

BAB II

LANDASAN TEORI

2.1 Posisi Penelitian

Posisi penelitian merupakan bagian yang menggambarkan kedudukan penelitian ini di antara penelitian-penelitian terdahulu yang relevan dengan topik yang dikaji. Bagian ini disusun berdasarkan hasil penelusuran literatur ilmiah, terutama dari jurnal nasional dan internasional yang membahas sistem rekomendasi, metode *Content-Based Filtering*, teknik TF-IDF dan *Cosine Similarity*, pengolahan teks Bahasa Indonesia, serta penerapannya pada domain pariwisata. Melalui penelusuran tersebut, dapat diidentifikasi metode, objek, dan pendekatan yang telah digunakan oleh peneliti sebelumnya, sekaligus ditemukan celah penelitian yang menjadi peluang bagi penelitian ini untuk memberikan kontribusi baru.

Kajian terhadap penelitian terdahulu dilakukan dengan meninjau lima belas penelitian yang dipublikasikan dalam kurun waktu 2021 hingga 2026. Penelitian-penelitian tersebut dipilih karena memiliki keterkaitan langsung dengan metode maupun objek yang digunakan dalam penelitian ini, sehingga dapat dijadikan acuan untuk menentukan kebaruan dan posisi penelitian yang diusulkan. Ringkasan penelitian-penelitian terdahulu tersebut disajikan pada Tabel 2.1, sedangkan penelitian yang diusulkan dalam skripsi ini disajikan pada Tabel 2.2.

Tabel 2. 1 *State Of The Art*

No	Penulis	Judul	Metode	Objek Penelitian	Isi
1	Prasetyo, B., Atina, V., & Purwanto, E. Vol 14(1), Hal 51-58, 2021 ISSN: 2086- 9436	Sistem Rekomendasi Pariwisata Dengan Metode <i>Content Based Recommenda</i>	<i>Content- Based Filtering</i>	Pariwisata (Surakarta)	Membangun sistem rekomendasi pariwisata di Surakarta berbasis <i>website</i> . Sistem memberikan saran destinasi sesuai preferensi pengguna berdasarkan

No	Penulis	Judul	Metode	Objek Penelitian	Isi
	DOI : 10.47701/dutacom.v14i1.2017 https://doi.org/10.47701/dutacom.v14i1.2017 https://ojs.udb.ac.id/index.php/dutacom/article/view/2017	tion Berbasis Website			kemiripan karakteristik antar tempat wisata yang ada di kota tersebut.
2	Roy, D., & Dutta, M. Vol 9(1), Hal no. 59, 2022 ISSN: 2196-1115 DOI: 10.1186/s40537-022-00592-5 https://doi.org/10.1186/s40537-022-00592-5 https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00592-5	<i>A Systematic Review and Research Perspective on Recommender Systems</i>	<i>Content-Based, Collaborative Filtering (Review)</i>	Sistem rekomendasi umum	Melakukan tinjauan sistematis terhadap perkembangan sistem rekomendasi dari berbagai pendekatan, meliputi <i>Content-Based Filtering</i> , <i>Collaborative Filtering</i> , dan <i>Hybrid</i> . Penelitian ini menjelaskan tantangan, peluang, serta arah pengembangan sistem rekomendasi di berbagai bidang.
3	Huda, A. A., Fajarudin, R., & Hadinegoro, A. Vol 4(3), Hal 1679-1686, 2022 ISSN: 2685-3310 DOI : 10.47065/bits.v4i3.2511	Sistem Rekomendasi <i>Content-Based Filtering</i> Menggunakan <i>TF-IDF Vector Similarity</i>	TF-IDF, <i>Cosine Similarity</i>	Artikel berita	Mengimplementasikan metode <i>Content-Based Filtering</i> yang dikombinasikan dengan TF-IDF dan <i>Cosine Similarity</i> untuk merekomendasikan artikel berita. Hasil penelitian menunjukkan bahwa metode TF-IDF efektif untuk mengukur kemiripan antar dokumen berita.

No	Penulis	Judul	Metode	Objek Penelitian	Isi
	https://doi.org/10.47065/bits.v4i3.2511 https://ejurnal.seminar-id.com/index.php/bits/article/view/2511				
4	Faurina, R., & Sitanggang, E. Vol 22(4), Hal 870-881, 2023 ISSN: 2356-2579 DOI : 10.33633/tc.v22i4.8556 https://doi.org/10.33633/tc.v22i4.8556 https://publikasi.dinus.ac.id/index.php/technoc/article/view/8556	Implementasi Metode <i>Content-Based Filtering dan Collaborative Filtering</i> pada Sistem Rekomendasi Wisata di Bali	CBF + CF	Wisata (Bali)	Menggabungkan pendekatan dalam membangun sistem rekomendasi destinasi wisata di Bali. Pendekatan <i>Hybrid</i> terbukti dapat meningkatkan akurasi rekomendasi dibandingkan jika hanya menggunakan satu metode saja.
5	Yulita, W., Untoro, M. C., Praseptiawan, M., Ashari, I. F., Afriansyah, A., & Pee, A. N. B. C. Vol 10(2), Hal 93–104, 2023 ISSN: 2460-0040 DOI : 10.15294/sji.v10i2.42209	<i>Automatic Scoring Using Term Frequency Inverse Document Frequency and Cosine Similarity</i>	TF-IDF, <i>Cosine Similarity</i>	Penilaian teks otomatis	membangun sistem penilaian teks otomatis. Penelitian ini membuktikan bahwa kombinasi kedua metode dapat menghitung kemiripan antar dokumen teks dengan akurat dan cocok diterapkan pada berbagai jenis data berbasis teks.

