# BAB II
# TINJAUAN PUSTAKA

## 2.1 Pariwisata dan Sistem Informasi

Pariwisata merupakan aktivitas perjalanan untuk tujuan rekreasi dan pengembangan pribadi dalam jangka waktu sementara. Menurut UU RI No. 10 Tahun 2009, pariwisata adalah berbagai kegiatan wisata yang didukung fasilitas dan layanan dari masyarakat, pengusaha, dan pemerintah. Sektor pariwisata berperan penting dalam perekonomian daerah melalui peningkatan pendapatan, penciptaan lapangan kerja, dan pelestarian budaya (Indonesia, 2009).

Sistem informasi pariwisata modern telah berkembang dari platform statis menjadi interaktif yang memanfaatkan kecerdasan buatan. Tren saat ini mencakup personalisasi berdasarkan preferensi, informasi *real-time*, pemanfaatan *user-generated content* (ulasan dan rating), serta *conversational interface* seperti *chatbot*. Ulasan pengunjung mengandung informasi berharga tentang pengalaman nyata, namun volume data yang besar dan tidak terstruktur menyulitkan pengguna menemukan informasi spesifik secara cepat (Sunaryo, 2024).

## 2.2 Api Abadi Mrapen sebagai Studi Kasus

Api Abadi Mrapen adalah fenomena alam unik di Kecamatan Godong, Kabupaten Grobogan, Jawa Tengah, berupa api yang menyala terus-menerus akibat rembesan gas metana dari dalam tanah. Meskipun memiliki potensi wisata menarik, informasi mengenai lokasi ini masih terbatas dan belum terorganisir dengan baik. Calon pengunjung mengalami kesulitan memperoleh informasi komprehensif mengenai fasilitas, aksesibilitas, dan tips praktis berkunjung. Penelitian ini memanfaatkan ulasan pengunjung dari platform digital sebagai sumber pengetahuan utama untuk membangun sistem tanya jawab yang memberikan informasi akurat berbasis pengalaman nyata, sehingga meningkatkan aksesibilitas informasi dan mendorong kunjungan wisata (Hidayah, 2023).

## 2.3 Natural Language Processing dan Large Language Models

*Natural Language Processing* (NLP) adalah cabang AI yang memungkinkan komputer memahami, menginterpretasi, dan menghasilkan bahasa manusia. Komponen utama NLP meliputi *tokenization*, *part-of-speech tagging*, *named entity recognition*, dan *semantic analysis*. Perkembangan NLP modern ditandai dengan penggunaan *deep learning* yang mampu memahami konteks bahasa lebih baik (Vaswani, 2017).

*Large Language Models* (LLM) adalah model *neural network* berukuran besar yang dilatih pada dataset teks masif, seperti GPT dan BERT. LLM menggunakan arsitektur *transformer* dengan mekanisme atensi untuk memprediksi token berikutnya berdasarkan konteks. Meskipun mampu menghasilkan teks koheren, LLM memiliki keterbatasan berupa kecenderungan halusinasi ketika tidak memiliki pengetahuan yang cukup tentang suatu topik (Wang, 2025).

## 2.4 Question Answering System

*Question Answering System* (QA) adalah aplikasi yang menjawab pertanyaan secara otomatis dalam bahasa natural. Berbeda dengan mesin pencari yang mengembalikan daftar dokumen, sistem QA memberikan jawaban langsung dan spesifik. Jenis QA meliputi: *Extractive QA* yang mengekstrak jawaban dari teks sumber; *Generative QA* yang menghasilkan jawaban baru; *Open-domain QA* untuk berbagai topik; dan *Closed-domain QA* untuk domain spesifik (Labadze, 2023).

Sistem QA modern terdiri dari tiga komponen utama: *Question Processing* untuk menganalisis pertanyaan; *Information Retrieval* untuk mencari informasi relevan dari basis pengetahuan; dan *Answer Generation* untuk menghasilkan jawaban berdasarkan informasi yang ditemukan.

## 2.5 Retrieval-Augmented Generation (RAG)

RAG adalah paradigma yang menggabungkan *retrieval-based methods* dengan *generative models* untuk meningkatkan kualitas dan akurasi output. RAG dirancang mengatasi keterbatasan LLM dengan menambahkan konteks faktual dari sumber eksternal.

Metode RAG bekerja dalam tiga tahap: (1) **Retrieval** - mencari dokumen relevan dari *knowledge base* menggunakan *semantic search*; (2) **Augmentation** - menambahkan dokumen ke *context prompt*; (3) **Generation** - LLM menghasilkan jawaban berdasarkan kombinasi pertanyaan dan dokumen (Lewis P. P., 2020).

Keunggulan RAG meliputi: mengurangi halusinasi dengan *grounding* pada data faktual; *update* pengetahuan tanpa *retraining* model; *cost-effective* untuk pengetahuan *domain-specific*; dan memberikan transparansi melalui *source attribution*.

