

## Analisis Efektivitas Dua Jenis Gaya Prompt dalam Model LLM Berbasis RAG

Muhammad Ainur Rizky<sup>1\*</sup>

<sup>1</sup>Teknik Informatika / Fakultas Teknik, Universitas Maarif Hasyim Latif

\*email: [ainur.rizky@scuderia-web.my.id](mailto:ainur.rizky@scuderia-web.my.id)

DOI: <https://doi.org/10.31603/komtika.v9i1.13488>

Received: 02-05-2025, Revised: 15-05-2025, Accepted: 18-05-2025

### ABSTRACT

*This study aims to analyse the effectiveness of two prompt styles, namely guided prompt and free prompt, in influencing the quality of answers generated by a Retrieval-Augmented Generation (RAG)-based Large Language Model (LLM) system using the META-Llama 3 model. The system is designed to answer questions based on reference documents stored in vector form through an embedding process. The research was conducted using questions formed in two versions of the prompt style, and the answer results were evaluated using two metrics ROUGE and BERTScore. The results showed that guided prompts resulted in higher scores on ROUGE-1, ROUGE-2, and ROUGE-L metrics reflecting a better level of precision and lexical agreement. Meanwhile, the BERTScore between the two prompt styles did not show any significant difference, meaning that in terms of meaning or semantic similarity, they provided relatively equivalent results. These findings suggest that prompt design has a real impact on the structure and precision of answers.*

**Keywords:** Large Language Model, Retrieval-Augmented Generation, ROUGE, BERTScore.

### ABSTRAK

Penelitian ini bertujuan untuk menganalisis efektivitas dua gaya prompt, yaitu *prompt* terbimbing dan *prompt* bebas, dalam memengaruhi kualitas jawaban yang dihasilkan oleh sistem *Large Language Model (LLM)* berbasis *Retrieval-Augmented Generation (RAG)* menggunakan model *META-Llama 3*. Sistem dirancang untuk menjawab pertanyaan berdasarkan dokumen referensi yang disimpan dalam bentuk vektor melalui proses *embedding*. Penelitian dilakukan dengan menggunakan pertanyaan yang dibentuk dalam dua versi gaya prompt, dan hasil jawaban dievaluasi menggunakan dua metrik *ROUGE* dan *BERTScore*. Hasil penelitian menunjukkan bahwa prompt terbimbing menghasilkan skor yang lebih tinggi pada metrik *ROUGE-1*, *ROUGE-2*, dan *ROUGE-L* yang mencerminkan tingkat presisi dan kesesuaian leksikal yang lebih baik. Sementara itu, nilai *BERTScore* antara kedua gaya *prompt* tidak menunjukkan perbedaan yang signifikan, yang berarti bahwa dari segi makna atau kesamaan semantik, keduanya memberikan hasil yang relatif setara. Temuan ini menunjukkan bahwa desain prompt memiliki dampak nyata terhadap struktur dan ketepatan jawaban.

**Keywords:** Model Bahasa Besar, Generasi Pengambilan-Penambahan, *ROUGE*, *BERTScore*

### PENDAHULUAN

Perkembangan teknologi kecerdasan buatan (AI) telah menghasilkan model bahasa besar (LLM) yang mampu menghasilkan teks dengan kualitas yang mendekati kemampuan manusia. salah satunya adalah *Large Language Model* yang disebut *Llama 3* yang dibuat oleh *Meta*. *Llama 3* merupakan kawanan model yang secara native mendukung multibahasa, pengkodean, penalaran, dan penggunaan alat [1]. Salah satu keunggulan utama LLM adalah pembelajaran dalam konteks, di mana model dilatih untuk menulis teks berdasarkan konteks atau perintah tertentu. Sementara itu, pendekatan NLP tradisional seperti chatbot berbasis

aturan tetap banyak digunakan dalam skenario yang lebih terstruktur, seperti layanan akademik [2]. Hal ini memungkinkan LLM untuk menghasilkan respons yang lebih koheren dan relevan secara kontekstual, sehingga cocok untuk aplikasi dan diskusi interaktif [3]. *Large Language Model (LLM)* secara umum memiliki kemampuan untuk menghasilkan teks yang jelas dan tampak faktual. Namun, informasi yang dihasilkan dapat mencakup ketidakakuratan faktual atau pernyataan yang tidak berdasarkan kenyataan. Karena LLM memiliki kesenjangan informasi dan menghasilkan output yang secara faktual tidak akurat, maka menambahkan informasi dari sumber pengetahuan eksternal dapat menjadi solusi yang menjanjikan untuk meningkatkan kinerja model bahasa tersebut [4]. Salah satu pendekatan dari LLM adalah implementasi *Retrieval-Augmented-Generation (RAG)*, RAG dapat menjembatani kesenjangan informasi tersebut dengan cara mencari dan mengkombinasikan pengetahuan dari informasi eksternal [5]. RAG dapat mengumpulkan informasi yang relevan terkait dengan pertanyaan pengguna. Informasi-informasi ini, digabungkan dengan pertanyaan asli, membentuk sebuah pertanyaan komprehensif yang memberdayakan LLM untuk menghasilkan jawaban yang terinformasi dengan baik [5]. Selain itu, dalam *Retrieval Augmented Generation (RAG)*, tidak perlu menyimpan pengetahuan secara implisit dalam parameter model LLM, tetapi dapat diperoleh secara eksplisit dengan gaya *plug-and-play*, sehingga menciptakan skalabilitas yang tinggi [6].

