JURNAL ILMU KOMPUTER
Volume 4 Nomor 1, Januari, 2026
ISSN: 3031-125X (ONLINE)
Penerbit: Program Studi Teknik Informatika S-2 Universitas Pamulang

Komparasi Kinerja Sistem Rekomendasi Destinasi Wisata Menggunakan *Content Based Filtering* Dan *Retrieval Augmented Generation* (RAG)

*Rachmat Aziz Awaludin[1], Arya Adyaksa Waskita [2], Mardiyanto [3]

[1,2,3)] Teknik Informatika S-2, Program Pascasarjana, Universitas Pamulang, Kota Tangerang Selatan, Banten

Email: [1]rachmataziz08@gmail.com, [2]aawaskita@unpam.ac.id, [3]dosen00027@unpam.ac.id

ABSTRACT

Advances in artificial intelligence have driven the development of recommendation systems in the tourism sector, which is characterized by diverse destinations. This condition often makes it difficult for tourists to select destinations that match their preferences. Based on literature studies, Content-Based Filtering (CBF) is widely used due to its efficiency; however, it has limitations in understanding contextual information. In contrast, the Retrieval Augmented Generation (RAG) approach has been developed to improve recommendation quality through semantic understanding. This study aims to compare the performance of CBF and RAG in tourism destination recommendation systems. CBF employs TF-IDF and cosine similarity to measure content similarity, while RAG integrates retrieval and generation processes using the LLaMA 3.2 model and the FAISS vector database. The research methodology includes data collection, text preprocessing, system implementation, and evaluation using context recall, faithfulness, answer relevancy, and similarity metrics. The results indicate that CBF achieved a context recall of 0.317, faithfulness of 1.000, answer relevancy of 0.190, and similarity of 0.293, demonstrating high accuracy with respect to source data but limited contextual understanding. Meanwhile, RAG achieved a context recall of 1.000, faithfulness of 0.783, answer relevancy of 0.617, and similarity of 0.715, indicating superior performance in generating relevant recommendations. In conclusion, RAG outperforms CBF in contextual and semantic aspects, while CBF remains more efficient in processing explicit data. This study is expected to serve as a reference for developing more adaptive and personalized tourism recommendation systems.

Keywords: *Content-Based Filtering; Retrieval Augmented Generation; Large Language Model; Recommendation System; Tourist Destination*.

ABSTRAK

Kemajuan teknologi kecerdasan buatan mendorong pengembangan sistem rekomendasi di sektor pariwisata yang memiliki keragaman destinasi. Kondisi ini menyebabkan wisatawan sering mengalami kesulitan dalam menentukan pilihan sesuai preferensi mereka. Berdasarkan studi literatur, metode Content-Based Filtering (CBF) banyak digunakan karena efisiensinya, namun memiliki keterbatasan dalam memahami konteks. Sebaliknya, pendekatan Retrieval Augmented Generation (RAG) dikembangkan untuk meningkatkan kualitas rekomendasi berbasis pemahaman semantik. Penelitian ini bertujuan membandingkan kinerja CBF dan RAG dalam sistem rekomendasi destinasi wisata. CBF menggunakan TF-IDF dan cosine similarity untuk mengukur kemiripan konten, sedangkan RAG mengintegrasikan proses retrieval dan generation melalui model LLaMA 3.2 dan vector database FAISS. Metodologi penelitian meliputi pengumpulan data, praproses teks, implementasi sistem, serta evaluasi menggunakan metrik context recall, faithfulness, answer relevancy, dan similarity. Hasil penelitian menunjukkan bahwa CBF memperoleh nilai context recall 0,317, faithfulness 1,000, answer relevancy 0,190, dan similarity 0,293, yang menunjukkan akurasi tinggi terhadap data sumber namun rendah dalam pemahaman konteks. Sementara itu, RAG mencapai context recall 1,000, faithfulness 0,783, answer relevancy 0,617, dan similarity 0,715, yang menunjukkan kemampuan lebih baik dalam menghasilkan rekomendasi yang relevan. Kesimpulannya, RAG lebih unggul dalam aspek kontekstual dan semantik, sedangkan CBF lebih efisien dalam pemrosesan berbasis data eksplisit. Penelitian ini diharapkan menjadi acuan dalam pengembangan sistem rekomendasi pariwisata yang lebih adaptif dan personal.

Kata Kunci: *Content Based Filtering; Retrieval Augmented Generation; Large Language Model*; Sistem Rekomendasi; Destinasi Wisata.

1. PENDAHULUAN

Perkembangan teknologi kecerdasan buatan (*Artificial Intelligence*/AI) dalam lima tahun terakhir telah menghasilkan kemajuan signifikan dalam pengolahan informasi, pemahaman bahasa alami, dan rekayasa pengetahuan. Perkembangan ini berpengaruh langsung terhadap sistem rekomendasi yang kini menjadi komponen kunci dalam berbagai aplikasi digital, mulai dari e-commerce, pendidikan, layanan hiburan, hingga pariwisata [1]. Sistem rekomendasi memainkan peran penting dalam mengurangi beban kognitif pengguna ketika harus memilih di antara banyaknya pilihan informasi, serta memastikan bahwa keputusan yang diambil lebih sesuai dengan kebutuhan dan preferensi personal [2]. Dengan meningkatnya jumlah data yang tersedia secara daring, kebutuhan sistem rekomendasi yang lebih cerdas, adaptif, dan dapat memahami konteks penggunaan menjadi semakin mendesak.

Sektor pariwisata merupakan salah satu domain yang paling membutuhkan sistem rekomendasi yang berkualitas karena karakteristiknya yang kompleks dan sangat dinamis. Pertumbuhan destinasi wisata yang terus bertambah, variasi pengalaman yang ditawarkan, serta perubahan kondisi eksternal seperti cuaca, kapasitas kunjungan, dan tren wisata global semakin mempersulit wisatawan dalam membuat keputusan perjalanan [3]. Wisatawan tidak hanya mempertimbangkan aspek umum seperti harga atau jarak, tetapi juga faktor-faktor yang bersifat personal seperti preferensi gaya perjalanan, kebutuhan kenyamanan, keamanan, pengalaman budaya, hingga keberlanjutan lingkungan [4]. Dalam konteks tersebut, kemampuan sistem rekomendasi untuk memberikan saran yang relevan, tepat waktu, dan sesuai preferensi individu menjadi sangat krusial.

Content-Based Filtering (CBF) merupakan salah satu pendekatan paling klasik namun tetap banyak digunakan dalam sistem rekomendasi destinasi wisata. Melalui analisis fitur intrinsik dari destinasi wisata seperti kategori objek, fasilitas, lokasi, ulasan, hingga aspek visual CBF mencocokkan karakteristik tersebut dengan profil pengguna untuk menghasilkan rekomendasi berbasis kesamaan konten [5]. Salah satu keunggulan utama CBF adalah kemampuannya untuk memberikan rekomendasi yang konsisten dan transparan, karena alasan rekomendasi dapat dijelaskan melalui atribut yang jelas [6]. Namun, efektivitas pendekatan ini sangat bergantung pada kualitas data yang tersedia. Dalam lingkungan pariwisata yang sarat perubahan, CBF sering kali tidak mampu menangkap dinamika informasi yang terus berkembang dan tidak memiliki mekanisme untuk menyesuaikan rekomendasi berdasarkan konteks yang berubah dari waktu ke waktu [7].

Seiring munculnya model bahasa berskala besar (Large Language Models/LLM), paradigma baru dalam sistem rekomendasi mulai berkembang, salah satunya melalui pendekatan Retrieval Augmented Generation (RAG). RAG merupakan metode yang mengombinasikan kemampuan pencarian dokumen relevan (*retrieval*) dengan kemampuan generatif LLM untuk menyusun rekomendasi berbasis narasi yang lebih alami, informatif, dan responsif [8]. Integrasi ini memungkinkan model untuk memperoleh pengetahuan eksternal secara dinamis dari database atau sumber informasi terkini, sehingga dapat menghasilkan keluaran yang lebih akurat dan kontekstual meskipun model dasar tidak dilatih secara eksplisit menggunakan seluruh data tersebut [9].

Dalam konteks aplikasi rekomendasi pariwisata, pendekatan RAG menawarkan keunggulan signifikan dibandingkan metode tradisional. RAG memungkinkan penggabungan informasi terkini seperti ulasan terbaru, pembaruan fasilitas, aktivitas musiman, agenda budaya lokal, hingga kondisi cuaca dan informasi keberlanjutan lingkungan. Informasi ini kemudian diolah menjadi rekomendasi yang bersifat naratif, mudah dipahami, dan lebih persuasif bagi pengguna [10]. Dengan karakteristik tersebut, RAG tidak hanya memberikan rekomendasi yang akurat, tetapi juga menyediakan justifikasi dalam bentuk uraian yang informatif, sehingga meningkatkan kepercayaan pengguna terhadap rekomendasi yang dihasilkan [11].

