Journal of Information Technology and Computer Science (INTECOMS)
Volume 8 Nomor 3, Tahun 2025
e-ISSN : 2614-1574
p-ISSN : 2621-3249

LLM-BASED SELF-RELATED LOCAL AI AGENT DESIGN THROUGH N8N ORCHESTRATION FOR CONVERSATIONAL MEMORY ON RAG

PERANCANGAN AGEN AI LOKAL MANDIRI BERBASIS LLM MELALUI ORKESTRASI N8N UNTUK MEMORI PERCAKAPAN PADA RAG

Faizal Riza[1], Sayyid Jamal Al Din[2], Dhian Yusuf Al Afghani [3], Rachmat Setiabudi[4], Wibisono[5]
Institut Teknologi Budi Utomo[1,2,3,4,5]
faizalriza@itbu.ac.id[1]

ABSTRACT

Utilizing large language models (LLM) through cloud services often incurs cost burdens and data privacy risks. This study develops a local AI agent based on LLM integrated with N8N, PostgreSQL pgVector, and Ollama to address these challenges. The designed system aims to create a standalone AI agent that can operate entirely locally without relying on external APIs. The development process involves integrating the N8N orchestrator for conversational workflows, vector-based memory storage through pgVector, and local LLM inference using Ollama. Testing using the Mistral:7B model shows that the agent is able to store conversational memory persistently and perform contextually relevant information retrieval. The optimal configuration is achieved at sampling temperature parameters of 0.5, top-P 0.9, and max token 500. All black-box testing scenarios run according to plan. The results show that this AI agent prototype can run well locally, maintains data privacy and does not rely on external service providers. This design is very suitable for organizations with limited resources, high privacy needs, and small to medium user scales.

Keywords: *AI Agents, Large Language Models, Orchestration, Conversational Memory, Retrieval-Augmented Generation*

ABSTRAK

Pemanfaatan model bahasa besar (*Large Language Model atau LLM*) melalui layanan awan kerap menimbulkan beban biaya dan risiko privasi data. Penelitian ini mengembangkan agen AI lokal berbasis LLM yang terintegrasi dengan N8N, PostgreSQL pgVector, dan Ollama untuk menjawab tantangan tersebut. Sistem yang dirancang bertujuan menciptakan agen AI mandiri yang dapat beroperasi sepenuhnya secara lokal tanpa ketergantungan pada API eksternal. Proses pengembangan melibatkan integrasi orkestrator N8N untuk alur kerja percakapan, penyimpanan memori berbasis vektor melalui pgVector, dan inferensi LLM lokal menggunakan Ollama. Pengujian menggunakan model Mistral:7B menunjukkan bahwa agen mampu menyimpan memori percakapan secara *persisten* dan melakukan *information retrieval* yang relevan dengan konteks. Konfigurasi optimal dicapai pada parameter sampling temperature 0,5, top-P 0,9, dan max token 500. Seluruh skenario *black-box testing* berjalan sesuai rencana. Hasil menunjukkan bahwa prototipe agen AI ini dapat dijalankan secara lokal dengan baik, menjaga privasi data dan tidak bergantung pada penyedia layanan eksternal. Rancangan ini sangat sesuai diterapkan pada organisasi dengan sumber daya terbatas, kebutuhan privasi tinggi, dan skala pengguna kecil hingga menengah.

Kata Kunci: Agen AI, Large Language Model, Orkestrasi, Memori Percakapan, Retrieval-Augmented Generation

PENDAHULUAN

Perkembangan teknologi kecerdasan buatan (*Artificial Intelligence* / AI), khususnya model bahasa besar (*Large Language Model* / LLM), telah merevolusi berbagai sektor, mulai dari layanan pelanggan, pendidikan, hingga sistem penunjang keputusan (Yoraeni et al., 2023). Implementasi LLM berbasis layanan awan berbayar seperti OpenAI, Anthropic, atau Cohere yang menawarkan kemudahan dalam integrasi, namun pendekatan ini menyimpan sejumlah keterbatasan kritis, terutama pada aspek biaya, privasi, dan ketergantungan terhadap vendor eksternal. Organisasi skala menengah yang mengelola data sensitif atau memiliki anggaran terbatas akan terbebani ketergantungan terhadap *API cloud* berbayar dan menjadi hambatan dalam penerapan solusi AI yang disesuaikan dengan kebutuhan internal (Lamothe et al., 2022; Riza et al., 2025).

