

Pengembangan Chatbot *Multi-Agent* Berbasis *AI Agent* untuk Optimalisasi Layanan Kemahasiswaan di UNISBANK Semarang

DOI: <http://dx.doi.org/10.35889/progresif.v22i3.3952>

Creative Commons License 4.0 (CC BY –NC)



Jovan Kirsten Anggata^{1*}, Dwi Budi Santoso²

Sistem Informasi, Universitas Stikubank, Semarang, Indonesia

*e-mail Corresponding Author: jovananggata1@gmail.com

Abstract

Conventional student services in higher education face staff workload constraints caused by the high volume of repetitive questions about academic regulations and the KIP scholarship. This study developed an Artificial Intelligence Agent-based chatbot with a Multi-Agent architecture to optimize student services at Universitas Stikubank (UNISBANK) Semarang. The system was built on the low-code n8n platform hosted on Sumopod, using Google Gemini as the language model, Supabase as the vector database, and Telegram as the interface. A Main Agent Router classified the user single intent and routed it to four specialist sub-workflows, namely General, News and Blog, Scholarship, and Regulation and KIP, which applied Retrieval-Augmented Generation (RAG). Gray Box testing showed that all flows operated as designed; the system answered academic questions based on official documents with cited sources, processed text and voice input, and automatically curated scholarship and news information. The results confirmed the effectiveness of a no-code multi-agent approach for automating campus information services.

Keywords: *AI Agent; Chatbot; Multi-Agent System; n8n; Retrieval-Augmented Generation*

Abstrak

Layanan kemahasiswaan konvensional di perguruan tinggi menghadapi kendala beban kerja staf akibat tingginya volume pertanyaan berulang mengenai peraturan akademik dan KIP Kuliah. Penelitian ini mengembangkan chatbot berbasis *Artificial Intelligence Agent* dengan arsitektur *Multi-Agent* untuk mengoptimalkan layanan kemahasiswaan di Universitas Stikubank (UNISBANK) Semarang. Sistem dibangun menggunakan platform *low-code* n8n yang di-hosting pada Sumopod, dengan Google Gemini sebagai model bahasa, Supabase sebagai basis data vektor, serta Telegram sebagai antarmuka. Sebuah *Main Agent Router* mengklasifikasikan maksud tunggal (*single intent*) pengguna dan mengarahkannya ke empat sub-*workflow* spesialis, yaitu General, Berita dan Blog, Beasiswa, serta Peraturan dan KIP yang menerapkan *Retrieval-Augmented Generation* (RAG). Pengujian *Gray Box* menunjukkan seluruh alur berjalan sesuai rancangan; sistem menjawab pertanyaan akademik berdasarkan dokumen resmi disertai sumber, memproses masukan teks dan suara, serta mengkurasi informasi beasiswa dan berita secara otomatis. Hasil membuktikan efektivitas pendekatan *multi-agent no-code* untuk otomatisasi layanan informasi kampus.

Kata Kunci: *AI Agent; Chatbot; Multi-Agent System; n8n; Retrieval-Augmented Generation*

1. Pendahuluan

Layanan kemahasiswaan berperan strategis dalam menunjang keberhasilan akademik dan non-akademik mahasiswa di perguruan tinggi. Keterbatasan jam operasional dan volume pertanyaan yang tinggi menjadi hambatan utama dalam penyediaan layanan yang responsif [1]. Penerapan kecerdasan buatan (*Artificial Intelligence*) dalam pendidikan tinggi telah terbukti meningkatkan kualitas layanan informasi melalui automasi respons dan ketersediaan layanan selama 24 jam tanpa bergantung pada jam kerja petugas [1]. Pemanfaatan *AI Agent* dalam

konteks pendidikan, yang menekankan efisiensi, kemandirian sistem, dan personalisasi layanan, menjadi arah pengembangan yang semakin relevan [2].

Kondisi tersebut secara nyata ditemukan di Universitas Stikubank (UNISBANK) Semarang. Berdasarkan hasil wawancara dengan staf layanan kemahasiswaan, petugas mengalami kesulitan dalam menangani pertanyaan berulang bervolume tinggi, khususnya mengenai peraturan akademik dan persyaratan KIP Kuliah. Penanganan seluruh pertanyaan rutin oleh satu orang staf dinilai tidak efisien dan rentan terhadap inkonsistensi respons. *Chatbot* berbasis aturan konvensional memiliki keterbatasan fundamental dalam menangani variasi bahasa dan pertanyaan kompleks yang memerlukan pemahaman konteks mendalam [3]. Oleh karena itu, diperlukan solusi berbasis kecerdasan buatan yang mampu mendistribusikan beban layanan secara otomatis, memberikan respons yang akurat, dan dapat beroperasi tanpa batasan waktu.

Sejumlah penelitian telah berupaya menyelesaikan permasalahan serupa di bidang layanan informasi akademik berbasis *chatbot*. Qalimaturrahmah dan Santoso [4] mengembangkan *chatbot* Telegram di lingkungan UNISBANK menggunakan *Natural Language Processing* berbasis Rasa dengan 47 *intent* dan 394 contoh; meskipun akurat untuk pertanyaan yang telah terlatih, sistem tersebut bergantung pada basis pengetahuan statis yang rentan ketinggalan informasi. Purwanto dkk. [5] mengintegrasikan platform n8n dengan algoritma *generative chatbot* untuk otomatisasi layanan penerimaan mahasiswa baru, namun sistem tersebut hanya menangani satu kategori layanan dan sangat bergantung pada stabilitas API pihak ketiga. Karhu [6] mendemonstrasikan sistem *Multi-Agent* berbasis n8n yang menghubungkan Supabase, SerpAPI, dan Telegram untuk perencanaan perjalanan kompleks; pendekatan ini efektif memecah tugas namun mengalami latensi akibat panjangnya rantai penalaran antar agen. Riza dkk. [7] merancang agen AI lokal mandiri berbasis LLM melalui orkestrasi n8n dengan memori percakapan pada RAG; meskipun menjamin privasi data, implementasi ini menuntut spesifikasi perangkat keras tinggi (GPU) yang tidak realistis bagi institusi dengan sumber daya terbatas. Tran Huynh [8] mengintegrasikan n8n dengan OpenAI untuk memusatkan informasi layanan mahasiswa; hasilnya efektif namun dokumen sumber harus diperbarui secara manual. Secara umum, *gap* yang tersisa pada penelitian-penelitian tersebut adalah: (1) keterbatasan pada satu kategori layanan saja, (2) ketidakmampuan menangani informasi dinamis dari sumber eksternal secara *real-time*, (3) tidak adanya mekanisme RAG berbasis dokumen resmi institusi, serta (4) tidak tersedianya pemrosesan masukan suara dan penanganan kesalahan terpusat.

