

Pengembangan Chatbot Berbasis Agentic Retrieval-Augmented Generation pada Aplikasi Pencatatan Kalori

Haffidz Aimar Maulana

Program Studi Teknik Informatika, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta
Depok, Jawa Barat

Haffidz.aimar.maulana.tik22@mhsw.pnj.ac.id

Abstract - Calorie-tracking applications can support dietary self-monitoring, but manual input, limited local food coverage, and declining adherence often reduce sustained use. This study developed an agentic retrieval-augmented generation chatbot integrated into the NutriVision Android application to enable nutrition inquiries and food logging through Indonesian natural-language conversations. The system was developed through sequential requirement analysis, design, implementation, and testing. Its architecture combines a Kotlin Android client, a Flask application programming interface, PostgreSQL, a LangChain agent, and the GPT-OSS-120B reasoning model served through Groq at a temperature of 0.3. A structured knowledge base containing 769 Indonesian food records was collected from FatSecret Indonesia and enriched with serving variants, nutrient values, semantic embeddings, and a hierarchical navigable small-world index. Eight tools allow the agent to search foods, retrieve servings, read daily intake, add, edit, or delete food logs, and retrieve or update user profiles. Evaluation involved 100 agent-trajectory scenarios assessed using an LLM-as-a-Judge rubric, 20 black-box functional cases, and usability testing with 16 participants aged 20-25 years. The chatbot achieved a mean trajectory score of 0.90 with a 95% pass rate, 100% functional success, a System Usability Scale score of 80.63, and a Net Promoter Score of 62.5. These results indicate that an agentic architecture grounded in a local structured food database can provide functional, context-aware, and well-accepted conversational calorie tracking.

Keywords: agentic chatbot, calorie tracking, large language model, retrieval-augmented generation

Abstrak - Aplikasi pencatatan kalori dapat mendukung pemantauan pola makan, tetapi input manual, keterbatasan data makanan lokal, dan penurunan kepatuhan sering menghambat penggunaan berkelanjutan. Penelitian ini mengembangkan chatbot berbasis agentic retrieval-augmented generation yang terintegrasi pada aplikasi Android NutriVision agar pengguna dapat menanyakan informasi gizi dan mencatat makanan melalui percakapan bahasa Indonesia. Pengembangan dilakukan secara berurutan melalui analisis kebutuhan, perancangan, implementasi, dan pengujian. Arsitektur sistem menggabungkan aplikasi Android Kotlin, application programming interface Flask, PostgreSQL, LangChain Agent, serta model reasoning GPT-OSS-120B melalui Groq dengan temperature 0,3. Basis pengetahuan berisi 769 data makanan Indonesia yang dikumpulkan dari FatSecret Indonesia dan dilengkapi variasi porsi, kandungan gizi, embedding semantik, serta indeks hierarchical navigable small-world. Delapan tool memungkinkan agent mencari makanan, mengambil porsi, membaca konsumsi harian, menambah, mengubah, atau menghapus food log, serta mengambil dan memperbarui profil pengguna. Evaluasi mencakup 100 skenario trajectory dengan LLM-as-a-Judge, 20 kasus Black Box Testing, dan pengujian usability terhadap 16 responden berusia 20-25 tahun. Sistem memperoleh skor rata-rata trajectory 0,90 dengan pass rate 95%, keberhasilan fungsional 100%, skor System Usability Scale 80,63, dan Net Promoter Score 62,5. Hasil tersebut menunjukkan bahwa arsitektur agentic yang di-grounding pada basis data makanan lokal terstruktur mampu mendukung pencatatan kalori yang fungsional, kontekstual, dan diterima dengan baik oleh pengguna.

Kata kunci: agentic chatbot, large language model, pencatatan kalori, retrieval-augmented generation

I. PENDAHULUAN

Berat badan berlebih dan obesitas masih menjadi masalah kesehatan global. Organisasi Kesehatan Dunia melaporkan bahwa 43% populasi dewasa mengalami berat badan berlebih dan 16% hidup dengan obesitas pada 2022 [1]. Di Indonesia, Survei Kesehatan Indonesia 2023 mencatat prevalensi berat badan berlebih sebesar 14,4% dan obesitas sebesar 23,4% pada penduduk dewasa [2]. Salah satu pendekatan pengelolaan berat badan adalah *dietary self-monitoring*, yaitu mencatat makanan agar individu dapat mengevaluasi kesesuaian asupan energi dengan kebutuhan hariannya [3].

