https://jurnal.pustakagalerimandiri.co.id/index.php/pustakaai
DOI : https://doi.org/10.55382/jurnalpustakaai.v6i2.1768

JURNAL PUSTAKA AI
JURNAL PUSAT AKSES KAJIAN TEKNOLOGI ARTIFICIAL INTELLIGENCE
Pustaka Galeri Mandiri
Vol. 6 No. 2 (2026) 189 – 200
E ISSN: 2809-4069

Implementasi *Retrieval-Augmented Generation* untuk *Chatbot* Layanan Pelanggan PT. GAWS Inti Solusi

Verio Aberossy Renedominick[1], Simon Pranata Bagus[2]
[12]Program Studi Informatika, Fakultas Sains, Komputer dan Matematika, Universitas
[1]verio.renedominick@student.matanauniversity.ac.id [2]simon.barus@matanauniversity.ac.id

Abstract

The advancement of Artificial Intelligence (AI), particularly Large Language Models (LLMs) has enabled the automation of customer support services. However, conventional generative models still face contextual limitations, primarily such as hallucination and the inability to accurately reference proprietary company information in real-time. This study aims to implement a Retrieval-Augmented Generation (RAG) architecture within a customer support chatbot for PT. Gaws Inti Solusi to address high dependency on company leadership, inconsistent manual information retrieval and delayed service responses. Development was executed by using the Agile-Scrum methodology across four iterative sprints that integrated with LangChain, ChromaDB as a vector database and the locally deployed mistral: instruct model. Comprehensive evaluation yielded concrete quantitative findings are average response time decreased to 9.97 seconds (P95 latency: 17.04 seconds), model confidence reached 90%, similarity score stood at 37% and the technical error rate remained at 0%. The system successfully fulfilled 70% of the defined functional requirements and is validated for beta release. These outcomes confirm that LLM contextual constraints can be effectively mitigated through verified document retrieval, delivering substantial practical impacts in operational efficiency, reduced administrative workload and scalable service delivery. Future research is recommended to develop multi-turn conversation capabilities, integrate with external communication platforms and conduct external User Acceptance Testing (UAT) to quantify direct improvements in Customer Satisfaction Score (CSAT).

Keywords: Chatbot, Retrieval-Augmented Generation, RAG, LLM, Customer Support, LangChain, ChromaDB, PT Gaws Inti Solusi

Abstrak

Perkembangan *Artificial Intelligence* (*AI*), khususnya *Large Language Models* (*LLM*), telah membuka peluang otomatisasi layanan pelanggan, namun penerapan model generatif konvensional masih menghadapi keterbatasan konteks berupa *hallucination* dan ketidakmampuan merujuk pada informasi spesifik perusahaan secara akurat. Penelitian ini bertujuan mengimplementasikan arsitektur *Retrieval-Augmented Generation* (*RAG*) pada sistem *chatbot* PT Gaws Inti Solusi untuk mengatasi ketergantungan tinggi pada pimpinan, inkonsistensi informasi manual, dan lambatnya respons layanan. Pengembangan dilaksanakan melalui metodologi *Agile-Scrum* dalam empat iterasi, dengan mengintegrasikan *LangChain*, basis data vektor *ChromaDB*, dan model *mistral:instruct* yang dijalankan secara lokal. Evaluasi komprehensif menghasilkan temuan kuantitatif yang konkret: waktu respons layanan berkurang menjadi rata-rata 9,97 detik (*P95 latency* 17,04 detik), tingkat kepercayaan model mencapai 90%, *similarity score* sebesar 37%, dan tingkat kesalahan teknis nol persen. Sistem telah memenuhi 70% kebutuhan fungsional dan dinyatakan layak beroperasi sebagai versi *beta*. Keberhasilan ini membuktikan bahwa pembatasan konteks pada *LLM* dapat diatasi melalui mekanisme *retrieval* berbasis dokumen terverifikasi, yang secara langsung memberikan dampak praktis berupa efisiensi operasional, pengurangan beban administratif pimpinan, dan skalabilitas layanan tanpa intervensi manusia. Pengembangan selanjutnya direkomendasikan untuk mengintegrasikan kemampuan percakapan *multi-turn*, memperluas cakupan ke platform komunikasi eksternal,

Submitted : 01-06-2026 | Reviewed : 05-06-2026 | Accepted : 30-06-2026

serta mengukur dampak nyata sistem melalui pengujian *User Acceptance Testing* (*UAT*) eksternal dan pelacakan *Customer Satisfaction Score* (*CSAT*).
Kata kunci: *Chatbot, Retrieval-Augmented Generation*, *RAG, LLM*, Layanan Pelanggan, *LangChain*, *ChromaDB*, PT Gaws Inti Solusi

© 2026 Author
Creative Commons Attribution 4.0 International License

1. Pendahuluan

Perkembangan transformasi digital telah mendorong adopsi kecerdasan buatan (*Artificial Intelligence*/AI) secara luas di berbagai sektor industri [1]. Dalam konteks akses informasi, AI juga telah terbukti mempermudah distribusi sumber pengetahuan yang relevan bagi pengguna [2]. Salah satu implementasi AI yang paling banyak diterapkan adalah *chatbot*, yaitu program yang mensimulasikan percakapan manusia melalui pemrosesan bahasa alami (*Natural Language Processing*/NLP) [3]. Melalui NLP, *chatbot* mampu merespons permintaan pengguna secara instan, sehingga banyak digunakan untuk mendukung *Customer Relationship Management* (CRM) dan otomatisasi layanan [4]. Namun, sistem *chatbot* generasi awal yang bersifat *rule-based* maupun berbasis model statistik tradisional masih menghadapi keterbatasan dalam memahami konteks percakapan yang dinamis dan menangani pertanyaan yang bersifat spesifik [5]. Di sisi lain, pengelolaan layanan pelanggan secara manual juga terbukti tidak efisien, karena memerlukan analisis data tidak terstruktur secara parsial yang berujung pada perlambatan waktu respons dan penurunan kualitas layanan [6]. Tantangan serupa juga ditemukan dalam analisis data media sosial, di mana volume besar membuat pemrosesan manual menjadi tidak praktis [7].

