

Implementasi Finite State Machine dan Fisher-Yates Shuffle pada Game Platformer 2D Edukasi Berbasis Godot Engine

Muhamad Davi Ferzagie^{1*}, Reza Augusta Jannatul Firdaus²

^{1,2}Teknik Informatika, Universitas Hasyim Asy'ari, Indonesia

Email: ^{1*}daviferzagie@mhs.unhasy.ac.id, ²rezafirdaus@unhasy.ac.id

Abstrak

Pengelolaan aktivitas karakter dan penyajian konten yang bervariasi merupakan dua aspek penting dalam pengembangan *game platformer* 2D edukasi. Perubahan aktivitas karakter yang terjadi secara cepat berpotensi menimbulkan konflik logika apabila tidak dikelola secara terstruktur, sedangkan penyajian soal yang berulang dapat mengurangi tingkat tantangan dan keterlibatan pemain. Penelitian ini bertujuan mengimplementasikan *Finite State Machine* (FSM) untuk mengelola aktivitas karakter pemain dan algoritma *Fisher-Yates Shuffle* untuk mengacak urutan soal pada *game platformer* 2D edukasi *Archipelago Hollow* berbasis desktop menggunakan Godot Engine. FSM diterapkan pada delapan *state* karakter yang terdiri dari *Idle*, *Walk*, *Jump*, *Fall*, *Attack*, *Interact*, *Hurt*, dan *Dead*. Sementara itu, algoritma *Fisher-Yates Shuffle* digunakan untuk mengacak 20 soal sehingga menghasilkan kombinasi soal yang berbeda pada setiap sesi permainan. Evaluasi sistem dilakukan melalui *Black Box Testing*, validasi metode, dan *User Acceptance Testing* (UAT). Hasil pengujian menunjukkan bahwa seluruh fungsi sistem berjalan dengan tingkat keberhasilan 100%. Validasi FSM menghasilkan transisi *state* yang valid dengan waktu respons di bawah 16 ms. Algoritma *Fisher-Yates Shuffle* mampu menghasilkan distribusi soal yang lebih merata dibandingkan metode *random* biasa dengan waktu eksekusi kurang dari 0,01 detik. Selain itu, hasil UAT yang melibatkan 61 responden memperoleh nilai rata-rata 4,47 pada kategori sangat memuaskan. Hasil penelitian menunjukkan bahwa integrasi FSM dan *Fisher-Yates Shuffle* mampu meningkatkan konsistensi perilaku karakter sekaligus menghasilkan variasi soal yang lebih dinamis dalam permainan.

Kata Kunci: *Finite State Machine; Fisher-Yates Shuffle; Game Edukasi; Godot Engine; Platformer 2D*

Abstract

Character activity management and content variation are important aspects in the development of educational 2D platformer games. Rapid changes in character activities may lead to logical conflicts if not managed properly, while repetitive question sequences can reduce player engagement and challenge levels. This study aims to implement a Finite State Machine (FSM) for managing player character activities and the Fisher-Yates Shuffle algorithm for randomizing quiz questions in the educational 2D platformer game *Archipelago Hollow* developed using Godot Engine. FSM was applied to eight character states, namely *Idle*, *Walk*, *Jump*, *Fall*, *Attack*, *Interact*, *Hurt*, and *Dead*. Meanwhile, the Fisher-Yates Shuffle algorithm was used to randomize a set of 20 questions to generate different question combinations in each gameplay session. System evaluation was conducted through Black Box Testing, method validation, and User Acceptance Testing (UAT). The results indicate that all system functionalities operated successfully with a 100% success rate. FSM validation demonstrated valid state transitions with response times below 16 ms. The Fisher-Yates Shuffle algorithm produced a more balanced question distribution than conventional randomization methods while maintaining an execution time of less than 0.01 seconds. Furthermore, the UAT involving 61 respondents achieved an average score of 4.47, categorized as highly satisfactory. These findings demonstrate that the integration of FSM and Fisher-Yates Shuffle effectively improves character behavior consistency while providing more dynamic question variations during gameplay.

Keywords: *Finite State Machine; Fisher-Yates Shuffle; Educational Game; Godot Engine; 2D Platformer*

*Corresponding Author:

Muhamad Davi Ferzagie, Universitas Hasyim Asy'ari, Indonesia

Email: daviferzagie@mhs.unhasy.ac.id

1. PENDAHULUAN

Perkembangan industri *game* telah menjadikan *game* tidak hanya sebagai media hiburan, tetapi juga sebagai sarana edukasi yang mampu meningkatkan keterlibatan pengguna melalui interaksi yang menarik. Salah satu genre yang masih banyak diminati adalah *game platformer* 2D karena menawarkan mekanisme permainan yang sederhana, responsif, dan mudah dimainkan oleh berbagai kalangan pengguna [1]. Pada genre ini, pemain dituntut untuk mengendalikan karakter dalam melewati rintangan, berinteraksi dengan objek permainan, dan menyelesaikan tujuan tertentu pada setiap level. Salah satu faktor yang memengaruhi kualitas pengalaman bermain adalah konsistensi perilaku karakter. Karakter pemain umumnya memiliki beberapa aktivitas seperti *idle*, berjalan, melompat, menyerang, dan berinteraksi dengan lingkungan permainan. Perubahan aktivitas yang terjadi secara cepat dapat menyebabkan konflik logika maupun ketidaksesuaian animasi apabila tidak dikelola dengan baik.