Aplikasi pencatatan kalori telah menyediakan basis data makanan, target energi, dan ringkasan makronutrien. Namun, penggunaan jangka panjang masih terkendala oleh input manual yang berulang, cakupan makanan yang terbatas, dan menurunnya kepatuhan pengguna [4]. Antarmuka percakapan dapat menurunkan beban interaksi karena pengguna cukup menyampaikan makanan yang dikonsumsi dalam bahasa sehari-hari, sebagaimana ditunjukkan pada studi pencatatan diet berbasis suara [23]. Intervensi *chatbot* juga dilaporkan efektif mendukung perubahan perilaku gaya hidup, termasuk pola makan dan aktivitas fisik [5].

Pemanfaatan *large language model* (LLM) secara langsung belum cukup karena model dapat menghasilkan informasi gizi yang tidak akurat, khususnya untuk makanan lokal yang kurang terwakili [20]. *Retrieval-Augmented Generation* (RAG) mengatasi masalah tersebut dengan mengambil fakta dari sumber eksternal sebelum respons dibentuk [6], [7]. Pendekatan *agentic* memperluas RAG melalui penalaran multi-langkah, penggunaan *tool*, dan memori sehingga model dapat membaca maupun mengubah data aplikasi secara terkontrol [8], [21].

Penelitian terdahulu telah memanfaatkan RAG untuk rekomendasi nutrisi [9], pemantauan diet berbasis percakapan [10], personalisasi rekomendasi makanan [11], dan agen kesehatan dengan *tool* eksternal [12]. Meskipun demikian, sebagian besar sistem berfokus pada rekomendasi atau analisis, menggunakan basis data pangan non-Indonesia, dan belum mengintegrasikan operasi transaksional seperti menambah, mengubah, dan menghapus catatan makan pada aplikasi pengguna. Celah tersebut menjadi dasar kontribusi penelitian ini: mengintegrasikan *agentic RAG* ke NutriVision menggunakan basis data makanan Indonesia, delapan *tool* backend, serta memori percakapan persisten.

Basis data lokal menjadi komponen penting karena nama makanan Indonesia sering memiliki variasi ejaan, istilah daerah, serta ukuran porsi yang tidak selalu tersedia pada korpus LLM. Sistem yang hanya mengandalkan pengetahuan parametrik dapat memberikan nilai gizi yang tampak meyakinkan tetapi tidak sesuai dengan sumber aplikasi. Oleh sebab itu, penelitian ini menggunakan basis data terstruktur dan pencarian semantik agar jawaban dapat ditelusuri hingga makanan dan porsi yang benar.

Keunikan penelitian terletak pada kombinasi empat lapisan: basis data makanan lokal, agent dengan operasi baca-tulis, memori percakapan persisten, dan evaluasi trajectory. Kombinasi tersebut membedakan NutriVision dari *chatbot* tanya jawab biasa karena sistem dapat mengeksekusi tujuan pengguna, memvalidasi hasil melalui layanan backend, kemudian mengembalikan ringkasan yang konsisten dengan kondisi akun.

Tujuan penelitian adalah mengembangkan *chatbot* yang mampu memberikan informasi gizi, mengelola *food log*, menampilkan progres kalori, dan memperbarui profil melalui bahasa alami. Kualitas sistem dievaluasi pada level proses agent, fungsionalitas, kemudahan penggunaan, dan kepuasan pengguna agar hasil tidak hanya menunjukkan keberhasilan respons akhir.

II. METODOLOGI PENELITIAN

A. Rancangan Pengembangan

Penelitian menggunakan pengembangan sistem secara sekuensial yang mencakup studi literatur, analisis kebutuhan, perancangan, implementasi, dan pengujian. Alur tersebut mengikuti prinsip *Waterfall* karena kebutuhan utama NutriVision telah terdefinisi dan setiap keluaran tahap digunakan sebagai masukan tahap berikutnya [18]. Analisis kebutuhan menghasilkan fungsi pencarian gizi, pengelolaan catatan makan, ringkasan harian, pembaruan profil, pemahaman konteks, serta pengelolaan histori percakapan.