Meskipun sistem RAG memiliki potensi yang besar, kualitas teks yang dihasilkan sangat bergantung pada *prompt* yang diberikan sebagai input awal. *Prompt* digunakan untuk menentukan bagaimana model merespons, dan perubahan kecil pada *prompt* dapat memengaruhi struktur, relevansi, dan keakuratan jawaban yang dihasilkan. Secara khusus, *prompt* menentukan konteks percakapan dan memberi tahu LLM informasi apa yang penting serta bentuk dan konten keluaran yang diinginkan. Wei dkk.[7] menunjukkan bahwa penelitian yang menunjukkan bahwa *prompt* dapat meningkatkan kinerja pada berbagai tugas aritmatika, akal sehat, dan penalaran simbolik.

Penelitian ini bertujuan untuk menganalisis pengaruh dua gaya penulisan *prompt* yaitu *prompt* terbimbing dan *prompt* bebas terhadap performa sistem *Large Language Model (LLM)* Llama 3 berbasis *Retrieval-Augmented Generation (RAG)*. Dalam sistem RAG, peran *prompt* sangat penting karena menjadi jembatan antara dokumen yang diambil dari basis data dan jawaban yang dihasilkan oleh model. *Prompt* terbimbing biasanya memiliki struktur instruksi yang jelas dan formal, sedangkan *prompt* bebas cenderung bersifat natural dan lebih menyerupai pertanyaan sehari-hari.

Melalui eksperimen ini, penelitian mengevaluasi kualitas teks hasil generasi menggunakan dua gaya *prompt* tersebut dengan metrik evaluasi otomatis seperti *ROUGE* dan *BERTScore*. Dengan membandingkan hasil dari kedua pendekatan, penelitian ini diharapkan dapat memberikan wawasan tentang desain gaya penulisan *prompt* yang lebih efektif dalam meningkatkan relevansi, ketepatan, dan kejelasan jawaban LLM dalam konteks pengambilan informasi berbasis dokumen.

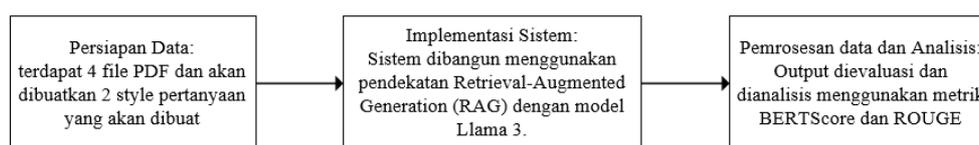
## METODE

Metodologi penelitian ini mencakup pembahasan mengenai desain penelitian, dataset atau dokumen yang digunakan sebagai sumber referensi dalam sistem *Retrieval-Augmented*

Generation (RAG), penerapan metrik evaluasi, serta implementasi model dan keseluruhan proses sistem yang digunakan.

### Rancangan Penelitian

Penelitian ini menggunakan metode kuantitatif dengan pendekatan eksperimental untuk mengeksplorasi pengaruh variasi prompt terhadap hasil evaluasi metrik *ROUGE* dan *BERTScore* pada sistem *Retrieval-Augmented Generation (RAG)* di dalam *Large Language Model (LLM)*. Dalam implementasinya, sebuah aplikasi dikembangkan untuk mempermudah proses pengumpulan dan evaluasi data berdasarkan metrik *BERTScore* dan *ROUGE*. Alur penelitian pada Gambar 1 berisi informasi terkait alur perancangan penelitian ini secara umum.