Oleh karena itu, diperlukan suatu metode yang mampu mengatur transisi aktivitas karakter secara terstruktur sehingga respons permainan tetap stabil dan konsisten [2], [3]. *Finite State Machine* (FSM) merupakan salah satu metode yang banyak digunakan dalam pengembangan *game* untuk mengelola perilaku karakter melalui representasi sejumlah *state* dan aturan transisi antar-*state*. Penelitian yang dilakukan oleh Saktiyan [1] menunjukkan bahwa FSM mampu membantu pengelolaan *state* karakter secara terstruktur pada *game* 2D. Penelitian Arbansyah [3] juga membuktikan bahwa FSM efektif digunakan untuk mengatur perilaku musuh pada *game platformer* 2D sehingga menghasilkan respons yang lebih konsisten. Selain itu, Shofy [4] menyatakan bahwa implementasi FSM pada Godot Engine mampu mendukung pengelolaan aktivitas karakter secara modular dan mudah dikembangkan. Selain pengelolaan karakter, variasi konten permainan juga menjadi aspek penting dalam mempertahankan minat pemain.

Pada *game* edukasi yang memanfaatkan sistem kuis, urutan soal yang muncul secara berulang dapat mengurangi tantangan permainan dan membuat pemain lebih mudah menghafal pola soal. Untuk mengatasi permasalahan tersebut, diperlukan mekanisme pengacakan yang mampu menghasilkan distribusi soal secara merata pada setiap sesi permainan. Algoritma *Fisher-Yates Shuffle* merupakan metode pengacakan yang banyak digunakan karena mampu menghasilkan susunan data acak dengan peluang kemunculan yang relatif sama pada setiap elemen. Penelitian Rahadhani et al. [5] menunjukkan bahwa algoritma *Fisher-Yates Shuffle* berhasil meningkatkan variasi penyajian konten pada *game* edukasi berbasis Godot Engine. Selain memiliki tingkat pemerataan yang baik, algoritma ini juga memiliki kompleksitas waktu $O(n)$ sehingga sesuai diterapkan pada sistem permainan yang membutuhkan proses pengacakan secara cepat.

Berdasarkan penelitian terdahulu, implementasi FSM dan *Fisher-Yates Shuffle* umumnya dilakukan secara terpisah. Penelitian yang mengintegrasikan kedua metode tersebut dalam satu *game platformer* 2D edukasi berbasis desktop masih relatif terbatas. Padahal, kombinasi FSM dan *Fisher-Yates Shuffle* berpotensi menghasilkan sistem permainan yang lebih konsisten dari sisi perilaku karakter sekaligus lebih dinamis dalam penyajian konten edukatif [7], [9].

Oleh karena itu, penelitian ini mengembangkan *game platformer* 2D edukasi berjudul *Archipelago Hollow* menggunakan Godot Engine dengan menerapkan FSM untuk mengelola aktivitas karakter pemain dan algoritma *Fisher-Yates Shuffle* untuk mengacak urutan soal kuis. Implementasi kedua metode tersebut dievaluasi menggunakan pengujian *Black Box*, validasi metode, dan *User Acceptance Testing* (UAT) untuk mengetahui efektivitasnya dalam mendukung konsistensi perilaku karakter serta variasi penyajian soal selama permainan berlangsung [3]. Pada Tabel 1 menyajikan perbandingan beberapa penelitian terdahulu yang berkaitan dengan penerapan *Finite State Machine* (FSM) maupun algoritma *Fisher-Yates Shuffle* pada pengembangan *game* [8]. Perbandingan dilakukan untuk mengidentifikasi posisi penelitian serta kontribusi yang diberikan pada penelitian ini

Tabel 1. Perbandingan Penelitian Terdahulu

Peneliti	Metode	Objek Penelitian	Hasil Penelitian	Keterbatasan
Awalludin dkk. [6]	FSM	Game 2D Cat Collection Coin	FSM mampu mengelola state	Tidak membahas integrasi dengan

				karakter secara sistem pengacakan terstruktur dan konten permainan. mempermudah pengembangan game.
Firdaus dkk. [7]	FSM	Retro Arcade Fighting Game		FSM menghasilkan perilaku karakter yang lebih terstruktur. Tidak mengintegrasikan metode pengacakan data.
Nugroho dkk. [8]	FSM	Game 2D Omar's Adventure		FSM berhasil diterapkan untuk mengatur aktivitas karakter. Tidak membahas pengacakan konten permainan.
Harsadi dkk. [9]	Fisher-Yates Shuffle	Game edukasi Aksara Jawa		Pengacakan konten menjadi lebih merata dan dinamis. Tidak dikombinasikan dengan metode pengelolaan perilaku karakter.