Data makanan dikumpulkan melalui *web scraping* FatSecret Indonesia menggunakan Python, *requests*, BeautifulSoup, dan pandas. Data dibersihkan untuk menyatukan kategori, menghapus duplikasi, dan memisahkan entitas makanan dari variasi porsi. Proses menghasilkan 769 makanan dengan informasi kalori, protein, lemak, karbohidrat, serat, gula, sodium, dan kalium. Nama makanan dan porsi diubah menjadi vektor 768 dimensi; indeks HNSW pada PostgreSQL digunakan untuk mendukung pencarian semantik yang tetap terhubung dengan data relasional.

B. Pengolahan Data dan Basis Pengetahuan

Proses *scraping* mengambil daftar kategori, halaman makanan, dan seluruh variasi porsi. Setiap makanan dapat memiliki porsi berbasis gram, mililiter, sendok, butir, atau satuan kemasan. Tahap *cleaning* membentuk identitas makanan yang unik, memetakan kategori yang tidak konsisten, menggabungkan data makanan umum dengan label deteksi citra yang telah dimiliki NutriVision, dan memilih porsi default. Struktur ini mencegah satu makanan disimpan berulang hanya karena memiliki beberapa ukuran saji.

Data disimpan pada tabel *foods* dan *servings*. Tabel *foods* memuat nama, kategori, label citra, dan embedding nama, sedangkan *servings* menyimpan unit, jumlah, deskripsi porsi, nilai gizi, dan embedding porsi. Hubungan ini memungkinkan agent mencari nama makanan secara semantik, lalu memilih ukuran saji yang sesuai dengan satuan pada pesan pengguna. Indeks HNSW digunakan untuk mempercepat pencarian vektor tanpa memisahkan data dari transaksi PostgreSQL.

C. Arsitektur dan Konfigurasi Agent

Sistem terdiri atas aplikasi Android berbasis Kotlin, *backend* Flask, database PostgreSQL, dan layanan Groq. Flask menangani autentikasi, pengguna, makanan, *food log*, resep, serta sesi percakapan. LangChain Agent, yang mendukung orkestrasi LLM, retrieval, dan *tool*, menerima

system prompt, pesan pengguna, delapan giliran percakapan terakhir, dan identitas pengguna [22]. Agent kemudian menentukan apakah permintaan dapat dijawab langsung atau membutuhkan pemanggilan *tool*. Checkpointer PostgreSQL mempertahankan konteks lintas permintaan.

Dua model produksi Groq dibandingkan pada sepuluh skenario. GPT-OSS-120B memperoleh skor trajectory rata-rata 0,875, lebih tinggi daripada GPT-OSS-20B sebesar 0,842, sehingga dipilih sebagai model utama. Pengujian temperature 0,0-1,0 menunjukkan nilai 0,3 memberikan respons yang paling stabil, ringkas, dan natural. Konfigurasi akhir menggunakan GPT-OSS-120B dengan temperature 0,3.

TABEL I. PERBANDINGAN MODEL REASONING

Model	Rata-rata	Keputusan
GPT-OSS-20B	0,842	Tidak dipilih
GPT-OSS-120B	0,875	Model utama

Perbedaan skor paling terlihat pada operasi edit, hapus, dan penggunaan profil. Pada skenario perbandingan dua makanan lalu mencatat pilihan yang lebih rendah, kedua model masih memperoleh skor di bawah 0,5. Hasil tersebut menunjukkan bahwa tugas multi-langkah yang memerlukan pengambilan beberapa kandidat, perbandingan, dan transaksi lanjutan tetap menjadi tantangan meskipun model yang lebih besar digunakan.

TABEL II. TOOL PADA AGENTIC CHATBOT

Tool	Fungsi
get_today_food_logs	Membaca ringkasan asupan hari ini.
add_food_logs	Menambah satu atau beberapa makanan.
edit_food_logs	Mengubah catatan makanan.
delete_food_logs	Menghapus catatan makanan.
search_foods	Mencari makanan secara semantik.
get_food_servings	Mengambil variasi porsi.
get_user_data	Membaca profil dan target pengguna.
update_user_profile	Memperbarui profil dan target.

