

Pengembangan Game Edukasi Role-Playing Game (RPG) Kimia Menggunakan Godot Engine Berbasis Finite State Machine

^{1*}Muhammad Ghufon, ²Muh Aris Saputra, ³Risa Helilintar

^{1,2,3} Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: ¹muhghufon005@gmail.com, ²aris.saputra@unpkdr.ac.id, ³risa.helilintar@unpkediri.ac.id

Penulis Korespondens : Muhammad Ghufon

Abstrak— Pembelajaran kimia yang abstrak membutuhkan media interaktif. Game edukasi *Role-Playing Game* (RPG) menjadi solusi, namun pemrosesan kecerdasan buatan (AI) musuh sering membebani kinerja memori komputer. Penelitian ini bertujuan mengembangkan game edukasi kimia 2D menggunakan Godot Engine dengan menerapkan algoritma *Finite State Machine* (FSM) untuk mengontrol perilaku AI musuh serta mekanik peracikan laboratorium. Pengujian sistem dilakukan melalui metode *Black-Box*. Hasil pengujian menunjukkan bahwa logika peracikan unsur berjalan akurat sesuai resep reaksi kimia. Selain itu, transisi AI musuh merespons kehadiran pemain secara optimal tanpa perulangan kalkulasi jarak yang memberatkan komputasi. Integrasi algoritma FSM dan mekanik eksperimen laboratorium ini berhasil menghasilkan purwarupa game edukasi yang stabil guna meningkatkan minat belajar siswa terhadap ilmu sains.

Kata Kunci— Algoritma FSM, Edukasi Kimia, Game RPG, Godot Engine, Kecerdasan Buatan

Abstract— Abstract chemistry learning requires interactive media. Role-Playing Game (RPG) educational games offer a solution, but processing enemy artificial intelligence (AI) often burdens computer memory performance. This study aims to develop a 2D chemistry educational game using the Godot Engine by applying the Finite State Machine (FSM) algorithm to control enemy AI behavior and laboratory mixing mechanics. System testing was conducted using the Black-Box method. The test results show that the element mixing logic runs accurately according to chemical reaction recipes. Furthermore, enemy AI transitions respond to the player optimally without computationally heavy distance calculation loops. The integration of the FSM algorithm and laboratory experimental mechanics successfully produces a stable, ducational game prototype to increase student learning interest in science.

Keywords— Artificial Intelligence, Chemistry Education, FSM Algorithm, Godot Engine, RPG Game

This is an open access article under the CC BY-SA License.



I. PENDAHULUAN

Perkembangan teknologi informasi dalam sektor pendidikan sekarang telah menjadi suatu kebutuhan untuk mendukung pembelajaran yang lebih responsif [1]. Namun, kemajuan teknologi ini ternyata belum sepenuhnya sejalan dengan peningkatan minat belajar siswa, terutama dalam mata pelajaran sains [2]. Ini sesuai dengan data *Programme for International Student Assessment* (PISA) 2022 yang menunjukkan tren penurunan literasi sains yang mengkhawatirkan, di mana

Indonesia hanya memperoleh skor 383, masih di bawah standar rata-rata global OECD yang sebesar 485 [3].

Sebagai solusi alternatif untuk mengatasi hal tersebut penggunaan media digital interaktif dapat dimanfaatkan. Metode yang dianggap efektif untuk mendorong keterlibatan kognitif siswa adalah dengan menggunakan game edukasi bertipe *Role-Playing Game* (RPG) [4]. Dengan mekanik RPG, pemain tidak hanya diberikan materi teori, tetapi juga dilibatkan untuk secara langsung mensimulasikan proses eksperimen, seperti mencampurkan unsur-unsur dasar menjadi senyawa aplikatif di dalam sebuah meja laboratorium virtual.

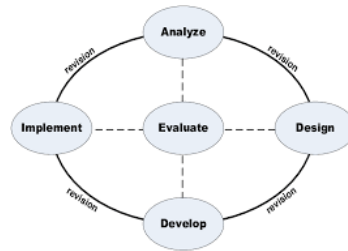
Tentu saja, tantangan teknis dalam menciptakan game edukasi semacam ini adalah mempertahankan pengalaman bermain yang imersif melalui kehadiran *Non-Player Character* (NPC) atau lawan yang responsif [5]. Sayangnya, isu performa sering terjadi pada game 2D ketika kecerdasan buatan (AI) musuh menerapkan metode pengecekan jarak (*distance checking*) secara berkelanjutan dalam proses frame [6]. Pelaksanaan logika yang berlangsung puluhan kali dalam satu detik ini membutuhkan banyak memori, sehingga berisiko menyebabkan *frame drop* atau *lag*, terutama pada perangkat dengan spesifikasi rendah.

