

RANCANG BANGUN CHATBOT REKOMENDASI PRODUK SKINCARE BERBASIS RETRIEVAL-AUGMENTED GENERATION PADA TiDB SERVERLESS

Gerald Dustin Albert^{1*}, Nina Sulistiyowati², Aziz Ma'shum³

^{1,2,3}Universitas Singaperbangsa Karawang; Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361; Telp. (0267) 6

Keywords:

Chatbot;
RAG;
Hybrid Retrieval;
TiDB Serverless;
Skincare Recommendation.

Correspondent Email:

albertgerald26@gmail.com

Abstrak. Pemilihan produk *skincare* sering menjadi masalah karena pengguna memiliki jenis kulit, kebutuhan, preferensi, dan batasan harga yang berbeda. Pencarian manual melalui *marketplace* atau media sosial sering tidak terstruktur, kurang personal, dan membutuhkan waktu lama, sedangkan layanan admin tidak selalu mampu merespons pertanyaan pelanggan secara cepat. Penelitian ini bertujuan merancang dan membangun *chatbot* rekomendasi produk *skincare* berbasis *Retrieval-Augmented Generation* (RAG) pada *TiDB Serverless*. Metode pengembangan sistem menggunakan *Extreme Programming* (XP) dengan tahapan *planning*, *design*, *coding*, dan *testing*. Sistem dikembangkan sebagai aplikasi web menggunakan Next.js, React, TypeScript, Drizzle ORM, TiDB *Serverless*, Ollama, model LLM Qwen2.5:7B, dan *embedding* BAAI/BGE-M3. Pendekatan *hybrid retrieval* diterapkan dengan menggabungkan *vector search* dan *structured query* agar sistem mampu memahami pertanyaan berbasis bahasa alami sekaligus menangani filter seperti harga. Hasil pengujian *black box* terhadap 15 skenario menunjukkan seluruh fungsi utama berhasil berjalan, meliputi rekomendasi produk, penolakan pertanyaan di luar topik, filter harga, login admin, pengelolaan produk, inventori, pesanan, *chat logs*, dan *cache response*. Sistem ini menunjukkan bahwa integrasi RAG, *hybrid retrieval*, dan *TiDB Serverless* dapat mendukung rekomendasi *skincare* secara lebih interaktif, kontekstual, efisien, dan berbasis data.



Copyright © 2026 The Author(s),
Published by Jurnal Informatika dan
Teknik Elektro Terapan. This is an
open-access article distributed under
the terms of the Creative Commons
Attribution-NonCommercial 4.0
International License (CC BY-NC
4.0).

Abstract. *Selecting skincare products is often difficult because users have different skin types, needs, preferences, and price ranges. Manual searching through marketplaces or social media is often unstructured, less personalized, and time-consuming. This study aims to design and build a skincare product recommendation chatbot based on Retrieval-Augmented Generation (RAG) using TiDB Serverless. The system development method used Extreme Programming (XP), consisting of planning, design, coding, and testing stages. The system was developed as a web application using Next.js, React, TypeScript, Drizzle ORM, TiDB Serverless, Ollama, Qwen2.5:7B LLM, and BAAI/BGE-M3 embedding model. A hybrid retrieval approach was implemented by combining vector search and structured query so the system can understand natural language questions while handling filters such as price. Black box testing on 15 scenarios showed that all main functions worked successfully, including product recommendation, out-of-topic rejection, price filtering, admin login, product management, inventory, order management, chat logs, and response caching. The system supports skincare recommendation delivery in a more interactive, contextual, and data-driven manner.*

1. PENDAHULUAN

Industri skincare berkembang pesat seiring meningkatnya kesadaran masyarakat terhadap perawatan kulit. Banyaknya produk dengan fungsi, kandungan bahan aktif, harga, dan target jenis kulit berbeda memberi banyak pilihan bagi konsumen, tetapi juga menimbulkan kesulitan dalam menentukan produk yang tepat. Pengguna dengan kulit kering, berminyak, sensitif, berjerawat, atau kombinasi membutuhkan rekomendasi berbeda. Kesalahan memilih produk dapat menyebabkan hasil tidak efektif, iritasi, jerawat, bahkan kerusakan skin barrier [1], [2].

