

PENINGKATAN SERIOUS GAME THE ARCHITECT: REBIRTH OF SERAYA MENGUNAKAN METODE ENGAGEMENT-ORIENTED DYNAMIC DIFFICULTY ADJUSTMENT (EDDA)

Tenno Young Dalana, Paulus Harsadi

Informatika, Universitas Tiga Serangkai Surakarta
Jl. K.H. Samanhudi No. 84-86 57142 Laweyan Jawa Tengah
tennoyoungdalana123@gmail.com

ABSTRAK

Tantangan utama dalam *serious game* simulasi tata kota adalah mempertahankan keterlibatan pemain akibat tingkat kesulitan statis yang kaku. Hal ini memicu kebosanan pada pemain ahli atau kecemasan pada pemain pemula. Penelitian ini bertujuan merancang dan mengimplementasikan perangkat lunak *serious game* PC berjudul "The Architect: Rebirth of Seraya" sebagai media edukasi kebijakan publik makro. Pengembangan dilakukan menggunakan *game engine* Godot 4.x dengan bahasa pemrograman C#, dipandu metodologi *Game Development Life Cycle* (GDLC). Untuk menjaga kondisi psikologis pemain agar tetap berada dalam zona *Flow State*, sistem mengintegrasikan metode *Engagement-Oriented Dynamic Difficulty Adjustment* (EDDA), *Finite State Machine* (FSM), dan algoritma *Random Forest*. EDDA secara *real-time* mengevaluasi performa dan mengadaptasi variabel krisis ekonomi secara dinamis. Pengujian efektivitas diukur menggunakan *Game Engagement Questionnaire* (GEQ). Hasil pengujian menunjukkan bahwa sistem berhasil mempertahankan keterlibatan pemain dengan skor rata-rata "Baik" (4,087). Kesimpulannya, penerapan EDDA terbukti efektif mengoptimalkan keterlibatan pemain dan memberikan pengalaman simulasi kebijakan publik yang adaptif dan realistis.

Kata kunci : *Serious game, PC desktop, Godot, EDDA, Random forest, FSM.*

1. PENDAHULUAN

Ekosistem permainan video interaktif dewasa ini telah berevolusi jauh melampaui fungsi asali sebagai sarana hiburan empiris semata. Medium ini kini diakui sebagai instrumen komputasional tingkat lanjut yang mampu merepresentasikan sistem kerumitan dunia nyata ke dalam sebuah ruang virtual yang aman. Transformasi fungsional inilah yang melahirkan konsep *serious game*, yakni sebuah rekayasa perangkat lunak yang dirancang secara terstruktur untuk mengemban misi pendidikan, pelatihan kognitif, maupun simulasi strategis. Di ranah pendidikan tinggi dan literasi publik, pemahaman mengenai tata kelola kota (*urban governance*), alokasi anggaran, dan dampak berkelanjutan dari regulasi politik sering kali menjadi materi yang sangat abstrak dan sulit dicerna jika hanya disampaikan melalui pendekatan tekstual konvensional. Melalui intervensi *serious game*, batas-batas teoretis tersebut dapat diruntuhkan; memberikan pemain wewenang absolut untuk bereksperimen dengan berbagai kebijakan berisiko tinggi tanpa menghasilkan kerusakan empiris di dunia nyata [1].

Meskipun potensi edukatif yang ditawarkan sangat revolusioner, pengembangan *serious game* masih dibayangi oleh satu permasalahan fundamental: rendahnya tingkat retensi dan keterlibatan emosional (*player engagement*) pengguna dalam sesi jangka panjang. Kesenjangan ini utamanya bersumber dari penerapan arsitektur tingkat kesulitan yang bersifat statis (*static difficulty*). Pada sistem statis, parameter rintangan telah diprogram secara baku sejak awal kompilasi, memaksa seluruh pemain—tanpa mempedulikan tingkat literasi strategi mereka—untuk

tunduk pada satu kurva linear yang kaku [2]. Ketidakhadiran mekanisme adaptif ini sangat bertentangan dengan prinsip dasar *Flow Theory* yang digagas oleh Mihaly Csikszentmihalyi. Ketidakseimbangan relasional antara tingkat keterampilan (*skill*) yang dimiliki pengguna dengan beban tantangan (*challenge*) yang disimulasikan oleh mesin akan menghasilkan respons psikologis yang sangat destruktif. Pemain yang belum memiliki kecakapan manajerial akan secara instan terpuruk dalam kondisi kecemasan (*anxiety*) ketika kas kota simulasi bangkrut. Sebaliknya, pemain dengan kemampuan analitis tinggi akan mendominasi simulasi dengan sangat cepat, yang kemudian berujung pada kondisi kebosanan akut (*boredom*). Kedua kondisi ekstrem ini merupakan katalis utama pemicu penghentian sesi permainan secara permanen (*player churn rate*) sebelum transfer pengetahuan terkait kebijakan publik dapat terjadi [3].