Gambar 1. Alur Penelitian

### Dataset

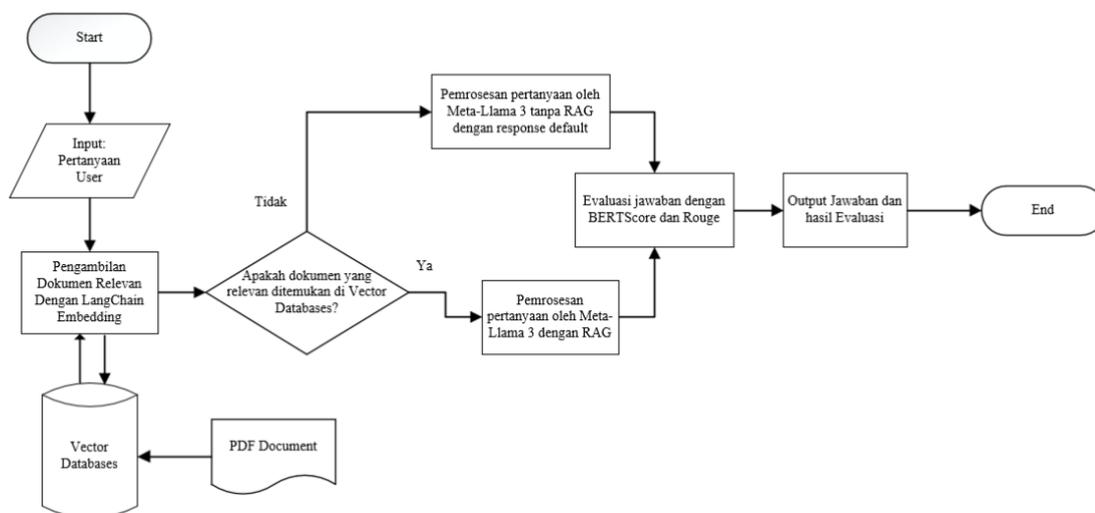
*Dataset* yang digunakan dalam penelitian ini berupa berbagai dokumen PDF dengan teks berbahasa Indonesia. dokumen yang digunakan juga bermacam-macam, ini digunakan untuk mencakup berbagai gaya penulisan, tingkat kompleksitas, dan konteks isi.

### Implementasi Sistem

Tujuan implementasi ini adalah untuk membangun sistem yang mengintegrasikan model *Large Language Model (LLM) Llama 3* berukuran 8 miliar parameter dengan pendekatan *Retrieval-Augmented Generation (RAG)*. Sistem ini dirancang untuk memproses *input* pertanyaan pengguna dan menghasilkan jawaban berbasis informasi yang diambil dari dokumen referensi.

Sistem yang diimplementasikan menggunakan pendekatan *Retrieval-Augmented Generation (RAG)*. Sistem ini dirancang untuk menjawab pertanyaan pengguna dengan memanfaatkan dokumen referensi. Dokumen tersebut akan berupa file *PDF* atau *text files* yang lainnya yang akan diolah dan disimpan dalam bentuk *Vector Databases* menggunakan *ChromaDB*, yang merupakan salah satu penyimpanan vektor yang didukung oleh *library LangChain*. *Vector Databases* sendiri adalah jenis basis data yang menyimpan data dalam bentuk vektor dimensi tinggi dengan representasi matematis matematis yang mewakili fitur atau atribut. Setiap vektor memiliki dimensi tertentu, yang bervariasi dari puluhan hingga ribuan, tergantung pada detail dan kompleksitas data [8]. Dalam proses penyimpanannya diperlukan yang namanya *embedding* yang merupakan proses mengubah setiap kata atau frasa dari kosakata menjadi vektor, sehingga terbentuk hubungan semantik antara kata-kata [9]. Untuk menghasilkan representasi vektor dari dokumen, proses *embedding* dilakukan melalui *LangChain* yang merupakan *framework python* sumber terbuka untuk membangun aplikasi

berbasis LLM yang memungkinkan pengembang menghubungkan model bahasa dengan sumber data dan layanan eksternal [10]. Ketika pengguna mengajukan pertanyaan, sistem melakukan proses retrieval untuk menemukan potongan kata (*token*) dalam dokumen dalam bentuk konteks kecil yang sesuai dengan pertanyaan yang telah disimpan dalam *Vector Databases*, Proses ini dikenal sebagai *chunking* yang merupakan proses membagi dokumen menjadi bagian teks yang lebih kecil agar dapat disesuaikan dengan pertanyaan yang diajukan pengguna saat mencari informasi [11].



Gambar 2. *Flowchart* Sistem

Sebuah *prompt* dibuat dengan menggabungkan hasil pencarian dari *Vector Databases*. *Prompt* gabungan ini dirancang untuk memberikan konteks dan panduan bagi model untuk menghasilkan respons [12]. Metode ini memungkinkan model menggunakan data yang kaya dan terarah untuk membuat respons yang tidak hanya relevan tetapi juga akurat dan berkualitas tinggi.

Model yang digunakan adalah *Meta Llama 3 8B*, model ini digunakan untuk menghasilkan jawaban berdasarkan input yang telah disesuaikan. Model akan ini menghasilkan response yang spesifik dan relevan sesuai konteks. pada implementasi ini model *Meta Llama 3* dijalankan secara lokal pada sistem, sehingga tidak memerlukan koneksi internet untuk menjalankan proses inference. Ini memungkinkan sistem untuk tetap berjalan dalam lingkungan jaringan yang terbatas dan seluruh proses pengolahan pertanyaan dan dokumen referensi dilakukan secara lokal tanpa harus mengirimkan data ke server eksternal. untuk menjalankan model ini secara lokal dibutuhkan *framework Ollama*. *Ollama* adalah *framework* untuk menjalankan banyak jenis model *Large Language Model* secara lokal tanpa perlu bergantung pada model secara *cloud* [13].

