

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Jenis Penelitian**

Penelitian ini merupakan penelitian terapan (*applied research*) dengan pendekatan **Research and Development (R&D)** menggunakan model **ADDIE** (Analysis, Design, Development, Implementation, Evaluation) yang dikemukakan oleh Model ADDIE dipilih karena sistematis dan cocok untuk pengembangan sistem berbasis teknologi informasi (Branch, 2009).

#### **Karakteristik penelitian:**

1. **Terapan:** Mengaplikasikan metode RAG untuk masalah nyata
2. **Pengembangan:** Menghasilkan sistem tanya jawab pariwisata
3. **Evaluatif:** Menilai kualitas sistem melalui evaluasi kualitatif dengan indikator objektif

#### **3.2 Waktu Penelitian**

1. **Objek penelitian:** Api Abadi Mrapen, Kabupaten Grobogan
2. **Pengumpulan data:** Platform Google Maps (online)
3. **Waktu:** Februari - Juli 2026 (6 bulan)

#### **3.3 Sumber Data**

##### **Data Primer:**

1. Ulasan pengunjung dari Google Maps (899)
2. Hasil User Acceptance Testing (UAT) dari 34 partisipan
3. Output sistem RAG (retrieved contexts dan generated answers)

##### **Data Sekunder:**

1. Literatur ilmiah (jurnal, buku metodologi)
2. Dokumentasi teknis (LangChain, LangGraph, LLM)
3. Informasi umum tentang Api Abadi Mrapen

#### **3.4 Teknik Pengumpulan Data**

##### **3.4.1 Web Scraping**

1. **Tools:** Python/Google Collab (Selenium/Google Places API)

2. **Target:** Minimal 500-2000 ulasan dengan rating, teks, tanggal
3. **Proses Cleaning Data:** Data mentah hasil scraping sejumlah 899 ulasan (kolom nama\_tempat, user, review, rating) diproses melalui auto-deteksi kolom, penghapusan data duplikat (deduplication), dan penyaringan panjang teks minimal 20 karakter, sehingga menghasilkan 643 ulasan valid yang digunakan sebagai basis pengetahuan sistem.

#### **3.4.2 User Acceptance Testing (UAT)**

1. Instrumen UAT terdiri dari 5 item pernyataan berskala Likert 1–5
2. Pertanyaan terbuka untuk feedback
3. Observasi selama penggunaan