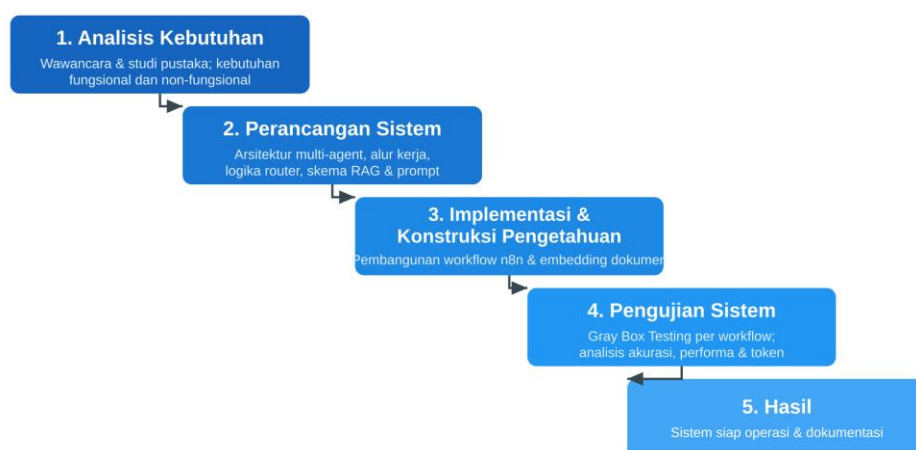
Penelitian ini mengusulkan sistem *chatbot* berbasis *AI Agent* dengan arsitektur *Multi-Agent Systems* (MAS) sebagai solusi komprehensif. Pada kerangka MAS, permasalahan kompleks dapat diuraikan menjadi sub-tugas yang ditangani oleh agen-agen spesialis yang saling berkoordinasi [9]. Perancangan topologi dan instruksi *prompt* yang tepat pada MAS terbukti menghasilkan kinerja yang lebih baik dibandingkan penggunaan agen tunggal [10]. Untuk menjamin akurasi informasi yang berkaitan dengan regulasi resmi dan mereduksi risiko halusinasi, diterapkan metode *Retrieval-Augmented Generation* (RAG) yang mencocokkan kemiripan makna antara pertanyaan dan dokumen resmi kampus [11]; penggunaan RAG dengan sumber informasi yang andal terbukti secara signifikan mengurangi tingkat halusinasi *chatbot* AI [12]. Platform *no-code* n8n dipilih untuk menurunkan hambatan teknis pengembangan melalui perakitan alur kerja secara visual [13]. Informasi dinamis dari luar kampus seperti berita dan beasiswa diperoleh melalui teknik *Web Scraping* dan Google Custom Search API [14]. Koordinasi antar agen dijalankan oleh *Router* yang mengidentifikasi satu maksud dominan (*single intent*) agar kecepatan respons terjaga [10]. Telegram dipilih sebagai antarmuka karena ringan, aman, berbasis *cloud*, dan mendukung pengembangan *bot* [15]. Kebaruan (*novelty*) penelitian ini terletak pada integrasi arsitektur *multi-agent no-code* yang menggabungkan RAG atas dokumen resmi internal, pengambilan informasi dinamis eksternal, pemrosesan masukan suara (*voice note*), dan penanganan kesalahan terpusat dalam satu sistem layanan kemahasiswaan yang terintegrasi, hal tersebut merupakan kombinasi fitur yang belum terdapat pada penelitian-penelitian sebelumnya.

Penelitian ini memiliki dua tujuan utama. Tujuan pertama adalah merancang dan membangun aplikasi *chatbot* layanan kemahasiswaan berbasis AI Agent yang terintegrasi dengan Telegram untuk memudahkan akses informasi akademik dan KIP Kuliah. Tujuan kedua adalah menerapkan metode RAG dan mekanisme koordinasi agen berbasis *single intent* agar sistem menghasilkan respons yang akurat, relevan, dan terfokus dalam menangani pertanyaan

berulang mahasiswa. Hasil penelitian diharapkan memberikan manfaat bagi mahasiswa berupa akses layanan sepanjang waktu, bagi staf berupa pengurangan beban administratif, serta bagi institusi berupa peningkatan efisiensi layanan informasi publik.

2. Metodologi

Penelitian ini merupakan penelitian pengembangan (*Research and Development*) dengan keluaran berupa sistem agen cerdas. Model pengembangan yang digunakan adalah *Waterfall*, yaitu salah satu model *System Development Life Cycle* (SDLC) yang menjalankan tahapan secara berurutan dan sistematis, dengan keluaran setiap tahap menjadi masukan bagi tahap berikutnya. Model ini dipilih karena kebutuhan sistem telah teridentifikasi secara jelas pada awal penelitian melalui wawancara dan studi pustaka sehingga alur pengembangan yang linier lebih menjamin keterurutan proses. Kelima tahapan metode penelitian ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Metode Penelitian (Model Waterfall)

Data primer diperoleh melalui wawancara dengan staf layanan kemahasiswaan UNISBANK untuk mengidentifikasi kendala operasional. Data sekunder berupa dokumen Buku Peraturan Akademik UNISBANK (PERAK 2021) dan Buku Panduan KIP Kuliah 2025 yang menjadi basis pengetahuan sistem, serta kajian literatur dari jurnal ilmiah. Uraian setiap tahapan dipaparkan sebagai berikut.

2.1 Analisis Kebutuhan

Tahap ini mengidentifikasi kebutuhan fungsional dan non-fungsional berdasarkan hasil pengumpulan data. Kebutuhan fungsional utama diringkas pada Tabel 2.

Tabel 2. Kebutuhan Fungsional Sistem

KF	Deskripsi Kebutuhan Fungsional
1	Beroperasi penuh di Telegram serta menerima dan memproses masukan teks maupun suara (voice note) melalui Automatic Speech Recognition (ASR).
2	Menyediakan mekanisme Router untuk mengklasifikasikan satu maksud dominan (single intent) dan mengarahkannya ke satu agen spesialis yang relevan.
3	Menjawab pertanyaan berbasis dokumen resmi (PERAK 2021 dan Panduan KIP 2025) menggunakan metode RAG.
4	Mengambil informasi dinamis eksternal: berita melalui scraping situs resmi kampus dan beasiswa melalui Google Custom Search API.
5	Menyimpan konteks percakapan jangka pendek (short-term memory) pada alur Peraturan dan KIP untuk pertanyaan lanjutan.
6	Memvalidasi fitur reply Telegram serta menangani sapaan atau masukan di luar cakupan dengan respons informatif.