### 2.6 Embedding dan Vector Database

*Embedding* adalah representasi numerik teks dalam bentuk vektor berdimensi tinggi yang menangkap makna semantik. Model seperti Sentence-BERT, OpenAI *embeddings*, dan model multilingual mengkonversi teks menjadi vektor yang dapat dibandingkan secara matematis. Teks dengan makna serupa memiliki representasi vektor yang berdekatan dalam ruang vektor, memfasilitasi pencarian semantik berdasarkan kesamaan makna (Taipalus, 2024).

*Vector database* seperti Chroma, Pinecone, atau FAISS dirancang khusus untuk menyimpan dan mengindeks *embedding* guna memungkinkan *similarity search* yang cepat. Pencarian menggunakan metrik seperti *cosine similarity* atau *Euclidean distance* untuk mengukur kedekatan antar vektor, sehingga dapat mengidentifikasi dokumen paling relevan dengan pertanyaan pengguna.

### 2.7 LangChain dan LangGraph

LangChain adalah *framework open-source* untuk membangun aplikasi berbasis LLM dengan komponen modular. Komponen utama meliputi: *Document Loaders* untuk memuat berbagai format dokumen; *Text Splitters* untuk membagi dokumen menjadi *chunks* optimal; *Embeddings Interface* untuk model *embedding*; *Vector Stores* untuk integrasi *vector database*; *Retrievers* untuk mencari dokumen relevan; *Chains* untuk orkestrasi komponen; dan *Agents* untuk *decision-making.*

LangGraph adalah *library* untuk membangun aplikasi *stateful multi-actor* dengan LLM menggunakan pendekatan berbasis graf. Fitur utama meliputi: *State Management* untuk

mengelola konteks sepanjang percakapan; *Conditional Logic* untuk *routing* berbasis kondisi; *Human-in-the-Loop* untuk integrasi persetujuan manusia; *Parallel Execution* untuk menjalankan beberapa *nodes* secara paralel; dan *Cyclic Graphs* untuk mendukung iterasi dan *self-correction*. LangGraph sangat berguna untuk *multi-step reasoning* dan manajemen percakapan kompleks dalam sistem RAG (Bansal, 2024).

## 2.8 Text Preprocessing

*Preprocessing* adalah tahap penting untuk membersihkan dan menyiapkan data teks. Tahapan meliputi: *Cleaning* - menghapus *noise* seperti HTML *tags* dan karakter khusus; *Normalization* - *case folding*, *stemming*/*lemmatization*; *Tokenization* - memecah teks menjadi *tokens*; dan *Stopword Removal* - menghapus kata umum (opsional dalam RAG). Untuk bahasa Indonesia, *tools* seperti Sastrawi, NLTK, atau spaCy dapat digunakan (Dana, 2024).

## 2.9 Evaluasi Sistem RAG

### 2.9.1 *Framework* RAGAS

RAGAS (*Retrieval-Augmented Generation Assessment*) adalah *framework* evaluasi khusus untuk sistem RAG dengan metrik: *Context Precision* - ketepatan relevansi dokumen yang diambil; *Context Recall* - kelengkapan informasi yang terambil; *Faithfulness* - kesetiaan jawaban pada konteks; *Answer Relevancy* - relevansi jawaban terhadap pertanyaan; dan *Answer Semantic Similarity* - kemiripan semantik dengan jawaban referensi. RAGAS memastikan evaluasi objektif dan terstandarisasi (Es S. J.-A., 2023).

***Tabel 2. 1 Framework Evaluasi RAGAS***

| Aspek | Definisi | Target | Interpretasi Kualitatif |
|---|---|---|---|
| ***Context Precision*** | Ketepatan dokumen yang di-retrieve | > 0.70 | Sistem baik/kurang dalam mencari ulasan relevan |
| ***Context Recall*** | Kelengkapan informasi ter-retrieve | > 0.70 | Sistem lengkap/tidak dalam mengambil informasi |

| Aspek | Definisi | Target | Interpretasi Kualitatif |
|---|---|---|---|
| ***Faithfulness*** | Jawaban sesuai fakta (bukan halusinasi) | > 0.80 | Jawaban factual/mengandung halusinasi |
| ***Answer Relevancy*** | Jawaban menjawab pertanyaan | > 0.75 | Jawaban on-topic/melenceng |
| ***Answer Semantic Similarity*** | Kemiripan dengan jawaban ideal | > 0.80 | Jawaban berkualitas tinggi/rendah |
| **Answer Correctness** | Kebenaran faktual keseluruhan | > 0.75 | Jawaban benar & lengkap/kurang tepat |

### 2.9.2 Evaluasi Kualitatif (UAT)

Evaluasi kualitatif melalui *User Acceptance Testing* (UAT) menilai kepuasan pengguna terhadap: *Relevance* - kesesuaian jawaban dengan pertanyaan; *Accuracy* - ketepatan fakta informasi; *Helpfulness* - manfaat informasi dalam perencanaan kunjungan; dan *Ease of Use* - kemudahan berinteraksi dengan sistem. Penilaian menggunakan skala Likert (1-5) dan wawancara terstruktur untuk masukan deskriptif (Sugiyono, 2021).

