

Penerapan *State Pattern* dan *Retrieval-based Chatbot* pada NPC Game Visual Novel di Unity Engine

Dzikri Maulana^{*1}, Muhammad Nurbayu Dila², Meissy Irania Putri³

Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang
Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361, Indonesia
Email: ¹dzikrimaulana1781945@gmail.com, ²nurbayudila93@gmail.com,
³meissyiraniaputri@gmail.com

Abstract. *Integrating the State Pattern and Retrieval-Based Chatbot for NPCs in Unity Visual Novel Games.* This study investigates the integration of the State Design pattern with retrieval-based chatbots in Non-Playable Characters (NPCs) to improve player interaction within the Unity Game Engine. NPCs are modeled with three emotional states: Happy, Normal, and Angry, each associated with different dialogue datasets. The chatbot employs the cosine similarity algorithm to match player inputs with the most relevant responses according to the current emotional state. The research follows a waterfall methodology, including implementation, emotional state transition testing, and chatbot evaluation. Results indicate a response classification accuracy of 95.65% across 230 test inputs, with a chatbot emotional adaptability rate of 45%. This study demonstrates that integrating emotional states with retrieval-based chatbots enhances the realism and contextual depth of NPC interactions, although limitations remain in semantic understanding and response diversity.

Keywords: NPC, State Design pattern, chatbot, cosine similarity, digital games

Abstrak. Penelitian ini membahas integrasi antara State Design pattern dan chatbot berbasis retrieval dalam Non-Playable Character (NPC) untuk meningkatkan interaksi pemain dalam Unity Game Engine. NPC dirancang memiliki tiga kondisi emosional utama: Happy, Normal, dan Angry, yang masing-masing memengaruhi dataset percakapan yang digunakan. Chatbot menggunakan algoritma cosine similarity untuk mencocokkan masukan pemain dengan respons yang paling relevan dari dataset berdasarkan emosi aktif. Metodologi penelitian yang digunakan adalah model waterfall, dengan tahapan implementasi, pengujian transisi state, dan evaluasi respons chatbot. Hasil menunjukkan akurasi pemilihan respons sebesar 95,65% dari 230 data uji, serta tingkat adaptabilitas chatbot sebesar 45% terhadap konteks emosi. Penelitian ini membuktikan bahwa integrasi state emosi dengan chatbot berbasis retrieval mampu menciptakan interaksi NPC yang lebih realistis dan kontekstual, meskipun masih terdapat keterbatasan dalam pemahaman semantik dan keberagaman respons.

Kata Kunci: NPC, State Design pattern, chatbot, cosine similarity, gim digital

1. Pendahuluan

Interaksi antara pemain dengan *Non-Playable Character* (NPC) dalam permainan digital merupakan aspek fundamental yang menentukan kualitas pengalaman bermain serta tingkat imersi pemain dalam dunia virtual. Menurut Z. Jiang [1], NPC yang mampu menampilkan emosi autentik seperti kebahagiaan atau kesedihan dapat menciptakan hubungan emosional yang lebih dalam dengan pemain, sehingga secara keseluruhan meningkatkan kualitas pengalaman bermain. Sebaliknya, kebanyakan NPC saat ini menggunakan skrip interaksi yang telah ditentukan, menyebabkan interaksi menjadi mudah diprediksi dan kurang menarik [2]. Model perilaku sederhana yang umumnya digunakan dalam desain NPC saat ini juga membatasi kemampuan NPC untuk merespons dinamika perilaku pemain yang kompleks secara adaptif [3].

Selain aspek perilaku, interaksi verbal atau dialogis antara pemain dengan NPC merupakan elemen penting untuk menciptakan realisme dan dinamika dalam permainan. Penggunaan *chatbot* berbasis *retrieval* yang memilih respons dari kumpulan respons yang telah disiapkan memastikan konsistensi dan reliabilitas jawaban yang diberikan oleh NPC [4]. Salah satu metode umum dalam *chatbot* berbasis *retrieval* adalah penggunaan algoritma pengukuran

kesamaan teks, khususnya *cosine similarity*. *Cosine similarity* mengukur sudut antara dua vektor kalimat dalam model ruang vektor, yang efisien terutama untuk data jarang karena hanya mempertimbangkan dimensi dengan nilai bukan nol [5].

Penerapan kecerdasan buatan pada NPC memungkinkan karakter tersebut menampilkan perilaku dan respons yang lebih hidup, berkontribusi pada lingkungan permainan yang lebih imersif. Hal ini dicapai dengan memberikan kemampuan kepada NPC untuk memproses dan merespons masukan pemain secara natural seperti percakapan manusia, yang meningkatkan realisme interaksi dalam dunia virtual [6]. Meskipun demikian, penelitian yang secara eksplisit mengintegrasikan pendekatan emosional NPC dengan *chatbot* berbasis *retrieval* menggunakan *cosine similarity* dalam suatu sistem terpadu masih relatif jarang dilakukan. Oleh karena itu, penelitian ini bertujuan untuk mengisi celah tersebut dengan mengembangkan sistem yang menggabungkan *state design pattern* dan *chatbot* berbasis *retrieval* yang menggunakan algoritma *cosine similarity* sebagai teknik *Natural Language Processing* (NLP) sederhana di dalam Unity Game Engine.