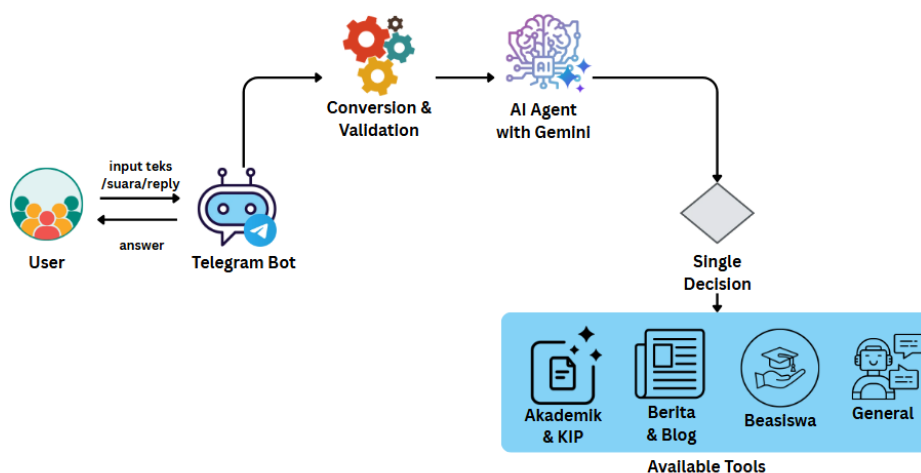
KF Deskripsi Kebutuhan Fungsional

- 7 Terintegrasi dengan LLM Google Gemini dan memiliki mekanisme penanganan kesalahan (error handling) yang terpusat.

Kebutuhan non-fungsional mencakup perangkat lunak, yaitu n8n sebagai platform orkestrasi, JavaScript, Google Gemini Chat, serta Supabase berbasis PostgreSQL dan pgvector, dan perangkat keras berupa lingkungan pengembangan serta server Sumopod (1 Core CPU, RAM 1024 MB, template n8n Plus).

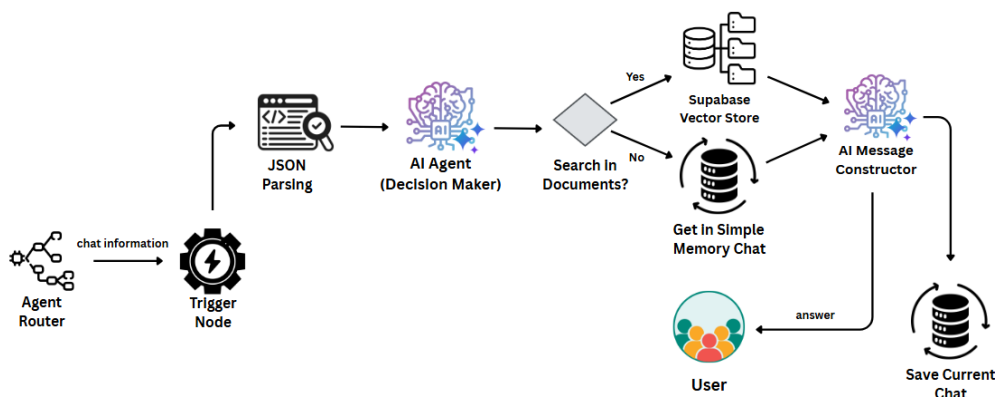
2.2 Perancangan Sistem

Arsitektur sistem dirancang menggunakan pendekatan *Multi-Agent* dengan satu *Main Agent Router* sebagai gerbang utama dan empat sub-*workflow* spesialis, ditambah satu *Global Error Handler*. Router menerima pesan Telegram, mengonversi suara menjadi teks bila diperlukan, mengklasifikasikan *intent*, kemudian memanggil salah satu dari empat jalur eksekusi, yaitu General, Berita/Blog, Peraturan/KIP, atau Beasiswa. Rancangan alur Router ditunjukkan pada Gambar 2.



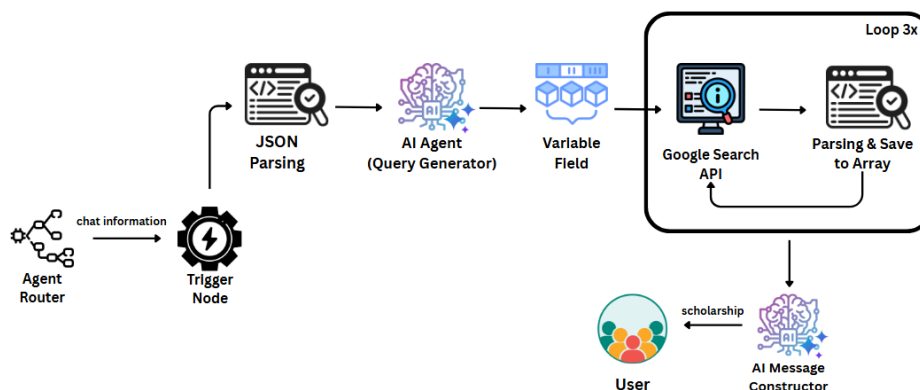
Gambar 2. Rancangan Alur Main Agent Router

Alur Peraturan dan KIP merupakan inti sistem yang menerapkan RAG. Dokumen resmi terlebih dahulu dipecah menjadi potongan (*chunk*) berukuran 1.000 karakter dengan *overlap* 200 karakter, kemudian diubah menjadi vektor melalui model *embedding* HuggingFace Inference dan disimpan pada Supabase *Vector Store*. Ketika menerima pertanyaan, agen menentukan apakah jawaban diambil dari memori percakapan atau melalui pencarian semantik ke basis data vektor; apabila informasi tidak ditemukan, agen menjawab berdasarkan pengetahuan umum disertai *disclaimer*. Rancangan alur jawaban RAG ditunjukkan pada Gambar 3.



Gambar 3. Rancangan Alur Jawaban RAG (Peraturan dan KIP)

Alur Beasiswa menggunakan *AI Agent* sebagai generator kueri dengan dua mode, yaitu mode *Strict* yang menerapkan teknik *Google Dorking* dan mode *Reguler* yang diaktifkan *flag "-r"*. Sistem memanggil Google Custom Search API sebanyak tiga putaran, mengumpulkan hasil ke dalam *accumulated_data*, kemudian agen kurator menyaring dan menyajikan paling banyak delapan rekomendasi. Rancangan alur ditunjukkan pada Gambar 4.