D. Prosedur Pengujian

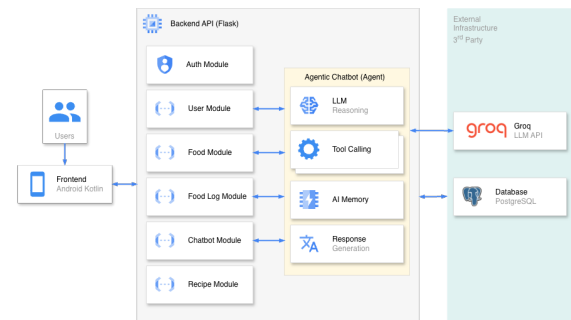
Evaluasi trajectory menggunakan 100 skenario yang mencakup pencarian gizi, operasi *food log*, ringkasan harian, pembaruan profil, konteks percakapan, dan pertanyaan di luar cakupan. Setiap rekaman *tool call* dan hasilnya dinilai oleh GPT-5.5 [19] pada lima aspek: kelogisan alur, progres menuju tujuan, ketepatan *tool*, efisiensi, dan kesesuaian respons akhir. Skor berada pada rentang 0-1 dan dinyatakan PASS apabila minimal 0,5, sejalan dengan penggunaan

LLM sebagai evaluator proses agent [13], [14]. Fungsionalitas diuji melalui 20 kasus *Black Box Testing* [15]. Usability diukur dengan sepuluh item SUS [16], [24], sedangkan kecenderungan rekomendasi diukur menggunakan NPS [17]. Pengujian pengguna melibatkan 16 responden berusia 20-25 tahun setelah mencoba fungsi utama aplikasi.

III. HASIL DAN PEMBAHASAN

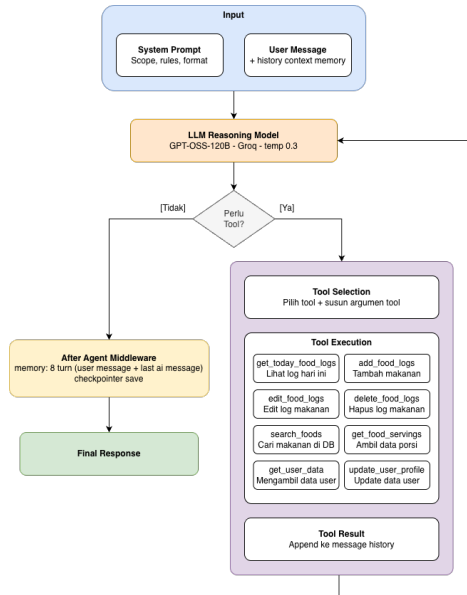
A. Implementasi Sistem

Arsitektur yang dihasilkan ditunjukkan pada Gambar 1. Pengguna berinteraksi melalui aplikasi Android, sedangkan Flask meneruskan pesan ke agent dan menyediakan akses terkontrol terhadap modul pengguna, makanan, *food log*, *chatbot*, dan resep. Agent menggabungkan penalaran LLM, pemanggilan *tool*, memori, dan pembentukan respons. Pemisahan modul ini mencegah model mengakses database secara langsung; seluruh operasi baca-tulis tetap mengikuti validasi layanan backend dan identitas pengguna.



Gambar 1. Arsitektur sistem NutriVision

Alur internal agent dirangkum pada Gambar 2. Model terlebih dahulu menilai kebutuhan *tool*. Jika diperlukan, agent memilih fungsi dan argumen, mengeksekusinya, serta menambahkan hasil ke histori pesan sebelum melakukan penalaran berikutnya. Siklus dapat berulang hingga informasi cukup. Setelah respons akhir dibentuk, middleware menyimpan pesan pengguna dan jawaban terakhir serta memangkas konteks menjadi delapan giliran agar biaya token tetap terkendali.



