Perbandingan Kinerja Sequential Processing dan Routing Questioning Dalam Optimasi RAG Untuk Domain Spesifik

1st Muhammad Hanafi Choirulloh
Universitas Telkom Surabaya
Fakultas Informatika
Surabaya, Indonesia
hanafichoi@student.telkomuniversity.ac.id

2nd Alqis Rausanfita, S.Kom., M.Kom.
Universitas Telkom Surabaya
Fakultas Informatika
Surabaya, Indonesia
alqisfita@telkomuniversity.ac.id

3rd Daud Muhajir, S.Kom., M.Kom.
Universitas Telkom Surabaya
Fakultas Informatika
Surabaya, Indonesia
daudmuhajir@telkomuniversity.ac.id

***Abstrak* — Penelitian ini mengevaluasi dua pendekatan arsitektur dalam sistem Retrieval-Augmented Generation (RAG) untuk domain spesifik kesehatan: sequential processing dan routing questioning. Eksperimen dilakukan menggunakan data obat dari MIMS Indonesia dengan melibatkan model domain spesifik (Bio-Mistral) dan model generatif umum (GPT-5). Hasil penelitian menunjukkan bahwa pendekatan routing questioning secara signifikan mengungguli sequential processing dalam hal efisiensi dan kualitas luaran. Routing questioning mencatat latensi rata-rata yang jauh lebih rendah (16,33 detik) dibandingkan sequential processing (53,69 detik) serta unggul dalam metrik evaluasi F1, ROUGE, semantic similarity, dan groundedness. Temuan ini mengindikasikan bahwa mekanisme klasifikasi kueri lebih efektif daripada pemrosesan linear bertingkat dalam optimasi sistem RAG**

Kata kunci*— *Retrieval-Augmented Generation (RAG), Sequential Processing, Routing Questions, Akurasi, Kecepatan Pemrosesan, Optimasi Kinerja, Domain Khusus, Teknologi AI, Pembelajaran Mesin, Sistem Pencarian dan Generasi, Refleksi Pemrosesan Bertahap, Pengurangan Waktu Pemrosesan

I. PENDAHULUAN

Perkembangan Artificial Intelligence (AI), khususnya dalam Natural Language Processing (NLP) dan Generative AI, telah memungkinkan pemrosesan teks yang adaptif. Namun, bagi perusahaan di sektor spesifik seperti kesehatan (Perusahaan X), penggunaan model bahasa generik sering kali tidak memadai karena risiko halusinasi dan kurangnya pengetahuan mendalam mengenai prosedur medis atau regulasi farmasi. Untuk mengatasi hal ini, metode Retrieval-Augmented Generation (RAG) diterapkan guna mengintegrasikan sumber data eksternal yang relevan [1].

Terdapat dua pendekatan arsitektur utama dalam implementasi RAG yang diteliti, yaitu Sequential Processing dan Routing Questioning. Sequential Processing melibatkan proses berurutan di mana model domain khusus memproses informasi terlebih dahulu sebelum disempurnakan oleh model umum. Sebaliknya, Routing Questioning menggunakan mekanisme klasifikasi untuk mendeteksi relevansi pertanyaan; jika pertanyaan bersifat medis, sistem akan merutekan ke model khusus, dan jika umum, ke model general.

Penelitian ini bertujuan untuk membandingkan kinerja kedua metode tersebut guna menentukan pendekatan yang paling efisien dan akurat untuk diterapkan pada domain kesehatan,. Hipotesis penelitian adalah integrasi retrieval spesifik akan meningkatkan akurasi, namun perbedaan arsitektur pemrosesan akan berdampak signifikan pada latensi dan kualitas jawaban.

II. KAJIAN TEORI

A. Retrieval-Augmented Generation (RAG)

RAG adalah kerangka kerja yang menggabungkan kemampuan pencarian informasi (retrieval) dengan kemampuan generatif model bahasa. Sistem ini mengatasi keterbatasan pengetahuan statis pada Large Language Models (LLM) dengan memungkinkan akses ke data terkini dari sumber eksternal sebelum menghasilkan respons. Dengan cara ini, RAG tidak hanya bergantung pada pengetahuan internal yang ada dalam model, tetapi juga dapat menarik informasi terkini yang relevan dari sumber eksternal, seperti basis data atau dokumen yang terus diperbarui [2].

