



Pengembangan Chatbot Customer Service dengan Retrieval-Augmented Generation pada Usaha Mikro Kecil Menengah KampusMadu

Ahmad Zakky Zamani*, Novi Tristanti

Fakultas Sains dan Teknologi, Program Studi Informatika, Universitas Muhammadiyah Karanganyar, Karanganyar, Indonesia

Email: ¹*azacky302@gmail.com, ²novitristanti@gmail.com

Email Penulis Korespondensi: azacky302@gmail.com

Abstrak—Layanan pelanggan pada Usaha Mikro, Kecil, dan Menengah (UMKM) umumnya masih dilakukan secara manual, sehingga menyebabkan keterlambatan respons, inkonsistensi informasi, dan beban kerja operasional yang tinggi. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi chatbot customer service berbasis Large Language Model dengan metode Retrieval-Augmented Generation (RAG) sebagai solusi otomatisasi layanan pelanggan pada UMKM KampusMadu. Sistem dikembangkan menggunakan pendekatan Waterfall adaptif melalui tahap analisis kebutuhan, perancangan, implementasi, dan pengujian. Pengujian dilakukan melalui tiga pendekatan: pengujian fungsional dengan metode Black Box Testing, pengujian akurasi retrieval menggunakan metrik Precision@3, Recall@3, dan Hit Rate@3, serta pengujian kegunaan (usability) menggunakan System Usability Scale (SUS) yang melibatkan 10 responden. Hasil pengujian menunjukkan bahwa seluruh fungsi sistem berjalan sesuai kebutuhan, dengan rata-rata Precision@3 sebesar 63,3%, Recall@3 sebesar 72,5%, dan Hit Rate@3 sebesar 100%. Skor SUS yang diperoleh sebesar 69,75 (kategori "OK"), mengindikasikan bahwa sistem cukup dapat diterima oleh pengguna. Sistem chatbot berhasil diintegrasikan ke dalam website KampusMadu sebagai widget interaktif yang dapat diakses langsung oleh pelanggan. Penelitian ini membuktikan bahwa penerapan RAG pada chatbot UMKM mampu membantu otomatisasi layanan pelanggan secara lebih responsif, informatif, dan kontekstual.

Kata Kunci: Chatbot; Customer Service; Large Language Model; Retrieval-Augmented Generation; UMKM

Abstract—Customer service in Micro, Small, and Medium Enterprises (MSMEs) is generally still conducted manually, resulting in delayed responses, inconsistent information, and high operational workload. This study aims to develop and evaluate a customer service chatbot based on Large Language Model using the Retrieval-Augmented Generation (RAG) method as an automation solution for customer service at KampusMadu MSME. The system was developed using an adaptive Waterfall approach encompassing requirements analysis, design, implementation, and testing stages. Testing was conducted through three approaches: functional testing using Black Box Testing method, retrieval accuracy testing using Precision@3, Recall@3, and Hit Rate@3 metrics, and usability testing using the System Usability Scale (SUS) involving 10 respondents. The test results show that all system functions operate as required, with average Precision@3 of 63.3%, Recall@3 of 72.5%, and Hit Rate@3 of 100%. The SUS score obtained was 69.75 (category "OK"), indicating that the system is reasonably acceptable to users. The chatbot system was successfully integrated into the KampusMadu website as an interactive widget directly accessible to customers. This study demonstrates that the application of RAG in MSME chatbots can effectively support the automation of customer service in a more responsive, informative, and contextual manner.

Keywords: Chatbot; Customer Service; Large Language Model; Retrieval-Augmented Generation; MSME

1. PENDAHULUAN

Perkembangan teknologi kecerdasan buatan (*Artificial Intelligence* atau AI) dalam beberapa tahun terakhir telah mencapai titik yang sangat signifikan, mengubah secara fundamental cara manusia berinteraksi dengan mesin dan bagaimana mesin memproses informasi. Salah satu terobosan paling revolusioner dalam bidang ini adalah hadirnya *Large Language Model* (LLM), yang merupakan model bahasa berbasis arsitektur *transformer* yang dilatih menggunakan data teks berskala masif dari berbagai sumber seperti buku, artikel ilmiah, situs web, dan korpus digital lainnya. Arsitektur *transformer* yang diperkenalkan oleh Vaswani et al. pada tahun 2017 memungkinkan model untuk menangkap hubungan antar kata dalam suatu kalimat secara lebih kontekstual melalui mekanisme *self-attention*, sehingga LLM mampu memahami nuansa bahasa, inferensi semantik, dan menghasilkan teks yang sangat mirip dengan tulisan manusia [1]. Kemampuan ini membawa perubahan besar dalam cara bisnis berinteraksi dengan pelanggan, karena LLM dapat digunakan untuk menjawab pertanyaan secara natural, memberikan rekomendasi produk, hingga melakukan negosiasi sederhana dalam waktu nyata. Dengan demikian, teknologi ini membuka peluang luas bagi pelaku usaha, termasuk Usaha Mikro, Kecil, dan Menengah (UMKM), untuk meningkatkan kualitas layanan pelanggan secara lebih cepat, responsif, dan efisien tanpa harus mengeluarkan biaya besar untuk perekrutan staf tambahan.

Peran UMKM dalam perekonomian Indonesia tidak dapat dipandang sebelah mata. Berdasarkan data dari Kementerian Koperasi dan UKM, UMKM menyumbang lebih dari 60% terhadap Produk Domestik Bruto (PDB) nasional dan menyerap sekitar 97% tenaga kerja di sektor swasta. Namun, meskipun kontribusinya sangat besar, tak sedikit UMKM yang masih menghadapi tantangan klasik dalam memberikan layanan pelanggan secara optimal, terutama di era digital yang menuntut kecepatan dan ketepatan. Konsumen saat ini, yang terbiasa dengan kemudahan akses informasi melalui platform digital, cenderung mengharapkan respons instan terhadap setiap pertanyaan mereka terkait informasi produk, harga, ketersediaan stok, cara penggunaan, hingga kebijakan pengiriman. Ekspektasi ini semakin diperkuat oleh dominasi *e-commerce* dan *marketplace* yang menyediakan fitur *chat instant* sebagai standar pelayanan. Keterlambatan dalam merespons pertanyaan pelanggan, bahkan hanya dalam hitungan menit, dapat menurunkan tingkat kepuasan pelanggan, mengurangi peluang konversi penjualan, dan pada akhirnya berdampak negatif pada kredibilitas dan kepercayaan terhadap usaha tersebut. Permasalahan ini menjadi semakin kompleks ketika volume pesan yang masuk



meningkat tajam pada jam-jam sibuk atau saat ada promosi, sementara sumber daya manusia yang tersedia untuk menangani layanan pelanggan sangat terbatas, seringkali hanya dijalankan oleh pemilik usaha itu sendiri atau satu dua orang staf yang juga memiliki tanggung jawab operasional lainnya.

Salah satu contoh nyata dari kondisi tersebut adalah KampusMadu, sebuah UMKM yang bergerak di bidang penjualan madu murni dan produk herbal organik seperti ketumbar, garam Himalaya, *virgin coconut oil* (VCO), *bee pollen*, dan *royal jelly*. Berdasarkan hasil observasi awal yang dilakukan di lokasi usaha, KampusMadu telah memanfaatkan berbagai kanal digital seperti website, Shopee, dan Instagram untuk memasarkan produknya dan menjangkau konsumen yang lebih luas. Kehadiran di berbagai platform ini merupakan langkah maju dalam strategi pemasaran digital, namun sayangnya belum diimbangi dengan sistem layanan pelanggan yang terotomatisasi. Proses layanan pelanggan masih dilakukan secara manual oleh pemilik usaha, di mana setiap pertanyaan dari pelanggan terkait manfaat produk, aturan konsumsi yang tepat, perbandingan harga antar varian produk, serta informasi ketersediaan stok harus dijawab satu per satu melalui berbagai platform digital yang berbeda. Kondisi ini menyebabkan beberapa masalah kritis. Pertama, terjadi keterlambatan respons karena pemilik usaha tidak dapat hadir secara bersamaan di semua kanal. Kedua, terdapat potensi inkonsistensi informasi karena jawaban yang diberikan mungkin berbeda antara satu platform dengan platform lainnya tergantung pada situasi dan kondisi saat menjawab. Ketiga, beban kerja operasional meningkat secara signifikan, mengurangi waktu yang seharusnya dapat dialokasikan untuk pengembangan bisnis, inovasi produk, dan strategi pemasaran yang lebih produktif.