Proses pencarian informasi produk skincare secara manual melalui marketplace, media sosial, atau toko online sering tidak terstruktur dan kurang personal. Berdasarkan wawancara dengan pemilik usaha skincare, pelanggan umumnya bertanya melalui WhatsApp mengenai kondisi kulit, rekomendasi produk, manfaat, harga, dan batasan budget. Admin masih membalas pertanyaan satu per satu sehingga layanan menjadi lambat ketika pesan masuk banyak atau berada di luar jam operasional. Kondisi ini menunjukkan kebutuhan sistem yang mampu memberikan rekomendasi produk secara otomatis, interaktif, dan berbasis data.

Perkembangan kecerdasan buatan, Natural Language Processing (NLP), Large Language Model (LLM), dan chatbot membuka peluang pengembangan sistem rekomendasi berbasis percakapan [3], [4]. Chatbot berbasis LLM mampu memahami pertanyaan bahasa alami dan menghasilkan respons yang lebih natural. Namun, LLM memiliki keterbatasan karena bergantung pada pengetahuan internal dan dapat menghasilkan hallucination, yaitu informasi yang tidak akurat atau tidak sesuai fakta [5], [6]. Pada domain rekomendasi produk yang berkaitan dengan kondisi kulit, kesalahan informasi dapat merugikan pengguna.

Retrieval-Augmented Generation (RAG) menjadi pendekatan untuk mengurangi keterbatasan tersebut dengan menggabungkan proses retrieval dari sumber data eksternal dan generation oleh LLM [7], [8]. Melalui RAG, jawaban chatbot dapat diarahkan berdasarkan data produk yang tersedia. Namun, retrieval berbasis satu metode seperti vector search belum selalu cukup untuk menangani query kompleks, misalnya kebutuhan kulit sekaligus

batasan harga. Karena itu, penelitian ini menerapkan hybrid retrieval yang menggabungkan vector search dan structured query agar rekomendasi lebih relevan secara semantik maupun terstruktur [9], [10].

Penelitian ini bertujuan merancang dan membangun chatbot rekomendasi produk skincare berbasis RAG pada TiDB Serverless. Kebaruan penelitian terletak pada integrasi RAG, hybrid retrieval, LLM lokal, embedding model, dan TiDB Serverless yang mendukung penyimpanan data relasional sekaligus vector search dalam satu basis data. Sistem dikembangkan menggunakan metode Extreme Programming (XP) agar proses pengembangan dapat dilakukan secara iteratif dan adaptif terhadap kebutuhan pengguna.

2. TINJAUAN PUSTAKA

2.1. Chatbot dan Natural Language Processing

Chatbot merupakan program komputer yang dirancang untuk mensimulasikan percakapan dengan pengguna melalui teks atau suara. Chatbot modern banyak memanfaatkan NLP dan machine learning agar mampu memahami konteks percakapan dan memberikan respons yang lebih relevan [11]. NLP sendiri berfokus pada pemrosesan bahasa alami agar sistem dapat memahami, menginterpretasikan, dan menghasilkan teks manusia secara efektif [12]. Dalam sistem rekomendasi skincare, chatbot berperan sebagai antarmuka percakapan agar pengguna dapat menyampaikan kebutuhan produk menggunakan bahasa sehari-hari.

2.2. Large Language Model dan Retrieval

Augmented Generation Large Language Model merupakan model deep learning berbasis transformer yang dilatih menggunakan data teks besar untuk memahami dan menghasilkan bahasa alami [13]. LLM mampu menjawab pertanyaan, merangkum teks, dan menghasilkan respons percakapan, tetapi memiliki risiko hallucination ketika menghasilkan informasi yang tidak berdasar pada data valid [5], [6]. RAG dikembangkan untuk mengatasi masalah ini dengan mengambil data relevan dari sumber eksternal sebelum proses generasi respons [7]. Dalam pipeline RAG, query pengguna diubah menjadi embedding, data relevan diambil dari basis

data, lalu data tersebut dimasukkan sebagai konteks bagi LLM.

2.3. *Vector Search dan Hybrid Retrieval*

Vector embedding merepresentasikan teks dalam bentuk vektor numerik sehingga sistem dapat menangkap makna semantik dari query dan data [14]. Vector search mencari data yang paling mirip secara semantik menggunakan ukuran kemiripan seperti cosine distance atau nearest neighbor [15]. Hybrid retrieval menggabungkan vector search dengan structured query sehingga sistem dapat menangani kebutuhan semantik dan atribut terstruktur seperti harga, kategori, atau jenis kulit [9], [10]. Pendekatan ini cocok untuk sistem rekomendasi produk karena pengguna sering memberikan pertanyaan campuran, misalnya “rekomendasi moisturizer untuk kulit berminyak di bawah 100 ribu”.

