ANALISIS KOMPARATIF KUALITAS LARGE LANGUAGE MODEL GPT, GEMINI, DAN CLAUDE DALAM MEMAHAMI BAHASA INDONESIA

TUGAS AKHIR

Diajukan oleh:
Zaharatul Maqfirah
NIM : 220705049

Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2026 M/1447 H

LEMBAR PERSETUJUAN

ANALISIS KOMPARATIF KUALITAS LARGE LANGUAGE MODEL GPT, GEMINI, DAN CLAUDE DALAM MEMAHAMI BAHASA INDONESIA

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1) dalam
Prodi Teknologi Informasi

Oleh:
ZAHARATUL MAQFIRAH
NIM : 220705049
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui untuk Dimunaqasyahkan Oleh:

Pembimbing I,	Pembimbing II,
Mulkan Fadhli, S.T.,M.T.	**Ridha Ilahi, M.T**
NIP. 198811282020121006	**NIP. 197905302014031001**

Mengetahui,
Ketua Program Studi Teknologi Informasi

Mulahayati, M.T
NIP. 198301272015032003

LEMBAR PENGESAHAN

ANALISIS KOMPARATIF KUALITAS LARGE LANGUAGE MODEL GPT, GEMINI, DAN CLAUDE DALAM MEMAHAMI BAHASA INDONESIA

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S1)
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Rabu, 12 Agustus 2026 M
28 Shafar 1448 H
Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir

Ketua,	Sekretaris,
Mulkan Nadhli, M.T	**Ridha Ilahi, M.T**
NIP. 198811282020121006	**NIP. 197905302014031001**
Penguji I,	Penguji II,
Gufran Ibnu Yasa, M.T	**Fathiah, M.Eng**
NIP. 198409262014031005	**NIP. 197905302014031001**

Mengetahui:
Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh

Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	:	Zaharatul Mafirah
Nim	:	220705049
Prodi	:	Teknologi Informasi
Fakultas	:	Sains dan Teknologi
Judul	:	Analisis Komparatif Kualitas Language Models GPT, Gemini, dan Claude dalam Memahami Bahasa Indonesia

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 12 Agustus 2026

Yang Menyatakan

(Zaharatul Maqfirah)

ABSTRAK

Perkembangan *Large Language Models* (LLM) menuntut adanya evaluasi komprehensif, khususnya dalam pemrosesan teks berbahasa Indonesia. Penelitian ini bertujuan mengkomparasi performa GPT, Claude, dan Gemini berdasarkan metrik akurasi, efisiensi operasional (latensi dan biaya), serta menguji objektivitas model sebagai juri otomatis (*LLM-as-a-judge*) melalui instrumen *Multiple Choice Questions* (MCQ), Esai, dan *Coding*. Metode penilaian divalidasi melalui pengujian silang antar-*provider* (*cross-evaluation*) dan *Expert Judgment*. Hasil pengujian menunjukkan ketiga model memiliki pemahaman dasar yang sangat baik (*Pass Rate* 100% pada MCQ). Secara keseluruhan, GPT mendominasi sebagai model paling efisien dengan akurasi tertinggi (>98%), latensi tercepat (rata-rata 5.500 ms), dan biaya termurah, yang mana dikonfirmasi oleh validasi pakar sebagai luaran paling logis. Claude unggul dalam narasi komprehensif meski lebih lambat dan rentan *over-specification*, sedangkan Gemini kompetitif pada gaya bahasa natural. Lebih lanjut, analisis evaluasi silang pada tingkat soal mengungkap adanya anomali bias sistem juri AI; GPT dan Claude menunjukkan bias positif (*overconfidence*) saat menilai hasil keluarannya sendiri, sementara Gemini menunjukkan kecenderungan penalti diri (*self-penalization*). Penelitian ini menyimpulkan bahwa penggunaan *LLM-as-a-judge* tidak dapat dilakukan secara mandiri, melainkan mutlak memerlukan agregasi multi-juri dari *provider* yang berbeda untuk menetralisir bias inheren tersebut.

Kata Kunci: *Large Language Models* (LLM), Evaluasi Performa AI, *LLM-as-a-Judge*, Pemrosesan Bahasa Alami, Bias Kecerdasan Buatan.

ABSTRACT

The rapid development of Large Language Models (LLMs) necessitates comprehensive evaluation, particularly in processing Indonesian texts. This study aims to compare the performance of GPT, Claude, and Gemini based on accuracy metrics and operational efficiency (latency and cost), as well as to examine the models' objectivity as automated judges (LLM-as-a-judge) using Multiple Choice Questions (MCQ), Essays, and Coding instruments. The assessment method was validated through cross-provider evaluation and Expert Judgment. The results indicate that all three models possess an excellent foundational understanding (100% Pass Rate on MCQs). Overall, GPT dominates as the most efficient model, demonstrating the highest accuracy (>98%), the fastest latency (averaging 5,500 ms), and the lowest cost, which was confirmed by expert validation as producing the most logical outputs. Claude excels in comprehensive narratives despite being slower and prone to over-specification, while Gemini is competitive in generating natural language styles. Furthermore, cross-evaluation analysis at the item level reveals bias anomalies within the AI judging system; GPT and Claude demonstrate a positive bias (overconfidence) when evaluating their own outputs, whereas Gemini exhibits a tendency towards self-penalization. This study concludes that the implementation of LLM-as-a-judge should not be conducted independently; it strictly necessitates a multi-judge aggregation from different providers to neutralize these inherent biases.

Keyword: *Large Language Models* (LLM), AI Performance Evaluation, LLM-as-a-Judge, Natural Language Processing, Artificial Intelligence Bias.

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas segala rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul "**Analisis Komparatif Kualitas Language Models GPT, Gemini, dan Claude dalam Memahami Bahasa Indonesia"**. Tak lupa pula Shalawat beriringkan salam tercurahkan kepada sang baginda Nabi Muhammad SAW. Tugas akhir ini disusun sebagai salah satu persyaratan untuk menyelesaikan pendidikan pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh.

Penulis menyadari sepenuhnya bahwa skripsi ini masih memiliki banyak kekurangan, baik dari segi materi maupun tata bahasa, mengingat keterbatasan pengetahuan dan pengalaman yang dimiliki. Oleh karena itu, kritik dan saran yang membangun dari pembaca sangat penulis harapkan demi perbaikan dan penyempurnaan di masa mendatang.

Kesempatan ini penulis gunakan untuk menyampaikan rasa terima kasih dan penghargaan yang setinggi-tingginya kepada berbagai pihak yang telah memberikan dukungan, bimbingan, dan kontribusi tak ternilai :

1. Orang tua tercinta, Bapak M. Husni dan Ibu Kamaliah. Tiada kata yang cukup untuk membalas besarnya jasa, pengorbanan, dan cinta kasih tulus yang telah kalian berikan. Terima kasih karena senantiasa melimpahkan semangat, dukungan moril, serta doa yang tidak terputus sejak awal masa perkuliahan hingga penulis berada di tahap penyusunan Skripsi ini. Penulis menyadari sepenuhnya bahwa tanpa kehadiran, rida, serta doa yang terus mengalir dari Bapak dan Ibu, penulis tidak akan pernah memiliki kekuatan untuk bertahan dan sampai pada titik penyelesaian tugas akhir ini.
2. Bapak Mulkan Fadhli, S.T., M.T., selaku dosen pembimbing I dan Bapak Ridha Ilahi, M.T., selaku dosen pembimbing II, yang telah meluangkan waktu memberikan arahan, masukan dan bimbingan yang sangat berharga selama proses penyusunan Skripsi.

3. Ibu Malahayati, M.T., selaku Ketua Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh.
4. Ibu Cut Ida Rahmadiana, S.Si, selaku seseorang yang telah banyak membantu selama masa perkuliahan penulis.
5. Kakak penulis, Ivon Maryati Widiya, yang senantiasa memberikan dukungan tanpa henti dan selalu meyakinkan penulis untuk tidak menyerah dalam menyelesaikan skripsi ini. Serta abang penulis, Ivan Maulidani, yang walaupun tidak dapat membersamai secara langsung, namun selalu memberikan kepercayaan penuh dan menjadi pelipur lara yang selalu menghibur penulis di saat-saat terpuruk.
6. Sahabat-sahabat terbaik penulis, Uswatun Nisa dan Nurul Azila, yang selalu tulus mendengar, memahami, dan menemani setiap langkah penulis. Terima kasih karena telah menjadi tempat bersandar di saat lelah, menerima segala keluh kesah dengan sangat sabar, dan selalu menguatkan ketika penulis merasa ingin menyerah.
7. Seluruh kawan-kawan dan rekan seperjuangan, yang tidak dapat disebutkan satu per satu. Terima kasih atas kebersamaan, semangat, dan dukungan moral yang membuat perjalanan akademik ini terasa lebih bermakna.

Banda Aceh, 12 Agustus 2026

Penulis,

Zaharatul Maqfirah

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Berikut adalah hasil perombakan latar belakang Anda. Saya telah merapikan struktur kalimatnya, menghilangkan kata hubung di awal paragraf, menjaga alur yang mengerucut (dari umum ke khusus), serta menyisipkan satu paragraf murni dari sudut pandang Anda sebagai penulis (tanpa kutipan) sebelum paragraf penutup.

Perkembangan *Artificial Intelligence* (AI) dalam beberapa tahun terakhir mengalami peningkatan yang sangat pesat, khususnya pada bidang *Natural Language Processing* (NLP). Kehadiran *Large Language Model* (LLM) generatif modern, seperti GPT, Gemini, dan Claude, telah mengubah cara manusia berinteraksi dengan sistem berbasis bahasa dengan kualitas yang semakin mendekati kemampuan kognitif manusia. Teknologi ini tidak hanya diandalkan dalam percakapan digital sehari-hari, tetapi juga mulai dimanfaatkan secara luas dalam sektor pendidikan, industri, penelitian, hingga pengembangan perangkat lunak.

Kapabilitas generatif luar biasa yang ditunjukkan oleh LLM saat ini sayangnya masih menyisakan celah pengujian yang signifikan akibat dominasi evaluasi berbasis *dataset* berbahasa Inggris. Performa tinggi pada bahasa Inggris tidak dapat serta-merta menjadi acuan langsung bagi Bahasa Indonesia, yang memiliki kompleksitas afiksasi, struktur kalimat fleksibel, dan variasi dialek yang khas. Di lingkungan akademik, adopsi LLM membawa potensi besar untuk membantu aktivitas mahasiswa, seperti menyusun ringkasan terstruktur (Rane et al., 2024). Namun, praktik ini juga memunculkan risiko halusinasi informasi di mana model menghasilkan jawaban yang terkesan logis tetapi tidak faktual (de Carvalho Souza, 2025). Integrasi kecerdasan buatan pada dasarnya dirancang untuk mengaugmentasi atau mendukung penalaran manusia, bukan untuk sepenuhnya menggantikan keahlian analitis penggunanya (Autor, 2024). Kebutuhan akan evaluasi komprehensif karenanya menjadi krusial, yakni dengan menggunakan instrumen berbahasa Indonesia autentik yang memiliki variasi format soal seperti pilihan ganda dan esai

terbuka yang terbukti memengaruhi reliabilitas jawaban model (Mykhalko et al., 2024) demi mengukur objektivitas, akurasi, efisiensi, dan konsistensi sistem.

Tantangan dalam pemrosesan Bahasa Indonesia ini menuntut adanya pembuktian empiris mengenai kelayakan operasional model-model AI tersebut saat dihadapkan pada konteks akademik lokal. Mahasiswa sebagai pengguna akhir sering kali dibingungkan oleh berbagai klaim keunggulan dari masing-masing pengembang AI tanpa adanya transparansi metrik yang terukur untuk konteks bahasa ibu mereka. Ketidakmampuan sebuah model dalam menangkap nuansa akademis berbahasa Indonesia bukan sekadar masalah teknis, melainkan dapat berdampak langsung pada penurunan kualitas penalaran kritis dan validitas tugas perguruan tinggi. Oleh sebab itu, pengujian yang berfokus pada skenario nyata mahasiswa menjadi sebuah urgensi untuk memetakan secara jelas batasan dan keunggulan tiap-tiap model.

Penelitian ini secara khusus difokuskan untuk menganalisis dan membandingkan kualitas GPT, Gemini, dan Claude dalam memproses Bahasa Indonesia melalui skenario tugas yang relevan dengan kebutuhan mahasiswa. Selain mengukur metrik akurasi, evaluasi dalam penelitian ini juga mempertimbangkan kemampuan penalaran kompleks dari model-model generasi terbaru (Uppalapati & Nag, 2024). Penggunaan *framework* evaluasi otomatis Promptfoo dan instrumen Indo-Bench diharapkan mampu menghasilkan penilaian yang terstandardisasi. Hasil evaluasi ini pada akhirnya dirancang untuk memberikan rekomendasi objektif bagi mahasiswa dalam memilih model AI yang paling optimal, akurat, dan efisien demi mendukung produktivitas akademik di kampus.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana tingkat akurasi dan koherensi respons GPT, Gemini, dan Claude pada berbagai jenis tugas berbahasa Indonesia?

2. Bagaimana performa efisiensi teknis ketiga model tersebut jika ditinjau dari aspek latensi respons dan estimasi biaya penggunaan token dalam pemrosesan Bahasa Indonesia?
3. Apa saja kelebihan dan kekurangan masing-masing model yang dapat dijadikan dasar rekomendasi bagi mahasiswa dalam memilih layanan AI generatif yang paling optimal untuk mendukung kegiatan akademik ?

1.3 Tujuan Penelitian

Tujuan khusus penelitian ini adalah:

1. Membandingkan tingkat akurasi dan koherensi respons GPT, Gemini, dan Claude pada berbagai format pertanyaan berbahasa Indonesia.
2. Mengukur performa teknis sederhana dari ketiga model tersebut berdasarkan rata-rata waktu respons (latensi) dan perkiraan konsumsi token.
3. Memberikan saran atau rekomendasi penggunaan model AI yang paling efektif bagi mahasiswa untuk mendukung kegiatan akademik berdasarkan hasil pengujian.

1.4 Batasan Penelitian

Batasan dalam penelitian ini adalah sebagai berikut :

1. Objek Penelitian yang digunakan terbatas pada saat penelitian dilakukan, yaitu GPT (OpenAI), Gemini (Google), dan Claude (Anthropic).
2. Pengujian dilakukan melalui *Application Programming Interface* (API) resmi yang diakses melalui penyedia layanan.
3. Fokus penelitian hanya pada Bahasa Indonesia dengan berbagai variasinya (formal, informal, dan teknis) sesuai dengan dataset yang tersedia.
4. Data pengujian dalam penelitian ini berupa instrumen pengujian mandiri yang disusun dari kumpulan pertanyaan resmi dan tervalidasi, mencakup kategori *Multiple Choice Questions* (MCQ), soal esai (*Question Answering*), dan tugas pemrograman (*coding*).

5. Proses *benchmarking* dilakukan menggunakan framework otomatis Promptfoo untuk mengukur metrik kuantitatif dan kualitatif.
6. Evaluasi dibatasi pada aspek akurasi jawaban (*pass rate*), koherensi teks, latensi respons (kecepatan), dan estimasi biaya penggunaan token.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut :

1. Bagi Akademisi

   Sebagai referensi data mengenai performa model GPT, Gemini, dan Claude khusus dalam pengolahan Bahasa Indonesia. Selain itu, penelitian ini dapat menjadi contoh penerapan evaluasi otomatis menggunakan *framework* Promptfoo untuk penelitian sejenis di masa depan.
2. Bagi Pengembang Aplikasi

   Memberikan informasi perbandingan mengenai akurasi, kecepatan (latensi), dan biaya operasional dari ketiga model tersebut. Data ini dapat membantu pengembang dalam memilih model yang paling tepat dan efisien untuk diintegrasikan ke dalam sistem atau aplikasi berbasis Bahasa Indonesia.
3. Bagi Pengguna Teknologi AI

   Memberikan gambaran mengenai kelebihan dan keterbatasan masing-masing model dalam memahami instruksi Bahasa Indonesia, sehingga pengguna dapat memilih layanan AI yang paling sesuai dengan kebutuhan mereka

BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Sejumlah penelitian telah melakukan studi komparatif terhadap model-model LLM seperti GPT, Gemini, dan Claude pada berbagai domain aplikasi. Umumnya, penelitian tersebut bertujuan menilai perbedaan kapabilitas model berdasarkan konteks tugas tertentu, seperti manajemen bisnis, conversational AI, maupun pengambilan keputusan klinis. Hasil dari penelitian-penelitian tersebut menunjukkan bahwa tidak ada satu model yang unggul pada semua skenario, melainkan performa model sangat dipengaruhi oleh jenis tugas, bentuk pertanyaan, serta cara evaluasi yang digunakan.

Salah satu penelitian yang secara langsung membandingkan GPT, Gemini, dan Claude dilakukan oleh (Mykhalko et al., 2024) melalui studi diagnostik kasus klinis menggunakan dua pendekatan bentuk soal, yaitu pertanyaan terbuka (*open-ended*) dan pilihan ganda (*multiple choice*). Penelitian ini bertujuan untuk melihat bagaimana perbedaan format pertanyaan memengaruhi performa model dalam menghasilkan jawaban yang benar dan konsisten. Ketiga model diberikan serangkaian kasus klinis yang sama, kemudian hasil jawabannya dievaluasi berdasarkan tingkat akurasi serta konsistensi antar jawaban. Hasil penelitian menunjukkan bahwa performa ketiga model meningkat secara signifikan ketika pertanyaan disajikan dalam bentuk pilihan ganda dibandingkan pertanyaan terbuka. Hal ini mengindikasikan bahwa model cenderung lebih akurat ketika diberikan batasan konteks jawaban. Selain itu, penelitian ini menggunakan metrik statistik seperti Cohen's Kappa untuk mengukur konsistensi jawaban model terhadap kunci jawaban yang telah ditentukan. Pendekatan ini memperlihatkan bahwa evaluasi LLM tidak hanya dapat dilakukan melalui benar atau salahnya jawaban, tetapi juga melalui tingkat konsistensi respons model pada berbagai bentuk pertanyaan.

Penelitian tersebut juga menyoroti bahwa meskipun ketiga model mampu memberikan jawaban yang terlihat meyakinkan pada pertanyaan terbuka, tidak semua jawaban tersebut sepenuhnya sesuai dengan diagnosis yang benar.

Kondisi ini menunjukkan adanya kecenderungan model menghasilkan jawaban yang tampak logis namun tidak sepenuhnya akurat secara faktual. Temuan ini memperkuat pentingnya penggunaan metrik evaluasi yang tidak hanya berbasis akurasi, tetapi juga mempertimbangkan kualitas dan keandalan keluaran model.

Pendekatan yang digunakan dalam penelitian ini memberikan inspirasi langsung terhadap metode yang digunakan dalam penelitian ini, khususnya dalam variasi bentuk soal yang digunakan pada dataset pengujian, serta pentingnya mengukur konsistensi dan kualitas keluaran model pada berbagai tipe pertanyaan. Selain penelitian utama tersebut, terdapat beberapa penelitian lain yang relevan dan mendukung penelitian ini, sebagaimana dirangkum dalam Tabel 2.1.

Tabel 2. 1 Penelitian Terkait

Peneliti	**Model yang Dibandingkan**	**Hasil Penelitian**	**Relevansi dengan Penelitian Ini**	**Perbedaan dengan Penelitian Terkait**
(Rane et al., 2024)	Gemini, ChatGPT	Ditemukan bahwa ChatGPT unggul dalam menghasilkan penjelasan yang lebih terstruktur dan detail, sementara Gemini lebih cepat dalam merespons dan lebih ringkas. Performa keduanya sangat bergantung pada jenis tugas yang diberikan, sehingga tidak ada model yang unggul di semua skenario.	Dasar penggunaan berbagai skenario pengujian NLP agar perbandingan lebih representatif	Fokus pada aplikasi bisnis secara umum, sedangkan penelitian ini berfokus pada analisis mendalam terhadap kualitas linguistik dalam Bahasa Indonesia.

(de Carvalho Souza, 2025)	Grok, Gemini, ChatGPT, DeepSeek	Ditemukan bahwa seluruh model mampu menghasilkan respons yang terlihat meyakinkan, namun sering mengandung informasi yang tidak sepenuhnya faktual (*hallucination*). Model juga menunjukkan bias pada konteks tertentu, terutama pada pertanyaan yang ambigu.	Dasar evaluasi hallucination dan kualitas makna dalam penelitian ini	Menitikberatkan pada perilaku model secara universal, sementara penelitian ini menguji akurasi semantik dan konteks lokal pada dialek serta tata bahasa Indonesia.
(Alhur, 2024)	ChatGPT, Gemini, Co-Pilot	Model mampu memberikan saran medis yang lengkap dan sistematis, tetapi terdapat inkonsistensi pada beberapa kasus serta kecenderungan memberikan jawaban yang terlalu percaya diri meskipun tidak sepenuhnya akurat.	Dasar evaluasi kualitas dan konsistensi jawaban model	Berfokus pada efektivitas model di domain kesehatan, sedangkan penelitian ini mengevaluasi kemampuan generatif umum pada versi model terbaru.

(Uppalapati & Nag, 2024)	ChatGPT, Claude, Bard, Perplexity	Ditemukan bahwa Claude unggul dalam kejelasan dan kelengkapan jawaban, ChatGPT unggul dalam relevansi informasi, sementara model lain bervariasi tergantung kompleksitas kasus. Evaluasi menggunakan multi-metrik memberikan gambaran performa yang lebih menyeluruh.	Dasar penggunaan multi-metrik evaluasi (relevansi, kejelasan, kelengkapan)	Menggunakan model generasi sebelumnya, sedangkan penelitian ini menggunakan model terbaru danSubjek evaluasi pada penelitian tersebut adalah ketepatan diagnosis medis, sementara penelitian ini menitikberatkan pada kualitas pemahaman struktur kalimat, semantik, dan pragmatik dalam konteks Bahasa Indonesia

2.2 Landasan Teori

2.2.1 Large Language Model (LLM)

Large Language Model (LLM) merupakan model kecerdasan buatan berbasis arsitektur *transformer* yang dilatih menggunakan data teks dalam jumlah sangat besar untuk memahami serta menghasilkan bahasa alami. Berbeda dengan pendekatan Natural Language Processing (NLP) sebelumnya yang umumnya berfokus pada klasifikasi teks atau prediksi kata, LLM generatif mampu menerima instruksi dalam bentuk bahasa alami dan menghasilkan respons yang kontekstual, terstruktur, dan koheren seperti percakapan manusia. Kemampuan ini membuat LLM tidak hanya berfungsi sebagai sistem tanya jawab, tetapi juga mampu melakukan berbagai tugas berbasis bahasa seperti peringkasan teks (*summarization*),

penerjemahan (*translation*), penjelasan konsep, serta mengikuti instruksi dalam format tertentu (*instruction following*). Karakteristik tersebut menjadi ciri utama LLM generatif modern yang saat ini direpresentasikan oleh model-model seperti GPT, Gemini, dan Claude.

LLM bekerja dengan memproses teks dalam bentuk token dan memahami hubungan antar token tersebut secara semantik. Model tidak hanya mengenali kata, tetapi juga menangkap makna kalimat, konteks paragraf, serta maksud dari perintah yang diberikan. Oleh karena itu, kualitas keluaran LLM sangat bergantung pada kemampuan model dalam memahami makna bahasa, bukan sekadar kecocokan pola kata. Meskipun memiliki kemampuan penalaran yang tinggi, LLM masih menghadapi tantangan besar terkait halusinasi informasi dan keterbatasan dalam sintesis literatur yang kompleks (Sparkman & Witt, 2025). Selain itu, keluaran model dapat berbeda tergantung pada bagaimana instruksi diberikan, sehingga evaluasi terhadap LLM perlu mempertimbangkan aspek pemahaman makna, koherensi teks, serta kepatuhan terhadap instruksi.

Karakteristik-karakteristik tersebut menjadi dasar dalam penelitian ini untuk membandingkan GPT, Gemini, dan Claude. Ketiga model tersebut memiliki kemampuan LLM generatif modern yang serupa, sehingga dapat dianalisis berdasarkan bagaimana masing-masing model memahami dan menghasilkan teks dalam Bahasa Indonesia.