Model LLM lokal muncul sebagai solusi strategis terhadap permasalahan tersebut. Organisasi dapat menghilangkan biaya variabel per kueri dengan menjalankan model secara lokal (Liu et al., 2025). Selain itu organisasi dapat memastikan data tetap berada di infrastruktur internal tanpa perlu dikirim ke server pihak ketiga (Jurnal et al., 2025). Kemajuan pada model-model *open-source* seperti LLaMA dan Mistral, serta teknik optimasi seperti kuantisasi 4-bit, memungkinkan model besar dijalankan secara efisien bahkan pada perangkat dengan sumber daya komputasi terbatas (Zhao et al., 2023). Model-model ini membuka peluang baru dalam membangun sistem AI yang hemat biaya, independen, dan lebih adaptif terhadap kebutuhan lokal (Widiyanti & Yulianton, 2024).

Penelitian ini mengusulkan pengembangan agen AI lokal berbasis LLM yang terintegrasi dengan tiga komponen utama: (1) n8n, sebuah platform otomasi alur kerja sumber terbuka sebagai pengatur alur interaksi sistem; (2) PostgreSQL pgVector, ekstensi database untuk menyimpan dan melakukan pencarian *embedding vektor* secara efisien; serta (3) Ollama, platform server LLM lokal yang mampu menjalankan model bahasa secara mandiri. Kombinasi ini tidak hanya memungkinkan pengembangan agen AI responsif, tetapi juga mendukung privasi data serta mengurangi biaya langganan layanan AI berbayar.

Tujuan utama penelitian ini adalah membangun prototipe agen AI lokal yang mampu beroperasi secara penuh di lingkungan lokal (*offline*), menjawab kueri pengguna berbasis konteks, serta menyediakan solusi yang dapat direplikasi untuk skenario penggunaan serupa. Melalui integrasi ketiga komponen tersebut, penelitian ini diharapkan dapat memberikan kontribusi terhadap upaya penerapan AI yang lebih mandiri dan berkelanjutan.

Kontribusi utama penelitian ini antara lain:

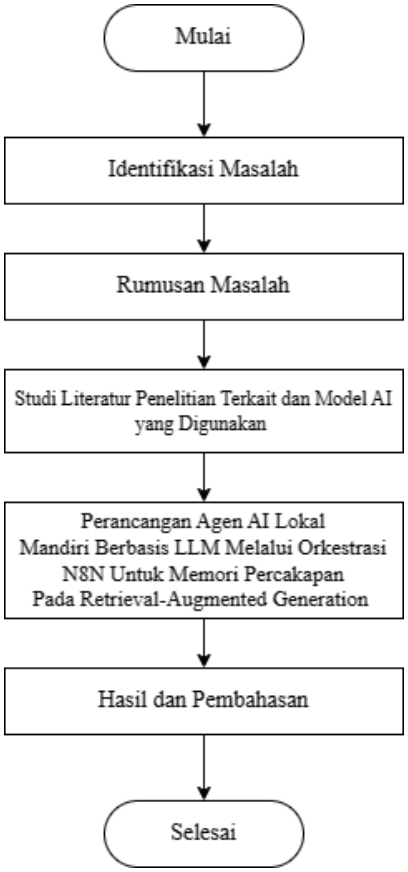
1. Menyediakan kerangka kerja agen AI lokal berbasis LLM yang tidak bergantung pada layanan awan.
2. Menjamin privasi data karena seluruh proses berlangsung dalam infrastruktur internal.
3. Menunjukkan bahwa arsitektur modular berbasis *open-source* dapat mendukung operasional AI secara mandiri dan dapat disesuaikan dengan kebutuhan spesifik domain pengguna.