Menghadapi masalah tersebut, dapat menerapkan sistem yang lebih terorganisir dengan memanfaatkan algoritma *Finite State Machine* (FSM) [7]. Algoritma FSM mengelompokkan perilaku musuh ke dalam sejumlah kondisi yang terbatas, sehingga pergerakannya menjadi lebih ringan dan mudah diprediksi. Dalam *Godot Engine*, kinerja FSM ini ditingkatkan dengan memanfaatkan sistem Signal dan pengenalan kolisi, memungkinkan AI hanya menghitung perubahan gerak saat ada pemicu secara langsung, bukan melakukan perhitungan secara terus-menerus [8].

Saat ini, pengembangan game edukasi yang menggabungkan mekanika eksperimen kimia secara spesifik dengan sistem AI musuh berbasis FSM masih jarang ditemukan [9]. Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengembangkan purwarupa *game* edukasi kimia RPG 2D menggunakan *Godot Engine*. Kontribusi utama penelitian ini berfokus pada implementasi algoritma FSM untuk mengendalikan transisi perilaku musuh (*Idle*, *Chasing*, dan *Dead*) yang dipicu oleh sensor jarak visual secara *real-time*. Melalui arsitektur sistem yang efisien ini, diharapkan dapat dihasilkan media pembelajaran interaktif yang stabil secara komputasi sekaligus menantang, guna merangsang keaktifan mandiri siswa dalam mengeksplorasi ilmu sains.

II. METODE

Prosedur yang tepat diperlukan agar proses penelitian menghasilkan produk media pembelajaran yang berkualitas. Proses pengembangan perangkat lunak pada penelitian ini merujuk pada kerangka kerja *Research and Development* (R&D) melalui model ADDIE (*Analysis, Design, Development, Implementation, Evaluation*)[10]. Pemilihan model ini didasarkan pada strukturnya yang sistematis dalam merancang produk interaktif yang terukur. Tahapan pengembangannya dijabarkan sebagai berikut:



Gambar 1. Model pengembangan ADDIE

A. Tahapan penelitian awal (*Analysis*)

Pada tahap ini, dilakukan analisis kebutuhan sistem dan perumusan masalah komputasi.:

1. Analisis Perangkat: Penentuan spesifikasi pengembangan *game* 2D menggunakan lingkungan integrasi *Godot Engine* versi 4 dan bahasa pemrograman *GDScript*.
2. Analisis Algoritma: Mengkaji batasan memori (*CPU load*) akibat perulangan pengecekan jarak (*distance checking*) pada musuh, dan merumuskan penerapan algoritma *Finite State Machine* (FSM) sebagai solusi optimasinya.

B. Tahapan Fase desain (*Design*)

Fase desain terdiri dari perancangan arsitektur logika dan alur mekanik:

1. Pemodelan FSM Musuh: Merancang graf transisi perilaku Kecerdasan Buatan (AI) musuh yang dibagi menjadi tiga *state* utama, yaitu *Idle* (diam), *Chasing* (mengejar), dan *Dead* (mati). Sistem sensor pandangan musuh dimodelkan menggunakan bentuk fisik *Area2D* dan *CircleShape2D* sebagai parameter jarak visual.
2. Desain Mekanik Edukasi: Merancang alur logika laboratorium kimia di mana pemain diwajibkan menyusun material dasar menjadi senyawa eksperimental (seperti energi panas/eksotermik) untuk dapat menyerang musuh.

C. Tahapan pengembangan *Game* (*Development*)

Pengembangan dilakukan dalam lingkungan IDE *Godot Engine* untuk menerjemahkan desain logika menjadi produk fungsional:

1. Implementasi Skrip FSM: Menuliskan kode *GDScript* dengan tidak menggunakan perulangan *_process(delta)* untuk AI musuh, melainkan memanfaatkan sistem *Signal* bawaan *engine* (seperti *body_entered* dan *body_exited*). Hal ini bertujuan agar transisi *state* hanya tereksekusi saat terjadi tabrakan poligon karakter utama dengan sensor musuh.
2. Penyusunan Lingkungan: Membangun *sprite* 2D, integrasi modul *inventori*, dan pemetaan *tilemap* arena eksplorasi pemain.