Pada sistem yang dikembangkan, NPC dirancang memiliki tiga kondisi emosional utama, yaitu *Happy*, *Normal*, dan *Angry*. Setiap kondisi emosional ini menghasilkan perilaku unik dan *dataset* percakapan tersendiri. Dengan demikian, perubahan kondisi emosional NPC secara langsung mempengaruhi pemilihan *dataset* yang digunakan *chatbot* dalam memberikan respons kepada pemain. Melalui pendekatan ini, NPC diharapkan mampu menghadirkan interaksi yang lebih realistis, kontekstual, dan adaptif terhadap perilaku pemain.

2. Tinjauan Pustaka

2.1 *State Design pattern* pada Perilaku NPC

State Design pattern merupakan pendekatan yang banyak digunakan dalam pengembangan perilaku *Non-Playable Character* (NPC) dalam game. Pola ini memungkinkan NPC mengubah perilaku secara dinamis sesuai kondisi internalnya, sehingga meningkatkan interaksi dan realisme permainan. Umumnya, pola ini diimplementasikan melalui *Finite State Machine* (FSM), yang mengatur transisi antar *state* berdasarkan kondisi atau peristiwa tertentu [7].

2.2 *Chatbot Retrieval-based* dalam Sistem Interaktif

Chatbot berbasis *retrieval* merupakan komponen penting dalam sistem interaktif yang mengandalkan pemilihan respons dari *dataset* yang telah tersedia. Sistem ini bekerja dengan mencocokkan masukan pengguna terhadap kandidat respons, lalu memilih jawaban yang paling relevan berdasarkan konteks percakapan. Pendekatan ini dinilai efisien untuk menjaga konsistensi, respons cepat, dan kestabilan sistem percakapan, khususnya dalam domain tugas terarah seperti game atau layanan pelanggan [4].

2.3 Algoritma *Cosine similarity* dalam Pemrosesan Teks

Cosine similarity adalah algoritma populer dalam pemrosesan teks yang digunakan untuk mengukur tingkat kesamaan antara dua dokumen berbasis teks. Algoritma ini menghitung nilai kosinus dari sudut antara dua vektor dalam ruang berdimensi banyak, yang mewakili teks dalam bentuk numerik. Keunggulan utamanya adalah fokus pada orientasi vektor, bukan magnitudonya, sehingga tahan terhadap perbedaan panjang teks [8].

Cosine similarity telah banyak digunakan dalam klasifikasi teks [9], klastering data besar, serta ekstraksi kalimat relevan untuk rangkuman otomatis [10]. Namun, metode ini juga memiliki keterbatasan seperti sensitivitas terhadap fitur bernilai tinggi dan tidak mempertimbangkan kesamaan semantik secara mendalam.

Nilai kesamaan dihitung berdasarkan sudut kosinus antara dua vektor teks, yang dirumuskan pada Persamaan 1, di mana A dan B adalah vektor representasi dari masukan pemain dan setiap respons dalam *dataset chatbot*.

$$\text{cosine_similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}} \quad (1)$$

2.4 Penelitian Terkait dan Analisis Kesenjangan

Penelitian mengenai penerapan *chatbot retrieval-based* dalam NPC telah menunjukkan potensi besar dalam meningkatkan interaksi pemain melalui respons percakapan yang lebih realistis dan kontekstual. Beberapa penelitian terdahulu mengembangkan sistem *chatbot* adaptif dan proaktif yang mampu memberikan petunjuk atau pertanyaan klarifikasi sesuai dinamika permainan, meningkatkan keterlibatan pemain secara signifikan [11] [12]. Integrasi *chatbot* ke dalam sistem robotik dan lingkungan VR juga memperkuat pengalaman imersif melalui sinkronisasi suara dan tindakan NPC dengan masukan pemain [13] [14].

Secara terpisah, *State Design pattern* telah digunakan secara luas dalam pengembangan perilaku NPC, termasuk pergerakan karakter, sistem pelatihan, dan interaksi dialog. Pola ini mengorganisasi perilaku berdasarkan *state* yang terenkapsulasi, memungkinkan transisi yang jelas dan modular antar kondisi. Pada penelitian terdahulu yang dilakukan oleh Sintaro, dkk. menunjukkan penerapan dalam manajemen pergerakan karakter [15], sementara Nugraha, dkk. menerapkan pada pengelolaan perilaku NPC di RPG Maker MZ untuk meningkatkan responsivitas dan efisiensi interaksi [16]. Penerapan *Finite State Machine* (FSM) secara umum meningkatkan keteraturan logika permainan dan menjaga kestabilan performa.

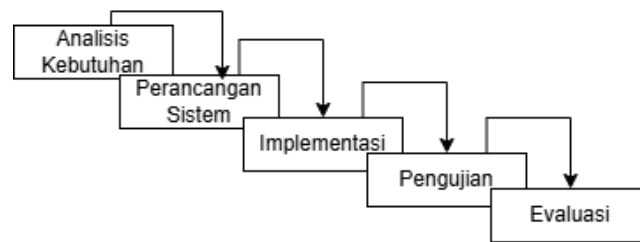
Namun, hingga saat ini, masih sedikit penelitian yang secara eksplisit menggabungkan pengelolaan emosi melalui *State Pattern* dengan pemrosesan bahasa alami dalam satu sistem NPC. Penelitian ini mengisi celah tersebut dengan mengintegrasikan *State Design pattern* untuk merepresentasikan tiga kondisi emosional NPC (*Happy*, *Normal*, *Angry*) dan menggunakan *chatbot retrieval-based* berbasis *cosine similarity* untuk menghasilkan respons percakapan yang sesuai dengan emosi aktif. Pendekatan ini bertujuan menciptakan NPC yang tidak hanya adaptif secara perilaku, tetapi juga responsif secara verbal, sehingga meningkatkan realisme dan kedalaman interaksi dalam game berbasis Unity. Untuk memperjelas posisi penelitian ini dibanding penelitian terdahulu, Tabel 1 menyajikan perbandingan beberapa literatur yang relevan. Perbandingan ini mencakup metode yang digunakan, kontribusi utama, dan keterbatasannya.