### Matrik Evaluasi

Pada evaluasi metrik ini digunakan untuk menentukan kelayakan jawaban sistem yang telah dikembangkan. evaluasi ini dilakukan untuk melihat kecocokan antara jawaban dengan referensi data yang diambil dari proses *retrieval* menggunakan matrik *BERTScore* dan *ROUGE* untuk mengukur kualitas jawaban dari model *Llama 3*.

### 1. BERTScore

BERTScore merupakan matrik evaluasi otomatis untuk generatif text dengan menghitung skor kemiripan untuk setiap *token* dalam kalimat prediksi dengan kalimat referensi. BERTScore ini memproses kemiripan token dengan menghitung kesamaan antara keduanya dengan proses *embedding*. Metrik ini menggunakan model bahasa BERT (*Bidirectional Encoder Representations from Transformers*) untuk mengumpulkan representasi teks yang mendalam. Kemudian, untuk mengevaluasi kalimat potensial dengan mengacu pada kalimat referensi, mereka menghitung kesamaan antara representasi teks tersebut [14], [15]. Dalam BERTScore memiliki nilai dari 0 sampai 1, nilai 1 menunjukkan nilai yang mendekati dari dua kalimat, sedangkan nilai 0 menunjukkan tidak mendekati dari dua kalimat [16]. Berikut ini adalah persamaan untuk menghitung BERTScore [15]:

a. Precision ( $P_{BERT}$ )

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max x_i^\top \hat{x}_j \quad (1)$$

b. Recall ( $R_{BERT}$ )

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max x_i^\top \hat{x}_j \quad (2)$$

c. F1-score ( $F_{BERT}$ )

$$F_{BERT} = 2 \cdot \frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}} \quad (3)$$

### 2. ROUGE

ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) adalah matrik yang dirancang untuk mengevaluasi ringkasan text secara otomatis. ROUGE sendiri menerima ringkasan text yang dihasilkan sistem dan ringkasan model serta memperkirakan konten yang dibagikan dengan menghasilkan skor yang berbeda [17]. ROUGE menentukan kualitas ringkasan otomatis dengan membandingkan unit yang tumpang tindih seperti n-gram, urutan kata, dan pasangan kata dengan ringkasan yang ditulis oleh manusia.

ROUGE-N terdiri dari skor *precision*, *recall*, dan *F1*, dengan nilai N adalah 1, 2 dan L. Nilai 1 menghitung satu kata (*unigram*) dan nilai 2 menghitung dua kata yang berhubungan (*bigram*) yang ada di kedua dokumen yang dibandingkan. ROUGE-N bernilai L adalah *Longest Common Subsequence (LCS)*, yaitu kata-kata berurutan terpanjang yang ada di kedua dokumen yang dibandingkan. berikut persamaan untuk menghitung ROUGE [18]:

a. Precision

$$ROUGE\ 1\ Precision = \frac{\text{Jumlah unigram katasama}}{\text{keseluruhan kata ringkasan sistem}} \quad (4)$$

$$ROUGE\ 2\ Precision = \frac{\text{Jumlah bigram katasama}}{\text{keseluruhan kata ringkasan sistem}} \quad (5)$$

$$ROUGE\ L\ Precision = \frac{\text{Longest Common Subsequence (LCS)}}{\text{keseluruhan kata ringkasan sistem}} \quad (6)$$

b. Recall

$$ROUGE\ 1\ Recall = \frac{\text{Jumlah unigram yang sama}}{\text{keseluruhan kata ringkasan text manusia}} \quad (7)$$

$$ROUGE\ 2\ Recall = \frac{\text{Jumlah bigram yang sama}}{\text{keseluruhan kata ringkasan text manusia}} \quad (8)$$

$$ROUGE\ L\ Recall = \frac{\text{Longest Common Subsequence (LCS)}}{\text{keseluruhan kata ringkasan text manusia}} \quad (9)$$

c. F1-score

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

## HASIL DAN PEMBAHASAN

### Persiapan

Data yang dipakai dalam pengetesan ini memakai dokumen PDF yang didapatkan dari berbagai sumber media *online*, baik itu berjenis artikel atau *e-book*. data ini dipakai untuk menilai sumber jawaban dan referensi untuk hasil hasil evaluasi matrik *BERTScore* dan *ROGUE*. Setiap dokumen dilengkapi dengan beberapa pertanyaan yang dibuat untuk mengukur kemampuan *llama 3* dalam memahami dan menjawab berdasarkan informasi yang terkandung dalam dokumen tersebut.