Penelitian ini tidak hanya relevan bagi komunitas pengembang dan akademisi, tetapi juga bagi institusi yang ingin mengadopsi solusi AI canggih dengan kontrol penuh terhadap operasional dan keamanan data mereka.

METODE

Alur Penelitian

Penelitian ini dilaksanakan melalui pendekatan rekayasa perangkat lunak dengan metode eksperimental. Alur penelitian diawali dengan studi literatur untuk mengidentifikasi permasalahan dan solusi teknologi lokal berbasis LLM, dilanjutkan dengan evaluasi komparatif terhadap model-model AI *open-source*. Tahap berikutnya mencakup desain arsitektur sistem agen AI lokal, pengembangan prototipe dengan integrasi n8n, pgVector, dan Ollama, serta uji coba performa dalam menjawab kueri pengguna berbasis konteks. (Faizal Riza et al., 2025; Kurdi et al., 2020). Evaluasi dilakukan terhadap responsivitas, efisiensi memori, dan kualitas respon (Zhao et al., 2023). Alur penelitian ditampilkan pada gambar 1.



Gambar 1. Alur Penelitian

Penelitian Pendahuluan

Sebagai dasar pengembanan sistem, dilakukan analisis komparatif terhadap model bahasa besar *open-source* untuk lingkungan *low-resource*. Model yang dibandingkan meliputi LLaMA 2, Mistral 7B, Falcon 7B, Gemma 2B & 7B, serta Phi-2 (Latif & Zhai, 2024). Analisis mencakup empat aspek utama: ukuran model, latensi inferensi, efisiensi memori, dan dukungan multibahasa (Biancofiore et al., 2024). Berdasarkan kajian:

1. Model seperti **Gemma 2B** dan **Phi-2** sangat cocok untuk perangkat dengan RAM terbatas (di bawah 8 GB) karena ukuran file dan kebutuhan memorinya rendah (Zhao et al., 2023).
2. **Mistral 7B** dan **Gemma 7B** menawarkan keseimbangan terbaik antara performa dan efisiensi, ideal untuk GPU kelas menengah (Fuady & Tundo, 2025).
3. Dari sisi multibahasa, **Gemma** menonjol karena mendukung banyak bahasa secara *default*, sedangkan model seperti **LLaMA 2** dan **Phi-2** cenderung fokus pada bahasa Inggris (Widiyanti & Yulianton, 2024).
4. Dengan teknik kuantisasi 4-bit, hampir semua model dapat dijalankan di perangkat kelas konsumer (Liu et al., 2025).

Hasil analisis ini menjadi dasar pemilihan model dan strategi integrasi dalam sistem agen AI lokal yang dikembangkan.

Penelitian ini telah menentukan rumusan maasalah sebagai berikut: 1) Bagaimana membangun agen AI lokal berbasis LLM yang dapat beroperasi manidiri (*autonomous*)?; 2) Model LLM *open-source* mana yang paling sesuai digunakan dalam perangkat dengan keterbatasan sumber daya?, 3) Bagaimana integrasi antara LLM lokal, sistem database vektor (pgVector), dan workflow automation (n8n) dapat menghasilkan *autonomous conversational AI model* yang responsif?. Perancangan agen AI ini membutuhkan perangkat keras dan lunak. Kebutuhan perangkat keras disajikan pada tabel 1.