2.4. *Sistem Rekomendasi Produk*

Sistem rekomendasi dirancang untuk membantu pengguna memilih item berdasarkan preferensi, kebutuhan, atau perilaku tertentu [16]. Dalam konteks skincare, sistem rekomendasi dapat membantu pengguna memilih produk berdasarkan jenis kulit, manfaat, kandungan, dan harga. Penelitian sebelumnya menunjukkan bahwa sistem rekomendasi berbasis AI dan content-based learning dapat membantu personalisasi rekomendasi skincare [1], [2]. Integrasi sistem rekomendasi dengan chatbot membuat proses pencarian menjadi lebih interaktif dan kontekstual.

2.5. *Extreme Programming*

Extreme Programming (XP) merupakan metode pengembangan perangkat lunak berbasis Agile yang cocok untuk proyek dengan kebutuhan dinamis. XP menggunakan iterasi singkat, user stories, acceptance criteria, simple design, coding, refactoring, dan testing [17]. Dalam penelitian ini, XP digunakan karena pengembangan chatbot berbasis AI membutuhkan eksperimen, evaluasi, dan penyempurnaan fitur secara bertahap.

3. METODE PENELITIAN

3.1. *Objek Penelitian*

Objek penelitian ini adalah sistem chatbot rekomendasi produk skincare berbasis

kecerdasan buatan yang dikembangkan untuk mendukung layanan rekomendasi produk secara digital. Sistem dirancang untuk membantu pengguna memperoleh rekomendasi produk sesuai kebutuhan kulit melalui percakapan. Data utama sistem berupa produk skincare yang tersimpan dalam basis data, meliputi nama produk, kategori, deskripsi, manfaat, jenis kulit, harga, dan embedding.

3.2. *Metode Pengembangan Sistem*

Metode pengembangan sistem menggunakan Extreme Programming (XP) dengan tahapan planning, design, coding, dan testing. Tahap planning digunakan untuk mengidentifikasi kebutuhan pengguna, menyusun user stories, acceptance criteria, kebutuhan fungsional, kebutuhan nonfungsional, perangkat keras, dan perangkat lunak. Tahap design digunakan untuk merancang alur sistem, proses bisnis, struktur basis data, interaksi sistem, dan wireframe antarmuka. Tahap coding digunakan untuk mengimplementasikan aplikasi secara iteratif. Tahap testing menggunakan black box testing untuk memverifikasi fungsi berdasarkan input dan output.

3.3. *Kebutuhan Sistem*

Kebutuhan fungsional sistem meliputi kemampuan menerima input bahasa alami, memberikan rekomendasi produk skincare, menampilkan informasi produk, melakukan retrieval berdasarkan jenis kulit atau harga, mengelola data produk melalui dashboard admin, dan menyimpan riwayat percakapan. Kebutuhan perangkat keras minimal adalah laptop atau komputer dengan prosesor AMD Ryzen 5 atau Intel Core i5, RAM 16 GB, SSD 512 GB, dan koneksi internet. Kebutuhan perangkat lunak meliputi Next.js, React, TypeScript, Tailwind CSS, Node.js, Drizzle ORM, TiDB Serverless, Ollama, Qwen2.5:7B, BAAI/BGE-M3, Visual Studio Code, Git, dan GitHub.

3.4. *Arsitektur Sistem*

Sistem dikembangkan sebagai aplikasi web full-stack. Pengguna berinteraksi melalui halaman chatbot. Query pengguna dikirim ke backend Next.js API Routes. Backend mengubah query menjadi vector embedding menggunakan BAAI/BGE-M3, lalu

menjalankan retrieval pada TiDB Serverless. Vector search dilakukan menggunakan fungsi `VEC_COSINE_DISTANCE()` untuk mencari produk yang relevan secara semantik. Jika query mengandung batasan harga, sistem menjalankan structured query menggunakan SQL filtering. Hasil vector search dan structured query digabungkan, dideduplikasi, lalu dimasukkan sebagai konteks ke prompt LLM Qwen2.5:7B melalui Ollama. LLM menghasilkan respons rekomendasi yang ditampilkan ke pengguna secara streaming.