Selain itu, penelitian ini menggunakan 2 jenis gaya pertanyaan yang dirancang untuk menguji performa setiap *prompt* atau pertanyaa dalam menghasilkan respons. *Prompt* yang digunakan meliputi:

Tabel 1. Sampel Pertanyaan

Prompt Bebas	Prompt Terbimbing
Jelaskan aturan dasar dalam penulisan kode Python	Jelaskan secara sistematis aturan dasar dalam penulisan kode Python, termasuk penamaan variabel, indentasi, komentar, dan struktur umum penulisan program.
Bagaimana dampak privatisasi terhadap kinerja keuangan BUMN?	Uraikan dampak privatisasi terhadap kinerja keuangan BUMN, khususnya terhadap rasio keuangan seperti ROE (Return on Equity), ROA (Return on Assets), dan ROS (Return on Sales).
Bagaimana Kurikulum Merdeka berbeda dari kurikulum tradisional di Indonesia?	Bandingkan Kurikulum Merdeka dengan kurikulum tradisional di Indonesia, terutama dalam hal pendekatan pembelajaran, fleksibilitas, dan peran guru serta siswa.
Apa fungsi dari Jupyter Notebook?	Jelaskan fungsi utama dari Jupyter Notebook dalam pengembangan dan dokumentasi program Python, serta manfaatnya dalam dunia penelitian.

Dan berikut adalah sample jawaban dari ke dua gaya penulisan prompt dari pertanyaan berikut:

### 1. “Jelaskan aturan dasar dalam penulisan kode Python”

Response:

Aturan dasar dalam penulisan kode Python adalah:

- Menggunakan tanda pagar (#) untuk menulis komentar.
- Menggunakan indentasi (spasi atau tab) untuk menandai blok kode.
- Menggunakan enter untuk memisahkan baris program.
- Menggunakan tanda semikolon (;) untuk memisahkan baris program yang tidak perlu diakhiri dengan enter.

Gambar 3. Sample jawaban *Prompt* Bebas

### 2. “Jelaskan secara sistematis aturan dasar dalam penulisan kode Python, termasuk penamaan variabel, indentasi, komentar, dan struktur umum penulisan program”

Dalam penulisan kode Python, ada beberapa aturan dasar yang perlu diikuti untuk menjaga kualitas kode dan meningkatkan kerja sama dalam pengembangan perangkat lunak. Berikut adalah aturan dasar tersebut:

1. **Indentasi:** Indentasi digunakan untuk menandai blok kode dalam Python. Indentasi harus konsisten dan sesuai dengan struktur program. Jika indentasi tidak konsisten atau tidak sesuai, maka program tidak akan berjalan dengan benar atau dapat menyebabkan kesalahan sintaks.
2. **Komentar:** Komentar digunakan untuk memberikan informasi tambahan tentang kode. Dalam Python, komentar ditulis menggunakan tanda pagar (#) diikuti oleh teks komentar.
3. **Penamaan Variabel:** Penamaan variabel tidak perlu secara eksplisit dalam Python. Variabel akan dideklarasikan secara otomatis ketika digunakan. Namun, penamaan variabel harus jelas dan menggambarkan konten yang disimpan di dalamnya.
4. **Struktur Umum Penulisan Program:** Struktur umum penulisan program pada Python terdiri dari beberapa bagian, yaitu:
  - **Blok Kode:** Blok kode adalah sekumpulan pernyataan yang dieksekusi bersama sebagai satu unit.
  - **Baris Program:** Baris program tidak memerlukan semicolon (“titik koma”), cukup menekan enter untuk membedakan baris satu dengan baris berikutnya.

Dengan mengikuti aturan dasar ini, Anda dapat meningkatkan kualitas kode Python dan memastikan bahwa kode Anda dapat dijalankan dengan benar tanpa kesalahan sintaks.

Gambar 4. Sample Jawaban *Prompt* Terbimbing

## Pengujian dan hasil

Hasil pengujian akan mencakup skor evaluasi menggunakan metrik *BERTScore* dan *ROUGE*. Setiap gaya penulisan yang diuji akan dinilai secara otomatis yang akan menghitung nilai kedua metrik. Setiap gaya penulisan yang diuji akan diberi skor berdasarkan kedua metrik tersebut, yang kemudian akan dibandingkan untuk menilai kualitas respons secara keseluruhan. Perbandingan ini memungkinkan analisis mendalam tentang kinerja dalam menghasilkan respons yang sesuai dengan referensi yang telah ditetapkan.

Evaluasi ini bertujuan untuk mengukur sejauh mana respon yang dihasilkan mendekati referensi yang diharapkan, baik dari segi kesamaan semantik maupun kesesuaian struktur teks. Dengan demikian, hasil evaluasi dapat digunakan untuk mengidentifikasi kekuatan dan juga area yang perlu ditingkatkan, sehingga kualitas output aplikasi dapat terus ditingkatkan.