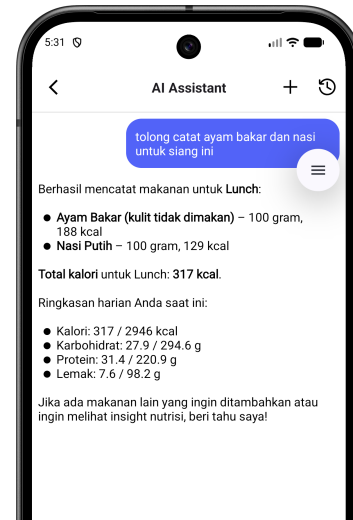
Gambar 2. Alur pemrosesan agentic chatbot

Saat menerima perintah, agent memadukan aturan pada *system prompt*, konteks pesan, dan profil pengguna. Untuk permintaan "catat ayam bakar dan nasi untuk siang", agent mencari kandidat makanan, menentukan porsi, menambahkan dua *food log*, lalu mengambil ringkasan harian untuk menyusun konfirmasi. Hasil operasi disimpan sebagai pesan *tool* sehingga respons akhir dapat dilacak ke data aktual, bukan perhitungan bebas model.

Pencarian semantik membantu menangani variasi bahasa pengguna, sedangkan tabel porsi menjaga konversi jumlah makanan ke kalori dan makronutrien. Ketika makanan tidak ditemukan, aturan agent melarang substitusi tanpa persetujuan. Kegagalan pada sebagian kecil skenario terutama terjadi saat model memilih makanan pengganti atau belum meminta klarifikasi pada input ambigu. Temuan ini menegaskan bahwa kualitas *agentic RAG* tidak hanya ditentukan oleh LLM, tetapi juga oleh desain *tool*, batasan *prompt*, dan kelengkapan basis data.

B. Integrasi Antarmuka Percakapan

Halaman percakapan menyediakan saran input, pengiriman pesan, indikator pemrosesan, pembuatan sesi baru, dan akses histori. Respons dapat memuat hasil pencatatan per item, total kalori pada waktu makan, serta perbandingan konsumsi harian dengan target pengguna. Gambar 3 memperlihatkan contoh pencatatan ayam bakar dan nasi yang menghasilkan total 317 kkal beserta ringkasan kalori dan makronutrien harian.



Gambar 3. Respons pencatatan makanan pada aplikasi

Integrasi ini memperluas aplikasi pencatatan kalori konvensional dari navigasi berbasis formulir menjadi operasi berbasis tujuan. Pengguna tidak perlu memilih makanan, porsi, dan waktu makan pada beberapa layar; informasi tersebut dapat disampaikan dalam satu pesan. Pendekatan tersebut sejalan dengan sistem nutrisi berbasis RAG [9], [10] dan implementasi chatbot RAG pada domain layanan informasi [25], tetapi NutriVision menambahkan transaksi *food log* dan pembaruan profil melalui agent, bukan hanya rekomendasi atau analisis.

C. Hasil Pengujian

Ringkasan hasil pengujian ditampilkan pada Tabel II. Penilaian 100 trajectory menghasilkan rata-rata 0,90, median 0,94, standar deviasi 0,13, nilai minimum 0,32, dan maksimum 1,00. Sebanyak 95 skenario mencapai ambang 0,5. Skor tinggi menunjukkan bahwa agent umumnya memilih *tool* yang tepat dan menyusun jawaban sesuai hasil eksekusi. Lima skenario gagal memperlihatkan kebutuhan perbaikan pada klarifikasi input, penanganan data tidak ditemukan, dan keputusan multi-langkah.

TABEL III. RINGKASAN HASIL PENGUJIAN

Metode	Sampel	Hasil Utama
LLM-as-a-Judge	100	Mean 0,90; pass rate 95%
Black Box	20	Seluruh skenario PASS (100%)
SUS	16	80,63; Acceptable; grade B
NPS	16	62,5; kategori Excellent

Seluruh 20 skenario *Black Box Testing* berstatus PASS. Fungsi membuka halaman, mencegah pesan kosong, mencari makanan, menangani data tidak ditemukan, mencatat

beberapa item, menampilkan ringkasan, mengubah dan menghapus catatan, memperbarui profil, melanjutkan konteks, serta mengelola histori berjalan sesuai spesifikasi. Hasil ini menunjukkan pemisahan antara agent dan layanan backend tidak mengurangi reliabilitas fungsi aplikasi.

