, hasil pengujian pengguna dengan 13 responden dan 6 pernyataan menunjukkan 70,51% penilaian positif dan 29,48% penilaian negatif, yang berarti mayoritas pengguna menilai gameplay, kontrol, AI musuh, dan desain peta sudah baik, meskipun masih ada aspek yang perlu ditingkatkan terutama pada kualitas suara dan keseimbangan permainan. Adapun saran sebagai acuan terhadap penelitian atau pengembangan selanjutnya antara lain adalah menambahkan jumlah *level* dan memperkaya variasi misi guna meningkatkan tantangan serta daya tarik permainan, mengimplementasikan sistem *save* dan *load* antar *level* agar pemain bisa terus bermain tanpa perlu mulai dari awal, serta meningkatkan kualitas animasi dan efek suara untuk menghasilkan suasana permainan yang lebih mendalam dan menarik.

DAFTAR PUSTAKA

- [1] Setyaningrum, A. A., (2024). “ Penerapan Metode *Finite State Machine* Pada Game Adventure “Noir” ” JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 8 No. 2, April 2024.
- [2] Zahra, A. I. N., 2024. Penerapan Metode Fsm (*Finite State Machine*) Pada Game “Tsunami Rescue” Berbasis Android. JATI (Jurnal Mahasiswa Teknik Informatika).
- [3] Karmanto, A. L. A., (2023). “Implementasi Metode *Finite State Machine* (Fsm) Pada Game “Cipher Club” 2d Berbasis Android”. JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 7 No. 1, Februari 2023.
- [4] Aulia, M. K., (2021). “Penerapan Metode *Finite State Machine* Pada Game Pandemic Nightmare Berbasis Android “. JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 5 No. 1, Maret 2021.
- [5] Ramadhan, D. A., “Procedural Content Generation Pada Game World Exploration Sandbox Menggunakan Algoritma Perlin Noise.” JINACS: Volume 04 Nomor 01, 2022 (Journal of Informatics and Computer Science) ISSN : 2686-2220.
- [6] Prasetyo, R. A., (2023). “Perancangan Game 2d “Barbossa The Savior” Menggunakan *Finite State Machine* ”, JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 7 No. 5, Oktober 2023.
- [7] Satrio, I., (2022). “Penerapan A* Pathfinding Dan Fsm (*Finite State Machine*) Pada Game “Lost Civilization” Berbasis Android”. JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 6 No. 2, September 2022.
- [8] Pratama, D. S., Vendyansyah, N., & Prasetya, R. P. (2024). PEMBUATAN GAME 2D “KEN AROK” MENGGUNAKAN METODE *FINITE STATE MACHINE* DAN PATHFINDING. JATI (Jurnal Mahasiswa Teknik Informatika), 8(5), 8389-8397.
- [9] Atthariq, M. H., 2022. “Implementasi Procedural Content Generation pada Game Menggunakan *Unity*”. Jurnal Teknik Komputer, vol. 1, no. 2, pp. 62-72, November 2022.
- [10] Syu’aibi, M. H., (2023). “Perancangan Dan Implementasi Metode Fsm (*Finite State Machine*) Pada Game Military Defence 2d Berbasis Android” JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 7 No. 5, Oktober 2023.
- [11] Muslim, M. A., (2021). Pembuatan Game 2D “Escape Plan” Dengan Metode *Finite State Machine*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X Vol. 3, No. 5, Mei 2021, hlm. 4406-4414 <http://j-ptiik.ub.ac.id>.
- [12] Setiawan, D. A., (2021). “ Penerapan Metode *Finite State Machine* Dan Fuzzy Pada Game “Black Warrior” ” JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 5 No. 2, September 2021