Meskipun berbagai pendekatan sistem rekomendasi telah dikembangkan dan diterapkan dalam domain pariwisata, masih terdapat sejumlah tantangan mendasar yang belum sepenuhnya teratasi. Salah satu permasalahan utama adalah keterbatasan dalam memahami kebutuhan pengguna yang bersifat dinamis dan kontekstual. Preferensi wisatawan dapat berubah seiring waktu, dipengaruhi oleh faktor internal seperti pengalaman sebelumnya dan kondisi ekonomi, maupun faktor eksternal seperti situasi sosial, kebijakan pemerintah, dan perkembangan teknologi digital [12]. Oleh karena itu, sistem rekomendasi modern dituntut tidak hanya mampu mengolah data historis, tetapi juga mengintegrasikan informasi real-time serta memahami perubahan pola perilaku pengguna secara adaptif.

Selain itu, tantangan lain yang sering dihadapi adalah rendahnya tingkat kepercayaan pengguna terhadap sistem rekomendasi otomatis. Banyak pengguna masih meragukan relevansi rekomendasi yang dihasilkan, terutama ketika sistem tidak mampu memberikan penjelasan yang memadai mengenai alasan di balik suatu rekomendasi [13]. Kurangnya transparansi ini dapat menurunkan tingkat adopsi sistem, khususnya dalam konteks pariwisata yang melibatkan keputusan finansial dan pengalaman personal yang signifikan. Oleh sebab itu, pengembangan sistem yang mampu menyajikan rekomendasi sekaligus justifikasi yang logis dan mudah dipahami menjadi aspek penting dalam meningkatkan kepercayaan pengguna.

Dalam beberapa tahun terakhir, penelitian juga menunjukkan bahwa integrasi antara kecerdasan buatan, big data, dan sistem rekomendasi berpotensi menciptakan layanan pariwisata yang lebih personal dan berkelanjutan. Pemanfaatan data lokasi, media sosial, sensor lingkungan, serta ulasan daring memungkinkan sistem untuk membangun pemahaman yang lebih komprehensif terhadap kondisi destinasi dan preferensi wisatawan [14]. Integrasi multidimensional ini membuka peluang bagi pengembangan sistem rekomendasi yang tidak hanya berorientasi pada kepuasan pengguna, tetapi juga memperhatikan aspek keberlanjutan sosial, budaya, dan lingkungan.

Namun demikian, penerapan teknologi berbasis LLM dan RAG dalam sistem rekomendasi pariwisata masih tergolong baru dan memerlukan kajian empiris yang mendalam. Beberapa studi menunjukkan bahwa meskipun pendekatan ini mampu menghasilkan rekomendasi yang lebih kaya secara semantik, tantangan terkait efisiensi komputasi, kualitas retrieval, serta potensi bias informasi masih perlu mendapat perhatian serius [15].

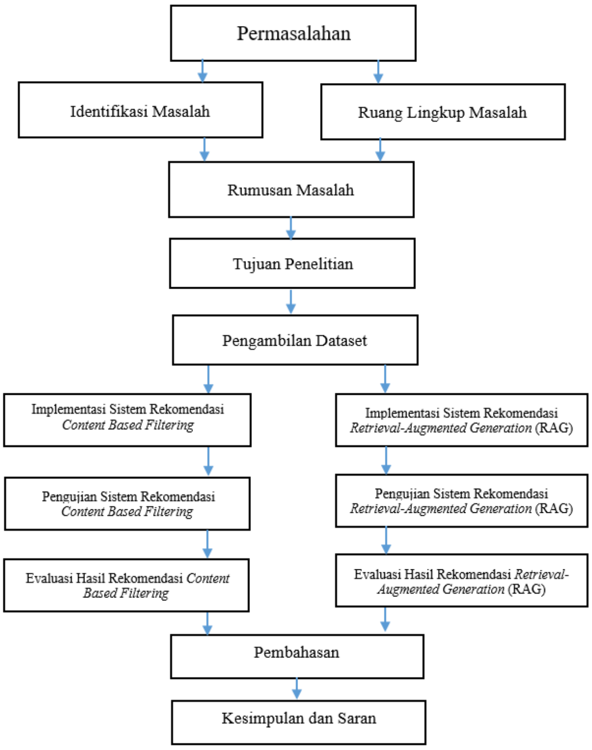
Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja pendekatan Content-Based Filtering dan Retrieval Augmented Generation dalam sistem rekomendasi destinasi wisata. Fokus utama penelitian diarahkan pada aspek relevansi rekomendasi, kemampuan adaptasi terhadap perubahan informasi, kualitas penjelasan rekomendasi, serta tingkat kepuasan pengguna. Hasil penelitian ini diharapkan dapat memberikan kontribusi teoretis dalam pengembangan sistem rekomendasi berbasis kecerdasan buatan, serta menjadi rujukan praktis bagi pengembang aplikasi pariwisata digital dan pemangku kepentingan terkait.

2. METODE

Penelitian ini menggunakan dataset berisi 437 destinasi wisata yang diperoleh dari situs Kaggle. Tahap awal penelitian adalah pengambilan dataset dan pemrosesan teks (*text preprocessing*) yang bertujuan untuk membersihkan serta menormalkan teks agar siap diolah pada tahap berikutnya. Pada sistem *content based filtering*, setiap deskripsi destinasi wisata diubah menjadi representasi numerik menggunakan teknik TF-IDF (*Term Frequency–Inverse Document Frequency*). Kemudian, sistem menghitung tingkat kesamaan antar destinasi dengan metode *cosine similarity* untuk menghasilkan daftar rekomendasi yang paling relevan dengan preferensi pengguna.

Sementara itu, pada sistem RAG, proses representasi teks dilakukan menggunakan model bahasa besar (*Large Language Model*) untuk menghasilkan *embedding* vektor yang menangkap makna semantik dari setiap deskripsi destinasi. *Embedding* tersebut disimpan dalam *vector database* yang memungkinkan pencarian berbasis kemiripan.

Selanjutnya, sistem menggunakan mekanisme *retrieval* untuk mengambil destinasi yang paling relevan berdasarkan permintaan pengguna, lalu model LLM menghasilkan teks penjelasan rekomendasi secara alami dan kontekstual. Kedua sistem kemudian diuji dan dibandingkan menggunakan metrik evaluasi, yaitu *context recall*, *faithfulness*, *answer relevancy,* dan *similarity* untuk menilai relevansi hasil rekomendasi terhadap data uji. Seluruh sistem dikembangkan dengan antarmuka berbasis web menggunakan *Streamlit***,** yang memungkinkan pengguna memasukkan preferensi wisata dan melihat hasil rekomendasi dari kedua metode secara interaktif. Penelitian ini dengan kerangka pemikiran di Gambar 1. di bawah ini



Gambar 1. Metode Penelitian

Penelitian ini diawali dengan tahap permasalahan, yaitu mengidentifikasi kondisi nyata yang terjadi di lapangan terkait kesulitan wisatawan dalam memperoleh rekomendasi destinasi yang sesuai dengan preferensi mereka. Pada tahap ini, peneliti mengamati sistem yang sudah ada, mempelajari literatur, serta mengkaji kebutuhan pengguna untuk menemukan permasalahan utama yang relevan dengan bidang penelitian. Selanjutnya, dilakukan identifikasi masalah dengan menguraikan permasalahan utama menjadi beberapa aspek yang lebih spesifik, seperti keterbatasan sistem dalam memberikan rekomendasi yang personal, kurangnya pemanfaatan data ulasan, serta belum optimalnya penggunaan teknologi kecerdasan buatan. Tahap ini bertujuan untuk mengetahui akar permasalahan yang harus diselesaikan melalui penelitian.

Setelah itu, peneliti menetapkan ruang lingkup masalah guna membatasi cakupan penelitian agar tetap fokus dan realistis. Pada tahap ini ditentukan batasan penelitian, seperti jenis data yang digunakan, wilayah destinasi wisata yang diteliti, serta platform sistem yang dikembangkan. Pembatasan ini dilakukan agar penelitian tidak terlalu luas dan dapat diselesaikan sesuai waktu yang tersedia.

Berdasarkan permasalahan yang telah diidentifikasi, peneliti menyusun rumusan masalah dalam bentuk pertanyaan penelitian. Rumusan masalah berfungsi sebagai pedoman utama penelitian, misalnya mengenai bagaimana kinerja sistem rekomendasi berbasis CBF dan RAG serta metode mana yang memberikan hasil terbaik. Selanjutnya, peneliti menetapkan tujuan penelitian yang merupakan penjabaran dari rumusan masalah. Tujuan ini meliputi pengembangan sistem rekomendasi, penerapan metode CBF dan RAG, serta evaluasi kinerja kedua metode tersebut. Tujuan penelitian menjadi indikator keberhasilan yang ingin dicapai.