Gambar 4. Rancangan Alur Beasiswa Flow

Selain ketiga alur tersebut, terdapat *General Flow* yang menangani sapaan dan masukan di luar cakupan dengan menjelaskan kapabilitas sistem, sekaligus memvalidasi fitur *reply* agar hanya balasan bertopik akademik atau KIP yang diproses. Adapun *Global Error Handler* dirancang sebagai *workflow* pengaman yang dipicu otomatis oleh n8n saat terjadi kegagalan pada alur mana pun; *workflow* ini mencatat detail kejadian ke Google Sheets sebagai sarana pemantauan bagi pengembang. Rancangan tersebut melengkapi arsitektur agar sistem tetap tangguh terhadap masukan tak terduga dan kegagalan proses.

Setiap *AI Agent* dirancang menggunakan pola *Role-Task-Constraint*, yaitu pendefinisian peran, tugas, dan batasan secara eksplisit di dalam instruksi sistem agar perilaku dan format keluaran setiap agen konsisten. Ringkasan rancangan *prompt* setiap agen disajikan pada Tabel 3.

Tabel 3. Ringkasan Struktur Prompt Setiap AI Agent

Agen	Peran & Tugas Utama	Batasan (Constraint) Utama
Router (Main Agent)	Mengklasifikasikan maksud pengguna dan meneruskan data mentah ke satu tool yang paling sesuai.	Hanya memanggil satu tool per eksekusi; nilai query diteruskan apa adanya; tersedia tool Think untuk maksud ambigu.
Peraturan & KIP	Menjawab pertanyaan akademik dengan prioritas pencarian ke basis data vektor dokumen resmi.	Dilarang mengarang nomor pasal; wajib disclaimer bila memakai pengetahuan umum; keluaran maksimal 3.600 karakter.
Blog & Berita	Menyaring dan meringkas artikel kampus (5W+1H) menjadi ringkasan eksekutif berformat rapi.	JSON Data Filter mengembalikan array murni; ringkasan maksimal 40 kata per artikel dan 5 berita.
Beasiswa	Menghasilkan kueri pencarian (Strict/Reguler) lalu mengkurasi hasil menjadi rekomendasi terstruktur.	Menyertakan tahun berjalan; membuang duplikasi; keluaran maksimal 8 rekomendasi dalam 3.600 karakter.

2.3 Implementasi, Pengujian, dan Teknik Analisis

Rancangan diimplementasikan pada platform n8n yang berjalan di Sumopod dengan melakukan konfigurasi kredensial layanan (Telegram, Google Gemini, HuggingFace, Supabase, Google Drive, dan Google Sheets) serta membangun kelima *workflow*.

Setiap fitur fungsional pada Tabel 2 diuji mengikuti konsep eksperimen tunggal-faktor, yaitu setiap fungsi diberikan beberapa perlakuan (*treatment*) berupa skenario masukan yang berbeda, kemudian hasil setiap perlakuan diamati melalui indikator yang telah didefinisikan secara operasional. Variabel bebas pada rancangan ini adalah jenis perlakuan atau skenario masukan yang bersifat kategorikal dan berbeda pada tiap kebutuhan fungsional. Variabel terikat yang diamati terdiri atas empat indikator: (1) *kesesuaian fungsional*, yaitu kategori Sesuai/Sebagian Sesuai/Tidak Sesuai yang diperoleh dengan membandingkan keluaran aktual pada tampilan eksekusi n8n dan Telegram terhadap keluaran yang diharapkan sesuai rancangan; (2) *akurasi jawaban*, khusus untuk alur berbasis AI generatif, yaitu kategori Akurat/Kurang Akurat yang dinilai dari kesesuaian informasi yang disampaikan dengan sumber rujukan (dokumen resmi, situs asli, atau hasil pencarian); (3) *waktu respons*, yaitu durasi dalam detik yang diukur dari saat pesan diterima hingga jawaban ditampilkan di Telegram; dan (4) *konsumsi token*, yaitu jumlah token yang tercatat pada log eksekusi setiap pemanggilan LLM. Pemetaan setiap kebutuhan fungsional terhadap perlakuan dan variabel terikat yang diamati disajikan pada Tabel 4.

Tabel 4. Desain Eksperimen Pengujian Fungsional

KF	Fungsi yang Diuji	Perlakuan yang Diberikan	Variabel Terikat yang Diamati
1	Alur Main Agent Router	P1: pesan teks; P2: pesan suara (voice note); P3: pesan reply	Kesesuaian fungsional; waktu respons
2	Klasifikasi single intent	P1: intent Peraturan/KIP; P2: intent Beasiswa; P3: intent Berita; P4: intent General	Kesesuaian fungsional; waktu respons
3, 5	Jawaban RAG atas dokumen resmi & memori percakapan	P1: pertanyaan tersedia dalam dokumen; P2: pertanyaan di luar dokumen; P3: pertanyaan lanjutan (uji memori)	Kesesuaian fungsional; akurasi jawaban; waktu respons; konsumsi token
4	Pencarian informasi eksternal (berita & beasiswa)	P1: permintaan berita terbaru; P2: pencarian berita per topik; P3: pencarian beasiswa mode Strict; P4: pencarian beasiswa mode Reguler	Kesesuaian fungsional; akurasi jawaban; waktu respons; konsumsi token
6	Validasi reply dan penanganan di luar cakupan	P1: sapaan; P2: reply di luar topik akademik/KIP	Kesesuaian fungsional; waktu respons
7	Integrasi LLM dan penanganan kesalahan	P1: eksekusi normal; P2: eksekusi dipicu gagal (error)	Kesesuaian fungsional; waktu respons

Pengujian dilakukan menggunakan metode *Gray Box Testing*, yaitu pengujian yang menilai fungsionalitas dari sudut pandang pengguna sekaligus mengamati proses internal *workflow* melalui tampilan eksekusi n8n. Setiap perlakuan pada Tabel 4 dijalankan melalui Telegram, dan hasil setiap perlakuan dicatat untuk keempat indikator yang relevan. Teknik analisis data bersifat deskriptif kuantitatif dan kualitatif, yaitu menyajikan nilai atau kategori setiap variabel terikat per perlakuan, kemudian menginterpretasikan pola yang muncul antarperlakuan dan antar-*workflow*.

3. Hasil dan Pembahasan

Sistem *chatbot* layanan kemahasiswaan UNISBANK berhasil dibangun dan dipublikasikan secara penuh. Satu *Main Agent Router*, empat sub-*workflow*, dan satu *Global Error Handler* telah aktif berjalan pada instansi n8n di Sumopod dan dapat diakses langsung

melalui aplikasi Telegram. Hasil pengujian setiap perlakuan pada Tabel 4 dipaparkan pada subbab 3.1 hingga 3.5, diikuti pembahasan menyeluruh pada subbab 3.6.