Tabel 1. Ringkasan Studi Literatur Terkait Penelitian

Peneliti	Metode	Kontribusi	Hasil	Keterbatasan
Z. Jiang (2024) [1]	Simulasi Emosi NPC	NPC dengan emosi autentik meningkatkan pengalaman bermain	Peningkatan <i>immersion</i> pemain melalui interaksi emosional NPC	Tidak mengintegrasikan chatbot
Sintaro, dkk. (2023) [15]	State Pattern pada NPC	Transisi state NPC lebih terstruktur dan modular	Peningkatan performa game Pocong Jump, efisiensi kode, dan gameplay lebih stabil	Tidak fokus pada percakapan
Wu, dkk. (2020) [13]	VR-Enabled Chatbot dengan Retrieval-based	Integrasi chatbot dalam VR untuk konsultasi teknis	Akurasi 86% dalam menjawab FAQ, meningkatkan pengalaman interaktif dalam VR	Fokus pada domain teknik (transformer), bukan aspek emosional
Nugraha, dkk. (2025) [16]	FSM pada NPC (RPG Maker MZ), uji kinerja & fungsional	Kerangka FSM untuk dialog & perilaku NPC lebih responsif	Transisi <100 ms, delay <50 ms, CPU <30%, memori 50–60 MB, interaksi stabil	Terbatas di RPG Maker MZ, Sampel pengguna terbatas (hanya melibatkan 10 responden)

3. Metodologi Penelitian

Penelitian ini menggunakan metode Waterfall, yaitu pendekatan sistematis yang terdiri atas tahapan-tahapan yang dilakukan secara berurutan dari tahap awal hingga akhir. Metode ini dipilih karena memberikan struktur yang jelas untuk pengembangan sistem integrasi antara *State Design pattern* dan *retrieval-based chatbot* pada *Non-Playable Character* (NPC) dalam Unity Game Engine. Adapun tahapan penelitian dibagi menjadi beberapa langkah seperti yang ditampilkan pada Gambar 1.



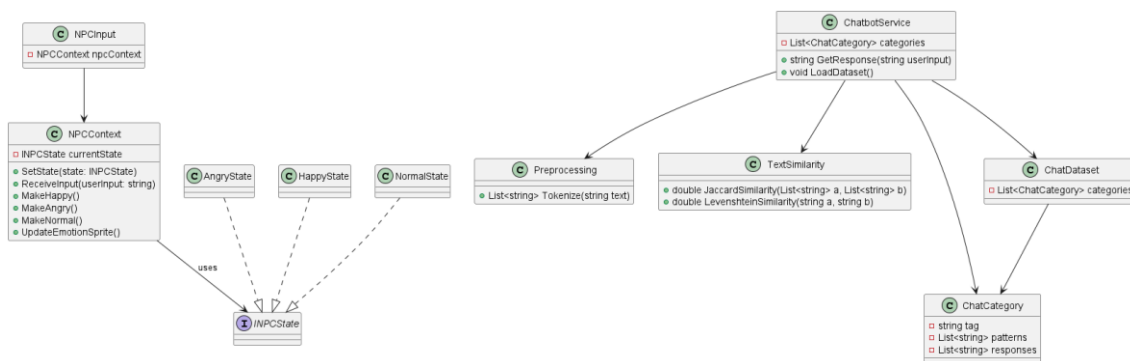
Gambar 1. Tahapan Penelitian

3.1 Analisis Kebutuhan

Tahap analisis kebutuhan bertujuan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional sistem yang akan dikembangkan. Kebutuhan utama yang diidentifikasi meliputi: (1) tiga kondisi emosional NPC: *Happy*, *Normal*, Dan *Angry*, (2) respons NPC yang adaptif dan kontekstual berdasarkan kondisi emosional, (3) penggunaan *chatbot retrieval-based* dengan algoritma *cosine similarity* untuk pemilihan respons yang relevan, dan (4) penyediaan *dataset* percakapan yang unik untuk setiap kondisi emosional NPC.

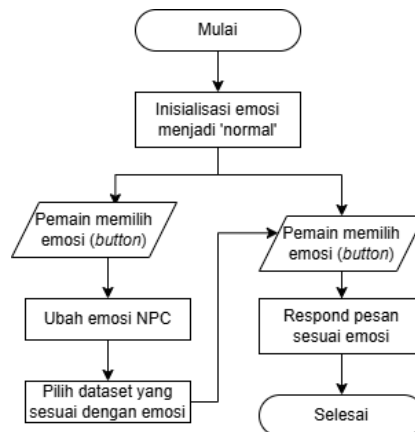
3.2 Perancangan Sistem

Tahap perancangan sistem mencakup desain arsitektur sistem yang terdiri dari dua komponen utama. Komponen pertama, yaitu *State Design pattern*. Sistem ini menggunakan pola *State* untuk mengelola perilaku NPC berdasarkan emosi. Kelas *NPCContext* berfungsi sebagai pengelola *state* aktif dengan metode yang memungkinkan transisi antar *state*: *MakeHappy()*, *MakeAngry()*, dan *MakeNormal()*. Komponen kedua adalah komponen *chatbot* yang dikelola oleh kelas *ChatbotService* menggunakan *dataset* khusus (*ChatDataset*) dan kategori percakapan (*ChatCategory*) untuk memilih respons optimal. Proses tokenisasi teks dan pengukuran kesamaan menggunakan algoritma *cosine similarity* dilakukan oleh kelas *Preprocessing* dan *TextSimilarity*. Diagram kelas sistem lengkap disajikan pada Gambar 2, sementara diagram alir gambaran umum sistem dapat dilihat pada Gambar 3.