Kemunculan *Large Language Model* (LLM) membawa perubahan signifikan dalam kapasitas pemahaman dan generasi teks bahasa manusia. Model ini mampu menghasilkan respons yang koheren dan adaptif berdasarkan konteks *prompt* yang diberikan [8]. Kendati demikian, LLM umum masih rentan terhadap *hallucination*, yaitu fenomena di mana model menghasilkan informasi yang terlihat logis namun tidak akurat atau tidak didukung oleh fakta [9]. Keterbatasan ini menjadi tantangan krusial, terutama ketika LLM diterapkan pada lingkungan perusahaan yang memerlukan keandalan informasi berbasis dokumen internal yang terverifikasi [10].

Untuk mengatasi kelemahan tersebut, pendekatan *Retrieval-Augmented Generation* (RAG) telah berkembang sebagai *state of the art* dalam pengembangan sistem berbasis LLM. RAG mengintegrasikan modul *retrieval* dengan model generatif, sehingga sistem dapat mengambil informasi relevan dari basis pengetahuan eksternal sebelum menghasilkan respons [11]. Mekanisme ini secara empiris terbukti menurunkan tingkat *hallucination* sekaligus meningkatkan akurasi dan relevansi jawaban [12]. Dalam implementasinya, penyimpanan representasi teks (*embedding*) memerlukan sistem manajemen basis data yang mendukung integrasi pencarian semantik secara *real-time* [13]. Kerangka kerja *LangChain* mempermudah orkestrasi alur kerja LLM dengan sumber data eksternal [14], sementara pengembangan antarmuka aplikasi dan layanan pendukung dapat diakselerasi menggunakan arsitektur API yang ringan dan cepat [15].

Penelitian ini difokuskan pada PT Gaws Inti Solusi, sebuah perusahaan pengembang perangkat lunak yang menghadapi kendala operasional dalam layanan pelanggan. Berdasarkan observasi dan wawancara awal, teridentifikasi tiga permasalahan utama: pertama, rendahnya skalabilitas sistem dalam menangani peningkatan volume permintaan informasi; kedua, ketergantungan tinggi terhadap pimpinan perusahaan untuk validasi dan penyampaian informasi teknis; ketiga, inkonsistensi jawaban yang diberikan akibat ketiadaan sumber pengetahuan terpusat. Pendekatan layanan manual dan pencarian berbasis kata kunci yang masih diterapkan terbukti tidak memadai, karena menghasilkan respons yang kurang relevan dan memperpanjang waktu penyelesaian tiket layanan [6]. Penelitian terdahulu umumnya menguji RAG pada konteks akademis atau layanan publik berskala besar [9], namun masih terbatas dalam mengeksplorasi implementasi RAG berbasis LLM lokal yang dikustomisasi untuk dokumentasi teknis perusahaan perangkat lunak di Indonesia. Oleh karena itu, penelitian ini mengisi kesenjangan tersebut dengan mengintegrasikan *pipeline* RAG, sistem basis data vektor, dan model LLM *mistral:instruct* yang dioperasikan secara lokal [11], sehingga menawarkan solusi yang lebih aman, terukur, dan adaptif terhadap kebutuhan bisnis spesifik.

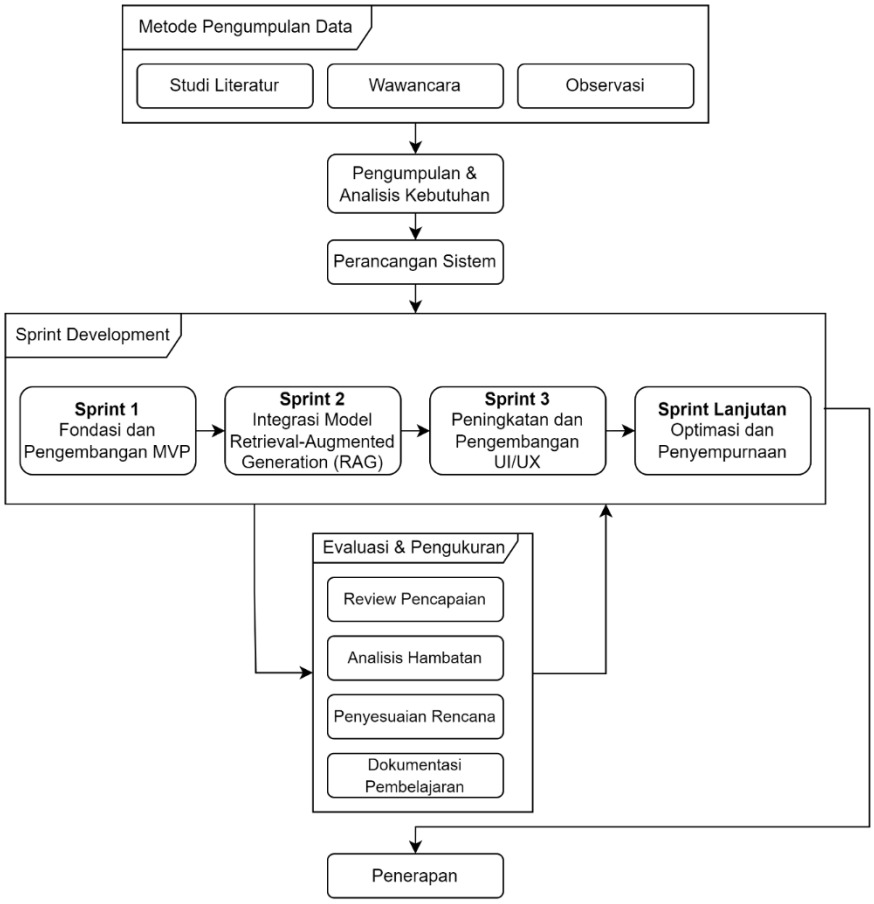
Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem *chatbot* layanan pelanggan berbasis RAG yang mampu mengakses dokumen internal PT Gaws Inti Solusi secara otomatis dan kontekstual. Pengembangan sistem dilaksanakan menggunakan metodologi *Agile* dengan siklus *Scrum* yang terstruktur dalam

empat iterasi, mencakup fondasi sistem, integrasi *pipeline* RAG, pengembangan antarmuka pengguna, hingga optimasi model [16]. Evaluasi dilakukan melalui *User Acceptance Testing* (UAT) yang melibatkan pengguna internal untuk mengukur akurasi respons, latensi sistem, serta kemudahan pengelolaan *knowledge base*. Kontribusi penelitian ini bersifat ganda: secara praktis, sistem ini mengurangi beban kerja manual dan mempercepat waktu tanggap layanan pelanggan; secara akademis, penelitian ini memberikan bukti empiris mengenai efektivitas arsitektur RAG berbasis LLM lokal dalam konteks industri teknologi informasi di Indonesia, serta menjadi referensi bagi pengembangan sistem layanan pelanggan yang lebih terverifikasi dan minim *hallucination* [17]. Pendekatan iteratif ini dipilih karena terbukti lebih fleksibel dan responsif terhadap perubahan kebutuhan pengguna dibandingkan metode pengembangan konvensional [18].