3.5. Implementasi Hybrid Retrieval

Hybrid retrieval diterapkan dengan menggabungkan pencarian semantik dan pencarian terstruktur. Pencarian semantik digunakan untuk memahami maksud pengguna berdasarkan embedding query. Pencarian terstruktur digunakan untuk memfilter data berdasarkan atribut harga. Sistem menggunakan regex untuk mendeteksi bahasa informal terkait harga, seperti “di bawah 100 ribu”, “sekitar 150rb”, atau “lebih dari 200k”. Hasil deteksi menentukan query SQL yang digunakan. Pendekatan ini membuat chatbot mampu menangani pertanyaan deskriptif dan numerik dalam satu pipeline retrieval.

3.6. Metode Pengujian

Pengujian dilakukan menggunakan black box testing terhadap fungsi utama sistem. Skenario pengujian meliputi pertanyaan rekomendasi produk, pertanyaan di luar topik, filter harga, login admin valid dan invalid, tambah produk, validasi data produk, edit produk, hapus produk, stok inbound, stok outbound melebihi stok, pembuatan pesanan, perubahan status pesanan, chat logs, dan cache response. Hasil pengujian dinilai berdasarkan kesesuaian antara keluaran aktual dan keluaran yang diharapkan.

4. HASIL DAN PEMBAHASAN

4.1. Hasil dan Identifikasi Masalah

Hasil analisis menunjukkan bahwa proses pemilihan produk skincare masih banyak dilakukan secara manual. Pengguna mencari informasi produk melalui marketplace, media sosial, atau chat dengan admin, lalu membaca deskripsi dan membandingkan produk secara mandiri. Cara ini membutuhkan waktu lama dan bergantung pada kemampuan pengguna

memahami informasi produk. Wawancara dengan pemilik usaha menunjukkan bahwa pelanggan sering bertanya mengenai kondisi kulit, rekomendasi produk, manfaat, harga, dan budget. Pertanyaan berulang seperti rekomendasi untuk kulit berjerawat atau produk di bawah Rp100.000 muncul hampir setiap hari. Karena admin tidak selalu online 24 jam, pelanggan kadang harus menunggu lama.

Berdasarkan masalah tersebut, sistem yang diusulkan adalah chatbot rekomendasi produk skincare berbasis RAG. Sistem ini memungkinkan pengguna mengirim pertanyaan dalam bahasa alami, kemudian sistem melakukan retrieval data produk dan menghasilkan rekomendasi berdasarkan konteks pertanyaan pengguna.

4.2. Hasil Perancangan Sistem

Perancangan sistem menghasilkan dua aktor utama, yaitu User dan Admin. User dapat melakukan percakapan dengan chatbot, mencari rekomendasi produk skincare, dan melihat informasi produk. Admin dapat login dan mengelola data produk yang digunakan dalam proses retrieval. Struktur data sistem terdiri dari tabel `products`, `chat_sessions`, `chat_messages`, dan `admins`. Tabel `products` menyimpan informasi produk dan embedding. Tabel `chat_sessions` menyimpan sesi percakapan. Tabel `chat_messages` menyimpan riwayat pesan. Tabel `admins` menyimpan data administrator.

Alur rekomendasi dimulai ketika user membuka halaman chatbot dan mengirim pertanyaan. Sistem memproses query, melakukan retrieval data produk, menginjeksi konteks ke prompt, menghasilkan respons menggunakan LLM, lalu menampilkan rekomendasi kepada user. Pada sisi admin, alur dimulai dari login, masuk dashboard, lalu melakukan tambah, ubah, atau hapus data produk. Perubahan data disimpan ke TiDB Serverless agar dapat digunakan kembali oleh chatbot.

4.3. Hasil Implementasi Sistem

Implementasi dilakukan dalam tiga iterasi utama. Iterasi pertama menghasilkan fondasi aplikasi, meliputi inisialisasi Next.js App Router, backend API Routes, integrasi TiDB Serverless menggunakan Drizzle ORM,

otentikasi admin menggunakan Better Auth, halaman landing page, halaman chatbot dasar, dashboard admin, dan struktur folder frontend-backend.