D. Tahapan pengujian dan validasi (*Implementation & Evaluation*)

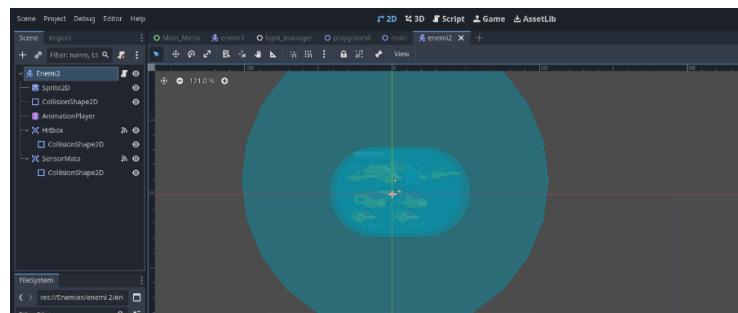
Pengujian purwarupa (*prototype*) dilakukan untuk memvalidasi algoritma komputasi dan fungsionalitas sistem. Pengujian difokuskan menggunakan metode *Black-Box Testing* terhadap alur transisi FSM musuh (validasi perubahan *state* dari *Idle* ke *Chasing* secara *real-time*) serta akurasi kondisi logika (*if-else*) pada sistem peracikan unsur kimia di meja laboratorium. Evaluasi dilakukan dengan memastikan tidak terjadinya galat kode (*bug*) maupun kebocoran memori selama pergantian *scene* permainan.

III. HASIL DAN PEMBAHASAN

A. Hasil Implementasi

a) Konfigurasi Node dan FSM Musuh

Untuk menghindari beban memori yang berlebihan akibat pengecekan jarak (*distance checking*) di setiap *frame* melalui fungsi *_process* (delta), sistem ini memanfaatkan fitur *event-driven* bawaan *engine*, yaitu *Signal*. Deteksi jarak (*proximity detection*) dimodelkan secara visual menggunakan *node Area2D* yang diberi nama *SensorMata*, dipadukan dengan *CircleShape2D* untuk membatasi radius pandangan musuh.



Gambar 2. Konfigurasi Node dan Radius Sensor FSM pada Godot Engine

Sinyal *body_entered* dan *body_exited* pada *node* tersebut bertindak sebagai *trigger* utama yang memicu transisi antar *state*. Ketika karakter utama (pemain) menembus batas *poligon CircleShape2D*, sistem langsung mengubah variabel logika pertarungan menjadi *True*, yang secara instan mengalihkan *state* musuh dari *Idle* menjadi *Chasing*.

Tabel 1. Parameter Transisi *Finite State Machine* (FSM)

State Saat Ini	Event / Kondisi (Trigger Sistem)	Aksi yang Dieksekusi	State Berikutnya
Idle	Pemain memicu sinyal <i>body_entered</i> pada <i>SensorMata</i> .	Mengunci koordinat pemain, memanggil <i>BgmManager</i> untuk mengubah musik latar menjadi tegang.	Chasing
Idle	Tidak ada objek dengan fungsionalitas <i>Player</i> di area sensor.	Menjalankan animasi rotasi atau patroli konstan.	Idle
Chasing	Pemain menjauh dan memicu sinyal <i>body_exited</i>	Mereset koordinat target, memanggil <i>BgmManager</i> untuk mengembalikan musik normal.	Idle
Chasing	Musuh menerima <i>damage</i> dari proyektil kimia yang tepat, sehingga HP < 0.	Menghentikan pergerakan, memutar animasi kehancuran.	Dead
Dead	Animasi kehancuran selesai dieksekusi.	Menjalankan perintah <i>queue_free()</i> untuk menghapus <i>node</i> dari RAM.	Exit

b) Antarmuka Laboratorium Kimia

Data yang dihasilkan dan disajikan dalam sistem ini tidak lepas dari elemen edukasi sains yang menjadi inti permainan. Untuk dapat mencapai *state Dead* pada musuh, pemain diwajibkan untuk menguasai materi IPAS (Ilmu Pengetahuan Alam dan Sosial). Pemain tidak dibekali senjata fisik biasa, melainkan harus melakukan tahapan eksplorasi untuk mengumpulkan unsur-unsur material mentah (seperti Litium, Tembaga, dan Seng). Misalnya, penggabungan elemen tertentu dapat menghasilkan senyawa proyektil aktif seperti "Baterai Listrik" atau "Ledakan Termal".