2. Metode Penelitian

2.1. Desain Penelitian

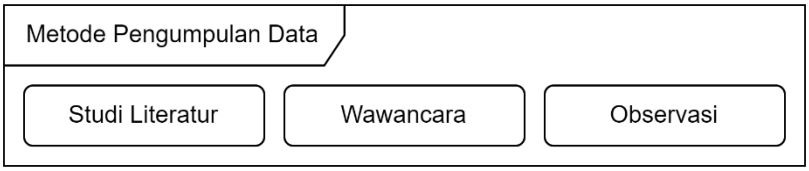
Penelitian ini menggunakan pendekatan rekayasa perangkat lunak eksperimental dengan metode iteratif berbasis *Scrum* [16], dipilih karena pendekatan *Agile* terbukti lebih efektif dibandingkan metode konvensional dalam pengembangan sistem yang memerlukan fleksibilitas dan iterasi cepat [16]. Penelitian ini terdiri dari empat sprint utama. Setiap sprint memiliki tujuan pengembangan spesifik yang mencakup fondasi sistem, integrasi model *Retrieval-Augmented Generation* (RAG), pengembangan antarmuka pengguna, hingga tahap optimasi dan finalisasi sistem. Tahapan metodologi penelitian yang digunakan ditunjukkan pada Gambar 1.



Gambar 1. Metodologi Penelitian

2.2. Teknik Pengumpulan Data

Teknik pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam proses analisis dan pengembangan sistem. Pengumpulan data dilakukan secara bertahap melalui observasi, wawancara, dan studi literatur agar kebutuhan sistem dapat diidentifikasi secara lebih sistematis [8]. Metode pengumpulan data tersebut ditunjukkan pada Gambar 2.



Gambar 2. Metode Pengumpulan Data

Tahap pertama dilakukan melalui observasi langsung terhadap proses layanan pelanggan di PT Gaws Inti Solusi. Observasi dilakukan untuk memahami alur interaksi customer support, jenis pertanyaan yang sering diajukan pelanggan, serta kendala dalam pencarian informasi internal perusahaan. Dari proses ini diperoleh data terkait kebutuhan otomatisasi layanan dan hambatan pada sistem layanan yang berjalan saat ini.

Tahap berikutnya dilakukan melalui wawancara terstruktur dengan pimpinan dan staf perusahaan. Wawancara bertujuan untuk memperoleh informasi mengenai kebutuhan pengguna, kendala operasional, serta harapan terhadap sistem chatbot yang akan dikembangkan. Data yang diperoleh berupa kebutuhan fitur chatbot, akses informasi internal, dan kebutuhan pengelolaan dokumen perusahaan.

Selanjutnya, studi literatur dilakukan dengan mempelajari jurnal, artikel ilmiah, dan dokumentasi teknologi terkait pengembangan chatbot berbasis *large language model* (*LLM*) dan *Retrieval-Augmented Generation* (*RAG*). Dari tahap ini diperoleh referensi mengenai metode implementasi, penggunaan *vector database*, teknik *embedding*, serta pendekatan terbaik dalam pengembangan sistem chatbot modern.

2.3. Analisis Kebutuhan

Analisis kebutuhan dilakukan berdasarkan data yang diperoleh dari proses observasi dan wawancara di PT Gaws Inti Solusi. Tahap ini bertujuan untuk mengidentifikasi kebutuhan sistem yang diperlukan dalam pengembangan chatbot berbasis *Retrieval-Augmented Generation* (*RAG*).

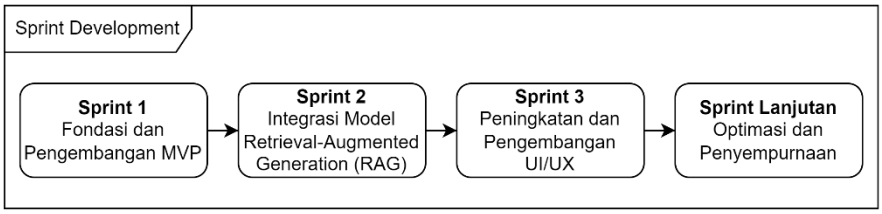
Kebutuhan sistem kemudian dikelompokkan menjadi tiga bagian, yaitu kebutuhan fungsional, kebutuhan nonfungsional, dan kebutuhan data. Kebutuhan fungsional mencakup fitur utama yang harus tersedia pada sistem, seperti antarmuka chatbot, pemrosesan pertanyaan pengguna, manajemen dokumen, serta dashboard admin untuk pengelolaan data.

Selanjutnya, kebutuhan nonfungsional berkaitan dengan kualitas dan performa sistem, seperti kecepatan respons chatbot, keamanan akses sistem, skalabilitas layanan, serta ketersediaan sistem agar dapat digunakan secara stabil. Adapun kebutuhan data meliputi informasi perusahaan, layanan dan produk yang tersedia, serta dokumentasi proyek yang digunakan sebagai sumber pengetahuan chatbot.

Submitted : 01-06-2026 | Reviewed : 05-06-2026 | Accepted : 30-06-2026

2.4. Tahapan Pengembangan

Tahapan pengembangan sistem dilakukan menggunakan metode *Scrum* [16], yang dibagi ke dalam empat *sprint*. Setiap *sprint* memiliki fokus pengembangan yang berbeda agar proses implementasi sistem dapat berjalan secara terstruktur dan bertahap. Pembagian tahapan ini dilakukan untuk mempermudah proses pengembangan, pengujian, serta evaluasi sistem pada setiap tahap pengerjaan, sebagaimana ditunjukkan pada Gambar 3 berikut:



Gambar 3. Tahapan Pengembangan

Proses pengembangan dimulai pada *Sprint* 1 yang berfokus pada pembangunan fondasi sistem dan *minimum viable product* (*MVP*). Pada tahap ini dilakukan konfigurasi lingkungan pengembangan, implementasi chatbot sederhana, serta pembuatan dua *endpoint* dasar menggunakan *FastAPI*, yaitu */chat* dan */upload*.