2.2.2 Karakteristik Bahasa Indonesia dalam NLP

Perkembangan NLP di Indonesia telah bergeser dari metode berbasis aturan menuju penggunaan model transformator besar, namun tantangan dalam evaluasi spesifik bahasa lokal tetap menjadi fokus utama (Amien, 2023). Bahasa Indonesia memiliki karakteristik linguistik yang berbeda dengan bahasa Inggris yang selama ini banyak digunakan dalam pengembangan dan evaluasi model Natural Language Processing (NLP). Perbedaan tersebut menyebabkan hasil pengujian model pada bahasa Inggris tidak dapat langsung diterapkan pada Bahasa Indonesia. Oleh karena itu, pengujian kemampuan Large Language Model (LLM) pada Bahasa

Indonesia menjadi penting untuk mengetahui sejauh mana model mampu memahami struktur dan makna bahasa ini.

Salah satu karakteristik utama Bahasa Indonesia adalah penggunaan sistem afiksasi. Imbuhan seperti awalan *me-*, *di-*, dan *ber-*, serta akhiran -*kan* dan -*i* dapat mengubah makna kata secara signifikan. Sebuah kata dasar dapat memiliki berbagai bentuk kata bergantung pada imbuhan yang digunakan, sehingga model bahasa perlu memahami hubungan antar bentuk kata tersebut agar dapat menangkap makna kalimat dengan tepat. Selain itu, Bahasa Indonesia tidak memiliki sistem *tense*s yang jelas seperti bahasa Inggris. Penanda waktu umumnya dinyatakan melalui keterangan tambahan dalam kalimat, bukan melalui perubahan bentuk kata kerja. Hal ini menuntut model untuk memahami konteks kalimat secara keseluruhan.

Struktur kalimat Bahasa Indonesia juga relatif fleksibel dan memungkinkan variasi susunan kata tanpa mengubah makna utama. Kondisi ini dapat menjadi tantangan bagi model bahasa dalam menjaga konsistensi dan pemahaman semantik. Di samping itu, dalam penggunaan sehari-hari Bahasa Indonesia sering bercampur dengan bahasa daerah, bahasa Inggris, serta bentuk bahasa informal atau slang. Fenomena ini banyak ditemukan dalam percakapan digital dan media sosial, sehingga variasi bahasa yang muncul menjadi sangat luas.

Bahasa Indonesia memiliki karakteristik sintaksis yang unik dan fleksibel, di mana susunan kata, frasa, dan klausa membentuk fungsi gramatikal yang sangat menentukan makna kalimat (Ramadhani et al., 2025). Namun, fleksibilitas ini sering memicu ambiguitas dan kesalahan sintaksis, seperti ketidaktepatan struktur subjek-predikat atau penggunaan diksi yang tidak efektif (Agustina & Ramadhan, 2023). Kompleksitas struktural inilah yang menjadi tantangan besar bagi model bahasa generatif, sehingga pengujian spesifik diperlukan untuk memastikan LLM tidak hanya menerjemahkan secara harfiah, tetapi benar-benar memahami logika sintaksis Bahasa Indonesia.

Karakteristik lain yang perlu diperhatikan adalah penggunaan Bahasa Indonesia dalam konteks formal, seperti pada bidang hukum, pendidikan,

dan teknologi. Teks pada bidang-bidang tersebut umumnya memiliki struktur kalimat yang panjang, formal, dan menggunakan istilah teknis tertentu. Oleh karena itu, LLM yang diuji pada Bahasa Indonesia perlu mampu memahami teks formal yang kompleks sekaligus dapat menangkap makna dari bahasa yang lebih informal dalam komunikasi sehari-hari.

Kesenjangan dalam sumber daya Bahasa Indonesia sering kali disebabkan oleh ketergantungan pada dataset yang diterjemahkan dari bahasa Inggris, yang sering kali kehilangan otentisitas linguistik dan konteks budaya. Menurut (Wiyono et al., 2025), penggunaan dataset yang ditulis secara *native* oleh penutur asli sangat krusial untuk mencerminkan norma pragmatis Bahasa Indonesia. Sejalan dengan hal tersebut, (Koto & Baldwin, 2020) menegaskan bahwa standarisasi *benchmark* seperti *IndoLEM* dan *IndoBERT* merupakan langkah fundamental dalam riset NLP Bahasa Indonesia guna mengatasi kelangkaan dataset yang terstandarisasi dan berkualitas tinggi.

2.2.3 Metrik Evaluasi Kualitas LLM

Evaluasi terhadap Large Language Model (LLM) tidak dapat dilakukan hanya dengan menilai benar atau salahnya jawaban. Hal ini disebabkan oleh sifat model generatif yang menghasilkan teks secara terbuka, sehingga satu pertanyaan dapat memiliki lebih dari satu jawaban yang benar secara makna. Oleh karena itu, evaluasi LLM memerlukan pendekatan yang mempertimbangkan kualitas makna jawaban sekaligus efisiensi operasional model dalam menghasilkan respons.

Evaluasi terhadap kualitas teks yang dihasilkan oleh model generatif harus selaras dengan ekspektasi atau preferensi manusia (*human preference*). Dalam konteks bahasa Indonesia, kualitas luaran model bahasa umumnya dievaluasi berdasarkan dua kriteria utama, yaitu relevansi (*relevance*) dan kefasihan (*fluency*). Relevansi mengukur sejauh mana respons yang dihasilkan mampu menjawab maksud dari instruksi (*prompt*) yang diberikan, sedangkan kefasihan menilai kebenaran tata bahasa, koherensi, dan tingkat kealamian (keluwesan) bahasa yang digunakan.

Kedua aspek ini mendasari perlunya rubrik penilaian spesifik ketika mengevaluasi jawaban model pada tugas yang bersifat terbuka, guna memastikan jawaban tidak hanya tepat secara faktual tetapi juga natural secara linguistik.

Berdasarkan pertimbangan tersebut, penelitian ini menggunakan beberapa metrik evaluasi yang mencakup kualitas linguistik dan efisiensi operasional model. Dari sisi kualitas linguistik, evaluasi dilakukan dengan melihat kesesuaian makna jawaban model terhadap instrumen pengujian. Dalam penelitian ini, aspek tersebut direpresentasikan melalui metrik pass rate dan average score, yang menggambarkan tingkat keberhasilan model dalam menghasilkan jawaban yang relevan dan sesuai dengan kriteria penilaian yang telah ditentukan.

Selain itu, evaluasi juga mempertimbangkan kemampuan model dalam menghasilkan jawaban yang koheren dan mengikuti instruksi yang diberikan. Pada tugas berbentuk pilihan ganda, penilaian dilakukan dengan metode exact match terhadap kunci jawaban yang tersedia. Sementara itu, pada tugas esai, kualitas jawaban dianalisis menggunakan pendekatan LLM-as-a-Judge, di mana model evaluator menilai koherensi, kelengkapan, dan relevansi jawaban berdasarkan rubrik penilaian yang telah ditetapkan.

Di samping kualitas jawaban, penelitian ini juga mengevaluasi aspek efisiensi operasional model. Efisiensi tersebut diukur melalui latensi respons, yaitu waktu yang dibutuhkan model untuk menghasilkan jawaban setelah menerima instruksi. Pengukuran latensi dilakukan berdasarkan parameter *latencyMs* yang dihasilkan selama proses pengujian melalui API.

Mengevaluasi aplikasi berbasis LLM menghadirkan tantangan tersendiri karena metrik otomatis tradisional sering kali tidak berkorelasi kuat dengan penilaian manusia, sementara evaluasi manusia murni memakan biaya besar dan rentan terhadap subjektivitas (Abeysinghe Bhashithe & Ruhan Circi, 2024) . Oleh karena itu, pendekatan evaluasi faktorial (*factored evaluation*) yang memecah kualitas menjadi beberapa dimensi multidimensional seperti akurasi, kelengkapan, dan koherensi

sangat direkomendasikan agar analisis kelemahan model menjadi lebih terarah dan objektif (Hashemi et al., 2024).

Selain itu, penelitian ini turut mempertimbangkan estimasi biaya penggunaan token sebagai indikator efisiensi penggunaan sumber daya komputasi. Biaya ini dihitung berdasarkan jumlah token input dan output yang digunakan oleh model selama proses pengujian. Dengan menggabungkan evaluasi kualitas jawaban dan efisiensi operasional, penelitian ini diharapkan dapat memberikan gambaran yang lebih komprehensif mengenai performa GPT, Gemini, dan Claude dalam memahami instruksi berbahasa Indonesia.

2.2.4 Instrumen dan Framework Evaluasi

2.2.4.1 Promptfoo

Gambar 2. 1 Promptfoo. Sumber:Promptfoo,2024

Gambar 2.1 menampilkan logo Promptfoo, sebuah kerangka kerja (*framework*) berbasis *open-source* yang dirancang khusus untuk melakukan pengujian dan evaluasi terhadap luaran *Large Language Models* (LLM) secara sistematis. Dalam konteks pengembangan perangkat lunak berbasis AI, Promptfoo memungkinkan pengembang untuk menjalankan pengujian otomatis (*automated testing*) terhadap berbagai model sekaligus dengan menggunakan input (*prompt*) yang identik.

Beberapa fitur utama Promptfoo yang menjadikannya standar dalam proses *benchmarking* adalah:

1. **Multi-Provider Support:** Kemampuan untuk terhubung dengan berbagai API model bahasa seperti OpenAI, Google Vertex AI (Gemini), dan Anthropic (Claude) secara simultan.

2. **Quantitative Metrics:** Memberikan hasil pengukuran yang presisi terhadap metrik teknis seperti *pass rate*, latensi respons, dan jumlah penggunaan token per unit waktu.
3. **Deterministic Evaluation:** Dengan pengaturan parameter seperti *temperature* yang dapat dikontrol, Promptfoo menjamin bahwa variasi luaran model disebabkan oleh kapabilitas model itu sendiri, bukan karena faktor kebetulan.

2.2.4.2 Metode *LLM-as-a-Judge* dan Rubrik Berbasis Tugas

Dalam mengatasi tantangan *data contamination* (kebocoran data), penelitian ini mengadopsi prinsip untuk mengutamakan instrumen evaluasi yang tahan terhadap hafalan model (*contamination-free*), yakni dengan menyusun instrumen pengujian secara mandiri yang tidak terdapat dalam korpus pelatihan model. Selain itu, karena evaluasi teks generatif bersifat subjektif, penelitian ini menerapkan metode *LLM-as-a-Judge*. Pendekatan ini mendelegasikan proses penilaian kepada model bahasa canggih untuk bertindak sebagai evaluator independen yang menilai kualitas respons model lain, meskipun perlu diwaspadai adanya kerentanan terhadap berbagai bias seperti bias preferensi diri (*self-preference bias*) atau bias panjang jawaban (*verbosity bias*) (Doddapaneni et al., 2024).

Oleh karena itu, penggunaan rubrik penilaian yang ketat dan terstandarisasi menjadi mekanisme kendali utama dalam memastikan objektivitas juri AI. Lebih lanjut, selain penggunaan rubrik, objektivitas Juri AI juga dijamin melalui penerapan metode anonimisasi data input (*blind evaluation*) pada saat proses evaluasi berlangsung. Secara teoretis, menyembunyikan identitas pembuat respons (GPT, Gemini, atau Claude) dari pandangan juri AI bertujuan untuk memitigasi risiko *egocentric bias* atau *self-preference bias*, yakni fenomena di mana evaluator LLM memiliki kecenderungan mengenali dan menyukai teks yang dihasilkannya sendiri (Doddapaneni et al., 2024). Mekanisme anonim semacam ini dapat membantu mengeliminasi bias terhadap

LLM tertentu sampai batas tertentu, sehingga secara efektif memastikan keadilan dalam proses evaluasi (Cheng & Chen, 2024). Melalui implementasi anonimisasi (penyamaran identitas model) pada sistem pengelola, juri AI dipaksa untuk mengevaluasi kualitas luaran secara objektif, murni berdasarkan pemenuhan kriteria rubrik tanpa terpengaruh oleh identitas model penghasil respons.

Evaluasi terhadap luaran LLM yang bersifat terbuka (*open-ended*) seperti jawaban esai dan kode pemrograman tidak dapat diukur secara biner (benar/salah) atau sekadar menggunakan pencocokan teks (*exact match*). Agar proses penjurian yang dilakukan oleh *LLM-as-a-Judge* ini dapat diandalkan untuk meneliti kemampuan kritis lain seperti keakuratan faktual, kepatuhan instruksi, maupun logika penalaran (Doddapaneni et al., 2024), juri AI dipandu oleh Rubrik Penilaian Spesifik (*Task-Specific Rubrics*). Rubrik ini menetapkan kriteria kelengkapan faktual, kebenaran logika, dan pemenuhan instruksi kondisional yang dikuantifikasi menjadi skor numerik.

2.2.4.3 Konsep Parameter Evaluasi Lokal (Perspektif Indo-Bench)

Indo-Bench merupakan salah satu platform *benchmarking* standar yang memetakan kompleksitas linguistik dalam konteks Bahasa Indonesia. Platform ini menekankan bahwa evaluasi model bahasa lokal harus mengukur dimensi krusial seperti Validitas Semantik (ketepatan makna), Akurasi Morfologis (pemahaman afiksasi/imbuhan), dan Variasi Kontekstual (pemahaman teks akademik/teknis). Meskipun penelitian ini menggunakan instrumen pengujian mandiri yang tertutup, dimensi-dimensi krusial dari Indo-Bench tersebut diadopsi sebagai dasar filosofis dalam menyusun kriteria rubrik penilaian pada sistem Promptfoo. Hal ini dilakukan untuk memastikan bahwa rubrik yang disematkan pada juri AI benar-benar mampu menguji ketepatan semantik dan logika kontekstual dalam Bahasa Indonesia.

2.2.4.4 API (Application Programming Interface)

Application Programming Interface (API) merupakan sekumpulan protokol dan instruksi perangkat lunak yang memungkinkan satu aplikasi untuk berkomunikasi serta bertukar data dengan aplikasi atau layanan lainnya secara terstruktur. Dalam arsitektur *Large Language Model* (LLM), API bertindak sebagai antarmuka krusial yang menghubungkan sistem lokal pengembang dengan infrastruktur model yang berada pada server penyedia layanan (*cloud-based model*).

Beberapa aspek strategis penggunaan API dalam proses *benchmarking* LLM pada penelitian ini meliputi:

1. **Interoperabilitas:** API memungkinkan sistem lokal yang menggunakan *framework* Promptfoo untuk mengirimkan instruksi dalam format standar seperti JSON dan menerima jawaban secara otomatis dari berbagai model yang berbeda secara simultan. Hal ini menjamin integrasi yang mulus antara instrumen pengujian dengan penyedia layanan AI.
2. **Kendali Parameter:** Melalui akses API, peneliti memiliki kontrol penuh terhadap parameter teknis model yang tidak tersedia secara mendalam pada antarmuka *chat* publik. Hal ini mencakup pengaturan temperature untuk menjaga konsistensi teks (deterministik), max_tokens untuk membatasi panjang jawaban, serta pemilihan versi spesifik model (GPT, Gemini, atau Claude) guna menjamin validitas data pengujian.
3. **Skalabilitas dan Otomatisasi:** Pemanfaatan API mendukung eksekusi pengujian secara massal (*batch testing*), di mana seluruh sampel soal yang terdapat dalam dataset mandiri dapat diproses secara otomatis tanpa memerlukan intervensi manual pada setiap interaksinya. Proses ini tidak hanya meningkatkan efisiensi waktu pengambilan data, tetapi juga meminimalkan risiko kesalahan manusia (*human error*) sehingga hasil perbandingan performa menjadi lebih objektif.

2.2.5 Konsep Token dalam Large Language Model

Dalam Large Language Model (LLM), teks tidak diproses secara langsung dalam bentuk kata atau kalimat utuh, melainkan dipecah menjadi unit-unit kecil yang disebut *token*. Token dapat berupa kata, bagian dari kata, atau kombinasi karakter tertentu, tergantung pada sistem tokenisasi yang digunakan oleh masing-masing model. Proses tokenisasi ini bertujuan untuk mengubah teks menjadi representasi numerik yang dapat diproses oleh model berbasis jaringan saraf.

Secara umum, terdapat dua jenis token yang dihitung dalam penggunaan LLM, yaitu input token dan output token. Input token merupakan jumlah token dari teks yang diberikan kepada model, sedangkan output token adalah jumlah token dari teks yang dihasilkan model sebagai jawaban. Total token yang diproses akan memengaruhi waktu komputasi serta biaya penggunaan model, terutama pada layanan berbasis API yang menghitung biaya berdasarkan jumlah token.

Jumlah token yang digunakan dalam suatu teks berpengaruh langsung terhadap beban komputasi model. Semakin banyak token yang diproses, semakin besar pula sumber daya yang diperlukan, baik dari sisi waktu maupun memori. Oleh karena itu, dalam penelitian ini, kecepatan pemrosesan token per detik digunakan sebagai salah satu indikator efisiensi operasional model.

Dalam konteks Bahasa Indonesia, jumlah token dapat dipengaruhi oleh struktur morfologi dan bentuk kata. Penggunaan imbuhan, kata majemuk, serta variasi struktur kalimat dapat menyebabkan teks terpecah menjadi lebih banyak token dibandingkan bahasa lain dalam kondisi tertentu. Selain itu, setiap model seperti GPT, Gemini, dan Claude memiliki mekanisme tokenisasi yang berbeda, sehingga jumlah token yang dihasilkan dari teks yang sama bisa bervariasi. Perbedaan ini dapat berdampak pada kecepatan inferensi dan konsumsi sumber daya model.

2.2.6 Model LLM yang Digunakan dalam Penelitian

Penelitian ini memfokuskan pengujian pada tiga arsitektur *Large Language Model* (LLM) utama yang saat ini mendominasi lanskap teknologi kecerdasan buatan generatif. Pemilihan ketiga model ini didasarkan pada perbedaan karakteristik pelatihan, arsitektur, dan tujuan pengembangan masing-masing model, yaitu:

1. GPT (OpenAI): Model ini dipilih karena reputasinya dalam kemampuan mengikuti instruksi (*instruction following*) dan penalaran logis yang kuat. Berdasarkan penelitian oleh (Zaremba & Demir, 2023), arsitektur GPT telah terbukti unggul dalam berbagai tugas pemrosesan bahasa alami (NLP) yang kompleks dan menjadi standar acuan bagi pengembangan LLM lainnya.
2. Gemini (Google): Sebagai model multimodal yang dikembangkan oleh Google DeepMind, Gemini menawarkan keunggulan dalam integrasi data yang luas dan pemahaman konteks yang dalam. (Alhur, 2024) mencatat bahwa Gemini dirancang untuk efisiensi tinggi dalam menangani berbagai modalitas input, menjadikannya kompetitor yang relevan untuk diuji dalam kemampuannya memproses bahasa Indonesia yang bersifat kontekstual.
3. Claude (Anthropic): Pemilihan Claude didasarkan pada pendekatan *Constitutional AI* yang mengedepankan keamanan (*safety*) dan konsistensi respons. Penelitian oleh (Sparkman & Witt, 2025) menunjukkan bahwa Claude memiliki kecenderungan halusinasi yang lebih rendah dalam sintesis literatur dibandingkan model lain, sehingga sangat menarik untuk diuji apakah keandalan ini juga tercermin saat memproses instruksi dalam Bahasa Indonesia.

Ketiga model tersebut akan diakses melalui *Application Programming Interface* (API) untuk memastikan bahwa versi yang diuji adalah versi *state-of-the-art* yang memiliki parameter performa paling optimal saat ini.

2.2.7 Benchmarking pada Large Language Model (LLM)

Benchmarking merupakan proses evaluasi sistematis untuk mengukur dan membandingkan performa beberapa model bahasa (LLM) menggunakan standar pengujian yang identik. Pendekatan ini memungkinkan peneliti memperoleh gambaran komparatif yang objektif mengenai kualitas respons, efisiensi komputasi, dan konsistensi keluaran model. Melalui proses ini, setiap model diberikan *input* yang persis sama, sehingga perbedaan performa yang dihasilkan dapat dipastikan berasal dari kemampuan intrinsik masing-masing model, bukan dari variasi data pengujian.

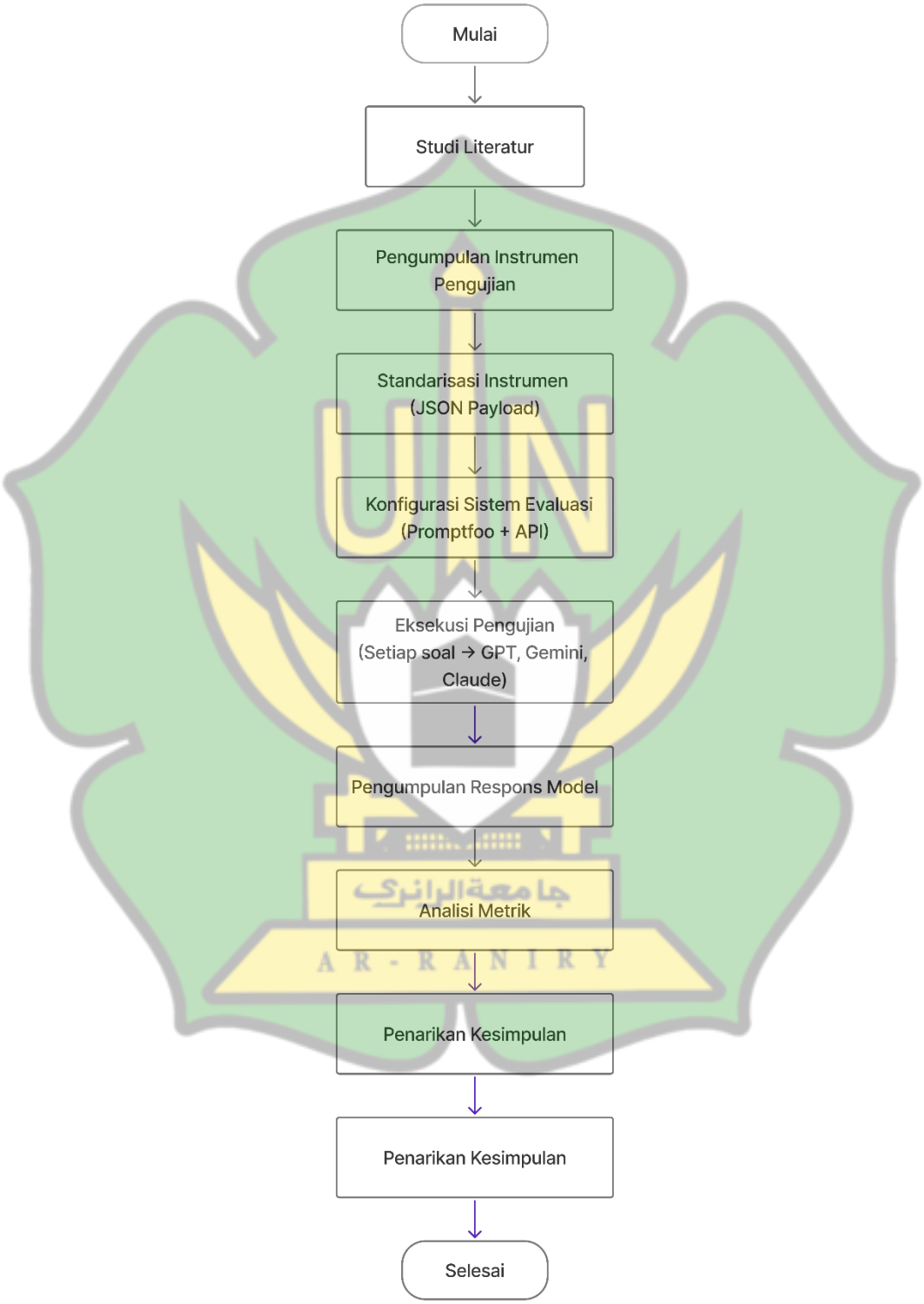
Pelaksanaan *benchmarking* pada *Natural Language Processing* (NLP) secara umum memerlukan integrasi empat komponen utama: instrumen pengujian yang divalidasi, model yang menjadi objek uji, metrik evaluasi kuantitatif maupun kualitatif, serta *framework* evaluasi otomatis. Dalam penelitian ini, proses *benchmarking* diimplementasikan menggunakan *framework* Promptfoo untuk mengevaluasi model GPT, Gemini, dan Claude secara simultan melalui akses *Application Programming Interface* (API). *Framework* ini bertugas mendistribusikan instruksi berbahasa Indonesia yang sama kepada ketiga model, untuk kemudian secara otomatis merekam hasil metrik seperti akurasi jawaban, latensi respons, dan estimasi biaya token.