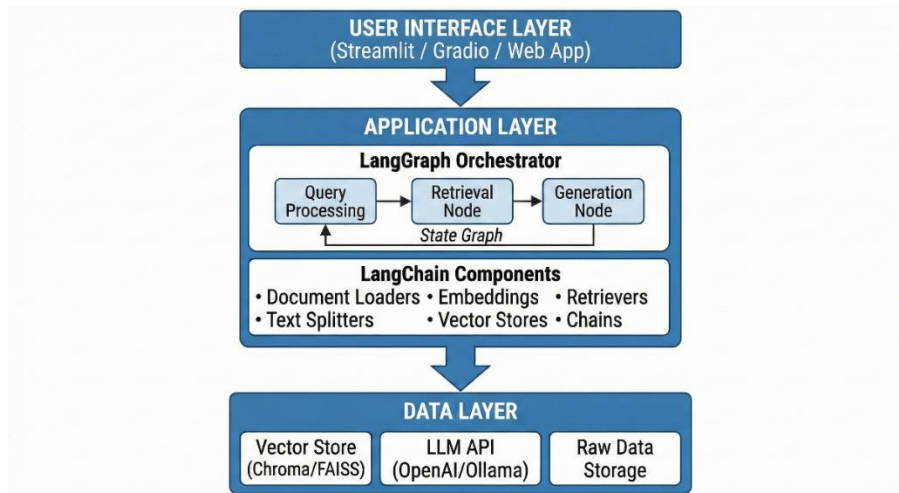
#### **3.4.3 Wawancara Pengelola**

Wawancara dilakukan secara langsung dengan pengelola objek wisata Api Abadi Mrapen (Bapak Annas Rofiqi, Penata Layanan Operasional), serta didukung oleh referensi resmi dari Pemerintah Kabupaten (Pemkab) Grobogan dan Pusat Vulkanologi dan Mitigasi Bencana Geologi (PVMBG) untuk memperoleh informasi otoritatif mengenai sejarah, fenomena alam, dan pengelolaan situs. Hasil wawancara disusun ke dalam 20 pasangan tanya-jawab yang mencakup 11 topik tematik, yaitu: sejarah, fenomena alam, tiket, jam buka, fasilitas, parkir, akses, daya tarik, program, pengelolaan, dan edukasi. Data ini digunakan sebagai data sekunder terstruktur yang melengkapi ulasan pengunjung dalam pembentukan basis pengetahuan sistem RAG.

#### **3.4.4 Text Preprocessing**

Sebelum diindeks ke dalam vector store, seluruh data ulasan valid melalui tahap text preprocessing yang meliputi: (1) case folding, yaitu penyeragaman huruf menjadi huruf kecil; (2) penghapusan URL; (3) penghapusan karakter non-alfanumerik; (4) normalisasi kata tidak baku atau slang menggunakan kamus khusus berisi 22 entri (contoh: gak → tidak, udah → sudah, yg → yang); serta (5) penyaringan ulasan yang bersifat duplikat atau berulang (deduplication). Tahapan ini bertujuan menghasilkan teks yang bersih dan konsisten sebelum proses embedding dilakukan.

### 3.5 Arsitektur Sistem



Gambar 3. 1 Arsitektur Sistem

#### 3.5.1. User Interface Layer (Lapisan Antarmuka)

Ini adalah lapisan teratas tempat pengguna berinteraksi langsung dengan sistem.

1. **Fungsi:** Menerima input pertanyaan (query) dari pengguna dan menampilkan jawaban akhir yang dihasilkan sistem.
2. **Contoh Tool:** Streamlit atau Gradio (sering digunakan untuk prototipe cepat berbasis Python) atau Web App umum.

#### 3.5.2. Application Layer (Lapisan Aplikasi)

Ini adalah "otak" dari sistem yang memproses logika bisnis. Terbagi menjadi dua bagian besar:

**A. LangGraph Orchestrator** Bagian ini mengatur alur kerja menggunakan konsep *State Graph* (grafik berstatus).

1. **Query Processing:** Menganalisis dan menyiapkan pertanyaan pengguna agar siap diproses.
2. **Retrieval Node:** Mengambil informasi yang relevan dari basis data berdasarkan pertanyaan.
3. **Generation Node:** Menggunakan informasi yang sudah diambil untuk menyusun jawaban melalui LLM.

4. **State Graph:** Garis panah yang berputar menunjukkan bahwa sistem ini bisa bersifat iteratif (misalnya, jika jawaban belum memuaskan, sistem bisa kembali memproses ulang query).

Implementasi teknis dari LangGraph Orchestrator ini terdiri atas 6 node kerja yang saling terhubung, yaitu: cek\_relevansi (memeriksa kesesuaian pertanyaan dengan domain wisata menggunakan pencocokan kata kunci), retrieve (mengambil top-5 dokumen relevan dari vector store), validasi\_konteks (mengevaluasi kecukupan konteks menggunakan cosine similarity dengan threshold 0,30), retrieve\_ulang (melakukan retrieval ulang dengan query yang diperluas apabila konteks belum memadai), generate (menyusun jawaban akhir menggunakan LLM), dan fallback (memberikan respons standar untuk pertanyaan di luar domain). Keenam node inilah yang merealisasikan konsep State Graph yang bersifat iteratif dan self-correcting.

**B. LangChain Components** Ini adalah "kotak peralatan" yang berisi modul-modul fungsional untuk mendukung operasional:

1. **Document Loaders:** Untuk membaca file (PDF, TXT, dll).
2. **Text Splitters:** Memecah dokumen besar menjadi potongan kecil (*chunks*).
3. **Embeddings:** Mengubah teks menjadi representasi angka (vektor).
4. **Vector Stores:** Tempat menyimpan data yang sudah di-embedding.
5. **Retrievers & Chains:** Algoritma untuk mencari data dan menghubungkan berbagai proses.