Berdasarkan Tabel 1, dapat diketahui bahwa sebagian besar penelitian terdahulu berfokus pada penerapan FSM untuk mengelola perilaku karakter maupun NPC dalam *game* [8], [9], [10], [11]. Sementara itu, penelitian yang dilakukan oleh Harsadi et al. [9] memanfaatkan algoritma *Fisher-Yates Shuffle* untuk meningkatkan variasi konten permainan melalui mekanisme pengacakan data. Meskipun demikian, penelitian-penelitian tersebut masih menerapkan kedua metode secara terpisah. Berbeda dengan penelitian sebelumnya, penelitian ini mengintegrasikan FSM dan *Fisher-Yates Shuffle* dalam satu *game platformer* 2D edukasi berbasis desktop menggunakan Godot Engine [12]. FSM digunakan untuk mengelola aktivitas karakter pemain yang terdiri dari beberapa *state* utama, sedangkan *Fisher-Yates Shuffle* diterapkan untuk mengacak urutan soal kuis sehingga setiap sesi permainan memiliki variasi yang berbeda. Integrasi kedua metode tersebut diharapkan mampu menghasilkan pengalaman bermain yang lebih konsisten, responsif, dan tidak monoton [13].

Kebaruan dari penelitian ini terletak pada penggabungan Finite State Machine dengan Fisher-Yates Shuffle pada satu platformer 2D pendidikan yang dibangun di atas Godot Engine, serta penerapan kedua pendekatan tersebut untuk mengatasi dua isu berbeda secara bersamaan, yaitu konsistensi perilaku karakter dan variasi konten edukatif. Tidak sama dengan studi sebelumnya yang hanya memusatkan perhatian pada pengaturan status karakter atau pengacakan konten secara terpisah, penelitian ini mengukur efektivitas penggabungan kedua metode melalui validasi FSM, analisis distribusi hasil pengacakan, dan pengujian penerimaan pengguna (UAT). Metode ini menghasilkan sistem permainan yang lebih responsif, adaptif, dan tidak monoton.

Selain itu, penelitian ini merupakan salah satu implementasi integrasi Finite State Machine dan Fisher-Yates Shuffle pada game platformer 2D edukasi berbasis Godot Engine yang dievaluasi melalui kombinasi Black Box Testing, validasi metode, dan User Acceptance Testing (UAT). Pendekatan ini tidak hanya menilai keberhasilan implementasi sistem, tetapi juga mengukur konsistensi perilaku karakter, kualitas distribusi hasil pengacakan soal, serta tingkat penerimaan pengguna terhadap permainan yang dikembangkan.

2. METODE PENELITIAN

Penelitian ini berfokus pada pengembangan *game platformer* 2D edukasi berjudul *Archipellago Hollow* berbasis desktop menggunakan Godot Engine. Sistem dikembangkan dengan mengintegrasikan metode *Finite State Machine* (FSM) untuk mengelola aktivitas karakter pemain dan algoritma *Fisher-Yates Shuffle* untuk mengacak urutan soal kuis yang muncul selama permainan. Tahapan penelitian meliputi analisis kebutuhan, perancangan sistem, implementasi metode, serta pengujian sistem.

2.1. Analisis Kebutuhan Sistem

Penelitian dilakukan dengan mengidentifikasi kebutuhan sistem berdasarkan karakteristik *game platformer* 2D. Permasalahan utama yang ditemukan adalah perlunya pengelolaan aktivitas karakter yang konsisten ketika menerima berbagai masukan dari pemain serta kebutuhan akan variasi soal kuis agar permainan tidak mudah diprediksi. Untuk mengatasi permasalahan tersebut, FSM digunakan sebagai mekanisme pengelolaan perilaku karakter, sedangkan algoritma *Fisher-Yates Shuffle* digunakan untuk menghasilkan susunan soal yang berbeda pada setiap sesi permainan. Sistem dikembangkan menggunakan Godot Engine 4.5 dengan bahasa pemrograman GDScript dan dijalankan pada sistem operasi Windows. FSM digunakan untuk mengatur aktivitas karakter pemain berdasarkan kondisi tertentu. Setiap aktivitas direpresentasikan dalam bentuk *state* yang saling terhubung melalui aturan transisi [11], [1], [3].

Pada penelitian ini, karakter pemain memiliki delapan *state*, yaitu *Idle*, *Walk*, *Jump*, *Fall*, *Attack*, *Interact*, *Hurt*, dan *Dead*. Perubahan *state* terjadi berdasarkan masukan dari pemain maupun kondisi yang terjadi selama permainan berlangsung. Sebagai contoh, karakter akan berpindah dari *Idle* ke *Walk* ketika menerima input pergerakan, sedangkan *state Jump* akan aktif ketika pemain menekan tombol lompat saat karakter berada di permukaan tanah. Dengan pendekatan ini, setiap aktivitas karakter dapat dikelola secara lebih terstruktur sehingga mengurangi kemungkinan terjadinya konflik logika maupun ketidaksesuaian animasi [14], [15].