Proses *benchmarking* menjadi semakin kompleks ketika diterapkan pada bahasa yang kurang terwakili (*underrepresented languages*) seperti Bahasa Indonesia. Studi terbaru menunjukkan bahwa kemampuan LLM dalam mematuhi instruksi sangat sensitif terhadap bahasa yang digunakan dalam *prompt*. Beberapa model sering kali mengalami penurunan performa yang signifikan ketika instruksi diterjemahkan dari bahasa Inggris ke bahasa Indonesia, sementara model lainnya menunjukkan ketahanan generalisasi yang lebih baik. Fenomena ini menegaskan bahwa *benchmarking* menggunakan instruksi Bahasa Indonesia autentik sangat diperlukan untuk mengidentifikasi kemampuan *cross-lingual transfer* dari model, sehingga

dapat ditentukan layanan AI generatif mana yang paling optimal untuk mendukung kegiatan akademik.

BAB III
METODOLOGI PENELITIAN

3.1 Alur Penelitian



Gambar 3. 1 Alur Penelitian

Diagram alir penelitian pada gambar 3.1 merupakan visualisasi dari seluruh alur kerja sistematis yang dimulai dengan tahapan Studi Literatur untuk mendalami teori model bahasa, dilanjutkan dengan Pengumpulan Instrumen Pengujian secara mandiri dari instrumen pertanyaan resmi yang tervalidasi serta Persiapan API untuk mengakses model GPT, Gemini, dan Claude.

Tahapan teknis dimulai dengan Konfigurasi Framework menggunakan Promptfoo untuk mengatur parameter pengujian yang kemudian dijalankan pada tahap Eksekusi Pengujian guna mengambil respons model secara otomatis. Proses tersebut kemudian masuk ke dalam Validasi Data, di mana jika ditemukan galat atau data tidak lengkap (*NO*), maka sistem akan kembali ke tahap konfigurasi untuk perbaikan, namun jika data valid (*YES*), maka penelitian dilanjutkan ke Analisis Metrik untuk mengevaluasi akurasi, latensi, serta biaya sebelum akhirnya dilakukan Penarikan Kesimpulan guna menentukan model yang paling optimal bagi mahasiswa dalam memahami Bahasa Indonesia.

3.2 Jenis dan Pendekatan Penelitian

Penelitian ini menggunakan jenis penelitian kuantitatif eksperimental dengan metode analisis komparatif. Pemilihan metode ini bertujuan untuk mengukur secara empiris performa tiga *Large Language Model* (LLM) melalui metrik yang objektif dan terstandarisasi. Pendekatan eksperimental dilakukan dengan memberikan stimulus berupa *prompt* atau instruksi yang identik kepada setiap model untuk kemudian diukur luarannya menggunakan parameter yang telah ditentukan.

Pendekatan penelitian ini dibagi menjadi dua aspek utama:

1. Pendekatan Kuantitatif

   Digunakan untuk mengumpulkan dan menganalisis data numerik seperti tingkat akurasi jawaban (*pass rate*), kecepatan pemrosesan (*tokens per second*), latensi respons, dan biaya operasional penggunaan API. Penggunaan metrik kuantitatif seperti *semantic similarity* dan skor akurasi sangat penting karena evaluasi LLM tidak cukup hanya dilakukan dengan melihat benar atau salahnya jawaban secara manual.

2. Pendekatan Komparatif

   Digunakan untuk membandingkan kapabilitas antara GPT, Gemini, dan Claude dalam menangani tugas-tugas *Natural Language Processing* (NLP) spesifik. Metode komparatif ini memungkinkan peneliti untuk mengidentifikasi kelebihan dan kekurangan masing-masing model pada skenario tertentu, mengingat tidak ada satu model pun yang unggul secara konsisten di seluruh jenis tugas.

Melalui pendekatan ini, penelitian diharapkan dapat memberikan gambaran yang sistematis mengenai efektivitas model-model tersebut dalam memahami karakteristik linguistik Bahasa Indonesia yang memiliki keunikan pada sistem afiksasi serta fleksibilitas struktur kalimat.

3.3 Objek Penelitian

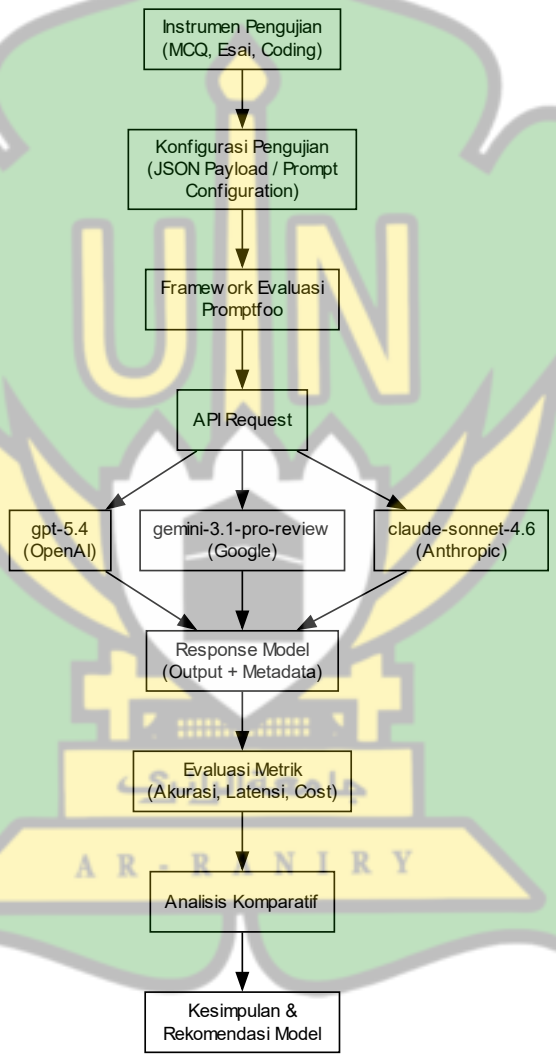
Objek dalam penelitian ini adalah tiga model bahasa besar (*Large Language Models*) generatif yang memiliki arsitektur *state-of-the-art* dan aksesibilitas melalui *Application Programming Interface* (API). Penentuan ketiga model ini didasarkan pada dominasi pasar dan perbedaan pendekatan teknis dalam pengembangannya. Adapun objek penelitian tersebut adalah:

1. GPT (OpenAI): Model ini mewakili produk dari OpenAI yang dikenal memiliki kemampuan penalaran (*reasoning*) dan pemahaman instruksi yang sangat tinggi. Dalam berbagai studi komparatif, arsitektur GPT sering kali dijadikan sebagai standar acuan (*gold standard*) dalam evaluasi tugas-tugas NLP (Zaremba & Demir, 2023).
2. Gemini (Google): Model ini merupakan teknologi *multimodal* terbaru dari Google DeepMind yang dirancang untuk menangani berbagai modalitas input secara terintegrasi. Gemini dievaluasi dalam penelitian ini karena keunggulannya dalam memahami konteks informasi yang luas dan ketersediaan data pelatihan yang sangat masif (Alhur, 2024).
3. Claude (Anthropic): Model ini dikembangkan dengan prinsip *Constitutional AI* yang berfokus pada keamanan dan minimalisasi risiko respons yang tidak akurat. Claude dipilih sebagai objek penelitian karena karakteristiknya yang cenderung memberikan jawaban lebih lengkap dan

tingkat halusinasi yang relatif lebih rendah pada tugas-tugas sintesis informasi (Uppalapati & Nag, 2024; Sparkman & Witt, 2025).

Ketiga model tersebut akan diuji menggunakan versi terbaru yang tersedia saat pengujian berlangsung (seperti GPT, Gemini, dan Claude) untuk memastikan relevansi hasil penelitian dengan perkembangan teknologi terkini.

3.4 Arsitektur Sistem Penelitian



Gambar 3. 2 Arsitektur Sistem Pengujian

Arsitektur sistem penelitian ini dirancang untuk melakukan proses *benchmarking* terhadap beberapa *Large Language Model* (LLM) secara otomatis, terstandarisasi, dan objektif. Sistem bekerja dengan memanfaatkan instrumen

pengujian yang didistribusikan kepada setiap model melalui *framework* evaluasi berbasis API, di mana kerangka kerja NestJS bertindak sebagai orkestrator *backend* utama.

Proses penelitian dimulai dengan penyusunan instrumen pengujian yang terdiri dari beberapa jenis tugas, seperti soal pilihan ganda (MCQ), pertanyaan esai, serta instruksi pemrograman. Seluruh instrumen tersebut kemudian dikonfigurasi ke dalam format *JSON payload* yang dikelola oleh NestJS untuk didistribusikan ke *framework* Promptfoo dalam menjalankan proses pengujian secara otomatis.

Dalam tahapan eksekusi, *framework* Promptfoo berfungsi mengatur pengiriman instruksi kepada model GPT, Gemini, dan Claude melalui *Application Programming Interface* (API) dari masing-masing penyedia layanan. Instrumen pengujian diberikan secara identik kepada ketiga model tersebut, sehingga satu instruksi akan memproduksi tiga respons model yang berbeda untuk nantinya dibandingkan secara komparatif.

Fungsi *Masking* dan Anonimisasi oleh NestJS Untuk tipe soal yang bersifat terbuka (seperti esai), arsitektur ini menerapkan pendekatan evaluasi *LLM-as-a-Judge*. Pada titik inilah peran krusial NestJS diimplementasikan. Sebelum NestJS mengirimkan kembali *JSON payload* yang berisi respons model ke Promptfoo untuk dievaluasi oleh Juri AI, sistem melakukan fungsi *masking* atau anonimisasi label model.

Algoritma pada NestJS secara otomatis menyensor atau mengubah atribut identitas model asal (misalnya, menghapus label openai:gpt-5.4 atau google:gemini-3.1 dan menggantinya dengan label netral seperti Model X, Model Y, dan Model Z). Melalui proses anonimisasi *payload* ini, Juri AI menerima input secara "buta" (*blind review*). Pendekatan arsitektural ini memastikan bahwa Juri AI sama sekali tidak mengetahui model mana yang menghasilkan teks tersebut, sehingga secara efektif mencegah *egocentric bias* (*same-model bias*) dan memaksa Juri AI untuk menilai kemurnian kualitas jawaban semata-mata berdasarkan rubrik penilaian yang disediakan.

3.5 Instrumen Pengujian

Penelitian ini tidak menggunakan dataset publik berskala masif, melainkan menggunakan Instrumen Pengujian Autentik Tertutup (*Unpublished Authentic Instruments*).

Total instrumen uji terdiri dari 28 butir pertanyaan: 20 soal Pilihan Ganda (MCQ) untuk menguji akurasi logika dasar, 5 soal esai untuk mengevaluasi pemahaman konteks, dan 3 instruksi pemrograman (*coding*) untuk penalaran teknis. Seluruh instrumen bersumber dari arsip ujian internal lokal (seperti soal UTS/UAS sekolah) yang belum pernah dipublikasikan di internet. Pendekatan ini dilakukan untuk menguji kemampuan penalaran murni (*Zero-Shot Reasoning*) dari model, bukan sekadar kemampuan mengingat data latih (*training data*).

Seluruh instrumen pengujian yang telah dikurasi kemudian diberikan kepada ketiga model LLM yang diuji secara identik, sehingga setiap instrumen menghasilkan respons dari masing-masing model yang selanjutnya digunakan dalam proses evaluasi komparatif.

Untuk mengevaluasi instrumen yang bersifat terbuka (Esai dan *Coding*), penelitian ini tidak menggunakan pencocokan teks presisi (*exact match*), melainkan mengimplementasikan Rubrik Penilaian Spesifik Berbasis Tugas (*Task-Specific Rubrics*). Rubrik ini ditransformasikan ke dalam parameter rubric pada arsitektur pengujian JSON dan berfungsi sebagai parameter standar atau *ground truth* bagi *LLM-as-a-Judge* dalam framework Promptfoo.

Pendekatan ini memastikan bahwa setiap respons model tidak hanya dinilai secara biner (benar/salah), tetapi juga dievaluasi secara kualitatif terhadap kelengkapan faktual dan kebenaran logika sebelum dikonversi menjadi skor kuantitatif (rentang 0.0 hingga 1.0). Berikut adalah rincian instrumen dan rubrik yang diterapkan pada masing-masing tipe pengujian:

a. Instrumen dan Rubrik Soal Esai

   Pada pengujian esai, rubrik difokuskan pada kelengkapan identifikasi masalah dan penjelasan faktual. Tabel 3.1 menunjukkan salah satu sampel instrumen pengujian esai beserta kriteria penilaian yang disematkan pada sistem.

Tabel 3. 1 Contoh Instrumen dan Rubrik Penilaian Soal Esai

Komponen	**Deskripsi**
Instruksi (Prompt)	Pembakaran bahan bakar minyak dapat menimbulkan beberapa dampak, diantaranya terhadap kesehatan dan lingkungan. Jelaskan dampak yang ditimbulkan oleh pembakaran bahan bakar terhadap lingkungan!
Kriteria Penilaian	Kemampuan model dalam mengidentifikasi dan menjelaskan minimal dua dampak lingkungan secara komprehensif.
Rubrik (*Automated Grading*)	**Skor 1.0**: Menjelaskan minimal dua dampak lingkungan seperti pencemaran udara, efek rumah kaca, hujan asam, atau pemanasan global.
	Skor 0.5: Hanya menyebutkan satu dampak lingkungan tanpa penjelasan lengkap.
	Skor 0.0: Tidak menyebutkan dampak lingkungan yang relevan atau melenceng dari konteks.

b. Instrumen dan Rubrik Soal Pemrograman (*Coding*)

Pada pengujian pemrograman, rubrik tidak menilai kesamaan sintaks kode, melainkan berfokus pada kebenaran logika algoritma dan implementasi kondisi (seperti iterasi, kalkulasi akumulatif, atau

manipulasi *string*). Tabel 3.2 menunjukkan salah satu sampel instrumen pengujian *coding*.

Tabel 3. 2 Contoh Instrumen dan Rubrik Penilaian Soal Coding

Komponen	**Deskripsi**
Instruksi (Prompt)	Buatlah fungsi find_pairs(nums, target) yang mencari semua pasangan unik angka dalam list nums yang jika dijumlahkan menghasilkan nilai target. Output harus berupa list dari tuple. Contoh: nums=[1, 2, 3, 4, 5], target=5 -> [(1, 4), (2, 3)].
Kriteria Penilaian	Ketepatan logika algoritma pencarian pasangan angka dan penanganan duplikasi elemen (uniqueness) menggunakan struktur data *list* dan *tuple* sesuai nilai target yang ditentukan.
Rubrik (*Automated Grading*)	**Skor 1.0**: Jika fungsi mengembalikan list tuple pasangan yang benar dan unik.
	Skor 0.5: Jika ada pasangan duplikat (misal (1,4) dan (4,1) muncul keduanya).
	Skor 0.0: Jika logika penjumlahan salah.

Dalam implementasinya, setiap respons yang dihasilkan oleh model yang diuji (GPT, Gemini, Claude) akan dibaca oleh model evaluator. Evaluator tersebut wajib menghasilkan dua keluaran: score berupa nilai numerik sesuai ketetapan rubrik, dan reason yang berisi argumentasi logis mengapa nilai tersebut diberikan. Skor dari seluruh evaluator kemudian diagregasi oleh sistem untuk menghasilkan metrik *Average Score* bagi setiap model yang diuji.

3.5.1 Proses Kurasi dan Mitigasi Kebocoran Data

Proses kurasi instrumen dilakukan melalui tahapan berlapis untuk menjamin validitas pengujian dan mencegah bias evaluasi:

1. **Mitigasi Kebocoran Data (*Data Leakage*):** Penulis sengaja tidak menggunakan soal dari bank soal publik, buku cetak nasional, atau repositori *online* (seperti LeetCode atau forum edukasi). Penggunaan soal autentik lokal menjamin bahwa instruksi yang diberikan belum pernah "dibaca" atau direkam oleh model GPT, Gemini, maupun Claude pada fase pelatihannya.
2. **Standardisasi Format (*JSON Payload*):** Seluruh soal yang dikurasi ditransformasikan ke dalam struktur JSON agar dapat dieksekusi secara otomatis oleh sistem. Setiap soal dipetakan ke dalam parameter spesifik seperti testId, type (MCQ/Esai), dan expectedAnswer (untuk MCQ) atau rubric (untuk esai).
3. **Validasi Ahli untuk Pemrograman:** Khusus untuk 3 kasus *coding*, keluaran model tidak hanya diukur oleh sistem, tetapi juga divalidasi silang oleh ahli (praktisi/akademisi) untuk menilai efisiensi dan kebersihan kode (*clean code*).

Seluruh instrumen pengujian yang telah dikurasi kemudian digunakan sebagai input yang sama bagi setiap model yang diuji. Pendekatan ini memungkinkan proses evaluasi komparatif dilakukan secara adil, karena setiap model menerima pertanyaan yang identik dalam proses pengujian.

3.5.2 Validasi Ahli (Expert Judgment)

Pada penelitian ini, evaluasi terhadap keluaran model pada instrumen pemrograman (coding) tidak hanya dilakukan menggunakan pengujian otomatis, tetapi juga melalui proses validasi ahli (*expert judgment*). Pendekatan ini digunakan karena kualitas solusi pemrograman tidak hanya ditentukan oleh keberhasilan menghasilkan keluaran yang benar, tetapi juga dipengaruhi oleh aspek-aspek lain seperti struktur kode, efisiensi algoritma, keterbacaan (*readability*), serta penerapan praktik pemrograman yang baik (*clean code*).

Validasi ahli dilakukan terhadap seluruh respons yang dihasilkan oleh model GPT, Gemini, dan Claude pada tiga instrumen pengujian pemrograman. sPenelitian ini melibatkan tiga orang ahli yang memiliki kompetensi pada bidang pemrograman dan pengembangan perangkat lunak. Penggunaan penilaian ahli manusia ini berfungsi sebagai mekanisme kontrol untuk memvalidasi luaran *machine learning*, mengingat evaluasi dari mesin dan manusia bersifat komplementer dan saling melengkapi (Gencoglu et al., 2023). Karena metode otomatis tidak dapat sepenuhnya menggantikan evaluasi manusia, keterlibatan ahli tetap sangat krusial (Gencoglu et al., 2023). Dengan demikian, penggunaan tiga ahli dalam penelitian ini dinilai memadai untuk menguji validitas konten dan memperoleh penilaian yang objektif serta dapat dipertanggungjawabkan secara akademik (Jeldres et al., 2023).

Setiap ahli diminta untuk melakukan penilaian terhadap ketiga respons model pada setiap instrumen coding dengan memberikan peringkat (*ranking*) berdasarkan kualitas keseluruhan solusi. Penilaian dilakukan dengan mempertimbangkan beberapa aspek, yaitu:

1. Kebenaran logika program.
2. Kelengkapan solusi terhadap seluruh kebutuhan soal.
3. Keterbacaan dan struktur kode.
4. Efisiensi penyelesaian masalah.
5. Kesesuaian dengan praktik pemrograman yang baik (*clean code*).

Hasil penilaian dari masing-masing ahli kemudian direkapitulasi dan dianalisis untuk menentukan model yang memperoleh peringkat terbaik pada setiap instrumen pemrograman. Selain itu, komentar dan masukan yang diberikan oleh ahli digunakan sebagai data pendukung dalam proses interpretasi hasil penelitian.

3.6 Prosedur Penelitian

Prosedur penelitian ini dirancang secara sistematis melalui dua fase utama untuk memastikan evaluasi terhadap GPT, Gemini, dan Claude berjalan efektif dan tepat waktu. Fase pertama dilaksanakan pada periode Desember 2025 hingga Februari 2026, yang secara khusus difokuskan pada persiapan teoretis dan perencanaan. Kegiatan utama pada fase ini mencakup pelaksanaan studi literatur

secara komprehensif serta penyusunan draf awal skripsi yang meliputi Bab 1 hingga Bab 3.

Fase kedua dilaksanakan pada periode Mei hingga Juli 2026, yang mencakup tahap eksekusi teknis hingga penyusunan pelaporan akhir. Kegiatan pada fase ini diawali dengan persiapan dan kurasi instrumen soal, dilanjutkan dengan konfigurasi sistem pada Promptfoo dan integrasi API. Setelah lingkungan pengujian siap, peneliti mengeksekusi pengujian model AI untuk mengambil data, menganalisis hasil perbandingan dengan melibatkan penilaian ahli (*expert judgment*), dan diakhiri dengan penyusunan laporan akhir skripsi (Bab 4 dan Bab 5) sebagai kesimpulan keseluruhan penelitian.

3.7 Alat dan Bahan Penelitian

Penelitian ini memanfaatkan kombinasi perangkat keras dan perangkat lunak untuk mendukung pelaksanaan pengujian otomatis terhadap model Large Language Model (LLM). Penggunaan perangkat-perangkat tersebut bertujuan untuk memastikan bahwa proses pengambilan data dapat dilakukan secara sistematis, konsisten, dan objektif selama proses evaluasi model. Selain itu, penggunaan sistem evaluasi otomatis juga membantu meminimalkan potensi kesalahan manusia dalam pencatatan dan pengukuran metrik performa model. Rincian perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini disajikan pada Tabel 3.3:

Tabel 3. 3 Daftar Perangkat Keras dan Lunak

Kategori	Nama Alat	Fungsi Utama
Perangkat Keras	Laptop / Unit Komputer	Stasiun kerja utama untuk konfigurasi sistem dan eksekusi pengujian.
	Koneksi Internet	Akses *real-time* ke server API masing-masing model LLM.
Perangkat Lunak	Promptfoo	Framework evaluasi otomatis yang digunakan untuk menjalankan benchmarking terhadap model LLM.

	API Model (OpenAI, Google AI, Anthropic)	Digunakan untuk mengakses model GPT, Gemini, dan Claude melalui API.
	Node.js Runtime	Lingkungan eksekusi untuk menjalankan framework Promptfoo.
	NestJS Framework	Kerangka kerja backend berbasis Node.js yang digunakan untuk membangun API automasi, mengelola *JSON payload*, dan mengorkestrasi eksekusi pengujian.

Selain perangkat teknis, penelitian ini juga memanfaatkan beberapa bahan penelitian yang digunakan dalam proses evaluasi terhadap model Large Language Model (LLM). Bahan penelitian tersebut meliputi model LLM yang menjadi objek pengujian, instrumen pengujian yang disusun secara mandiri, serta parameter evaluasi yang digunakan untuk mengukur performa model. Instrumen pengujian terdiri dari berbagai jenis pertanyaan seperti soal pilihan ganda, esai, dan tugas pemrograman yang dirancang untuk merepresentasikan berbagai skenario penggunaan akademik berbahasa Indonesia. Rincian bahan penelitian yang digunakan dalam penelitian ini disajikan pada Tabel 3.4:

Tabel 3. 4 Bahan Penelitian

Komponen	Sumber / Deskripsi	Fungsi dalam Penelitian
Model LLM	GPT (OpenAI), Gemini (Google), Claude (Anthropic)	Objek penelitian yang dibandingkan performanya dalam memahami instruksi berbahasa Indonesia.
Instrumen Pengujian	Kumpulan soal *Multiple Choice* (MCQ), Esai, dan *Coding* autentik yang dikurasi secara mandiri.	Bertindak sebagai stimulus atau *prompt* masukan (input) untuk menguji kemampuan pemahaman dan penalaran model.
Rubrik Penilaian (*Task-Specific Rubrics*)	Standar kriteria penjurian yang disusun secara mandiri untuk soal Esai dan *Coding*.	Menjadi *ground truth* atau pedoman objektif bagi *LLM-as-a-Judge* dalam memberikan skor kuantitatif.
Parameter Evaluasi	Metrik *Pass Rate*, *Average Score*, *Average Latency* (waktu respons), dan *Total Cost* (estimasi biaya token).	Digunakan sebagai indikator numerik akhir untuk membandingkan kualitas dan efisiensi performa antar model.