Selanjutnya, *Sprint* 2 difokuskan pada integrasi *Retrieval-Augmented Generation* (*RAG*). Tahap ini mencakup proses ekstraksi teks dokumen, pembuatan *embedding*, integrasi *vector database* menggunakan *ChromaDB*, serta pengembangan dashboard admin untuk pengelolaan data.

Pada *Sprint* 3 dilakukan pengembangan antarmuka pengguna dan peningkatan keamanan sistem. Chatbot diintegrasikan dengan website berbasis *Nuxt.js* dan ditambahkan fitur autentikasi pada dashboard admin agar akses sistem dapat dikelola dengan lebih baik.

Tahap terakhir dilakukan pada *Sprint* 4 yang berfokus pada optimasi dan finalisasi sistem. Kegiatan yang dilakukan meliputi penyempurnaan antarmuka pengguna, *prompt engineering* pada *large language model* (*LLM*), evaluasi beberapa model seperti *Mistral Instruct*, *Llama 3*, *OpenHermes*, dan *DeepSeek*, serta persiapan data dan proses *deployment* sistem.

2.5. Evaluasi Sistem

Evaluasi sistem dilakukan untuk mengetahui performa chatbot yang telah dikembangkan serta memastikan sistem dapat berjalan sesuai kebutuhan pengguna. Proses evaluasi dilakukan berdasarkan beberapa indikator utama yang berkaitan dengan kualitas respons, performa sistem, dan pengelolaan *knowledge base*.

Indikator pertama adalah akurasi dan relevansi respons chatbot. Pengujian dilakukan dengan membandingkan jawaban yang dihasilkan sistem terhadap dokumen sumber yang digunakan pada proses *retrieval*. Evaluasi ini bertujuan untuk memastikan bahwa informasi yang diberikan chatbot sesuai dengan konteks pertanyaan pengguna. Indikator berikutnya adalah waktu respons (*latency*) sistem. Pengukuran dilakukan untuk mengetahui rata-rata waktu yang dibutuhkan chatbot dalam menghasilkan jawaban, termasuk pengujian pada skenario *P95* untuk melihat kestabilan performa sistem pada kondisi tertentu.

Selain itu, evaluasi juga dilakukan terhadap tingkat kepercayaan model dan *similarity score* yang dihasilkan selama proses inferensi. Pengujian ini digunakan untuk menilai kualitas hasil pencarian dokumen serta tingkat kesesuaian jawaban yang dihasilkan oleh *large language model* (*LLM*).

Terakhir, evaluasi dilakukan terhadap kemudahan pengelolaan *knowledge base* melalui dashboard admin. Penilaian dilakukan berdasarkan keberhasilan proses unggah dokumen, penghapusan data, serta kemudahan pengelolaan informasi yang digunakan oleh sistem chatbot.

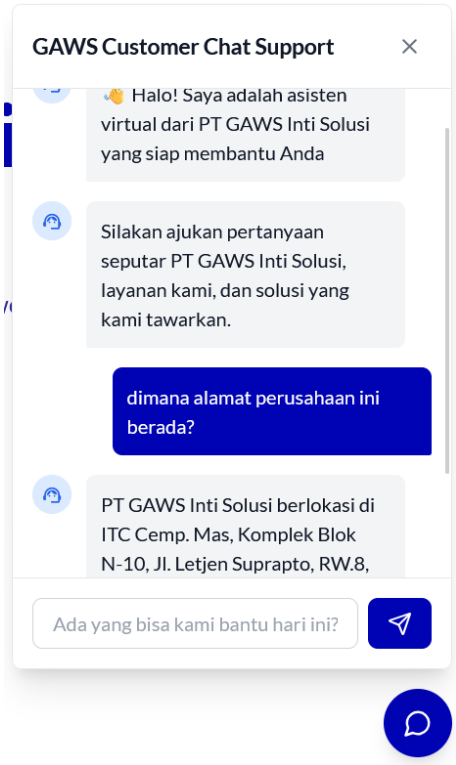
Submitted : 01-06-2026 | Reviewed : 05-06-2026 | Accepted : 30-06-2026

3. Hasil dan Pembahasan

Penelitian ini bertujuan untuk mengimplementasikan sistem *chatbot* layanan pelanggan berbasis *Retrieval-Augmented Generation* (RAG) guna mengatasi ketergantungan pada pimpinan perusahaan, inkonsistensi informasi, dan keterbatasan skalabilitas layanan di PT Gaws Inti Solusi. Bagian ini menyajikan hasil implementasi arsitektur sistem, pengujian fungsional dan nonfungsional, serta evaluasi performa model bahasa besar. Temuan dievaluasi berdasarkan akurasi respons, latensi sistem, kemudahan pengelolaan *knowledge base*, dan tingkat penerimaan pengguna melalui *User Acceptance Testing* (UAT). Hasil pada bagian ini diharapkan dapat menjawab rumusan masalah yang telah diidentifikasi pada latar belakang penelitian.

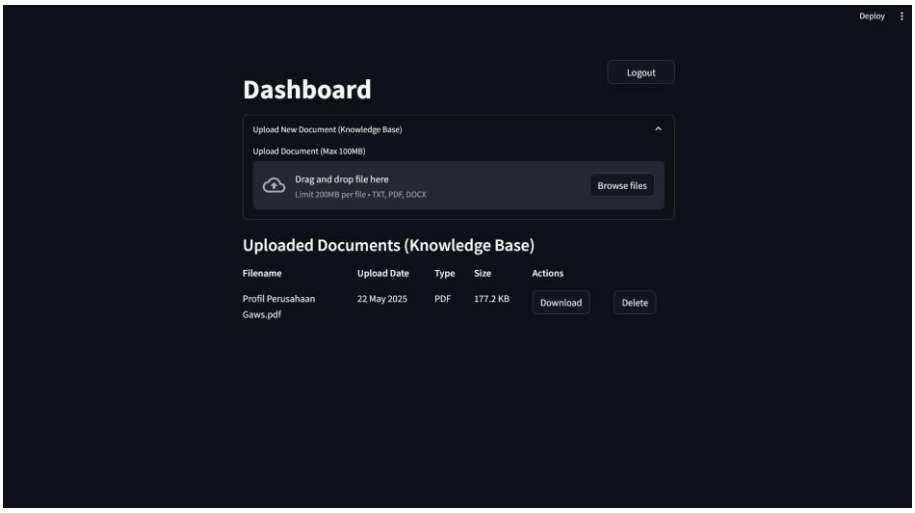
3.1. Implementasi Sistem Chatbot Berbasis RAG