### 3.5.3. Data Layer (Lapisan Data)

Tempat penyimpanan dan sumber daya eksternal:

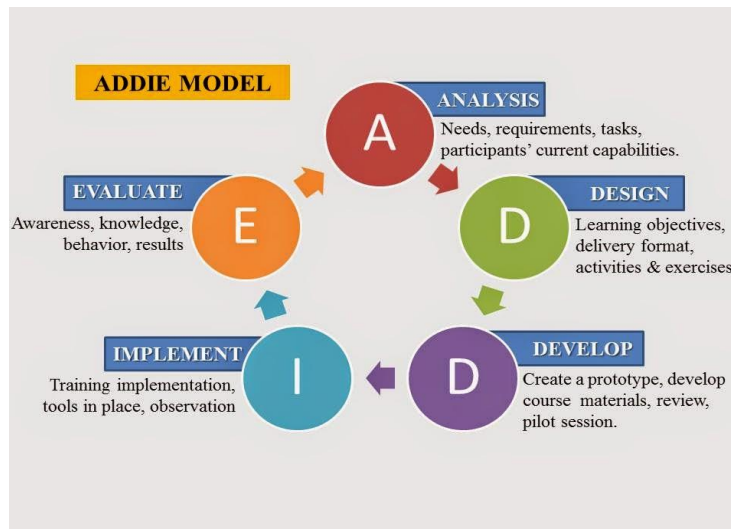
1. **Vector Store:** Database khusus (seperti Chroma atau FAISS) untuk mencari informasi relevan berdasarkan makna (embedding).
2. **LLM API:** Model bahasa besar (seperti OpenAI atau Ollama) yang berfungsi memproses bahasa alami.
3. **Raw Data Storage:** Tempat penyimpanan data mentah sebelum diproses.

Pada implementasi penelitian ini, LLM API yang digunakan adalah LLaMA-3.3-70B melalui Groq API (temperature=0,3; max\_tokens=1024) untuk proses generasi jawaban, serta model gpt-4o-mini dari OpenAI API (temperature=0) sebagai LLM-as-a-judge pada tahap evaluasi RAGAS. Vector Store diimplementasikan menggunakan FAISS dengan model embedding paraphrase-multilingual-MiniLM-L12-v2 (dimensi vektor 384). Proses pembentukan basis pengetahuan menggunakan RecursiveCharacterTextSplitter dengan chunk\_size=500 dan chunk\_overlap=100, serta metadata terstruktur (sumber, nama, rating, dan sentimen untuk data ulasan; sumber dan topik untuk data wawancara) pada setiap dokumen yang diindeks.

#### **3.5.4 Skema Perbandingan: Agentic RAG vs RAG Baseline**

Untuk mengetahui kontribusi mekanisme self-correction pada pipeline Agentic RAG, penelitian ini turut membangun sistem RAG Baseline sebagai pembandingan. RAG Baseline merupakan implementasi RAG sederhana yang hanya terdiri atas proses retrieve dan generate secara langsung, tanpa melalui node validasi\_konteks, retrieve\_ulang, maupun mekanisme grading lainnya yang terdapat pada Agentic RAG. Kedua sistem menggunakan basis pengetahuan (vector store FAISS) dan model LLM yang identik, sehingga perbedaan hasil evaluasi mencerminkan pengaruh mekanisme self-correction pada pipeline Agentic. Perbandingan ini digunakan pada tahap evaluasi RAGAS Diperkuat (30 test case) untuk menilai apakah kompleksitas tambahan pada Agentic RAG memberikan peningkatan kualitas jawaban dibandingkan pendekatan baseline yang lebih sederhana.