Proses ini dimulai dengan mengubah kueri (pertanyaan) pengguna menjadi vektor (embedding) yang merepresentasikan informasi dari pertanyaan tersebut. Vektor ini kemudian dicocokkan dengan basis data vektor untuk menemukan dokumen yang relevan yang dapat memberikan konteks bagi model generatif. Dengan adanya akses ke data eksternal, RAG memungkinkan model untuk memberikan jawaban yang lebih akurat, berbasis fakta terkini, dan lebih adaptif dibandingkan dengan model yang hanya bergantung pada pengetahuan yang telah ditanamkan selama pelatihan [3].

B. Sequential Processing

Sequential Processing dalam RAG merujuk pada eksekusi bertahap di mana setiap model memiliki tugas spesifik secara berurutan. Proses ini dimulai dengan embedding, dilanjutkan dengan retrieval, dan akhirnya generation. Pendekatan ini memberikan kontrol ketat pada setiap langkah, di mana hasil dari model spesifik (misalnya Bio-Mistral) digunakan sebagai konteks untuk model generatif umum (seperti GPT-5) guna menghasilkan teks yang lebih alami dan relevan [4].

Namun, meskipun pendekatan ini memberikan kontrol yang lebih ketat, ketergantungan antar-tahap ini berpotensi meningkatkan latensi sistem. Setiap langkah harus diproses secara berurutan, yang dapat menyebabkan penundaan jika

salah satu tahap memakan waktu lebih lama daripada yang lain. Misalnya, jika proses retrieval memerlukan waktu yang lama untuk menemukan dokumen yang relevan, hal ini dapat memperlambat keseluruhan sistem, terutama untuk aplikasi yang membutuhkan respons waktu nyata.

C. Routing Questioning

Pendekatan Routing Questioning melibatkan klasifikasi awal terhadap pertanyaan pengguna untuk menentukan jalur pemrosesan yang paling tepat. Dalam pendekatan ini, kueri dibagi menjadi kategori-kategori yang lebih spesifik, seperti "Medis" atau "Umum". Dengan cara ini, sistem dapat mengalokasikan sumber daya komputasi secara lebih efisien dan memilih model yang paling tepat berdasarkan kategori kueri yang diajukan [5].

Jika kueri terdeteksi sebagai medis, sistem akan mengarahkan ke model spesifik yang memiliki pengetahuan mendalam tentang topik tersebut (misalnya Bio-Mistral). Namun, jika kueri tersebut lebih umum dan tidak memerlukan pengetahuan domain khusus, sistem akan langsung menggunakan model generatif umum seperti GPT-5, menghindari pemanggilan model yang lebih berat dan tidak diperlukan. Pendekatan ini tidak hanya meningkatkan efisiensi komputasi tetapi juga dapat mengurangi latensi, karena hanya model yang relevan yang dipanggil berdasarkan jenis pertanyaan yang diterima.

III. METODE

A. Perancangan Sistem

Penelitian ini menggunakan dataset obat anti-diare yang diambil melalui scraping dari situs MIMS Indonesia, dengan total populasi sebanyak 447 data. Untuk memastikan representasi yang akurat dari populasi, digunakan rumus Slovin dengan margin kesalahan 10%, yang menghasilkan jumlah sampel uji sebanyak 82 data yang telah divalidasi oleh pakar medis. Dataset ini digunakan untuk melatih dan menguji sistem dalam memberikan informasi medis yang relevan dan akurat.

Arsitektur sistem dibangun dengan Node.js sebagai platform utama untuk pengembangan server-side, memungkinkan aplikasi untuk menangani permintaan secara asinkron dengan efisien. LangChain digunakan sebagai orkestrator untuk mengelola alur pemrosesan data dan koordinasi antar-model. Untuk menyimpan embedding, Milvus dipilih sebagai database vektor, sementara PostgreSQL digunakan untuk menyimpan data terstruktur yang diperlukan untuk pencarian dan analisis tambahan.

Sistem ini dirancang dengan pendekatan berbasis mikroservis, memungkinkan komponen-komponen yang berbeda dapat saling berinteraksi tanpa mengganggu keseluruhan sistem, serta memberikan kemudahan dalam pemeliharaan dan pengembangan lebih lanjut.

B. Konfigurasi Model

Sistem mengintegrasikan dua jenis model model Domain Spesifik: Bio-Mistral 7B yang di-hosting secara lokal menggunakan Flask untuk memproses konteks medis. Model Umum: GPT-5-nano digunakan untuk generasi teks yang koheren.