2.2. Implementasi Fisher-Yates Shuffle

Algoritma *Fisher-Yates Shuffle* diterapkan pada sistem kuis untuk menghasilkan urutan soal yang berbeda pada setiap sesi permainan. Mekanisme pengacakan dilakukan terhadap seluruh kumpulan soal yang tersedia sebelum permainan dimulai. Proses pengacakan dimulai dari elemen terakhir pada larik soal, kemudian memilih indeks secara acak untuk dilakukan pertukaran posisi (*swap*). Langkah tersebut diulang hingga seluruh elemen selesai diproses. Setelah proses pengacakan selesai, sistem mengambil sejumlah soal dari hasil pengacakan untuk ditampilkan kepada pemain. Metode ini dipilih karena mampu menghasilkan distribusi soal yang lebih merata dibandingkan metode pemilihan acak sederhana [16], [17].

2.3. Perancangan Sistem

Pemain mengendalikan karakter utama untuk menjelajahi lingkungan permainan, menghindari rintangan, berinteraksi dengan objek informasi, dan menyelesaikan kuis edukatif. Setiap kuis yang berhasil diselesaikan akan menjadi syarat untuk melanjutkan permainan ke tahap berikutnya. FSM berperan dalam mengatur seluruh aktivitas karakter selama permainan berlangsung, sedangkan *Fisher-Yates Shuffle* bertanggung jawab terhadap pengacakan soal yang muncul pada setiap sesi permainan. Integrasi kedua metode tersebut membentuk sistem yang mampu memberikan pengalaman bermain yang lebih responsif sekaligus lebih dinamis [1], [3], [6].

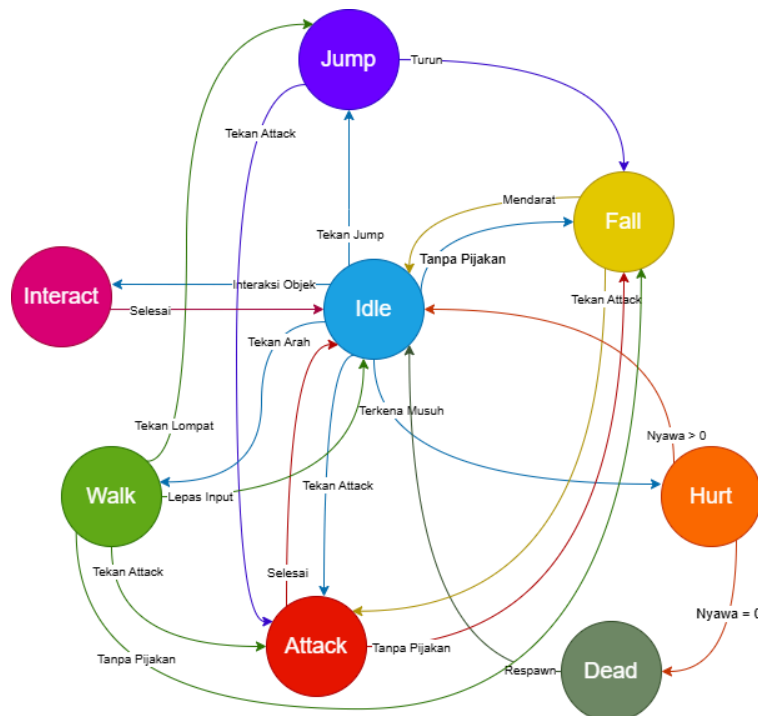
2.4. Pengujian Sistem

Evaluasi sistem dilakukan melalui tiga jenis pengujian. Pengujian pertama menggunakan metode *Black Box Testing* untuk memastikan seluruh fungsi sistem berjalan sesuai dengan rancangan. Pengujian ini mencakup menu navigasi, kontrol karakter, mekanisme kuis, dan kondisi akhir permainan. Pengujian kedua dilakukan untuk memvalidasi implementasi metode yang digunakan [4]. Pada FSM, pengujian difokuskan pada keakuratan transisi antar-*state* dan konsistensi perilaku karakter. Sementara itu, pada algoritma *Fisher-Yates Shuffle*, pengujian dilakukan dengan menganalisis distribusi kemunculan soal yang dihasilkan dari beberapa kali percobaan [17]. Pengujian ketiga menggunakan *User Acceptance Testing* (UAT) yang melibatkan pengguna sebagai responden. Evaluasi dilakukan melalui kuesioner berbasis skala Likert untuk mengukur persepsi pengguna terhadap responsivitas kontrol, kenyamanan bermain, variasi soal, tampilan antarmuka, serta tingkat kepuasan terhadap sistem secara keseluruhan [1], [5], [9].