No	Penulis	Judul	Metode	Objek Penelitian	Isi
	https://doi.org/10.15294/sji.v10i2.42209 https://journal.unnes.ac.id/nju/index.php/sji/article/view/42209				
6	Saputra, D. M., Angelia, N., & Yusliani, N. Vol 5(3), Hal 122-127, 2024 ISSN: 2722-2934 DOI 10.62527/jitsi.5.3.254 https://doi.org/10.62527/jitsi.5.3.254 https://jurnal-itsi.org/index.php/jitsi/article/view/254	<i>Recommender System for Tourist Destinations in Indonesia Using Matrix Factorization Method</i>	<i>Matrix Factorization</i>	Wisata (Indonesia)	Membangun sistem rekomendasi destinasi wisata di Indonesia menggunakan metode <i>Matrix Factorization</i> . Sistem ini menggunakan pendekatan berbasis interaksi pengguna untuk menghasilkan saran destinasi yang relevan secara personal.
7	Albab, J. F. U., & Rohman, A. N. Vol 10(1), Hal 935-945, 2026 ISSN: 2548-6861 DOI : 10.30871/jaic.v10i1.11751 https://doi.org/10.30871/jaic.v10i1.11751 https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/11751	<i>Recommendation System Yogyakarta Tourism Using TF-IDF and Cosine Similarity Methods with Word Normalizer</i>	TF-IDF, Cosine Similarity, Word Normalizer	Wisata (Yogyakarta)	Membangun sistem rekomendasi yang dilengkapi <i>Word Normalizer</i> . Penambahan <i>Word Normalizer</i> , akurasi sistem dalam menangani variasi kata yang tidak baku pada deskripsi destinasi wisata.

No	Penulis	Judul	Metode	Objek Penelitian	Isi
8	Setiawan, G. H., & Adnyana, I. M. B. Vol 7(2), Hal 252-257, 2023 ISSN: 2548- 6861 DOI : 10.30871/jaic.v7 i2.6527 https://doi.org/10.30871/jaic.v7i2.6527 https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/6527	<i>Improving Helpdesk Chatbot Performance with TF-IDF and Cosine Similarity Models</i>	TF-IDF, <i>Cosine Similarity</i>	<i>Chatbot helpdesk</i>	Mengembangkan <i>chatbot helpdesk</i> sistem mampu menghasilkan jawaban relevan terhadap pertanyaan pengguna berdasarkan kemiripan teks dengan basis pengetahuan yang telah disiapkan sebelumnya.
9	Christyawan, F., Rohman, A. N., & Hartanto, A. D. Vol 6(3), Hal 1559-1576, 2024 ISSN: 2656- 4882 DOI : 10.51519/journali.v6i3.806 https://doi.org/10.51519/journali.v6i3.806 https://journal-isi.org/index.php/isi/article/view/806	<i>Application of Content-Based Filtering Method Using Cosine Similarity in Restaurant Selection Recommendation System</i>	CBF, <i>Cosine Similarity</i>	Restoran	Menerapkan <i>Content-Based Filtering</i> dengan <i>Cosine Similarity</i> untuk membangun sistem rekomendasi restoran. Sistem memberikan saran restoran berdasarkan kemiripan karakteristik seperti jenis menu, lokasi, dan harga sesuai preferensi pengguna.
10	Nandita, I. A. W., Suarjaya, I. M. A. D., &	<i>News Recommendation System</i>	CBF, TF-IDF,	Berita	Membangun sistem rekomendasi melalui layanan kustomisasi

No	Penulis	Judul	Metode	Objek Penelitian	Isi
	Bayupati, I. P. A. Vol 9(4), Hal 1858-1867, 2025 ISSN: 2548-6861 DOI : 10.30871/jaic.v9i4.9807 https://doi.org/10.30871/jaic.v9i4.9807 https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/9807	<i>Using Content-Based Filtering through RSS Customization Service</i>	<i>Cosine Similarity</i>		RSS. Sistem berhasil memberikan saran berita yang sesuai dengan minat pembaca berdasarkan konten yang pernah dibaca.
11	Wicaksono, A. W., Rohman, A. N., & Hartanto, A. D. Vol 6(3), Hal 1598-1615, 2024 ISSN : 2656-4882 DOI : 10.51519/journalisi.v6i3.813 https://doi.org/10.51519/journalisi.v6i3.813 https://journal-isi.org/index.php/isi/article/view/813	<i>Café Recommendation Using the CBF Method</i>	CBF, <i>Cosine Similarity</i>	Kafe (Yogyakarta)	Mengembangkan sistem rekomendasi kafe di Yogyakarta dengan metode <i>Content-Based Filtering</i> dan <i>Cosine Similarity</i> . Sistem ini memanfaatkan deskripsi dan atribut kafe untuk menghasilkan rekomendasi tempat yang sesuai dengan preferensi pengguna.
12	Pasaribu, Y. S., & Sitompul, T. S.	Rekomendasi Destinasi Wisata Kota Bandung Menggunakan	<i>Collaborative Filtering</i>	Wisata (Bandung)	Membangun sistem rekomendasi destinasi wisata Kota Bandung dengan algoritma