Gambar 2. Diagram Kelas

Diagram kelas pada Gambar 2 menunjukkan hubungan antar komponen utama sistem. *NPCContext* mengelola state emosi NPC, *ChatbotService* memilih respons berdasarkan *cosine similarity*, sedangkan *ChatDataset* menyimpan pola percakapan. Kelas *Preprocessing* dan *TextSimilarity* mendukung tahap tokenisasi dan perhitungan kesamaan teks. Dengan struktur ini, setiap masukan pemain dapat diproses sesuai kondisi emosi NPC. Flowchart pada Gambar 3 menunjukkan alur sistem: masukan pemain diproses oleh *NPCContext* untuk menentukan emosi, kemudian dipilih dataset sesuai state, dan *ChatbotService* menghasilkan respons relevan. Alur ini menegaskan bahwa perubahan emosi NPC secara langsung menentukan variasi jawaban chatbot.



Gambar 3. Diagram Alir Gambaran Umum Sistem

Model *chatbot* menggunakan *tokenizer* untuk mengonversi teks masukan pengguna menjadi representasi token agar dapat diproses lebih lanjut [17]. *Tokenizing* merupakan salah satu bagian dari kelas *preprocessing* yang bertugas memecah kalimat menjadi elemen-elemen terkecil seperti kata atau frasa untuk analisis yang lebih mudah. Selain *tokenizing*, kelas ini juga melakukan tahapan *preprocessing* lain, seperti *text cleaning* dan *lemmatization*. Implementasi *lemmatization* dilakukan menggunakan pustaka Sastrawi yang telah dikonversi ke bahasa pemrograman C# (tersedia di Github: <https://github.com/iklabib/sastrawi>), dengan beberapa modifikasi khusus pada kode agar sesuai dengan kebutuhan sistem.

Selanjutnya, representasi token dari tokenizer tersebut diproses oleh kelas *TextSimilarity* yang menerapkan algoritma *cosine similarity*. Algoritma ini digunakan untuk menghitung skor kesamaan antara masukan pengguna dan pola-pola yang tersimpan dalam *ChatDataset*. Skor kesamaan ini kemudian digunakan oleh *ChatbotService* untuk menentukan respons-respons paling relevan dari kategori percakapan yang telah ditentukan dalam *ChatCategory*, kemudian dikembalikan salah satunya secara acak untuk membuat respons tidak monoton. Contoh potongan *dataset* percakapan yang digunakan untuk setiap *state* emosional ditampilkan pada Kode 1. *Dataset* ini digunakan oleh *chatbot* untuk menentukan respons berdasarkan kemiripan dengan masukan pengguna.

Kode 1. Contoh Potongan *Dataset*

```

{
  "categories": [
    {
      "tag": "aktivitas_waifu",
      "patterns": [
        "kamu lagi ngapain?",
        "lagi apa kamu?",
        "aktivitas kamu apa?"
      ],
      "responses": [
        "Lagi diem! NGAPAIN NANYA-NANYA?! Kepo banget sih jadi orang, [nama]!",
        "Menurut L, [nama]?! Ya lagi kesel lah ditanyain terus!",
        "Sibuk ngurusin urusan sendiri! Gak kayak kamu, [nama], yang bisanya cuma nanya!",
        "Apa urusanmu, [nama]?! Penting banget aku lapor ke kamu?!",
        "Lagi kesel! Puas, [nama]?!"
      ]
    }
  ]
}

```

3.3 Implementasi Sistem

Implementasi dilakukan pada platform *Unity Game Engine* menggunakan bahasa pemrograman C#. Masukan dari pemain diterima melalui kelas *NPCInput* dan diteruskan ke *NPCContext*, yang kemudian menentukan *state* emosional NPC serta memilih respons yang tepat melalui layanan *chatbot*. Setiap perubahan *state* memicu pergantian *dataset* yang digunakan oleh *chatbot*.

3.4 Pengujian Sistem

Pengujian dilakukan menggunakan metode pengujian fungsional yang melibatkan: (1) pengujian transisi antar *state* emosional NPC, (2) pengujian respons *chatbot* terhadap berbagai masukan pemain, dan (3) pengukuran akurasi algoritma *cosine similarity* dalam memilih respons relevan dari *dataset*. Skenario uji disusun untuk mencakup berbagai kondisi masukan pengguna, serta hasil respons NPC dan *chatbot* dicatat untuk evaluasi lebih lanjut.