### 3.6 Model Pengembangan ADDIE



Gambar 3. 2 ADDIE Model

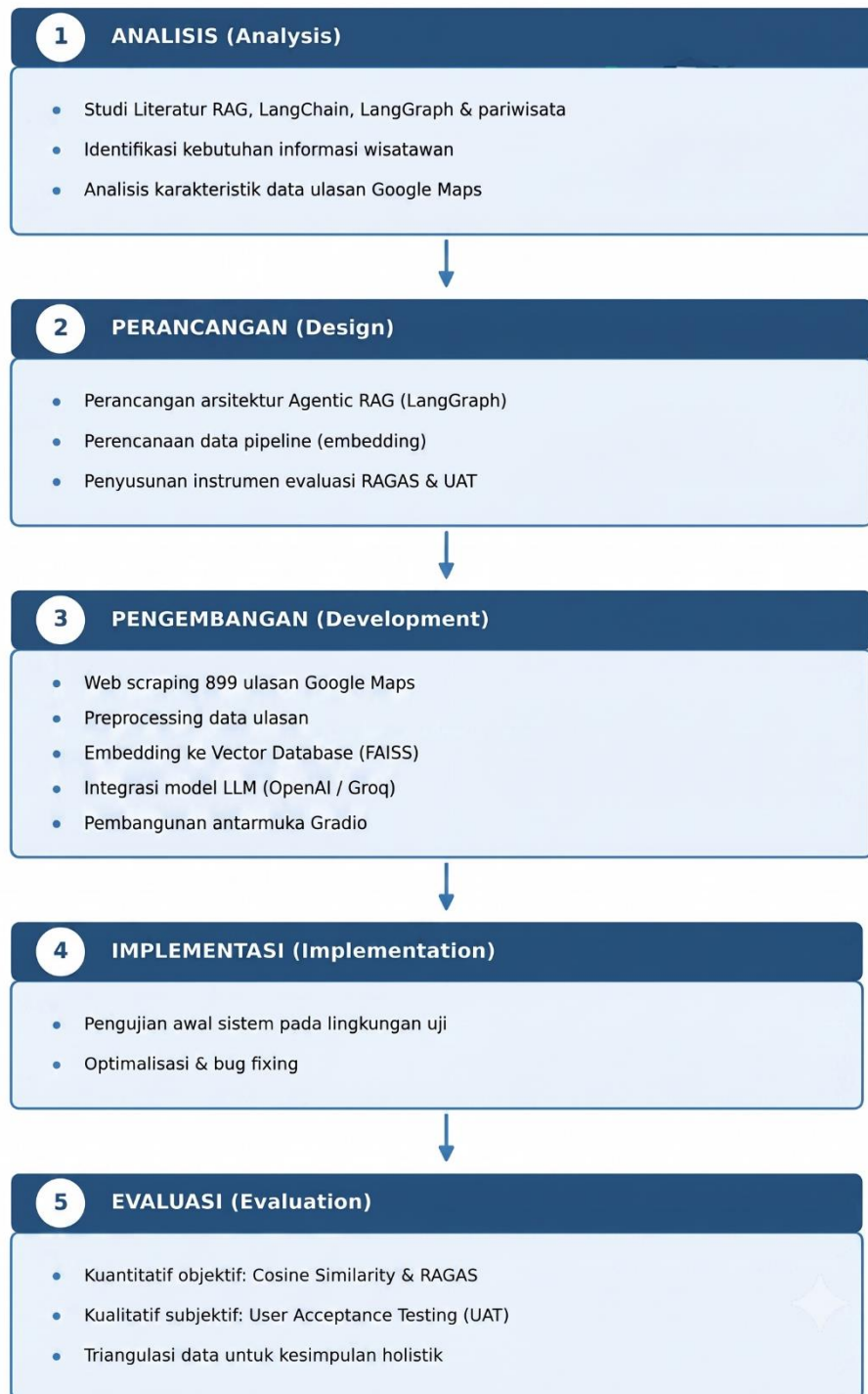
#### 3.6.1 Tahapan Pengembangan Model ADDIE