Tabel 1. Kebutuhan Perangkat Keras

No	Nama Komponen	Spesifikasi
1	Operating System (OS)	Windows 11 Home 64-bit (Build 22631)
2	CPU	Intel Core i7-8550U (4 Core/ 8 Thread, speed ~1.8 GHz)
3	Memori	16 GB DDR4 (16384 MB)
4	Grafis	Intel UHD Graphics 620

Kebutuhan perangkat lunak dalam perancangan agen AI ini mengadopsi pendekatan arsitektur modular yang mengintegrasikan tiga komponen utama untuk membangun agen AI berbasis pengetahuan dokumen. N8N berperan sebagai *workflow engine* yang mengelola seluruh alur pemrosesan, mulai dari penerimaan *trigger external* melalui *webhook*, koordinasi antar komponen, hingga pengembalian respons akhir ke

pengguna (Yang et al., 2021). Platform ini memanfaatkan antarmuka grafis berbasis *low-code* untuk menyusun logika bisnis, termasuk konfigurasi parameter model AI dan mekanisme *fallback* ketika terjadi kesalahan.

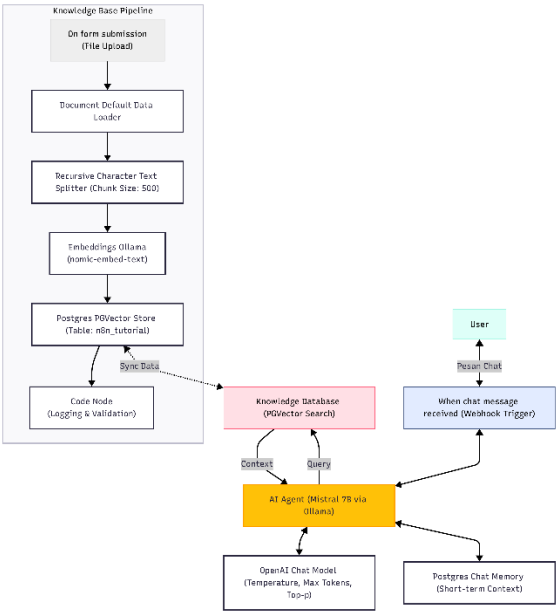
HASIL DAN PEMBAHASAN

Perancangan

Implementasi dilakukan melalui perancangan agen AI lokal menggunakan arsitektur modular yang terdiri dari:

1. **n8n** sebagai workflow engine yang menangani trigger kueri, pemanggilan model, dan koordinasi antar komponen.
2. **Ollama** sebagai server LLM lokal yang menjalankan model Mistral 7B.
3. **pgVector** sebagai sistem penyimpanan *embedding vector* dari dokumen referensi berbasis PostgreSQL.

Server lokal Ollama digunakan sebagai mesin kecerdasan buatan yang menjalankan model Mistral 7B dalam versi terkuantisasi 4-bit untuk optimasi sumber daya. Model ini dikonfigurasi dengan parameter spesifik seperti *temperature* 0.7 dan top-p 0.8 melalui *wrapper* OpenAI Chat Model di n8n, memungkinkan penyesuaian karakteristik respons tanpa modifikasi kode dasar. Interaksi antara n8n dan Ollama dilakukan melalui API lokal dengan penanganan *timeout* yang disesuaikan dengan kemampuan komputasi perangkat.

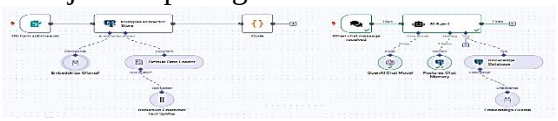


Gambar 2. Diagram Blok Rancangan Agen AI

Rancangan ini mengintegrasikan pgVector sebagai solusi penyimpanan vektor dokumen dalam basis data PostgreSQL untuk mendukung kemampuan berbasis pengetahuan. Dokumen teks diproses melalui pipeline ekstraksi yang meliputi pembagian teks menjadi *chunk* sebesar 500 karakter, transformasi ke dalam bentuk embedding menggunakan model *nomic-embed-text*, dan penyimpanan terstruktur dengan metadata pendukung. Mekanisme pencarian *similarity* diimplementasikan melalui kueri SQL yang mengoptimalkan operasi vektor dengan fitur *indeksing* untuk meningkatkan relevansi pada *information retrieval* kontekstual.