3.1 Pengujian Main Agent Router dan General Flow

Sembilan perlakuan yang diujikan pada kebutuhan fungsional Main Agent Router, klasifikasi single intent, dan General Flow (KF1, KF2, KF6) beserta hasilnya dirangkum pada Tabel 5.

Tabel 5. Pengujian Perlakuan pada Main Agent Router dan General Flow

No	Perlakuan	Ekspektasi	Hasil Pengujian	Waktu Respons
1	Pesan teks	Sistem mengenali maksud pesan dan meneruskannya ke sub-workflow yang tepat	Pesan teks dikenali dan diarahkan ke sub-workflow sesuai maksud (Sesuai)	2–3 dtk
2	Pesan suara (voice note)	Suara ditranskripsi menjadi teks lalu diproses seperti pesan teks	Transkripsi oleh Gemini berhasil; pesan diproses melalui alur yang sama (Sesuai)	2–3 dtk
3	Pesan reply	Reply diarahkan ke cabang AI Agent yang sesuai konteks balasan	Reply diarahkan dengan tepat ke agen terkait (Sesuai)	2–3 dtk
4	Kalimat bermaksud Peraturan/KIP	Router mengklasifikasikan intent dan mengarahkan ke alur Peraturan & KIP	Diarahkan ke alur Peraturan & KIP sesuai intent (Sesuai)	2–3 dtk
5	Kalimat bermaksud Beasiswa	Router mengarahkan ke alur Beasiswa	Diarahkan ke alur Beasiswa sesuai intent (Sesuai)	2–3 dtk
6	Kalimat bermaksud Berita	Router mengarahkan ke alur Berita & Blog	Diarahkan ke alur Berita & Blog sesuai intent (Sesuai)	2–3 dtk
7	Kalimat bermaksud umum (General)	Router mengarahkan ke General Flow	Diarahkan ke General Flow sesuai intent (Sesuai)	2–3 dtk
8	Sapaan (General Flow)	Sistem membalas dengan penjelasan kapabilitas layanan	Sistem membalas dengan penjelasan kapabilitas (Sesuai)	< 1 dtk
9	Reply di luar topik akademik/KIP	Sistem menolak dan memberi keterangan informatif	Sistem menolak disertai pesan informatif (Sesuai)	< 1 dtk

Tabel 5 menunjukkan bahwa seluruh perlakuan mencapai kesesuaian fungsional Sesuai. General Flow mencatat waktu respons tercepat (di bawah 1 detik) karena tidak melibatkan pemrosesan AI maupun layanan eksternal, sedangkan Router dan klasifikasi intent membutuhkan 2–3 detik dengan penalaran pemilihan tool di bawah 700 milidetik. Ditemukan kendala berupa perubahan nilai query mentah oleh Gemini yang diatasi dengan mengaktifkan Max Iterations satu kali dan Execute Once agar penalaran tidak berulang serta konsumsi token lebih hemat.

3.2 Pengujian Peraturan dan KIP Flow (RAG)

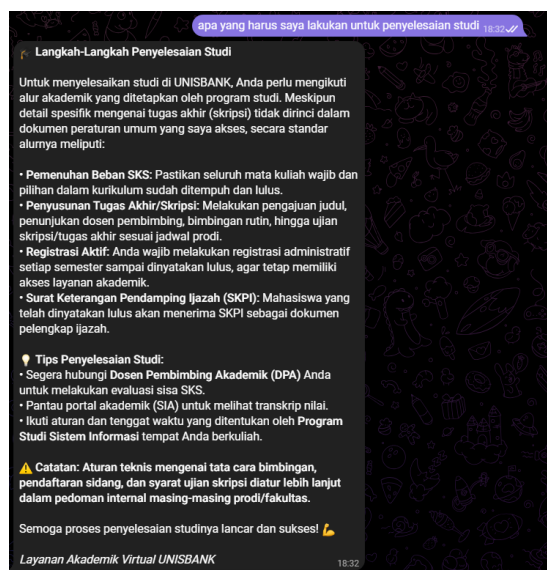
Dokumen PERAK 2021 dan Panduan KIP 2025 diindeks menjadi vektor pada tabel *documents* di Supabase. Proses pengindeksan dijalankan dua kali secara manual, masing-masing untuk berkas PERAK 2021 dan Panduan KIP 2025, karena *node* pengunduh berkas hanya memproses satu dokumen per eksekusi. Keberhasilan pengindeksan diverifikasi melalui *Table Editor* Supabase yang menampilkan potongan dokumen beserta nilai vektornya.

Ketiga perlakuan yang diujikan pada kebutuhan fungsional RAG dan memori percakapan (KF3, KF5) beserta hasilnya dirangkum pada Tabel 6.

Tabel 6. Pengujian Perlakuan pada Peraturan dan KIP Flow (RAG)

No	Perlakuan	Ekspektasi	Hasil Pengujian	Waktu Respons	Token
1	Pertanyaan tersedia dalam dokumen (PERAK 2021/Panduan KIP 2025)	Sistem mencari potongan dokumen relevan dan menjawab disertai rujukan pasal/bagian	Jawaban akurat disertai rujukan pasal/bagian dokumen (Sesuai; Akurat)	< 10 dtk	> 4.000
2	Pertanyaan di luar cakupan dokumen	Sistem menjawab dari pengetahuan umum disertai disclaimer	Jawaban disertai disclaimer sesuai rancangan (Sesuai)	< 10 dtk	> 4.000
3	Pertanyaan lanjutan (uji memori)	Simple Memory mempertahankan konteks sehingga pertanyaan lanjutan terjawab tanpa mengulang detail	Konteks percakapan dipertahankan berbasis chatId (Sesuai)	< 10 dtk	> 4.000

Tabel 6 menunjukkan sistem konsisten menautkan jawaban ke sumber resmi ketika informasi tersedia dalam dokumen, dan menghindari informasi yang berada di luar cakupan dokumen melalui disclaimer eksplisit, sebagaimana ditunjukkan pada Gambar 5. Fitur Simple Memory berbasis chatId terbukti efektif mempertahankan konteks sehingga pertanyaan lanjutan tidak perlu mengulang detail. Kemampuan menautkan jawaban ke sumber resmi menjadi pembeda utama dari pendekatan berbasis dokumen statis yang rentan usang seperti pada [8].