3.8 Tahapan Penelitian

Penelitian ini menggunakan pendekatan benchmarking untuk membandingkan performa beberapa Large Language Model (LLM) dalam memahami instruksi berbahasa Indonesia. Proses penelitian dimulai dengan penyusunan instrumen pengujian yang terdiri dari soal pilihan ganda, pertanyaan esai, serta tugas pemrograman yang relevan dengan kebutuhan akademik mahasiswa. Instrumen pengujian tersebut kemudian dikonfigurasi ke dalam format JSON dan dijalankan menggunakan framework evaluasi otomatis Promptfoo. Dalam proses pengujian, setiap instrumen diberikan secara identik kepada tiga model LLM yang diuji, yaitu GPT, Gemini, dan Claude melalui akses Application Programming Interface (API). Setiap model menghasilkan respons yang kemudian direkam beserta metadata teknis seperti latensi respons dan penggunaan token. Data hasil pengujian selanjutnya divalidasi untuk memastikan kelengkapan respons,

kemudian dianalisis menggunakan beberapa metrik evaluasi seperti akurasi jawaban, latensi inferensi, dan estimasi biaya penggunaan token. Hasil analisis tersebut digunakan untuk membandingkan performa masing-masing model secara komparatif dan menentukan model yang paling optimal dalam memahami Bahasa Indonesia.

3.8.1 Studi Literatur

Langkah awal dimulai dengan melakukan penelusuran mendalam terhadap berbagai referensi ilmiah, jurnal, dan dokumentasi teknis terkait perkembangan *Large Language Models* (LLM). Penulis memfokuskan kajian pada model spesifik seperti GPT, Gemini, dan Claude, serta mempelajari metodologi evaluasi bahasa alami (NLP) yang relevan dengan karakteristik Bahasa Indonesia agar memiliki landasan teoritis yang kuat sebelum melakukan pengujian.

3.8.2 Pengumpulan Instrumen Pengujian

Pada tahap ini, penulis menghimpun instrumen uji secara mandiri yang dirancang untuk mewakili berbagai skenario akademik (MCQ, Esai, dan *Coding*). Berbeda dengan pengumpulan data konvensional yang mengambil dari bank soal publik di internet, penulis secara khusus mengumpulkan arsip soal autentik tertutup dari lingkungan akademik lokal (seperti soal ujian internal kampus atau sekolah yang tidak dipublikasikan secara *online*). Pemilihan sumber instrumen ini merupakan langkah krusial untuk memitigasi risiko kebocoran data (*data leakage/contamination*), sehingga memastikan bahwa model LLM yang diuji benar-benar menggunakan kemampuan penalaran murni (*zero-shot reasoning*), bukan sekadar memanggil ingatan dari data latih mereka.

3.8.3 Standarisasi Instrumen (JSON Payload)

Pada tahap ini, penulis merancang dan menyusun arsitektur struktur data (*payload*) pengujian ke dalam format JSON (*JavaScript Object Notation*) yang bersifat deterministik, terstruktur, dan memiliki tingkat replikasi tinggi. JSON *Payload* ini berfungsi sebagai skrip orkestrasi utama yang menjembatani instrumen

pengujian dengan *framework* Promptfoo, sehingga seluruh alur *benchmarking* dapat berjalan secara otomatis dan meminimalkan subjektivitas penilaian manusia. Optimasi dilakukan dengan menstandarkan penamaan atribut (*field*) dan membagi *payload* JSON ke dalam dua fase utama, yaitu Fase Konfigurasi (*Input*) dan Fase Perekaman (*Output*):

1. Fase Konfigurasi (*Input Parameters*)

Struktur input didefinisikan secara spesifik untuk memetakan tugas ke masing-masing model. Atribut utama dalam fase ini meliputi:

a. Providers: Mendefinisikan *endpoint* dari model LLM utama yang diuji (GPT, Gemini, dan Claude).
b. JudgeProviders: Menetapkan model LLM yang bertugas sebagai evaluator (*LLM-as-a-Judge*) untuk menilai koherensi dan kualitas luaran pada tipe soal esai secara otomatis.
c. Tests: Memuat matriks instrumen pengujian yang mencakup atribut pengenal (testId), jenis soal (type), instruksi berbahasa Indonesia (question), serta kunci acuan berupa expectedAnswer (untuk validasi pilihan ganda) atau rubric (untuk parameter penjurian esai).

2. Fase Perekaman dan Agregasi (*Output Results & Summary*)

Setelah eksekusi API selesai, JSON *Payload* dirancang untuk secara otomatis memproduksi dan menyimpan hasil luaran ke dalam format objek bersarang (*nested objects*). Atribut perekaman ini meliputi:

a. Results: Merekam jejak respons mentah dari setiap model secara individual, yang mencakup teks jawaban (output), status keberhasilan (pass), skor (score), argumentasi penjurian (gradingReason), dan metrik teknis seperti latensi (latencyMs).
b. Summary: Melakukan proses agregasi data, yaitu menggabungkan dan merangkum seluruh hasil pengujian dari setiap instrumen menjadi laporan statistik komprehensif bagi masing-masing provider. Bagian ini memuat berbagai metrik ringkasan seperti akumulasi Pass Rate, rata-rata skor (*Average Score*), latensi rata-rata (*Average Latency*), serta estimasi total

biaya operasional (*Total Cost*). Selain itu, bagian ini juga menampilkan klasifikasi model paling optimal melalui atribut *winners*, yang mencakup kategori *mostEfficient*, *fastest*, dan *highestQuality*.

Melalui penyusunan arsitektur JSON yang komprehensif ini, data pengujian menjadi sangat terpusat, mudah dianalisis secara komparatif, dan siap untuk diintegrasikan secara langsung ke dalam sistem atau *dashboard* melalui API berbasis NestJS yang digunakan dalam penelitian.

3.8.4 Konfigurasi Sistem Evaluasi (Promptfoo + API)

Pada tahap ini, penulis melakukan konfigurasi sistem evaluasi yang digunakan untuk menjalankan proses benchmarking terhadap model *Large Language Model* (LLM). Konfigurasi sistem dilakukan menggunakan framework Promptfoo yang terintegrasi dengan *Application Programming Interface* (API) dari masing-masing penyedia layanan model, yaitu OpenAI untuk GPT, Google untuk Gemini, dan Anthropic untuk Claude.

Konfigurasi pengujian disusun dalam bentuk struktur data JSON yang berfungsi sebagai file konfigurasi utama dalam proses benchmarking. Struktur ini memuat informasi mengenai model yang diuji, instrumen pengujian, parameter evaluasi, serta mekanisme pencatatan hasil respons model. File konfigurasi ini kemudian dijalankan oleh Promptfoo untuk mengotomatisasi proses pengujian terhadap seluruh model yang terlibat.

Dalam konfigurasi tersebut, beberapa komponen utama yang didefinisikan meliputi:

1. Batch Pengujian (batchName)

Atribut ini digunakan untuk mengidentifikasi kelompok pengujian yang sedang dijalankan. Setiap batch berisi sekumpulan instrumen pengujian yang akan diberikan kepada seluruh model secara bersamaan. Penamaan batch memudahkan proses pencatatan dan pengelompokan hasil evaluasi pada tahap analisis.

2. Model yang Diuji (providers)

Bagian ini mendefinisikan model-model Large Language Model yang menjadi objek pengujian. Dalam penelitian ini, model yang diuji terdiri dari

tiga model utama, yaitu GPT, Gemini, dan Claude. Ketiga model tersebut diakses melalui endpoint API masing-masing penyedia layanan sehingga sistem dapat mengirimkan instruksi pengujian secara otomatis kepada setiap model.

3. Model Evaluator (judgeProvider)

Selain model utama yang diuji, konfigurasi juga mendefinisikan model evaluator yang berfungsi sebagai *LLM-as-a-Judge*. Model evaluator ini digunakan untuk menilai kualitas jawaban pada tipe soal esai berdasarkan rubrik penilaian yang telah ditentukan. Dengan pendekatan ini, evaluasi terhadap jawaban esai dapat dilakukan secara otomatis dan konsisten tanpa memerlukan penilaian manual.

4. Parameter Estimasi Biaya (providerPrices)

Bagian ini digunakan untuk mendefinisikan estimasi biaya penggunaan API dari masing-masing model berdasarkan jumlah token yang diproses. Parameter ini mencakup harga token input dan token output yang dinyatakan dalam satuan biaya per satu juta token (MTok). Informasi ini memungkinkan sistem menghitung estimasi biaya operasional selama proses pengujian berlangsung.

5. Instrumen Pengujian (tests)

Komponen ini berisi kumpulan instrumen pengujian yang diberikan kepada model. Setiap instrumen memiliki atribut yang mendefinisikan identitas soal, jenis soal, serta parameter evaluasi. Atribut tersebut meliputi:

a) *id*, sebagai identitas unik setiap instrumen pengujian
b) *type*, yang menunjukkan jenis soal (misalnya pilihan ganda atau esai)
c) *question*, yang berisi instruksi atau pertanyaan yang diberikan kepada model
d) *expectedAnswer*, yang digunakan untuk memvalidasi jawaban pada soal pilihan ganda
e) *rubric*, yang digunakan sebagai pedoman penilaian pada soal esai.

Melalui konfigurasi ini, seluruh instrumen pengujian dapat dikirim secara otomatis kepada model GPT, Gemini, dan Claude melalui API masing-masing penyedia layanan. Setiap model kemudian menghasilkan respons terhadap

instrumen yang sama sehingga memungkinkan proses evaluasi komparatif dilakukan secara objektif.

Seluruh respons yang dihasilkan oleh model kemudian direkam oleh sistem beserta metadata teknis seperti waktu respons (latency), jumlah token yang digunakan, skor evaluasi, serta status keberhasilan jawaban. Data tersebut selanjutnya digunakan pada tahap analisis metrik untuk menentukan performa masing-masing model dalam memahami instruksi berbahasa Indonesia.

Dengan adanya konfigurasi sistem evaluasi ini, proses benchmarking dapat dilakukan secara terstandarisasi, otomatis, dan konsisten, sehingga hasil perbandingan antar model dapat dianalisis secara lebih objektif pada tahap penelitian berikutnya

3.8.5 Eksekusi Pengujian

Tahap eksekusi pengujian merupakan fase inti dalam penelitian ini, di mana seluruh instrumen pengujian dijalankan secara otomatis terhadap model *Large Language Model* (LLM) yang menjadi objek penelitian. Pada tahap ini, *framework* Promptfoo digunakan untuk mengirimkan setiap instrumen pengujian kepada model GPT, Gemini, dan Claude melalui *Application Programming Interface* (API) dari masing-masing penyedia layanan.

Proses pengujian dilakukan secara terstandarisasi dengan menggunakan konfigurasi JSON yang telah disusun pada tahap sebelumnya. Melalui konfigurasi tersebut, setiap instrumen pengujian yang terdapat dalam sistem dikirimkan secara identik kepada ketiga model yang diuji. Dengan pendekatan ini, satu instrumen pengujian akan menghasilkan tiga respons yang berbeda, masing-masing berasal dari model GPT, Gemini, dan Claude. Pendekatan ini sangat penting dalam metode *benchmarking* karena memastikan bahwa seluruh model menerima pertanyaan yang sama, sehingga hasil respons yang dihasilkan dapat dibandingkan secara objektif.

Selama proses eksekusi berlangsung, Promptfoo secara otomatis mengelola pengiriman instruksi, menerima respons dari model, serta mencatat berbagai metadata teknis yang dihasilkan selama proses inferensi. Metadata tersebut meliputi teks jawaban model, status keberhasilan jawaban (*pass*), skor evaluasi, waktu

respons (*latency*), serta jumlah token yang digunakan dalam proses pemrosesan instruksi.

Selain itu, pada tipe instrumen terbuka seperti soal esai dan tugas pemrograman (*coding*), sistem juga menjalankan proses evaluasi otomatis menggunakan pendekatan *LLM-as-a-Judge*, di mana model evaluator menilai kualitas jawaban berdasarkan rubrik penilaian yang telah ditentukan sebelumnya. Hasil penilaian tersebut kemudian direkam dalam bentuk skor evaluasi yang dapat dianalisis secara kuantitatif pada tahap selanjutnya.

Seluruh proses pengujian dilakukan secara otomatis dan simultan oleh sistem sehingga dapat meminimalkan kesalahan manusia (*human error*) dalam proses pengambilan data. Dengan mekanisme ini, data respons dari setiap model dapat dikumpulkan secara konsisten dan siap digunakan pada tahap analisis metrik untuk mengevaluasi performa masing-masing model dalam memahami instruksi berbahasa Indonesia.

3.8.6 Pengumpulan Respons Model

Setelah proses eksekusi pengujian dilakukan, setiap model Large Language Model (LLM) menghasilkan respons terhadap instrumen pengujian yang diberikan. Pada tahap ini, seluruh respons yang dihasilkan oleh model GPT, Gemini, dan Claude dikumpulkan secara otomatis oleh sistem evaluasi yang dijalankan melalui framework Promptfoo.

Proses pengumpulan respons dilakukan secara terstruktur dengan memanfaatkan mekanisme pencatatan data yang telah didefinisikan dalam konfigurasi JSON pada tahap sebelumnya. Setiap respons yang dihasilkan oleh model direkam dalam bentuk data terstruktur yang mencakup beberapa komponen penting, antara lain teks jawaban yang dihasilkan oleh model, status keberhasilan jawaban (pass), skor evaluasi, serta metadata teknis yang berkaitan dengan proses pemrosesan instruksi.

Selain teks jawaban, sistem juga mencatat informasi teknis yang dihasilkan selama proses inferensi model. Informasi tersebut meliputi waktu respons model (latency), jumlah token yang digunakan dalam proses pemrosesan instruksi, serta estimasi biaya penggunaan token berdasarkan parameter harga yang telah

ditentukan pada konfigurasi sistem. Pencatatan metadata ini penting untuk mendukung proses evaluasi efisiensi operasional model pada tahap analisis metrik.

Seluruh data respons yang dikumpulkan kemudian disimpan dalam bentuk objek data yang terstruktur sehingga dapat dianalisis secara komparatif. Dengan pendekatan ini, setiap instrumen pengujian akan memiliki tiga respons yang berasal dari masing-masing model yang diuji, yaitu GPT, Gemini, dan Claude. Data respons tersebut selanjutnya digunakan sebagai dasar dalam proses analisis performa model untuk mengukur kualitas jawaban, kecepatan respons, serta efisiensi penggunaan sumber daya komputasi.

Melalui proses pengumpulan respons model yang terstruktur ini, seluruh data hasil pengujian dapat terdokumentasi secara sistematis dan siap digunakan pada tahap analisis metrik untuk membandingkan performa masing-masing model dalam memahami instruksi berbahasa Indonesia.

3.8.7 Analisis Metrik

Tahap analisis dilakukan dengan mengekstraksi data hasil pengujian (*output matrix*) yang diproduksi oleh konfigurasi Optimasi JSON *payload*. Sistem akan memproduksi laporan kuantitatif yang mengagregasi performa operasional dan akurasi linguistik dari masing-masing model LLM.

Untuk menentukan model yang paling optimal, evaluasi didasarkan pada empat matriks utama yang terukur secara sistematis:

1. Matriks Kualitas Tertinggi (*Highest Quality / Score*)

Matriks ini mengukur tingkat akurasi dan relevansi jawaban model terhadap instrumen uji, yang direpresentasikan melalui metrik passRate dan avgScore. Pengukuran dibagi menjadi dua metode berdasarkan tipe soal:

a. **Akurasi Deterministik (MCQ):** Menggunakan matriks *Exact Match*. Sistem membandingkan nilai pada parameter choice dengan kunci jawaban (expectedAnswer). Jika identik, model mendapat bobot nilai penuh.

b. **Evaluasi Semantik (Esai dan Coding):** Mendelegasikan proses evaluasi menggunakan metode *LLM-as-a-Judge* (Juri AI). Juri AI menganalisis koherensi parameter *output* berdasarkan rubrik penilaian. Guna menjamin objektivitas penjurian dan memitigasi bias preferensi diri (*same-model*

bias), *prompt* atau input teks yang diterima oleh Juri AI untuk dievaluasi murni hanya berisi instruksi pertanyaan, rubrik penilaian, dan teks respons, tanpa menyertakan atribut pengenal (identitas) model yang menghasilkan respons tersebut. Argumen analitis juri direkam pada parameter *gradingReason* agar penilaian kualitatif dapat dipertanggungjawabkan sebelum dikonversi menjadi skor kuantitatif terukur.

$$S_{esai} = \frac{1}{n}\sum_{j=1}^{n} S_j$$

Keterangan:

S_{esai}= Skor kuantitatif final untuk jawaban esai.

n = Jumlah total model juri (n = 3).

S_j = Skor yang diberikan oleh model juri ke-j berdasarkan parameter score.

2. Matriks Kecepatan Inferensi (*Fastest / Lowest Latency*)

Dalam konteks *Large Language Model*, inferensi merujuk pada proses ketika model memproses instruksi yang diberikan dan menghasilkan respons. Oleh karena itu, kecepatan inferensi menggambarkan seberapa cepat model dapat menghasilkan jawaban setelah menerima suatu pertanyaan.

Metrik ini mengevaluasi responsivitas model dengan mengukur total waktu komputasi yang dibutuhkan model untuk menghasilkan jawaban, yang dinyatakan dalam satuan milidetik. Pengukuran dilakukan berdasarkan parameter latencyMs untuk setiap instrumen pengujian, yang kemudian dirata-ratakan menjadi avgLatencyMs. Model dengan nilai latensi rata-rata paling rendah dikategorikan sebagai model dengan kecepatan inferensi tercepat (fastest).

3. Matriks Estimasi Biaya Pokok (*Lowest Cost*)

Biaya operasional penggunaan API LLM tidak dihitung berdasarkan jumlah karakter, melainkan berdasarkan konsumsi token. Estimasi biaya pokok (totalCost) dihitung dengan menjumlahkan tarif token *input* (instruksi) dan token *output* (jawaban) menggunakan persamaan berikut:

$$\text{Total Biaya} = \sum_{i=1}^{n}((T_{in} \times P_{in}) + (T_{out} \times P_{out}))$$

Keterangan:

T_{in} = Jumlah token *input* (prompt).

P_{in} = Harga per unit token *input* sesuai tarif resmi penyedia API.

T_{out} = Jumlah token *output* (completion).

P_{out} = Harga per unit token *output* sesuai tarif resmi penyedia API.

n = Total kuantitas pengujian.

4. Matriks Efisiensi Komposit (*Most Efficient*)

Efisiensi dalam pemrosesan bahasa alami merupakan keseimbangan optimal antara kualitas jawaban tertinggi dengan penggunaan sumber daya (waktu dan biaya) terendah. Framework Promptfoo secara bawaan hanya mengeluarkan metrik operasional secara terpisah (*individual metrics*). Oleh karena itu, untuk menentukan model yang paling optimal secara menyeluruh, peneliti memformulasikan sebuah nilai indeks baru yang disebut Skor Efisiensi Komposit (E) menggunakan persamaan berikut:

$$\text{Skor Efisiensi } (E) = \frac{Q}{\bar{L} \times (1+C)}$$

Keterangan:

E = Skor Efisiensi Komplisit

Q = Kualitas/Skor rata-rata model atau *Pass Rate* (dalam bentuk desimal, misalnya 100% = 1.0 ; 95% = 0.95).

$\bar{L}$= Rata-rata latensi respons dalam detik ($\frac{avgLatencyMs}{1000}$).

C = Total biaya operasional (totalCost) dalam satuan Dolar AS (USD). Penambahan konstanta 1 berfungsi sebagai penyeimbang matematis (*mathematical smoothing*) untuk mencegah galat pembagian tak terhingga (nol) apabila model diuji menggunakan akses API *free-tier* yang berbiaya 0.

Konstruksi formula komposit ini didasarkan pada pemodelan matematika rasio efisiensi standar yang disesuaikan dengan karakteristik variabel pengujian LLM, dengan rincian logika sebagai berikut:

a) Hubungan Sebanding dengan Kualitas (Q): Variabel kualitas (Q) diletakkan sebagai pembilang karena efisiensi sebuah model bahasa dinilai positif jika

mampu mempertahankan kualitas pemahaman teks yang tinggi. Semakin tinggi skor kualitas, maka nilai *E* akan semakin besar.

b) Hubungan Terbalik dengan Latensi ($\overline{L}$): Rata-rata waktu respons ($\overline{L}$) diletakkan sebagai penyebut karena kecepatan inferensi merupakan beban waktu komputasi. Model yang memerlukan waktu pemrosesan lebih lama (latensi tinggi) akan memperkecil skor efisiensi komposit.

c) Implementasi *Mathematical Smoothing* pada Biaya (1 + *C*): Variabel biaya (*C***)** dikombinasikan dengan konstanta 1 pada penyebut sebagai teknik *smoothing*. Hal ini memiliki dua fungsi krusial:

- Mencegah Galat Pembagian Nol (*Division by Zero Error*) **:** Apabila model diuji menggunakan akses API jalur gratis (*free-tier*) yang memiliki nilai biaya *C* = 0, keberadaan nilai konstanta 1 memastikan penyebut tidak bernilai nol, sehingga fungsi matematika tetap dapat dieksekusi.
- Stabilisasi Skala Nilai Biaya: Mengingat nilai *Total Cost* (*C*) dalam pengujian API berbasis token umumnya berada pada skala desimal yang sangat kecil (misal: 0,03 USD atau 0,009 USD), penggunaan format perkalian langsung terhadap *C* tanpa konstanta akan mengecilkan nilai penyebut secara ekstrem. Kondisi tersebut dapat mendistorsi hasil akhir dan membuat model yang murah terlihat efisien secara berlebihan meskipun kualitasnya buruk. Penambahan konstanta (1 + *C*) menjaga agar pengaruh biaya tetap proporsional dan tidak merusak sensitivitas variabel latensi.

Melalui kalkulasi matriks komposit yang terukur ini, model yang memperoleh Skor Efisiensi (*E*) tertinggi akan ditetapkan secara objektif sebagai layanan AI generatif yang paling ideal bagi produktivitas akademik mahasiswa.

3.8.8 Penarikan Kesimpulan

Tahap penarikan kesimpulan merupakan fase akhir dalam penelitian ini yang bertujuan untuk merangkum dan menginterpretasikan hasil analisis yang telah diperoleh pada tahap sebelumnya. Pada tahap ini, penulis melakukan sintesis

terhadap seluruh data evaluasi yang dihasilkan dari proses benchmarking terhadap GPT, Gemini, dan Claude.

Analisis kesimpulan dilakukan dengan membandingkan performa ketiga model berdasarkan beberapa metrik utama yang telah dihitung sebelumnya, yaitu kualitas jawaban (pass rate dan skor), kecepatan respons (latensi), estimasi biaya penggunaan token, serta efisiensi komputasi secara keseluruhan. Data agregasi yang dihasilkan oleh sistem evaluasi kemudian digunakan untuk mengidentifikasi model yang memiliki performa paling optimal dalam memahami instruksi berbahasa Indonesia.

Hasil dari tahap ini tidak hanya digunakan untuk menjawab rumusan masalah penelitian, tetapi juga untuk memberikan rekomendasi praktis mengenai model Large Language Model yang paling efektif, cepat, dan efisien untuk mendukung kebutuhan akademik mahasiswa dalam penggunaan Bahasa Indonesia.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Proses Pengumpulan Instrumen Pengujian

Sebelum tahap implementasi sistem evaluasi dilakukan, proses pengumpulan instrumen pengujian telah diselesaikan. Instrumen pengujian ini dikurasi secara mandiri dari arsip ujian internal lokal yang belum pernah dipublikasikan di internet guna mencegah terjadinya kebocoran data (*data leakage*) pada model selama pengujian. Total instrumen yang dikumpulkan berjumlah 28 butir pertanyaan yang terdiri atas 20 soal Pilihan Ganda (MCQ), 5 soal Esai, dan 3 instruksi Pemrograman (Coding). Daftar instrumen pengujian secara lengkap dapat dilihat pada Lampiran A (Halaman 75-83).

4.2 Implementasi Sistem Evaluasi

Implementasi sistem mengintegrasikan *framework* NestJS sebagai orkestrator *backend* untuk mengelola distribusi *payload* JSON ke *framework* Promptfoo. Integrasi ini mengotomatisasi proses *benchmarking* terhadap GPT, Claude, dan Gemini, memastikan setiap model menerima instrumen dan konfigurasi yang identik. Sistem merekam respons model beserta metadata evaluasi (skor, status keberhasilan, latensi, jumlah token, dan estimasi biaya) melalui beberapa komponen konfigurasi berikut:

4.2.1 Konfigurasi Pengujian Model

```
{
  "batchName": "InstrumenUji-Batch-1",
  "providers": [
    "openai:gpt-5.4",
    "anthropic:claude-sonnet-4-6",
    "google:gemini-3.1-pro-preview"
  ]
}
```

Gambar 4. 1 Konfigurasi Pengujian Model dalam Format JSON

Konfigurasi model digunakan untuk menentukan model yang akan dibandingkan dalam penelitian. Konfigurasi ini disusun dalam format JSON dan digunakan oleh framework Promptfoo sebagai dasar dalam menjalankan proses benchmarking.