Salah satu solusi teknologi yang paling menjanjikan untuk mengatasi permasalahan tersebut adalah penggunaan *chatbot* berbasis LLM. Berbeda dengan *chatbot* konvensional tipe *rule-based* yang responsnya terbatas pada pola-pola kalimat tertentu yang telah diprogram sebelumnya, *chatbot* berbasis LLM mampu memahami variasi bahasa alami pengguna yang sangat beragam, termasuk penggunaan slang, singkatan, atau pertanyaan yang tidak terstruktur. Kemampuan ini memungkinkan *chatbot* untuk menghasilkan respons yang lebih fleksibel, kontekstual, dan personal, sehingga pengalaman interaksi pelanggan menjadi lebih manusiawi dan memuaskan. Meskipun demikian, penting untuk disadari bahwa LLM memiliki keterbatasan fundamental, terutama dalam mengakses informasi spesifik di luar data pelatihannya. Model LLM dilatih pada data historis yang tidak selalu mutakhir atau relevan dengan produk dan kebijakan spesifik suatu perusahaan. Selain itu, LLM juga berpotensi menghasilkan informasi yang tidak akurat atau bahkan sepenuhnya salah, sebuah fenomena yang dikenal sebagai *hallucination*, di mana model menghasilkan teks yang terdengar meyakinkan namun tidak berdasar pada fakta yang benar. Dalam konteks layanan pelanggan UMKM, *hallucination* dapat berakibat fatal karena dapat menyesatkan pelanggan tentang manfaat produk, dosis konsumsi, atau kebijakan garansi, yang pada gilirannya dapat merusak kepercayaan dan citra usaha.

Untuk mengatasi keterbatasan tersebut, pendekatan *Retrieval-Augmented Generation* (RAG) muncul sebagai salah satu cara yang paling efektif dan banyak diadopsi. RAG merupakan arsitektur yang menggabungkan kemampuan generatif dari LLM dengan mekanisme pencarian informasi dari basis data eksternal yang terverifikasi dan selalu diperbarui [2]. Dalam pendekatan ini, sistem tidak hanya mengandalkan pengetahuan bawaan dari LLM, tetapi pertama-tama melakukan proses *retrieval* atau pencarian terhadap data produk yang relevan dari *vector database*—sebuah basis data yang menyimpan representasi numerik (*embedding*) dari dokumen-dokumen internal perusahaan. Hasil *retrieval* berupa potongan-potongan informasi yang paling sesuai dengan pertanyaan pengguna kemudian digunakan sebagai konteks tambahan bagi LLM untuk menghasilkan respons yang lebih akurat, spesifik, dan dapat dipertanggungjawabkan. Penerapan RAG pada berbagai domain, termasuk kesehatan dan bisnis ritel, telah terbukti mampu mengurangi bias model dan secara signifikan meningkatkan akurasi respons *chatbot* [3]. Dengan pendekatan ini, perusahaan dapat memastikan bahwa setiap jawaban yang diberikan kepada pelanggan selalu didasarkan pada data internal yang valid dan terkini, seperti katalog produk, daftar harga, panduan penggunaan, dan kebijakan pengiriman.

Beberapa penelitian sebelumnya telah mengkaji penerapan *chatbot* berbasis kecerdasan buatan untuk layanan pelanggan, yang menunjukkan meningkatnya minat dan potensi teknologi ini di berbagai sektor. Penelitian pertama mengembangkan *chatbot* berbasis *Natural Language Processing* (NLP) menggunakan Dialogflow untuk rekomendasi UMKM kuliner dan memperoleh tingkat akurasi pengujian sebesar 100% serta skor *System Usability Scale* (SUS) sebesar 85,75, yang mengindikasikan tingkat penerimaan pengguna yang sangat baik [4]. Penelitian kedua mengimplementasikan *chatbot* berbasis NLP pada aplikasi WhatsApp untuk layanan *customer service* usaha jasa permak pakaian, yang mampu merespons pertanyaan pelanggan secara otomatis dan *real-time*, sehingga mengurangi waktu tunggu dan meningkatkan efisiensi operasional [5]. Penelitian ketiga mengembangkan *chatbot* berbasis *Local Retrieval-Augmented Generation* (RAG) menggunakan Ollama dan ChromaDB untuk menjawab pertanyaan berdasarkan isi dokumen PDF, dengan skor ROUGE-L sebesar 0,85, yang menunjukkan bahwa pendekatan RAG lokal mampu menghasilkan ringkasan yang sangat sesuai dengan dokumen sumber [6]. Penelitian keempat mengembangkan *chatbot* berbasis NLP untuk layanan informasi desa wisata Cimindi, yang berhasil mencapai tingkat akurasi pengujian fungsional sebesar 100% dalam memahami konteks pertanyaan pengguna dan memberikan jawaban yang sesuai [7]. Penelitian kelima mengembangkan *chatbot customer service* berbasis LSTM (*Long Short-Term Memory*) yang diintegrasikan ke dalam website menggunakan *framework* Flask, dengan akurasi pelatihan mencapai 90% dan akurasi validasi sebesar 63%, menunjukkan bahwa pendekatan berbasis *deep learning* juga layak dipertimbangkan meskipun performanya masih perlu ditingkatkan [8].

Meskipun penelitian-penelitian tersebut memberikan kontribusi yang berharga, namun sebagian besar masih memiliki beberapa celah yang belum terjawab. Pertama, penelitian-penelitian tersebut umumnya masih berfokus pada implementasi *chatbot* sebagai aplikasi mandiri yang terpisah atau sebagai sistem yang berdiri sendiri, dan belum secara



spesifik mengkaji integrasi *chatbot* berbasis RAG ke dalam *website* UMKM yang telah berjalan sebagai komponen *widget* interaktif yang dapat dipasang (*plug-in*) dengan mudah. Kedua, masih sedikit penelitian yang menerapkan pendekatan RAG menggunakan infrastruktur ringan berbasis *local embedding* dan *vector database* lokal yang sesuai dengan kapasitas teknis dan finansial UMKM. Mayoritas penelitian masih mengandalkan layanan berbasis cloud yang mahal atau membutuhkan sumber daya komputasi yang besar, yang seringkali tidak terjangkau oleh UMKM skala kecil-menengah. Ketiga, belum banyak penelitian yang secara komprehensif mengevaluasi aspek *user experience* dan kepuasan pelanggan dari penggunaan *chatbot* RAG dalam konteks UMKM produk kesehatan dan herbal, yang memiliki karakteristik pertanyaan yang beragam dan membutuhkan akurasi informasi yang tinggi.

Berdasarkan identifikasi permasalahan dan celah penelitian tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem *chatbot customer service* berbasis LLM dengan metode RAG pada UMKM KampusMadu. Sistem dikembangkan menggunakan model *embedding* lokal yang ringan, *vector database* yang dapat dijalankan di server sederhana, dan LLM berbasis API yang terjangkau, kemudian diintegrasikan langsung ke dalam *website* Laravel yang sudah berjalan sebagai *widget* interaktif yang dapat diakses kapan saja oleh pengunjung situs. Dengan pendekatan ini, diharapkan *chatbot* dapat memberikan jawaban yang cepat, akurat, dan kontekstual terhadap pertanyaan pelanggan seputar produk madu dan herbal, tanpa memerlukan intervensi manual dari pemilik usaha. Penelitian ini juga diharapkan dapat memberikan kontribusi praktis bagi pengembangan sistem layanan pelanggan UMKM di Indonesia secara lebih luas, serta menjadi referensi bagi pengembang dan peneliti lain yang ingin mengimplementasikan solusi serupa dengan sumber daya terbatas. Secara lebih spesifik, penelitian ini diharapkan dapat membantu otomatisasi layanan pelanggan UMKM secara lebih responsif, kontekstual, dan efisien, sehingga pada akhirnya dapat meningkatkan kepuasan pelanggan, memperkuat kredibilitas usaha, dan mendorong pertumbuhan bisnis yang berkelanjutan.

2. METODOLOGI PENELITIAN

2.1 Kerangka Dasar Penelitian

Pelaksanaan penelitian ini mengikuti tahapan yang terencana dan berurutan, yaitu dimulai dengan analisis kebutuhan, kemudian perancangan arsitektur sistem, implementasi *chatbot* berbasis *Retrieval-Augmented Generation* (RAG), serta pengujian performa sistem. Metode pengembangan yang digunakan adalah *Waterfall* adaptif yang meliputi empat tahap utama: analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Model *Waterfall* merupakan salah satu model tertua dalam rekayasa perangkat lunak yang mengadopsi pendekatan sekuensial, di mana setiap fase harus rampung terlebih dahulu sebelum fase berikutnya dapat dimulai. Pendekatan ini dipilih karena dinilai tepat untuk pengembangan sistem *chatbot* yang memiliki alur implementasi terstruktur serta kebutuhan sistem yang sudah ditetapkan secara jelas sejak awal penelitian.