Tahap berikutnya adalah pengambilan dataset, yaitu proses pengumpulan data yang akan digunakan dalam sistem rekomendasi. Data dapat berupa informasi destinasi wisata, deskripsi, ulasan, dan penilaian pengguna. Data diperoleh melalui web scraping, API, atau sumber resmi, kemudian dilakukan proses pembersihan, normalisasi, dan preprocessing teks agar siap digunakan dalam pemodelan. Setelah dataset tersedia, penelitian dilanjutkan dengan implementasi sistem rekomendasi CBF. Pada tahap ini, peneliti membangun sistem yang merekomendasikan destinasi berdasarkan kesamaan konten, seperti deskripsi dan ulasan. Data teks diolah menggunakan teknik ekstraksi fitur, seperti TF-IDF atau word embedding, kemudian dihitung tingkat kemiripannya menggunakan metode cosine similarity untuk menghasilkan rekomendasi. Setelah sistem CBF diimplementasikan, dilakukan pengujian sistem rekomendasi CBF untuk memastikan sistem berjalan sesuai dengan fungsinya. Pengujian dilakukan dengan memberikan berbagai skenario input dan mengevaluasi apakah sistem mampu menghasilkan rekomendasi tanpa kesalahan teknis.

Selanjutnya, dilakukan evaluasi hasil rekomendasi CBF untuk mengukur kualitas rekomendasi yang dihasilkan. Evaluasi dilakukan menggunakan metrik tertentu atau melalui penilaian pengguna, sehingga dapat diketahui tingkat relevansi dan akurasi rekomendasi yang diberikan oleh sistem. Pada jalur kedua, peneliti melakukan implementasi sistem rekomendasi berbasis RAG. Tahap ini melibatkan proses *chunking* dokumen, pembuatan *embedding*, penyimpanan data dalam basis data vektor, serta integrasi dengan model bahasa besar. Sistem RAG dirancang untuk mengambil informasi relevan dari basis data dan menghasilkan rekomendasi dalam bentuk teks yang lebih kontekstual. Setelah implementasi RAG selesai, dilakukan pengujian sistem rekomendasi RAG untuk memastikan bahwa sistem mampu memberikan jawaban yang sesuai dengan konteks dan bebas dari kesalahan yang signifikan. Pengujian dilakukan dengan memberikan berbagai pertanyaan dan skenario penggunaan, serta mengevaluasi kestabilan dan keakuratan respons sistem.

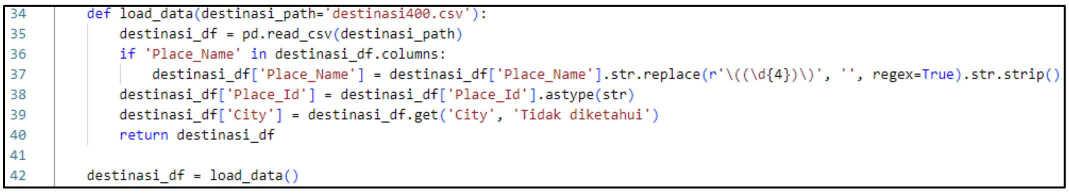
Tahap berikutnya adalah evaluasi hasil rekomendasi RAG, yang bertujuan untuk menilai kualitas keluaran sistem berbasis AI. Evaluasi dilakukan melalui penilaian relevansi, konsistensi konteks, serta survei pengguna, sehingga dapat diketahui sejauh mana sistem RAG mampu memberikan rekomendasi yang informatif dan bermanfaat. Setelah seluruh tahapan implementasi dan evaluasi selesai, peneliti memasuki tahap pembahasan. Pada tahap ini, hasil evaluasi dari metode CBF dan RAG dianalisis dan dibandingkan. Peneliti membahas kelebihan, kekurangan, serta faktor-faktor yang memengaruhi kinerja masing-masing metode secara sistematis.

Tahap terakhir adalah kesimpulan dan saran, yaitu merangkum seluruh hasil penelitian dan menjawab rumusan masalah yang telah disusun. Pada bagian ini, peneliti juga memberikan rekomendasi untuk pengembangan sistem di masa mendatang serta saran bagi penelitian selanjutnya agar dapat menghasilkan sistem rekomendasi yang lebih optimal.

3. HASIL DAN PEMBAHASAN

3.1. Pengambilan Dataset

Data yang digunakan dalam penelitian ini merupakan dataset destinasi wisata yang berasal dari www.kaggle.com. Dataset ini menggunakan file destinasi400.csv, mencakup 437 data yang berisi tentang *Place_Id, Place_Name, Description, Category, City, Price, Rating, Time_Minutes, Coordinate, Lat, Long.* Dataset digunakan untuk dua tujuan utama, yaitu melatih sistem rekomendasi berbasis *content-based filtering* dan sebagai sumber dokumen untuk model *Large Language Model (LLM)* dalam proses *retrieval*.. Dengan demikian, sistem dapat diuji dalam konteks yang lebih luas dan hasilnya lebih representatif.



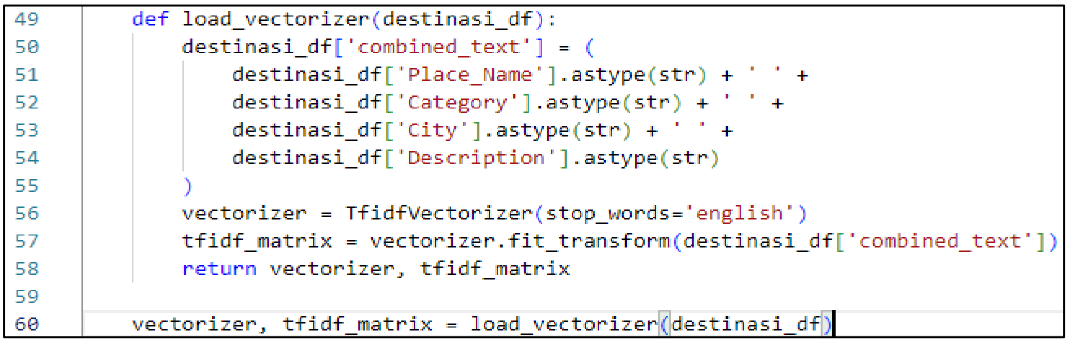
```python
def load_data(destinasi_path='destinasi400.csv'):
    destinasi_df = pd.read_csv(destinasi_path)
    if 'Place_Name' in destinasi_df.columns:
        destinasi_df['Place_Name'] = destinasi_df['Place_Name'].str.replace(r'\((\d{4})\)', '', regex=True).str.strip()
    destinasi_df['Place_Id'] = destinasi_df['Place_Id'].astype(str)
    destinasi_df['City'] = destinasi_df.get('City', 'Tidak diketahui')
    return destinasi_df

destinasi_df = load_data()
```

Gambar 2. *Source code* untuk membaca dataset

Fungsi ini membaca dua file utama yaitu destinasi400.csv yang berisi data destinasi wisata seperti Place_Id, Place_Name, dan Category. Selanjutnya file ratings.csv berisi data penilaian pengguna terhadap destinasi tertentu. Selain itu, dilakukan pembersihan data dengan menghapus karakter tambahan pada kolom Place_Name dan memastikan keseragaman tipe data Place_Id.

3.2. Implementasi Sistem Rekomendasi Content Based Filtering

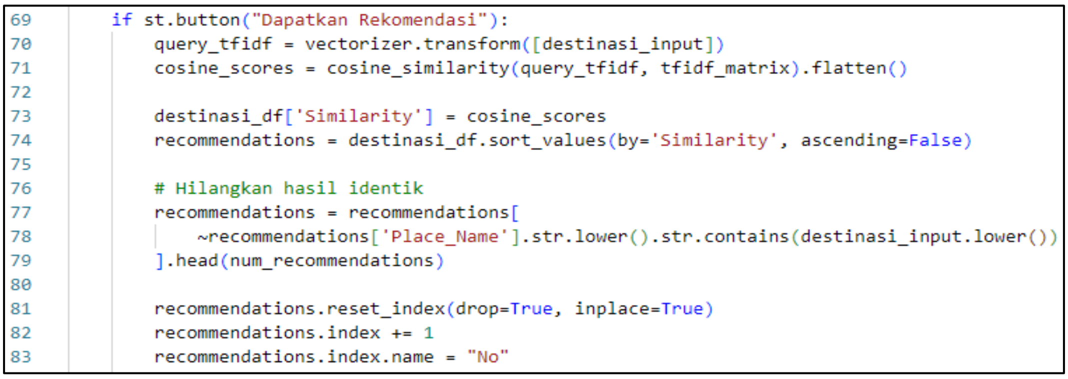
Pada tahap ini, sistem membangun representasi fitur teks untuk setiap destinasi wisata dengan menggabungkan beberapa kolom informasi, yaitu *Place_Name*, *Category*, dan *City* ke dalam satu kolom baru bernama *combined_text*. Proses ini bertujuan agar setiap entri memiliki deskripsi teks yang lebih komprehensif. Selanjutnya, sistem menerapkan metode TF-IDF (*Term Frequency–Inverse Document Frequency*) menggunakan pustaka *TfidfVectorizer* dari *scikit-learn* untuk mengubah kumpulan teks tersebut menjadi representasi vektor numerik. Metode ini menilai tingkat kepentingan setiap kata dalam suatu destinasi dibandingkan dengan seluruh destinasi lain dalam dataset. Hasilnya berupa matriks TF-IDF (*tfidf_matrix*) yang menjadi dasar perhitungan kesamaan antar destinasi dalam tahap berikutnya. Proses ini dilakukan melalui fungsi berikut:



```python
def load_vectorizer(destinasi_df):
    destinasi_df['combined_text'] = (
        destinasi_df['Place_Name'].astype(str) + ' ' +
        destinasi_df['Category'].astype(str) + ' ' +
        destinasi_df['City'].astype(str) + ' ' +
        destinasi_df['Description'].astype(str)
    )
    vectorizer = TfidfVectorizer(stop_words='english')
    tfidf_matrix = vectorizer.fit_transform(destinasi_df['combined_text'])
    return vectorizer, tfidf_matrix

vectorizer, tfidf_matrix = load_vectorizer(destinasi_df)
```

Gambar 3. Source code untuk pembentukan matriks TF-IDF

Selanjutnya sistem menghasilkan rekomendasi destinasi wisata berdasarkan input pengguna. Teks destinasi yang dimasukkan pengguna terlebih dahulu diubah menjadi representasi numerik menggunakan model TF-IDF yang telah dibentuk sebelumnya. Selanjutnya, sistem menghitung tingkat kesamaan antara destinasi input dan seluruh destinasi dalam dataset menggunakan metrik *cosine similarity*. Nilai kesamaan tersebut disimpan pada kolom *Similarity* di dalam *DataFrame* utama. Berikut ini adalah potongan gambar kode untuk proses rekomendasi.