Skor SUS 80,63 berada pada kategori *Acceptable* dengan grade B dan mendekati adjective rating *Excellent*. Nilai ini menunjukkan alur percakapan cukup mudah dipelajari, meskipun variasi skor antarresponden mengindikasikan perlunya contoh perintah dan panduan singkat. NPS 62,5 terdiri atas 62,5% *Promoter*, 37,5% *Passive*, dan tanpa *Detractor*. Kombinasi keduanya menunjukkan sistem tidak hanya berfungsi, tetapi juga memiliki penerimaan dan kecenderungan rekomendasi yang tinggi.

Analisis per kategori menunjukkan fungsi pencarian gizi dan pencatatan satu item cenderung stabil karena alurnya pendek. Variasi skor lebih besar muncul pada perintah multi-item, pengubahan catatan terdahulu, dan penggunaan referensi seperti "yang tadi" atau "yang lebih rendah". Pada kondisi tersebut, agent harus mempertahankan konteks, mengidentifikasi catatan yang tepat, dan menghindari eksekusi ganda. Checkpointer membantu mempertahankan konteks, tetapi aturan klarifikasi masih perlu diperketat ketika terdapat lebih dari satu catatan yang mungkin dimaksud.

Hasil pengguna juga perlu dibaca bersama karakteristik sampel. Responden berusia 20-25 tahun relatif terbiasa dengan aplikasi percakapan sehingga skor usability yang tinggi belum tentu langsung berlaku pada kelompok usia lanjut. Meski demikian, tidak adanya *Detractor* dan keberhasilan seluruh fungsi utama menunjukkan desain antarmuka, latensi, serta format konfirmasi telah memenuhi ekspektasi dasar pada kelompok uji.

Secara keseluruhan, hasil mendukung temuan bahwa RAG dapat meningkatkan keandalan informasi pada domain nutrisi [9], [10] dan bahwa agen dengan *tool* serta memori memungkinkan personalisasi yang lebih kuat [12]. Kontribusi utama penelitian adalah penerapan konsep tersebut pada operasi pencatatan kalori berbahasa Indonesia dengan evaluasi trajectory. Namun, hasil belum membuktikan akurasi klinis jangka panjang. Basis data 769 makanan masih dapat gagal mengenali menu sangat spesifik, LLM-as-a-Judge berpotensi membawa bias evaluator [13], dan pengujian usability melibatkan kelompok usia yang relatif homogen.

D. Analisis Kegagalan dan Implikasi

Lima trajectory yang berada di bawah ambang kelulusan tidak menunjukkan satu jenis

kesalahan yang sama. Pada kasus makanan tidak tersedia, agent pernah memilih kandidat yang dianggap mirip dan meneruskannya sebagai hasil. Perilaku tersebut meningkatkan peluang pencatatan data yang salah meskipun respons terlihat lancar. Pada skenario multi-langkah seperti membandingkan dua makanan lalu mencatat pilihan berkalori lebih rendah, agent juga dapat berhenti setelah proses pencarian atau menyusun argumen transaksi yang tidak konsisten. Kedua pola tersebut memperlihatkan bahwa keberhasilan retrieval tidak otomatis menjamin keberhasilan keputusan dan aksi berikutnya.

Input ambigu menjadi sumber kegagalan lain. Pernyataan seperti "saya makan banyak tadi" tidak menyediakan jenis makanan, jumlah, maupun waktu makan yang cukup. Respons terbaik seharusnya meminta klarifikasi, bukan menebak. Demikian pula, perintah edit dengan rujukan "yang tadi" dapat mengacu pada lebih dari satu catatan. Karena itu, kebijakan agent perlu menetapkan kondisi wajib klarifikasi apabila skor kemiripan kandidat rendah, lebih dari satu objek memenuhi konteks, atau satuan porsi tidak dapat dipetakan secara pasti.