Tabel 2. Hasil Evaluasi Gaya Jawaban Bebas

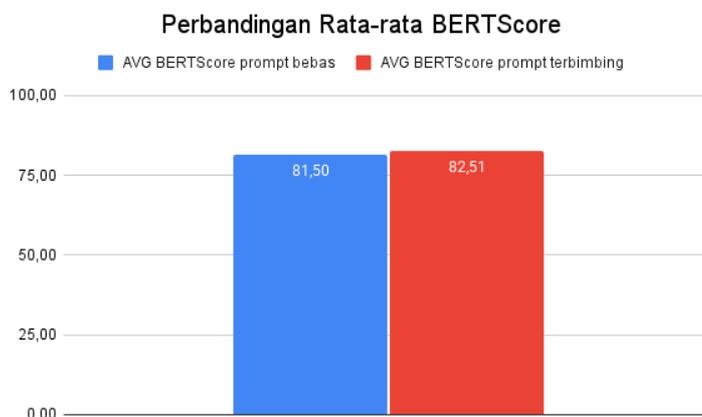
No.	BERTScore			ROUGE								
				ROUGE-1			ROUGE-2			ROUGE-L		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
1	89,48	82,08	85,62	30,57	79,55	44,16	17,11	44,83	24,76	16,16	42,05	23,34
2	89,59	76,94	82,78	7,48	94,12	13,85	5,63	75,00	10,48	7,01	88,24	12,99
3	88,39	79,13	83,50	7,98	89,47	14,66	4,72	55,56	8,70	6,57	73,68	12,07
4	80,58	73,17	76,70	2,44	66,67	4,71	1,22	37,50	2,37	2,03	55,56	3,92
5	81,56	78,89	80,20	8,09	73,33	14,57	2,58	24,14	4,67	5,88	53,33	10,60
6	88,20	75,76	81,51	6,53	100,00	12,26	6,15	100,00	11,58	5,71	87,50	10,73
7	82,89	79,99	81,41	18,11	32,84	23,34	1,65	3,01	2,13	9,05	16,42	11,67
8	85,26	76,63	80,72	4,27	54,55	7,92	1,79	23,81	3,32	3,56	45,45	6,60
9	87,59	78,73	82,92	8,5	80,77	15,38	4,47	44,00	8,12	8,1	76,92	14,65
10	85,59	74,38	79,60	4,85	91,67	9,21	3,98	81,82	7,59	4,85	91,67	9,21
AVG	85,91	77,57	81,50	9,88	76,30	16,01	4,93	48,97	8,37	6,89	63,08	11,58

Tabel 2 menunjukkan hasil evaluasi performa pada *prompt* 1 dengan menggunakan matriks *BERTScore* dan *ROUGE*. Terlihat bahwa hasil yang didapatkan menunjukkan performa model secara keseluruhan dengan persentase *BERTScore F1* 81.50%, Sedangkan untuk *ROUGE*, hasil *F1 ROUGE-1* sebesar 16.01%, *F1 ROUGE-2* sebesar 8.37%, dan *F1 ROUGE-L* sebesar 11.58%.

Tabel 3. Hasil Evaluasi Gaya Jawaban Terbimbing

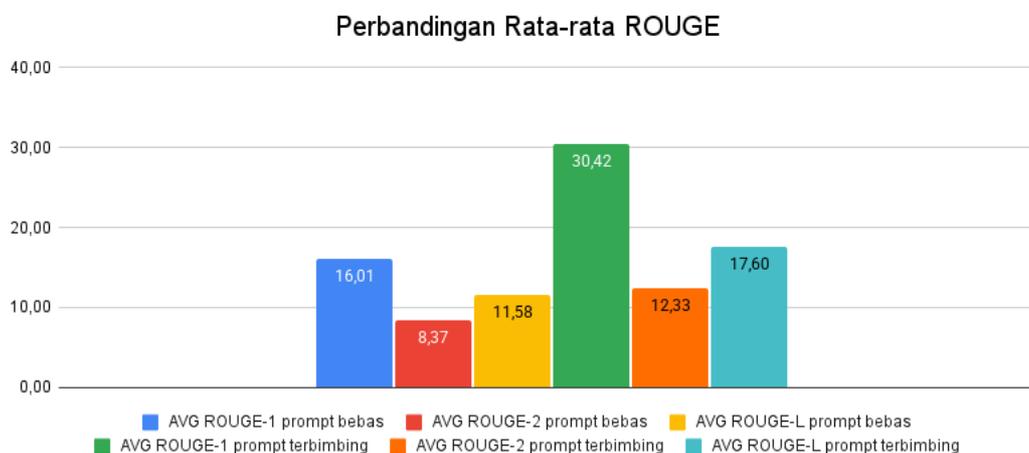
No.	BERTScore			ROUGE								
				ROUGE-1			ROUGE-2			ROUGE-L		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
1	85,14	82,66	83,88	41,92	38,71	40,25	14,47	13,36	13,89	17,47	16,13	16,77
2	88,73	78,21	83,14	14,02	85,71	24,10	8,45	52,94	14,57	9,81	60,00	16,87
3	84,89	80,08	82,42	8,92	45,24	14,90	2,36	12,20	3,95	5,63	28,57	9,41
4	87,01	80,42	83,59	14,63	80,00	24,74	6,12	34,09	10,38	8,13	44,44	13,75
5	79,01	80,37	79,69	20,96	66,28	31,84	9,95	30,59	14,61	15,44	48,84	23,46
6	84,40	79,23	81,74	25,31	29,21	33,42	10,66	20,80	14,09	14,29	27,78	18,87
7	84,43	78,76	81,49	8,64	42,86	14,38	2,07	10,42	3,45	6,58	32,65	10,96
8	84,49	80,31	82,56	20,64	51,79	29,52	6,79	17,12	9,72	10,68	26,79	15,27
9	84,29	81,16	82,70	41,30	48,57	44,64	14,23	16,75	15,38	19,43	22,86	21,01
10	84,83	83,03	83,92	51,10	42,49	46,40	25,66	21,32	23,29	32,60	27,11	29,6
AVG	84,72	80,42	82,51	24,74	53,09	30,42	10,08	22,96	12,33	14,01	33,52	17,60