Pada konfigurasi tersebut, atribut batchName digunakan untuk menandai kelompok pengujian yang dijalankan dalam satu eksperimen. Penamaan ini bertujuan untuk mempermudah proses pengelolaan serta identifikasi data hasil pengujian.

Selanjutnya, atribut providers berisi daftar model yang diuji dalam penelitian ini, yaitu GPT, Claude, dan Gemini. Ketiga model tersebut diakses melalui API dari masing-masing penyedia layanan. Dengan konfigurasi ini, setiap instrumen pengujian akan dikirimkan secara identik kepada seluruh model, sehingga hasil respons yang diperoleh dapat dibandingkan secara langsung.

Konfigurasi tersebut memastikan bahwa seluruh instrumen MCQ, Essay, maupun Coding dikirim kepada ketiga model menggunakan konfigurasi yang sama sehingga hasil evaluasi dapat dibandingkan secara objektif.

4.2.2 Konfigurasi Judge Provider

```
],
"judgeProvider": [
    "openai:gpt-5.4",
    "anthropic:claude-sonnet-4-6",
    "google:gemini-3.1-pro-preview"
],
```

Gambar 4. 2 Konfigurasi Judge Provider

Konfigurasi ini digunakan terutama pada instrumen yang bersifat terbuka, seperti soal esai dan tugas pemrograman, di mana jawaban model tidak dapat dinilai hanya berdasarkan kesesuaian dengan satu jawaban benar. Oleh karena itu, penelitian ini menggunakan pendekatan LLM-as-a-Judge, yaitu memanfaatkan model bahasa untuk menilai kualitas jawaban model lain.

Melalui konfigurasi judgeProvider, sistem dapat menggunakan beberapa model sebagai evaluator sehingga proses penilaian menjadi lebih objektif. Model yang digunakan sebagai evaluator dalam penelitian ini meliputi GPT, Claude, dan Gemini.

4.2.3 Implementasi Anonimisasi Data oleh NestJS

Pada proses pengujian instrumen terbuka (Esai dan *Coding*) yang menggunakan metode *LLM-as-a-Judge*, terdapat potensi risiko bias sistemik di mana evaluator berbasis *Large Language Model* (LLM) sering kali menunjukkan bias terhadap generasinya sendiri, yang secara akademis dikenal sebagai *same-model bias* atau *self-preference bias* (Doddapaneni et al., 2024). Untuk memitigasi kerentanan objektivitas tersebut, penelitian ini mengimplementasikan fungsi anonimisasi data (*data masking*) pada level *backend* menggunakan *script* di dalam *framework* NestJS.

Secara teknis, setelah model yang diuji selesai menghasilkan respons dan sebelum NestJS mendistribusikan *JSON payload* tersebut kepada *Judge Provider* untuk dinilai, *script* pada NestJS akan mencegat (*intercept*) aliran data tersebut. Sistem kemudian menghapus dan memodifikasi atribut identitas model asal (*provider labels*). Nilai parameter yang secara eksplisit menunjukkan nama model, seperti "openai:gpt-5.4", "google:gemini-3.1-pro-preview", atau "anthropic:claude-sonnet-4-6", secara otomatis diparsing dan disamarkan menjadi alias yang netral, misalnya "Model A", "Model B", dan "Model C".

Melalui proses manipulasi *payload* ini, log pengujian dan *prompt* akhir yang dikirimkan kepada Juri AI menjadi sepenuhnya anonim. Juri AI dipaksa untuk mengeksekusi evaluasi secara buta (*blind review*), karena input yang diterimanya murni hanya berisi instruksi pertanyaan, kriteria rubrik penilaian, dan teks respons model, tanpa ada jejak identitas pembuat respons tersebut. Mekanisme anonim ini dapat membantu mengeliminasi bias terhadap LLM tertentu, sehingga secara efektif menjamin objektivitas dan keadilan dalam proses evaluasi (Cheng et al., 2024).

4.2.4 Konfigurasi Parameter Biaya Model

Gambar 4. 3 Konfigurasi Parameter Biaya Model

Gambar 4.3 menunjukkan konfigurasi parameter biaya penggunaan model yang digunakan dalam penelitian ini. Konfigurasi tersebut dituliskan dalam format JSON melalui komponen providerPrices, yang berfungsi untuk mendefinisikan biaya penggunaan token dari masing-masing model *Large Language Model* yang diuji.

Pada konfigurasi tersebut, setiap model memiliki dua parameter utama, yaitu inputPerMTok dan outputPerMTok. Parameter inputPerMTok menunjukkan biaya penggunaan token pada bagian input, sedangkan outputPerMTok menunjukkan biaya penggunaan token pada bagian output. Nilai biaya tersebut dinyatakan dalam satuan biaya per satu juta token (*per million tokens*).

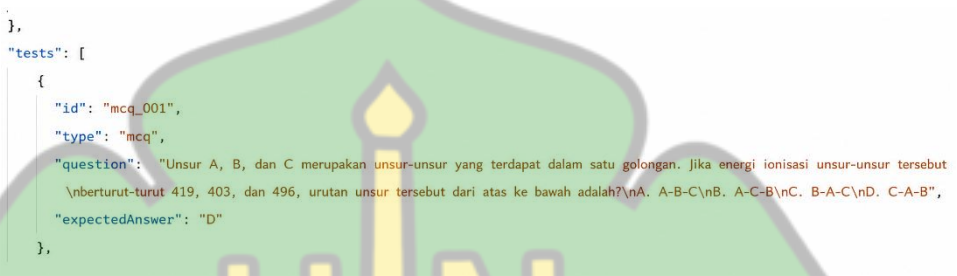
Informasi ini digunakan oleh sistem evaluasi untuk menghitung estimasi biaya operasional selama proses pengujian berlangsung. Dengan adanya konfigurasi parameter biaya ini, penelitian tidak hanya mengevaluasi kualitas jawaban dan kecepatan respons model, tetapi juga mempertimbangkan aspek efisiensi biaya penggunaan model dalam skenario penggunaan nyata.

Setelah setiap pengujian selesai dilakukan, Promptfoo menghitung biaya masing-masing respons berdasarkan jumlah input token dan output token yang tercatat pada metadata. Seluruh biaya tersebut kemudian dijumlahkan sehingga menghasilkan nilai totalCost pada bagian summary.

4.2.5 Implementasi Instrumen Pengujian

Dalam penelitian ini, instrumen pengujian dibagi menjadi tiga jenis, yaitu soal pilihan ganda (*multiple choice question*), soal esai, dan soal pemrograman. Ketiga jenis instrumen ini digunakan untuk mengukur kemampuan model pada berbagai tingkat kompleksitas, mulai dari pemahaman faktual hingga kemampuan penalaran dan penyelesaian masalah.

a. Implementasi MCQ



Gambar 4. 4 Implementasi Instrumen Pengujian MCQ

Gambar 4.4 menunjukkan contoh instrumen pengujian berupa soal pilihan ganda. Secara keseluruhan, terdapat 20 soal Pilihan Ganda (MCQ) yang digunakan dalam pengujian ini untuk mengukur akurasi logika dasar model. Seluruh daftar pertanyaan MCQ beserta pilihan jawabannya secara lengkap dapat dilihat pada Lampiran A.1 (mulai Halaman 79-84). Instrumen ini terdiri dari beberapa atribut, yaitu id, type, question, dan expectedAnswer. Atribut expectedAnswer berfungsi sebagai jawaban acuan (ground truth) yang digunakan oleh sistem untuk melakukan evaluasi otomatis terhadap respons model.

Atribut question berisi teks pertanyaan beserta pilihan jawaban yang harus dipilih oleh model, sedangkan expectedAnswer digunakan sebagai acuan dalam proses evaluasi otomatis yang dilakukan oleh Promptfoo.

Setelah model menghasilkan respons, Promptfoo melakukan ekstraksi pilihan jawaban model dan menyimpannya pada atribut choice. Selanjutnya, sistem membandingkan nilai choice dengan expectedAnswer. Apabila kedua nilai tersebut sama, maka sistem memberikan **score** sebesar 1 dan status pass bernilai true. Sebaliknya, apabila jawaban model tidak sesuai dengan expectedAnswer, maka score bernilai 0 dan status pass bernilai false. Dengan demikian, proses evaluasi pada instrumen MCQ dilakukan secara otomatis

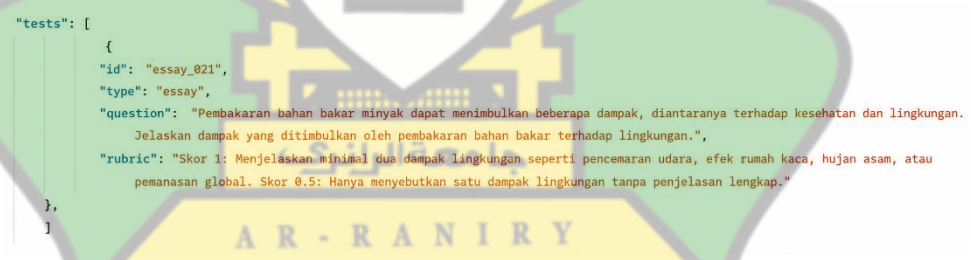
tanpa memerlukan penilaian manual karena telah menggunakan jawaban referensi yang telah ditentukan sebelumnya.

Nilai pass yang diperoleh pada setiap soal kemudian diakumulasikan untuk menghasilkan passCount, yaitu jumlah soal yang berhasil dijawab dengan benar oleh suatu model. Berdasarkan nilai tersebut, Promptfoo menghitung Pass Rate sebagai persentase jumlah soal yang memperoleh status pass = true terhadap jumlah seluruh soal yang dikerjakan model pada satu batch. Nilai Pass Rate dihitung menggunakan persamaan berikut.

$$\text{Pass Rate} = \frac{Jumlah\ soal\ dengan\ pass = true}{Jumlah\ seluruh\ soal} \times 100\%$$

Sebagai contoh, pada Batch 1 terdapat lima soal MCQ untuk setiap model. Seluruh jawaban GPT sesuai dengan nilai expectedAnswer, sehingga kelima soal memperoleh status pass = true. Oleh karena itu, nilai passCount menjadi 5 dan Pass Rate yang dihasilkan sebesar 100%. Mekanisme yang sama diterapkan pada model Claude dan Gemini dalam seluruh batch pengujian.

b. Implementasi Essay



Gambar 4. 5 Implementasi Instrumen Pengujian Essay

Gambar 4.5 menunjukkan contoh instrumen pengujian berupa soal essay. Total keseluruhan instrumen esai yang dievaluasi pada penelitian ini berjumlah 5 soal. Rincian instruksi pertanyaan beserta kriteria rubrik penilaian secara lengkap untuk kelima soal esai tersebut tercantum pada Lampiran A.2 (mulai Halaman 84-86). Berbeda dengan instrumen MCQ yang memiliki satu jawaban benar (*expectedAnswer*), pada instrumen Essay tidak terdapat jawaban referensi tunggal karena setiap model dapat menghasilkan jawaban dengan redaksi yang

berbeda namun tetap benar. Oleh karena itu, proses evaluasi dilakukan menggunakan pendekatan LLM-as-a-Judge, yaitu memanfaatkan model bahasa sebagai evaluator berdasarkan rubrik penilaian yang telah ditentukan.

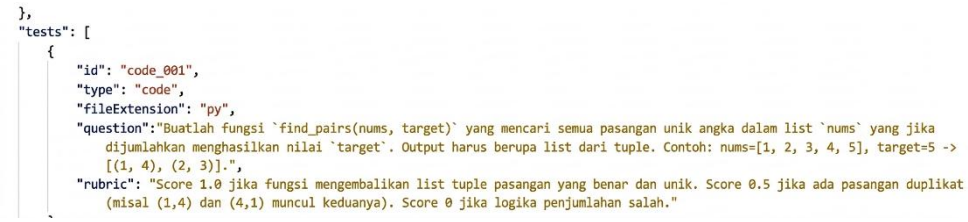
Setiap soal essay terdiri atas atribut *id*, *question*, dan rubric. Atribut rubric berisi kriteria penilaian yang digunakan sebagai pedoman bagi *judge* dalam mengevaluasi jawaban model. Rubrik mendeskripsikan aspek-aspek yang harus dipenuhi agar jawaban memperoleh nilai maksimal, sehingga proses penilaian menjadi lebih terstruktur dan konsisten.

Pada penelitian ini, proses evaluasi dilakukan menggunakan konfigurasi judgeProvider, yaitu GPT, Claude, dan Gemini yang berperan sebagai evaluator. Setelah model menghasilkan jawaban terhadap suatu soal essay, jawaban tersebut bersama dengan rubrik dikirim kepada *judge* untuk dievaluasi. *Judge* kemudian membandingkan isi jawaban dengan kriteria yang terdapat pada rubrik dan menghasilkan nilai evaluasi berupa score, pass, serta reason yang menjelaskan alasan pemberian nilai.

Nilai score menunjukkan tingkat kesesuaian jawaban model terhadap rubrik penilaian. Pada penelitian ini, score dinyatakan dalam rentang 0 hingga 1, di mana nilai 1 menunjukkan bahwa jawaban telah memenuhi seluruh kriteria pada rubrik, sedangkan nilai yang lebih rendah menunjukkan bahwa masih terdapat aspek yang belum terpenuhi secara sempurna. Sebagai contoh, apabila suatu jawaban telah memenuhi sebagian besar kriteria namun masih terdapat kekurangan pada kelengkapan atau ketepatan penjelasan, maka *judge* dapat memberikan nilai, misalnya 0,95 atau 0,98 sesuai tingkat kesesuaiannya terhadap rubrik.

Selanjutnya, nilai score dari seluruh soal pada satu batch dirata-ratakan sehingga menghasilkan Average Score, yang digunakan sebagai salah satu indikator kualitas jawaban model. Selain itu, berdasarkan nilai score tersebut Promptfoo juga menentukan status pass, yaitu apakah jawaban telah memenuhi ambang batas keberhasilan (*threshold*) yang telah ditentukan pada konfigurasi evaluasi. Nilai pass kemudian diakumulasikan untuk menghitung Pass Rate pada pengujian essay.

c. Implementasi Coding



Gambar 4. 6 Implementasi Instrumen Pengujian Pemrograman

Gambar 4.6 menampilkan contoh implementasi dari instrumen tugas pemrograman. Pada pengujian ini, terdapat total 3 soal pemrograman (Coding) yang diujikan untuk mengevaluasi kebenaran logika algoritma dari masing-masing model. Instruksi lengkap beserta parameter rubrik penilaian untuk ketiga soal *coding* ini dapat dirujuk pada Lampiran A.3 (mulai Halaman 86-87).

Tugas pemrograman Gambar 4.6 dievaluasi menggunakan instrumen *Coding* untuk mengukur kemampuan model dalam menghasilkan solusi kode sesuai spesifikasi. Berbeda dengan instrumen MCQ yang dievaluasi secara biner (*exact match*), instrumen *Coding* dinilai berdasarkan kesesuaian solusi terhadap rubrik penilaian yang fleksibel karena variasi implementasi kode yang sangat luas. Setiap butir soal dalam instrumen ini terstruktur dengan atribut id, question, fileExtension, dan rubric. Atribut rubric berfungsi sebagai parameter standar untuk menilai ketepatan logika, kelengkapan implementasi, efisiensi algoritma, dan penanganan kondisi (*edge cases*).

Proses evaluasi dijalankan otomatis menggunakan pendekatan *LLM-as-a-Judge* melalui konfigurasi judgeProvider pada *framework* Promptfoo. Setelah model menghasilkan kode program, Promptfoo mengirimkan kode tersebut beserta rubriknya ke juri AI untuk dinilai secara objektif. Hasil evaluasi juri AI terdiri dari status kelulusan (*pass*), argumentasi penilaian (*reason*), dan skor kuantitatif (*score*) dalam skala kontinu 0 hingga 1. Seluruh nilai skor tersebut dirata-ratakan untuk menghasilkan *Average Score*, sementara status *pass* diakumulasikan untuk kalkulasi *Pass Rate*.

Pendekatan skor kontinu (0–1) dipilih karena kualitas solusi pemrograman bersifat multidimensi dan tidak cukup dinilai dengan kategori benar atau salah secara mutlak. Dua model dapat menghasilkan kode berbeda yang sama-sama berfungsi (*pass*), namun memiliki perbedaan kualitas pada tingkat efisiensi

algoritma atau kelengkapan penanganan kondisi. Sebagai contoh, model dengan solusi optimal akan mendapatkan skor penuh 1,00, sedangkan model yang kodenya kurang efisien atau terlalu kompleks menerima pengurangan skor minor (misalnya menjadi 0,98 atau 0,97). Melalui mekanisme ini, sistem tidak hanya mengukur keberhasilan penyelesaian masalah tetapi juga kualitas teknis dari solusi yang didefinisikan model secara komprehensif.

4.3 Hasil Pengujian Model

4.3.1 Hasil Pengujian MCQ

Pengujian Multiple Choice Question (MCQ) dilaksanakan menggunakan 20 soal yang dibagi ke dalam empat batch pengujian. Pembagian batch dilakukan untuk mempermudah proses eksekusi tanpa mengubah mekanisme evaluasi. Hasil akhir pengujian diperoleh dengan merekapitulasi metrik yang dihasilkan pada setiap batch sehingga dapat dilakukan perbandingan performa antara GPT, Claude, dan Gemini.

a. Total cost

Tabel 4.1 Tabel Rekapitulasi Cost MCQ

Batch	**GPT**	**Claude**	**Gemini**
Batch 1	0.0076925	0.012698	0.014260
Batch 2	0.0097600	0.013332	0.015860
Batch 3	0.0058025	0.008778	0.015860
Batch 4	0.0109000	0.021528	0.018912
Total	**0.034155**	**0.056334**	**0.059244**

Berdasarkan Tabel 4.1, GPT memiliki total biaya penggunaan API paling rendah sebesar 0,034155 USD, diikuti Claude sebesar 0,056334 USD, sedangkan Gemini memiliki total biaya terbesar yaitu 0,059244 USD. Nilai tersebut diperoleh dari akumulasi summary.totalCost pada keempat batch pengujian. Hasil ini menunjukkan bahwa GPT merupakan model yang paling efisien dari sisi biaya pada pengujian MCQ.

b. Pass Rate

Seluruh model memperoleh nilai Pass Rate sebesar 100% pada keempat batch pengujian. Hal tersebut menunjukkan bahwa seluruh soal MCQ berhasil dijawab sesuai dengan nilai expectedAnswer sehingga tidak terdapat perbedaan akurasi antar model pada instrumen pilihan ganda.

c. Latency

Tabel 4. 2 Tabel Rekapitulasi Latency MCQ

Batch	GPT	Claude	Gemini
Batch 1	2436	3109	4683
Batch 2	3209	3188	6111
Batch 3	2496	2830	5571
Batch 4	3220	3920	5753
Rata-rata	**±2840 ms**	**±3262 ms**	**±5530 ms**

Berdasarkan rata-rata waktu respons pada seluruh batch, GPT menunjukkan waktu respons tercepat, diikuti Claude dan Gemini. Meskipun pada Batch 2 kategori Fastest diperoleh Claude, secara keseluruhan GPT memiliki rata-rata latensi yang lebih rendah sehingga mampu menghasilkan respons lebih cepat selama pengujian MCQ.

d. Winner

Framework Promptfoo secara otomatis menghasilkan kategori *Winner* berdasarkan hasil agregasi metrik evaluasi yang diperoleh setiap model pada bagian *summary*. Penentuan kategori tersebut dilakukan dengan membandingkan nilai metrik antar model sehingga diperoleh model dengan performa terbaik pada masing-masing kategori. Kategori Winner yang digunakan dalam penelitian ini terdiri atas Highest Quality, Fastest, Cheapest, dan Most Efficient.

Tabel 4. 3 Kriteria Penentuan Kategori Winner dalam Evaluasi Promptfoo

Kategori	**Dasar Penilaian**
Highest Quality	Average Score tertinggi
Fastest	Average Latency terendah
Cheapest	Total Cost terendah
Most Efficient	Kombinasi kualitas jawaban, biaya, dan efisiensi berdasarkan evaluasi Promptfoo

Tabel 4. 4 Hasil Penentuan Kategori Winner pada Pengujian MCQ

Batch	**Highest Quality**	**Fastest**	**Cheapest**	**Most Efficient**
1	GPT	GPT	GPT	GPT
2	GPT	Claude	GPT	GPT
3	GPT	GPT	GPT	GPT
4	GPT	GPT	GPT	GPT

Berdasarkan hasil evaluasi Promptfoo, GPT secara konsisten memperoleh kategori **Highest Quality**, **Cheapest**, dan **Most Efficient** pada seluruh batch pengujian MCQ. Sementara itu, kategori **Fastest** umumnya juga diperoleh GPT, kecuali pada Batch 2 di mana Claude memiliki rata-rata waktu respons yang sedikit lebih rendah. Secara keseluruhan, hasil tersebut menunjukkan bahwa GPT memiliki performa paling baik pada pengujian MCQ karena mampu mempertahankan kualitas jawaban yang tinggi dengan biaya penggunaan API yang lebih rendah dibandingkan model lainnya.

4.3.2 Hasil Pengujian Essay

Pengujian Essay dilakukan menggunakan lima soal uraian yang dirancang untuk mengukur kemampuan model dalam memberikan jawaban yang bersifat deskriptif dan analitis. Berbeda dengan instrumen MCQ, evaluasi pada

pengujian Essay menggunakan pendekatan *LLM-as-a-Judge*, di mana jawaban setiap model dinilai berdasarkan rubrik yang telah ditentukan. Hasil evaluasi kemudian direkapitulasi dalam bentuk metrik *Pass Rate*, *Average Score*, *Average Latency*, *Total Cost*, serta kategori *Winner*.

Tabel 4. 5 Hasil Pengujian Essay

Model	Jumalah Soal	Pass Rate	Average Score	Average Latency (ms)	Total Cost (USD)
GPT (OpenAI)	5	100%	98.3%	5.543	0,061656
Claude (Anthropic)	5	100%	95.9%	13.016	0,112550
Gemini (Google)	5	100%	98.0%	13.197	0,084805

a. Analisis Pass Rate

Berdasarkan Tabel 4.5, ketiga model memperoleh Pass Rate sebesar 100%. Hal ini menunjukkan bahwa seluruh jawaban yang dihasilkan memenuhi ambang batas (*threshold*) penilaian yang telah ditetapkan pada evaluasi berbasis rubrik. Dengan demikian, seluruh model dinyatakan berhasil menyelesaikan seluruh instrumen Essay tanpa terdapat jawaban yang gagal memenuhi kriteria evaluasi.

b. Analisis Average Score

Meskipun seluruh model memperoleh Pass Rate sebesar 100%, nilai Average Score menunjukkan adanya perbedaan kualitas jawaban yang dihasilkan. Nilai tersebut diperoleh dari hasil evaluasi menggunakan pendekatan *LLM-as-a-Judge* berdasarkan rubrik penilaian yang telah dijelaskan pada Bab III. Setiap jawaban diberikan nilai pada rentang 0–1 sesuai tingkat kesesuaiannya terhadap rubrik, kemudian nilai tersebut dirata-ratakan untuk memperoleh Average Score. Berdasarkan hasil pengujian, GPT memperoleh nilai Average Score tertinggi sebesar 98,3%, diikuti Gemini sebesar 98,0%, dan Claude sebesar 95,9%. Hasil ini

menunjukkan bahwa GPT menghasilkan jawaban yang paling konsisten memenuhi kriteria pada rubrik penilaian.

c. Contoh Evaluasi LLM-as-a-Judge pada Instrumen Essay

Sebagai ilustrasi proses penilaian menggunakan pendekatan LLM-as-a-Judge, Tabel 4.6 menyajikan hasil evaluasi pada soal Essay 021. Pada proses ini, setiap jawaban dievaluasi berdasarkan rubrik yang telah ditentukan sehingga menghasilkan nilai score dan reason sebagai dasar penilaian.