### 2.9.3 Evaluasi Kuantitatif Kedekatan Semantik (*Cosine Similarity*)

Metrik *Cosine Similarity* digunakan untuk mengukur presisi teknis hasil pencarian (*retrieval*) secara objektif dengan menghitung kedekatan semantik antara vektor *embedding* pertanyaan pengguna dan dokumen ulasan (Manning, Raghavan, & Schütze, 2008). Persamaan matematisnya adalah:

$$\text{Similarity}(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

**Keterangan:**

1. $A$ *dan* $B$: Vektor representasi dari teks yang sedang dibandingkan (misalnya, pertanyaan pengguna dan dokumen basis data).
2. $n$ : Jumlah total dimensi atau fitur dalam vektor (misalnya, jumlah kosakata unik).
3. $A_i$ *dan* $B_i$ : Nilai bobot elemen ke-$i$ pada masing-masing vektor $A$ dan $B$.
4. $\sum(i=1)^n A_i B_i$: *Dot product* (perkalian titik) dari kedua vektor, yang nilainya membesar jika banyak kata yang beririsan.
5. $Similarity(A,B) = (\sum(i=1)^n A_i B_i) / \left( \sqrt{(\sum(i=1)^n A_i^2)} \sqrt{(\sum(i=1)^n B_i^2)} \right)$

   Hasil kali panjang (magnitudo) dari masing-masing vektor, yang berfungsi sebagai normalisasi agar perhitungan tidak bias terhadap panjang teks dokumen.

**Interpretasi Hasil:**

Hasil perhitungan *Cosine Similarity* pada teks berada pada rentang nilai **0 hingga 1**. Semakin nilainya mendekati angka **1**, maka kedua teks tersebut dianggap semakin mirip atau relevan. Sebaliknya, nilai yang mendekati **0** menunjukkan bahwa kedua teks tersebut tidak memiliki kemiripan makna atau kosakata.

Dalam pengembangan sistem RAG Api Abadi Mrapen ini, metrik tersebut berfungsi untuk:

1. **Akurasi *Retriever***: Memastikan ulasan yang diambil memiliki skor kemiripan tinggi dengan pertanyaan wisatawan.
2. **Validasi *Ground Truth***: Mengukur kemiripan jawaban sistem dengan referensi jawaban pakar.

Hasil perhitungan ini berperan sebagai instrumen pendukung dalam proses triangulasi data, yang nantinya akan dikonfrontasi dengan hasil *User Acceptance Testing* (UAT) untuk menghasilkan kesimpulan penelitian yang holistik.

## 2.10 Kerangka Teori

Kerangka teori penelitian ini mengintegrasikan konsep-konsep yang telah dijelaskan dalam alur kerja sistem tanya jawab pariwisata cerdas berbasis RAG:

1. ***Data Collection & Preprocessing****:* Pengumpulan ulasan dari Google Maps, pembersihan dan normalisasi teks, pemecahan dokumen menjadi *chunks* optimal menggunakan *RecursiveCharacterTextSplitter*.
2. ***Knowledge Base Creation****:* Konversi potongan teks menjadi vektor *embedding* menggunakan model seperti OpenAI *embeddings*, penyimpanan dalam *vector database* (Chroma/FAISS) menggunakan LangChain *VectorStore*.
3. ***Query Processing****:* Konversi pertanyaan pengguna menjadi vektor *embedding*, *similarity search* untuk menemukan *chunks* relevan, *retrieval* dokumen dengan skor kemiripan tertinggi (*top-k*).
4. ***Answer Generation*** **dengan LangGraph**: Inisialisasi *state* dengan pertanyaan dan dokumen, *conditional routing* untuk menentukan kebutuhan *retrieval* tambahan, *context augmentation*, pembangkitan jawaban oleh LLM, serta *post-processing* dan pemformatan.
5. ***Response Delivery***: Penyajian jawaban kepada pengguna, *source attribution* untuk verifikasi, dan pengumpulan *feedback* untuk perbaikan sistem.

Hubungan antar komponen: LangChain menyediakan komponen modular untuk implementasi *pipeline* RAG; LangGraph mengatur orkestrasi alur kerja kompleks; *Vector Database* mengelola basis pengetahuan; LLM menghasilkan jawaban natural; dan Ulasan Pengunjung sebagai sumber pengetahuan dinamis berbasis pengalaman nyata. Kerangka ini memungkinkan sistem memberikan jawaban yang relevan, kontekstual, dan *grounded* pada pengalaman pengunjung Api Abadi Mrapen (Ramlo, 2024).