Pada data Tabel 3, rata-rata hasil evaluasi mengalami peningkatan dibandingkan dengan *prompt* 1. Dapat dilihat bahwa persentase hasil *ROUGE-1 F1* sebesar 30.42%, *ROUGE-2 F1* sebesar 12.33%, dan *ROUGE-L F1* sebesar 17.60%. dan untuk *BERTScore F1* 82.51%.



Gambar 5. Diagram Rata-rata *BERTScore*

Dari diagram terlihat bahwa nilai rata-rata *BERTScore* untuk *prompt* terbimbing adalah 82,51, sedikit lebih tinggi dibandingkan *prompt* bebas yang memperoleh nilai sebesar 81,50. Selisih skor yang relatif kecil ini menunjukkan bahwa meskipun terdapat perbedaan struktur dan gaya penulisan pada masing-masing jenis *prompt*, makna atau isi utama dari respons yang dihasilkan oleh sistem tetap berada dalam tingkat kesamaan semantik yang cukup dekat. Artinya, baik *prompt* terbimbing maupun *prompt* bebas mampu menghasilkan jawaban yang secara makna tidak jauh berbeda. Namun demikian, *prompt* terbimbing menunjukkan performa yang lebih konsisten.



Gambar 6. Diagram Rata-rata *ROUGE*

Pada metrik *ROUGE-1* dengan *prompt* terbimbing memperoleh nilai sebesar 30,42, sedangkan *prompt* bebas hanya mencapai 16,01. Hal ini mengindikasikan bahwa *prompt* terbimbing mampu menghasilkan teks yang lebih presisi dalam hal kesesuaian kata-kata tunggal (*unigram*) dengan referensi. Sementara itu, pada metrik *ROUGE-2* yang mengukur kesesuaian dua kata berurutan (*bigram*), *prompt* terbimbing juga unggul dengan skor 12,33 dibandingkan *prompt* bebas yang memperoleh nilai 8,37. Ini menunjukkan bahwa struktur lokal dalam kalimat, seperti kombinasi dua kata, lebih terjaga dalam respons yang dihasilkan

oleh prompt terbimbing. Dan hasil pada metrik *ROUGE-L*, di mana *prompt* terbimbing juga mencatat skor lebih tinggi yaitu 17,60 dibandingkan *prompt* bebas yang hanya mencapai 11,58.

## KESIMPULAN

Berdasarkan hasil penelitian menunjukkan bahwa penggunaan *prompt* terbimbing memberikan hasil yang lebih unggul dalam hal ketepatan leksikal dan struktur lokal teks. Hal ini dibuktikan dengan nilai *ROUGE-1*, *ROUGE-2* dan *ROUGE-L* yang lebih tinggi dibandingkan dengan *prompt* bebas. *Prompt* terbimbing mampu menghasilkan respons yang lebih presisi terhadap teks referensi, terutama dalam kesesuaian kata-kata dan pasangan kata yang digunakan. Sementara itu, hasil perbandingan nilai BERTScore menunjukkan bahwa perbedaan antara kedua gaya prompt relatif kecil, dengan prompt terbimbing sedikit lebih unggul. Hal ini mengindikasikan bahwa secara semantik, kedua jenis prompt mampu menghasilkan jawaban yang memiliki tingkat kesamaan makna yang cukup tinggi dengan referensi. Dengan demikian, pemilihan gaya prompt dapat disesuaikan dengan kebutuhan aplikasi; prompt terbimbing lebih cocok untuk menghasilkan jawaban yang spesifik dan terarah, sedangkan prompt bebas lebih sesuai jika diutamakan fleksibilitas bahasa dan struktur yang natural.