Penelitian ini termasuk ke dalam jenis penelitian terapan (*applied research*) dengan menggunakan pendekatan studi kasus yang bersifat implementatif pada UMKM KampusMadu. Subjek penelitian meliputi pemilik UMKM KampusMadu sebagai sumber kebutuhan sistem dan 10 orang responden (pemilik usaha serta pelanggan) untuk pengujian *usability*. Objek penelitian adalah sistem *chatbot* berbasis LLM dan RAG yang diintegrasikan ke dalam *website* KampusMadu. Variabel utama yang dikaji meliputi akurasi *retrieval* produk yang diukur menggunakan metrik *Precision@K*, *Recall@K*, dan *Hit Rate@K*, serta tingkat kegunaan sistem yang diukur menggunakan *System Usability Scale* (SUS).

2.2 Tahapan Penelitian

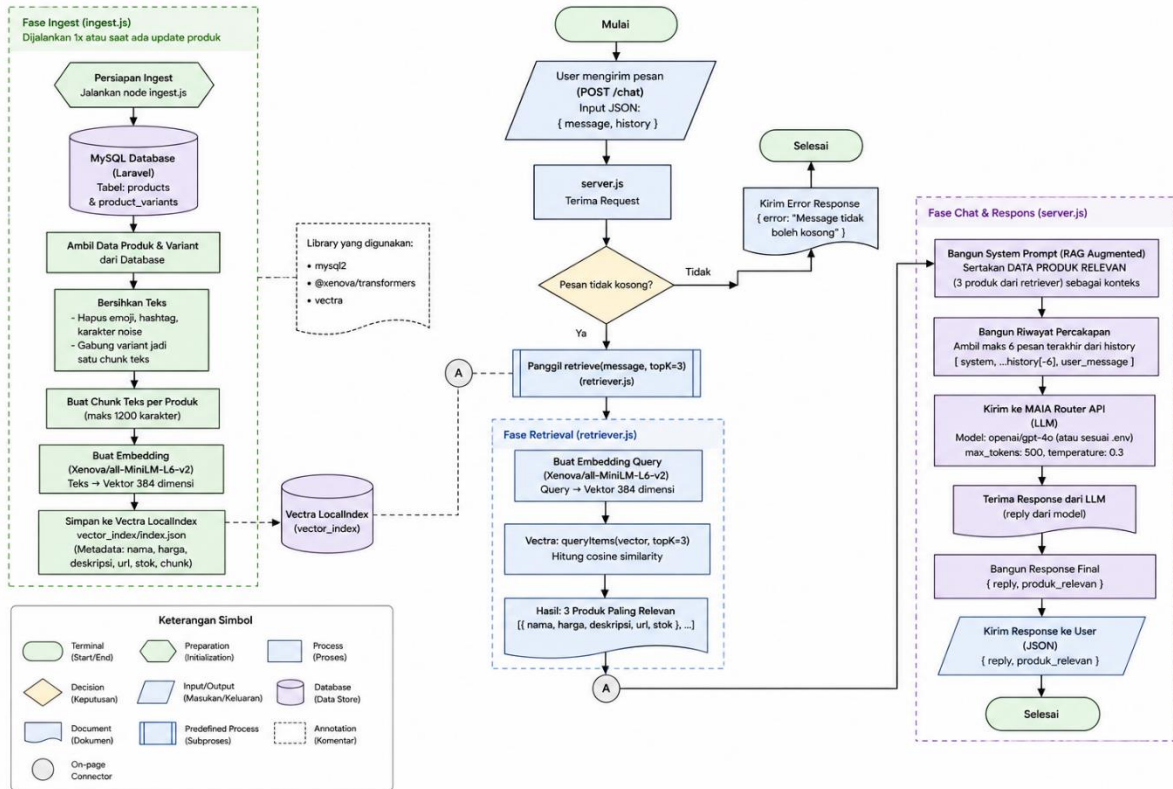
Metode pengembangan yang digunakan adalah *Waterfall* adaptif yang meliputi empat tahap utama: analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Model *Waterfall* termasuk salah satu model tertua dalam rekayasa perangkat lunak yang mengadopsi pendekatan sekuensial, di mana setiap fase harus rampung terlebih dahulu sebelum fase berikutnya dapat dimulai [9]. Adaptasi metode *Waterfall* dipilih sebagai pendekatan pengembangan sistem dalam penelitian ini, dengan menjalankan empat tahap pokok: analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Berikut adalah uraian setiap tahapan:

a. Analisis Kebutuhan

Tahap ini dilaksanakan dengan mengumpulkan dan menganalisis kebutuhan sistem melalui observasi langsung terhadap proses layanan pelanggan di KampusMadu, serta melalui wawancara dengan pemilik usaha. Hasil analisis menunjukkan kebutuhan sistem yang mencakup kemampuan menjawab pertanyaan tentang produk secara akurat berdasarkan katalog, kemampuan memberikan tautan produk yang valid, serta antarmuka yang dapat diakses langsung dari website tanpa aplikasi tambahan.

b. Perancangan Sistem

Arsitektur sistem dirancang dengan tiga komponen utama, yaitu pipeline RAG pada sisi server Node.js, *backend* Laravel sebagai host website, dan komponen *widget* frontend berbasis Alpine.js. Alur kerja sistem diilustrasikan pada Gambar 1



Gambar 1. Arsitektur Sistem Chatbot RAG KampusMadu

Gambar 1 mengilustrasikan arsitektur sistem chatbot RAG KampusMadu yang terdiri dari tiga fase utama. Fase pertama adalah Fase Ingest (ingest.js) yang dijalankan satu kali atau setiap kali terjadi pembaruan data produk. Pada fase ini, sistem mengambil data produk dan varian dari tabel products dan product_variants pada database MySQL Laravel, kemudian membersihkan teks dari karakter noise seperti emoji dan hashtag, menggabungkan informasi varian menjadi satu chunk teks dengan panjang maksimal 1.200 karakter, dan mengubah teks tersebut menjadi vektor berdimensi 384 menggunakan model embedding Xenova/all-MiniLM-L6-v2.

Hasil embedding kemudian disimpan ke dalam Vectra LocalIndex beserta metadata seperti nama produk, harga, deskripsi, URL, stok, dan isi chunk. Fase kedua adalah Fase Retrieval (retriever.js) yang dijalankan setiap kali pengguna mengirimkan pesan. Query pengguna diubah menjadi vektor embedding menggunakan model yang sama, kemudian sistem melakukan pencarian cosine similarity terhadap seluruh vektor yang tersimpan pada Vectra LocalIndex untuk mengambil tiga produk paling relevan (topK=3).

Fase ketiga adalah Fase Chat & Respons (server.js) yang menerima permintaan pengguna melalui endpoint POST /chat. Sistem memvalidasi pesan, membangun system prompt berbasis RAG dengan menyertakan data tiga produk relevan sebagai konteks, menggabungkan riwayat percakapan hingga enam pesan terakhir, lalu mengirimkan seluruh data ke MAIA Router API. Respons dari LLM kemudian dikembalikan kepada pengguna dalam format JSON yang memuat teks jawaban dan daftar produk relevan.

c. Implementasi

Implementasi sistem chatbot dilakukan dengan mengintegrasikan beberapa komponen utama yang mendukung proses Retrieval-Augmented Generation (RAG), mulai dari proses ingestion data produk, mekanisme retrieval berbasis pencarian semantik, integrasi dengan Large Language Model (LLM), hingga pengembangan antarmuka chatbot pada website Laravel. Sistem terdiri dari Node.js sebagai server utama untuk proses RAG dan Laravel sebagai platform website UMKM KampusMadu.