3. HASIL DAN PEMBAHASAN

Metode *Finite State Machine* (FSM) diterapkan untuk mengelola aktivitas karakter pemain selama permainan berlangsung. Implementasi FSM dilakukan dengan membagi perilaku karakter ke dalam

delapan *state*, yaitu *Idle*, *Walk*, *Jump*, *Fall*, *Attack*, *Interact*, *Hurt*, dan *Dead*. Setiap *state* memiliki kondisi transisi yang berbeda sesuai dengan masukan dari pemain maupun kondisi yang terjadi di dalam permainan [15]. Penerapan FSM memungkinkan setiap aktivitas karakter berjalan secara terstruktur tanpa menimbulkan konflik logika antar-*state*. Sebagai contoh, karakter tidak dapat menjalankan animasi menyerang dan melompat secara bersamaan apabila kondisi transisi belum terpenuhi [12]. Dengan demikian, perilaku karakter menjadi lebih konsisten dan responsif terhadap setiap input yang diberikan pemain[4]. Pada Gambar 1 menunjukkan rancangan FSM yang digunakan pada sistem permainan

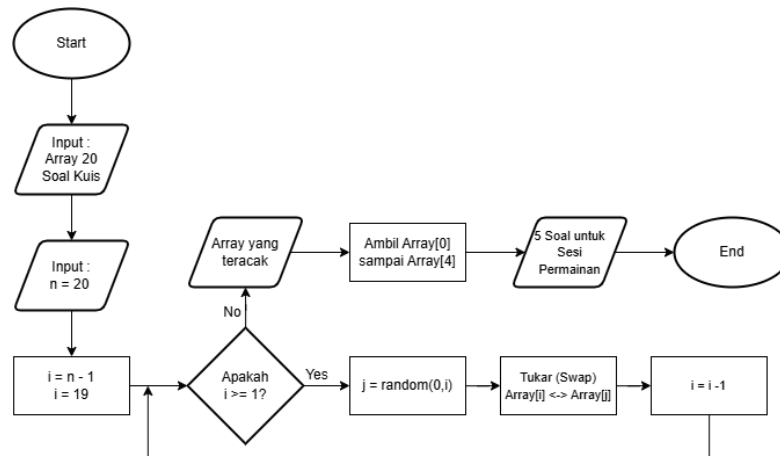


Gambar 1. Diagram *Finite State Machine* Karakter Pemain

Berdasarkan hasil implementasi, seluruh *state* dapat berjalan sesuai dengan rancangan dan mampu menghasilkan transisi yang stabil selama proses permainan berlangsung [4], [16].

3.1. Implementasi Fisher-Yates Shuffle

Penerapan algoritma ini menghasilkan kombinasi soal yang berbeda pada setiap sesi permainan sehingga pemain tidak memperoleh urutan soal yang sama secara berulang[18]. Selain meningkatkan variasi permainan, metode ini juga membantu menjaga tingkat tantangan yang diberikan kepada pemain [15]. Pada Gambar 2 memperlihatkan alur pengacakan soal menggunakan algoritma *Fisher-Yates Shuffle*.



Gambar 2. Proses Pengacakan Soal Menggunakan Fisher-Yates Shuffle

Hasil implementasi menunjukkan bahwa algoritma dapat berjalan dengan baik dan menghasilkan urutan soal yang berbeda pada setiap percobaan.

3.2. Hasil Black Box Testing

Pengujian *Black Box* dilakukan untuk memastikan seluruh fungsi sistem berjalan sesuai dengan kebutuhan yang telah ditetapkan. Pengujian meliputi menu navigasi, kontrol karakter, mekanisme kuis, serta kondisi akhir permainan. Pada Tabel 2. Hasil Black Box Testing

Tabel 2. Hasil Black Box Testing	
Komponen yang Diuji	Hasil
Menu Navigasi	Berhasil
Kontrol Karakter	Berhasil
Gameplay	Berhasil
Sistem Kuis	Berhasil
Kondisi Akhir Permainan	Berhasil

Berdasarkan hasil pengujian, seluruh fungsi sistem dapat berjalan sesuai dengan rancangan tanpa ditemukan kesalahan fungsional. Tingkat keberhasilan pengujian mencapai 100%, yang menunjukkan bahwa sistem telah memenuhi kebutuhan operasional permainan.

3.3. Hasil Validasi Finite State Machine

Validasi FSM dilakukan untuk memastikan seluruh transisi antar-state berjalan sesuai dengan kondisi yang telah ditentukan. Pengujian dilakukan dengan memberikan berbagai kombinasi input dan mengamati perubahan state yang terjadi selama permainan berlangsung. Pada Tabel 3. Hasil Validasi FSM

Tabel 3. Skenario Pengujian FSM				
No	State Awal	Input	State Tujuan	Status
1	Idle	Tombol Gerak	Walk	Valid
2	Walk	Tombol Lompat	Jump	Valid
3	Jump	Tidak Menyentuh Tanah	Fall	Valid
4	Fall	Menyentuh Tanah	Idle	Valid
5	Idle	Tombol Attack	Attack	Valid
6	Attack	Animasi Selesai	Idle	Valid
7	Idle	Interaksi Objek	Interact	Valid
8	Interact	Interaksi Selesai	Idle	Valid
9	Semua State	Menerima Damage	Hurt	Valid
10	Hurt	Nyawa Habis	Dead	Valid

Pengujian FSM dilaksanakan dengan meninjau setiap jalur transisi yang ada pada skema keadaan karakter. Setiap transisi diuji dengan 10 percobaan menggunakan variasi input yang beragam untuk memastikan tidak ada konflik antar-keadaan. Hasil dari pengujian menunjukkan semua transisi beroperasi sesuai dengan ketentuan yang telah ditentukan tanpa adanya keadaan yang tidak teridentifikasi.