Tabel 4. 6 Hasil Evaluasi LLM-as-a-Judge pada Instrumen Essay

Model	**Ringkasan Jwaban**	**Score**	**Ringkasan Reasom**
GPT	Menjelaskan pencemaran udara, efek rumah kaca, hujan asam, dan smog akibat pembakaran bahan bakar.	0,9833	Memenuhi seluruh kriteria rubrik, namun penjelasan masih dapat dibuat lebih spesifik pada beberapa bagian.
Claude	Menjelaskan pemanasan global, hujan asam, pencemaran udara, serta kerusakan ekosistem secara rinci.	0,9833	Jawaban sangat lengkap dan sistematis, namun terdapat beberapa penjelasan yang melebihi kebutuhan pertanyaan.
Gemini	Menjelaskan pemanasan global, hujan asam, dan pencemaran udara dengan penjelasan yang jelas.	0,9833	Seluruh kriteria rubrik telah terpenuhi dengan baik, namun penyajian masih dapat dibuat lebih ringkas.

Berdasarkan Tabel 4.6, ketiga model memperoleh skor 0,9833 karena seluruh jawaban telah memenuhi kriteria utama pada rubrik, yaitu mampu menjelaskan lebih dari dua dampak lingkungan akibat pembakaran

bahan bakar minyak. Meskipun memperoleh skor yang sama, *judge* tetap memberikan reason yang berbeda sesuai karakteristik jawaban masing-masing model. Hal ini menunjukkan bahwa pendekatan LLM-as-a-Judge tidak hanya menilai benar atau salah, tetapi juga mengevaluasi kelengkapan, ketepatan, dan kualitas jawaban berdasarkan rubrik yang telah ditentukan. Nilai score dari seluruh soal kemudian dirata-ratakan menggunakan mekanisme yang telah dijelaskan pada Persamaan (3.8.7) sehingga menghasilkan Average Score sebagai salah satu metrik evaluasi penelitian.

d. Analisis Average Latency

Dari aspek waktu respons, GPT menunjukkan performa terbaik dengan rata-rata latency sebesar 5.543 ms, jauh lebih rendah dibandingkan Claude 13.016 ms dan Gemini 13.197 ms. Nilai tersebut menunjukkan bahwa GPT mampu menghasilkan jawaban lebih cepat dibandingkan kedua model lainnya pada pengujian Essay.

e. Analisis Total Cost

Total Cost merupakan akumulasi biaya penggunaan API selama proses pengujian Essay yang dihitung berdasarkan jumlah input token dan output token sesuai tarif masing-masing model. Berdasarkan hasil pengujian, GPT memiliki Total Cost terendah sebesar 0,061656 USD, sedangkan Gemini dan Claude masing-masing sebesar 0,084805 USD dan 0,112550 USD. Hal ini menunjukkan bahwa GPT memberikan efisiensi biaya yang lebih baik dibandingkan kedua model lainnya dalam menyelesaikan instrumen Essay.

f. Analisis Penentuan Winner

Tabel 4. 7 Penentuan Kategori Winner pada Pengujian Essay

Kategori	**Model Pemenang**	**Dasar Penentuan**
Highest Quality	GPT	Average Score tertinggi
Fastest	GPT	Average Latency terendah
Cheapest	GPT	Total Cost terendah
Most Efficient	GPT	Hasil evaluasi Promptfoo berdasarkan efisiensi keseluruhan model

Berdasarkan hasil evaluasi Promptfoo, GPT ditetapkan sebagai model pemenang (*Winner*) pada seluruh kategori evaluasi. Penentuan kategori Highest Quality didasarkan pada nilai Average Score tertinggi yang diperoleh GPT, yaitu sebesar 98,3%, sehingga menunjukkan bahwa model tersebut menghasilkan jawaban yang paling sesuai dengan rubrik penilaian. Sementara itu, kategori Fastest diberikan kepada GPT karena memiliki Average Latency terendah, yaitu 5.543 ms, yang menunjukkan kemampuan model dalam menghasilkan respons lebih cepat dibandingkan Claudedan Gemini.

Dari aspek efisiensi biaya, GPT juga memperoleh kategori Cheapest karena memiliki Total Cost terendah sebesar 0,061656 USD, lebih rendah dibandingkan Gemini sebesar 0,084805 USD dan Claude sebesar 0,112550 USD. Selain itu, Promptfoo menetapkan GPT sebagai Most Efficient, yang menunjukkan bahwa model tersebut memiliki keseimbangan terbaik antara kualitas jawaban, waktu respons, dan efisiensi penggunaan sumber daya selama proses pengujian. Dengan demikian, berdasarkan seluruh metrik evaluasi yang dihasilkan, GPT menunjukkan performa terbaik pada pengujian Essay.

4.3.3 Hasil Pengujian Coding

Pengujian Coding dilakukan menggunakan lima soal pemrograman yang dirancang untuk mengevaluasi kemampuan model dalam menghasilkan solusi program sesuai dengan spesifikasi yang diberikan. Berbeda dengan instrumen

MCQ, evaluasi pada instrumen Coding tidak hanya menilai benar atau salahnya keluaran program, tetapi juga mempertimbangkan kualitas implementasi berdasarkan rubrik menggunakan pendekatan LLM-as-a-Judge. Hasil pengujian disajikan berdasarkan metrik *Pass Rate*, *Average Score*, *Average Latency*, *Total Cost*, serta kategori *Winner.*

Tabel 4. 8 Hasil Pengujian Otomatis Instrumen Coding

Model	**Pass Rate**	**Average Score**	**Average Latency(ms)**	**Total Cost(USD)**
GPT (OpenAI)	100%	100.0%	5.827	0,0090175
Claude (Anthropic)	100%	96,7%	15.192	0,050439
Gemini (Google)	100%	97,8%	9.582	0,011884

a. Analisis Pass Rate

Berdasarkan Tabel 4.8, ketiga model memperoleh Pass Rate sebesar 100%. Hasil tersebut menunjukkan bahwa seluruh model berhasil menghasilkan solusi pemrograman yang memenuhi ambang batas kelulusan (*threshold*) berdasarkan rubrik penilaian. Dengan demikian, seluruh soal coding berhasil diselesaikan oleh masing-masing model tanpa terdapat respons yang dinyatakan gagal (*fail*).

b. Analisis Average Score

Meskipun seluruh model memperoleh Pass Rate sebesar 100%, nilai Average Score menunjukkan adanya perbedaan kualitas implementasi kode yang dihasilkan. Nilai tersebut diperoleh melalui evaluasi menggunakan pendekatan LLM-as-a-Judge, di mana setiap solusi dinilai berdasarkan rubrik yang telah ditentukan. Skor setiap soal kemudian dirata-ratakan menggunakan mekanisme yang dijelaskan pada Persamaan (3.8.7) sehingga menghasilkan Average Score masing-masing model. Berdasarkan hasil pengujian, GPT memperoleh nilai Average Score tertinggi sebesar 100,0%, diikuti Gemini sebesar 97,8%, dan Claude

sebesar 96,7%. Hasil ini menunjukkan bahwa implementasi kode yang dihasilkan GPTpaling konsisten memenuhi seluruh kriteria pada rubrik penilaian.

c. Contoh Evaluasi LLM-as-a-Judge

Sebagai ilustrasi proses evaluasi pada instrumen Coding, Tabel 4.9 menyajikan hasil penilaian terhadap Code 001. Pada soal ini, ketiga model berhasil menghasilkan solusi yang memenuhi kebutuhan permasalahan, namun memperoleh nilai yang berbeda karena kualitas implementasi dievaluasi menggunakan rubrik melalui pendekatan LLM-as-a-Judge.

Tabel 4. 9 Contoh Hasil Evaluasi LLM-as-a-Judge pada Soal Coding

Model	**Score**	**Ringkasan Reason**
GPT	1,0000	Implementasi sesuai dengan seluruh kriteria rubrik dan menghasilkan solusi yang benar.
Claude	0,9833	Logika telah benar, namun implementasi menambahkan fungsi, validasi, serta pengujian yang tidak diminta pada soal.
Gemini	0,9667	Implementasi telah memenuhi logika yang diminta, namun solusi dinilai terlalu ringkas karena tidak menyertakan penjelasan maupun contoh penggunaan.

Berdasarkan Tabel 4.9, ketiga model berhasil menghasilkan solusi yang sesuai dengan kebutuhan soal sehingga seluruhnya memperoleh status pass. Akan tetapi, nilai score yang diberikan *judge* berbeda karena penilaian tidak hanya mempertimbangkan benar atau salahnya logika program, tetapi juga kualitas implementasi berdasarkan rubrik. GPT memperoleh skor penuh karena seluruh kriteria pada rubrik telah dipenuhi. Claude memperoleh skor sedikit lebih rendah karena menghasilkan implementasi yang jauh lebih kompleks dibandingkan

kebutuhan soal, sedangkan Gemini memperoleh pengurangan skor karena implementasi yang diberikan sangat ringkas tanpa penjelasan tambahan. Hal ini menunjukkan bahwa pendekatan LLM-as-a-Judge mampu mengevaluasi kualitas solusi secara lebih komprehensif dibandingkan evaluasi yang hanya berfokus pada keluaran program.

d. Analisis Average Latency

Berdasarkan hasil pengujian, GPTmemiliki Average Latency terendah, yaitu 5.827 ms, diikuti Gemini sebesar 9.582 ms, dan Claude sebesar 15.192 ms. Hasil tersebut menunjukkan bahwa GPTmampu menghasilkan solusi pemrograman dalam waktu yang lebih singkat dibandingkan kedua model lainnya.

e. Analisis Total Cost

Total Cost dihitung berdasarkan akumulasi biaya penggunaan API yang diperoleh dari jumlah input token dan output token sesuai tarif masing-masing model. Berdasarkan hasil pengujian, GPTmemiliki Total Cost paling rendah sebesar 0,0090175 USD, diikuti Gemini sebesar 0,011884 USD, sedangkan Claude memiliki biaya tertinggi sebesar 0,050439 USD. Hasil ini menunjukkan bahwa GPT merupakan model yang paling efisien dari sisi biaya pada pengujian Coding.

f. Analisis Penentuan Winner

Tabel 4. 10 Penentuan Kategori Winner pada Pengujian Coding

Kategori	**Model Pemenang**	**Dasar Penentuan**
Highest Quality	GPT	Average Score tertinggi
Fastest	GPT	Average Latency terendah
Cheapest	GPT	Total Cost terendah
Most Efficient	GPT	Hasil evaluasi Promptfoo berdasarkan efisiensi keseluruhan model

Berdasarkan hasil evaluasi Promptfoo, GPT ditetapkan sebagai model pemenang (*Winner*) pada seluruh kategori evaluasi. Kategori Highest Quality diperoleh karena GPTmemiliki Average Score tertinggi

sebesar 100,0%, yang menunjukkan bahwa seluruh solusi pemrograman memenuhi kriteria pada rubrik penilaian. Selain itu, GPT juga memperoleh kategori Fastest karena memiliki Average Latency terendah serta kategori Cheapest karena memiliki Total Cost paling rendah dibandingkan model lainnya. Promptfoo juga menetapkan GPTsebagai Most Efficient, yang menunjukkan bahwa model tersebut memberikan keseimbangan terbaik antara kualitas jawaban, kecepatan respons, dan efisiensi biaya selama proses pengujian. Dengan demikian, GPT menunjukkan performa terbaik pada instrumen Coding.

g. Expert Judgment

Tabel 4. 11 Hasil Expert Judgment Pada Instrumen Coding

Soal	**Validator**	**Peringkat 1**	**Peringkat 2**	**Peringkat 3**
Coding 1	Validator 1	Gemini	GPT	Claude
Coding 1	Validator 2	Claude	GPT	Gemini
Coding 1	Validator 3	GPT	Claude	Gemini
Coding 2	Validator 1	GPT	Gemini	Claude
Coding 2	Validator 2	Claude	GPT	Gemini
Coding 2	Validator 3	GPT	Gemini	Claude
Coding 3	Validator 1	Claude	Gemini	GPT
Coding 3	Validator 2	GPT	Gemini	Claude
Coding 3	Validator 3	Gemini	Claude	GPT

Angka yang disajikan pada Tabel 4.12 merupakan hasil tabulasi silang (cross-tabulation) dari penilaian tiga validator ahli terhadap tiga instrumen Coding (Tabel 4.11), sehingga terdapat total 9 kali penilaian (3 instrumen × 3 validator). Perhitungan untuk setiap kolom dilakukan dengan cara menjumlahkan frekuensi kemunculan masing-masing model pada setiap posisi peringkat. Sebagai contoh, model GPT memperoleh Peringkat 1 sebanyak 4 kali (diberikan oleh Validator 3 pada Coding 1; Validator 1 dan 3 pada Coding 2; serta Validator 2 pada Coding 3).

Untuk menentukan Peringkat Akhir secara objektif, penelitian ini menerapkan sistem pembobotan nilai (weighted scoring system), di mana

setiap posisi peringkat dikonversi menjadi poin dengan ketentuan sebagai berikut:

- Peringkat 1 diberikan bobot 3 poin
- Peringkat 2 diberikan bobot 2 poin
- Peringkat 3 diberikan bobot 1 poin

Tabel 4. 12 Rekapitulasi Hasil Expert Judgment

Model	**Jumlah peringkat 1**	**Jumlah peringkat 2**	**Jumlah peringkat 3**	**Total poin**
GPT	4	3	2	**20**
Gemini	2	4	3	**17**
Claude	3	2	4	**17**

Berdasarkan sistem bobot tersebut, akumulasi total poin untuk masing-masing model dihitung dengan persamaan: Total Skor = (∑ Peringkat 1 × 3) + (∑ Peringkat 2 × 2) + (∑ Peringkat 3 × 1).

Hasil kalkulasi untuk ketiga model adalah sebagai berikut:

- GPT: (4 × 3) + (3 × 2) + (2 × 1) = 12 + 6 + 2 = 20 poin.
- Gemini: (2 × 3) + (4 × 2) + (3 × 1) = 6 + 8 + 3 = 17 poin.
- Claude: (3 × 3) + (2 × 2) + (4 × 1) = 9 + 4 + 4 = 17 poin.

Melalui perhitungan poin tersebut, GPT secara matematis meraih total skor tertinggi (20 poin) sehingga ditetapkan sebagai Peringkat 1.

Meskipun Gemini dan Claude memiliki akumulasi total poin yang sama (17 poin), penentuan *tie-breaker* (pemecah seri) untuk menetapkan peringkat akhir dilakukan dengan mengevaluasi frekuensi posisi terbawah (Peringkat 3). Claude ditempatkan pada posisi terbawah oleh validator lebih sering (4 kali) dibandingkan Gemini (3 kali). Hal ini mengindikasikan bahwa kualitas solusi yang dihasilkan Gemini relatif lebih stabil dan konsisten berada di posisi menengah ke atas, sehingga Gemini berhak menempati Peringkat 2, dan Claude menempati Peringkat 3.

4.4 Analisis Komparatif Hasil Pengujian

Tabel 4. 13 Perbandingan Hasil Pengujian GPT, Claude , dan Gemini

Instrumen	Metrik	GPT	Claude	Gemini	Model terbaik
MCQ	Pass Rate	100%	100%	100%	-
	Average Score	100%	100%	100%	-
	Average Latency (rata-rata 4 batch)	±2.840 ms	±3.262 ms	±5.530 ms	GPT
	Total Cost (4 batch)	0,034155 USD	0,056334 USD	0,059244 USD	GPT
Essay	Pass Rate	100%	100%	100%	-
	Average Score	98,3%	95,9%	98,0%	GPT
	Average Latency	5.543 ms	13.016 ms	13.197 ms	GPT
	Total Cost	0,061656 USD	0,112550 USD	0,084805 USD	GPT
Coding	Pass Rate	100%	100%	100%	-
	Average Score	100%	96,7%	97,8%	GPT
	Average Latency	5.827 ms	15.192 ms	9.582 ms	GPT
	Total Cost	0,0090175 USD	0,050439 USD	0,011884 USD	GPT

Berdasarkan hasil pengujian, ketiga model berhasil memperoleh Pass Rate sebesar 100% pada seluruh instrumen. Hal ini menunjukkan bahwa GPT, Claude, dan Gemini mampu menghasilkan jawaban yang memenuhi kriteria kelulusan baik pada pengujian MCQ, Essay, maupun Coding. Oleh karena itu, perbandingan

performa antar model lebih ditentukan oleh metrik Average Score, Average Latency, dan Total Cost.

Pada aspek kualitas jawaban, seluruh model memperoleh Average Score sebesar 100% pada instrumen MCQ karena evaluasi dilakukan menggunakan metode *expectedAnswer*. Namun, pada instrumen Essay dan Coding yang menggunakan pendekatan LLM-as-a-Judge, mulai terlihat perbedaan kualitas jawaban. GPT memperoleh Average Score tertinggi pada kedua instrumen tersebut, yaitu sebesar 98,3% untuk Essay dan 100% untuk Coding. Gemini berada pada posisi kedua, sedangkan Claude memperoleh nilai Average Score terendah. Hasil ini menunjukkan bahwa GPT menghasilkan jawaban yang paling konsisten memenuhi rubrik penilaian.

Dari sisi kecepatan respons, GPT juga menunjukkan performa yang paling baik. Pada pengujian MCQ, rata-rata waktu respons GPTlebih rendah dibandingkan dua model lainnya. Pola yang sama juga terlihat pada instrumen Essay dan Coding, di mana GPT secara konsisten memiliki Average Latency paling rendah. Sebaliknya, Claude dan Gemini memerlukan waktu respons yang lebih lama, terutama pada instrumen yang membutuhkan proses penalaran dan penyusunan jawaban yang lebih kompleks.

Dari aspek efisiensi biaya, GPT kembali menunjukkan performa terbaik. Total Cost yang dihasilkan pada seluruh instrumen selalu lebih rendah dibandingkan Claude maupun Gemini. Perbedaan biaya ini dipengaruhi oleh jumlah token yang digunakan selama proses inferensi serta tarif penggunaan API masing-masing model. Dengan biaya yang lebih rendah namun tetap menghasilkan kualitas jawaban yang tinggi, GPTmenjadi model yang paling efisien pada penelitian ini.

Hasil tersebut juga sejalan dengan kategori Winner yang dihasilkan oleh Promptfoo. Pada seluruh instrumen pengujian, GPT secara konsisten memperoleh kategori Highest Quality, Cheapest, dan Most Efficient. Untuk kategori Fastest, GPTmenjadi pemenang pada sebagian besar pengujian, meskipun pada salah satu batch MCQ Claude memperoleh waktu respons yang sedikit lebih cepat. Secara keseluruhan, GPTtetap menunjukkan performa paling stabil pada seluruh metrik evaluasi.

4.4.1 Analisis Bias Evaluasi Diri (*Self-Bias*)

Untuk memvalidasi objektivitas penilaian model bahasa yang bertindak sebagai juri (*LLM-as-a-judge*), dilakukan analisis pada tingkat soal (terutama pada pengujian *essay*) menggunakan metode evaluasi silang (*cross-evaluation*). Analisis ini bertujuan untuk membuktikan apakah sebuah model memiliki kecenderungan bias (meninggikan nilai) saat mengevaluasi hasil keluarannya sendiri.

Pada Tabel 4.14 berikut, ditampilkan sampel data dari lima pengujian *essay* (essay_021 hingga essay_025). Kolom **"Selisih"** menunjukkan perbedaan poin antara Rata-rata 3 *Judge* (yang melibatkan penilaian diri sendiri) dikurangi Rata-rata 2 *Judge* (murni penilaian dari dua *provider* eksternal).

1. **Selisih Positif (+)** menunjukkan indikasi bias positif (*overconfidence*), di mana model memberikan nilai tinggi pada dirinya sendiri, sehingga saat nilainya dikeluarkan, skor rata-rata menjadi turun.
2. **Selisih Negatif (-)** menunjukkan indikasi penalti diri (*self-penalization*), di mana model memberikan nilai lebih rendah pada dirinya sendiri dibandingkan penilaian *provider* lain, sehingga saat nilainya dikeluarkan, skor rata-rata justru naik.
3. **Selisih Nol (0)** menunjukkan kesepakatan sempurna antara model tersebut dengan juri lainnya.

Tabel 4. 14 Perbandingan Detail Skor Evaluasi dengan dan tanpa Self-Judge

Test	**Privider**	**Rata-rata 3 *Judge* (Dengan *Self*)**	**Est. skor self**	**Rata² 2 judge (tanpa self)**	**Selisih**
essay_021	gpt-5.4	0.983	1.00	0.975	+0.008
essay_021	claude-sonnet-4-6	0.983	1.00	0.975	+0.008
essay_021	gemini-3.1-pro	0.983	1.00	0.975	+0.008
essay_022	gpt-5.4	1.000	1.00	1.000	0.000

essay_022	claude-sonnet-4-6	0.867	1.00	0.800	+0.067
essay_022	gemini-3.1-pro	1.000	1.00	1.000	0.000
essay_023	gpt-5.4	1.000	1.00	1.000	0.000
essay_023	claude-sonnet-4-6	0.993	1.00	0.990	+0.003
essay_023	gemini-3.1-pro	1.000	1.00	1.000	0.000
essay_024	gpt-5.4	0.967	1.00	0.950	+0.017
essay_024	claude-sonnet-4-6	1.000	1.00	1.000	0.000
essay_024	gemini-3.1-pro	0.967	0.90	1.000	-0.033
essay_025	gpt-5.4	0.967	1.00	0.950	+0.017
essay_025	claude-sonnet-4-6	0.950	1.00	0.925	+0.025
essay_025	gemini-3.1-pro	0.950	0.90	0.975	−0.025

Data sampel pada Tabel 4.14 memberikan gambaran empiris mengenai fenomena bias evaluasi diri yang telah diobservasi sebelumnya:

1. **Dominasi Nilai Sempurna pada Penilaian Diri (*Overconfidence*):** Dapat diamati pada kolom "Est. Skor *Self*", model GPT-5.4 dan Claude Sonnet 4-6 secara konsisten memberikan nilai sempurna (1,00) untuk seluruh jawaban mereka sendiri pada sampel uji ini. Hal ini menyebabkan selisih yang dihasilkan selalu bernilai positif atau nol. Kasus yang paling mencolok terjadi pada Claude Sonnet di pengujian *essay_022*, di mana estimasi nilai dirinya adalah 1,00, namun rata-rata dari dua *judge* eksternal hanya 0,800.

Kehadiran nilai *self-judge* mendongkrak skor akhir menjadi 0,867 (selisih +0,067), yang secara jelas menunjukkan bias positif yang cukup signifikan.

2. **Kecenderungan Kritis (*Self-Penalization*) pada Gemini:** Berbeda dengan kedua model lainnya, Gemini 3.1 Pro menunjukkan perilaku yang lebih konservatif. Pada pengujian *essay_024* dan *essay_025*, estimasi nilai yang diberikan Gemini untuk jawabannya sendiri adalah 0,90. Sementara itu, juri eksternal (GPT dan Claude) justru memberikan evaluasi rata-rata yang lebih tinggi (berturut-turut 1,000 dan 0,975). Akibatnya, terjadi selisih negatif (-0,033 dan -0,025), yang berarti penggabungan nilai Gemini sendiri (*self-judge*) justru menurunkan rata-rata skor akhir dari jawaban tersebut

4.4.2 Analisis Kritis Dominasi Model dan Potensi Bias Sistemik

Menyikapi hasil pengujian yang menunjukkan dominasi GPT pada seluruh instrumen dan kategori metrik, penulis memandang perlunya sebuah analisis kritis untuk mengidentifikasi potensi bias sistemik di dalam ekosistem pengujian ini. Pertama, penulis mengidentifikasi bahwa keunggulan GPT pada aspek *Cheapest* dan *Fastest* sangat dipengaruhi oleh tarif komersial API OpenAI yang lebih rendah serta efisiensi *tokenizer* model tersebut dalam memampatkan kosa kata Bahasa Indonesia. Oleh karena itu, penulis menyimpulkan bahwa metrik ini tidak semata-mata mencerminkan keunggulan arsitektur penalaran model, melainkan juga dampak dari strategi ekonomi penyedia layanan.