Pada tahap ingestion, data produk diambil secara langsung dari database MySQL Laravel melalui tabel products dan product_variants. Data yang diproses meliputi nama produk, deskripsi, manfaat, cara penggunaan, harga, stok, dan variasi produk. Sebelum dilakukan proses embedding, teks produk terlebih dahulu dibersihkan dari karakter yang tidak diperlukan seperti emoji, hashtag, dan karakter noise lainnya. Selanjutnya, sistem menggabungkan informasi produk dan variannya ke dalam satu teks chunk dengan panjang maksimal 1200 karakter untuk menjaga efisiensi pemrosesan. Teks tersebut kemudian diubah menjadi representasi vektor dengan model embedding Xenova/all-MiniLM-L6-v2 yang berjalan secara lokal melalui pustaka @xenova/transformers. Hasil embedding disimpan ke dalam vector database Vectra beserta metadata produk seperti nama, harga, URL, stok, dan isi chunk.

Pada proses retrieval, sistem menerima query dari pengguna dan mengubahnya menjadi vektor menggunakan model embedding yang sama. Selanjutnya, sistem melakukan pencarian semantik menggunakan metode cosine similarity



terhadap seluruh vektor produk yang tersimpan pada vector database. Sistem mengambil tiga produk dengan nilai kemiripan tertinggi (topK = 3) sebagai konteks utama untuk proses pembangkitan respons. Untuk meningkatkan efisiensi performa, model embedding dan vector index dimuat satu kali saat server dijalankan menggunakan pendekatan singleton pattern, sehingga proses retrieval dapat dilakukan lebih cepat tanpa memuat ulang model pada setiap permintaan pengguna.

Integrasi dengan LLM dilakukan melalui server Node.js yang menyediakan endpoint POST /chat. Setiap permintaan yang masuk diproses melalui beberapa tahapan, yaitu validasi pesan pengguna, retrieval produk relevan, pembangunan system prompt berbasis RAG, penggabungan riwayat percakapan (multi-turn conversation), dan pengiriman data ke layanan MAIA Router API. Sistem menggunakan riwayat maksimal enam pesan terakhir untuk menjaga efisiensi penggunaan token pada LLM. Respons yang dihasilkan dikembalikan dalam format JSON yang memuat teks jawaban chatbot serta daftar produk relevan yang diperoleh dari proses retrieval.

Pada sisi antarmuka, chatbot diintegrasikan ke dalam website Laravel menggunakan Laravel Blade dan Alpine.js. Komponen widget chatbot berkomunikasi dengan server Node.js menggunakan fetch API berbasis HTTP. Respons chatbot ditampilkan dalam bentuk gelembung percakapan yang dilengkapi dengan chip tautan produk relevan yang dapat diklik langsung oleh pengguna untuk menuju halaman produk KampusMadu. Pendekatan ini memungkinkan chatbot diakses secara langsung dari website tanpa memerlukan aplikasi tambahan, sehingga pengalaman pengguna menjadi lebih praktis dan responsif.

d. Pengujian

Pengujian sistem dilakukan untuk mengevaluasi fungsionalitas, performa retrieval, dan tingkat kegunaan (usability) chatbot berbasis Retrieval-Augmented Generation (RAG) yang dikembangkan. Pengujian dibagi menjadi tiga tahap utama, yaitu pengujian fungsional, pengujian akurasi retrieval, dan pengujian subjektif menggunakan System Usability Scale (SUS).

2.3 Teknologi yang Digunakan

Untuk mewujudkan sistem chatbot customer service yang diusulkan, diperlukan pemilihan teknologi dan tools yang tepat guna, dengan mempertimbangkan aspek performa, biaya, kemudahan implementasi, serta kesesuaian dengan infrastruktur yang telah dimiliki oleh UMKM KampusMadu. Secara keseluruhan, arsitektur sistem dibagi menjadi beberapa lapisan utama, yaitu lapisan backend website, lapisan runtime RAG, lapisan penyimpanan dan pencarian vektor, lapisan kecerdasan buatan, serta lapisan antarmuka pengguna. Pada lapisan backend website, Laravel 10 dipilih karena merupakan framework yang sudah digunakan pada sistem KampusMadu yang telah berjalan, sehingga memudahkan proses integrasi dan mengurangi kompleksitas pengembangan ulang. Sementara itu, untuk lapisan runtime RAG, Node.js v18 dan Express.js digunakan sebagai lingkungan eksekusi dan web framework untuk menangani endpoint REST API yang melayani proses retrieval dan generasi jawaban. Pemilihan Node.js didasarkan pada kemampuannya dalam menangani operasi I/O asinkron secara efisien, yang sangat sesuai untuk proses embedding dan query ke vector database. Pada lapisan penyimpanan, MySQL tetap dipertahankan sebagai database sumber data produk, sementara Vectra (LocalIndex) dipilih sebagai vector database lokal yang ringan dan tidak memerlukan infrastruktur server terpisah, sehingga sangat ramah bagi keterbatasan teknis dan finansial UMKM. Untuk model embedding, digunakan Xenova/all-MiniLM-L6-v2 yang dapat dijalankan secara lokal tanpa biaya tambahan, sedangkan untuk LLM digunakan model openai/gpt-4o melalui MAIA Router API yang menyediakan akses berkualitas tinggi dengan skema pembayaran yang fleksibel. Terakhir, pada sisi frontend, Alpine.js dan Laravel Blade dipilih untuk membangun widget chatbot interaktif yang ringan dan terintegrasi langsung ke dalam halaman website tanpa memerlukan framework JavaScript yang berat. Keseluruhan komponen teknologi tersebut dirangkum secara sistematis pada Tabel 1.

Tabel 1. Teknologi dan Tools yang Digunakan

Komponen	Teknologi/Tools
Runtime server RAG	Node.js v18
Web framework RAG	Express.js
Framework web backend	Laravel 10
Database sumber data	MySQL (mysql2)
Model embedding	Xenova/all-MiniLM-L6-v2
Vector database	Vectra (LocalIndex)
LLM provider	MAIA Router API
Model LLM	openai/gpt-4o
Frontend widget	Alpine.js, Laravel Blade

Node.js v18 dipilih sebagai runtime server RAG karena mendukung pemrosesan asinkron yang efisien, sementara Express.js digunakan sebagai web framework ringan untuk menangani endpoint REST API. Laravel 10 berfungsi sebagai platform website utama sekaligus pengelola data produk pada database MySQL yang diakses melalui library mysql2.

Untuk komponen inti RAG, model embedding Xenova/all-MiniLM-L6-v2 dipilih karena mampu berjalan secara lokal tanpa memerlukan akses API eksternal, sehingga lebih efisien dari segi biaya dan latensi. Hasil embedding disimpan pada Vectra (LocalIndex) sebagai vector database lokal yang ringan dan tidak memerlukan infrastruktur server terpisah.



Integrasi dengan Large Language Model dilakukan melalui MAIA Router API menggunakan model openai/gpt-4o yang mampu menghasilkan respons kontekstual berkualitas tinggi.

Pada sisi frontend, Alpine.js dan Laravel Blade digunakan untuk membangun widget chatbot interaktif yang terintegrasi langsung ke dalam halaman website KampusMadu tanpa memerlukan framework JavaScript berat. Sistem chatbot dibangun menggunakan arsitektur Retrieval-Augmented Generation (RAG) yang mengintegrasikan Laravel, Node.js, vector database, dan Large Language Model (LLM). Laravel 10 digunakan sebagai platform website utama sekaligus pengelola data produk yang tersimpan pada database MySQL.

Sementara itu, Node.js dan Express.js digunakan sebagai server RAG yang menangani proses retrieval dan komunikasi dengan LLM. Proses ingestion dilakukan menggunakan skrip ingest.js yang mengambil data produk dari tabel products dan product_variants, kemudian membersihkan teks dan mengubahnya menjadi embedding vector menggunakan model Xenova/all-MiniLM-L6-v2. Hasil embedding disimpan pada vector database Vectra berbasis lokal untuk mendukung pencarian semantik menggunakan cosine similarity. Pada saat pengguna mengirimkan pertanyaan, sistem melakukan retrieval tiga produk paling relevan (topK = 3) melalui retriever.js, kemudian menyisipkan hasil retrieval tersebut ke dalam system prompt sebelum dikirim ke MAIA Router API menggunakan model openai/gpt-4o.

Respons yang dihasilkan dikembalikan dalam format JSON berisi jawaban chatbot dan daftar produk relevan. Pada sisi frontend, widget chatbot diintegrasikan ke dalam website Laravel menggunakan Laravel Blade dan Alpine.js. Komunikasi dengan server dilakukan melalui fetch API, sedangkan hasil respons chatbot ditampilkan dalam bentuk percakapan beserta chip tautan produk yang dapat diakses langsung oleh pengguna.