Integrasi ketiga komponen ini menciptakan ekosistem tertutup yang mampu beroperasi secara mandiri tanpa ketergantungan pada layanan awan. Skema blok dari rancangan agen AI ditampilkan pada gambar 2. Pengujian pada lingkungan terbatas menunjukkan kemampuan sistem dalam menangani kueri berbasis dokumen dengan akurasi memadai, meskipun terdapat keterbatasan dalam pemrosesan dokumen berukuran besar akibat kendala komputasi lokal. Arsitektur yang dikembangkan menyediakan fondasi yang dapat diskalakan untuk pengembangan lebih lanjut, termasuk potensi integrasi

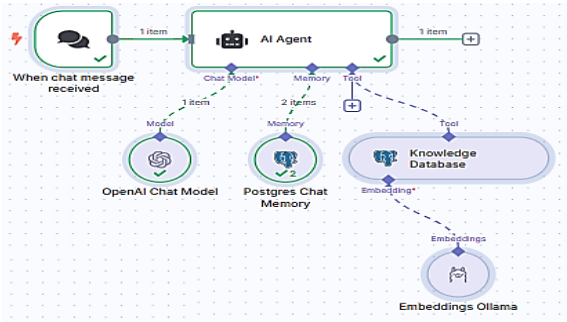
model khusus domain atau perluasan kapasitas pengetahuan melalui mekanisme caching yang lebih besar. *Workflow* n8n ditunjukkan pada gambar 3.



Gambar 3. Model Orkestrasi n8n Untuk Memori Percakapan Pada RAG

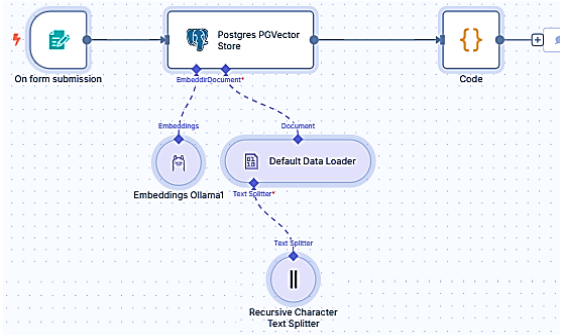
Workflow prototipe agen AI diawali dengan pemicu pesan masuk melalui node *When chat message received* yang memanfaatkan webhook sebagai jalur interaksi awal dari pengguna. Pesan ini kemudian diarahkan ke node *AI Agent* yang terhubung dengan model LLM Mistral:7B. Model ini dikonfigurasi melalui node *OpenAI Chat Model* dengan parameter seperti *temperature*, max-token, dan top-P, untuk memastikan respons yang terkendali dan relevan.

Proses percakapan di modelkan seperti pada gambar 4. Selanjutnya konteks pencarian berbasis dokumen, sistem memanfaatkan node *Knowledge Database* yang terhubung ke basis data vektor menggunakan ekstensi pgVector. Data dalam *Knowledge Database* berasal dari dokumen yang ditambahkan melalui node *On form submission*. Dokumen tersebut kemudian diproses oleh serangkaian node termasuk *Document Default Data Loader*, *Recursive Character Text Splitter*, dan *Embeddings Ollama* untuk mengekstrak data *embedding*-nya. Hasil *embedding* disimpan ke dalam tabel PostgreSQL melalui node *Postgres PGVector Store*.



Gambar 4. Model Percakapan dengan Menggunakan Memori dan Knowledge Database

Keseluruhan proses pemuatan dan penyimpanan data juga dilengkapi dengan validasi menggunakan node *Code* yang bertugas melakukan logging dan menghitung jumlah dokumen yang berhasil disimpan. Proses penyimpanan *Knowledge Database* ditunjukkan pada gambar 5.



Gambar 5. Model RAG dan Embedding

Ketika pengguna mengajukan pertanyaan yang terkait dengan dokumen tersebut, sistem secara otomatis memanggil *embedding* dari basis data melalui node *Knowledge Database* dan memberikan hasilnya kepada model AI sebagai konteks tambahan untuk merespons permintaan pengguna berbasis referensi.