1. **Analisis (Analysis)**: Melakukan identifikasi kebutuhan informasi wisatawan, analisis karakteristik data ulasan, dan penentuan jenis pertanyaan umum.
2. **Perancangan (Design)**: Menyusun arsitektur sistem RAG, merencanakan alur data (*data pipeline*), serta merancang instrumen evaluasi teknis dan instrumen *User Acceptance Testing (UAT)*.
3. **Pengembangan (Development)**: Melakukan pemrosesan data (*preprocessing*), membangun basis pengetahuan (*knowledge base*), serta mengintegrasikan model LLM dan antarmuka percakapan.
4. **Implementasi (Implementation)**: Menerapkan sistem pada lingkungan pengujian, melakukan perbaikan kutu (*bug fixing*), dan mengoptimalkan performa sistem.
5. **Evaluasi (Evaluation)**: Menilai kualitas sistem secara holistik melalui dua pendekatan:
  - a. **Analisis Kuantitatif**: Melakukan perhitungan skor *Cosine Similarity* untuk mengukur presisi teknis pada tahap *retrieval* dan kemiripan jawaban dengan *ground truth*.

- b. **Analisis Kualitatif:** Menggunakan indikator RAGAS untuk performa sistem dan pengujian UAT untuk mengukur tingkat kepuasan pengguna secara nyata.

### **3.7 Alur Kerja Penelitian**

Berikut adalah rincian alur penelitian sesuai Model ADDIE, sebagaimana diilustrasikan pada Gambar 3.3:



**Gambar 3. 3 Alur Kerja Penelitian**

Sebagaimana ditunjukkan pada Gambar 3.3, alur kerja penelitian mengikuti lima tahap Model ADDIE secara berurutan. Tahap Analisis menjadi dasar melalui studi



literatur, identifikasi kebutuhan informasi wisatawan, dan analisis karakteristik data ulasan. Hasil tahap ini digunakan pada tahap Perancangan untuk menyusun arsitektur sistem Agentic RAG, merencanakan alur data, serta menyiapkan instrumen evaluasi RAGAS dan UAT. Rancangan tersebut kemudian direalisasikan pada tahap Pengembangan melalui web scraping, preprocessing dan embedding data ke Vector Database, integrasi model LLM, dan pembangunan antarmuka sistem. Sistem yang telah dibangun selanjutnya diuji pada tahap Implementasi melalui pengujian awal dan optimalisasi (bug fixing), sebelum akhirnya dievaluasi secara holistik pada tahap Evaluasi melalui pendekatan kuantitatif (Cosine Similarity dan RAGAS) serta kualitatif (User Acceptance Testing), yang hasilnya ditriangulasi untuk menghasilkan kesimpulan penelitian.

## 1. Tahap Analisis (*Analysis*)

Pada tahap awal ini, fokus utama adalah memetakan kebutuhan sistem dan data:

- a. **Studi Literatur:** Mengkaji teori terkait RAG, LangChain, LangGraph, dan pariwisata.
- b. **Identifikasi Kebutuhan:** Menentukan jenis informasi yang paling sering dicari wisatawan (fasilitas, aksesibilitas, harga, dll.).
- c. **Analisis Data:** Menganalisis karakteristik ulasan dari Google Maps yang akan dijadikan basis pengetahuan.

## 2. Tahap Perancangan (*Design*)

Tahap ini menyusun "cetak biru" sistem sebelum masuk ke proses pengkodean:

- a. **Arsitektur Sistem:** Merancang alur kerja *Agentic RAG* menggunakan **LangGraph** dan komponen **LangChain**.
- b. **Data Pipeline:** Merencanakan bagaimana data mentah akan diolah menjadi vektor (*embedding*).
- c. **Instrumen Evaluasi:** Menyusun 30 *test cases* untuk metrik RAGAS dan kuesioner UAT (skala Likert) untuk pengujian pengguna.

### 3. Tahap Pengembangan (*Development*)

Proses konstruksi sistem dilakukan secara teknis:

- a. **Web Scraping:** 899 ulasan pengunjung dari Google Maps.
- b. **Preprocessing & Embedding:** Membersihkan teks ulasan dan mengonversinya menjadi vektor untuk disimpan di *Vector Database* (seperti Chroma atau FAISS).
- c. **Integrasi Model:** Menghubungkan basis pengetahuan dengan LLM (seperti OpenAI atau Ollama).
- d. **Interface:** Membangun antarmuka pengguna berbasis web menggunakan *framework* seperti Streamlit atau Gradio.