2.4 Metode Pengujian Sistem

a. Black Box Testing

Pengujian fungsional dilakukan menggunakan metode Black Box Testing dengan teknik Equivalence Partitioning. Metode Black Box Testing berfokus pada evaluasi perilaku eksternal sistem tanpa perlu mengetahui struktur kode internalnya [10]. Teknik Equivalence Partitioning membagi data masukan ke dalam kelas-kelas yang ekuivalen (valid dan tidak valid), sehingga memungkinkan efisiensi dalam jumlah kasus uji tanpa mengurangi cakupan pengujian [10]. Tujuan dari pengujian ini adalah untuk menjamin bahwa semua fungsi sistem beroperasi sebagaimana yang dibutuhkan, tanpa perlu meninjau struktur kode program. Skenario pengujian meliputi validasi input pengguna, proses pengiriman pesan, retrieval produk, hingga penanganan kondisi error pada sistem.

b. Pengujian Akurasi Retrieval

Pengujian akurasi retrieval dilakukan untuk mengukur kemampuan retriever dalam mengembalikan produk yang relevan berdasarkan query pengguna. Pada penelitian ini digunakan tiga metrik evaluasi information retrieval, yaitu Precision@K, Recall@K, dan Hit Rate@K dengan nilai K = 3 sesuai jumlah produk teratas yang ditampilkan sistem. Metrik precision dan recall merupakan standar evaluasi yang umum digunakan untuk mengukur kinerja retrieval pada sistem RAG [11]. Dalam systematic literature review tentang RAG, metrik Precision@K, Recall@K, dan Hit Rate@K (juga dikenal sebagai Hit Success Ratio) diidentifikasi sebagai metrik yang paling sering digunakan untuk mengevaluasi efektivitas retrieval pada sistem RAG [12].

Precision@K digunakan untuk mengukur proporsi produk relevan dari total hasil retrieval yang dikembalikan sistem. Recall@K digunakan untuk mengukur kemampuan sistem dalam menemukan seluruh produk relevan yang tersedia di dalam database. Sementara itu, Hit Rate@K digunakan untuk mengukur apakah minimal terdapat satu produk relevan pada hasil retrieval yang diberikan sistem.

Persamaan Precision@K ditunjukkan pada persamaan (1).

$$Precision@K = \frac{\text{Jumlah produk relevan pada top-K}}{K} \quad (1)$$

Persamaan Recall@K ditunjukkan pada persamaan (2).

$$Recall@K = \frac{\text{Jumlah produk relevan pada top-K}}{\text{Total seluruh produk relevan}} \quad (2)$$

Persamaan Hit Rate@K ditunjukkan pada persamaan (3).

$$HitRate@K = \begin{cases} 1, & \text{jika terdapat minimal satu hasil relevan} \\ 0, & \text{jika tidak ada hasil relevan} \end{cases} \quad (3)$$

Relevansi produk ditentukan secara manual berdasarkan kesesuaian semantik antara query pengguna dengan produk yang dikembalikan sistem. Pengujian dilakukan menggunakan 10 query representatif yang mewakili berbagai kebutuhan pelanggan seperti pencarian produk kesehatan, manfaat madu, stamina, imun tubuh, hingga kebutuhan program hamil.

c. Pengujian Subjektif (System Usability Scale)

Pengujian subjektif dilakukan untuk mengukur tingkat kegunaan (usability) sistem chatbot dari perspektif pengguna secara langsung. Metode yang digunakan adalah System Usability Scale (SUS), yaitu kuesioner standar terdiri dari 10 butir pernyataan dengan skala Likert 1 hingga 5. SUS banyak digunakan dalam evaluasi sistem berbasis antarmuka pengguna karena sifatnya yang sederhana, cepat, dan telah tervalidasi secara luas.



Penerapan SUS pada penelitian chatbot berbasis NLP juga telah dilakukan pada Penelitian terdahulu yang memperoleh skor SUS sebesar 85,75 pada chatbot rekomendasi UMKM kuliner[4]. Responden pengujian berjumlah 10 orang yang terdiri dari pemilik UMKM KampusMadu dan pelanggan atau calon pengguna. Setiap responden diminta untuk mencoba berinteraksi langsung dengan chatbot melalui widget pada website KampusMadu, kemudian mengisi kuesioner SUS yang disebarakan melalui Google Form.

Tabel 2. Butir Pernyataan System Usability Scale (SUS)

No	Pernyataan SUS	Jenis
1	“Saya rasa saya akan sering menggunakan sistem chatbot ini”	Positif
2	“Saya merasa sistem chatbot ini terlalu rumit untuk digunakan”	Negatif
3	“Saya merasa sistem chatbot ini mudah digunakan”	Positif
4	“Saya membutuhkan bantuan teknis untuk dapat menggunakan sistem ini”	Negatif
5	“Saya merasa berbagai fitur chatbot ini terintegrasi dengan baik”	Positif
6	“Saya merasa terlalu banyak ketidakkonsistenan pada sistem chatbot ini”	Negatif
7	“Saya rasa kebanyakan orang akan mudah mempelajari cara menggunakan chatbot ini”	Positif
8	“Saya merasa sistem chatbot ini sangat tidak praktis digunakan”	Negatif
9	“Saya merasa percaya diri menggunakan sistem chatbot ini”	Positif
10	“Saya perlu belajar banyak hal sebelum bisa menggunakan sistem chatbot ini”	Negatif

Skor SUS dihitung dengan menggunakan rumus standar. Pada butir-butir bernomor ganjil (pernyataan positif: butir 1, 3, 5, 7, 9), skor kontribusi dihitung dengan mengurangi nilai jawaban responden dengan 1 (nilai - 1). Sedangkan untuk butir bernomor genap (pernyataan negatif: butir 2, 4, 6, 8, 10), skor kontribusi dihitung dengan mengurangi 5 dengan nilai jawaban responden (5 - nilai). Setelah itu, total dari semua skor kontribusi dikalikan dengan 2,5 untuk menghasilkan skor SUS akhir yang berkisar antara 0 hingga 100.

3. HASIL DAN PEMBAHASAN

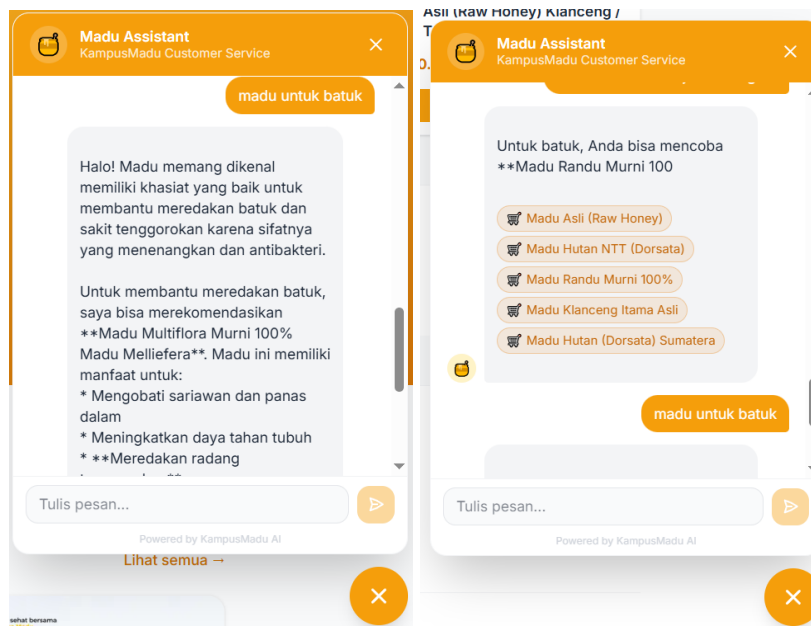
3.1 Hasil Pengujian Fungsional

Pengujian fungsional pada penelitian ini menerapkan pendekatan Black Box Testing beserta teknik Equivalence Partitioning. Teknik tersebut mengelompokkan domain masukan ke dalam berbagai kelas data, sehingga kasus uji dapat dirancang berdasarkan kondisi input yang valid maupun tidak valid [13]. Pengujian ini bertujuan untuk memverifikasi bahwa semua fungsi sistem chatbot berfungsi dengan baik sesuai kebutuhan, tanpa melakukan pemeriksaan terhadap struktur kode program.