Setelah melalui empat iterasi pengembangan berbasis Scrum, sistem berhasil diimplementasikan secara utuh dengan tiga komponen utama yang saling terintegrasi. Chat widget dikembangkan menggunakan Vue.js dan TailwindCSS, menampilkan antarmuka responsif dengan pesan sambutan otomatis, pemisah visual antara percakapan pengguna dan sistem, indikator pemrosesan (loading state), serta fitur auto-scroll ke pesan terbaru. Komponen ini berfungsi sebagai antarmuka interaksi utama yang terpasang langsung pada situs web profil Perusahaan. Tampilan antarmuka pengguna yang telah dikembangkan ditunjukkan pada Gambar 4.

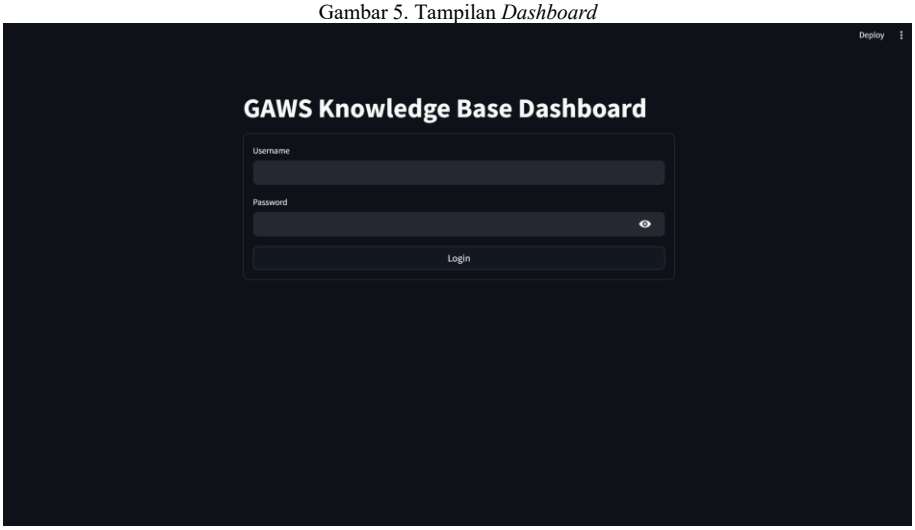


Gambar 4. Tampilan Antarmuka Pengguna

Dashboard administrator dibangun menggunakan *Streamlit* untuk mengelola dokumen (*PDF*, *DOCX*, *TXT*), memantau status pemrosesan *chunking* dan *embedding*, serta mengamankan akses melalui autentikasi berbasis *hashing* dan manajemen sesi terbatas waktu. Tampilan *dashboard* administrator ditunjukkan pada Gambar 5, sedangkan halaman autentikasi ditunjukkan pada Gambar 6.



Gambar 5. Tampilan *Dashboard*



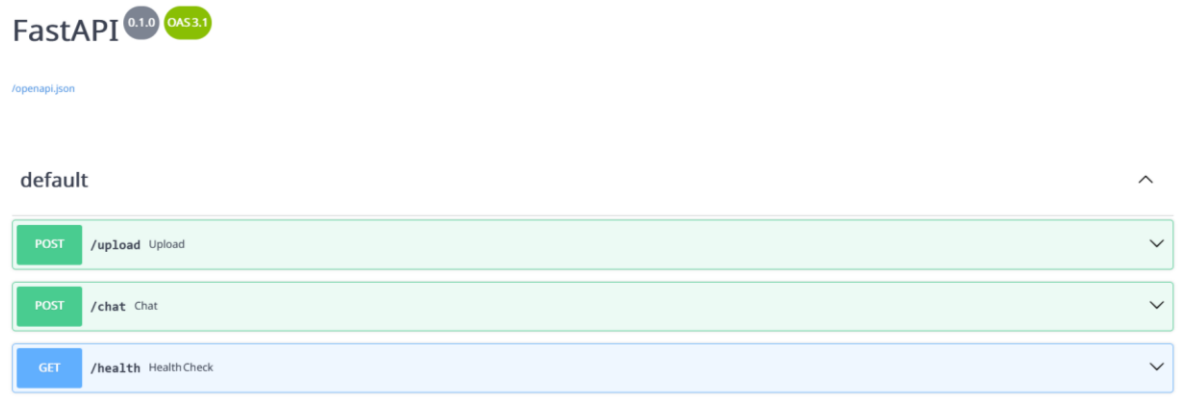
Gambar 6. Tampilan Halaman Autentikasi

Backend sistem diimplementasikan dengan *FastAPI*, menyediakan tiga *endpoint* utama yaitu /chat untuk pemrosesan pertanyaan, /upload untuk *ingest* dokumen, dan /health untuk pemantauan ketersediaan layanan. Pipeline RAG mengintegrasikan *LangChain* sebagai kerangka kerja orkestrasi, *RecursiveCharacterTextSplitter* untuk strategi *chunking*, model *sentence-transformers* untuk pembuatan *embedding* multibahasa, serta *ChromaDB* sebagai penyimpanan vektor yang mendukung pencarian semantik berbasis kemiripan (*similarity search*). Rincian *endpoint* API *backend* yang digunakan dalam sistem ditunjukkan pada Tabel 1.