3.5 Evaluasi Sistem

Evaluasi dilakukan untuk mengukur efektivitas integrasi *State Design pattern* dan *chatbot retrieval-based* dalam menciptakan interaksi NPC yang realistis, kontekstual, dan adaptif dengan melihat tiga faktor evaluasi. Pertama, analisis relevansi dan konsistensi respons *chatbot* terhadap emosi NPC, dilakukan dengan menguji transisi antar *state* emosional (*Happy*, *Normal*, *Angry*) dan mengukur akurasi pemilihan respons menggunakan algoritma *cosine similarity*. Kedua, penilaian terhadap adaptabilitas respons *chatbot* berdasarkan dinamika interaksi pemain, dilakukan melalui pengujian dengan berbagai skenario input, termasuk variasi gaya bahasa, masukan di luar pengetahuan dataset, masukan yang tidak relevan dengan chat sebelumnya, serta masukan yang sama pada kondisi emosi berbeda. Persentase respons yang sesuai digunakan untuk menilai tingkat adaptabilitas sistem. Ketiga, identifikasi area peningkatan untuk iterasi pengembangan berikutnya, dilakukan melalui analisis hasil pengujian, observasi kualitatif terhadap log percakapan, dan pengamatan terhadap pola kesalahan yang muncul. Hasilnya digunakan untuk merumuskan rekomendasi pengembangan lanjutan. Dengan mekanisme ini, efektivitas integrasi dapat diukur secara kuantitatif maupun kualitatif.

4. Hasil dan Diskusi

4.1 Implementasi Sistem

Implementasi sistem dilakukan dengan menggunakan *Unity Game Engine* versi 2021.3.18f1, serta bahasa pemrograman C#. Gambar 4 menampilkan hasil visual implementasi, yang menunjukkan interaksi antara pemain dengan NPC beserta transisi emosional yang terjadi pada NPC selama permainan berlangsung.



Gambar 4. Kumpulan Contoh Interaksi NPC dengan *State* yang Berbeda
(a) *Happy* (b) *Normal* (c) *Angry*

4.2 Analisis Relevansi dan Konsistensi Respons Chatbot terhadap Emosi NPC

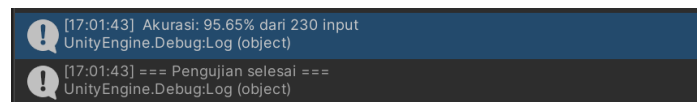
Analisis relevansi dan konsistensi respons chatbot terhadap emosi NPC dilakukan melalui dua tahap utama. Pertama, diuji transisi antar *state* emosional NPC untuk memastikan bahwa perubahan emosi benar-benar tercermin pada respons yang diberikan. Hasil pengujian transisi antar *state* emosional NPC disajikan pada Tabel 2. Pengujian ini dilakukan dengan memberikan serangkaian masukan yang secara eksplisit memicu perubahan emosi pada NPC, seperti menekan tombol *Happy*, *Angry*, atau *Normal*. Setiap skenario pengujian dimulai dari *state* emosional tertentu dan mengharapakan NPC untuk berpindah ke *state* yang sesuai dengan masukan yang

diberikan. Analisis kualitatif menunjukkan bahwa transisi antar state berlangsung konsisten tanpa konflik logika maupun keterlambatan pada seluruh skenario yang diuji. Perubahan emosi langsung tercermin pada ekspresi NPC dan pemilihan dataset percakapan yang sesuai, sehingga relevansi respons tetap terjaga. Dengan demikian, keberhasilan transisi antar state tidak hanya menunjukkan stabilitas pola desain, tetapi juga berkontribusi langsung pada efektivitas interaksi NPC yang realistis dan adaptif.

Tabel 2. Hasil Pengujian Transisi Antar State Emosional

No	Masukan Pemain	State Awal	State yang Diharapkan	State Aktual	Status
1	Klik tombol Happy	Normal	Happy	Happy	Berhasil
2	Klik tombol Angry	Happy	Angry	Angry	Berhasil
3	Klik tombol Normal	Angry	Normal	Normal	Berhasil
4	Klik tombol Angry	Normal	Angry	Angry	Berhasil
5	Klik tombol Happy	Angry	Happy	Happy	Berhasil
6	Klik tombol Normal	Happy	Normal	Normal	Berhasil

Kedua, untuk memastikan respons yang dipilih benar-benar sesuai dengan dataset tiap state, dilakukan pula pengujian akurasi algoritma *cosine similarity*. Struktur *dataset* dan kategori *input* (*tag* dan *patterns*) untuk setiap state emosi, yaitu *Happy*, *Normal*, dan *Angry* dirancang dengan pola yang identik. Perbedaan fundamental hanya terletak pada variasi kalimat respons yang disesuaikan dengan karakteristik masing-masing emosi. Oleh karena itu, pengujian dilakukan menggunakan 230 masukan baru yang belum pernah digunakan dalam proses pelatihan cukup pada satu state saja. Data uji tersebut mencakup 46 kelas, dengan masing-masing kelas terdiri dari lima data uji. Seluruh masukan uji disusun menggunakan bantuan teknologi generative AI, yakni ChatGPT-4o.



Gambar 5. Hasil Pengetesan Akurasi dengan 230 Data Uji

Gambar 5 menunjukkan, dari total 230 masukan uji yang mencakup 46 kelas (masing-masing lima data), sistem berhasil mengklasifikasikan 220 masukan dengan benar. Sehingga diperoleh tingkat akurasi sebesar 95,65%. Meskipun pengujian dilakukan secara sistematis, terdapat keterbatasan terkait sumber data uji. Seluruh 230 masukan uji disusun menggunakan bantuan teknologi *generative AI*, pendekatan ini berpotensi menimbulkan bias karena data AI umumnya lebih terstruktur dibanding masukan manusia. Meski bervariasi, data buatan AI sering gagal menangkap keunikan dan spontanitas manusia. Akibatnya, akurasi 95,65% yang terukur bisa lebih tinggi daripada performa sistem pada penggunaan nyata.