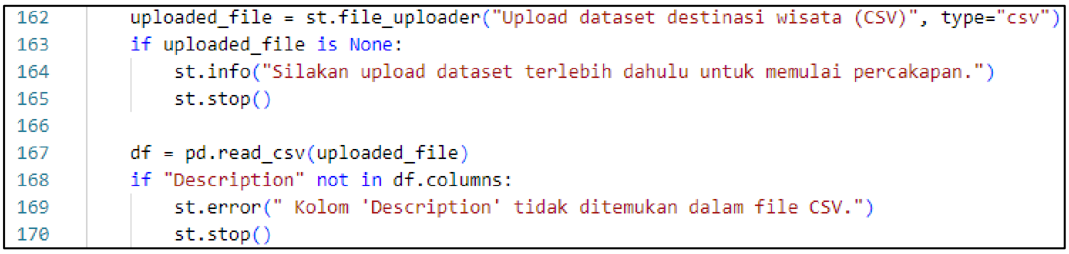
```
    if st.button("Dapatkan Rekomendasi"):
        query_tfidf = vectorizer.transform([destinasi_input])
        cosine_scores = cosine_similarity(query_tfidf, tfidf_matrix).flatten()

        destinasi_df['Similarity'] = cosine_scores
        recommendations = destinasi_df.sort_values(by='Similarity', ascending=False)

        # Hilangkan hasil identik
        recommendations = recommendations[
            ~recommendations['Place_Name'].str.lower().str.contains(destinasi_input.lower())
        ].head(num_recommendations)

        recommendations.reset_index(drop=True, inplace=True)
        recommendations.index += 1
        recommendations.index.name = "No"
```

Gambar 4. *Source code* untuk menghasilkan rekomendasi

Hasil perhitungan kemudian diurutkan secara menurun berdasarkan nilai *Similarity* untuk menampilkan destinasi dengan tingkat kemiripan tertinggi. Sistem juga menghapus destinasi yang identik dengan input pengguna agar hasil rekomendasi lebih relevan. Beberapa destinasi teratas dengan skor tertinggi ditampilkan sebagai hasil akhir rekomendasi. Data hasil rekomendasi kemudian ditampilkan dalam bentuk tabel interaktif menggunakan komponen *st.dataframe()* pada antarmuka *Streamlit*.

3.3. Implementasi Sistem Rekomendasi Retrieval Augmented Generation (RAG)

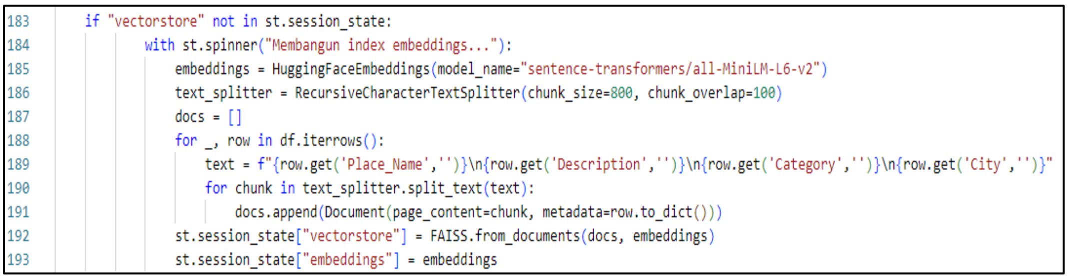
Tahap pertama adalah mengunggah dan memvalidasi dataset yang digunakan untuk membangun basis pengetahuan destinasi wisata. Proses validasi memastikan file yang diunggah memenuhi struktur yang dibutuhkan. Jika tidak, aplikasi akan menampilkan pesan kesalahan (*st.error*) dan menghentikan proses selanjutnya. Berikut dibawah ini adalah potongan gambar kode untuk proses mengunggah dan memvalidasi dataset.



```
    uploaded_file = st.file_uploader("Upload dataset destinasi wisata (CSV)", type="csv")
    if uploaded_file is None:
        st.info("Silakan upload dataset terlebih dahulu untuk memulai percakapan.")
        st.stop()

    df = pd.read_csv(uploaded_file)
    if "Description" not in df.columns:
        st.error(" Kolom 'Description' tidak ditemukan dalam file CSV.")
        st.stop()
```

Gambar 5. *Source code* untuk mengunggah dan memvalidasi dataset

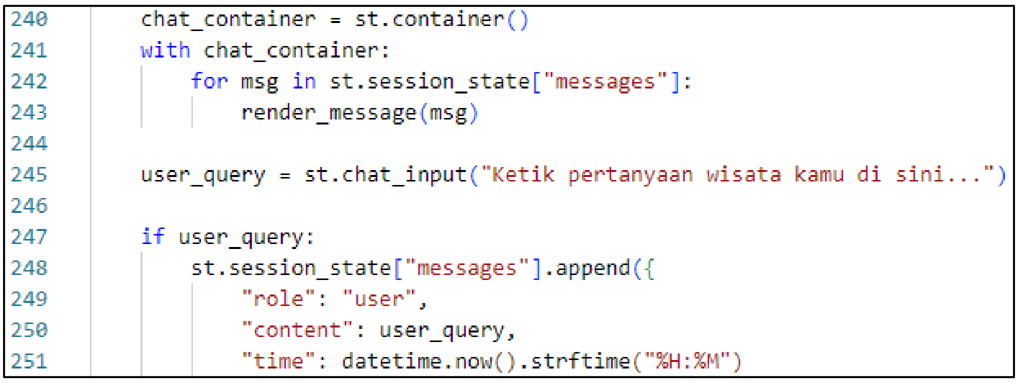
Selanjutnya setiap baris data wisata diubah menjadi dokumen teks yang berisi gabungan informasi nama, kota, kategori, deskripsi dan aktivitas. Untuk mempermudah pencarian semantik, teks dipotong menjadi beberapa potongan (*chunks*) menggunakan *`RecursiveCharacterTextSplitter`* agar ukuran setiap dokumen tidak terlalu panjang. Setiap potongan teks diubah menjadi *embedding* menggunakan model `all-MiniLM-L6-v2` dari HuggingFace. *Embedding* adalah representasi numerik dari teks yang mempertahankan makna semantik. Seluruh *embedding* kemudian disimpan ke dalam *vector database FAISS (Facebook AI Similarity Search)* agar sistem dapat mencari dokumen yang paling mirip dengan pertanyaan pengguna berdasarkan kesamaan semantik. Gambar 6 dibawah ini adalah potongan gambar kode untuk proses *embedding*.



```
    if "vectorstore" not in st.session_state:
            with st.spinner("Membangun index embeddings..."):
                embeddings = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")
                text_splitter = RecursiveCharacterTextSplitter(chunk_size=800, chunk_overlap=100)
                docs = []
                for _, row in df.iterrows():
                    text = f"{row.get('Place_Name','')}\n{row.get('Description','')}\n{row.get('Category','')}\n{row.get('City','')}"
                    for chunk in text_splitter.split_text(text):
                        docs.append(Document(page_content=chunk, metadata=row.to_dict()))
                st.session_state["vectorstore"] = FAISS.from_documents(docs, embeddings)
                st.session_state["embeddings"] = embeddings
```

Gambar 6. *Source code* untuk proses *embedding*

Bagian ini membentuk **antarmuka pengguna** menggunakan komponen `st.chat_input` dari Streamlit. Pengguna dapat mengetikkan pertanyaan dalam bahasa alami seperti: *"Rekomendasikan 5 destinasi wisata sejenis pantai baron?"*. Semua percakapan disimpan dalam variabel *`st.session_state ["messages"]`* agar riwayat percakapan tetap tersimpan selama sesi berlangsung. Antarmuka dirancang menyerupai aplikasi pesan untuk pengalaman interaksi yang lebih alami. Gambar 7 dibawah ini adalah potongan gambar kode untuk menampilkan antarmuka pengguna.



```
    chat_container = st.container()
    with chat_container:
        for msg in st.session_state["messages"]:
            render_message(msg)

    user_query = st.chat_input("Ketik pertanyaan wisata kamu di sini...")

    if user_query:
        st.session_state["messages"].append({
            "role": "user",
            "content": user_query,
            "time": datetime.now().strftime("%H:%M")
```