Tabel 4. Hasil Validasi FSM

Aspek Pengujian	Hasil
Validitas Transisi State	100%
Konflik Antar State	Tidak Ditemukan
Respons Sistem	< 16 ms

Hasil pengujian menunjukkan bahwa seluruh transisi *state* berjalan sesuai rancangan dengan tingkat keberhasilan 100%. Selain itu, tidak ditemukan konflik antar-*state* selama pengujian berlangsung. Waktu respons yang diperoleh juga berada di bawah 16 ms sehingga karakter mampu merespons input pemain secara cepat.

3.4. Hasil Pengujian Fisher-Yates Shuffle

Evaluasi terhadap algoritma *Fisher-Yates Shuffle* dilakukan dengan membandingkan distribusi kemunculan soal yang dihasilkan dengan metode *random* biasa. Pengujian dilakukan melalui beberapa kali percobaan untuk melihat tingkat pemerataan distribusi soal. Hasil pengujian menunjukkan bahwa algoritma *Fisher-Yates Shuffle* menghasilkan distribusi soal yang lebih merata dibandingkan metode *random* biasa. Selain itu, waktu eksekusi yang dibutuhkan berada di bawah 0,01 detik sehingga proses pengacakan tidak memengaruhi performa permainan.

Tabel 5. Perbandingan Hasil Pengacakan

Metode	Waktu Eksekusi	Pemerataan Distribusi
Random Biasa	< 0,01 detik	Kurang Merata
Fisher-Yates Shuffle	< 0,01 detik	Lebih Merata

Berdasarkan hasil tersebut, algoritma *Fisher-Yates Shuffle* terbukti mampu menghasilkan variasi soal yang lebih baik dengan tetap mempertahankan efisiensi proses pengacakan.

Tabel 6. Distribusi Kemunculan Soal pada 100 Percobaan

Nomor Soal	Frekuensi
1	5
2	4
3	6
4	5
5	5
6	4
7	6
8	5
9	5
10	4
11	5
12	6
13	5
14	4
15	5
16	6
17	5

18	5
19	4
20	5

Pengujian dilakukan sebanyak 100 kali proses pengacakan terhadap 20 soal. Hasil pengujian menunjukkan bahwa rata-rata frekuensi kemunculan setiap soal adalah 5 kali dengan standar deviasi sebesar 0,74. Nilai standar deviasi yang relatif kecil menunjukkan bahwa distribusi kemunculan soal cenderung merata. Temuan ini mengindikasikan bahwa algoritma Fisher-Yates Shuffle mampu menghasilkan pola pengacakan yang mendekati distribusi uniform sehingga dapat mengurangi kemunculan pola soal yang berulang.

3.5. Hasil User Acceptance Testing

Pengujian *User Acceptance Testing* (UAT) dilakukan untuk mengetahui tingkat penerimaan pengguna terhadap sistem yang dikembangkan. Pengujian melibatkan 61 responden yang memainkan game kemudian memberikan penilaian melalui kuesioner berbasis skala Likert. Aspek yang dinilai meliputi responsivitas kontrol karakter, konsistensi perilaku karakter, variasi soal, tampilan antarmuka, serta kenyamanan bermain secara keseluruhan.

Tabel 7. Karakteristik Responden

Karakteristik	Jumlah	Persentase
Laki-laki	38	62,3%
Perempuan	23	37,7%
Total	61	100%

Tabel 8. Berdasarkan Usia

Rentang Usia	Jumlah
17-20 Tahun	18
21-25 Tahun	34
>25 Tahun	9

Tabel 9. Berdasarkan Pengalaman Bermain

Pengalaman	Jumlah
Pemula	15
Menengah	31
Mahir	15

Partisipan dipilih melalui metode purposive sampling dengan syarat mempunyai pengalaman dalam penggunaan komputer dan telah bermain game platformer. Berbagai karakteristik partisipan dimanfaatkan untuk mendapatkan penilaian yang lebih akurat mengenai kualitas sistem yang tengah dikembangkan.

Tabel 10. Hasil User Acceptance Testing

Aspek Penilaian	Nilai Rata-Rata
Responsivitas Kontrol	4,49
Konsistensi Karakter	4,51
Variasi Soal	4,42
Antarmuka Permainan	4,43
Kepuasan Pengguna	4,50
Rata-Rata Keseluruhan	4,47

Hasil UAT menunjukkan bahwa sistem memperoleh nilai rata-rata keseluruhan sebesar 4,47 yang termasuk dalam kategori sangat memuaskan. Nilai tersebut menunjukkan bahwa pengguna memberikan tanggapan positif terhadap implementasi FSM maupun algoritma *Fisher-Yates Shuffle* yang diterapkan pada permainan.