Untuk mengatasi defisiensi arsitektur statis tersebut, penelitian rekayasa perangkat lunak mutakhir mengusulkan penerapan sistem kalibrasi otonom yang dikenal sebagai *Dynamic Difficulty Adjustment* (DDA). Secara lebih spesifik, penelitian ini merekayasa dan mengimplementasikan algoritma *Engagement-Oriented Dynamic Difficulty Adjustment* (EDDA). Berbeda dengan sistem DDA tradisional yang hanya memantau persentase kemenangan atau kekalahan teknis (seperti jumlah *health point*), metode EDDA beroperasi secara asinkron membedakan data telemetri yang lebih kompleks, seperti fluktuasi ekonomi kota, kecepatan resolusi krisis, dan durasi retensi pemain [4].

Berdasarkan landasan urgensi tersebut, dibangunlah sebuah perangkat lunak *serious game* bergenre simulasi tata kota dengan resolusi 2D isometrik yang diberi judul "The Architect: Rebirth of Seraya". Sistem komputasi ini dirancang secara eksklusif untuk arsitektur *PC Desktop Windows 64-bit* dengan memanfaatkan kapabilitas pemrosesan bahasa berorientasi objek C# di dalam lingkungan *game engine* Godot 4.x. Integrasi algoritma adaptif EDDA pada perangkat keras personal ini diproyeksikan mampu mendeteksi profil pemain secara *real-time* dan memodifikasi laju datangnya bencana secara transparan (*invisible hand*), sehingga pemain senantiasa terkunci di dalam *Flow channel*. Dengan demikian, tujuan utama untuk menanamkan pemahaman akan hukum kausalitas dalam ilmu manajemen kebijakan publik dapat terwujud secara holistik.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Beberapa penelitian terdahulu telah mengeksplorasi adaptasi kesulitan dan efektivitas *serious game*. Mi dan Gao [1] membuktikan bahwa metode Engagement-Oriented Dynamic Difficulty Adjustment (EDDA) efektif mencegah player churn dan mempertahankan keterlibatan pemain, meskipun implementasinya pada genre simulasi tata kelola makro belum diuji secara spesifik. Dari sisi arsitektur perilaku, Wiratama dkk. [2] menunjukkan bahwa integrasi Finite State Machine (FSM) mampu meningkatkan responsivitas interaksi secara signifikan tanpa membebani memori komputasi. Pada domain edukasi kebijakan publik, Brauner dkk. [4] dan Putra dkk. [11] menemukan bahwa *serious game* simulasi tata kelola memiliki potensi besar sebagai media pembelajaran yang interaktif, namun sering terkendala pada sistem kesulitan yang tidak responsif terhadap kemampuan pemain. Penelitian ini mengisi celah tersebut dengan mengkombinasikan EDDA, FSM, dan algoritma Random Forest di dalam engine Godot 4.x—sebuah arsitektur engine yang telah diuji efisiensinya oleh Alchimowicz dan Plechawska-Wójcik [8]. Kombinasi ini bertujuan menciptakan simulasi kebijakan publik dengan tingkat kesulitan yang terkalibrasi secara dinamis.

2.2. Serious Game dan Afordansi Genre City Building

Konstruksi *serious game* beroperasi di atas prinsip *stealth learning*, yakni sebuah metode pedagogi di mana pengguna menyerap materi pembelajaran yang sangat kompleks tanpa merasa sedang menempuh proses akademis yang kaku. Penggunaan genre City Building Management dipilih secara presisi karena genre ini memiliki afordansi (*affordances*) bawaan yang sangat identik dengan sistem birokrasi nyata. Pemain didorong untuk memahami logika berpikir sistemik (*systems thinking*), di mana sebuah entitas keputusan tunggal—seperti penetapan regulasi

pajak karbon—akan memicu efek domino terhadap puluhan variabel lain, mulai dari indeks kebahagiaan warga hingga probabilitas terjadinya krisis kesehatan massal [5].

2.3. Teori Keseimbangan Psikologis (Flow Theory)

Landasan perancangan metrik evaluasi kinerja sistem ini bertumpu sepenuhnya pada *Flow Theory*. Konsep ini memformulasikan bahwa individu akan mencapai tingkat efisiensi kognitif maksimal apabila mereka berada dalam kondisi penyerapan total, yang ditandai dengan hilangnya distorsi waktu dan gangguan eksternal. Di dalam arsitektur permainan digital, *Flow Channel* adalah sebuah koridor sempit yang membentang di antara sumbu vertikal (*Challenge*) dan sumbu horizontal (*Skill*). Tugas utama dari modul kecerdasan buatan yang dibangun dalam penelitian ini adalah untuk memastikan fluktuasi status kota senantiasa bermanuver di dalam koridor tersebut secara dinamis [6].