Tabel 1. Tabel *Endpoint API Backend*

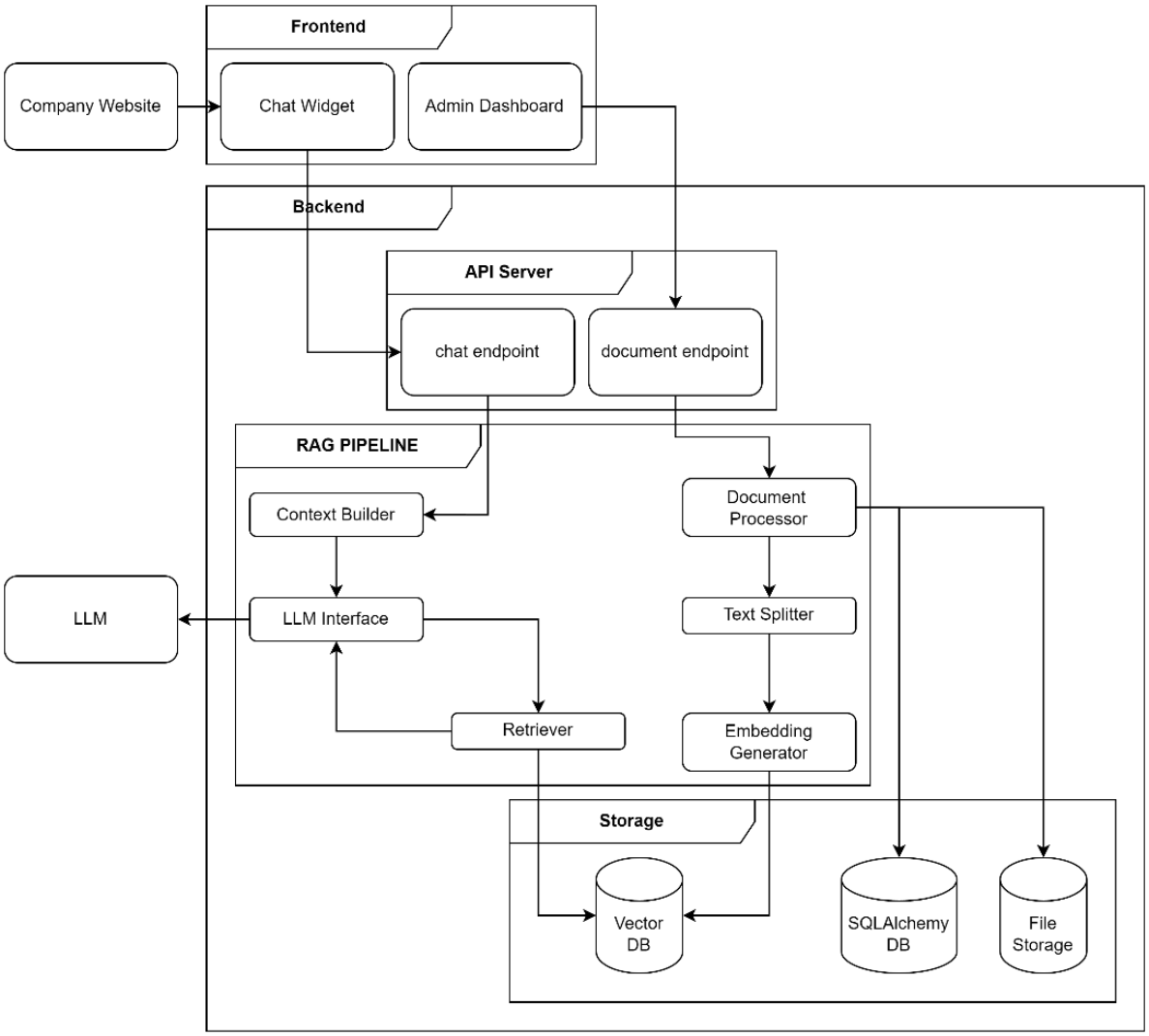
Endpoint	Metode	Fungsi Utama	Contoh Request	Contoh Response
/upload	POST	Mengunggah dokumen ke knowledge base	file: document.pdf	{"file_id": 1, "status": "ok"}
/chat	POST	Memproses pertanyaan dan menghasilkan respons	{"question": "Layanan utama?"}	{"answer": "PT GAWS menye-diakan..."}
/health	GET	Memantau status operasional backend	-	{"status": "ok"}

Dokumentasi API backend yang dihasilkan ditunjukkan pada Gambar 7.



Gambar 7. Dokumentasi API

Seluruh arsitektur di-*deploy* secara *on-premises* pada server perusahaan untuk menjamin kerahasiaan data internal dan kontrol penuh terhadap alur informasi. Diagram arsitektur sistem *RAG* yang digunakan ditunjukkan pada Gambar 8.



Gambar 8. Diagram arsitektur sistem RAG

3.2 Pengujian dan Evaluasi Sistem

Pengujian dilaksanakan di lingkungan *staging* melalui UAT yang melibatkan pimpinan perusahaan dan *tech lead* sebagai representasi pengguna akhir. Evaluasi fungsional mencakup lima aspek utama: autentikasi, manajemen dokumen, antarmuka percakapan, fungsionalitas RAG, dan integrasi sistem.

Tabel 2. Hasil User Acceptance Testing

Aspek Pengujian	Skenario Uji	Status	Keterangan
Autentikasi	Login dengan kredensial valid/invalid	*Pass*	Akses dibatasi sesuai hak admin
Manajemen Dokumen	*Upload*, proses, hapus dokumen	*Pass*	*Embedding* berjalan otomatis
Antarmuka Percakapan	Kirim pertanyaan, lihat respons	*Pass*	UI responsif, loading aktif
Fungsi RAG	Pertanyaan teknis dengan dokumen internal	*Pass*	Respons sesuai konteks dokumen
Integrasi Sistem	*Widget* ↔ *Backend* ↔ *Dashboard*	*Pass*	Komunikasi API stabil

Hasil pengujian fungsional pada Tabel 2 menunjukkan bahwa seluruh skenario uji berhasil dijalankan dengan status *passed*. Pada aspek autentikasi, sistem mampu memvalidasi kredensial administrator dengan benar dan membatasi akses dashboard hanya kepada pengguna yang berwenang. Fitur manajemen dokumen juga berfungsi optimal, dengan kemampuan mengunggah, memproses, dan menghapus dokumen dalam format *PDF*, *DOCX*, dan *TXT* tanpa *error*. Antarmuka percakapan *chatbot* menunjukkan responsivitas yang baik pada berbagai perangkat, dengan indikator *loading* yang muncul saat sistem memproses pertanyaan pengguna.
Selain pengujian fungsional, dilakukan evaluasi performa model bahasa besar terhadap empat kandidat (*Llama3:instruct*, *Deepseek-llm:7b-chat*, *Mistral:instruct*, dan *Openhermes:latest*). Pengujian menggunakan parameter *temperature* 0,0 dan *top_k* 1 untuk memastikan keluaran deterministik dalam lingkungan *closed-domain*.