Sistem ini mengintegrasikan dua jenis model: model Domain Spesifik dan model Umum, yang saling melengkapi dalam proses pembuatan jawaban. Model Bio-Mistral 7B, yang digunakan untuk menangani kueri medis, di-hosting secara lokal menggunakan Flask untuk memastikan pemrosesan yang cepat dan akurat dalam konteks medis. Model ini sangat efektif dalam menangani terminologi medis dan informasi domain khusus.

Di sisi lain, GPT-5-nano digunakan sebagai model umum untuk generasi teks yang koheren. Model ini dapat merespons kueri yang lebih umum dengan cara yang alami dan fleksibel. Kombinasi kedua model ini memungkinkan sistem untuk menangani berbagai jenis kueri, baik yang memerlukan pengetahuan medis mendalam maupun yang bersifat lebih umum, sambil menjaga kualitas dan relevansi jawaban.

C. Skenario Pengujian

Pengujian sistem dilakukan menggunakan metode Prompt Budgeting untuk membatasi penggunaan token maksimal sebanyak 220 token per proses. Hal ini bertujuan untuk memastikan keadilan dalam perbandingan kinerja antara kedua alur pemrosesan yang diuji.

Pada alur Sequential, kueri diproses melalui Model Spesifik terlebih dahulu menggunakan 100 token untuk ekstraksi informasi medis, kemudian diteruskan ke Model Umum yang menggunakan 120 token untuk menghasilkan jawaban akhir.

Pada alur Routing, Routing System mengalokasikan 40 token untuk memilih antara Model Bio-Mistral (menggunakan 180 token) atau Model Umum (juga menggunakan 180 token), tergantung pada jenis kueri yang diajukan, apakah medis atau umum. Dengan metode ini, tujuan utamanya adalah untuk membandingkan kinerja kedua alur dalam memberikan respons yang cepat dan relevan, sambil mengontrol penggunaan token untuk keadilan pengujian.

D. Metrik Evaluasi Kinerja dievaluasi berdasarkan

Evaluasi kinerja sistem dilakukan berdasarkan dua aspek utama: latensi dan kualitas. Latensi mengukur waktu pemrosesan dalam detik, yaitu waktu yang dibutuhkan sistem untuk memproses dan menghasilkan respons dari kueri. Pengujian dilakukan untuk mengevaluasi seberapa cepat sistem dapat merespons pertanyaan dalam berbagai skenario.

Sedangkan kualitas jawaban dievaluasi menggunakan beberapa metrik. F1 Score digunakan untuk mengukur keseimbangan antara presisi dan recall dalam respons yang dihasilkan. ROUGE (1, 2, L) mengukur kesamaan n-gram antara jawaban yang dihasilkan dengan jawaban referensi, memberikan indikasi tentang kedekatan jawaban yang dihasilkan dengan jawaban yang diinginkan. Semantic Similarity mengukur seberapa mirip makna antara jawaban yang dihasilkan dan jawaban yang ideal, menggunakan teknik pengukuran jarak seperti cosine similarity. Terakhir, Groundedness mengukur sejauh mana jawaban yang dihasilkan sesuai dengan fakta yang terkandung dalam sumber referensi yang relevan, memastikan bahwa jawaban tidak hanya koheren, tetapi juga berbasis fakta yang dapat dipercaya.

IV. HASIL DAN PEMBAHASAN

Hasil Pengujian Latensi Hasil eksperimen menunjukkan perbedaan signifikan dalam efisiensi waktu. Metode Routing Questioning mencatat rata-rata latensi sebesar 16,33 detik dengan varians rendah (±4,41 detik). Sebaliknya, Sequential Processing menunjukkan inefisiensi dengan rata-rata latensi 53,69 detik dan varians yang sangat tinggi (±67,19 detik). Tingginya latensi pada metode sekuensial disebabkan oleh

keharusan memanggil model spesifik untuk setiap kueri tanpa memandang relevansinya, yang menambah overhead sistem.