2.4. Algoritma Adaptif EDDA, FSM, dan Random Forest

Engagement-Oriented Dynamic Difficulty Adjustment (EDDA) beroperasi sebagai struktur otak utama yang memproses kumpulan data mentah menjadi metrik *Player Skill Score* (PSS). Keluaran dari EDDA tidak dieksekusi secara langsung, melainkan dikirimkan ke dalam arsitektur *Finite State Machine* (FSM). FSM berfungsi sebagai pengatur transisi status operasional kebijakan (*Ready, Active, Cooldown*) secara deterministik, memastikan bahwa interupsi bencana tidak saling tumpang tindih (*overlapping*) yang dapat merusak logika permainan [7]. Sebagai lapisan analitik final, modul *Random Forest* diimplementasikan menggunakan arsitektur *ensemble learning*. Algoritma ini memecah evaluasi ke dalam lima pohon keputusan independen untuk mengkalulasi kecenderungan moral dan strategis pemain. Hasil komputasi dari setiap pohon kemudian diagregasi melalui mekanisme *majority voting* untuk memberikan label gelar kepemimpinan di layar pasca-permainan secara objektif [8].

Dari tinjauan ketiga konsep tersebut, teridentifikasi sebuah kesenjangan penelitian yang menjadi justifikasi utama studi ini. Studi terdahulu mengenai *serious game* edukatif umumnya berhasil membuktikan efektivitas pedagogis genre ini, namun mengabaikan persoalan retensi pemain dalam sesi jangka panjang akibat desain kesulitan yang statis [4]. Di sisi lain, penelitian mengenai *Dynamic Difficulty Adjustment* (DDA) yang ada lebih banyak diterapkan pada game hiburan konvensional dengan metrik sesederhana persentase kesehatan karakter, bukan pada variabel makroekonomi yang kompleks seperti indeks stabilitas kota atau saldo kas [7]. Penelitian ini hadir untuk menutup kesenjangan tersebut dengan merancang EDDA sebagai evaluator berbasis data telemetri kota, FSM sebagai eksekutor deterministik

yang mencegah tumpang tindih krisis, dan Random Forest sebagai pengklasifikasi profil pemain—sebuah arsitektur terpadu yang belum pernah diimplementasikan secara bersamaan dalam konteks simulasi kebijakan publik sebelumnya.

3. METODE PENELITIAN

Metodologi yang menjadi tulang punggung rekayasa sistem dalam penelitian ini adalah *Game Development Life Cycle* (GDLC). Metodologi ini dipilih karena kemampuannya yang sangat elastis dalam mengakomodasi sinkronisasi antara penulisan kode matematis kompleks dengan integrasi aset multimedia resolusi tinggi secara iteratif [9].

3.1. Tahapan Pengembangan Sistem

Pengembangan sistem dibagi menjadi lima tahapan utama:

- inisialisasi: Penentuan genre *City Building* dan target platform PC Desktop (Windows 64-bit).
- Pra-Produksi: Perancangan *Game Design Document* (GDD), pemetaan alur FSM, dan perumusan matematis algoritma EDDA.
- Produksi: Implementasi kode menggunakan bahasa C# .NET 8 pada *engine* Godot 4.x.
- Pengujian: Validasi *Black Box Testing* terhadap seluruh skenario aksi pemain.
- Rilis: Kompilasi akhir aplikasi *standalone*.

