

## BAB 5. PENUTUP

### 5.1. Kesimpulan

Berdasarkan penelitian Implementasi *Intelligent Tutoring System* Berbasis *Large Language Model* dan Visualisasi Dinamis pada Platform Pembelajaran Struktur Data, dapat disimpulkan bahwa sistem ini berhasil dirancang, diimplementasikan, dan dievaluasi dengan baik melalui tahapan *Requirements Planning*, *User Design*, *Construction*, dan *Cutover*. Sistem dibangun dengan arsitektur *service-oriented* berbasis Docker yang terdiri dari komponen-komponen terisolasi seperti *API Gateway*, *Auth Service*, *User Service*, *Quest Service*, dan *AI Service*. Interaksi *real-time* antara pengguna dan sistem diwujudkan melalui komunikasi *WebSocket* pada *Quest Service*, sementara kecerdasan sistem ditopang oleh dua peran LLM utama, yaitu *Director* yang menyusun skenario dan pemetaan visual secara dinamis, serta *Oracle* yang berperan sebagai tutor *Socratic* interaktif. Visualisasi dinamis sendiri dirancang melalui *Visual Engine* berbasis instruksi JSON yang di-*render* langsung di sisi klien, sehingga tampilan struktur data dapat berubah secara *real-time* sesuai konteks pembelajaran. Seluruh kebutuhan fungsional dan non-fungsional yang direncanakan berhasil diwujudkan, termasuk dukungan interaksi percakapan berbasis teks maupun suara melalui *Speech-to-Text* dan *Text-to-Speech*, serta komponen RAG yang menggunakan *Vector Database* sebagai mesin pencari semantik untuk memastikan jawaban *Oracle* selalu berlandaskan materi yang terverifikasi, sehingga meminimalkan risiko halusinasi AI. Seluruh sistem ini di-deploy menggunakan Docker Compose dengan infrastruktur pendukung seperti Redis untuk pemrosesan data *real-time*, PostgreSQL untuk penyimpanan data permanen, dan RabbitMQ untuk penanganan antrian tugas AI secara asinkron.

Evaluasi dilakukan melalui beberapa metode pengujian yang saling melengkapi. *Black Box Testing* membuktikan bahwa seluruh skenario uji fungsional berjalan tanpa galat, mengonfirmasi keberhasilan integrasi antara backend, visualisasi interaktif, dan layanan kecerdasan buatan. *User Acceptance Testing* menggunakan instrumen PSSUQ menunjukkan tingkat penerimaan pengguna yang sangat positif terhadap platform secara keseluruhan, baik dari kalangan mahasiswa maupun pakar industri. Evaluasi kinerja RAG dengan kerangka kerja Ragas menunjukkan bahwa sistem *retrieval Oracle* bekerja baik dalam menyajikan dokumen referensi yang relevan, sementara skor *Faithfulness* dan *Answer Relevancy* yang relatif rendah bukan menjadi indikasi kegagalan, melainkan justru membuktikan bahwa *Oracle* berhasil menjalankan perannya sebagai tutor *Socratic* yang sengaja membimbing pengguna melalui pertanyaan balik dan analogi, bukan memberikan jawaban langsung. Hal ini divalidasi melalui metrik khusus *tutoring* yang menunjukkan hasil yang baik pada aspek *Socratic*, analogi, dan *alignment* pembelajaran. Selain itu, pengujian performa *real-time* menunjukkan bahwa seluruh komponen interaktif sistem beroperasi dalam batas

latensi yang dapat diterima dan dipersepsikan positif oleh pengguna, baik pada sisi validasi aksi WebSocket, komunikasi antar layanan, maupun waktu respons Oracle. Dengan demikian, secara keseluruhan sistem Struktura terbukti layak dan efektif digunakan sebagai platform pembelajaran struktur data yang mampu mendukung interaksi *real-time* berbasis LLM-RAG sekaligus menyajikan visualisasi dinamis yang bermanfaat bagi proses belajar mahasiswa.

## 5.2. Saran

Berdasarkan hasil pengembangan dan evaluasi yang telah dilakukan, terdapat beberapa saran untuk penelitian dan pengembangan lebih lanjut:

1. Pengukuran Efektivitas Pembelajaran  
Penelitian lanjutan perlu melakukan studi *pre-test/post-test* untuk mengukur apakah penggunaan Struktura secara statistik meningkatkan kemampuan mahasiswa dalam menyelesaikan soal struktur data. Konsep studi yang direkomendasikan adalah membandingkan kelompok yang belajar menggunakan Struktura selama 4 - 6 minggu dengan kelompok kontrol yang belajar secara konvensional, menggunakan soal-soal standar yang setara dengan tes teknis rekrutmen sebagai instrumen evaluasi.
2. Perluasan Cakupan Kurikulum  
Sistem perlu dikembangkan untuk menjangkau lebih banyak variasi struktur data yang lebih kompleks, seperti *Graph*, *Hash Table*, atau variasi *Tree* lanjutan. Penambahan materi ini akan membuat ekosistem pembelajaran menjadi jauh lebih lengkap dan komprehensif bagi pengguna.
3. Eksplorasi Model Bahasa Lokal Berskala Lebih Besar  
Penelitian ini mengeliminasi model lokal (Llama 3.1 8B, Qwen2.5 14B) karena keterbatasan *hardware* pengembangan. Penelitian selanjutnya disarankan mengeksplorasi model lokal berskala lebih besar (70B parameter atau lebih) yang dijalankan pada *server GPU dedicated*, guna mengevaluasi apakah model lokal dapat menjadi alternatif yang kompetitif dan bebas ketergantungan API berbayar.
4. Pengembangan *Dashboard* Administrator  
Untuk memudahkan pengelolaan jangka panjang, sangat krusial untuk membangun antarmuka khusus admin (Admin UI) yang memungkinkan pengajar menambah materi belajar baru, memantau kemajuan mahasiswa, dan mengelola basis pengetahuan RAG tanpa harus bersentuhan langsung dengan kode di backend.
5. Pengujian Ketahanan Beban (*Load Testing*)  
Penelitian ini belum menguji perilaku sistem ketika diakses oleh banyak pengguna secara bersamaan. Penelitian selanjutnya perlu melakukan *load testing* untuk mengidentifikasi kapasitas maksimum pengguna konkuren yang dapat ditangani sistem, serta menentukan strategi skalabilitas infrastruktur yang tepat.

6. Pengembangan LLM mandiri (*self-hosted LLM*).

Ketergantungan sistem Struktura pada API LLM eksternal dari Google merupakan titik risiko yang perlu dimitigasi dalam jangka panjang, baik dari sisi ketersediaan layanan, biaya operasional yang meningkat seiring bertambahnya pengguna, maupun aspek privasi data mahasiswa yang dikirimkan ke *server* pihak ketiga. Penelitian lanjutan perlu mengeksplorasi pengembangan dan *fine-tuning* model bahasa besar yang dapat dijalankan secara mandiri (*self-hosted*). Arah yang direkomendasikan adalah *fine-tuning* model berukuran 7B hingga 13B parameter seperti LLaMA pada dataset yang terdiri dari pasangan pertanyaan-respons *Socratic* tentang struktur data yang dapat dikurasi dari log interaksi Oracle selama sistem beroperasi. Dengan dataset yang mencukupi dan teknik *fine-tuning* seperti LoRA (*Low-Rank Adaptation*) yang efisien dalam penggunaan memori GPU, model mandiri berpotensi menghasilkan performa yang kompetitif.