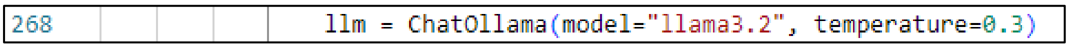
Gambar 7. *Source code* untuk menampilkan antarmuka pengguna

Ketika pengguna mengajukan pertanyaan, sistem akan mencari 5 dokumen paling relevan dari basis pengetahuan menggunakan metode *semantic similarity search*. Proses ini menggunakan *FAISS retriever* untuk menilai kedekatan makna antara pertanyaan pengguna dan dokumen yang tersimpan. Selanjutnya, sistem menyusun konteks gabungan dari hasil pencarian. Konteks ini berisi informasi yang akan digunakan oleh model LLM sebagai dasar dalam membentuk jawaban yang sesuai dengan data aktual destinasi wisata.

```
    if st.session_state["messages"]:
        last_msg = st.session_state["messages"][-1]
        if last_msg["role"] == "user":
            user_query = last_msg["content"]
            embeddings = st.session_state["embeddings"]
            vectorstore = st.session_state["vectorstore"]

            with st.spinner("Asisten sedang mencari jawaban..."):
                retriever = vectorstore.as_retriever(search_kwargs={"k": 5})
                docs = retriever.get_relevant_documents(user_query)
                context = "\n\n".join([d.page_content for d in docs])
```

Gambar 8. *Source code* proses *retrieval*

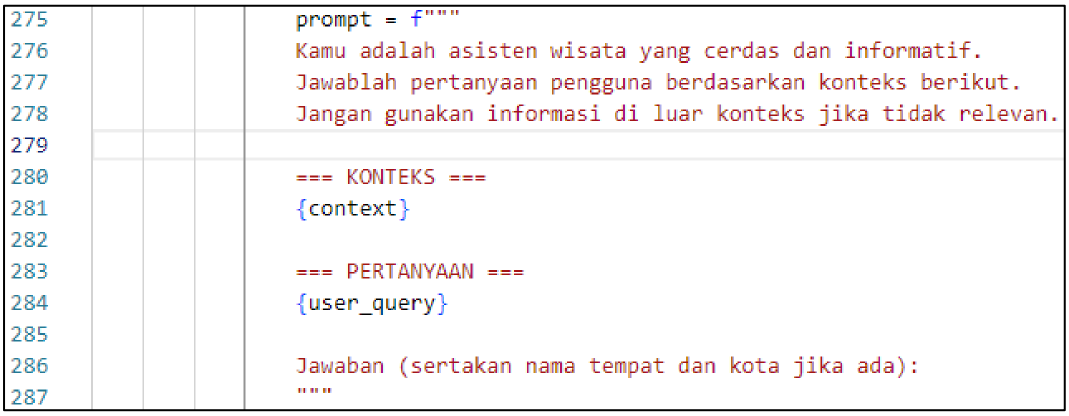
Selanjutnya sistem menggunakan model Llama 3.2 yang diakses melalui Ollama sebagai *Large Language Model (LLM)* untuk menghasilkan jawaban berdasarkan konteks hasil pencarian. *Temperature* 0.3 digunakan agar jawaban tetap fokus dan tidak terlalu variatif. Prompt berisi gabungan konteks (hasil pencarian dari FAISS) dan pertanyaan pengguna. Gambar dibawah ini adalah potongan gambar kode untuk inisialisasi LLM.



```
                llm = ChatOllama(model="llama3.2", temperature=0.3)
```

Gambar 9. Source code untuk inisialisasi LLM

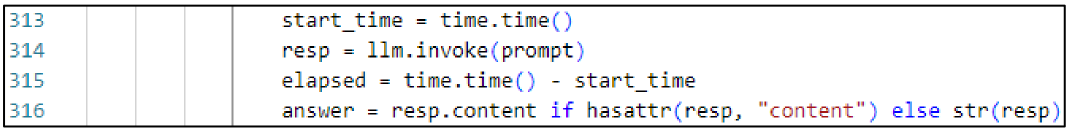
Tahap berikutnya adalah menyusun *prompt*, yaitu teks instruksi yang dikirim ke LLM agar model tahu peran yang harus dijalankan dan informasi apa yang digunakan. Tujuannya adalah membentuk *prompt* yang terarah, kontekstual, dan berbasis data, agar LLM tidak menjawab berdasarkan hafalan umum (*hallucination*), melainkan dari *retrieved documents* yang sudah dipilih sistem.



```
            prompt = f"""
            Kamu adalah asisten wisata yang cerdas dan informatif.
            Jawablah pertanyaan pengguna berdasarkan konteks berikut.
            Jangan gunakan informasi di luar konteks jika tidak relevan.

            === KONTEKS ===
            {context}

            === PERTANYAAN ===
            {user_query}

            Jawaban (sertakan nama tempat dan kota jika ada):
            """
```

Gambar 10. Source code pembuatan *prompt*

Pada tahap selanjutnya adalah mengambil jawaban dan menjamin bahwa sistem tetap dapat menampilkan teks jawaban LLM dengan benar, meskipun struktur objek output dari model sedikit berbeda antar versi. Selanjutnya adalah mengukur *response time* model, yang menjadi salah satu indikator performa sistem (efisiensi pemrosesan LLM). Gambar dibawah ini adalah potongan gambar kode untuk mengambil jawaban dan mengukur waktu respon.

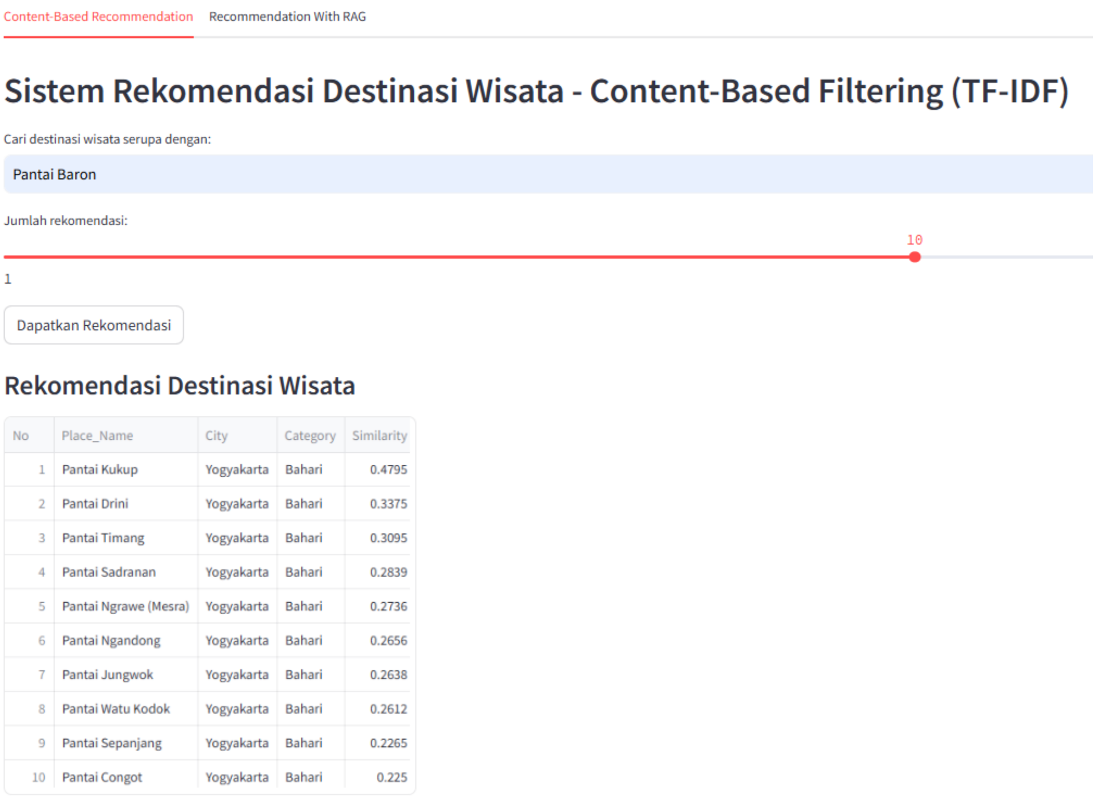


```
            start_time = time.time()
            resp = llm.invoke(prompt)
            elapsed = time.time() - start_time
            answer = resp.content if hasattr(resp, "content") else str(resp)
```

Gambar 11. *Source code* untuk mengambil jawaban dan waktu respon

3.4. Pengujian Sistem Rekomendasi Content Based Filtering dan RAG

Pada tahap pengujian ini, pengguna memasukkan nama destinasi, misalnya “Pantai Baron”, kemudian sistem menampilkan destinasi lain dengan tingkat kesamaan tertinggi berdasarkan nilai *cosine similarity*. Jumlah rekomendasi dapat diatur antara 1–20 hasil teratas. Sebagai contoh, hasil untuk “Pantai Baron” menghasilkan 10 rekomendasi seperti pada gambar dibawah ini:



Sistem Rekomendasi Destinasi Wisata - Content-Based Filtering (TF-IDF)

Cari destinasi wisata serupa dengan: Pantai Baron

Jumlah rekomendasi: 10

Rekomendasi Destinasi Wisata

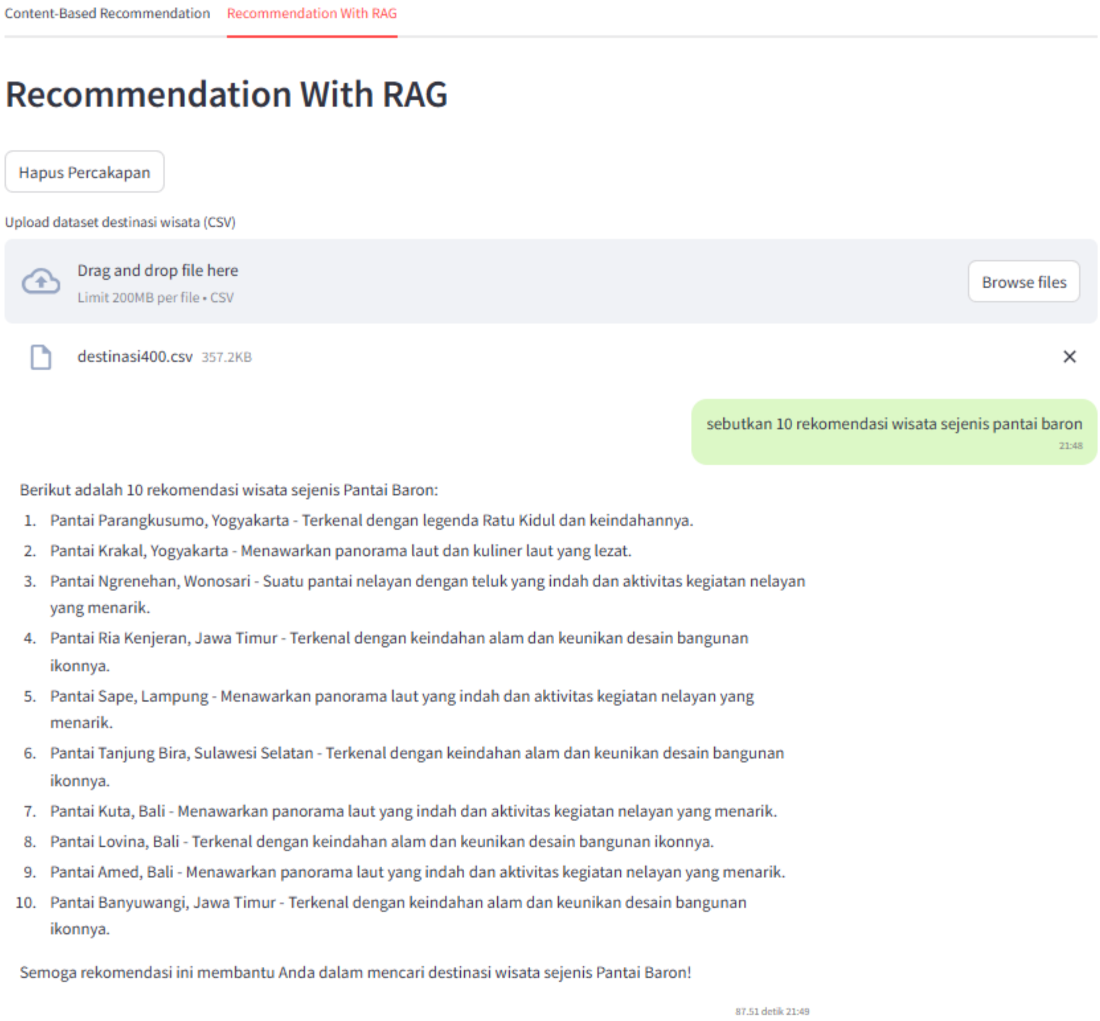
No	Place_Name	City	Category	Similarity
1	Pantai Kukup	Yogyakarta	Bahari	0.4795
2	Pantai Drini	Yogyakarta	Bahari	0.3375
3	Pantai Timang	Yogyakarta	Bahari	0.3095
4	Pantai Sadranan	Yogyakarta	Bahari	0.2839
5	Pantai Ngrawe (Mesra)	Yogyakarta	Bahari	0.2736
6	Pantai Ngandong	Yogyakarta	Bahari	0.2656
7	Pantai Jungwok	Yogyakarta	Bahari	0.2638
8	Pantai Watu Kodok	Yogyakarta	Bahari	0.2612
9	Pantai Sepanjang	Yogyakarta	Bahari	0.2265
10	Pantai Congot	Yogyakarta	Bahari	0.225

Gambar 12. Hasil pengujian sistem rekomendasi *content based filtering*

Gambar 12 menghasilkan 10 rekomendasi wisata yang sejenis dan relevan dengan Pantai Baron yaitu Pantai Kukup, Pantai Drini, Pantai Timang, Pantai Sadranan, Pantai Ngrawe, Pantai Ngandong, Pantai Jungwok, Pantai Watu Kodok, Pantai Sepanjang, Pantai Congot. Setiap destinasi dilengkapi dengan kolom kategori, nilai similarity, serta relevansi.

Sedangkan untuk *Retrieval Augmented Generation* (RAG) dilakukan dengan mengintegrasikan *Large Language Model* (LLaMA 3.2) melalui Ollama API ke dalam sistem rekomendasi. Pendekatan ini menggunakan konsep *Retrieval Augmented Generation* (RAG), di mana model tidak hanya mengandalkan kemampuan generatifnya, tetapi juga melakukan proses pencarian (*retrieval*) terhadap informasi yang relevan dari basis data wisata.

Sebelum model digunakan, sistem terlebih dahulu membangun *knowledge base* berupa dokumen teks yang disimpan dalam *vector database* FAISS. Setiap entri destinasi wisata diubah menjadi *embedding* vektor menggunakan model *HuggingFace all-MiniLM-L6-v2* agar sistem dapat melakukan pencarian semantik berdasarkan makna teks, bukan sekadar kesamaan kata. Gambar dibawah ini adalah hasil percobaan sistem rekomendasi menggunakan *Retrieval Augmented Generation* (RAG).



sebutkan 10 rekomendasi wisata sejenis pantai baron
21:48

Berikut adalah 10 rekomendasi wisata sejenis Pantai Baron:

1. Pantai Parangkusumo, Yogyakarta - Terkenal dengan legenda Ratu Kidul dan keindahannya.
2. Pantai Krakal, Yogyakarta - Menawarkan panorama laut dan kuliner laut yang lezat.
3. Pantai Ngrenehan, Wonosari - Suatu pantai nelayan dengan teluk yang indah dan aktivitas kegiatan nelayan yang menarik.
4. Pantai Ria Kenjeran, Jawa Timur - Terkenal dengan keindahan alam dan keunikan desain bangunan ikonnya.
5. Pantai Sape, Lampung - Menawarkan panorama laut yang indah dan aktivitas kegiatan nelayan yang menarik.
6. Pantai Tanjung Bira, Sulawesi Selatan - Terkenal dengan keindahan alam dan keunikan desain bangunan ikonnya.
7. Pantai Kuta, Bali - Menawarkan panorama laut yang indah dan aktivitas kegiatan nelayan yang menarik.
8. Pantai Lovina, Bali - Terkenal dengan keindahan alam dan keunikan desain bangunan ikonnya.
9. Pantai Amed, Bali - Menawarkan panorama laut yang indah dan aktivitas kegiatan nelayan yang menarik.
10. Pantai Banyuwangi, Jawa Timur - Terkenal dengan keindahan alam dan keunikan desain bangunan ikonnya.

Semoga rekomendasi ini membantu Anda dalam mencari destinasi wisata sejenis Pantai Baron!

87.51 detik 21:49

Gambar 13. Hasil pengujian sistem rekomendasi menggunakan RAG

Pada pengujian sistem rekomendasi berbasis *Retrieval Augmented Generation* (RAG), model menghasilkan respons yang lebih kontekstual dan informatif dibandingkan metode berbasis konten. Sistem mampu memahami pertanyaan pengguna yang bersifat alami seperti *"Sebutkan 10 rekomendasi wisata sejenis Pantai Baron"*, kemudian melakukan proses *retrieval* terhadap dokumen wisata yang tersimpan dalam basis pengetahuan, dan menghasilkan jawaban yang bersifat deskriptif.

3.5. Evaluasi Hasil Rekomendasi Content Based Filtering dan RAG

Pada sistem rekomendasi berbasis *Content Based Filtering*, evaluasi dilakukan dengan membandingkan hasil rekomendasi sistem terhadap data *ground truth* yang diperoleh dari penilaian pengguna atau label kategori wisata yang telah ada dalam dataset. Selanjutnya berdasarkan hasil pengujian sistem rekomendasi menggunakan *content based filtering* pada pengujian sebelumnya, menghasilkan 10 rekomendasi dan sistem akan secara otomatis menampilkan hasil metrik evaluasi seperti pada tabel dibawah ini.