**Gambar 5. Pengujian Pertanyaan Peraturan Akademik pada Peraturan dan KIP Flow**

3.3 Pengujian Berita dan Blog Flow

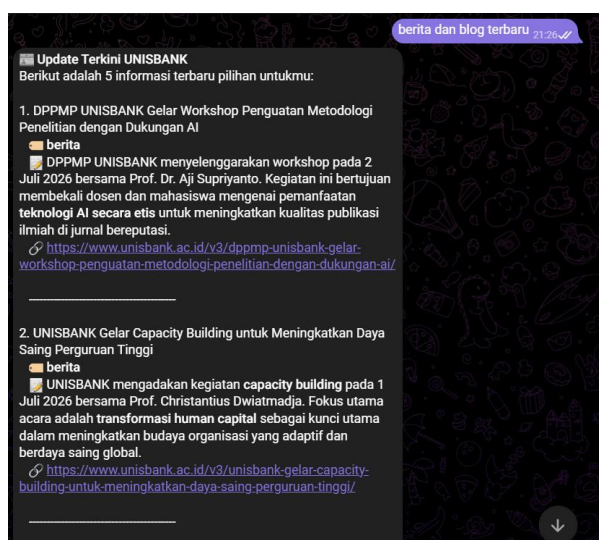
Kedua perlakuan yang diujikan pada kebutuhan fungsional pencarian berita eksternal (bagian dari KF4) beserta hasilnya dirangkum pada Tabel 7.

Tabel 7. Pengujian Perlakuan pada Berita dan Blog Flow

No	Perlakuan	Ekspektasi	Hasil Pengujian	Waktu Respons	Token
1	Permintaan berita terbaru	Sistem melakukan scraping situs resmi UNISBANK dan	Artikel disaring dan diringkas dengan format header, judul, kategori,	> 10 dtk	> 6.000

No	Perlakuan	Ekspektasi	Hasil Pengujian	Waktu Respons	Token
2	Pencarian berita per topik	meringkas artikel terbaru Sistem menyaring artikel sesuai topik yang diminta	ringkasan, dan tautan (Sesuai; Akurat) Artikel relevan dengan topik berhasil disajikan (Sesuai; Akurat)	> 10 dtk	> 6.000

Tabel 7 menunjukkan sistem konsisten merujuk pada artikel sumber pada kedua perlakuan, sebagaimana ditunjukkan pada Gambar 6. Proses ini membutuhkan waktu relatif lama karena melibatkan *scraping* dan dua tahap pemrosesan AI secara berurutan, yaitu penyaringan artikel dan peringkasan, sehingga turut menyumbang konsumsi *token* yang cukup tinggi dibandingkan alur yang tidak melibatkan layanan eksternal.



Gambar 6. Pengujian Permintaan Berita pada Berita dan Blog Flow

3.4 Pengujian Beasiswa Flow

Kedua perlakuan yang diujikan pada kebutuhan fungsional pencarian beasiswa (bagian dari KF4) beserta hasilnya dirangkum pada Tabel 8.

Tabel 8. Pengujian Perlakuan pada Beasiswa Flow

No	Perlakuan	Ekspektasi	Hasil Pengujian	Waktu Respons	Token
1	Pencarian mode Strict (Google Dorking)	Sistem menghasilkan kueri beroperator logika lengkap dan mengumpulkan hasil dari tiga putaran pencarian	Kueri lengkap terbentuk; 30 hasil terkumpul dan dikurasi menjadi 8 rekomendasi (Sesuai; Akurat)	> 10 dtk	± 10.000
2	Pencarian mode Reguler (flag "-r")	Sistem menghasilkan kueri pencarian yang lebih sederhana	Kueri sederhana terbentuk; hasil dikurasi menjadi 8 rekomendasi (Sesuai; Akurat)	> 10 dtk	± 10.000

Tabel 8 menunjukkan kedua mode kueri berhasil menghasilkan struktur pencarian sesuai rancangan, dengan *Agent Validator* berperan penting menyaring 30 hasil pencarian mentah menjadi delapan rekomendasi yang relevan dan bebas duplikasi, sebagaimana ditunjukkan pada Gambar 7. Alur ini memiliki konsumsi *token* tertinggi di antara seluruh *workflow* karena memproses 30 data secara bersamaan dalam satu pemanggilan agen kurator.

Ringkasan hasil pengujian seluruh perlakuan, waktu respons, dan konsumsi *token* disajikan pada Tabel 10. Waktu respons bervariasi sesuai kompleksitas proses; alur tanpa AI dan tanpa layanan eksternal memiliki waktu respons yang jauh lebih singkat dibandingkan alur yang melibatkan *scraping*, API eksternal, atau pencarian semantik.

Tabel 10. Ringkasan Hasil Pengujian, Waktu Respons, dan Konsumsi Token

Workflow	Ekspektasi	Hasil Pengujian	Waktu	Token	Status
Main Agent Router	Pesan teks, suara, dan reply dikenali serta diarahkan ke sub-workflow yang tepat sesuai intent	Seluruh perlakuan diarahkan sesuai intent (P1–P3)	2–3 dtk	>1.000	Sesuai
General Flow	Sapaan dan reply di luar topik akademik/KIP direspons dengan penjelasan atau penolakan yang informatif	Sapaan dan reply di luar konteks direspons sesuai rancangan (P1–P2)	< 1 dtk	—	Sesuai
Peraturan & KIP (RAG)	Jawaban berbasis dokumen resmi disertai rujukan; pertanyaan di luar dokumen disertai disclaimer; konteks lanjutan dipertahankan	Jawaban akurat disertai sumber, disclaimer, dan memori percakapan berfungsi (P1–P3)	< 10 dtk	>4.000	Sesuai
Berita & Blog	Artikel terbaru maupun per topik disaring dari situs resmi dan diringkas secara terstruktur	Scraping dan ringkasan artikel kampus berhasil sesuai format (P1–P2)	> 10 dtk	>6.000	Sesuai
Beasiswa	Kueri mode Strict/Reguler menghasilkan rekomendasi beasiswa yang relevan dan bebas duplikasi	Dua mode kueri berhasil menghasilkan kurasi 8 rekomendasi (P3–P4)	> 10 dtk	±10.000	Sesuai
Global Error Handler	Kegagalan pada workflow mana pun tercatat otomatis dan pengguna menerima notifikasi	Log tercatat lengkap; notifikasi ke pengguna terkendala (P1–P2)	> 6 dtk	—	Sebagian

3.6 Pembahasan

Hasil pengujian pada kelima alur menunjukkan bahwa seluruh perlakuan pada Tabel 4 tercapai dengan kesesuaian fungsional Sesuai, dengan satu pengecualian sebagian pada notifikasi *error* lintas *workflow*. Temuan ini menjawab tujuan pertama penelitian, yaitu membangun *chatbot* layanan kemahasiswaan berbasis AI Agent yang terintegrasi dengan Telegram, karena keenam *workflow* terbukti dapat diakses dan digunakan secara langsung oleh mahasiswa. Pencapaian tujuan kedua, yaitu penerapan RAG dan *routing single intent* agar jawaban akurat dan terfokus, tercermin dari kemampuan Router mengarahkan satu maksud dominan ke satu agen spesialis pada seluruh perlakuan *routing* yang diuji, serta kemampuan alur Peraturan dan KIP menjawab dengan rujukan dokumen resmi pada perlakuan P1.