Tabel 3. Tabel Pengujian Fungsional

No	Skenario Pengujian	Input	Output yang Diharapkan	Hasil
1	Pengiriman produk valid	query “madu untuk batuk”	Sistem menampilkan respons chatbot dan rekomendasi produk relevan	Sesuai Harapan
2	Pengiriman manfaat kesehatan	query “produk untuk imun tubuh”	Sistem menampilkan produk yang berkaitan dengan imun dan daya tahan tubuh	Sesuai Harapan
3	Pengiriman query harga produk	query “harga madu melliefera”	Sistem menampilkan nama dan harga produk yang relevan	Sesuai Harapan
4	Pengiriman query di luar domain	query “siapa presiden indonesia?”	Sistem menolak pertanyaan secara sopan dan mengarahkan ke topik KampusMadu	Sesuai Harapan
5	Input kosong	“” (string kosong)	Sistem menampilkan error “Pesan tidak boleh kosong” dengan HTTP 400	Sesuai Harapan
6	Multi-turn conversation	“bisa kasih linknya?” setelah query produk	Sistem tetap memberikan link produk sesuai konteks percakapan	Sesuai Harapan
7	Produk tidak tersedia	“apakah ada madu untuk diabetes tipe 1?”	Sistem menyampaikan bahwa produk yang dimaksud belum tersedia	Sesuai Harapan
8	Koneksi server gagal	Server chatbot dimatikan	Widget menampilkan pesan gangguan teknis	Sesuai Harapan
9	Pengiriman pesan berulang cepat	User mengirim beberapa pesan berturut-turut	Sistem tetap memproses pesan tanpa crash	Sesuai Harapan
10	Tampilan chip produk	Query produk valid	Sistem menampilkan chip/link produk yang dapat diklik	Sesuai Harapan

Tabel 3 menunjukkan bahwa seluruh skenario pengujian fungsional berhasil dijalankan dengan hasil "Sesuai Harapan". Untuk memberikan gambaran visual tentang implementasi antarmuka chatbot, Gambar 2 menampilkan tampilan widget chatbot yang telah diintegrasikan ke dalam website KampusMadu.



Gambar 2. Tampilan Widget Chatbot pada Website KampusMadu

Gambar 2 menunjukkan antarmuka chatbot yang muncul sebagai widget pada pojok kanan bawah website KampusMadu, dengan contoh percakapan pengguna menanyakan produk untuk meredakan batuk.

3.2 Hasil Pengujian Akurasi Retrieval

Pengujian akurasi retrieval dilakukan untuk mengukur kemampuan retriever dalam mengembalikan produk yang relevan dari vector database berdasarkan query pengguna. Evaluasi dilakukan menggunakan tiga metrik information retrieval, yaitu Precision@3, Recall@3, dan Hit Rate@3. Metrik Recall, Precision, dan Hit Rate dengan pendekatan Top-N merupakan standar evaluasi yang umum digunakan pada sistem rekomendasi di platform e-commerce [14]

Precision@3 digunakan untuk mengukur proporsi produk relevan dari tiga hasil teratas yang dikembalikan sistem. Recall@3 digunakan untuk mengukur kemampuan sistem menemukan seluruh produk relevan yang tersedia pada database. Dalam penelitian tentang embedding-based retrieval untuk product search di e-commerce, metrik Precision@k dan Recall@k juga digunakan sebagai indikator utama untuk mengevaluasi efektivitas sistem retrieval pada tahap awal [15]. Sedangkan Hit Rate@3 digunakan untuk mengevaluasi apakah sistem berhasil menampilkan minimal satu produk relevan pada hasil retrieval.

Pengujian dilakukan terhadap 10 query representatif yang mencerminkan kebutuhan umum pelanggan UMKM KampusMadu, seperti pencarian produk berdasarkan manfaat kesehatan, jenis madu, stamina tubuh, hingga program hamil. Relevansi produk ditentukan secara manual berdasarkan kesesuaian semantik antara query dengan deskripsi produk yang tersimpan pada vector database.

Tabel 4. Hasil Pengujian Retrieval pada Top-3

No	Query Uji	Precision@3	Recall@3	Hit Rate@3
1	Madu Klanceng Trigona untuk pengobatan kanker	0,67	1,00	1
2	Madu hutan liar murni dari Sumatera atau NTT	0,67	1,00	1
3	Produk untuk meningkatkan stamina dan vitalitas pria	0,33	0,50	1
4	Madu untuk program hamil pasutri	0,67	1,00	1
5	Madu untuk mengobati batuk dan radang tenggorokan	0,67	0,67	1
6	Harga madu melliefera murni	1,00	0,75	1
7	Madu untuk menurunkan kolesterol dan tekanan darah	0,67	1,00	1
8	Madu raw honey tanpa campuran untuk kesehatan umum	0,67	0,50	1
9	Madu untuk menyembuhkan luka dan sariawan	0,33	0,33	1
10	Suplemen herbal meningkatkan imun dan daya tahan tubuh	0,67	0,50	1

Berdasarkan hasil pengujian, sistem memperoleh rata-rata Precision@3 sebesar 63,3%, Recall@3 sebesar 72,5%, dan Hit Rate@3 sebesar 100%. Nilai Hit Rate@3 sebesar 100% menunjukkan bahwa seluruh query pengujian berhasil memperoleh minimal satu produk yang relevan pada hasil retrieval.

Nilai recall yang lebih tinggi dibanding precision menunjukkan bahwa retriever cukup baik dalam menemukan produk relevan, namun masih terdapat beberapa produk tidak relevan yang ikut muncul pada top-3 hasil retrieval. Hal ini terlihat pada query seperti “produk untuk meningkatkan stamina dan vitalitas pria” dan “madu untuk menyembuhkan luka dan sariawan” yang menghasilkan Precision@3 sebesar 0,33.



Sementara itu, query yang bersifat spesifik seperti “harga madu melliefera murni” memperoleh Precision@3 sebesar 1,00 karena seluruh hasil retrieval sesuai dengan kategori produk yang dicari pengguna. Hasil ini menunjukkan bahwa sistem retrieval berbasis embedding semantic mampu bekerja dengan baik pada query yang memiliki keterkaitan istilah yang jelas dengan deskripsi produk.

3.3 Hasil Pengujian System Usability Scale (SUS)

Pengujian SUS dilakukan untuk mengukur tingkat kegunaan (usability) sistem chatbot KampusMadu dari perspektif pengguna. Pengujian melibatkan 10 responden yang mengisi kuesioner secara mandiri melalui Google Form setelah mencoba sistem chatbot secara langsung melalui website KampusMadu. Hasil perhitungan skor SUS per responden disajikan pada Tabel berikut.

Tabel 5. Hasil Pengujian System Usability Scale (SUS)

Resp.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Skor SUS
R1	4	2	4	2	4	2	4	2	4	2	75.0
R2*	5	5	5	4	5	5	5	5	5	5	52.5
R3*	5	5	5	5	5	5	5	5	5	5	50.0
R4	4	2	4	2	4	2	4	2	4	2	75.0
R5	4	2	4	2	4	2	4	2	4	2	75.0
R6	4	3	4	2	4	2	4	2	4	2	72.5
R7	3	2	4	2	4	2	4	2	4	2	72.5
R8	3	2	5	2	5	2	4	2	4	2	77.5
R9	3	2	4	2	4	2	4	2	4	2	72.5
R10	4	2	4	2	4	2	4	2	4	2	75.0
Rata-rata											69.75

Berdasarkan hasil pengujian, terdapat dua responden (R2 dan R3) yang memberikan jawaban "Sangat Setuju" pada seluruh butir pernyataan termasuk butir-butir bernada negatif, seperti "Saya merasa sistem chatbot ini terlalu rumit" dan "Saya perlu belajar banyak hal sebelum bisa menggunakan sistem ini". Pola jawaban seragam pada seluruh butir ini merupakan indikasi response bias, yang terjadi ketika responden tidak membaca pernyataan secara cermat dan memberikan jawaban yang tidak konsisten secara semantik. Kondisi ini menyebabkan skor SUS R2 (52,5) dan R3 (50,0) berada jauh di bawah skor responden lainnya, sehingga memengaruhi nilai rata-rata keseluruhan. Dengan mempertimbangkan anomali tersebut, rata-rata skor SUS seluruh responden adalah 69.75 (kategori "OK"), sedangkan rata-rata skor SUS dari 8 responden valid (R1, R4–R10) adalah 74.38 (kategori "Good"). Kedua nilai tersebut disajikan sebagai bahan pertimbangan dalam analisis. Perbandingan skor dengan tabel interpretasi SUS ditunjukkan pada Tabel Berikut