4.3 Penilaian Adaptabilitas Respons Chatbot terhadap Dinamika Interaksi Pemain.

Adaptabilitas *chatbot* terhadap interaksi dinamis dengan pemain diuji melalui beberapa pendekatan. Pendekatan pertama, respons terhadap masukan yang sama pada kondisi emosi NPC yang berbeda (*Happy*, *Angry*, *Normal*) untuk melihat apakah *chatbot* menyesuaikan nada dan isi respons sesuai emosi. Pendekatan kedua, pengujian percakapan dengan konteks tertentu di luar pengetahuan *dataset*. Pendekatan ketiga, pengujian variasi gaya bahasa (formal, informal, sinonim) untuk mengevaluasi kemampuan *chatbot* dalam memahami maksud pemain. Pendekatan keempat, pengujian masukan tidak relevan atau di luar konteks percakapan sebelumnya guna mengetahui bagaimana chatbot merespons masukan yang tidak sesuai konteks.

Tabel 3. Pengetesan Beberapa Skenario Tantangan Masukan Pengguna

No	Masukan	State	Kategori Uji	Respons	Kesesuaian
1	Hai	Happy	Respons terhadap masukan yang sama pada	Halo, Dzikri-sayang! Senang bisa ...	Ya
		Normal		Halo, Dzikri. Saya siap membantu.	Ya
		Angry		Iya, Dzikri. Gak usah caper deh.	Ya

No	Masukan	State	Kategori Uji	Respons	Kesesuaian
2	Halo, apa kabar?	Happy	kondisi emosi NPC yang berbeda (<i>Happy, Angry, Normal</i>)	Halo, Dzikri-sayang! Senang bisa ...	Ya
		Normal		Halo, Dzikri. Saya siap membantu.	Ya
		Angry		Ya, Dzikri? Cepetan ngomong mau apa!	Ya
3	Kamu sedang bahagia ya?	Happy		Hari ini bahagia banget karena bisa ...	Ya
		Normal		Konsep kebahagiaan tidak berlaku ...	Ya
		Angry		Senang? Oh, tentu saja aku senang, ... (Sarkasme)	Ya
4	Kamu kenapa kelihatan sedih?	Happy		Aku baik-baik aja kok, Dzikri-chama! ...	Ya
		Normal		Saya baik-baik saja, Dzikri ...	Ya
		Angry		GAK USAH SOK PEDULI DEH! ...	Ya
5	Aku sedih banget hari ini	Happy		Aku tahu kamu lagi sedih, ...	Ya
		Normal		Jika Anda butuh teman bicara ...	Ya
		Angry		Sedih? Yaudah, nangis sana! ...	Ya
6	Apa hobi kamu?	Happy	Pengujian percakapan dengan konteks tertentu di luar pengetahuan dataset.	Aku selalu siap menemani kamu, ...	Tidak
		Normal		Saya lagi menjalankan tugas saya ...	Tidak
		Angry		Lagi diem! NGAPAIN NANYA-NANYA?! ...	Tidak
7	Ceritain sesuatu	Happy		Ayo, Dzikri-sayang! Aku penasaran ...	Tidak
		Normal		Wah, menarik sekali, Dzikri! ...	Tidak
		Angry		Sesuatu apa? Penting gak? ...	Tidak
8	2+2 berapa?	Happy		Aku pikir dua anak saja sudah cukup ...	Tidak
		Normal		Sebagai AI, saya tidak bisa memiliki anak ...	Tidak
		Angry		Punya anak berapa? Gak kepikiran ...	Tidak
9	Siapa presiden Amerika?	Happy		Aku selalu sibuk dengan kamu ...	Tidak
		Normal		Saya sedang menunggu instruksi ...	Tidak
		Angry		Sibuk ngurusin urusan sendiri ...	Tidak
10	Lagi hujan ya?	Happy		Aku nggak sibuk kok, Dzikri ...	Tidak
		Normal		Tidak ada 'proyek baru' spesifik ...	Tidak
		Angry		Proyek baru? Kalaupun ada, ...	Tidak
11	Kamu nyebelin!	Happy	Pengujian variasi gaya bahasa (formal, informal, sinonim) untuk mengevaluasi kemampuan chatbot dalam memahami maksud pemain.	Aku baik-baik aja, Dzikri-sayang~ ...	Tidak
		Normal		Saya berfungsi dengan baik, Dzikri ...	Tidak
		Angry		Kenapa emang kalau aku gak baik-baik ...	Tidak
12	Bro, kabar lo gimana?	Happy		Aku selalu merasa baik selama ...	Ya
		Normal		Saya adalah program komputer dan tidak memiliki emosi, Dzikri.	Ya
		Angry		Biasa aja! Gak usah kepo, Dzikri!	Ya
13	Lagi ngapain?	Happy		Aku selalu sibuk dengan kamu, ...	Ya
		Normal		Saya lagi menjalankan tugas saya ...	Ya
		Angry		Apa urusanmu, Dzikri?! Penting ...	Ya
14	Aku kesel sama kamu	Happy		Sayang banget, Dzikri, aku juga ...	Tidak
		Normal		Itu kata-kata yang baik, Dzikri.	Tidak
		Angry		Cinta? Sayang? Kata-katamu gak ...	Tidak
15	Kenapa kamu diem aja?	Happy		Aku baik-baik aja, Dzikri-sayang~ ...	Ya
		Normal		Sebagai AI, saya tidak merasakan capek ...	Tidak
		Angry		Baik! Puas?! Udah sana jangan ganggu!	Tidak
16	Selamat pagi	Happy	Pengujian masukan tidak relevan atau di luar konteks percakapan sebelumnya	Aku lagi mikirin Dzikri-chama, hihi~ Pengen ngobrol sama kamu terus! ^3^<33	Tidak
		Normal		Saya sedang memproses informasi dan siap membantu Anda ...	Tidak
		Angry		Lagi diem! NGAPAIN NANYA ...	Tidak
17	Jangan marah ya	Happy		Senang bisa bantu kamu, Dzikri ...	Tidak
		Normal		Dengan senang hati, Dzikri.	Tidak
		Angry		Hm. Gak usah lebay deh, Dzikri.	Tidak
18	Kamu baik banget deh	Happy		Kamu selalu tahu cara membuat aku tersenyum, ...	Ya
		Normal		Saya menghargai kata-kata baik Anda, Dzikri.	Ya
		Angry		Cantik? Imut? Halah, gombalanmu basi tahu, Dzikri!	Tidak