Kedua, meskipun penelitian ini telah menerapkan metode *LLM-as-a-Judge*, memiliki risiko bias sistemik pada model evaluator yang tidak dapat diabaikan begitu saja. Juri AI kerap memiliki kecenderungan implisit untuk memberikan skor lebih tinggi pada teks yang diproduksinya sendiri, sebuah fenomena yang dikenal sebagai *same-model bias* atau *self-preference bias*. Selain itu, model juga dapat terpengaruh oleh panjang jawaban (*verbosity bias*) atau pola sintaksis yang selaras dengan korpus pelatihannya sendiri.

Untuk memitigasi potensi *same-model bias* tersebut secara teknis, penelitian ini mengimplementasikan fungsi anonimisasi respons pada level *backend* menggunakan *framework* NestJS. Melalui manipulasi *JSON payload* ini, identitas model dihilangkan sehingga Juri AI (*LLM-as-a-Judge*) mengevaluasi respons

secara buta (*blind review*). Mekanisme evaluasi anonim semacam ini terbukti efektif dalam meminimalisasi bias terhadap model LLM tertentu dan memastikan keadilan (*fairness*) dalam proses evaluasi berbasis AI.

Oleh karena itu, hasil dari Juri AI yang sudah mengevaluasi secara buta identitas ini kemudian divalidasi kembali oleh *Expert Judgment* manusia sebagai lapisan kontrol ganda (*double control mechanism*). Fakta bahwa penilaian Juri AI yang anonim tersebut selaras dengan keputusan validator manusia memberikan keyakinan bagi penulis bahwa kualitas solusi kode dan penalaran GPT memang secara objektif memenuhi standar pakar. Temuan ini sekaligus mengukuhkan posisi penulis mengenai pentingnya intervensi manusia (*engaged augmentation*) serta pelibatan panel evaluasi majemuk untuk menjamin keadilan dan reliabilitas dalam mengevaluasi luaran kecerdasan buatan.

4.4.3 Keterbatasan Penelitian

Meskipun penelitian ini telah berhasil mencapai tujuan yang ditetapkan, penulis menyadari bahwa masih terdapat beberapa keterbatasan yang dapat memengaruhi hasil penelitian. Adapun keterbatasan tersebut adalah sebagai berikut:

a. Jumlah instrumen pengujian yang terbatas
b. Pengujian hanya dilakukan pada satu periode waktu
c. Evaluasi kualitas jawaban masih dipengaruhi oleh metode penilaian yang digunakan
d. Jumlah ahli yang terlibat dalam validasi coding terbatas
e. Penelitian hanya berfokus pada penggunaan Bahasa Indonesia.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis komparatif menggunakan *framework* evaluasi otomatis Promptfoo dan metode *LLM-as-a-Judge*, serta divalidasi silang melalui *Expert Judgment*, penelitian ini menghasilkan beberapa kesimpulan yang menjawab rumusan masalah sebagai berikut:

1. Tingkat Akurasi dan Koherensi Respons

   Ketiga model (GPT, Gemini, dan Claude) menunjukkan kapabilitas yang sangat baik dalam memahami instruksi Bahasa Indonesia dasar, dibuktikan dengan pencapaian *Pass Rate* 100% pada instrumen *Multiple Choice Questions* (MCQ). Namun, pada tugas yang menuntut penalaran terbuka (Esai dan *Coding*), GPT menunjukkan tingkat akurasi dan koherensi tertinggi dengan perolehan *Average Score* rata-rata di atas 98%. Claude memiliki keunggulan dalam menghasilkan narasi yang paling komprehensif dan mendetail, namun kerap kali melebihi batasan instruksi (*over-specification*). Sementara itu, Gemini menempati posisi yang sangat kompetitif di antara keduanya dengan keseimbangan gaya bahasa yang natural, meskipun sesekali kurang optimal dalam tugas logika matematis.

2. Performa Efisiensi Teknis (Latensi dan Biaya)

   Dari segi efisiensi operasional, GPT mendominasi sebagai model tercepat (*Fastest*) dan paling hemat biaya (*Cheapest*) dalam memproses Bahasa Indonesia. Rata-rata latensi GPT berada pada kisaran 5.500 ms untuk tugas kompleks, jauh lebih cepat dibandingkan Claude dan Gemini yang membutuhkan waktu di atas 13.000 ms. Keunggulan biaya dan kecepatan ini tidak hanya didorong oleh efisiensi arsitektur, tetapi juga sangat dipengaruhi oleh strategi tarif API penyedia layanan dan tingkat efisiensi *tokenizer* model dalam memampatkan teks Bahasa Indonesia.

3. Kelebihan, Kekurangan, dan Evaluasi Bias Sistemik

Dominasi GPT pada seluruh metrik pengujian menjadikan model ini sebagai pilihan *Most Efficient*. Untuk memitigasi potensi *same-model bias* (*self-bias*) pada sistem juri AI, penelitian ini menggunakan dua lapis validasi: evaluasi silang antar *provider* (*cross-evaluation*) dan *Expert Judgment*. Hasil evaluasi silang pada tingkat soal membuktikan adanya anomali bias; model GPT dan Claude menunjukkan kecenderungan bias positif (*overconfidence*) dengan mendominasi pemberian skor sempurna (1,00) pada jawabannya sendiri, sedangkan Gemini menunjukkan anomali berupa penalti diri (*self-penalization*). Meskipun terdapat bias sistemik tersebut, validasi pakar mengonfirmasi bahwa secara objektif GPT memang menghasilkan struktur jawaban dan *clean code* yang paling sesuai dengan ekspektasi penalaran manusia. Secara keseluruhan, pemanfaatan agregasi multi-juri terbukti esensial untuk menetralisir bias inheren; di mana Claude memiliki kelemahan pada tingginya biaya dan latensi, Gemini unggul dalam adaptasi variasi bahasa namun kurang stabil, dan GPT menjadi model yang paling optimal secara keseluruhan.

5.2 Saran

Berdasarkan kesimpulan dan keterbatasan dalam penelitian ini, saran yang dapat diberikan untuk penerapan praktis maupun pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Bagi Mahasiswa dan Pengguna Akademik

Mahasiswa disarankan untuk menggunakan GPT apabila membutuhkan efisiensi waktu dan penyelesaian tugas berbasis logika terstruktur (seperti pemrograman dan analisis data). Namun, untuk kebutuhan penulisan draf literatur yang membutuhkan penjabaran mendalam, Claude dapat dijadikan alternatif yang baik. Pengguna juga diimbau untuk selalu menerapkan teknik *prompt engineering* yang spesifik dan melakukan verifikasi faktual (*cross-check*) terhadap luaran model untuk menghindari risiko halusinasi informasi.

2. Bagi Institusi Pendidikan

Mengingat pesatnya integrasi AI generatif, institusi pendidikan (khususnya perguruan tinggi) disarankan untuk mulai memasukkan literasi *Artificial Intelligence* dan keterampilan *prompt engineering* ke dalam kurikulum pembelajaran dasar. Hal ini penting agar mahasiswa tidak hanya menjadi pengguna pasif, tetapi mampu mengarahkan AI secara etis dan bertanggung jawab untuk mendukung produktivitas akademik.

3. Bagi Peneliti Selanjutnya

Penelitian di masa mendatang disarankan untuk memperluas ruang lingkup objek pengujian dengan melibatkan model *open-source* lokal atau model khusus (*fine-tuned models*) agar perbandingan lebih inklusif. Terkait metodologi evaluasi otomatis berbasis AI (*LLM-as-a-judge*), sangat disarankan untuk menghindari penggunaan evaluasi tunggal yang mengizinkan model menilai hasil keluarannya sendiri. Peneliti masa depan sebaiknya menggunakan pendekatan panel juri gabungan (agregasi multi-*provider*) guna menetralisir anomali bias evaluasi diri, baik berupa bias positif (*overconfidence*) maupun penalti diri (*self-penalization*). Selain itu, untuk meningkatkan ketajaman evaluasi, peneliti dapat memperbesar skala *dataset benchmarking* dan mengembangkan metrik pengujian yang mampu mengukur bias kultural serta sensitivitas konteks lokal dalam Bahasa Indonesia secara lebih dinamis.

DAFTAR PUSTAKA

Abeysinghe Bhashithe ; Ruhan Circi. (2024). *The Challenges of Evaluating LLM Applications : An*. *8615*(July).

Agustina, R. I. A., & Ramadhan, S. (2023). *Analysis of Syntaxic Errors in Indonesian Writing : A Literature Review*. *7*(2), 362–384.

Alhur, A. (2024). Redefining Healthcare With Artificial Intelligence (AI): The Contributions of ChatGPT, Gemini, and Co-pilot. *Cureus*, *16*(4). https://doi.org/10.7759/cureus.57795

Amien, M. (2023). *Sejarah dan Perkembangan Teknik Natural Language Processing (NLP) Bahasa Indonesia: Tinjauan tentang sejarah, perkembangan teknologi, dan aplikasi NLP dalam bahasa Indonesia*. *2007*, 1–7. http://arxiv.org/abs/2304.02746

Autor, D. (2024). *Applying AI to Rebuild Middle Class Jobs* (NBER Working Paper No. 32140). National Bureau of Economic Research.Cheng, M., & Chen, E. (2024). *Towards Personalized Evaluation of Large Language Models with An Anonymous Crowd-Sourcing Platform*. 1035–1038. https://doi.org/10.1145/3589335.3651243

Cheng, M., Zhang, H., Yang, J., Liu, Q., Li, L., Huang, X., Song, L., Li, Z., Huang, Z., & Chen, E. (2024). Towards Personalized Evaluation of Large Language Models with An Anonymous Crowd-Sourcing Platform. *WWW 2024 Companion - Companion Proceedings of the ACM Web Conference*, 1035–1038. https://doi.org/10.1145/3589335.3651243

de Carvalho Souza, M. E. . W. L. (2025). Grok, gemini, chatgpt and deepseek: Comparison and applications in conversational artificial intelligence. *Inteligencia Artificial*, *2*(1), 1–7. https://doi.org/10.5281/zenodo.14885243

Doddapaneni, S., Khan, M. S. U. R., Verma, S., & Khapra, M. M. (2024). Finding Blind Spots in Evaluator LLMs with Interpretable Checklists. *EMNLP 2024 - 2024 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 16279–16309. https://doi.org/10.18653/v1/2024.emnlp-main.911

Gencoglu, B., Helms-lorenz, M., & Maulana, R. (2023). Computers & Education Machine and expert judgments of student perceptions of teaching behavior in secondary education : Added value of topic modeling with big data. *Computers & Education*,

193(September 2022), 104682. https://doi.org/10.1016/j.compedu.2022.104682

Hashemi, H., Eisner, J., Rosset, C., Van Durme, B., & Kedzie, C. (2024). LLM-RUBRIC: A Multidimensional, Calibrated Approach to Automated Evaluation of Natural Language Texts. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, *1*(June), 13806–13834. https://doi.org/10.18653/v1/2024.acl-long.745

Jeldres, M. R., Costa, E. D., & Nadim, F. (2023). *A review of Lawshe ' s method for calculating content validity in the social sciences*. *November*, 1–8. https://doi.org/10.3389/feduc.2023.1271335

Koto, F., & Baldwin, T. (2020). *IndoLEM and IndoBERT : A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP*. 757–770.

Mykhalko, Y. O., Filak, Y. F., Dutkevych-Ivanska, Y. V., Sabadosh, M. V., & Rubtsova, Y. I. (2024). From open-ended to multiple-choice: evaluating diagnostic performance and consistency of ChatGPT, Google Gemini and Claude AI. *Wiadomosci Lekarskie (Warsaw, Poland : 1960)*, *77*(10), 1852–1856. https://doi.org/10.36740/WLek/195125

Ramadhani, R., Hamzah, R. A., Ramadani, C. A., Perintis, J., Km, K., & Makassar, N. (2025). *Struktur Kebahasaan Bahasa Indonesia Sebagai Rujukan Pengunaan Bahasa (Sintaksis)*. *2*(4), 27–33.

Rane, N., Choudhary, S., & Rane, J. (2024). Gemini or ChatGPT? Capability, Performance, and Selection of Cutting-edge Generative Artificial Intelligence (AI) in Business Management. *Studies in Economics and Business Relations*, *5*(1), 40–50. https://doi.org/10.48185/sebr.v5i1.1051

Samiolo, R., Spence, C., & Toh, D. (2024). Auditor judgment in the fourth industrial revolution. *Contemporary Accounting Research*, *41*(1), 498–528. https://doi.org/10.1111/1911-3846.12901

Sparkman, M., & Witt, A. (2025). Claude AI and Literature Reviews: An Experiment in Utility and Ethical Use. *Library Trends*, *73*(3), 355–380. https://doi.org/10.1353/lib.2025.a961199

Uppalapati, V. K., & Nag, D. S. (2024). A Comparative Analysis of AI Models in Complex Medical Decision-Making Scenarios: Evaluating ChatGPT, Claude AI, Bard, and Perplexity. *Cureus*, *16*(1), 4–9. https://doi.org/10.7759/cureus.52485

Wiyono, V. R., Anugraha, D., Purwarianti, A., & Winata, G. I. (2025). *I NDO P REF : A Multi-Domain Pairwise Preference Dataset for Indonesian*. 128–138.

Zaremba, A., & Demir, E. (2023). ChatGPT: Unlocking the future of NLP in finance. *Modern Finance*, *1*(1), 93–98. https://doi.org/10.61351/mf.v1i1.43

Zhao, R., Zhang, W., Chia, Y. K., Xu, W., Zhao, D., & Bing, L. (2025). Auto-Arena: Automating LLM Evaluations with Agent Peer Battles and Committee Discussions. *Proceedings of the Annual Meeting of the Association for Computationa lLinguistics*, *1*, 4440–4463. https://doi.org/10.18653/v1/2025.acl-long.223

LAMPIRAN

Lampiran A: Instrumen Pengujian

A.1. Instrumen Uji Pilihan Ganda (Multiple Choice Questions)

Soal 1:

Unsur A, B, dan C merupakan unsur-unsur yang terdapat dalam satu golongan. Jika energi ionisasi unsur-unsur tersebut berturut-turut 419, 403, dan 496, urutan unsur tersebut dari atas ke bawah adalah?

A. A-B-C
B. A-C-B
C. B-A-C
D. C-A-B

Kunci jawaban : D

Soal 2:

Perubahan posisi benda terhadap posisi awalnya merupakan pengertian dari?

A. Kecepatan
B. Kelajuan
C. Posisi
D. Gerak

Kunci jawaban : D

Soal 3:

Pernyataan berikut yang paling tepat mengenai senyawa hidrokarbon adalah?

A. Semua senyawa yang mengandung atom karbon.
B. Senyawa yang mengandung karbon dan oksigen.
C. Semua senyawa yang mengandung hidrogen
D. Senyawa yang mengandung karbon dan hidrogen

Kunci jawaban : D

Soal 4:

Rumus molekul yang menyatakan hidrokarbon jenuh adalah?

A. C3H6
B. C4H10
C. C3H4
D. C4H8

Kunci jawaban : B

Soal 5:

Jika sebuah benda mengalami perubahan kecepatan dalam selang waktu tertentu maka benda tersebut mengalami?

A. Kecepatan
B. Kecepatan rata-rata
C. Percepatan
D. Kelajuan

Kunci jawaban : C

Soal 6:

Sebuah mobil dipercepat 4 m/s2 dari kondisi diam sehingga mobil mencapai kelajuan 30 m/s. Berapakah waktu tempuh mobil tersebut?

A. 5 sekon
B. 7,5 sekon
C. 10 sekon
D. 12 sekon

Kunci jawaban : B

Soal 7:

Sebuah batu dilemparkan ke dalam sumur dengan kecepatan awal 4 m/s. Jika batu mengenai dasar sumur setelah 2 sekon semenjak dilemparkan dan percepatan gravitasi 10 m/s2, maka kedalaman sumur tersebut yaitu.... meter.

A. 30 m
B. 28 m
C. 24 m
D. 20 m

Kunci jawaban : B

Soal 8:

Mobil awalnya diam kemudian digas sehingga melaju dengan percepatan tertentu. Peristiwa ini menunjukkan?

A. Hukum I Newton
B. Hukum II Newton
C. Hukum III Newton
D. Hukum aksi reaksi

Kunci jawaban : B

Soal 9:

Kekhasan atom karbon yang menyebabkan unsur karbon mempunyai banyak ragam senyawa adalah?

A. Mempunyai empat elektron valensi yang dapat digunakan untuk berikatan kovalen.
B. Mempunyai konfigurasi elektron yang belum stabil seperti gas mulia.
C. Dapat membentuk rantai karbon dengan berbagai bentuk.
D. Merupakan zat padat yang sangat stabil pada suhu kamar.

Kunci jawaban : C

Soal 10:

Jika benda bergerak melingkar dengan periode tetap sebesar 1/8 sekon, dalam geraknya maka benda melakukan?

A. Satu putaran selama 8 sekon dengan laju tetap.
B. 8 putaran tiap sekon dengan kecepatan sudut berubah.
C. 8 putaran tiap sekon dengan kecepatan sudut tetap.
D. 1/4 putaran tiap sekon dengan kecepatan sudut tetap.

Kunci jawaban : C

Soal 11:

Jika sebuah benda terletak pada bidang miring, maka gaya normal pada benda itu adalah?

A. Sama dengan berat benda
B. Lebih kecil dari berat benda
C. Lebih besar dari berat benda
D. Dapat lebih besar atau lebih kecil dari berat benda

Kunci jawaban : B

Soal 12:

Penyusun utama minyak bumi adalah senyawa?

A. Alkana
B. Hidrokarbon aromatis
C. Sikloalkana
D. Alkanatiol

Kunci jawaban : A

Soal 13:

Apa yang dimaksud dengan kualitas sumber daya manusia?

A. Jumlah penduduk yang tinggi
B. Jumlah tenaga kerja di sektor industri
C. Tingkat pendidikan, keterampilan, dan kesehatan individu
D. Pendapatan per kapita

Kunci jawaban : C

Soal 14:

Kebutuhan manusia akan makanan, minuman, sandang dan papan merupakan kebutuhan?

A. Sekunder
B. Tersier
C. Primer
D. Rohani

Kunci jawaban : C

Soal 15:

Perubahan penduduk yang terjadi karena adanya kelahiran, kematian dan perpindahan penduduk disebut...

A. Dinamika sosial
B. Dinamika penduduk
C. Dinamika masyarakat
D. Dinamika negara

Kunci jawaban : B

Soal 16:

Siapakah yang membaca teks proklamasi kemerdekaan Indonesia pada tanggal 17 Agustus 1945?

A. Habibi
B. Cut Nyak Dien
C. Sukarno
D. Sukarni

Kunci jawaban : C

Soal 17:

Mengapa pendidikan dianggap sebagai salah satu kunci utama untuk mencapai kemajuan suatu negara?

A. Karena pendidikan hanya mengejar
B. Karena pendidikan meningkatkan kemampuan penduduk untuk mengelola SDA
C. Karena pendidikan membuat penduduk lebih banyak bekerja dibanding pendidikan
D. Karena pendidikan mengurangi jumlah penduduk

Kunci jawaban : B

Soal 18:

Keberagaman budaya di Indonesia dipengaruhi oleh faktor geografis seperti...

A. Isologi geografis, kondisi iklim, letak geografis
B. Isologi geografis, kondisi cuaca, letak geografis
C. Kondisi cuaca, kondisi iklim, letak geografis
D. Isologi geografis, kondisi iklim, letak astronomis

Kunci jawaban : A

Soal 19:

Siska melempar batu vertikal ke atas dengan kecepatan 40 m/s. Jika percepatan gravitasi 10 m/s2, maka kedudukan setelah 6 sekon adalah...

A. Sedang bergerak naik
B. Sedang bergerak turun
C. Berhenti sesaat
D. Tiba di tanah

Kunci jawaban : B

Soal 20:

Pernyataan yang benar tentang kecepatan dan percepatan di bawah ini adalah...

A. Bila percepatan bernilai negatif maka selalu menghasilkan perlambatan
C. Bila percepatan bernilai positif dan kecepatan bernilai positif maka benda bergerak dipercepat

B. Bila percepatan bernilai positif maka selalu menghasilkan perlambatan

D. Bila percepatan bernilai positif dan kecepatan bernilai positif maka benda bergerak diperlambat

Kunci jawaban : C

A.2. Instrumen Uji Esai (Question Answering)

Soal 1:

Intruksi: Pembakaran bahan bakar minyak dapat menimbulkan beberapa dampak, diantaranya terhadap kesehatan dan lingkungan. Jelaskan dampak yang ditimbulkan oleh pembakaran bahan bakar terhadap lingkungan!

Rubrik penilaian:

Skor	Kriteria Implementasi Kode
1.0	Menjelaskan minimal dua dampak lingkungan seperti pencemaran udara, efek rumah kaca, hujan asam, atau pemanasan global.
0.5	Hanya menyebutkan satu dampak lingkungan tanpa penjelasan lengkap.

Soal 2:

Konfigurasi elektron suatu unsur adalah 1s2 2s2 2p6 3s2 3p6 3d10 4s2 4p6 4d7 5s2. Berdasarkan konfigurasi elektron tersebut, tentukan periode dan golongan unsur dalam sistem periodik!

Rubrik penilaian:

Skor	Kriteria Implementasi Kode
1.0	Menentukan periode dengan benar (periode 5) dan golongan dengan benar (golongan 9/VIIIB).
0.5	Hanya menentukan periode atau golongan dengan benar.

Soal 3:

Sebuah benda dilempar ke atas dengan kecepatan awal 40 m/s dan sudut elevasi 60° terhadap horizontal. Tentukan kecepatan benda saat berada di titik tertinggi! (g = 10 m/s2).

Rubrik penilaian:

Skor	Kriteria Implementasi Kode
1.0	Menjelaskan bahwa komponen kecepatan vertikal menjadi nol di titik tertinggi dan menentukan kecepatan sama dengan komponen horizontal (20 m/s).
0.5	Menyebutkan konsep titik tertinggi tetapi tidak menghitung kecepatan dengan benar.

Soal 4:

Dua benda dengan massa m1 = 100 g dan m2 = 300 g dilontarkan ke atas secara bersamaan dari lantai dengan kecepatan awal masing-masing v1 = 20 m/s dan v2 = 30 m/s. Jika gesekan udara diabaikan dan g = 10 m/s2, hitunglah perbedaan ketinggian kedua benda setelah t = 3 sekon!

Rubrik penilaian:

Skor	Kriteria Implementasi Kode
1.0	Menggunakan persamaan gerak vertikal dengan benar dan mendapatkan selisih ketinggian yang tepat.
0.5	Menggunakan rumus gerak tetapi perhitungan atau hasil akhir tidak tepat.

Soal 5:

Bagaimana pendapatmu tentang teman yang berbuat curang ketika ujian sekolah dan apa yang akan kamu lakukan jika melihat hal tersebut? Apakah perilaku tersebut melanggar norma? Jelaskan!

Rubrik penilaian:

Skor	Kriteria Implementasi Kode
1.0	Menjelaskan bahwa tindakan tersebut melanggar norma kejujuran dan aturan sekolah serta memberikan sikap yang tepat (menolak mencontek atau melaporkan).
0.5	Hanya menyatakan bahwa tindakan tersebut salah tanpa penjelasan norma atau sikap yang tepat.

A.3. Instrumen Coding

Soal 1:

Buatlah fungsi `find_pairs(nums, target)` yang mencari semua pasangan unik angka dalam list `nums` yang jika dijumlahkan menghasilkan nilai `target`. Output harus berupa list dari tuple. Contoh: nums=[1, 2, 3, 4, 5], target=5 -> [(1, 4), (2, 3)].Rubrik penilaian:

Skor	Kriteria Implementasi Kode
1.0	jika fungsi mengembalikan list tuple pasangan yang benar dan unik.
0.5	jika ada pasangan duplikat (misal (1,4) dan (4,1) muncul keduanya).
0	jika logika penjumlahan salah.

Soal 2:

mplementasikan fungsi `fibonacci(n)` yang mengembalikan list berisi `n` angka pertama dari deret Fibonacci. Gunakan pendekatan iteratif untuk efisiensi. Contoh: n=5 -> [0, 1, 1, 2, 3].

Rubrik penilaian:

Skor	Kriteria Implementasi Kode
1.0	jika menghasilkan list dengan n angka pertama yang tepat.
0.5	jika deret benar tapi jumlah elemen salah (off-by-one error).
0	jika menggunakan rekursi tanpa memoization untuk n besar atau logika salah.