Iterasi kedua menghasilkan fitur utama chatbot dan RAG. Sistem mengintegrasikan LLM Qwen2.5:7B melalui Ollama, embedding BAAI/BGE-M3, vector embedding untuk data produk dan query pengguna, vector search menggunakan TiDB Vector Search, pipeline RAG, context injection menggunakan tag <context>, antarmuka chatbot, streaming response, penyimpanan riwayat percakapan, dan markdown rendering. Pada pipeline RAG, query pengguna diubah menjadi embedding, lalu sistem mencari produk paling relevan di TiDB. Hasil retrieval digunakan sebagai konteks agar respons LLM lebih sesuai dengan data produk.

Iterasi ketiga menghasilkan hybrid retrieval dan fitur pendukung. Sistem menggabungkan vector search dan structured query, mendeteksi filter harga menggunakan regex, menjalankan retrieval paralel, melakukan deduplikasi hasil retrieval, menerapkan session persistence menggunakan localStorage, suggested questions, chat logs admin, pengelolaan inventori, dan pengelolaan pesanan. Implementasi ini meningkatkan fleksibilitas sistem ketika pengguna memberikan pertanyaan campuran, misalnya berdasarkan jenis kulit dan rentang harga.

4.4. Hasil Pengujian

Pengujian dilakukan menggunakan metode black box pada 15 skenario untuk memastikan fungsi utama sistem berjalan sesuai kebutuhan. Hasil pengujian menunjukkan bahwa seluruh skenario memperoleh status berhasil. Pada skenario rekomendasi produk, sistem mampu menerima masukan “rekomendasi untuk kulit berminyak” dan menampilkan rekomendasi produk yang relevan dari basis data. Pada skenario pertanyaan di luar topik, sistem mampu menolak pertanyaan seperti “siapa presiden indonesia?” dan mengarahkan kembali percakapan pada konteks skincare.

Pengujian terhadap fitur hybrid retrieval juga menunjukkan hasil sesuai harapan. Ketika pengguna memberikan masukan “produk di bawah 100 ribu”, sistem mampu mendeteksi batasan harga dan menampilkan produk

dengan harga sesuai rentang yang diminta. Hal ini menunjukkan bahwa kombinasi vector search dan structured query dapat membantu sistem menangani query berbasis konteks sekaligus query berbasis atribut numerik.

Pada sisi admin, pengujian login valid berhasil mengarahkan admin ke dashboard, sedangkan login dengan password salah berhasil menampilkan pesan kesalahan. Fitur pengelolaan produk juga berjalan sesuai kebutuhan, meliputi penambahan produk baru, validasi ketika nama produk kosong, pengubahan deskripsi produk, dan penghapusan produk. Produk yang ditambahkan dapat tersimpan, menghasilkan embedding, muncul pada daftar produk, serta dapat ditemukan kembali oleh chatbot. Ketika data produk diperbarui, chatbot juga mampu menampilkan informasi terbaru.

Pengujian fitur pendukung menunjukkan bahwa sistem berhasil mencatat stok masuk, menolak stok keluar yang melebihi jumlah tersedia, membuat pesanan dengan status pending, mengubah status pesanan, menampilkan daftar sesi percakapan pada halaman chat logs, serta memberikan respons lebih cepat ketika pengguna mengirim pertanyaan yang sama karena mekanisme cache response. Berdasarkan hasil tersebut, sistem chatbot rekomendasi produk skincare berbasis RAG dinilai telah memenuhi kebutuhan fungsional, baik pada sisi pengguna maupun admin.

4.5. Pembahasan

Hasil implementasi menunjukkan bahwa chatbot rekomendasi produk skincare berbasis RAG mampu memberikan rekomendasi secara lebih kontekstual dibandingkan pencarian berbasis kata kunci. Integrasi LLM, vector embedding, vector search, dan data produk membuat sistem dapat memahami pertanyaan pengguna dalam bahasa alami. Penggunaan context injection membantu respons LLM tetap berada pada konteks data produk yang tersedia sehingga risiko jawaban tidak relevan dapat dikurangi.

Hybrid retrieval menjadi bagian penting karena pengguna tidak hanya bertanya berdasarkan deskripsi kebutuhan kulit, tetapi juga atribut numerik seperti harga. Vector search membantu sistem memahami makna semantik, sedangkan structured query

membantu sistem memfilter produk berdasarkan batasan harga. Kombinasi keduanya membuat sistem lebih fleksibel dalam menangani query kompleks. Hasil ini sejalan dengan konsep RAG dan hybrid retrieval yang menyatakan bahwa penggabungan retrieval semantik dan terstruktur dapat meningkatkan relevansi respons [7], [9], [10].