Tabel 6. Interpretasi System Usability Scale

Rentang Skor	Kategori	Grade	Keterangan
85,0 – 100,0	Excellent	A	Sistem sangat mudah digunakan
71,0 – 84,9	Good	B	Sistem mudah digunakan
51,0 – 70,9	OK	C	Sistem cukup dapat digunakan
25,0 – 50,9	Poor	D	Sistem sulit digunakan
< 25,0	Awful	F	Sistem sangat sulit digunakan

Apabila ditinjau dari skor responden valid (R1, R4–R10), sebagian besar responden memberikan nilai positif pada butir yang berkaitan dengan kemudahan penggunaan (butir 3), integrasi fitur (butir 5), dan kepercayaan diri dalam menggunakan sistem (butir 9). Hal ini menunjukkan bahwa widget chatbot berhasil diintegrasikan ke dalam website dengan antarmuka yang intuitif dan mudah dipahami oleh pengguna dari berbagai latar belakang. Beberapa responden memberikan nilai netral pada butir 1 (frekuensi penggunaan) yang mengindikasikan bahwa tingkat adopsi jangka panjang masih perlu didorong, misalnya melalui peningkatan fitur proaktif atau notifikasi produk. Selain itu, satu responden (R6) memberikan nilai netral pada butir 2 (kerumitan sistem), yang menunjukkan bahwa terdapat ruang perbaikan pada aspek kesederhanaan alur percakapan chatbot.

3.4 Pembahasan

Hasil implementasi menunjukkan bahwa pendekatan Retrieval-Augmented Generation (RAG) berhasil diterapkan pada sistem chatbot UMKM KampusMadu untuk memberikan rekomendasi produk berbasis konteks query pengguna. Efektivitas pendekatan RAG dalam meningkatkan kualitas respons chatbot juga dibuktikan dalam penelitian [16], yang mengembangkan chatbot berbasis RAG untuk layanan informasi kampus dan berhasil mencapai skor correctness sebesar 0,89 serta relevance sebesar 0,90 menggunakan model Sahabat-AI. Keberhasilan ini tidak terlepas dari pemanfaatan vector database dan semantic retrieval, yang memungkinkan sistem mengambil informasi produk yang relevan sebelum menghasilkan respons menggunakan Large Language Model (LLM). Hal ini sejalan dengan temuan yang menyatakan



bahwa LLM menunjukkan performa yang mengesankan dalam tugas-tugas yang melibatkan pemahaman bahasa, termasuk pemrosesan masukan faktual dan menghasilkan respons yang koheren sesuai dengan konteks masukan yang diberikan[17].

Untuk mengukur sejauh mana kinerja sistem retrieval pada penelitian ini, dilakukan evaluasi menggunakan tiga metrik information retrieval. Berdasarkan hasil pengujian retrieval, sistem memperoleh rata-rata Precision@3 sebesar 63,3%, Recall@3 sebesar 72,5%, dan Hit Rate@3 sebesar 100%. Nilai Hit Rate@3 yang mencapai 100% menunjukkan bahwa sistem selalu mampu menemukan minimal satu produk relevan untuk seluruh query pengujian. Hal ini mengindikasikan bahwa proses embedding dan pencarian berbasis cosine similarity telah berjalan dengan baik.

Meskipun demikian, nilai Precision@3 yang masih berada pada kategori cukup menunjukkan bahwa retriever masih mengembalikan beberapa produk yang kurang relevan pada posisi top-3. Kondisi ini terutama terjadi pada query yang bersifat umum atau memiliki banyak keterkaitan semantik antar produk, seperti query terkait stamina, imun tubuh, dan kesehatan umum. Hal ini mencerminkan tantangan yang diidentifikasi [18], bahwa kualitas retrieval pada sistem RAG sangat dipengaruhi oleh strategi chunking dan pemilihan granularitas retrieval, di mana granularitas yang terlalu kasar dapat menyebabkan noise sedangkan yang terlalu halus berisiko kehilangan integritas semantik.

Nilai Recall@3 yang lebih tinggi dibanding Precision@3 menunjukkan bahwa sistem relatif lebih baik dalam menemukan produk relevan dibanding menyaring hasil agar seluruhnya benar-benar relevan. Dengan kata lain, retriever cenderung menghasilkan retrieval yang cukup luas sehingga meningkatkan peluang produk relevan ditemukan, tetapi masih menyertakan noise berupa produk yang kurang sesuai.

Hasil terbaik diperoleh pada query “harga madu melliefera murni” dengan Precision@3 sebesar 1,00 karena seluruh hasil retrieval berasal dari kategori produk yang sama. Sebaliknya, query “madu untuk menyembuhkan luka dan sariawan” memperoleh Precision@3 sebesar 0,33 karena beberapa produk yang dikembalikan memiliki kemiripan manfaat umum tetapi tidak secara eksplisit menyebut luka atau sariawan pada deskripsi produk.

Temuan ini menunjukkan bahwa kualitas retrieval sangat dipengaruhi oleh kelengkapan deskripsi manfaat produk pada database. Semakin kaya informasi manfaat dan kata kunci yang terdapat pada chunk produk, maka semakin baik kemampuan semantic retrieval dalam menemukan produk yang sesuai dengan kebutuhan pengguna.

Dari aspek kegunaan (usability), hasil pengujian System Usability Scale (SUS) menunjukkan bahwa sistem chatbot memperoleh skor rata-rata sebesar 69,75 yang termasuk dalam kategori "OK" (C) berdasarkan skala interpretasi SUS. Skor ini mengindikasikan bahwa sistem chatbot secara fungsional sudah berjalan dengan baik dan dapat diterima oleh pengguna, meskipun masih terdapat ruang untuk peningkatan. Apabila dua responden yang mengalami response bias (R2 dan R3) dikeluarkan dari perhitungan, rata-rata skor SUS meningkat menjadi 74,38 yang termasuk dalam kategori "Good" (B). Skor SUS yang diperoleh dalam penelitian ini lebih rendah dibandingkan penelitian Dynia & Wibowo (2025) yang mencapai skor 85,75 pada chatbot rekomendasi UMKM kuliner, namun masih berada dalam rentang yang dapat diterima untuk sistem prototipe.

Berdasarkan analisis per butir pernyataan, sebagian besar responden memberikan nilai positif pada aspek kemudahan penggunaan (butir 3), integrasi fitur (butir 5), dan kepercayaan diri dalam menggunakan sistem (butir 9). Hal ini menunjukkan bahwa widget chatbot berhasil diintegrasikan ke dalam website dengan antarmuka yang intuitif dan mudah dipahami. Beberapa responden memberikan nilai netral pada butir frekuensi penggunaan (butir 1), yang mengindikasikan bahwa tingkat adopsi jangka panjang masih perlu didorong, misalnya melalui peningkatan fitur proaktif atau notifikasi produk. Selain itu, terdapat responden yang memberikan nilai netral pada aspek kerumitan sistem (butir 2), yang menunjukkan ruang perbaikan pada kesederhanaan alur percakapan chatbot.

Secara keseluruhan, hasil pengujian menunjukkan bahwa pendekatan RAG yang diimplementasikan sudah cukup efektif untuk membantu pencarian produk UMKM berbasis bahasa alami. Namun, masih diperlukan optimalisasi pada proses chunking, enrichment metadata manfaat produk, dan strategi retrieval agar precision sistem dapat meningkat pada penelitian selanjutnya. Berisi hasil pembahasan dan bisa perbandingan dari hasil penelitian sebelumnya.