### 4. Tahap Implementasi (*Implementation*)

Sistem yang telah dibangun kemudian diuji secara operasional:

- a. **Pengujian Awal:** Menjalankan sistem pada lingkungan pengujian untuk mendeteksi adanya *bug*.
- b. **Optimalisasi:** Melakukan perbaikan teknis (*bug fixing*) dan mengoptimalkan kecepatan serta ketepatan respons sistem.

### 5. Tahap Evaluasi (*Evaluation*)

Tahap akhir untuk mengukur kualitas sistem melalui tiga pilar utama:

- a. **Kuantitatif Objektif:** Mengukur presisi teknis menggunakan metrik **Cosine Similarity** dan kerangka kerja **RAGAS** (*faithfulness, precision, recall, relevancy*).
- b. **Kualitatif Subjektif:** Melakukan **User Acceptance Testing (UAT)** kepada 34 partisipan untuk menilai kepuasan dan kegunaan nyata sistem.
- c. **Triangulasi Data:** Menyintesis hasil dari berbagai metode evaluasi untuk menarik kesimpulan penelitian yang holistik.

### 3.8 Teknik Analisis Data

#### 3.8.1 Analisis Data Ulasan (Content Analysis)

Analisis konten (\*content analysis\*) dilakukan terhadap 899 ulasan pengunjung Api Abadi Mrapen yang diperoleh dari Google Maps melalui proses *web scraping*. Analisis mengacu pada metode Krippendorff dengan tiga tahapan utama:

1. **Coding:** Setiap ulasan diklasifikasikan ke dalam kategori tematik yang relevan dengan kebutuhan informasi wisatawan Api Abadi Mrapen, meliputi: FAC (Fasilitas), AKS (Aksesibilitas & Lokasi), HRG (Harga & Tiket), PGM (Pengalaman & Atmosfer), WKT (Waktu Kunjungan), TIP (Tips Praktis), dan KLH (Keluhan). Proses *coding* ini menjadi dasar pembentukan kategori pertanyaan dalam 8 *test cases* evaluasi sistem RAG.
2. **Categorizing:** Informasi yang serupa dikelompokkan untuk membentuk *knowledge base* yang terstruktur. Ulasan yang telah dibersihkan melalui tahap *preprocessing* (normalisasi, penghapusan karakter khusus, dan pemecahan teks menggunakan *RecursiveCharacterTextSplitter*) diindeks ke dalam FAISS *vector store* sebagai basis pengetahuan utama sistem.
3. **Interpreting:** Pola dan tema dominan dari ulasan diidentifikasi untuk menarik kesimpulan mengenai kebutuhan informasi wisatawan Api Abadi Mrapen. Temuan ini digunakan sebagai dasar penyusunan *ground truth* pada tahap evaluasi sistem serta sebagai acuan interpretasi kualitas jawaban yang dihasilkan oleh LLaMA-3.3-70B melalui pipeline LangGraph.

#### 3.8.2 Analisis Kuantitatif Kedekatan Semantik

1. **Perhitungan Skor Similarity:** Mengukur presisi *retrieval* menggunakan *Cosine Similarity* dengan model *paraphrase-multilingual-MiniLM-L12-v2* pada FAISS. Evaluasi dilakukan terhadap 8 pasangan kueri dan *ground truth* yang mewakili kategori utama informasi wisata.
2. **Penentuan Ambang Batas (Threshold):** Klasifikasi relevansi dibagi menjadi: Sangat Relevan ( $\geq 0,80$ ), Relevan (0,60–0,79), dan Kurang Relevan ( $< 0,60$ ). Nilai

0,30 ditetapkan pada node `validasi_konteks` LangGraph sebagai pemicu retrieve ulang melalui *expanded query*.

3. **Evaluasi Akurasi Retriever:** Menghitung rata-rata skor kemiripan dari 8 pasangan uji dan 40 *test case* RAGAS. Hasilnya kemudian diselaraskan dengan metrik *Faithfulness*, *Answer Relevancy*, *Context Precision*, dan *Context Recall* dalam proses triangulasi data.