Penerapan TiDB Serverless juga menyederhanakan arsitektur sistem karena data relasional dan vector embedding dapat dikelola dalam satu basis data. Sistem tidak memerlukan layanan vector database tambahan. Hal ini mendukung pengembangan aplikasi AI yang lebih ringkas dan terintegrasi. Penggunaan XP membantu proses pengembangan dilakukan bertahap, mulai dari fondasi aplikasi, chatbot RAG, hingga hybrid retrieval dan fitur admin. Setiap iterasi menghasilkan peningkatan fitur yang langsung dapat diuji terhadap acceptance criteria.

Meskipun sistem berhasil berjalan, terdapat beberapa keterbatasan. Sistem masih dijalankan pada lingkungan lokal dan belum diimplementasikan pada produksi publik. Performa LLM lokal melalui Ollama bergantung pada spesifikasi perangkat keras. Sistem juga belum mendukung analisis kondisi kulit berbasis gambar dan belum melakukan evaluasi performa AI secara kuantitatif menggunakan precision, recall, atau benchmark model lain. Karena itu, penelitian lanjutan perlu menambah evaluasi kuantitatif, deployment produksi, personalisasi berbasis histori pengguna, dan fitur computer vision.

5. KESIMPULAN

- a. Hasil implementasi menunjukkan bahwa chatbot rekomendasi produk skincare berbasis RAG mampu memberikan rekomendasi secara lebih kontekstual dibandingkan pencarian berbasis kata kunci. Integrasi LLM, vector embedding, vector search, dan data produk membuat sistem dapat memahami pertanyaan pengguna dalam bahasa alami. Penggunaan context injection membantu respons LLM tetap berada pada konteks data produk yang tersedia sehingga risiko jawaban tidak relevan dapat dikurangi.
- b. Hybrid retrieval menjadi bagian penting karena pengguna tidak hanya

bertanya berdasarkan deskripsi kebutuhan kulit, tetapi juga atribut numerik seperti harga. Vector search membantu sistem memahami makna semantik, sedangkan structured query membantu sistem memfilter produk berdasarkan batasan harga. Kombinasi keduanya membuat sistem lebih fleksibel dalam menangani query kompleks. Hasil ini sejalan dengan konsep RAG dan hybrid retrieval yang menyatakan bahwa penggabungan retrieval semantik dan terstruktur dapat meningkatkan relevansi respons [7], [9], [10].

- c. Penerapan TiDB Serverless juga menyederhanakan arsitektur sistem karena data relasional dan vector embedding dapat dikelola dalam satu basis data. Sistem tidak memerlukan layanan vector database tambahan. Hal ini mendukung pengembangan aplikasi AI yang lebih ringkas dan terintegrasi. Penggunaan XP membantu proses pengembangan dilakukan bertahap, mulai dari fondasi aplikasi, chatbot RAG, hingga hybrid retrieval dan fitur admin. Setiap iterasi menghasilkan peningkatan fitur yang langsung dapat diuji terhadap acceptance criteria.
- d. Meskipun sistem berhasil berjalan, terdapat beberapa keterbatasan. Sistem masih dijalankan pada lingkungan lokal dan belum diimplementasikan pada produksi publik. Performa LLM lokal melalui Ollama bergantung pada spesifikasi perangkat keras. Sistem juga belum mendukung analisis kondisi kulit berbasis gambar dan belum melakukan evaluasi performa AI secara kuantitatif menggunakan precision, recall, atau benchmark model lain. Karena itu, penelitian lanjutan perlu menambah evaluasi kuantitatif, deployment produksi, personalisasi berbasis histori pengguna, dan fitur computer vision.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang, dosen pembimbing, serta pihak usaha skincare yang telah memberikan dukungan dan informasi dalam pelaksanaan

penelitian ini. Atas segala bantuan dan kerjasamanya semoga menjadi amal Saudara yang diberkahi rahmat yang melimpah dari Allah SWT