3.2. Formulasi Matematis Algoritma EDDA

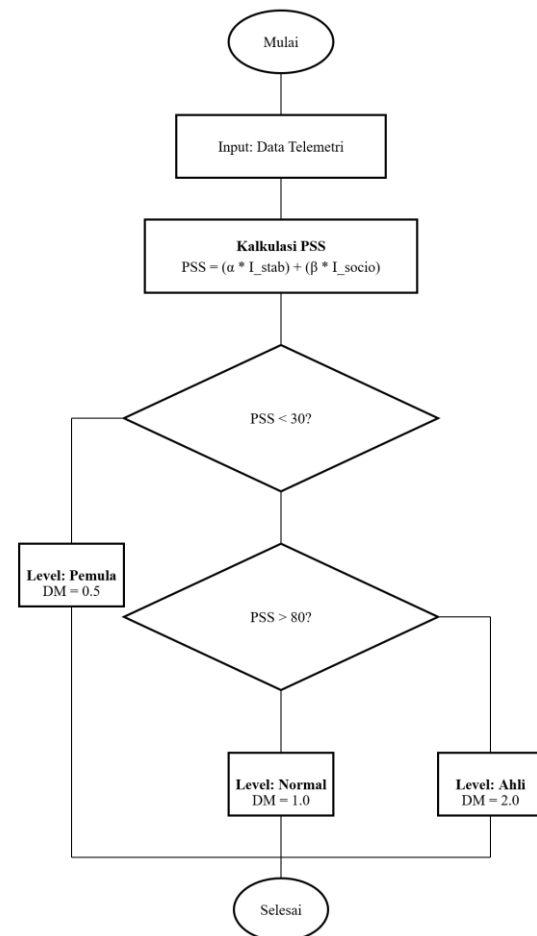
Inti dari penelitian ini adalah penerapan algoritma EDDA yang berfungsi sebagai mesin adaptasi kesulitan. Sistem mengonversi variabel telemetri kota menjadi *Player Skill Score* (PSS) secara *real-time*. Rumus perhitungan PSS adalah sebagai berikut:

$$PSS = (\alpha \times I_{stab}) + (\beta \times I_{socio})$$

Dimana I_{stab} merupakan Indeks Stabilitas kota (40% bobot) yang dihitung dari rata-rata variabel pendidikan, kesehatan, lingkungan, dan pangan. I_{socio} adalah Indeks Sosial-Ekonomi (60% bobot) yang diperoleh dari kepuasan warga, saldo kas kota, dan inversi kemiskinan. Nilai koefisien $\alpha = 0.4$ dan $\beta = 0.6$ ditetapkan untuk menyeimbangkan dampak makro ekonomi.

$$DM = \begin{cases} 0.5, & \text{jika } PSS < 30 \text{ (Level Pemula)} \\ 1.0, & \text{jika } 30 \leq PSS \leq 80 \text{ (Level Normal)} \\ 2.0, & \text{jika } PSS > 80 \text{ (Level Ahli)} \end{cases}$$

Gambar 1 merepresentasikan alur data di mana input telemetri kota diolah oleh mesin komputasi untuk menghasilkan *Difficulty Modifier*. Sistem ini memastikan bahwa jika $PSS < 30$, pemain akan menerima subsidi otomatis sebagai intervensi kognitif untuk menghindari *rage quit*.



Gambar 1. Diagram alur logika perhitungan PSS dan Difficulty Modifier

Secara arsitektural, algoritma EDDA memproses dua variabel komposit utama untuk menentukan tingkat kesulitan yang optimal. Variabel pertama adalah Indeks Stabilitas (I_{stab}), yang merepresentasikan ketahanan fundamental infrastruktur kota. Nilai ini dikalkulasi secara asinkron setiap 5 detik dengan mencari nilai rata-rata dari empat metrik vital: tingkat edukasi, kesehatan, lingkungan, dan ketersediaan pangan populasi. Formulasi matematis untuk indeks ini dinyatakan sebagai:

$$I_{stab} = (Edukasi + Kesehatan + Lingkungan + Pangan) / 4$$

Variabel kedua adalah Indeks Sosial-Ekonomi (I_{socio}), yang merepresentasikan kekuatan finansial dan stabilitas politik dari sudut pandang warga kota. Indeks ini dievaluasi melalui agregasi tingkat kepuasan faksi, inversi persentase kemiskinan (agar nilai tinggi berkonotasi positif), dan normalisasi saldo kas kota. Mengingat variabel saldo memiliki rentang nilai yang masif, sistem menerapkan teknik *Min-Max Scaling* kustom untuk memampatkan nilai kas ke dalam format persentase 0–100% sebelum dirata-ratakan. Formulasi untuk indeks ini adalah:

$$I_{socio} = (Kepuasan + (100 - Kemiskinan) + \min(Kas/300, 100)) / 3$$

Kedua nilai indeks tersebut kemudian diproses menggunakan nilai koefisien bobot α dan β untuk menghasilkan *Player Skill Score* (PSS). Keluaran PSS ini tidak langsung dieksekusi sebagai bentuk *punishment* atau *reward* kepada pemain, melainkan dikirimkan sebagai parameter pemicu ke dalam modul logika *Finite State Machine* (FSM).