Dari sisi rekayasa perangkat lunak, pemisahan LLM dan modul transaksi terbukti penting. Tool hanya menerima skema input yang tervalidasi dan menggunakan *user_id* dari sesi autentikasi, sehingga model tidak dapat mengubah data pengguna lain. Akan tetapi, validasi struktural belum selalu menjamin validitas semantik. Nama makanan yang benar tetapi porsi yang tidak sesuai tetap dapat menghasilkan nilai gizi yang keliru. Lapisan verifikasi tambahan dapat membandingkan satuan, rentang jumlah, dan porsi default sebelum transaksi dikomit.

Evaluasi trajectory memberikan informasi diagnostik yang tidak diperoleh dari pengujian output akhir. Skor dapat menurun ketika agent menggunakan tool berlebih, mengulangi pencarian, atau menghasilkan jawaban yang tidak sesuai dengan hasil eksekusi, walaupun tujuan akhirnya tercapai. Hal ini mendukung pentingnya evaluasi proses untuk sistem agentic [13], [14]. Namun, keputusan akhir tetap perlu dilengkapi peninjauan manusia karena evaluator LLM dapat memiliki bias terhadap panjang, gaya bahasa, atau struktur respons.

TABEL IV. TEMUAN KEGAGALAN DAN ARAH PERBAIKAN

Temuan	Arah Perbaikan
Makanan tidak ditemukan tetapi diganti kandidat lain.	Larangan substitusi dan konfirmasi eksplisit.

Temuan	Arah Perbaikan
Perintah perbandingan dan transaksi tidak tuntas.	Pisahkan tahap retrieval, verifikasi, dan commit.
Rujukan konteks atau porsi ambigu.	Klarifikasi wajib sebelum tool tulis dipanggil.
Tool dipanggil berulang tanpa informasi baru.	Batas iterasi dan pemeriksaan kecukupan bukti.

Implikasi praktis penelitian adalah bahwa *chatbot* pencatat kalori tidak cukup dibangun sebagai lapisan percakapan di atas aplikasi. Sistem memerlukan basis data yang dapat ditelusuri, tool dengan kontrak jelas, kontrol akses, memori yang dibatasi, dan aturan penolakan atau klarifikasi. Arsitektur tersebut juga memudahkan perluasan fungsi, misalnya rekomendasi berbasis sisa target kalori, pencatatan resep, atau integrasi hasil deteksi citra, tanpa memberi LLM akses langsung ke operasi database.

IV. KESIMPULAN DAN SARAN

Penelitian berhasil mengintegrasikan *agentic retrieval-augmented generation* pada NutriVision melalui Android Kotlin, Flask, PostgreSQL, LangChain Agent, GPT-OSS-120B, delapan *tool*, dan memori percakapan. Basis data 769 makanan Indonesia memungkinkan agent mencari informasi gizi dan menjalankan operasi *food log* menggunakan bahasa alami. Evaluasi menunjukkan mean trajectory 0,90 dengan pass rate 95%, keberhasilan fungsional 100%, SUS 80,63, dan NPS 62,5. Hasil tersebut menunjukkan sistem telah memenuhi tujuan fungsional dan diterima dengan baik oleh pengguna.

Pengembangan berikutnya perlu memperluas basis data makanan dan sinonim lokal, menambahkan mekanisme klarifikasi wajib untuk input ambigu, serta mengevaluasi akurasi terhadap penilaian ahli gizi. Pengujian pada kelompok usia dan latar belakang yang lebih beragam juga diperlukan. Untuk mengurangi bias evaluasi, trajectory sebaiknya dinilai menggunakan beberapa evaluator LLM dan sampel penilaian manusia.

REFERENSI

- [1] World Health Organization, "Obesity and overweight," Dec. 8, 2025.
- [2] Kementerian Kesehatan Republik Indonesia, "Survei Kesehatan Indonesia Tahun 2023," 2023.
- [3] M. Raber et al., "A systematic review of the use of dietary self-monitoring in behavioural weight loss interventions," *Public Health Nutrition*, vol. 24, no. 17, pp. 5885-5913, 2021, doi: 10.1017/S136898002100358X.
- [4] K. Dugas et al., "Calorie-counting apps for monitoring and managing calorie intake in adults living with weight-related chronic diseases," *JMIR mHealth and uHealth*, vol. 14, e64139, 2026, doi: 10.2196/64139.
- [5] B. Singh et al., "Systematic review and meta-analysis of the effectiveness of chatbots on lifestyle behaviours," *npj Digital Medicine*, vol. 6, no. 1, 2023, doi: 10.1038/s41746-023-00856-1.