### 3.8.3 Analisis UAT (Thematic Analysis)

1. **Analisis Deskriptif (Skala Likert):** Mengolah data dari 5 indikator (Q1–Q5) menggunakan nilai *mean* dan standar deviasi untuk mengukur tingkat kepuasan 34 responden. Data diambil secara otomatis melalui antarmuka Gradio ke format CSV untuk diinterpretasikan secara kuantitatif.
2. **Analisis Tematik:** Menganalisis komentar terbuka melalui empat tahap: familiarisasi data, *coding* segmen bermakna, kategorisasi (kelebihan, kekurangan, saran), dan identifikasi tema dominan sebagai dasar rekomendasi perbaikan sistem.
3. **Analisis Observasi:** Mengevaluasi catatan interaksi selama sesi UAT untuk mengidentifikasi pola kesulitan pengguna (*usability issues*) pada antarmuka Streamlit, efektivitas perumusan pertanyaan, dan respon sistem terhadap input di luar domain wisata.

### 3.8.4 Triangulasi Data

1. **Pilar Kuantitatif Objektif:** Pengukuran presisi matematis melalui skor *Cosine Similarity* (8 pasang data) menggunakan model *embedding* MiniLM-L12-v2 pada FAISS, serta skor RAGAS dari 40 *test case* (10 dasar + 30 gabungan).
2. **Pilar Kualitatif Terstandarisasi:** Evaluasi kualitas penalaran dan *retrieval* menggunakan metrik RAGAS (*Faithfulness*, *Answer Relevancy*, *Context Precision*, *Context Recall*) dengan *gpt-4o-mini* sebagai *LLM-as-a-judge*.
3. **Pilar Kualitatif Subjektif:** Penilaian kepuasan pengguna melalui pengujian UAT kepada 34 responden, mencakup 5 indikator skala Likert (Q1–Q5) dan umpan balik terbuka mengenai kualitas informasi sistem.

### 3.8.5 Penarikan Kesimpulan

Penarikan kesimpulan dilakukan melalui sintesis hasil triangulasi tiga pilar utama: kuantitatif objektif (skor *Cosine Similarity* dari 8 pasangan pertanyaan–*ground truth*), kualitatif terstandarisasi (metrik RAGAS pada 10 *test case* dasar dan 30 *test case* gabungan yang mencakup perbandingan Agentic RAG vs. RAG Baseline), dan kualitatif subjektif (hasil UAT 34 responden serta *feedback* terbuka pengguna). Proses ini bertujuan untuk mengukur pencapaian target berdasarkan ambang batas yang ditetapkan (skor RAGAS >0.70–0.80 sesuai indikator masing-masing dan tingkat kepuasan UAT >70%), sekaligus mengidentifikasi kekuatan serta kelemahan sistem RAG Api Abadi Mrapen. Hasil akhir berupa rekomendasi teknis terkait pipeline LangGraph dan FAISS, serta rekomendasi *user experience* (UX) antarmuka Gradio, yang divalidasi melalui konsistensi data lintas metode untuk menghasilkan simpulan penelitian yang holistik.

## 3.9 Instrumen Penelitian

### 3.9.1 Instrumen Evaluasi

1. **Test Case RAGAS:** 40 kasus uji (10 dasar + 30 diperkuat) untuk evaluasi komparatif *Agentic RAG* vs *Baseline*.
2. **Kuesioner UAT:** 5 aspek skala Likert (Akurasi, Kelengkapan, Keterbacaan, Relevansi, Kepuasan) untuk 34 responden.
3. **Pasangan Pertanyaan–Ground Truth Cosine Similarity:** 8 pasang pertanyaan dan jawaban referensi untuk mengukur kedekatan semantik jawaban sistem.
4. **Log Interaksi Sistem (*hasil\_user\_testing.csv*):** File CSV otomatis berisi data interaksi, skor Q1–Q5, komentar, dan rata-rata skor UAT.
5. **Panduan Pertanyaan Wawancara Pengelola:** 20 pasangan tanya-jawab dari 11 topik, bersumber dari wawancara pengelola, Pemkab Grobogan, dan PVMBG.
6. **Lembar Rekapitulasi Skor Evaluasi Kuantitatif:** Rekapitulasi skor *Cosine Similarity* dan RAGAS seluruh skenario pengujian.