FSM yang bertindak sebagai mesin eksekutor krisis dalam arsitektur ini memiliki tiga transisi status utama: Inactive, Monitoring, dan DisasterActive. Saat mode permainan adaptif dimulai, FSM akan bertransisi dari status Inactive menuju Monitoring. Pada fase ini, sistem secara aktif mengakumulasi metrik telemetri menggunakan *event timer* independen. Apabila algoritma mendeteksi dominasi pemain yang ditandai dengan nilai *Difficulty Modifier* yang ekstrem, FSM akan mengevaluasi batas toleransi (*threshold*) dari faksi-faksi virtual di dalam kota. Jika stabilitas faksi berada di bawah 20%, FSM akan memutus siklus pemantauan pasif dan memaksa transisi ke status DisasterActive. Status ini bertindak sebagai mekanisme pengaman komputasional (*safety lock*) yang secara instan menghentikan perhitungan waktu untuk bencana eksternal lainnya. Penghentian sementara ini sangat krusial dalam desain rekayasa perangkat lunak gim untuk memastikan pemain tidak dibombardir oleh berbagai skenario krisis secara serentak, yang berpotensi merusak logika *gameplay* dan memicu hilangnya keseimbangan ruang lingkup *Flow*.

3.3. Data Teknik Pengumpulan Data

Penelitian ini menggunakan *purposive sampling* untuk memastikan responden telah berinteraksi dengan dinamika sistem. Jumlah sampel minimal dihitung menggunakan rumus Lemeshow:

$$n = \frac{Z^2 \times P(1-P)}{d^2}$$

Berdasarkan rumus tersebut, dengan nilai $Z = 1.96$, $P = 0.5$, dan batas toleransi kesalahan $d = 0.15$, diperoleh nilai $n = 42.68$, yang dibulatkan menjadi 50 responden untuk memenuhi kuota buffer penelitian.

3.4. Instrumen Evaluasi (GEQ)

Instrumen yang digunakan adalah *Game Engagement Questionnaire* (GEQ) yang mengukur lima dimensi psikometrik: *Competence*, *Immersion*, *Flow*, *Tension*, dan *Challenge* [2]. Setiap butir pertanyaan menggunakan skala Likert 1-5. Data mentah diolah menggunakan uji validitas *Pearson Product Moment* dan uji reliabilitas *Cronbach's Alpha*.

Gambar 2 menampilkan *Heads-Up Display* (HUD) utama dalam simulasi tata kota. Panel 'Monitor EDDA' pada sisi kiri secara *real-time* memvisualisasikan kalkulasi *Player Skill Score* (PSS) dan *Difficulty Modifier* (DM) yang diproses di latar belakang. Panel ini menyajikan transparansi data mengenai stabilitas makroekonomi kota, yang menjadi basis bagi algoritma FSM dan *Random Forest* dalam mengeksekusi respons adaptif terhadap dinamika permainan.



Gambar 2. Antarmuka Permainan dan Panel Pemantau Algoritma EDDA

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengujian Fungsionalitas (Black Box Testing)

Pengujian fungsionalitas dilakukan menggunakan metode *Black Box Testing* yang berfokus pada pengamatan fungsionalitas *Input-Output* dari mekanika interaksi dasar permainan tanpa melihat struktur kode di baliknya. Pengujian ini bertujuan untuk memastikan sistem manajemen sumber daya dan antarmuka pemain berjalan sesuai dengan batasan aturan (*ruleset*) yang ditetapkan. Hasil seluruh skenario pengujian tercantum pada Tabel 1.

Tabel 1. Hasil Pengujian Fungsionalitas (Black Box Testing)

ID Uji	Skenario Pengujian	Hasil Aktual	Hasil yang Diharapkan	Status
BB-01	Pemain meletakkan bangunan saat saldo Kas Kota mencukupi.	Saldo berkurang, bangunan terinstansiasi dengan presisi Y-Sorting, dan SFX konstruksi berbunyi.	Sistem memotong saldo pemain, memunculkan sprite bangunan di atas TileMap, dan memainkan SFX konstruksi.	Valid
BB-02	Pemain mencoba meletakkan bangunan saat dana kurang dari harga bangunan.	Bangunan batal dibuat, muncul teks peringatan UI warna merah, SFX Error berbunyi.	Sistem menolak aksi konstruksi dan memunculkan notifikasi "Dana Tidak Cukup".	Valid
BB-03	Pemain membangun Klinik, dana cukup, namun level Edukasi kota di bawah syarat (<70%).	Konstruksi dibatalkan, panel UI memberikan umpan balik alasan kegagalan spesifik.	Sistem memblokir peletakan bangunan dan menampilkan notifikasi spesifik "Syarat Edukasi Tidak Terpenuhi".	Valid

ID Uji	Skenario Pengujian	Hasil Aktual	Hasil yang Diharapkan	Status
BB-04	Pemain menghapus pabrik yang sedang aktif mempekerjakan warga.	Bangunan terhapus, saldo bertambah setengah harga, angka kemiskinan instan melonjak.	Bangunan hilang (QueueFree), pemain mendapat dana refund 50%, pekerja dikembalikan ke status pengangguran.	Valid

4.2. Hasil Pengujian Algoritma Adaptif (EDDA, FSM, dan Random Forest)

Pengujian logika dilakukan untuk memvalidasi presisi perhitungan matematis dan respons dinamis dari algoritma EDDA, FSM, serta klasifikasi *Random*

Forest berdasarkan kondisi spesifik yang disimulasikan. Hasil pengujian menunjukkan seluruh algoritma merespons dengan benar terhadap skenario yang diuji, sebagaimana ditunjukkan pada Tabel 2.