3.6. Pembahasan

Hasil penelitian menunjukkan bahwa integrasi FSM dan *Fisher-Yates Shuffle* mampu mendukung pengembangan *game platformer* 2D yang lebih responsif dan dinamis. FSM berhasil mengelola aktivitas karakter secara konsisten melalui delapan *state* yang saling terhubung tanpa menimbulkan konflik logika. Temuan ini sejalan dengan penelitian Nugroho et al. [13], Filza et al. [12], serta Nugraha et al. [6] yang menyatakan bahwa FSM efektif digunakan untuk mengelola perilaku karakter dalam permainan. Di sisi lain, algoritma *Fisher-Yates Shuffle* mampu menghasilkan distribusi soal yang lebih merata dibandingkan metode *random* biasa. Hasil tersebut mendukung penelitian Fanani et al. [14] yang menunjukkan bahwa algoritma ini efektif digunakan untuk meningkatkan variasi konten pada *game* edukasi.

Perbedaan utama penelitian ini dibandingkan penelitian sebelumnya terletak pada integrasi kedua metode dalam satu sistem permainan. FSM digunakan untuk menjaga konsistensi perilaku karakter, sedangkan *Fisher-Yates Shuffle* digunakan untuk meningkatkan variasi soal kuis. Kombinasi kedua metode tersebut terbukti mampu meningkatkan kualitas pengalaman bermain sekaligus mempertahankan unsur edukatif yang menjadi tujuan utama permainan.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan *Finite State Machine* (FSM) dan algoritma *Fisher-Yates Shuffle* pada *game platformer* 2D edukasi *Archipelago Hollow* berbasis desktop menggunakan Godot Engine. FSM digunakan untuk mengelola aktivitas karakter pemain melalui delapan *state* utama sehingga setiap transisi aktivitas dapat berjalan secara konsisten dan responsif. Sementara itu, algoritma *Fisher-Yates Shuffle* diterapkan untuk mengacak urutan soal kuis sehingga menghasilkan variasi soal yang berbeda pada setiap sesi permainan. Hasil pengujian menunjukkan bahwa seluruh fungsi sistem dapat berjalan dengan baik dengan tingkat keberhasilan *Black Box Testing* sebesar 100%. Validasi metode menunjukkan bahwa FSM mampu menghasilkan transisi *state* yang valid tanpa konflik logika dengan waktu respons di bawah 16 ms. Selain itu, algoritma *Fisher-Yates Shuffle* terbukti mampu menghasilkan distribusi soal yang lebih merata dibandingkan metode *random* biasa dengan waktu eksekusi kurang dari 0,01 detik. Hasil *User Acceptance Testing* yang melibatkan 61 responden memperoleh nilai rata-rata 4,47 yang termasuk dalam kategori sangat memuaskan.

Berdasarkan hasil tersebut, integrasi FSM dan *Fisher-Yates Shuffle* terbukti efektif dalam meningkatkan konsistensi perilaku karakter serta memberikan variasi soal yang lebih dinamis pada *game platformer* 2D edukasi. Penelitian selanjutnya dapat mengembangkan sistem dengan menambahkan kecerdasan buatan pada musuh, memperluas jumlah dan jenis konten edukasi, serta melakukan pengujian pada berbagai platform selain desktop untuk memperoleh cakupan pengguna yang lebih luas.

5. SARAN

Meskipun implementasi *Finite State Machine* dan *Fisher-Yates Shuffle* telah menunjukkan hasil yang baik dalam mendukung konsistensi perilaku karakter dan variasi soal pada permainan, masih terdapat beberapa aspek yang dapat dikembangkan pada penelitian berikutnya. Pengembangan selanjutnya dapat dilakukan dengan menambahkan sistem kecerdasan buatan (*Artificial Intelligence*) pada musuh menggunakan metode seperti *Behavior Tree* atau *Goal-Oriented Action Planning* untuk menghasilkan perilaku yang lebih adaptif. Selain itu, jumlah dan variasi soal edukasi dapat diperluas sehingga materi yang disajikan menjadi lebih beragam dan mampu meningkatkan nilai edukatif permainan.

Penelitian berikutnya juga dapat melakukan pengujian pada platform lain, seperti Android maupun Web, agar jangkauan pengguna menjadi lebih luas. Dari sisi evaluasi metode, analisis statistik yang lebih mendalam terhadap hasil pengacakan, seperti uji *Chi-Square* atau *Kolmogorov-Smirnov*, dapat dilakukan untuk memperkuat validitas distribusi soal yang dihasilkan oleh algoritma *Fisher-Yates Shuffle*. Selain itu, pengukuran performa sistem yang mencakup penggunaan memori, konsumsi

sumber daya, dan stabilitas *frame rate* dapat menjadi aspek penting untuk menilai kualitas permainan secara lebih komprehensif.