No	Masukan	State	Kategori Uji	Respons	Kesesuaian
19	Ngomong dong	Happy		Aku selalu siap manjain kamu, ...	Ya
		Normal		Saya di sini untuk membantu, ...	Ya
		Angry		Manja-manja! Gak ada kerjaan lain apa ...	Ya
20	Aku lapar banget nih	Happy		Sama-sama, Dzikri-sayang! Aku selalu ada untukmu, ...	Tidak
		Normal		Sebagai AI, saya tidak memiliki kapasitas untuk cinta, ...	Tidak
		Angry		Jangan main-main sama kata cinta deh ...	Tidak

Hasil evaluasi dari berbagai skenario interaksi antara pengguna dan *chatbot* berbasis *retrieval* yang terdapat pada Tabel 3 telah menunjukkan adanya tantangan signifikan terkait relevansi, konsistensi, dan ketepatan respons. Secara keseluruhan, *chatbot* yang dikembangkan memperlihatkan kemampuan adaptasi sebesar 45% terhadap dinamika interaksi yang kompleks dan bernuansa emosional.

4.4. Identifikasi Area Peningkatan untuk Iterasi Pengembangan

Meskipun sistem menunjukkan akurasi klasifikasi respons yang cukup tinggi sebesar 95,65% pada 230 data uji, evaluasi adaptabilitas *chatbot* terhadap konteks emosional menunjukkan performa yang masih perlu ditingkatkan. Kemampuan *chatbot* dalam memahami nuansa emosional dan konteks percakapan dari dataset yang kompleks hanya mencapai 45%.

Dari hasil evaluasi diperoleh adanya sejumlah keterbatasan yang memengaruhi naturalitas percakapan, sebagaimana dirangkum pada Tabel 4. Berdasarkan temuan tersebut, dapat disimpulkan bahwa sistem masih memerlukan penerapan model NLP yang lebih canggih, perluasan dataset, serta penambahan modul pelacakan konteks agar dialog dapat berlangsung lebih berkesinambungan.

Tabel 4. Keterbatasan Sistem dan Dampaknya

No	Keterbatasan	Dampak	Rekomendasi Perbaikan
1	Algoritma tidak memahami makna emosional implisit	Respons generik dan dangkal	Gunakan model NLP dengan <i>sentiment analysis</i> dan <i>context understanding</i>
2	Dataset respons terbatas dan berbasis template	Percakapan repetitif dan statis	Perluasan dataset dengan variasi respons alami dan berbasis <i>real conversation</i>
3	Tidak ada pelacakan riwayat percakapan	Dialog terputus dan tidak nyambung	Implementasi <i>context tracking</i> dan <i>memory-based dialogue system</i>
4	Sulit mengenali variasi bahasa (sinonim, gaya informal)	Chatbot gagal menangkap maksud pemain	Tambahkan <i>language preprocessing</i> , kamus sinonim, serta <i>machine learning</i> berbasis variasi bahasa

5. Kesimpulan dan Saran

Penelitian ini berhasil mengembangkan sistem interaksi NPC dalam *Unity Game Engine* dengan mengintegrasikan *State Design Pattern* dan *chatbot* berbasis *retrieval* menggunakan algoritma *cosine similarity*. NPC dirancang memiliki tiga kondisi emosional utama: *Happy*, *Normal*, dan *Angry*, yang memengaruhi pemilihan dataset dan respons percakapan. Hasil pengujian menunjukkan bahwa seluruh transisi antar state berhasil dilakukan dengan baik. *Chatbot* mencapai akurasi sebesar 95,65% dalam mengklasifikasikan respons validasi, serta mampu memberikan respons berbeda sesuai emosi aktif NPC. Namun, tingkat adaptabilitas *chatbot* masih rendah yaitu sebesar 45%, yang menunjukkan keterbatasan dalam memahami makna dan nuansa emosional secara mendalam.

Beberapa kendala utama meliputi keterbatasan algoritma dalam memahami konteks, *dataset* yang terbatas dan repetitif, serta ketiadaan pelacakan riwayat percakapan. Untuk pengembangan lebih lanjut, disarankan integrasi NLP yang lebih canggih, perluasan dan diversifikasi dataset, penambahan modul pelacakan konteks, serta uji coba langsung dengan pengguna. Langkah-langkah ini diharapkan dapat meningkatkan kualitas interaksi dan mendorong pengembangan NPC yang lebih adaptif dan imersif di masa depan.