Gambar 3. Antarmuka laboratorium kimia dalam game

c) Transisi AI Mengejar (*Chasing*)

Implementasi ini menunjukkan bagaimana AI merespons pemain secara *real-time*. Saat karakter utama menembus batas poligon visual musuh, sinyal *body_entered* pada *Godot Engine* akan terpicu. Secara instan, hal ini mengalihkan *state* musuh dari yang sebelumnya *Idle* (diam/berpatroli) menjadi *Chasing* (mengejar target) dan diikuti dengan perubahan musik latar.



Gambar 4. Transisi State Chasing Saat Karakter Pemain Memasuki Radius Sensor

d) Mekanik Eksekusi Musuh (*Dead*)

Logika kimia pada laboratorium dikaitkan langsung dengan FSM musuh ini. Musuh tipe tertentu memiliki resistansi khusus, misalnya "Robot Besi" hanya bisa dikalahkan (*state Dead*) jika diserang menggunakan senyawa kelistrikan (*Sel Volta*), sedangkan "Monster Es" rentan terhadap reaksi eksotermik (Ledakan Termal). Jika berhasil mengenai musuh dengan material yang tepat, musuh akan hancur dan memorinya dihapus dari sistem menggunakan perintah `queue_free()`.



Gambar 5. Implementasi Mekanik Proyektil Pintar dan Eksekusi State Dead

B. Hasil Pengujian Fungsional Sistem (*Black-Box Testing*)

Pengujian fungsionalitas algoritma dilakukan menggunakan metode *Black-Box Testing* untuk memvalidasi apakah transisi pada mesin *game* beroperasi sesuai dengan desain FSM yang telah dimodelkan.

Tabel 2. Hasil Pengujian *Black-Box* Algoritma FSM dan Mekanik Laboratorium

Skenario Pengujian	Kondisi Awal	Input / Event	Hasil yang Diharapkan	Status
Transisi Deteksi AI	<i>State Idle.</i>	Pemain menggerakkan karakter masuk ke dalam radius SensorMata musuh..	Status berubah menjadi <i>Chasing</i> , musuh bergerak secara dinamis mengejar vektor pemain, musik berubah otomatis.	Berasil
Logika Pelepasan Target	<i>State Chasing</i>	Pemain melakukan <i>dash</i> (menghindar) keluar dari radius SensorMata.	Status kembali ke <i>Idle</i> , musuh berhenti mengejar, musik kembali santai	Berasil
Validasi Resep Kimia Gagal	Laboratorium Terbuka	Pemain memasukkan kombinasi unsur yang tidak ada di dalam skrip daftar_resep	Fungsi peleburan dibatalkan (<i>return</i>), unsur dikembalikan ke inventori, notifikasi "Resep Gagal" muncul.	Berasil
Kalkulasi Serangan Efektif	<i>State Chasing</i>	Pemain melempar hasil racikan senyawa yang tepat ke arah musuh (HP < 0).	Sinyal kehancuran terpicu, musuh memasuki <i>state Dead</i> , memori dikosongkan (<i>queue_free()</i>)	Berasil

C. Pembahasan Analisis Performa Komputasi

Pembuktian efisiensi algoritma ini divalidasi secara empiris menggunakan fitur Debugger Monitors pada *Godot Engine*. Hasil pemantauan metrik komputasi (Gambar 6) menunjukkan bahwa permainan mampu berjalan stabil pada frame rate maksimal yaitu 60 FPS tanpa adanya indikasi penurunan performa (*frame drop*). Tingkat eksekusi logika (*Process*) tercatat sangat

efisien di kisaran 17,58 milidetik dengan grafik yang berjalan stabil. Selain itu, metrik Static Memory menunjukkan garis mendatar yang sempurna pada alokasi 89,29 MiB. Kestabilan grafis memori ini mengonfirmasi bahwa penggantian metode kalkulasi jarak menjadi arsitektur FSM berbasis *event-driven* terbukti sukses mempertahankan performa perangkat lunak yang ringan serta memastikan sistem terbebas dari kebocoran memori (*memory leak*).



