

BAB 5. PENUTUP

5.1. Kesimpulan

Penelitian ini bertujuan merancang, mengimplementasikan, dan mengevaluasi sistem berbasis LLM-RAG untuk *automated document generation and validation* pada proposal Program Kreativitas Mahasiswa di FIK UPNVJ. Kesimpulan disusun berdasarkan tiga tahap *Rapid Application Development* (RAD) yang menjadi metodologi pengembangan sistem, yaitu perancangan, implementasi, dan evaluasi.

1. Perancangan Sistem

Peneliti merancang arsitektur sistem menggunakan pendekatan *Service Based Architecture* (SBA) dengan empat layanan, yaitu *frontend service* berbasis Next.js menangani antarmuka pengguna, *web backend service* berbasis Express.js bertindak sebagai *orchestrator* yang memverifikasi sesi dan peran pengguna sebelum meneruskan permintaan, *AI service* berbasis FastAPI dan Python menjembatani lapisan web dengan lapisan kecerdasan buatan Layanan AI untuk menjalankan pipeline LLM-RAG melalui LangChain. Layanan *database* menggunakan Supabase untuk tiga jenis penyimpanan sekaligus: data relasional, dokumen mentah, dan data vektor.

Peneliti merancang dua pipeline inti di atas arsitektur tersebut. Pipeline *automated document generation* mengekstrak teks Buku Panduan PKM menggunakan *PyMuPDF4LLM*, memecahnya menjadi *chunk* dengan *MarkdownTextSplitter*, mengubahnya menjadi *embedding* melalui *gemini-embedding-001*, lalu menyimpannya di Supabase sebagai basis pencarian semantik. Model *llama-3.3-70b-versatile* via Groq API mengolah hasil pencarian menjadi parameter format terstruktur, dengan Google Gemini sebagai model cadangan saat *rate limit* tercapai. Pipeline *automated document validation* bekerja pada arah sebaliknya: sistem mengambil parameter format tersimpan sebagai *ground truth*, mengekstrak properti format dari dokumen .docx yang diunggah *reviewer*, lalu membandingkan keduanya untuk mendeteksi ketidaksesuaian pada lima kategori (tipografi, tata letak halaman, spasi dan indentasi, struktur kelengkapan dokumen, penomoran halaman).

Kedua pipeline menyematkan *Human in The Loop* pada titik krusial. Admin meninjau hasil ekstraksi sebelum sistem menyimpannya sebagai *ground truth*. *Reviewer* menilai laporan validasi dan menetapkan status akhir proposal, karena laporan sistem bersifat asistensi, bukan keputusan otomatis.

Tahap perancangan ini menjawab aspek desain pada rumusan masalah pertama, yaitu bagaimana merancang arsitektur sistem berbasis LLM-RAG yang mampu mengekstrak parameter format dari Buku Panduan PKM sebagai basis *automated document generation and validation*.

2. Implementasi Sistem

Peneliti merealisasikan rancangan menjadi sistem *web* dengan tiga aktor: Admin, *Reviewer*, dan LLM sebagai layanan yang diintegrasikan lewat Groq API, bukan dibangun dari nol. Admin mengelola delapan fungsi utama, mencakup pengaturan periode *review*, manajemen akun *reviewer*, dan penugasan proposal. *Reviewer*

mengakses proposal yang ditugaskan, menjalankan validasi, memberi umpan balik tertulis, dan mengubah status kelulusan proposal.

Pada level kode, peneliti menangani tantangan utama selama implementasi, yaitu mengarahkan LLM agar menghasilkan keluaran dalam format terstruktur menggunakan metode *with_structured_output()*, skema Pydantic, bukan teks bebas, sehingga hasil ekstraksi parameter dapat langsung diproses oleh sistem tanpa risiko kesalahan *parsing*. Peneliti juga menerapkan tiga tingkat kunci akses Supabase sesuai kebutuhan tiap layanan: *anon key* untuk *frontend service*, kombinasi *anon* dan *service role key* untuk *web backend service*, dan *service role key* penuh untuk layanan AI.

Tahap implementasi ini melengkapi jawaban atas rumusan masalah pertama pada aspek realisasi teknis, sekaligus memenuhi tujuan penelitian pertama, yaitu merancang arsitektur sistem dan mengimplementasikan sistem berbasis LLM-RAG yang mampu mengekstrak parameter format dari Buku Panduan PKM sebagai basis *automated document generation and validation* pada dokumen proposal PKM di UPNVJ.

3. Evaluasi Sistem

Peneliti mengevaluasi sistem melalui tiga metode dengan cakupan berbeda. Pengujian *black box*, empat responden menguji tujuh skenario fungsional yang mewakili seluruh fitur MVP, mencakup autentikasi, manajemen data, pipeline *generation*, dan pipeline *validation* baik tunggal maupun massal. Seluruh tujuh skenario mencatat hasil berhasil (Tabel 4.14).