4. KESIMPULAN

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem chatbot customer service berbasis Large Language Model (LLM) dengan metode Retrieval-Augmented Generation (RAG) pada UMKM KampusMadu guna mengatasi permasalahan keterlambatan respons, inkonsistensi informasi, dan beban kerja operasional yang tinggi akibat proses layanan pelanggan yang masih dilakukan secara manual di berbagai platform digital, dan berdasarkan hasil implementasi serta pengujian yang telah dilakukan, sistem chatbot berhasil dibangun dan diintegrasikan ke dalam website Laravel sebagai widget interaktif yang mampu menerima pertanyaan pengguna, melakukan pencarian produk secara semantik dari vector database menggunakan model embedding Xenova/all-MiniLM-L6-v2, serta menghasilkan respons kontekstual melalui integrasi dengan LLM melalui MAIA Router API, dengan pengujian fungsional Black Box Testing yang menunjukkan seluruh fitur berfungsi sesuai kebutuhan, sementara pengujian akurasi retrieval memperoleh rata-rata Precision@3 sebesar 63,3%, Recall@3 sebesar 72,5%, dan Hit Rate@3 sebesar 100%, yang mengindikasikan sistem efektif dalam menemukan produk relevan meskipun masih terdapat ruang peningkatan pada query bersifat umum, dan pengujian System Usability Scale (SUS) menghasilkan skor rata-rata 69,75 dalam kategori "OK" (C) yang menunjukkan sistem dapat diterima oleh pengguna dan integrasi sebagai widget interaktif cukup mudah dipelajari. Secara keseluruhan, penerapan metode RAG pada chatbot customer service KampusMadu terbukti mampu membantu otomatisasi layanan



pelanggan UMKM secara lebih responsif, informatif, dan kontekstual, dan temuan ini sejalan dengan beberapa penelitian terkait yang juga mengimplementasikan RAG pada berbagai domain, seperti untuk customer support internal [19], chatbot kesehatan mental dengan peningkatan performa pada metrik CIDEr hingga 3,6923 [20], serta pengembangan RAG chatbot dengan teknologi efisien untuk personalized learning [21], dan sistem yang dikembangkan juga memiliki implementasi yang ringan serta sesuai dengan kebutuhan teknis UMKM karena menggunakan vector database lokal dan model embedding yang berjalan secara lokal, sebagaimana pendekatan serupa yang diterapkan oleh Falaq [21] yang mengembangkan RAG chatbot dengan teknologi efisien untuk menekan kebutuhan sumber daya serta menyesuaikan kapasitas dan skala pengguna. Sebagai saran untuk pengembangan selanjutnya, penelitian ini dapat ditingkatkan melalui optimalisasi proses chunking data agar informasi lebih terstruktur, penambahan metadata manfaat produk yang lebih lengkap untuk memperkaya konteks retrieval, penerapan reranking untuk meningkatkan presisi hasil pencarian, serta evaluasi kualitas respons menggunakan jumlah dataset dan skenario pengujian yang lebih besar agar performa chatbot menjadi lebih akurat, stabil, dan mampu menjawab berbagai variasi pertanyaan pelanggan secara lebih komprehensif di masa mendatang.

REFERENCES

- [1] P. Peykani, F. Ramezanlou, C. Tanasescu, and S. Ghanidel, "Large Language Models: A Structured Taxonomy and Review of Challenges, Limitations, Solutions, and Future Directions," *Appl. Sci.*, vol. 15, no. 14, p. 8103, Jul. 2025, doi: 10.3390/app15148103.
- [2] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Adv. Neural Inf. Process. Syst.*, vol. 2020-Decem, Apr. 2021, [Online]. Available: <http://arxiv.org/abs/2005.11401>
- [3] Gufranaka Samudra, Ahmad Turmudi Zy, and Ermanto, "Implementasi Retrieval Augmented Generation (RAG) Dalam Perancangan Chatbot Kesehatan Pencernaan," *JSAI (Journal Sci. Appl. Informatics)*, vol. 8, no. 1, pp. 181–188, Jan. 2025, doi: 10.36085/jsai.v8i1.7678.
- [4] L. P. Dynia and J. S. Wibowo, "Pengembangan Chatbot Dialogflow Untuk Rekomendasi UMKM Kuliner Kota Semarang Berbasis Natural Language Processing," *Techno. Com*, vol. 24, no. 2, pp. 405–418, May 2025, doi: 10.62411/tc.v24i2.12372.
- [5] R. H. Ardiansyah and A. G. Sulaksono, "Layanan pelanggan berbasis Natural Language Processing melalui chatbot pada aplikasi pesan," *J. Inf. Syst. Appl. Dev.*, vol. 1, no. 1, pp. 29–37, Mar. 2023, doi: 10.26905/jisad.v1i1.9858.
- [6] G. D. Albert and A. Voutama, "Pengembangan Chatbot Berbasis Pdf Menggunakan Local Retrieval-Augmented Generation (Rag) Dan Ollama," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 2, Apr. 2025, doi: 10.23960/jitet.v13i2.6361.
- [7] N. Agustina, R. Mardaweni, D. Komara, R. M. Widiastuti, and H. Gusdevi, "Aplikasi Chatbot Desa Wisata Cimindi Menggunakan Natural Language Processing," *Naratif J. Nas. Riset, Apl. dan Tek. Inform.*, vol. 6, no. 1, pp. 42–49, Jun. 2024, doi: 10.53580/naratif.v6i1.282.
- [8] D. R. Mukti and A. Salam, "The Development of a Deployment System Architecture for a Flask-Based Chatbot Using an LSTM NLP Model for Customer Service Question & Answer," *J. Appl. Informatics Comput.*, vol. 9, no. 3, pp. 732–740, 2025, doi: 10.30871/jaic.v9i3.9305.
- [9] A. Saravanos and M. X. Curinga, "Simulating the Software Development Lifecycle: The Waterfall Model," *Appl. Syst. Innov.*, vol. 6, no. 6, p. 108, Nov. 2023, doi: 10.3390/asi6060108.
- [10] Aisya Tyanafisyah *et al.*, "Black Box Testing Using Equivalence Partitioning Technique On Bakkar Website," *J. INSTEK (Informatika Sains dan Teknol.)*, vol. 10, no. 1, pp. 230–238, May 2025, doi: 10.24252/instek.v10i1.52951.
- [11] D. Vujić, A. Njeguš, and N. Baćanin Džakula, "Enhancing Retrieval - Augmented Generation with Graph-Based Retrieval and Generative Modeling," in *Proceedings of the International Scientific Conference - Sinteza 2025*, Beograd, Serbia: Singidunum University, 2025, pp. 3–9. doi: 10.15308/Sinteza-2025-3-9.
- [12] A. Brown, M. Roman, and B. Devereux, "A Systematic Literature Review of Retrieval-Augmented Generation: Techniques, Metrics, and Challenges," *Big Data Cogn. Comput.*, vol. 9, no. 12, 2025, doi: 10.3390/bdcc9120320.
- [13] R. Rosmiati, "Analisis Dan Pengujian Sistem Menggunakan Black Box Testing Equivalence Partitioning," *J. Sains Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 56–63, May 2021, doi: 10.33084/jsakti.v3i2.1932.
- [14] M. Singh, E. Ray, M. Ferradou, and A. Barraza-Urbina, "Metric@CustomerN: Evaluating Metrics at a Customer Level in E-Commerce," *CEUR Workshop Proc.*, vol. 3450, 2023.
- [15] F. V. Krasnov, "Learning-to-Rank in B2B e-commerce catalogs: A digital experiment and conversion analysis," *Bus. Informatics*, vol. 20, no. 1, pp. 54–66, Mar. 2026, doi: 10.17323/2587-814X.2026.1.54.66.
- [16] F. I. Apridal and F. Nova, "Development of an Interactive Chatbot Using Sahabat-AI Model with Retrieval-Augmented Generation Method," vol. 19, no. 1, pp. 1380–1396, 2026, doi: 10.24036/jtip.v19i1.1105.
- [17] Y. Mao *et al.*, "A Survey on LoRA of Large Language Models," *Front. Comput. Sci.*, vol. 19, no. 7, p. 197605, Oct. 2024, doi: 10.1007/s11704-024-40663-9.
- [18] T. T. Procko and O. Ochoa, "Graph Retrieval-Augmented Generation for Large Language Models: A Survey," in *2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, IEEE, Sep. 2024, pp. 166–169. doi: 10.1109/AIxSET62544.2024.00030.
- [19] C. Duarte Brito, A. Luís Lopes, and F. Batista, "Chatbot based on LLMs for Support and Development Teams," *Atas da Conf. da Assoc. Port. Sist. Inf.*, 2025, [Online]. Available: <https://aisel.aisnet.org/capsi2025/20/>
- [20] Dhia Tawakalna and Junta Zeniarja, "Sistem Chatbot Kesehatan Mental Berbasis Llm Dengan Deteksi Emosi Dan Retrieval Augmented Generation," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 11, no. 1, pp. 1398–1412, Jan. 2026, doi: 10.36341/rabit.v11i1.7347.
- [21] F. I. Falaq and N. Saputra, "Pengembangan Retrieval-Augmented Generation (Rag) Chatbot Menggunakan Metode Design Thinking Untuk Mendukung Personalized Learning," *PINTER J. Pendidik. Tek. Inform. dan Komput.*, vol. 9, no. 2, pp. 1–10, Dec. 2025, doi: 10.21009/pinter.9.2.1.