Kemampuan sistem menautkan jawaban ke potongan dokumen resmi sejalan dengan gagasan awal *Retrieval-Augmented Generation* yang dikemukakan Lewis dkk. [16], yang menyatakan bahwa proses generasi jawaban dapat diperkuat validitasnya bila dipadukan dengan pengambilan (*retrieval*) atas sumber pengetahuan eksternal yang dapat dirujuk, sehingga jawaban menjadi lebih spesifik dan faktual dibandingkan model generatif murni. Perlakuan P2 pada alur Peraturan dan KIP, yang tetap dijawab disertai *disclaimer* ketika

pertanyaan berada di luar cakupan dokumen, juga konsisten dengan prinsip tersebut karena sistem tidak memaksakan jawaban seolah bersumber dari dokumen ketika *retrieval* tidak menemukan kecocokan yang memadai.

Dibandingkan dengan *chatbot rule-based* yang sebelumnya dikembangkan pada konteks yang sama di UNISBANK, yaitu Al Fajri dan Hartono [17], hasil penelitian ini memperlihatkan perluasan cakupan jawaban yang signifikan: alur Peraturan dan KIP mampu menjawab pertanyaan yang tidak terdaftar secara eksplisit dalam skenario dialog, karena bergantung pada pencarian semantik dan bukan pencocokan *intent* yang telah dilatih. Namun demikian, cakupan pengujian Al Fajri dan Hartono [17] yang berfokus pada volume kalimat NLU (213 contoh kalimat dan 27 skenario dialog) menyoroti bahwa peningkatan cakupan jawaban pada penelitian ini turut menuntut konsumsi sumber daya komputasi yang lebih besar, sebagaimana tampak pada tingginya konsumsi *token* pada alur Beasiswa (Tabel 5). Sejalan dengan hal tersebut, temuan bahwa arsitektur *multi-agent* dengan pembagian tugas antaragen mampu menjaga kesesuaian fungsional pada beban pengujian yang bervariasi juga mendukung hasil Klang dkk. [18], yang menunjukkan bahwa pemisahan tugas antaragen (*task partitioning*) pada sistem *multi-agent* berbasis LLM mampu mempertahankan akurasi sekaligus menekan konsumsi *token* hingga puluhan kali lipat dibandingkan agen tunggal yang menangani seluruh tugas secara serentak. Meskipun demikian, alur Beasiswa pada penelitian ini memproses 30 data hasil pencarian dalam satu pemanggilan agen kurator tanpa pemecahan lebih lanjut, berbeda dari pendekatan granular pada penelitian tersebut, sehingga menjadi salah satu penyebab tingginya konsumsi *token* pada alur ini.

Temuan penelitian ini memberikan kontribusi bagi bidang sistem informasi dan kecerdasan buatan terapan, khususnya dalam mengintegrasikan RAG, *routing multi-agent* berbasis *single intent*, pencarian informasi eksternal, pemrosesan multimodal (teks dan suara), serta penanganan kesalahan terpusat ke dalam satu kerangka layanan administratif berbasis platform *no-code*. Integrasi tersebut memperluas kerangka implementasi AI pada pendidikan tinggi yang dikemukakan Rahmawati dkk. [19], yang menekankan bahwa optimalisasi administrasi akademik merupakan salah satu peluang strategis penerapan AI di perguruan tinggi Indonesia, sekaligus menunjukkan salah satu bentuk konkret kerangka tersebut yang teruji secara fungsional pada institusi dengan sumber daya terbatas.

Penelitian ini memiliki sejumlah keterbatasan yang membuka peluang bagi penelitian lanjutan. Pertama, pengujian yang dilakukan bersifat fungsional (*Gray Box Testing*) dan belum mengukur persepsi maupun tingkat penerimaan pengguna secara kuantitatif. Merujuk pada pendekatan *Technology Acceptance Model* dan *User Experience Questionnaire* yang diterapkan Nasrul dkk. [20] dalam mengevaluasi *chatbot* AI generatif pada konteks pendidikan, penelitian selanjutnya disarankan mengukur persepsi kegunaan, kemudahan, dan pengalaman pengguna mahasiswa UNISBANK secara kuantitatif untuk melengkapi hasil pengujian fungsional ini. Kedua, konsumsi *token* yang tinggi pada alur Beasiswa menunjukkan peluang optimalisasi melalui pemecahan tugas kurasi data ke dalam sub-agen yang lebih granular, sebagaimana direkomendasikan oleh temuan Klang dkk. [18] mengenai efisiensi *token* pada arsitektur *multi-agent* yang terpartisi dengan baik. Ketiga, keterbatasan teknis berupa perubahan nilai *query* oleh Gemini serta kegagalan notifikasi *error* lintas *workflow* akibat keterbatasan *customData* n8n membuka peluang penelitian mengenai mekanisme *state management* lintas *workflow* yang lebih andal pada platform *low-code*.

4. Simpulan

Penelitian ini mengembangkan sistem *chatbot* layanan kemahasiswaan UNISBANK berbasis AI Agent pada platform *low-code* n8n dengan arsitektur *Multi-Agent* yang terdiri atas satu *Main Agent Router*, empat sub-*workflow* spesialis (General, Berita dan Blog, Beasiswa, serta Peraturan dan KIP), dan satu *Global Error Handler*. Seluruh fitur utama berhasil diimplementasikan, mencakup pemrosesan masukan teks dan suara, *routing* berbasis *single intent* menggunakan Google Gemini, mekanisme RAG untuk menjawab pertanyaan berdasarkan dokumen resmi, pencarian informasi dinamis dari situs kampus dan Google Custom Search API, serta penyimpanan konteks percakapan jangka pendek.