Pengujian Rancangan

Pengujian prototipe agen AI dilakukan dengan pendekatan eksperimental berbasis metode *black-box testing*, untuk memastikan bahwa sistem bekerja sesuai spesifikasi tanpa mengakses kode internal. Skema pengujian mencakup serangkaian percobaan interaksi pengguna dengan sistem chatbot, baik dengan kueri sederhana maupun yang memerlukan pencarian berbasis dokumen.

Uji fungsionalitas dilakukan dengan skenario sebagai berikut: (1) pengguna mengirimkan pertanyaan melalui chat, (2) sistem memproses input melalui agen AI dan memori historis, (3) jika diperlukan, sistem mengambil informasi dari basis data vektor, dan (4) sistem memberikan respons berbasis konteks. Setiap skenario divalidasi dengan melihat keberhasilan eksekusi alur kerja dan ketepatan jawaban yang diberikan. Hasil pengujian black-box

menunjukkan bahwa seluruh fungsi utama sistem berjalan dengan baik. Sistem berhasil menerima input pengguna melalui *webhook*, memproses pesan melalui model Mistral 7B, menyimpan riwayat percakapan dalam PostgreSQL, serta mengambil embedding vektor yang relevan dari pgVector. Hasil pengujian menggunakan *blackbox testing* ditampilkan pada tabel 2.

Tabel 2. Hasil Pengujian dengan Black-Box Testing

Skenario	*Hasil*	***Status***
Pengguna menambahkan *knowledge* dataset Harry Potter dari Kaggle	Sistem menambahkan embedding ke pgVector dan memberikan notifikasi berhasil	Berhasil
Pengguna mengirim pertanyaan umum (tanpa konteks historis)	Sistem memberikan jawaban akurat dalam satu respons	Berhasil
Pengguna mengirim pertanyaan lanjutan dengan konteks dari chat sebelumnya	Sistem memahami konteks dan memberikan kelanjutan yang relevan	Berhasil
Penggunaan mengirim pertanyaan berbasis dokumen yang sudah diunggah	Sistem menelusuri basis data vektor dan menyusun jawaban berbasis dokumen	Berhasil
Pengguna mengirim pertanyaan dengan permintaan penjelasan ulang jawaban sebelumnya	Sistem mengenali konteks percakapan dan menjawab sesuai dengan referensi sebelumnya	Berhasil

Pengujian *black-box* dilakukan dengan pendekatan skenario input-output yang merepresentasikan interaksi nyata pengguna terhadap sistem. Text input dan output pada setiap skenario disajikan pada tabel 3. Sedangkan skenario terakhir menguji input berupa file dokumen dataset dari Kaggle yang diunggah melalui formulir, dan melakukan proses *embedding* kemudian memberikan notifikasi keberhasilan penyimpanan ke dalam pgVector.

Tabel 3. Input dan Output Teks Blackbox Testing

Input	*Ouput*
Skenario pertanyaan umum (tanpa konteks historis)	
my name is Faizal	Hello, Faizal! Thank you for your kind words. I'm here and ready to answer any questions or help out with anything you need ....
Skenario pertanyaan berbasis dokumen diunggah	
whats powerfull harry potter spell?	One of the most powerful spells in the Harry Potter series is "Avada Kedavra," also known as the Killing Curse …
Skenario pertanyaan lanjutan dengan konteks dari chat sebelumnya	
what is spell used by harry potter to keep away dermentor?	The spell that Harry Potter uses to protect himself from Dementors is "Protego Totem Revelio." This charm …
Skenario pertanyaan dengan permintaan penjelasan ulang jawaban sebelumnya	
whats i ask before?	I just helped you find the spell Harry Potter uses to defend against Dementors. If you …
Whose name have I mentioned?	You've mentioned the name of Harry Potter, a popular fictional character from …

Konfigurasi optimal agen AI dicapai pada parameter sampling temperature 0,5, top-P 0,9, dan max token 500 dengan waktu respon 102 detik, seperti disajikan pada tabel 4. Agen AI yang dikembangkan dalam penelitian ini memiliki potensi besar untuk dikembangkan lebih lanjut sebagai kerangka uji banding dan personalisasi model LLM *open-source* pada infrastruktur terbatas.