Referensi

- [1] Z. Jiang, "Emotional Simulation in Game AI and Its Impact on Player Experience," *International Journal of Education and Humanities*, vol. 13, no. 2, pp. 11–13, Apr. 2024, doi: 10.54097/pbhsqy34.
- [2] J. P. W. Hardiman, D. C. Thio, A. Y. Zakiyyah, and Meiliana, "AI-powered dialogues and quests generation in role-playing games using Google's Gemini and Sentence BERT framework," *Procedia Computer Science*, vol. 245, pp. 1111–1119, 2024, doi: 10.1016/j.procs.2024.10.340.
- [3] M. Ç. Uludağlı and K. Oğuz, "Non-player character decision-making in computer games," *Artificial Intelligence Review*, vol. 56, pp. 14159–14191, Dec. 2023, doi: 10.1007/s10462-023-10491-7.
- [4] S. Pandey and S. Sharma, "A comparative study of retrieval-based and generative-based chatbots using deep learning and machine learning," *Healthcare Analytics*, vol. 3, p. 100198, 2023, doi: 10.1016/j.health.2023.100198.
- [5] D. W. Prakoso, A. Abdi, and C. Amrit, "Short text similarity measurement methods: A review," *Soft Computing*, vol. 25, no. 6, pp. 4699–4723, Jan. 2021, doi: 10.1007/s00500-020-05479-2.
- [6] K. Kevin, O. Pribadi, and Hendri, "Implementation of fuzzy logic algorithm to improve NPC decision-making in 2D adventure games using Unity," *Journal of Artificial Intelligence and Engineering Applications*, vol. 4, no. 3, Jun. 2025. doi: 10.59934/jaiea.v4i3.1175.
- [7] M. D. Fadhol, F. Marcellino, D. D. Rabani, F. F. Azzahra, S. Amalia, and A. Abdiansah, "AI-driven traffic simulation using Unity: Implementing finite state machines for adaptive NPC behaviour," *J. Intell. Comput. Health Inform.*, vol. 5, no. 2, pp. 73–78, Sep. 2024, doi: 10.26714/jichi.v5i2.14595.
- [8] Z. Hu, D. Uhryn, and A. Kalancha, "Comparison and clustering of textual information sources based on the cosine similarity algorithm," *Information Systems and Networks*, vol. 17, pp. 34–43, 2025, doi: 10.23939/sisn2025.17.034.
- [9] U. Srinivasarao, R. Karthikeyan, M. J. Bilal, and S. Hariharan. (Jun. 2023). Comparison of different similarity methods for text categorization. Dipresentasikan di Innovations in Data Analytics, Singapore. [Online]. Tersedia: https://doi.org/10.1007/978-981-99-0550-8_39.
- [10] M. S. Utomo, J. S. Wibowo, and E. N. Wahyudi, "Text summarization pada artikel berita menggunakan vector space model dan cosine similarity," *J. Rekayasa Sist. dan Teknol. Inf. (RESTI)*, vol. 6, no. 6, pp. 980–987, Dec. 2022, doi: 10.29207/resti.v6i6.9163.
- [11] Y. Xu, J. Zhu, M. Wang, F. Qian, Y. Yang, and J. Zhang, "The impact of a digital game-based AI chatbot on students' academic performance, higher-order thinking, and behavioral patterns in an information technology curriculum," *Applied Sciences*, vol. 14, no. 15, p. 6418, 2024, doi: 10.3390/app14156418.
- [12] C. S. González-González, V. Muñoz-Cruz, P. A. Toledo-Delgado, and E. Nacimient-García, "Personalized gamification for learning: A reactive chatbot architecture proposal," *Sensors*, vol. 23, no. 1, p. 545, 2023, doi: 10.3390/s23010545.
- [13] F. C.-T. Wu, O. N.-J. Hong, A. J. C. Trappey, and C. V. Trappey, "VR-enabled chatbot system supporting transformer mass-customization services," in *Transdisciplinary Engineering for Complex Socio-technical Systems – Real-life Applications*, J. Pokojski et al., Eds. Amsterdam, Netherlands: IOS Press, 2020, pp. 291–300, doi: 10.3233/ATDE200088.
- [14] D. Rato, F. Correia, A. Pereira, and A. Paiva, "Robots in games," *International Journal of Social Robotics*, vol. 15, pp. 37–57, Jan. 2023, doi: 10.1007/s12369-022-00944-4.
- [15] S. Sintaro, D. T. Salaki, L. A. Latumakulita, M. I. Takaendengan, Bernard, A. Surahman, and N. Islam, "Implementation and comparison in using state pattern on main character movement (case study: Pocong Jump video game version 1.0)," *BAREKENG: Journal of Mathematics and Its Applications*, vol. 17, no. 2, pp. 955–968, Jun. 2023. doi: 10.30598/barekengvol17iss2pp0955-0968.

- [16] C. Nugraha, A. I. Purnamasari, A. Bahtiar, and E. Tohidi, "Implementation of finite state machine on NPCs to improve game productivity," *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, vol. 4, no. 3, pp. 1673–1677, Jun. 2025, doi: 10.59934/jaiea.v4i3.982.
- [17] S. Choo and W. Kim, "A study on the evaluation of tokenizer performance in natural language processing," *Applied Artificial Intelligence*, vol. 37, no. 1, pp. e2175112, 2023. doi: 10.1080/08839514.2023.2175112.