Tabel 2. Hasil Pengujian Algoritma Adaptif (White Box Testing)

Skenario Simulasi	Parameter Terdeteksi	Respons Algoritma (FSM & EDDA)	Status
Skenario Kritis (Ancaman Game Over)	Kas mendekati 0, tingkat kemiskinan tinggi.	EDDA mengkalkulasi PSS turun hingga <30 (Level Pemula). FSM menghentikan timer bencana eksternal dan memicu Relief Event (Bantuan Subsidi Tunai +150 Koin).	Valid
Skenario Dominasi (Overpowered)	Kas surplus ekstrem, seluruh faksi 100% puas.	EDDA mendeteksi PSS melonjak >80 (Level Ahli). FSM merespons dengan memicu krisis tingkat tinggi secara paksa (Krisis Ekonomi/Inflasi) untuk menyeimbangkan tantangan.	Valid
Skenario Pembuktian Random Forest	Pemain terus menaikkan pajak tinggi dan membiarkan polusi maksimal demi industri.	Modul Random Forest membaca fitur telemetri dan menggunakan Majority Voting. Sistem mencetak laporan akhir dengan label “Kapitalis/Oligarki”.	Valid

4.3. Hasil Pengujian Pengalaman Pengguna (GEQ)

Pengujian pengalaman pengguna menggunakan instrumen *Game Experience Questionnaire* (GEQ) dengan skala Likert 1–5 kepada 50 responden yang memenuhi kriteria inklusi penelitian. Sebelum analisis, instrumen diuji validitas menggunakan Korelasi *Pearson Product Moment* (r tabel = 0,20, N = 50) dan reliabilitas menggunakan *Cronbach's Alpha*. Seluruh 26 butir instrumen dinyatakan valid (r hitung = 0,5988–0,8608) dan seluruh tujuh dimensi dinyatakan reliabel (α > 0,70). Rekapitulasi skor rata-rata per dimensi disajikan pada Tabel 3.

Tabel 3. Rekapitulasi Skor Rata-Rata Dimensi GEQ

Dimensi	Mean	Cronbach's Alpha	Interpretasi
Immersion	4,120	0,8580	Baik
Presence	4,147	0,7248	Baik
Flow	4,044	0,8799	Baik
Absorption	3,813	0,8908	Baik
Kausalitas	4,156	0,9287	Baik
Mekanik	4,180	0,8539	Baik
Evaluasi	4,120	0,9023	Baik
Rata-Rata Keseluruhan	4,087	> 0,70	Baik

4.4. Pembahasan

Berdasarkan data pengujian, integrasi algoritma EDDA, FSM, dan *Random Forest* berhasil menciptakan sistem simulasi yang adaptif. Rata-rata skor GEQ keseluruhan sebesar 4,087 (kategori “Baik”) membuktikan bahwa *serious game* “The Architect: Rebirth of Seraya” berhasil mencapai tujuan utamanya

sebagai media edukasi kebijakan publik yang imersif dan efektif.

Dimensi *Presence* memperoleh skor tertinggi (mean = 4,147), mengindikasikan pemain merasakan kehadiran yang kuat sebagai pemimpin kota dan merasakan dampak nyata dari setiap keputusan. Dimensi *Flow* (mean = 4,044) secara implisit mengonfirmasi efektivitas sistem EDDA dalam menjaga keseimbangan kesulitan; butir B7 (EDDA: keseimbangan kesulitan) memperoleh mean = 3,98 dan butir B8 (EDDA: kemampuan adaptasi saat kesulitan) = 4,12. Butir B9 (FSM: kemunculan krisis) memperoleh mean terendah di antara butir *Flow* (3,90), mengindikasikan terdapat ruang penyempurnaan pada parameter FSM terkait penentuan waktu kemunculan krisis.