## DAFTAR PUSTAKA

- [1] A. Grattafiori *et al.*, “The Llama 3 Herd of Models,” Nov. 23, 2024, *arXiv*: arXiv:2407.21783. doi: 10.48550/arXiv.2407.21783.
- [2] D. Wicaksono, D. J. N. Salim, and D. P. Almeyda, “Pengembangan Layanan dan Pengenalan Akademik di Lingkungan Kampus berbasis Aplikasi Chatbot bagi Calon Mahasiswa Baru,” *JKKI*, vol. 8, no. 1, pp. 1–13, May 2024, doi: 10.31603/komtika.v8i1.11164.
- [3] Y. Chang *et al.*, “A Survey on Evaluation of Large Language Models,” *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, pp. 1–45, Jun. 2024, doi: 10.1145/3641289.
- [4] Z. Jiang *et al.*, “Active Retrieval Augmented Generation,” Oct. 22, 2023, *arXiv*: arXiv:2305.06983. doi: 10.48550/arXiv.2305.06983.
- [5] Y. Gao *et al.*, “Retrieval-Augmented Generation for Large Language Models: A Survey,” Mar. 27, 2024, *arXiv*: arXiv:2312.10997. doi: 10.48550/arXiv.2312.10997.
- [6] H. Li, Y. Su, D. Cai, Y. Wang, and L. Liu, “A Survey on Retrieval-Augmented Text Generation,” Feb. 13, 2022, *arXiv*: arXiv:2202.01110. doi: 10.48550/arXiv.2202.01110.
- [7] J. Wei *et al.*, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” Jan. 10, 2023, *arXiv*: arXiv:2201.11903. doi: 10.48550/arXiv.2201.11903.
- [8] Y. Han, C. Liu, and P. Wang, “A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge,” Oct. 18, 2023, *arXiv*: arXiv:2310.11703. doi: 10.48550/arXiv.2310.11703.
- [9] T. I. Z. M. Putra, S. Suprpto, and A. F. Bukhori, “Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Indobert Embedding dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia,” *JISTED*, vol. 1, no. 1, pp. 1–28, Nov. 2022, doi: 10.35912/jisted.v1i1.1509.

- [10] B. Auffarth, *Generative AI with LangChain: Build Large Language Model (LLM) Apps with Python, ChatGPT, and Other LLMs*, 2024th ed. Birmingham, UK: Packt Publishing Ltd, 2023.
- [11] O. Topsakal and T. C. Akinci, “Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast,” *ICAENS*, vol. 1, no. 1, pp. 1050–1056, Jul. 2023, doi: 10.59287/icaens.1127.
- [12] H. Tohir, N. Merlina, and M. Haris, “Utilizing Retrieval-Augmented Generation In Large Language Models To Enhance Indonesian Language NLP,” *jitk*, vol. 10, no. 2, pp. 352–360, Nov. 2024, doi: 10.33480/jitk.v10i2.5916.
- [13] J. B. Gruber and M. Weber, “rollama: An R package for using generative large language models through Ollama,” Apr. 11, 2024, *arXiv*: arXiv:2404.07654. doi: 10.48550/arXiv.2404.07654.
- [14] T. Sun, J. He, X. Qiu, and X. Huang, “BERTScore is Unfair: On Social Bias in Language Model-Based Metrics for Text Generation,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 3726–3739. doi: 10.18653/v1/2022.emnlp-main.245.
- [15] M. Hanna and O. Bojar, “A Fine-Grained Analysis of BERTScore,” *Association for Computational Linguistics*, vol. Proceedings of the Sixth Conference on Machine Translation, pp. 507–517, Nov. 2021.
- [16] F. V. P. Samosir, H. Toba, and M. Ayub, “BESKlus : BERT Extractive Summarization with K-Means Clustering in Scientific Paper,” *JuTISI*, vol. 8, no. 1, Apr. 2022, doi: 10.28932/jutisi.v8i1.4474.
- [17] M. Moradi, M. Dashti, and M. Samwald, “Summarization of biomedical articles using domain-specific word embeddings and graph ranking,” *Journal of Biomedical Informatics*, vol. 107, p. 103452, Jul. 2020, doi: 10.1016/j.jbi.2020.103452.
- [18] Z. Idhafi, S. Agustian, F. Yanto, and N. Safaat H, “Peringkat teks otomatis pada artikel berbahasa indonesia menggunakan metode maximum marginal relevance,” *CoSciTech*, vol. 4, no. 3, pp. 609–618, Dec. 2023, doi: 10.37859/coscitech.v4i3.6311.