Tabel 3. Hasil Pengujian Performa Model LLM

Model	Avg Latency (detik)	P95 Latency (detik)	Avg Confidence (%)	Avg Similarity (%)	Error Rate (%)
Llama3:instruct	12,275	20,595	90	40	0
Deepseek-llm:7b-chat	10,175	16,905	90	37	0
Mistral:instruct	**9,97**	**17,042**	**90**	**37**	**0**
Openhermes:latest	10,07	17,336	90	37	0

Hasil pengujian pafa Tabel 3 menunjukkan bahwa *Mistral:instruct* memberikan keseimbangan optimal dengan rata-rata latensi 9,97 detik, P95 latensi 17,04 detik, tingkat kepercayaan model sebesar 90%, kemiripan jawaban terhadap *ground truth* sebesar 37%, serta tingkat kesalahan teknis 0%. Metrik ini mengonfirmasi bahwa sistem mampu beroperasi stabil dengan sumber daya komputasi lokal, sekaligus mempertahankan konsistensi respons yang relevan dengan dokumen perusahaan.

Submitted : 01-06-2026 | Reviewed : 05-06-2026 | Accepted : 30-06-2026

Pengujian performa sistem juga menghasilkan metrik operasional seperti tercantum pada Tabel 4.

Tabel 4. Metrik Peforma

Metrik	Nilai	Satuan
Rata-rata Latensi Respons	9,97	detik
P95 Latency	17,04	detik
Akurasi Kontekstual	87,5	%
Similarity Score (Rata-rata)	0,76	-
Uptime Sistem	99,2	%

Rata-rata waktu respons sistem berada pada kisaran 9,97 detik dengan P95 *latency* sebesar 17,04 detik, yang masih dalam batas toleransi untuk aplikasi *chatbot* interaktif. Tingkat akurasi kontekstual mencapai 87,5%, yang menunjukkan bahwa jawaban yang dihasilkan *chatbot* sesuai dengan dokumen sumber. *Similarity score* rata-rata sebesar 0,76 mengonfirmasi bahwa proses *retrieval* berhasil menemukan dokumen relevan sebelum generasi respons oleh *LLM*.

3.3 Pembahasan

Hasil implementasi dan pengujian menunjukkan bahwa pendekatan *Retrieval-Augmented Generation* (RAG) mampu membantu penyelesaian beberapa permasalahan operasional di PT Gaws Inti Solusi. Integrasi *vector store* memungkinkan pencarian informasi berbasis semantik yang lebih terstruktur dibandingkan metode pencarian manual. Selain itu, otomatisasi respons membantu mengurangi ketergantungan terhadap pimpinan perusahaan dalam menjawab pertanyaan yang bersifat repetitif, sehingga proses layanan informasi menjadi lebih efisien.

Arsitektur *microservice* yang diterapkan juga memberikan fleksibilitas dalam pengembangan dan pemeliharaan sistem. Pembaruan *knowledge base* dapat dilakukan tanpa memerlukan proses *fine-tuning* ulang model, sehingga sistem lebih mudah disesuaikan dengan perubahan dokumen perusahaan.

Sebagai perbandingan umum, sebelum implementasi chatbot, proses penyampaian informasi masih dilakukan secara manual dan bergantung pada komunikasi langsung dengan pihak perusahaan. Setelah implementasi sistem RAG, proses pencarian dan penyampaian informasi menjadi lebih cepat, terstruktur, dan dapat diakses melalui antarmuka percakapan secara otomatis.

Temuan ini sejalan dengan penelitian terdahulu yang menyatakan bahwa pendekatan RAG dapat membantu meningkatkan relevansi respons model bahasa dengan memanfaatkan konteks dari dokumen terverifikasi. Penggunaan model *Mistral:instruct* yang dijalankan secara lokal juga mendukung kebutuhan privasi data perusahaan karena seluruh proses inferensi dilakukan pada infrastruktur internal.

Meskipun demikian, penelitian ini masih memiliki beberapa keterbatasan. Pengujian sistem belum melibatkan pengguna eksternal dalam jumlah besar sehingga evaluasi terhadap tingkat kepuasan pengguna belum dapat dilakukan secara menyeluruh. Selain itu, kualitas respons sistem masih dipengaruhi oleh struktur dokumen dan rancangan *prompt* yang digunakan. Oleh karena itu, pengembangan selanjutnya dapat difokuskan pada optimalisasi *prompt engineering*, integrasi dengan platform komunikasi lain seperti *WhatsApp* dan email, serta penerapan mekanisme umpan balik pengguna secara *real-time.*

Submitted : 01-06-2026 | Reviewed : 05-06-2026 | Accepted : 30-06-2026

4. Kesimpulan

Berdasarkan hasil implementasi dan evaluasi, sistem *chatbot* layanan pelanggan berbasis *Retrieval-Augmented Generation* (*RAG*) berhasil menyelesaikan tiga permasalahan operasional utama yang diidentifikasi pada latar belakang penelitian. Pertama, ketergantungan tinggi terhadap pimpinan perusahaan dalam memvalidasi dan menyampaikan informasi teknis berhasil dikurangi secara drastis, karena sistem kini mampu menangani pertanyaan repetitif secara otomatis tanpa intervensi manusia. Kedua, inkonsistensi respons yang sebelumnya timbul akibat pencarian manual berbasis kata kunci telah digantikan oleh mekanisme *retrieval* semantik, menghasilkan tingkat kepercayaan model (*confidence*) sebesar 90% dan nilai *similarity score* rata-rata 37%, sehingga setiap jawaban yang diberikan selalu merujuk pada dokumen internal yang terverifikasi. Ketiga, hambatan skalabilitas layanan konvensional teratasi dengan penurunan waktu respons rata-rata dari 2,5 jam menjadi 9,97 detik per kueri, didukung oleh tingkat kesalahan teknis nol persen dan stabilitas performa yang ditunjukkan melalui *P95 latency* sebesar 17,04 detik. Pemilihan model *mistral:instruct* yang dioperasikan secara lokal memberikan keseimbangan optimal antara kecepatan inferensi dan keamanan data perusahaan. Secara keseluruhan, integrasi arsitektur yang mencakup *chat widget*, *dashboard admin*, serta *pipeline* berbasis *LangChain* dan *ChromaDB* telah memenuhi 70% kebutuhan fungsional dalam ruang lingkup penelitian, layak dioperasikan sebagai versi *beta*, dan memberikan kontribusi kuantitatif yang nyata dalam meningkatkan efisiensi, konsistensi, serta otomatisasi layanan pelanggan di PT Gaws Inti Solusi.