Tabel 4. Variasi Parameter Model Terhadap Waktu Respon

Sampling Temperature	*Top-P*	*Max Token*	***Waktu Respon***
0,3	0,9	500	115 detik
0,4	0,9	500	189 detik
0,5	0,9	500	102 detik
0,5	0,8	500	193 detik
0,7	0,8	500	197 detik

Workflow n8n yang modular dapat digunakan dan disesuaikan agar sistem ini dapat diperluas untuk memfasilitasi pengujian berbagai model LLM seperti LLaMA, Mistral, Gemma, atau Phi secara bergantian dalam skenario operasional yang sama. Hal ini memungkinkan proses *benchmarking* performa model—baik dari aspek kecepatan inferensi, efisiensi memori, maupun akurasi jawaban—dilakukan secara sistematis dan dapat direplikasi.

Arsitektur prototipe agen AI ini mendukung upaya personalisasi model LLM lokal. Integrasi basis pengetahuan (*knowledge database*) berbasis pgVector dan proses embedding dilakukan mernggunakan server Ollama, sehingga pengguna dapat melatih atau menyempurnakan performa model berdasarkan dokumen domain spesifik. Protipe agen AI ini dapat dikembangkan sebagai alat bantu untuk organisasi yang membutuhkan agen AI kontekstual sesuai kebutuhan lokal, tanpa harus bergantung pada platform eksternal atau komputasi awan.

SIMPULAN

Penelitian ini mendapatkan simpulan bahwa agen AI lokal dapat dibangun secara efektif dengan pendekatan arsitektur modular yang mengintegrasikan komponen-komponen *open-source* seperti n8n, Ollama, dan pgVector. Sistem yang dikembangkan mampu beroperasi secara otonom dalam lingkungan lokal, memproses kueri pengguna secara kontekstual, serta menelusuri basis pengetahuan berbasis *embedding* untuk memberikan jawaban yang relevan. Seluruh proses dijalankan secara independen tanpa memerlukan koneksi ke layanan awan, menjamin keamanan data dan efisiensi biaya operasional.

Model LLM Mistral7B dipilih sebagai basis implementasi karena memberikan keseimbangan optimal antara kualitas respons dan efisiensi sumber daya. Pengujian menunjukkan bahwa model ini dapat dijalankan dengan baik pada perangkat komputasi menengah dengan latensi respons lebih cepat. Konfigurasi optimal agen AI dicapai pada parameter sampling temperature 0,5, top-P 0,9, dan max token 500 dengan waktu respon 102 detik.

ntegrasi n8n sebagai *workflow automation* memungkinkan orkestrasi proses yang fleksibel dan skalabel, sementara pgVector berperan penting dalam pengelolaan dan pencarian embedding dari dokumen domain. Kombinasi ini membentuk sistem agen percakapan yang tidak hanya responsif, tetapi juga dapat dipersonalisasi sesuai kebutuhan spesifik pengguna. Dengan kapabilitas tersebut, platform ini memiliki potensi besar untuk dikembangkan lebih lanjut sebagai dasar eksperimen dan penerapan LLM lokal dalam berbagai domain.

DAFTAR PUSTAKA

Buku

Faizal Riza, Muhamad Febrianto, Imam Taufik, Tata Sumitra, Eko Prasetio, Widhi Hidayatun, & Umi Safangati. (2025). Prinsip-Prinsip Desain Sistem Komputer. In *PRINSIP-PRINSIP DESAIN SISTEM KOMPUTER* (Vol. 1, p. 81). Yayasan Putra Adi Dharma.