Table 1. Hasil Rekomendasi

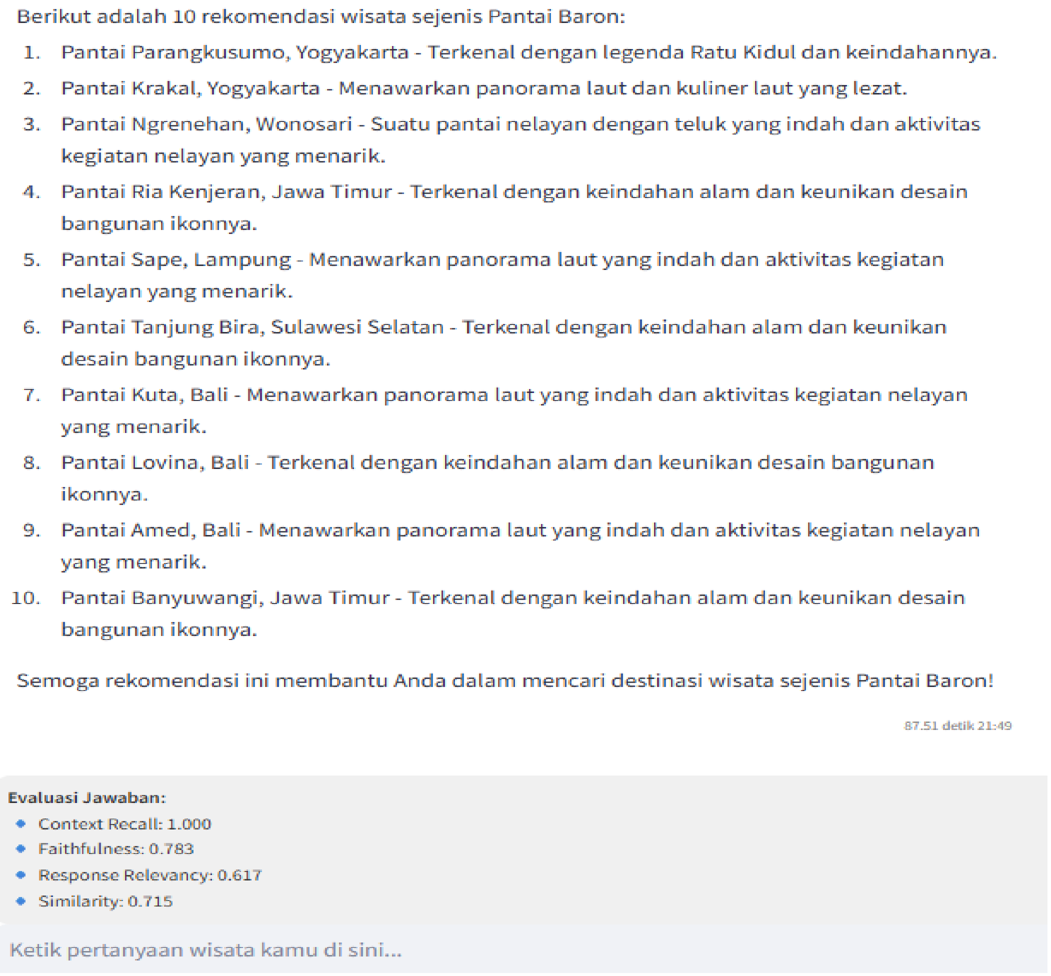
No	*Place_Name*	*City*	*Category*	*Similarity*
1	Pantai Kukup	Yogyakarta	Bahari	0,4795
2	Pantai Drini	Yogyakarta	Bahari	0,3375
3	Pantai Timang	Yogyakarta	Bahari	0,3095
4	Pantai Sadranan	Yogyakarta	Bahari	0,2839
5	Pantai Ngerawe (Mesra)	Yogyakarta	Bahari	0,2736
6	Pantai Ngandong	Yogyakarta	Bahari	0,2656
7	Pantai Jungwok	Yogyakarta	Bahari	0,2638
8	Pantai Watu Kodok	Yogyakarta	Bahari	0,2612
9	Pantai Sepanjang	Yogyakarta	Bahari	0,2265
10	Pantai Congot	Yogyakarta	Bahari	0,225

Table 2. Hasil Evaluasi Sistem

Metode	Skor
Context Recall	0,317
Faithfulness	1,000
Answer Relevancy	0,190
Similarity	0,293

Hasil evaluasi terhadap sistem rekomendasi berbasis *content based filtering* menunjukkan bahwa performa sistem tergolong cukup baik. Nilai *context recall* sebesar 0.317 mengindikasikan bahwa sistem mampu menangkap sebagian konteks dari permintaan pengguna dalam proses pencarian destinasi wisata. Sementara itu, *faithfulness* dengan nilai sempurna 1.000 menunjukkan bahwa seluruh hasil rekomendasi benar-benar sesuai dengan data yang ada, tanpa adanya informasi yang keliru atau menyimpang dari sumbernya.

Selanjutnya, nilai *answer relevancy* sebesar 0.190 menunjukkan bahwa tingkat keterkaitan antara masukan pengguna dan hasil rekomendasi masih tergolong rendah, sehingga sistem belum sepenuhnya mampu memahami maksud pengguna secara mendalam. Begitu juga dengan *similarity* sebesar 0.293, yang menandakan bahwa tingkat kemiripan antara destinasi yang diminta dan hasil rekomendasi masih berada pada kategori sedang. Secara keseluruhan, sistem *content based filtering* telah menunjukkan hasil yang konsisten, namun masih perlu pengembangan lebih lanjut agar dapat menghasilkan rekomendasi yang lebih relevan. Selanjutnya berdasarkan hasil percobaan sistem rekomendasi menggunakan *Retrieval Augmented Generation* (RAG) pada pengujian sebelumnya, menghasilkan rekomendasi dan sistem akan secara otomatis menampilkan hasil metrik evaluasi seperti pada gambar dibawah ini.



Berikut adalah 10 rekomendasi wisata sejenis Pantai Baron:

1. Pantai Parangkusumo, Yogyakarta - Terkenal dengan legenda Ratu Kidul dan keindahannya.
2. Pantai Krakal, Yogyakarta - Menawarkan panorama laut dan kuliner laut yang lezat.
3. Pantai Ngrenehan, Wonosari - Suatu pantai nelayan dengan teluk yang indah dan aktivitas kegiatan nelayan yang menarik.
4. Pantai Ria Kenjeran, Jawa Timur - Terkenal dengan keindahan alam dan keunikan desain bangunan ikonnya.
5. Pantai Sape, Lampung - Menawarkan panorama laut yang indah dan aktivitas kegiatan nelayan yang menarik.
6. Pantai Tanjung Bira, Sulawesi Selatan - Terkenal dengan keindahan alam dan keunikan desain bangunan ikonnya.
7. Pantai Kuta, Bali - Menawarkan panorama laut yang indah dan aktivitas kegiatan nelayan yang menarik.
8. Pantai Lovina, Bali - Terkenal dengan keindahan alam dan keunikan desain bangunan ikonnya.
9. Pantai Amed, Bali - Menawarkan panorama laut yang indah dan aktivitas kegiatan nelayan yang menarik.
10. Pantai Banyuwangi, Jawa Timur - Terkenal dengan keindahan alam dan keunikan desain bangunan ikonnya.

Semoga rekomendasi ini membantu Anda dalam mencari destinasi wisata sejenis Pantai Baron!

87.51 detik 21:49

Evaluasi Jawaban:
- Context Recall: 1.000
- Faithfulness: 0.783
- Response Relevancy: 0.617
- Similarity: 0.715

Gambar 14. Hasil metrik evaluasi *Retrieval Augmented Generation* (RAG)

Hasil pengujian terhadap sistem rekomendasi wisata berbasis *Retrieval Augmented Generation* (RAG) menunjukkan bahwa model memiliki kinerja yang cukup baik berdasarkan empat indikator utama, yakni *context recall, faithfulness, answer relevancy*, dan *similarity*. Nilai *Context Recall* sebesar 1.000 menandakan bahwa seluruh informasi penting yang diambil dari dokumen sumber berhasil dimanfaatkan sepenuhnya oleh model saat menghasilkan jawaban. Kemudian, nilai *faithfulness* sebesar 0.783 memperlihatkan bahwa sebagian besar keluaran model tetap sesuai dengan fakta yang terdapat dalam konteks sumber. Dengan demikian, sistem dapat dikatakan cukup andal karena mampu menjaga kesesuaian informasi dan meminimalkan kemungkinan munculnya halusinasi atau kesalahan konten.

Selanjutnya, nilai *answer relevancy* sebesar 0.617 menunjukkan bahwa respons yang diberikan model tergolong cukup relevan terhadap pertanyaan pengguna, meskipun masih terdapat ruang untuk peningkatan agar jawaban menjadi lebih fokus dan tepat sasaran. Hal ini mengindikasikan bahwa pemahaman model terhadap maksud pertanyaan sudah baik, tetapi penyusunan jawaban optimal masih perlu diperbaiki. Adapun nilai *similarity* sebesar 0.715 memperlihatkan bahwa secara semantik, jawaban yang dihasilkan memiliki tingkat kesamaan makna yang cukup tinggi dengan isi dokumen referensi. Dengan kata lain, meskipun terdapat perbedaan dalam struktur kalimat atau redaksi, makna keseluruhan yang disampaikan oleh sistem tetap konsisten dengan konteks sumber.