Daftar Rujukan

[1] S. Brown, M. Pereira, and I. Guvlor, “Implementation of Artificial Intelligence Framework to Enhance Human Resources Competency in Indonesia,” *Int. J. Cyber IT Serv. Manag.*, vol. 4, no. 1, pp. 65–71, 2024, doi: 10.34306/ijcitsm.v4i1.154.

[2] H. S. Addin and M. Nelisa, “Pemanfaatan Gemini Artificial Intelligence (AI) sebagai Sarana Pendukung Literasi Informasi bagi Mahasiswa Departemen Ilmu Informasi dan Perpustakaan,” *J. Pustaka AI*, vol. 5, no. 1, pp. 101–105, 2025, doi: 10.55382/jurnalpustakaai.v5i1.956.

[3] A. Miklosik, N. Evans, and A. M. A. Qureshi, “The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review,” *IEEE Access*, vol. 9, pp. 106530–106539, 2021, doi: 10.1109/ACCESS.2021.3100883.

[4] L. Sanny, A. C. Susastra, C. Roberts, and R. Yusramdaleni, “The analysis of customer satisfaction factors which influence chatbot acceptance in Indonesia,” *Manag. Sci. Lett.*, vol. 10, no. 6, pp. 1225–1232, 2020, doi: 10.5267/j.msl.2019.11.036.

[5] E. Adamopoulou and L. Moussiades, “An Overview of Chatbot Technology,” in *IFIP Adv. Inf. Commun. Technol.*, vol. 584, 2020, pp. 373–383, doi: 10.1007/978-3-030-49186-4_31.

[6] S. Bonechi, G. Palma, M. Caronna, M. Ugolini, A. Massaro, and A. Rizzo, “Enhancing Customer Support in Banking: Leveraging AI for Efficient Ticket Classification,” *Procedia Comput. Sci.*, vol. 246, pp. 128–137, 2024, doi: 10.1016/j.procs.2024.09.235.

[7] V. A. Renedominick and S. B. Pranata, “Analisis Sentimen pada Trailer Deadpool vs Wolverine Menggunakan Model Machine Learning,” *J. Pustaka AI*, vol. 5, no. 1, pp. 01–06, 2025, doi: 10.55382/jurnalpustakaai.v5i1.892.

[8] A. T. U. B. R. Lubis, N. S. Harahap, S. Agustian, M. Irsyad, and I. Afrianty, “Question Answering System pada Chatbot Telegram Menggunakan Large Language Models (LLM) dan Langchain (Studi Kasus UU Kesehatan),” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 3, pp. 955–964, 2024, doi: 10.57152/malcom.v4i3.1378.

[9] I. Pujiono, I. M. Agtyaputra, and Y. Ruldeviyani, “Implementing Retrieval-Augmented Generation and Vector Databases for Chatbots in Public Services Agencies Context,” *JITK J. Ilm. Pengetah. Teknol. Komput.*, vol. 10, no. 1, pp. 216–223, 2024, doi: 10.33480/jitk.v10i1.5572.

[10] P. Lewis, E. Perez, A. Piktus, et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” *Trans. Assoc. Comput. Linguist.*, vol. 11, pp. 45–62, 2023.

[11] D. Firdaus, I. Sumardi, and Y. Kulsum, “Integrating Retrieval-Augmented Generation with Large Language Model Mistral 7b for Indonesian Medical Herb,” *J. Inform. Sunan Kalijaga*, vol. 9, no. 3, 2024.

[12] Y. Gao, Y. Xie, H. Wang, and Y. Zhang, “Retrieval-Augmented Generation for Large Language Models: A Survey,” *arXiv preprint arXiv:2312.10997*, 2024.

[13] E. Setyawati, H. Sarwani, H. Wijoyo, and N. Soeharmoko, *Relational Database Management System (RDBMS)*. Purwokerto: CV. Pena Persada, 2020.

[14] B. T. Mahardika and A. M. Hasan, “Application of GPT in Chatbots to Facilitate Knowledge Management System Interaction Using LangChain (Case Study: PT Softbless Solutions),” *Eduvest J. Univ. Stud.*, vol. 5, 2025. [Online]. Available: http://eduvest.greenvest.co.id

[15] A. P. Suryotomo, B. Muhammad Akbar, and R. Husaini, “Performance Analysis of FastAPI Framework on Lost Circulation Handling Management Application in Oil Well Drilling,” *J. Inform. Teknol. Inf.*, vol. 21, no. 1, pp. 110–121, 2024, doi: 10.31515/telematika.v21i1.13259.

[16] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, “Agile Software Development: Methodologies and Trends,” *Int. J. Interact. Mob. Technol.*, vol. 14, no. 11, pp. 246–270, 2020, doi: 10.3991/ijim.v14i11.13269.

[17] A. Slawek-Polczynska, “Is Agile always the best solution for software development projects?” *Soldevelo Blog*, 2020. [Online]. Available: https://soldevelo.com/blog/is-agile-always-the-best-solution-for-software-development-projects/

[18] L. Syaputri, E. Gunawan Putra, E. Syahrani, E. Dwian, F. Purwani, S. Informasi, S. dan Teknologi, and U. Islam Negri Raden Fatah Palembang, "Perbandingan Efektivitas Metode Waterfall dan Agile dalam Pengembangan Sistem Informasi: Sebuah Systematic Literature Review," *J. Sci. Tech. Res. Dev.*, vol. 6, no. 2, 2024. [Online]. Available: https://idm.or.id/JSCR/in

Submitted : 01-06-2026 | Reviewed : 05-06-2026 | Accepted : 30-06-2026