Evaluasi kualitas jawaban menunjukkan bahwa selain unggul dalam kecepatan, Routing Questioning juga mendominasi metrik kualitas jawaban dibandingkan Sequential Processing. Pada metrik ROUGE-1, Routing mencapai nilai 0,1943 sementara Sequential hanya 0,1805. Begitu pula pada Semantic Similarity, Routing mencatatkan nilai 0,8953, sedangkan Sequential 0,8852. Untuk metrik Groundedness, Routing memperoleh nilai 0,1749, lebih tinggi dibandingkan Sequential yang hanya mencatatkan nilai 0,1042. Nilai groundedness yang lebih tinggi pada metode Routing menunjukkan bahwa jawaban yang dihasilkan lebih akurat dan berlandaskan pada data referensi. Sebaliknya, pada metode Sequential, proses bertingkat justru berpotensi mendilusi fakta atau menambah noise dari model perantara, sehingga menghasilkan skor yang lebih rendah.

Pembahasan Meskipun secara teori Sequential Processing dirancang untuk menghasilkan jawaban yang lebih terkontrol melalui pemrosesan bertahap, implementasinya terbukti kurang optimal untuk aplikasi real-time karena penggunaan sumber daya yang berlebihan,. Routing Questioning terbukti lebih efektif karena sistem hanya mengaktifkan model domain spesifik (Bio-Mistral) ketika benar-benar diperlukan (saat kueri diklasifikasikan sebagai medis), sehingga menghilangkan hambatan pemrosesan untuk pertanyaan umum. Hal ini tidak hanya mempercepat waktu respons tetapi juga menjaga konsistensi dan relevansi jawaban.

V. KESIMPULAN

Penelitian ini menyimpulkan bahwa Routing Questioning lebih unggul dibandingkan Sequential Processing dalam optimasi sistem RAG untuk domain spesifik. Pendekatan routing mampu menurunkan latensi secara drastis (sekitar 3 kali lebih cepat) dan meningkatkan akurasi serta relevansi jawaban berdasarkan metrik F1, ROUGE, dan groundedness.

Bagi pengembangan selanjutnya, disarankan untuk mengeksplorasi penggunaan Small Language Models (SLM) guna efisiensi komputasi lebih lanjut. Metode Sequential Processing sebaiknya hanya dipertimbangkan untuk skenario yang membutuhkan penalaran bertingkat yang sangat kompleks di mana waktu respons bukan prioritas utama.

REFERENSI

[1] S. Y. D. S. D. F. S. D. B. K. Z. M. B. T. &. K. V. Sharma, "Retrieval-Augmented Generation for Domain-Specific Question Answering," *Springer AI Journal.,* 2024.

[2] P. Lewis, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.," *arXiv preprint arXiv:2005.,* 2020.

[3] Z. W. H. &. T. B. Liu, "Retrieval-Augmented Generation and Its Impacts on Knowledge-Intensive Applications," *Journal of Artificial Intelligence Innovation.,* 2024.

[4] J. P. K. &. L. S. Han, "Sequential Processing in Retrieval-Augmented Generation for Financial Applications.," *Journal of Applied AI Research.,* 2024.

[5] S. C. Y. C. C. Y. Y. H. L. &. L. X. Wu, "Development and Testing of Retrieval-Augmented Generation for Healthcare Applications.," *Elsevier Applied Computing.,* 2024.

[6] L. L. Y. O. C. &. Y. Y. Cui, "Bailicai: A Domain-Optimized Retrieval-Augmented Generation Framework for Medical Applications," *IEEE Access, 12(2),* p. 3456–3470., 2024.

[7] J. B. A. &. D. M. Gino, "Integrating Ontologies and Large Language Models to Implement Retrieval Augmented Generation (RAG).," *Journal of Information Systems.,* 2024.

[8] P. Bechard, "Reducing Hallucination in Structured Outputs via Retrieval-Augmented Generation," *ACM Transactions on Intelligent Systems and Technology (TIST).,* 2024.

[9] A. Balakrishnan, "Enhancing Data Engineering Efficiency with AI: Utilizing Retrieval-Augmented Generation, Reinforcement Learning from Human Feedback, and Fine-Tuning Techniques.," *Journal of Data Science,* 2024.

[10] V. &. S. M. Patel, "Optimizing Performance in Healthcare Chatbots using RAG Systems," *IEEE Transactions on AI in Medicine.,* 2024.

[11] L. G. R. &. H. J. Martinez, "Evaluating Retrieval and Generation Models for Contextual AI Systems," *AI Systems Review Journal.,* 2024.

[12] K. T. P. K. K. &. T. A. Mandar, "Reinforcement Learning for Optimizing RAG for Domain Chatbots.," *Elsevier AI Advances.,* 2024.