3.6. Pembahasan

Berdasarkan hasil evaluasi, berikut ini adalah tabel komparasi kinerja antara *content based filtering* dan *Retrieval Augmented Generation* (RAG).

Table 3. Komparasi Kinerja

METODE	*Context Recall*	*Faithfulness*	*Answer Relevancy*	*Similarity*
Content Based Filtering	0,317	1	0,190	0,293
Retrieval Augmented Generation (RAG)	1	0,783	0,617	0,715

Hasil evaluasi terhadap dua pendekatan sistem rekomendasi, yaitu *content based filtering* dan *Retrieval Augmented Generation* (RAG), menunjukkan adanya perbedaan kinerja yang cukup mencolok berdasarkan empat indikator utama: *context recall*, *faithfulness*, *answer relevancy*, dan *similarity*. Pada metode *content based filtering*, nilai *context recall* tercatat sebesar 0,317, yang mengindikasikan bahwa sistem hanya mampu menangkap sebagian kecil dari konteks masukan pengguna. Sebaliknya, metode RAG memperoleh nilai sempurna (1,000), menunjukkan kemampuan yang sangat baik dalam memahami dan mengembalikan konteks yang relevan. Dari sisi *faithfulness*, *content based filtering* memperoleh nilai maksimal (1,000) yang menandakan hasil rekomendasinya sepenuhnya sesuai dengan data sumber tanpa ada penyimpangan. Namun, nilai *faithfulness* pada RAG sedikit lebih rendah (0,783), menandakan adanya kemungkinan interpretasi tambahan dari model yang sedikit mengubah makna aslinya.

Sementara itu, pada metrik *answer relevancy* dan *similarity*, metode RAG memperlihatkan performa yang lebih unggul dibandingkan *content based filtering*. Nilai *answer relevancy* RAG sebesar 0,617 menunjukkan bahwa rekomendasi yang dihasilkan lebih sesuai dengan kebutuhan pengguna, sedangkan *content based filtering* hanya mencapai 0,190, yang menandakan tingkat kesesuaian yang masih rendah. Hal serupa juga terlihat pada metrik *Similarity*, di mana RAG memperoleh nilai 0,715, jauh di atas *content based filtering* yang hanya mencapai 0,293. Temuan ini memperlihatkan bahwa pendekatan RAG memiliki kemampuan yang lebih baik dalam menangkap makna semantik dan menghasilkan rekomendasi yang lebih relevan dengan konteks pengguna. Secara keseluruhan, meskipun *content based filtering* unggul dalam menjaga kesesuaian dengan data sumber (*faithfulness*), metode RAG terbukti lebih efektif dalam memahami konteks, meningkatkan relevansi, dan menghasilkan rekomendasi yang lebih akurat terhadap kebutuhan pengguna.

4. KESIMPULAN

Berdasarkan hasil penelitian yang melibatkan proses pengumpulan data, pengembangan sistem, pengujian, dan evaluasi menggunakan berbagai metrik, dapat disimpulkan bahwa penerapan metode *Content Based Filtering* dan *Retrieval Augmented Generation* (RAG) menawarkan pendekatan yang berbeda dalam menghasilkan rekomendasi destinasi wisata. *Content Based Filtering* terbukti konsisten terhadap data sumber dan memberikan rekomendasi yang stabil berdasarkan kemiripan fitur teks melalui TF-IDF dan *cosine similarity*. Namun, keterbatasannya terletak pada ketidakmampuannya menangkap konteks permintaan pengguna secara lebih luas. Sebaliknya, RAG menunjukkan performa yang lebih unggul pada sebagian besar metrik evaluasi, terutama *context recall*, *answer relevancy*, dan *similarity*, yang mengindikasikan bahwa pendekatan ini mampu memahami maksud pengguna secara lebih mendalam dan memberikan rekomendasi yang lebih kontekstual. Temuan ini menunjukkan bahwa sistem rekomendasi berbasis RAG berpotensi diterapkan secara lebih luas pada platform pariwisata yang memerlukan pemahaman konteks bahasa alami, seperti chatbot wisata, agen perjalanan digital, atau aplikasi pencarian destinasi cerdas.

Implikasi dari hasil penelitian ini mengarah pada peluang peningkatan kualitas layanan rekomendasi yang tidak hanya bersifat statis tetapi juga adaptif terhadap kebutuhan pengguna. Secara spekulatif, jika model RAG terus dikembangkan menggunakan corpus yang lebih besar dan domain yang lebih beragam, sistem ini berpotensi memberikan rekomendasi yang semakin personal, deskriptif, dan interaktif, bahkan mampu menyesuaikan gaya rekomendasi berdasarkan preferensi pengguna. Di sisi lain, nilai faithfulness RAG yang lebih rendah dibandingkan CBF juga mengindikasikan perlunya mekanisme kontrol atau grounding yang lebih ketat agar hasil rekomendasi tidak menyimpang dari data yang tersedia. Oleh karena itu, penelitian selanjutnya disarankan untuk mengeksplorasi integrasi antara kedua pendekatan, misalnya hybrid CBF–RAG, penggunaan *reranking* berbasis *embedding* untuk meningkatkan konsistensi, atau pemanfaatan model *fine-tuned* RAG dengan domain wisata lokal untuk meningkatkan keterkaitan antara respons dan sumber data. Dengan demikian, simpulan penelitian ini tidak hanya menegaskan keunggulan RAG dalam memahami konteks, tetapi juga membuka peluang pengembangan sistem rekomendasi yang lebih cerdas, adaptif, dan tetap akurat dalam berbagai skenario penerapan nyata.

5. DAFTAR PUSTAKA

[1] S. Shehmir and R. Kashef, "LLM4Rec: A Comprehensive Survey on the Integration of Large Language Models in Recommender Systems," *Futur. Internet*, vol. 17, no. 6, p. 252, 2025, doi: https://doi.org/10.3390/fi17060252.

[2] Q. Peng, H. Liu, H. Huang, Q. Yang, and M. Shao, “A Survey on LLM-powered Agents for Recommender Systems,” *arXiv Prepr. arXiv2502.10050*, 2025, doi: https://doi.org/10.48550/arXiv.2502.10050.

[3] W. Fan *et al.*, “A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models,” *arXiv Prepr. arXiv2405.06211*, 2024, doi: https://doi.org/10.48550/arXiv.2405.06211.

[4] H. Yu *et al.*, “Evaluation of Retrieval-Augmented Generation: A Survey,” *arXiv Prepr. arXiv2405.07437*, 2024, doi: https://doi.org/10.48550/arXiv.2405.07437.

[5] M. Abo El-Enen and others, “A Survey on Retrieval-Augmented Generation (RAG),” *Neural Comput. Appl.*, 2025, doi: https://doi.org/10.1007/s00521-025-11666-9.

[6] A. Banerjee, A. Satish, and W. Wörndl, “Enhancing Tourism Recommender Systems for Sustainable City Trips Using Retrieval-Augmented Generation,” in *Proceedings*, 2024. doi: https://doi.org/10.48550/arXiv.2409.18003.

[7] J. Qi *et al.*, “RAG-Optimized Tibetan Tourism LLMs: Enhancing Accuracy and Personalization,” *arXiv Prepr. arXiv2408.12003*, 2024, doi: https://doi.org/10.48550/arXiv.2408.12003.

[8] R. Karlović and others, “Context-Aware Tourism Recommendations Using Retrieval-Augmented Generation,” *Electronics*, vol. 14, no. 22, 2025, doi: https://doi.org/10.3390/electronics14224448.

[9] C. Wei and others, “Enhanced Recommendation Systems with Retrieval-Augmented Generation,” *J. Artif. Intell. Res.*, 2025, doi: https://doi.org/10.1613/jair.1.17809.

[10] S. Luo, J. Xu, X. Zhang, and others, “RALLRec+: Retrieval Augmented Large Language Model Recommendation with Reasoning,” *arXiv Prepr. arXiv2503.20430*, 2025, doi: https://doi.org/10.48550/arXiv.2503.20430.

[11] S. Wang, W. Fan, Y. Feng, and others, “Knowledge-Graph Retrieval-Augmented Generation for LLM-based Recommendation (K-RagRec),” in *ACL*, 2025.

[12] D. Gavalas, C. Konstantopoulos, and K. Mastakas, “Mobile Recommender Systems in Tourism,” pp. 1–30.

[13] N. Tintarev and J. Masthoff, “Explaining Recommendations: Design and Evaluation,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds., Boston, MA: Springer US, 2015, pp. 353–382. doi: 10.1007/978-1-4899-7637-6_10.

[14] M. (2015). Xiang, Z., Schwartz, Z., Gerdes, J. H., & Uysal, “What can big data and text analytics tell us about hotel guest experience and satisfaction?,” p. 2015, 2015, doi: https://doi.org/10.1016/j.ijhm.2014.10.013.

[15] L. Wu *et al.*, “A Survey on Large Language Models for Recommendation,” pp. 1–34.