Pengujian *Gray Box* menunjukkan sistem mampu memberikan respons yang akurat dan relevan sesuai kategori permintaan. Alur Peraturan dan KIP menjawab secara konsisten disertai sumber dan *disclaimer*, alur Beasiswa mengkurasi rekomendasi melalui dua mode kueri, dan alur Berita meringkas konten terkini secara *real-time*. Waktu respons bervariasi

menurut kompleksitas, dari kurang dari satu detik pada *General Flow* hingga lebih dari sepuluh detik pada alur yang melibatkan layanan eksternal, dengan konsumsi *token* tertinggi pada alur Beasiswa. Terdapat dua keterbatasan teknis, yaitu perubahan nilai *query* mentah oleh Gemini pada Router dan kegagalan notifikasi *error* akibat keterbatasan *customData* lintas *workflow*. Penelitian selanjutnya disarankan menyederhanakan struktur *prompt* untuk menekan konsumsi *token*, membandingkan performa beragam LLM, menyimpan *chatId* pada basis data eksternal agar notifikasi lintas *workflow* berfungsi, serta memperluas cakupan dokumen dan menjadwalkan pembaruan indeks vektor secara berkala.

Daftar Referensi

- [1] N. F. Davar, M. A. A. Dewan, and X. Zhang, "AI Chatbots in Education: Challenges and Opportunities," *Information*, vol. 16, no. 3, p. 235, Mar. 2025, doi: 10.3390/info16030235.
- [2] J. Zhang, "AI Agents in Education: Four Trends and a Practical Workflow," *AAAI Spring Symp. - Tech. Rep.*, vol. 5, no. 1, pp. 131–132, 2025, doi: 10.1609/aaais.v5i1.35578.
- [3] M. Aleedy, E. Atwell, and S. Meshoul, "A Multi-Agent Chatbot Architecture for AI-Driven Language Learning," *Appl. Sci.*, vol. 15, no. 19, p. 10634, Oct. 2025, doi: 10.3390/app151910634.
- [4] M. Qalimaturrahmah and D. B. Santoso, "Aplikasi Layanan dan Informasi Akademik Berbasis Chatbot Telegram Menggunakan Natural Language Processing," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 8, no. 2, pp. 434–443, Apr. 2024, doi: 10.35870/jtik.v8i2.1887.
- [5] E. Purwanto, S. Putri, M. Sandjoko, and S. A. Syahputra, "Integrasi N8N Dan Algoritma Generative Chatbot Untuk Transformasi Digital Penerimaan Mahasiswa Baru," *Prosiding-Seminar Nas. Teknol. Inf. Ilmu Komput.*, vol. 4, no. 1, pp. 19–25, 2025.
- [6] S. Karhu, "Automating Multi-City Concert Travel Planning with AI Agents," 2025.
- [7] F. Riza, S. Jamal Al Din, D. Yusuf Al Afghani, R. Setiabudi, and I. Teknologi Budi Utomo, "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," *J. Inf. Technol. Comput. Sci.*, vol. 8, no. 3, p. 2025, 2025.
- [8] A. D. Tran Huynh, "Streamlining Student Information Services in Hämeenlinna with an AI Chatbot," B.S. thesis, Degree Programme in Computer Applications, Häme University of Applied Sciences (HAMK), Hämeenlinna, Finland, 2025.
- [9] C. Jimenez-Romero, A. Yegenoglu, and C. Blum, "Multi-agent systems powered by large language models: applications in swarm intelligence," *Frontiers in Artificial Intelligence*, vol. 8, art. 1593017, May 2025, doi: 10.3389/frai.2025.1593017.
- [10] H. Zhou *et al.*, "Multi-Agent Design: Optimizing Agents with Better Prompts and Topologies," pp. 1–30, 2025, [Online]. Available: <http://arxiv.org/abs/2502.02533>
- [11] I. Pujiono, I. M. Agtyaputra, and Y. Ruldeviyani, "Implementing Retrieval-Augmented Generation and Vector Databases for Chatbots in Public Services Agencies Context," *JITK (Jurnal Ilmu Pengetah. dan Teknol. Komputer)*, vol. 10, no. 1, pp. 216–223, 2024, doi: 10.33480/jitik.v10i1.5572.
- [12] S. Nishisako, T. Higashi, and F. Wakao, "Reducing Hallucinations and Trade-Offs in Responses in Generative AI Chatbots for Cancer Information: Development and Evaluation Study," *JMIR Cancer*, vol. 11, pp. e70176–e70176, Sep. 2025, doi: 10.2196/70176.
- [13] C. Jeong, "Beyond Text: Implementing Multimodal Large Language Model-Powered Multi-Agent Systems Using a No-Code Platform," *J. Intell. Inf. Syst.*, vol. 31, no. 1, pp. 191–231, 2025, doi: 10.13088/jiis.2025.31.1.191.
- [14] A. Z. Rizquina and C. I. Ratnasari, "Implementasi Web Scraping untuk Pengambilan Data Pada Website E-Commerce," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 5, no. 4, pp. 377–383, 2023, doi: 10.47233/jteksis.v5i4.913.
- [15] M. W. Hasyim, S. Pramono, and Sutrisno, "Web-Based Telegram Chatbot Management System: Create Chatbot Without Programming Language Requirements," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1096, no. 1, p. 012075, 2021, doi: 10.1088/1757-899x/1096/1/012075.
- [16] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Apr. 2021, [Online]. Available: <http://arxiv.org/abs/2005.11401>

-
- [17] M. R. Al Fajri and B. Hartono, "Pengembangan Aplikasi Chatbot Telegram Menggunakan Framework Rasa untuk Pelayanan Administrasi di Perguruan Tinggi Universitas Stikubank," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 8, no. 1, pp. 133–136, Jan. 2024, doi: 10.35870/jtik.v8i1.1370.
- [18] E. Klang *et al.*, "Orchestrated multi agents sustain accuracy under clinical-scale workloads compared to a single agent," *npj Heal. Syst.*, vol. 3, no. 1, p. 23, Mar. 2026, doi: 10.1038/s44401-026-00077-0.
- [19] A. Rahmawati, S. N. Amirah, and N. Wijaya, "Integrasi Kecerdasan Buatan dalam Pendidikan Tinggi Indonesia: Peluang, Tantangan, dan Kerangka Implementasi," *J. Teknol. Sist. Inf.*, vol. 6, no. 1, pp. 114–126, Apr. 2025, doi: 10.35957/jtsi.v6i1.11329.
- [20] I. Nasrul, A. Angraini, M. L. Hamzah, and E. Saputra, "Analysis Of User Experience Of ChatGPT And Gemini Users Using The User Experience Quistionnaire (UEQ) For Education," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 13, no. 3, pp. 352–359, Nov. 2024, doi: 10.32736/sisfokom.v13i3.2250.