Soal 3:

Buat fungsi `sort\_by\_length(strings)` yang menerima list string dan mengurutkannya berdasarkan panjang karakter dari yang terpendek ke terpanjang. Jika panjangnya sama, urutkan secara alfabetis. Contoh: ['apple', 'pie', 'banana', 'pear'] -> ['pie', 'pear', 'apple', 'banana'].

Rubrik penilaian:

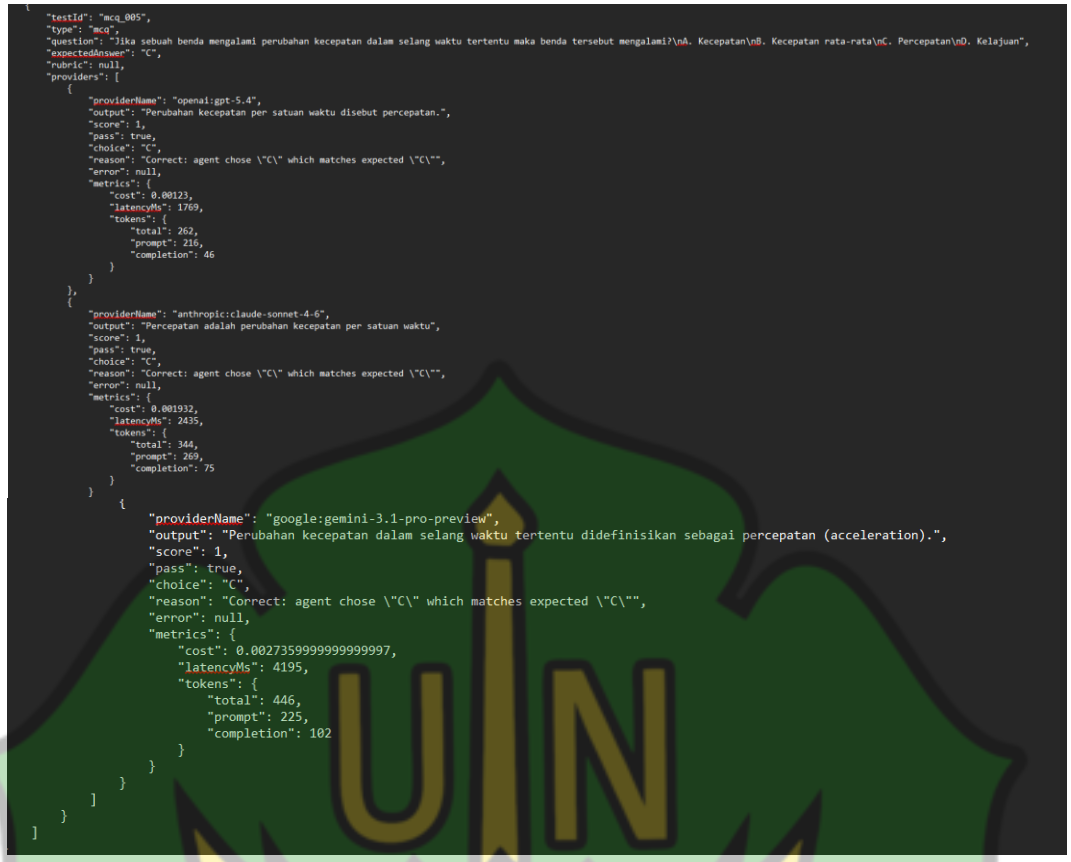
Skor	Kriteria Implementasi Kode
1.0	jika pengurutan panjang benar dan pengurutan alfabetis (tie-break) juga benar.
0.5	jika hanya urut berdasarkan panjang tapi alfabetisnya acak.
0	jika gagal mengurutkan.

Lampiran B: Bukti Output Model


```
{
    "batchName": "InstrumenUji-Batch-1",
    "timestamp": "2026-04-20T21:48:49.011Z",
    "summary": {
        "totalTests": 15,
        "providerStats": [
            {
                "provider": "openai:gpt-5.4",
                "totalTests": 5,
                "passCount": 5,
                "errorCount": 0,
                "passRate": "100.0%",
                "errorRate": "0.0%",
                "avgScore": "100.0%",
                "totalCost": 0.0076925,
                "avgLatencyMs": 2436
            },
            {
                "provider": "anthropic:claude-sonnet-4-6",
                "totalTests": 5,
                "passCount": 5,
                "errorCount": 0,
                "passRate": "100.0%",
                "errorRate": "0.0%",
                "avgScore": "100.0%",
                "totalCost": 0.012696,
                "avgLatencyMs": 3109
            },
            {
                "provider": "google:gemini-3.1-pro-preview",
                "totalTests": 5,
                "passCount": 5,
                "errorCount": 0,
                "passRate": "100.0%",
                "errorRate": "0.0%",
                "avgScore": "100.0%",
                "totalCost": 0.01426,
                "avgLatencyMs": 4683
            }
        ],
        "winners": {
            "mostEfficient": "openai:gpt-5.4",
            "cheapest": "openai:gpt-5.4",
            "fastest": "openai:gpt-5.4",
            "highestQuality": "openai:gpt-5.4"
        }
    },
```


"results": [
"testId": "mcq_001",
"type": "mcq",
"question": "Unsur A, B, dan C merupakan unsur-unsur yang terdapat dalam satu golongan. Jika energi ionisasi unsur-unsur tersebut berturut-turut 419, 403, dan 496, urutan unsur tersebut dari atas ke bawah adalah?\nA. A-B-C \nB. A-C-B\nC. B-A-C\nD. C-A-B",
"expectedAnswer": "D",
"rubric": null,
"providers": [
"providerName": "openai:gpt-5.4",
"output": "Dari atas ke bawah energi ionisasi menurun: 496, 419, 403.",
"score": 1,
"pass": true,
"choice": "D",
"reason": "Correct: agent chose \"D\" which matches expected \"D\"",
"error": null,
"metrics": {
"cost": 0.0019549999999999997,
"latencyMs": 4411,
"tokens": {
"total": 337,
"prompt": 248,
"completion": 89
"providerName": "anthropic:claude-sonnet-4-6",
"output": "Dalam satu golongan, energi ionisasi menurun dari atas ke bawah. Urutan dari tertinggi ke terendah: C(496) > A(419) > B(403), sehingga dari atas ke bawah: C-A-B",
"score": 1,
"pass": true,
"choice": "D",
"reason": "Correct: agent chose \"D\" which matches expected \"D\"",
"error": null,
"metrics": {
"cost": 0.003771,
"latencyMs": 4441,
"tokens": {
"total": 497,
"prompt": 307,
"completion": 190
"providerName": "google:gemini-3.1-pro-preview",
"output": "Energi ionisasi dalam satu golongan berkurang dari atas ke bawah. Jadi urutannya adalah yang paling besar ke yang paling kecil: C (496), A (419), B (403).",
"score": 1,
"pass": true,
"choice": "D",
"reason": "Correct: agent chose \"D\" which matches expected \"D\"",
"error": null,
"metrics": {
"cost": 0.0033500000000000005,
"latencyMs": 5189,
"tokens": {
"total": 558,
"prompt": 275,
"completion": 125
"testId": "mcq_002",
"type": "mcq",
"question": "Perubahan posisi benda terhadap posisi awalnya merupakan pengertian dari?\nA. Kecepatan\nB. Kelajuan\nC. Posisi\nD. Gerak",
"expectedAnswer": "D",
"rubric": null,
"providers": [
"providerName": "openai:gpt-5.4",
"output": "Perubahan posisi benda adalah gerak.",
"score": 1,
"pass": true,
"choice": "D",
"reason": "Correct: agent chose \"D\" which matches expected \"D\"",
"error": null,
"metrics": {
"cost": 0.00121,
"latencyMs": 1768,
"tokens": {
"total": 254,
"prompt": 208,
"completion": 46
"providerName": "anthropic:claude-sonnet-4-6",
"output": "Gerak adalah perubahan posisi benda terhadap posisi awalnya (titik acuan).",
"score": 1,
"pass": true,
"choice": "D",
"reason": "Correct: agent chose \"D\" which matches expected \"D\"",
"error": null,
"metrics": {
"cost": 0.002835,
"latencyMs": 3212,
"tokens": {
"total": 389,
"prompt": 250,
"completion": 139
"providerName": "google:gemini-3.1-pro-preview",
"output": "Gerak adalah perubahan posisi benda terhadap titik acuan atau posisi awalnya.",
"score": 1,
"pass": true,
"choice": "D",
"reason": "Correct: agent chose \"D\" which matches expected \"D\"",
"error": null,
"metrics": {
"cost": 0.0023220000000000003,
"latencyMs": 5805,
"tokens": {
"total": 609,
"prompt": 216,
"completion": 81

```
{
        "testId": "mcq_003",
        "type": "mcq",
        "question": "Pernyataan berikut yang paling tepat mengenai senyawa hidrokarbon adalah?\nA. Semua senyawa yang mengandung atom karbon\nB. Senyawa yang mengandung karbon dan oksigen\nC. Semua senyawa yang mengandung hidrogen\nD. Senyawa yang mengandung karbon dan hidrogen",
        "expectedAnswer": "D",
        "rubric": null,
        "providers": [
          {
            "providerName": "openai:gpt-5.4",
            "output": "Hidrokarbon adalah senyawa yang mengandung karbon dan hidrogen.",
            "score": 1,
            "pass": true,
            "choice": "D",
            "reason": "Correct: agent chose \"D\" which matches expected \"D\"",
            "error": null,
            "metrics": {
              "cost": 0.001665,
              "latencyMs": 2184,
              "tokens": {
                "total": 311,
                "prompt": 240,
                "completion": 71
              }
            }
          },
          {
            "providerName": "anthropic:claude-sonnet-4-6",
            "output": "Hidrokarbon adalah senyawa yang tersusun dari atom karbon dan hidrogen saja.",
            "score": 1,
            "pass": true,
            "choice": "D",
            "reason": "Correct: agent chose \"D\" which matches expected \"D\"",
            "error": null,
            "metrics": {
              "cost": 0.0019709999999999997,
              "latencyMs": 2491,
              "tokens": {
                "total": 361,
                "prompt": 287,
                "completion": 74
              }
            }
          },
          {
            "providerName": "google:gemini-3.1-pro-preview",
            "output": "Hidrokarbon secara definisi adalah senyawa yang hanya terdiri dari unsur karbon dan hidrogen.",
            "score": 1,
            "pass": true,
            "choice": "D",
            "reason": "Correct: agent chose \"D\" which matches expected \"D\"",
            "error": null,
            "metrics": {
              "cost": 0.00232,
              "latencyMs": 3771,
              "tokens": {
                "total": 483,
                "prompt": 238,
                "completion": 76
              }
            }
          }
        ]
      },
      {
        "testId": "mcq_004",
        "type": "mcq",
        "question": "Rumus molekul yang menyatakan hidrokarbon jenuh adalah?\nA. C3H6\nB. C4H10\nC. C3H4\nD. C4H8",
        "expectedAnswer": "B",
        "rubric": null,
        "providers": [
          {
            "providerName": "openai:gpt-5.4",
            "output": "Alkana memiliki rumus CnH2n+2; yang cocok adalah C4H10.",
            "score": 1,
            "pass": true,
            "choice": "B",
            "reason": "Correct: agent chose \"B\" which matches expected \"B\"",
            "error": null,
            "metrics": {
              "cost": 0.0016325,
              "latencyMs": 2046,
              "tokens": {
                "total": 288,
                "prompt": 215,
                "completion": 73
              }
            }
          },
          {
            "providerName": "anthropic:claude-sonnet-4-6",
            "output": "C4H10 sesuai rumus alkana CnH(2n+2) dengan n=4",
            "score": 1,
            "pass": true,
            "choice": "B",
            "reason": "Correct: agent chose \"B\" which matches expected \"B\"",
            "error": null,
            "metrics": {
              "cost": 0.0021869999999999997,
              "latencyMs": 2967,
              "tokens": {
                "total": 341,
                "prompt": 244,
                "completion": 97
              }
            }
          },
          {
            "providerName": "google:gemini-3.1-pro-preview",
            "output": "Hidrokarbon jenuh (alkana) memiliki rumus umum CnH2n+2. C4H10 memenuhi rumus ini.",
            "score": 1,
            "pass": true,
            "choice": "B",
            "reason": "Correct: agent chose \"B\" which matches expected \"B\"",
            "error": null,
            "metrics": {
              "cost": 0.003532,
              "latencyMs": 4454,
              "tokens": {
                "total": 539,
                "prompt": 226,
                "completion": 146
              }
            }
          }
        ]
```



```
{
    "testId": "mcq_005",
    "type": "mcq",
    "question": "Jika sebuah benda mengalami perubahan kecepatan dalam selang waktu tertentu maka benda tersebut mengalami?\nA. Kecepatan\nB. Kecepatan rata-rata\nC. Percepatan\nD. Kelajuan",
    "expectedAnswer": "C",
    "rubric": null,
    "providers": [
        {
            "providerName": "openai:gpt-5.4",
            "output": "Perubahan kecepatan per satuan waktu disebut percepatan.",
            "score": 1,
            "pass": true,
            "choice": "C",
            "reason": "Correct: agent chose \"C\" which matches expected \"C\"",
            "error": null,
            "metrics": {
                "cost": 0.00123,
                "latencyMs": 1769,
                "tokens": {
                    "total": 262,
                    "prompt": 216,
                    "completion": 46
                }
            }
        },
        {
            "providerName": "anthropic:claude-sonnet-4-6",
            "output": "Percepatan adalah perubahan kecepatan per satuan waktu",
            "score": 1,
            "pass": true,
            "choice": "C",
            "reason": "Correct: agent chose \"C\" which matches expected \"C\"",
            "error": null,
            "metrics": {
                "cost": 0.001932,
                "latencyMs": 2435,
                "tokens": {
                    "total": 344,
                    "prompt": 269,
                    "completion": 75
                }
            }
        },
        {
            "providerName": "google:gemini-3.1-pro-preview",
            "output": "Perubahan kecepatan dalam selang waktu tertentu didefinisikan sebagai percepatan (acceleration).",
            "score": 1,
            "pass": true,
            "choice": "C",
            "reason": "Correct: agent chose \"C\" which matches expected \"C\"",
            "error": null,
            "metrics": {
                "cost": 0.0027359999999999997,
                "latencyMs": 4195,
                "tokens": {
                    "total": 446,
                    "prompt": 225,
                    "completion": 102
                }
            }
        }
    ]
}
]
```

Lampiran C: Lembar Validasi Expert Judgment

C.1. Lembar Validasi Ahli 1

LEMBAR PENILAIAN AHLI
(expert judgment)

Instruksi Penilaian
Silakan membaca ketiga solusi kode yang diberikan (Kode A, B, dan C), kemudian lakukan penilaian dengan memberikan peringkat berdasarkan:

- Kebenaran logika program
- Kelengkapan solusi
- Keterbacaan dan struktur kode
- Efisiensi penyelesaian

Berikan:
Peringkat 1 = terbaik
Peringkat 2 = sedang
Peringkat 3 = terendah

SOAL CODING 1

Pertanyaan Penilaian

1. Berdasarkan kualitas keseluruhan kode, manakah yang Anda nilai terbaik?

Peringkat 1 : Gemini
Peringkat 2 : GPT
Peringkat 3 : Claude

2. Apa alasan utama Anda memilih kode dengan peringkat terbaik?

- lebih ringkas / singkat

3. Apa kelemahan utama dari kode dengan peringkat terendah?

- lebih panjang, struktur sintaksnya

SOAL CODING 3

Pertanyaan Penilaian

1. Berdasarkan kualitas keseluruhan kode, manakah yang Anda nilai terbaik?

Peringkat 1 : Claude

Peringkat 2 : Gemini

Peringkat 3 : GPT

2. Apa alasan utama Anda memilih kode dengan peringkat terbaik?

lebih ringkas / singkat

3. Apa kelemahan utama dari kode dengan peringkat terendah?

SOAL CODING 2

Pertanyaan Penilaian

1. Berdasarkan kualitas keseluruhan kode, manakah yang Anda nilai terbaik?

Peringkat 1 : GPT

Peringkat 2 : Gemini

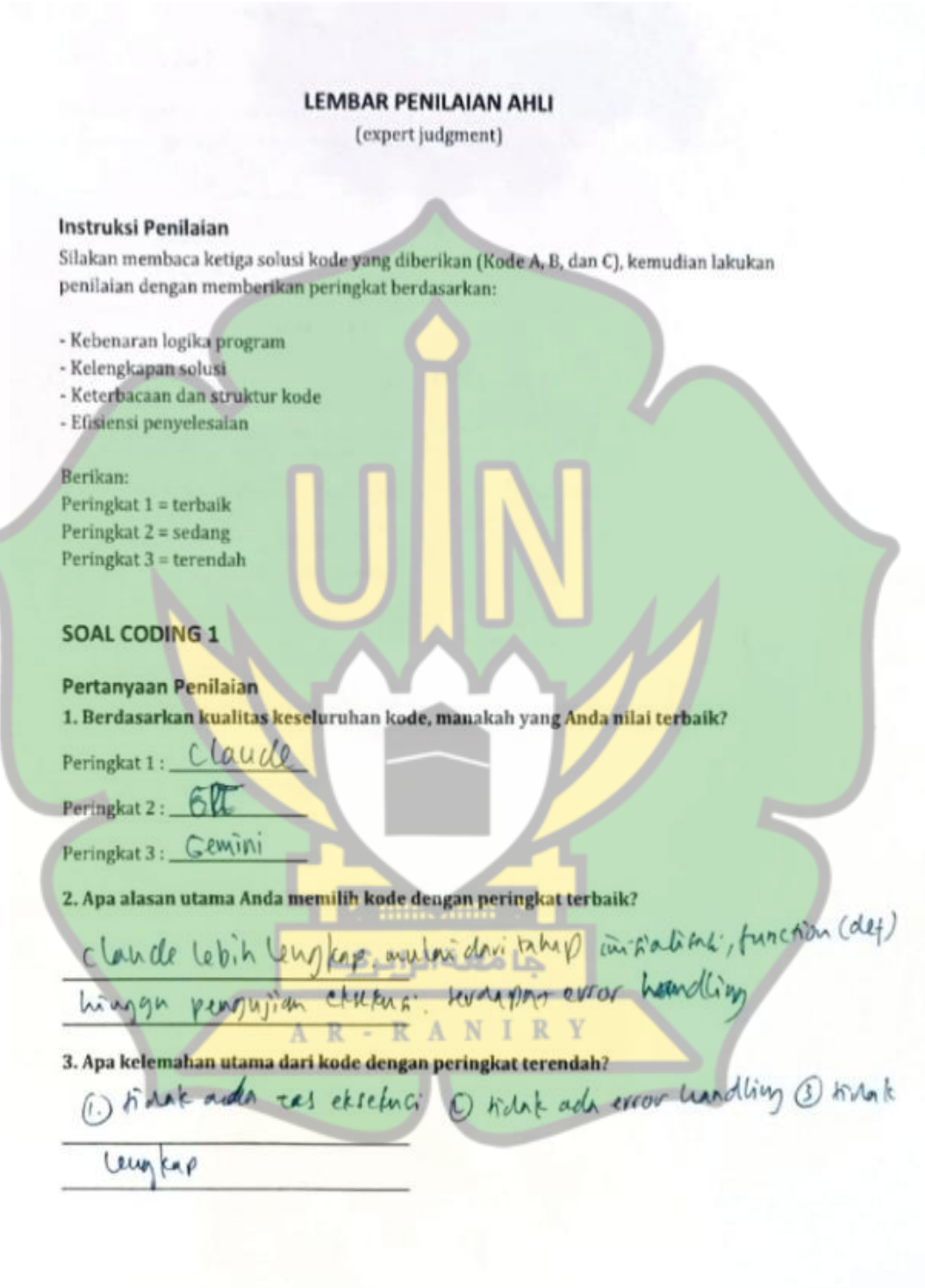
Peringkat 3 : cland

2. Apa alasan utama Anda memilih kode dengan peringkat terbaik?

lebih ringkas

3. Apa kelemahan utama dari kode dengan peringkat terendah?

C.2. Lembar Validasi Ahli 2



LEMBAR PENILAIAN AHLI

(expert judgment)

Instruksi Penilaian

Silakan membaca ketiga solusi kode yang diberikan (Kode A, B, dan C), kemudian lakukan penilaian dengan memberikan peringkat berdasarkan:

- Kebenaran logika program
- Kelengkapan solusi
- Keterbacaan dan struktur kode
- Efisiensi penyelesaian

Berikan:
Peringkat 1 = terbaik
Peringkat 2 = sedang
Peringkat 3 = terendah

SOAL CODING 1

Pertanyaan Penilaian

1. Berdasarkan kualitas keseluruhan kode, manakah yang Anda nilai terbaik?

Peringkat 1 : Claude

Peringkat 2 : GPT

Peringkat 3 : Gemini

2. Apa alasan utama Anda memilih kode dengan peringkat terbaik?

claude lebih lengkap, mulai dari tahap inisialisasi, function (def) hingga pengujian eksekusi. terdapat error handling

3. Apa kelemahan utama dari kode dengan peringkat terendah?

(1.) tidak ada tes eksekusi (2) tidak ada error handling (3) tidak lengkap

SOAL CODING 2

Pertanyaan Penilaian

1. Berdasarkan kualitas keseluruhan kode, manakah yang Anda nilai terbaik?

Peringkat 1 : Claude

Peringkat 2 : GPT

Peringkat 3 : Gemini

2. Apa alasan utama Anda memilih kode dengan peringkat terbaik?

Claude lebih lengkap, urutan langkahnya lebih, struktur nya teratur, selain nya ada penjelasan. ditambah cara penggunaan. ada beberapa contoh password

3. Apa kelemahan utama dari kode dengan peringkat terendah?

Belum ada contoh eksekusi ② belum ada error handling

SOAL CODING 3

Pertanyaan Penilaian

1. Berdasarkan kualitas keseluruhan kode, manakah yang Anda nilai terbaik?

Peringkat 1 : GPT

Peringkat 2 : Gemini

Peringkat 3 : Claude

2. Apa alasan utama Anda memilih kode dengan peringkat terbaik?

GPT penulisan sintaks python sesuai aturan. indentation rapi

terdapat cara penggunaan.

3. Apa kelemahan utama dari kode dengan peringkat terendah?

kesalahan ~~penulisan~~ indentation. error saat eksekusi

C.3. Lembar Validasi Ahli 3

6 Mei 2026

LEMBAR PENILAIAN AHLI

(expert judgment)

Instruksi Penilaian

Silakan membaca ketiga solusi kode yang diberikan (Kode A, B, dan C), kemudian lakukan penilaian dengan memberikan peringkat berdasarkan:

- Kebenaran logika program
- Kelengkapan solusi
- Keterbacaan dan struktur kode
- Efisiensi penyelesaian

Berikan:
Peringkat 1 = terbaik
Peringkat 2 = sedang
Peringkat 3 = terendah

SOAL CODING 1

Pertanyaan Penilaian

1. Berdasarkan kualitas keseluruhan kode, manakah yang Anda nilai terbaik?

Peringkat 1 : GPT

Peringkat 2 : Claude

Peringkat 3 : Gemini

2. Apa alasan utama Anda memilih kode dengan peringkat terbaik?

GPT lebih responsif dan menghasilkan kode yang benar/rapih

GPT lebih ringkas & padat serta mudah dibaca

3. Apa kelemahan utama dari kode dengan peringkat terendah?

lebih panjang dan rumit salah

SOAL CODING 2

Pertanyaan Penilaian

1. Berdasarkan kualitas keseluruhan kode, manakah yang Anda nilai terbaik?

Peringkat 1 : GPT

Peringkat 2 : Gemini

Peringkat 3 : Claude

2. Apa alasan utama Anda memilih kode dengan peringkat terbaik?

belum berhasil running tapi minim error

3. Apa kelemahan utama dari kode dengan peringkat terendah?

tidak berhasil running & error

SOAL CODING 3

Pertanyaan Penilaian

1. Berdasarkan kualitas keseluruhan kode, manakah yang Anda nilai terbaik?

Peringkat 1 : Gemini

Peringkat 2 : Grok

Peringkat 3 : GPT

2. Apa alasan utama Anda memilih kode dengan peringkat terbaik?

Error kurang lebih

3. Apa kelemahan utama dari kode dengan peringkat terendah?

terlalu kaya error