- [6] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," *arXiv:2005.11401*, 2021.
- [7] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," *arXiv:2312.10997*, 2024.
- [8] L. Wang et al., "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 41, 2024, doi: 10.1007/s11704-024-40231-1.
- [9] A. K. Gavai and J. van Hilleegersberg, "AI-driven personalized nutrition: RAG-based digital health solution for obesity and type 2 diabetes," *PLOS Digital Health*, vol. 4, no. 5, 2025, doi: 10.1371/journal.pdig.0000758.
- [10] Z. Jiang et al., "DietGlance: Dietary monitoring and personalized analysis at a glance with knowledge-empowered AI assistant," *arXiv:2502.01317*, 2026.
- [11] Z. Yang et al., "ChatDiet: Empowering personalized nutrition-oriented food recommender chatbots through an LLM-augmented framework," *Smart Health*, 2024, doi: 10.1016/j.smhl.2024.100465.
- [12] M. Abbasian, I. Azimi, A. M. Rahmani, and R. Jain, "Conversational health agents: A personalized LLM-powered agent framework," *arXiv:2310.02374*, 2024.
- [13] J. Gu et al., "A survey on LLM-as-a-Judge," *arXiv:2411.15594*, 2025.
- [14] M. Mohammadi, Y. Li, J. Lo, and W. Yip, "Evaluation and benchmarking of LLM agents: A survey," *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, vol. 2, pp. 6129-6139, 2025, doi: 10.1145/3711896.3736570.
- [15] G. S. Mahendra and I. K. A. Asmarajaya, "Evaluation using Black Box Testing and System Usability Scale in the Kidung Sekar Madya application," *Sinkron*, vol. 6, no. 4, pp. 2292-2302, 2022, doi: 10.33395/sinkron.v7i4.11755.
- [16] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," *Journal of Usability Studies*, vol. 4, no. 3, pp. 114-123, 2009.
- [17] S. Baehre, M. O'Dwyer, L. O'Malley, and N. Lee, "The use of Net Promoter Score to predict sales growth," *Journal of the Academy of Marketing Science*, 2021, doi: 10.1007/s11747-021-00790-2.
- [18] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York: McGraw-Hill Education, 2020.
- [19] OpenAI, "Introducing GPT-5.5," 2026.
- [20] M. A. K. Raiaan et al., "A review on large language models: Architectures, applications, taxonomies, open issues and challenges," *IEEE Access*, vol. 12, pp. 26839-26874, 2024, doi: 10.1109/ACCESS.2024.3365742.
- [21] F. Neha and D. Bhati, "Traditional RAG vs. Agentic RAG: A comparative study of retrieval-augmented systems," *TechRxiv*, 2025, doi: 10.36227/techrxiv.175624551.12254549/v1.
- [22] S. Vidivelli, M. Ramachandran, and A. Dharunbalaji, "Efficiency-driven custom chatbot development: Unleashing LangChain, RAG, and performance-optimized LLM fusion," *Computers, Materials and Continua*, vol. 80, no. 2, pp. 2423-2442, 2024, doi: 10.32604/cmc.2024.054360.
- [23] L. Chikwetu, S. Daily, B. J. Mortazavi, and J. Dunn, "Automated diet capture using voice alerts and speech recognition on smartphones," *JMIR Formative Research*, vol. 7, 2023, doi: 10.2196/46659.
- [24] N. Pratama, R. Anrahvi, and Stevani, "Penerapan metode System Usability Scale dalam mengukur kepuasan mahasiswa terhadap website direktori akademik," *IJBEM*, vol. 3, no. 2, pp. 74-80, 2024, doi: 10.57152/ijbem.v3i2.2024.
- [25] L. S. Hartono, E. I. Setiawan, and V. Singh, "Retrieval augmented generation-based chatbot for prospective and current university students," *International Journal of Engineering, Science and Information Technology*, vol. 5, no. 3, pp. 268-277, 2025, doi: 10.52088/ijesty.v5i3.951.