Gambar 6. Hasil validasi algoritma

IV. KESIMPULAN

Penelitian ini telah berhasil menyelesaikan rancang game edukasi kimia RPG 2D menggunakan *Godot Engine*. Penerapan algoritma *Finite State Machine* (FSM) dan sistem *Signal* terbukti secara efektif memecahkan kendala beban komputasi AI musuh, sehingga menghasilkan performa sistem yang ringan, stabil, dan terbebas dari galat memori. Selain pencapaian efisiensi teknis tersebut, logika kecerdasan buatan ini juga telah diintegrasikan sepenuhnya dengan mekanika eksperimen interaktif di laboratorium virtual. Keberhasilan perancangan purwarupa yang fungsional ini menjadi fondasi sistem yang solid dan siap untuk diekspansi lebih lanjut ke dalam skala permainan yang lebih luas guna meningkatkan minat belajar sains siswa.

DAFTAR PUSTAKA

- [1] A. Yani dan N. Rahmadani, "DAMPAK TEKNOLOGI INFORMASI DI ERA SOCIETY 5.0 DI BIDANG PENDIDIKAN," *MUMTAZ - Education Management and Islamic Studies*, vol. 4, no. 1, hlm. 35–39, Jan 2024, doi: 10.70936/mumtaz.v4i1.161.
- [2] E. Eveline dan S. Manzoor, "Pengukuran Sikap Siswa Sekolah Menengah di Remote Area terhadap Pembelajaran Sains," *Research in Science and Mathematics Education*, vol. 2, no. 1, hlm. 17–25, Apr 2025, doi: 10.62385/riseme.v2i1.180.
- [3] R. Umami, M. Rusdi, dan K. Kamid, "Pengembangan instrumen tes untuk mengukur higher order thinking skills (HOTS) berorientasi programme for international student assessment (PISA) pada peserta didik," *JP3M (Jurnal Penelitian Pendidikan dan Pengajaran Matematika)*, vol. 7, no. 1, hlm. 57–68, Apr 2021, doi: 10.37058/jp3m.v7i1.2069.
- [4] A. Harianto dan E. Yenti, "Desain dan uji coba game edukasi kimia berbasis role playing game (RPG) pada materi laju reaksi," *Jedchem (Journal Education and Chemistry)*, vol. 3, no. 1, hlm. 4–10, 2021.

- [5] S. Attaqwa, E. Nur Fadila, D. Fektor Orissa, O. Clara Gultom, dan P. Wirya Atmaja, “Memperkenalkan Unsur Kimia Dasar Melalui Pengembangan Game Edukasi 2D ‘Realm Of Shadows: School Of Secrets’ Dengan Metode Agile,” *Jurnal Ilmiah Teknologi Informasi dan Robotika*, vol. 6, no. 1, hlm. 1–13, Jun 2024, doi: 10.33005/jifti.v6i1.92.
- [6] D. Satria Pamenang, D. Wahyu Widodo, dan R. Aswi Ramadhani, “PENERAPAN ALGORITMA FINITE STATE MACHINE DALAM SISTEM KECERDASAN MUSUH PADA GAME AKSI TIGA DIMENSI BERBASIS UNITY,” 2025. doi: <https://doi.org/10.36040/jati.v9i5>.
- [7] R. Latifah, I. Azis, dan R. D. Risanty, “GAME EDUKASI SENYAWA IONIK SEBAGAI ALTERNATIF MEDIA PEMBELAJARAN MENGGUNAKAN FINITE STATE MACHINE,” *Komputasi: Jurnal Ilmiah Ilmu Komputer dan Matematika*, vol. 18, no. 2, hlm. 69–76, Jul 2021, doi: 10.33751/komputasi.v18i2.3629.
- [8] A. Kartadinata dan M. Akbar, “IMPLEMENTASI SISTEM CHARACTER PLAYER PADA GAME RPG 2D MENGGUNAKAN GAME ENGINE GODOT,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3, Jul 2025, doi: 10.23960/jitet.v13i3.7163.
- [9] K. Awalludin, S. Busono, U. Indahyanti, dan Y. Rahmawati, “Pengembangan Game 2D ‘Cat Collection Coin’ Berbasis Single Player Menggunakan Metode Finite State Machine untuk Manajemen State Karakter,” *Jurnal Algoritma*, vol. 22, no. 2, hlm. 140–151, Nov 2025, doi: 10.33364/algoritma/v.22-2.2424.
- [10] Agus Rustamana, Khansa Hasna Sahl, Delia Ardianti, dan Ahmad Hisyam Syauqi Solihin, “Penelitian dan Pengembangan (Research & Development) dalam Pendidikan,” *Jurnal Bima : Pusat Publikasi Ilmu Pendidikan bahasa dan Sastra*, vol. 2, no. 3, hlm. 60–69, Jun 2024, doi: 10.61132/bima.v2i3.1014.