Dimensi Kausalitas memperoleh mean = 4,156, menunjukkan game berhasil menjembatani pemahaman sebab-akibat kebijakan publik. Butir C3 (“Memimpin kota bukan perkara mudah”) memperoleh skor tertinggi keseluruhan (mean = 4,30; median = 5,00), mengindikasikan konsensus kuat di antara responden bahwa simulasi berhasil merepresentasikan kompleksitas tata kelola kebijakan. Dimensi Mekanik (mean = 4,180) mengonfirmasi FSM berhasil mendorong pemain berpikir lebih strategis. Dimensi Evaluasi (mean = 4,120) menunjukkan sistem *Random Forest* berhasil mengklasifikasikan gaya kepemimpinan yang dirasakan akurat oleh responden (butir C9: mean = 4,18; r hitung = 0,6914).

4.5. Komparasi Perilaku Sistem: Mode Statis vs. Mode Adaptif (EDDA)

Untuk mempertegas signifikansi kontribusi algoritma EDDA, dilakukan analisis komparatif sistematis antara dua kondisi operasional permainan: mode statis (EDDA dinonaktifkan) dan mode adaptif

(EDDA diaktifkan). Pengujian dilakukan dengan mensimulasikan skenario identik—yaitu kondisi kas kota yang mendekati titik kritis—pada kedua mode tersebut. Hasilnya menunjukkan perbedaan perilaku sistem yang sangat mendasar, sebagaimana dirangkum pada Tabel 4 berikut.

Tabel 4. Komparasi Perilaku Sistem Mode Statis vs. Mode Adaptif (EDDA)

Aspek Pengujian	Mode Statis (EDDA Dinonaktifkan)	Mode Adaptif (EDDA Diaktifkan)
Skenario Kas Kritis (Mendekati 0)	Sistem tidak merespons kondisi keuangan kota. Bencana eksternal tetap berjalan sesuai jadwal tetap. Pemain langsung menghadapi kondisi Game Over tanpa jeda atau intervensi apapun dari mesin permainan.	EDDA mendeteksi PSS <30 dan mengklasifikasikan pemain sebagai Level Pemula. FSM bertransisi ke state DisasterActive, menghentikan sementara timer bencana eksternal. Sistem memicu Relief Event berupa Bantuan Subsidi Tunai sebesar +150 Koin untuk mencegah Game Over prematur.
Skenario Dominasi (Kas Surplus Ekstrem)	Tidak ada mekanisme penyeimbang. Pemain dengan kemampuan tinggi mendominasi simulasi tanpa hambatan berarti. Tingkat tantangan stagnan hingga kondisi kemenangan terpenuhi secara mudah, memicu kebosanan akut (boredom) sesuai prediksi Flow Theory.	EDDA mendeteksi PSS >80 dan mengklasifikasikan pemain sebagai Level Ahli. Difficulty Modifier dinaikkan ke nilai 2.0x, mempercepat interval kedatangan bencana eksternal hingga separuhnya. FSM memicu Krisis Ekonomi atau Inflasi secara paksa untuk mengembalikan tekanan kompetitif dan menjaga pemain dalam Flow Channel.
Mekanisme Penyeimbang Otomatis	Tidak ada. Kurva kesulitan bersifat linier dan kaku sejak awal kompilasi. Seluruh pemain menerima beban tantangan yang identik tanpa penyesuaian.	Ada. EDDA mengevaluasi PSS setiap 5 detik dan menyesuaikan Difficulty Modifier (0.5x–2.0x) secara organik menggunakan fungsi Mathf.MoveToward. Pemain pemula menerima subsidi otomatis; pemain ahli menerima eskalasi krisis tambahan.
Dampak terhadap Player Retention	Tinggi churn rate. Pemain pemula cenderung berhenti karena frustrasi. Pemain ahli cenderung berhenti karena bosan. Transfer pengetahuan kebijakan publik gagal sebelum terjadi.	Churn rate ditekan. Pemain dari berbagai tingkat kemahiran dapat mencapai kondisi akhir simulasi. Skor GEQ dimensi Flow mencapai mean 4,044 dan dimensi Tantangan 4,180, mengkonfirmasi keberhasilan EDDA menjaga pemain di dalam Flow Channel.

Perbedaan mencolok antara kedua mode tersebut membuktikan bahwa ketiadaan sistem adaptif bukan sekadar mengurangi kenyamanan bermain, melainkan secara langsung menggagalkan misi edukatif permainan. Pada mode statis, skenario kas kritis berakhir dengan Game Over tanpa memberi pemain kesempatan untuk belajar dari kesalahannya. Sebaliknya, pada mode adaptif, sistem EDDA berperan sebagai penyangga kognitif yang memberikan waktu bagi pemain untuk memahami kausalitas kebijakan dan melakukan koreksi manajerial. Temuan ini selaras dengan argumen inti Flow Theory bahwa retensi pemain hanya dapat dipertahankan apabila tantangan yang diberikan selalu sepadan dengan kapasitas kemahiran pengguna—sebuah keseimbangan yang mustahil dicapai oleh kurva statis yang kaku.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan mengimplementasikan *serious game* “The Architect: Rebirth of Seraya” berbasis algoritma EDDA, FSM, dan *Random Forest* pada platform PC Desktop Windows 64-bit menggunakan *game engine* Godot 4.x dengan bahasa C#. Seluruh skenario pengujian *Black Box Testing* dan pengujian algoritma adaptif dinyatakan Valid. Pengujian GEQ terhadap 50 responden menghasilkan skor komposit 4,087