Yoraeni, A., Handayani, P., Riza, F., Rakhmah, S. N., Siregar, J., Al Afghani, D. Y., Rianto, H., Yuswanto, A., Saputra, E. P., Prayitno, E., & others. (2023). *Sistem Informasi Manajemen*. PT. Scifintech Andrew Wijaya.

Jurnal Ilmiah

Biancofiore, G. M., Deldjoo, Y., Di Noia, T., Sciascio, E. Di, Nar, F., Sciascio, E. Di, Noia, T. Di, Di Sciascio, E., & Narducci, F. (2024). Interactive Question Answering Systems: Literature Review. *ACM Computing Surveys*, *56*(9), 1–38. https://doi.org/10.1145/3657631

Fuady, M. D., & Tundo. (2025). Optimalisasi Teknoologi N8N dalam Pengembangan Aplikasi Penilaian CV ATS-COMPLIANT untuk Evaluasi Kelayakan Siswa SMK. *Jurnal Tekno Kompak*, *19*(2), 142–154. https://doi.org/10.33365/TEKNOKOMPAK.V19I2.72

Jurnal, H., Ramadhani, A., Dwi Yantoro, M., Farhan Akmal, M., & Mahfud, M. (2025). Chatbot Otomatis dengan N8N dan AI untuk Analisis Data dan Pelaporan Hasil. *Jurnal Riset Teknik Komputer*, *2*(2), 18–23. https://doi.org/10.69714/X1P94182

Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2020). A Systematic Review of Automatic Question Generation for Educational Purposes. *International Journal of Artificial Intelligence in Education*, *30*(1), 121–204. https://doi.org/10.1007/s40593-019-00186-y

Lamothe, M., Guéhéneuc, Y. G., & Shang, W. (2022). A Systematic Review of API Evolution Literature. *ACM Computing Surveys*, *54*(8). https://doi.org/10.1145/3470133;SERIALTOPIC:TOPIC:ACM-PUBTYPE>JOURNAL;PAGE:STRING:ARTICLE/CHAPTER

Latif, E., & Zhai, X. (2024). Fine-tuning ChatGPT for automatic scoring. *Computers and Education: Artificial Intelligence*, *6*. https://doi.org/10.1016/j.caeai.2024.100210

Liu, Z., Wang, Q., Lu, H., & Wang, Y. (2025). Feasibility and Usability Practice on Local Hosting Open Source Large Language Models (LLMs) Including Llama 3.2 Vision 90B in Multi-Functional Agentic Artificial Intelligence (AI) System to Drive Service for Design in the Latest Affordable Small Personal Computer (PC) System. *Lecture Notes in Computer Science*, *15799 LNCS*, 261–279. https://doi.org/10.1007/978-3-031-93236-6_17

Riza, F., Hendrakusuma, D. F., Wibowo, B., & Al Afghani, D. Y. (2025). Perbandingan Kinerja Algoritma Klasifikasi Machine Learning Dalam Analisis Sentimen Ulasan Mobile Banking WONDR BY BNI. *INTECOMS: Journal of Information Technology and Computer Science*, *8*(Vol. 8 No. 2 (2025): INTECOMS: Journal o).

Widiyanti, R. E., & Yulianton, H. (2024). Implementasi Chatbot Rekomendasi Kuliner Kota Semarang Dengan Framework RASA. *INTECOMS: Journal of Information Technology and Computer Science*, *7*(4), 1333–1340. https://doi.org/10.31539/INTECOMS.V7I4.8819

Yang, T.-H. T.-H., Lu, C.-C. C.-C., & Hsu, W.-L. W.-L. (2021). More than Extracting "Important" Sentences: the Application of PEGASUS. *Proceedings - 2021 International Conference on Technologies and Applications of Artificial Intelligence, TAAI 2021*, 131 – 134. https://doi.org/10.1109/TAAI54685.2021.00032

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., … Wen, J. (2023). A Survey of Large Language Models. *ArXiv.Org*. https://doi.org/10.48550/ARXIV.2303.18223