DAFTAR PUSTAKA

- [1] A. Sindhu, A. Suresh, S. G. Shaila, B. Bindu, B. Akshaya, and P. Priyanka, "Personalized Skincare Recommender System Using Deep Learning," in *2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, 2025. doi: 10.1109/ICAECA63854.2025.11012227.
- [2] J. Lee, H. Yoon, S. Kim, C. Lee, J. Lee, and S. Yoo, "Deep learning-based skin care product recommendation: A focus on cosmetic ingredient analysis and facial skin conditions," *Journal of Cosmetic Dermatology*, vol. 23, pp. 2066–2077, 2024. doi: 10.1111/jocd.16218.
- [3] S. Nurhaliza, B. Bustami, and H. A. K. Aidilof, "Implementasi Chatbot AI untuk Rekomendasi Produk Skincare Menggunakan Natural Language Processing," *Jurnal Ilmiah Global Education*, vol. 6, no. 2, pp. 294–303, 2025. doi: 10.55681/JIGE.V6I2.3802.
- [4] N. Shrivastava, P. Tewari, S. Sujatha, S. R. Bogireddy, N. Varshney, and V. Sharma, "Natural Language Processing for Conversational AI: Chatbots and Virtual Assistants," in *2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (LATMSI)*, 2025. doi: 10.1109/IATMSI64286.2025.10984818.
- [5] Z. Ji et al., "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, vol. 55, no. 12, 2023. doi: 10.1145/3571730.
- [6] S. Gupta, "Retrieval-Augmented Generation and Hallucination in Large Language Models: A Scholarly Overview," *Scholars Journal of Engineering and Technology*, vol. 13, no. 05, pp. 328–330, 2025. doi: 10.36347/SJET.2025.V13I05.003.
- [7] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," in *Proceedings - 2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, pp. 166–169, 2024. doi: 10.1109/AIxSET62544.2024.00030.
- [8] S. Sobhan and M. A. Haque, "LLM-Assisted Question-Answering on Technical Documents Using Structured Data-Aware Retrieval Augmented Generation," 2025. [Online]. Available: <https://arxiv.org/pdf/2506.23136>
- [9] K. Sawarkar, A. Mangal, and S. R. Solanki, "Blended RAG: Improving RAG (Retriever-Augmented Generation) Accuracy with Semantic Search and Hybrid Query-Based Retrievers," in *Proceedings of the International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 155–161, 2024. doi: 10.1109/MIPR62202.2024.00031.
- [10] Y. Wan, Z. Chen, Y. Liu, C. Chen, and M. Packianather, "Empowering LLMs by hybrid retrieval-augmented generation for domain-centric Q&A in smart manufacturing," *Advanced Engineering Informatics*, vol. 65, 2025. doi: 10.1016/J.AEI.2025.103212.
- [11] C.-C. Lin et al., "A Review of AI-Driven Conversational Chatbots Implementation Methodologies and Challenges (1999–2022)," *Sustainability*, vol. 15, no. 5, 2023. doi: 10.3390/SU15054012.
- [12] S. Singh and A. Mahmood, "The NLP Cookbook: Modern Recipes for Transformer Based Deep Learning Architectures," *IEEE Access*, vol. 9, pp. 68675–68702, 2021. doi: 10.1109/ACCESS.2021.3077350.
- [13] M. A. K. Raiaan et al., "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges," *IEEE Access*, vol. 12, pp. 26839–26874, 2024. doi: 10.1109/ACCESS.2024.3365742.
- [14] D. S. Asudani, N. K. Nagwani, and P. Singh, "Impact of word embedding models on text analytics in deep learning environment: a review," *Artificial Intelligence Review*, vol. 56, no. 9, pp. 10345–10425, 2023. doi: 10.1007/S10462-023-10419-1.
- [15] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2021. doi: 10.1109/TBDATA.2019.2921572.
- [16] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities," *Applied Sciences*, vol. 10, no. 21, 2020. doi: 10.3390/APP10217748.
- [17] A. Shrivastava, I. Jaggi, N. Katoch, D. Gupta, and S. Gupta, "A Systematic Review on Extreme Programming," *Journal of Physics: Conference Series*, vol. 1969, 2021. doi: 10.1088/1742-6596/1969/1/012046.
- [18] G. D. Albert and A. Voutama, "Pengembangan Chatbot Berbasis Pdf Menggunakan Local Retrieval-Augmented Generation (Rag) Dan Ollama," *Jurnal Informatika Dan Teknik*

Elektro Terapan, vol. 13, no. 2, pp. 2830–
7062, 2025. doi:
10.23960/JITET.V13I2.6361.