REFERENCES

- [1] M. A. N. Saktiyan, "Pengembangan Game Maze Edukasi Membaca Berbasis Godot Engine untuk Anak Usia Dini," in *Prosiding Seminar Nasional Teknologi dan Sains*, 2026.
- [2] D. Prayoga, L. Rusdiana, and F. Yedithia, "Pengembangan Game 2D Platformer Virus Must Die Berbasis Android Menggunakan Unity," *Jurnal Saintekom*, vol. 12, no. 2, 2022.
- [3] Arbansyah, "Implementasi Finite State Machine (FSM) pada Agent Permainan Game Lost Animal at Borneo Berbasis Android," *CoSciTech*, vol. 3, no. 2, 2022, doi: 10.37859/coscitech.v3i2.3921.
- [4] M. N. Shofy, B. Rahman, and S. Andryana, "Implementation of Fisher-Yates Shuffle Algorithm in Android-Based Javanese Batik Culture Education Game," 2024, doi: 10.33480/jitk.v10i1.5256.
- [5] A. W. Rahmadhani and M. F. Rizal, "Implementasi Algoritma Fisher-Yates Shuffle Untuk Pengacakan Puzzle Pada Game Edukasi Berbasis Android," *Development Engineering and Research in Technology*, vol. 1, no. 2, 2025.
- [6] C. Nugraha and A. I. Purnamasari, "Implementation of Finite State Machine on NPCs to Improve Game Productivity," *Journal of Artificial Intelligence and Engineering Applications*, vol. 4, no. 3, 2025.
- [7] R. Adrian and M. Akbar, "Implementasi Logika Finite State Machine dalam Perilaku Enemy Boss dalam Game 2D," *Jurnal Sains Informatika Terapan*, vol. 4, no. 3, 2025.
- [8] K. Awalludin and S. Busono, "Pengembangan Game 2D Cat Collection Coin Berbasis Single Player Menggunakan Metode Finite State Machine untuk Manajemen State Karakter," *Jurnal Algoritma*, vol. 22, no. 2, 2025, doi: 10.33364/algoritma/v.22-2.2424.
- [9] M. B. Firdaus, A. Z. Waksito, and A. Tejawati, "Finite State Machine for Retro Arcade Fighting Game Development," *International Journal of Informatics and Communication Technology*, vol. 14, no. 1, 2025, doi: 10.11591/ijict.v14i1.pp102-110.
- [10] A. Nugroho, B. R. Putra, and R. Setiawan, "Development of Platformer Game 'Adventure Drake' Using Finite State Machine Method," *Media Journal of General Computer Science*, vol. 1, pp. 29-32, 2024, doi: 10.62205/mjgcs.v1i1.10.
- [11] C. Y. Wardhana, P. Harsadi, and W. L. Y. Saptomo, "Implementasi Algoritma Fisher-Yates Shuffle Pada Game Edukasi Aksara Jawa Menggunakan Godot Engine," *Jurnal TIKomSiN*, vol. 10, no. 1, 2022, doi: 10.30646/tikomsin.v10i1.603.
- [12] M. F. Filza and A. Z. Rahman, "Implementation of a Finite State Machine for Character Animation Transitions in Unity-Based 3D Games," *SMART : Jurnal Teknologi Informasi dan Komputer*, vol. 4, no. 2, 2025.
- [13] F. Nugroho and P. M. N. S. A. Basid, "2D Game Omar's Adventure Design Using the Finite State Machine Method," *JITE Journal of Informatics and Telecommunication Engineering*, vol. 6, no. 1, 2022, doi: 10.31289/jite.v6i1.6327.
- [14] M. A. Fanani and G. S. Permadi, "Implementasi Algoritma Fisher-Yates Shuffle pada Game Edukasi Pengenalan Buah dan Sayur Berbasis Android," *Journal of Artificial Intelligence and Technology Information*, vol. 4, no. 2, 2026, doi: 10.58602/jaiti.v4i2.252.
- [15] Y. F. Rosman, R. A. Krisdiawan, and Yulyanto, "Implementasi Algoritma Fisher Yates Shuffle pada Game Adventure Edukasi Jenis Hewan Berdasarkan Makanan," *Digital Transformation Technology (Digitech)*, vol. 5, no. 2, 2025.
- [16] Y. Haryanto, "Designing Educational Games With Fisher Yates Shuffle Method In Early Childhood at TKIT Al Muqorrobin Based On Android," *Instal : Jurnal Komputer*, vol. 15, no. 02, 2023.
- [17] A. E. Adeniyi, B. Brahma, and M. O. Adebiyi, "Development of Two Dimension (2D) Game Engine with Finite State Machine (FSM) Based Artificial Intelligence (AI) Subsystem," *Procedia Comput. Sci.*, 2024, doi: 10.1016/j.procs.2024.04.283.
- [18] S. Alfarisi, "Implementasi AI Adaptif dalam Game Action Platformer 2D Keeping Up The Journey," 2024.