User Acceptance Testing, empat pengguna, yaitu Kepala UPA PKK, Admin UPA PKK, dan dua *Reviewer*, mengisi instrumen PSSUQ setelah menjalankan skenario penggunaan. Skor *overall* rata-rata mencapai 1,89 dari skala 1-7, masuk kategori Baik dengan tingkat kepuasan 85,2%. Ketua UPA PKK mencatat skor terbaik (1,75), sementara dua *Reviewer* mencatat skor 2,00. Dari 64 butir respons, dua butir tidak mencapai kategori "Setuju": P6 dari Admin soal kepercayaan diri menyelesaikan tugas dengan cepat, dan P7 dari *Reviewer* 1 soal kejelasan pesan kesalahan. Kedua butir ini menunjuk area perbaikan konkret untuk iterasi berikutnya.

Evaluasi kualitas pipeline RAG, kerangka RAGAS mengukur empat metrik pada pipeline *generation*. *Faithfulness* mencapai nilai sempurna 1,000, menandakan seluruh klaim jawaban model bersumber langsung dari *chunk* di basis data vektor tanpa halusinasi. *Answer relevancy* mencapai 0,9186 dan *context recall* mencapai 0,900. *Context precision* mencatat nilai terendah di antara keempat metrik, yaitu 0,8175, yang menunjukkan sebagian *chunk* yang diambil sistem belum sepenuhnya relevan dengan jawaban yang dibutuhkan.

Ketiga metode evaluasi ini menjawab rumusan masalah kedua, yaitu bagaimana hasil evaluasi sistem berdasarkan pengujian *black box testing* dan *User Acceptance Testing* terhadap fungsionalitas dan penerimaan pengguna. Hasil evaluasi sekaligus memenuhi tujuan penelitian kedua, yaitu mengevaluasi sistem melalui *black box testing* dan UAT untuk mengukur fungsionalitas dan tingkat penerimaan pengguna terhadap sistem yang dikembangkan.

5.2. Saran

Penelitian ini telah berhasil membangun sistem *automated document generation* dan validasi dokumen PKM berbasis pipeline RAG dan LLM secara *end-to-end*. Meski demikian, masih terdapat sejumlah aspek yang belum terjangkau dalam lingkup penelitian ini dan berpotensi untuk dikembangkan lebih lanjut. Berikut beberapa saran yang dapat menjadi arah pengembangan bagi peneliti selanjutnya.

1. Perluasan Validasi ke Format PDF

Sistem validasi saat ini hanya mendukung dokumen berformat DOCX. Peneliti selanjutnya dapat mengembangkan kemampuan validasi terhadap dokumen PDF, mengingat banyak mahasiswa mengumpulkan proposal PKM dalam format tersebut. Pengembangan ini memerlukan pustaka *parsing* PDF yang mampu mempertahankan informasi *layout* dan tipografi dokumen.

2. Penerapan Teknik *Re-Ranking* pada Proses *Retrieval*

Nilai *Context Precision* (0,8175) pada evaluasi RAGAS menunjukkan bahwa sebagian *chunk* yang diambil sistem belum sepenuhnya relevan. Peneliti selanjutnya dapat menerapkan *re-ranking* menggunakan model *cross-encoder* untuk menyaring dan mengurutkan ulang kandidat *chunk* sebelum dikirim ke LLM, sehingga konteks yang diproses lebih tepat sasaran dan skor presisi meningkat.

3. Penggunaan Model LLM Lokal

Seluruh pipeline saat ini bergantung pada layanan API berbasis *cloud* (Groq dan Google Generative AI) yang memiliki keterbatasan berupa biaya operasional, *rate limit*, dan ketergantungan pada konektivitas internet. Peneliti selanjutnya dapat mengevaluasi penggunaan model LLM *open-source* yang dijalankan secara lokal, misalnya melalui Ollama, sebagai alternatif yang lebih mandiri untuk lingkungan institusi pendidikan.

4. Ekstraksi Informasi yang Lebih Dinamis dan Adaptif

Proses ekstraksi metadata dalam sistem ini bergantung pada skema Pydantic yang didefinisikan secara statis (*rule-based structured output*), sehingga *field* yang dapat diekstrak terbatas pada apa yang sudah ditentukan pengembang. Peneliti selanjutnya dapat mengeksplorasi pendekatan ekstraksi yang lebih dinamis, misalnya dengan membiarkan LLM mengidentifikasi dan mengekstrak aturan format tanpa skema tetap, lalu memetakannya ke struktur yang diperlukan secara otomatis. Pendekatan ini memungkinkan sistem beradaptasi terhadap aturan baru tanpa perlu mengubah kode.

5. Peningkatan Pengalaman Pengguna

Peneliti selanjutnya dapat menggali kebutuhan pengguna secara lebih spesifik pada dua konteks, yaitu tahap mana dalam alur otomatisasi yang membutuhkan indikator progres tambahan, dan informasi apa yang dibutuhkan *Reviewer* saat menerima laporan ketidaksesuaian format, melalui wawancara lanjutan atau pengujian *usability* tambahan. Dua konteks ini mengacu kepada kuesioner PSSUQ terkait keyakinan pengguna untuk menjadi produktif dengan cepat dan kejelasan pesan kesalahan yang ditampilkan sistem.