(kategori “Baik”) dengan seluruh nilai *Cronbach's Alpha* di atas 0,70, membuktikan bahwa komputasi asinkron EDDA berhasil mengkalibrasi kurva kesulitan secara otonom dan menekan angka *player churn*. Untuk penelitian selanjutnya disarankan penambahan parameter biometrik pada EDDA, perluasan peta kota, serta pengujian komparatif antara mode kesulitan statis dan adaptif guna memperkuat validitas empiris manfaat edukatif sistem.

DAFTAR PUSTAKA

- [1] Q. Mi and T. Gao, "Engagement-Oriented Dynamic Difficulty Adjustment," *Applied Sciences*, vol. 15, no. 10, p. 5610, 2025.
- [2] J. Wiratama, Rian, M. E. Johan, and H. Santoso, "Enhancing Gaming Engagement through the Integration of Game Design Document and Finite State Machine: A Study on Optimizing Non-playable Character Responsiveness," *Indonesian Journal of Computer Science*, vol. 12, no. 4, pp. 1626-1638, 2023.
- [3] W. Oliveira and J. Hamari, "Flow Experience in Gameful Approaches: A Systematic Literature Review, Scientometric Analysis, and Research Agenda," *International Journal of Human-Computer Interaction*, vol. 41, no. 20, pp. 13113-13139, 2025.

- [4] S. Brauner, J. Moschner, A. Brumme, and F. Hölzlberger, "More than chair circles? A qualitative systematic review of serious games related to water governance and their potential effects on policy design," *Water International*, vol. 50, no. 7, pp. 799-822, 2025.
- [5] J. Chen, M. He, J. Chen, and C. Zhang, "Exploring the role and mechanisms of environmental serious games in promoting pro-environmental decision-making a focused literature review and future research agenda," *Frontiers in Psychology*, vol. 15, p. 1455005, Sep. 2024.
- [6] E. A. Dewanata and T. B. M. Willis, "Design and Construction of the Action Role-Playing Game: Supply Odyssey Using the Game Development Life Cycle Method," *JAIIT (Journal of Advances in Information and Industrial Technology)*, vol. 7, no. 2, Nov. 2024.
- [7] F. Nugroho, P. M. N. Safitri, A. Basid, F. S. Bahtiar, and I. G. P. Asto Buditjahjanto, "Dynamic Difficulty Adjustment of Serious-Game Based on Synthetic Fog using Activity Theory Model," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, 2023.
- [8] S. Alchimowicz and M. Plechawska-Wójcik, "Comparative analysis of the implementation performance using selected scripting languages in the Godot game engine," *Journal of Computer Sciences Institute (JCSI)*, vol. 31, pp. 68-72, Oct. 2023.
- [9] S. R. Fauzy, Asriyanik, and F. F. Azzahra, "Implementasi Game Development Life Cycle dalam Pembuatan Game Buana Ruh," *INDEXIA: Informatic and Computational Intelligent Journal*, vol. 5, no. 1, pp. 19-34, May 2023.
- [10] L. Cañete Sanz, S. Martens, and T. de la Hera, "The Case of Cities: Skylines Versions—Affordances in Urban Planning Education," *Media and Communication*, vol. 13, Art. no. 8747, Feb. 2025.
- [11] E. A. E. Winardy, H. Surasa, and B. Zaman, "PENGEMBANGAN ANOTHER WORLD REBORN DENGAN MENGGUNAKAN METODE GAME DEVELOPMENT LIFE CYCLE," *KHARISMA Tech*, vol. 20, no. 2, pp. 47-58, Oct. 2025.
- [12] G. H. Putra, E. W. Nugroho, and T. B. Chandrawati, "Simulation Game-Based Governance System Education," *Journal of Business and Technology*, vol. 3, no. 1, pp. 21-26, 2023.
- [13] E. A. Romero-Mendez, P. C. Santana-Mancilla, M. Garcia-Ruiz, O. A. Montesinos-López, and L. E. Anido-Rifón, "The Use of Deep Learning to Improve Player Engagement in a Video Game through a Dynamic Difficulty Adjustment Based on Skills Classification," *Applied Sciences*, vol. 13, no. 14, p. 8249, Jul. 2023.