No	Penulis	Judul	Metode	Objek Penelitian	Isi
	Vol 1(6), Hal 382-392, 2023 ISSN : 2986-7029 DOI : 10.59059/mutiara.v1i6.736 https://doi.org/10.59059/mutiara.v1i6.736 https://journal.staiyapiqbaubau.ac.id/index.php/Mutiara/article/view/736	n Algoritma Collaborative Filtering			<i>Collaborative Filtering</i> . Sistem memberikan saran destinasi berdasarkan pola perilaku wisatawan lain yang memiliki preferensi serupa, bukan berdasarkan karakteristik destinasi.
13	Al Rasyid, R., & Ningsih, D. H. U. Vol 8(1), Hal 170-178, 2024 ISSN : 2580-1643 DOI : 10.35870/jtik.v8i1.1416 https://doi.org/10.35870/jtik.v8i1.1416 https://journal.lembagakita.com/index.php/jtik/article/view/1416	Penerapan Algoritma TF-IDF dan <i>Cosine Similarity</i> untuk <i>Query</i> Pencarian Pada <i>Dataset</i> Destinasi Wisata	TF-IDF, <i>Cosine Similarity</i>	Wisata (Semarang)	Menerapkan TF-IDF dan <i>Cosine Similarity</i> untuk <i>query</i> pencarian pada <i>dataset</i> destinasi wisata Semarang.
14	Lubis, A. R., Lase, Y. Y., Rahman, D. A., & Witarsyah, D. Vol 28(5), Hal 1335-1342, 2023	<i>Improving Spell Checker Performance for Bahasa Indonesia Using Text Preprocessin</i>	<i>Case Folding, Tokenization, Stemming, Stopword Removal</i>	NLP Bahasa Indonesia	Meneliti penerapan teknik <i>Text Preprocessing</i> seperti <i>Case Folding, tokenization, stemming, dan stopwords removal</i> yang dikombinasikan

No	Penulis	Judul	Metode	Objek Penelitian	Isi
	ISSN : 1633-1311 DOI : 10.18280/isi.280522 https://doi.org/10.18280/isi.280522 https://www.iiet.a.org/journals/isi/paper/10.18280/isi.280522	<i>g Techniques with Deep Learning Models</i>			dengan <i>deep learning</i> untuk meningkatkan performa <i>spell checker</i> Bahasa Indonesia. Penelitian ini menegaskan pentingnya <i>Preprocessing</i> pada pengolahan teks Bahasa Indonesia.
15	Salsabilla, C., & Utomo, D. W. Vol 14(1), Hal 262-270, 2025 ISSN : 2540-9719 DOI : 10.32520/stmsi.v14i1.4839 https://doi.org/10.32520/stmsi.v14i1.4839 https://sistemasi.ftik.unisi.ac.id/index.php/stmsi/article/view/4839	<i>Pekalongan Regency Tourism Recommendation System with Content based Filtering</i>	CBF, TF-IDF, <i>Cosine Similarity</i>	Wisata (Pekalongan)	Membangun sistem rekomendasi wisata Kabupaten Pekalongan menggunakan <i>Content-Based Filtering</i> , TF-IDF, dan <i>Cosine Similarity</i> . Sistem berhasil memberikan rekomendasi destinasi wisata yang relevan berdasarkan kemiripan deskripsi tempat wisata yang ada di wilayah tersebut.

Tabel 2. 2 Penelitian yang Diusulkan

No	Penulis	Judul	Metode	Objek Penelitian	Isi
1	Mira Aldina Program Studi Teknik Informatika Fakultas Teknik dan Pertanian	Sistem Rekomendasi Destinasi Wisata Bandung Raya Berbasis <i>Website</i> dengan	CBF + TF-IDF + <i>Cosine Similarity</i>	Wisata Bandung Raya (331 destinasi)	Membangun sistem rekomendasi destinasi wisata Bandung Raya berbasis <i>website</i> dengan pendekatan <i>Machine Learning</i> menggunakan metode <i>Content-</i>

No	Penulis	Judul	Metode	Objek Penelitian	Isi
	Universitas Islam Nusantara Tahun 2026	Pendekatan <i>Machine Learning</i> Menggunakan Metode <i>Content-Based Filtering</i>			<i>Based Filtering</i> , TF-IDF, dan <i>Cosine Similarity</i> . Penelitian ini mengintegrasikan seluruh tahapan dari <i>Preprocessing</i> teks Bahasa Indonesia menggunakan Sastrawi, pembobotan TF-IDF, hingga perhitungan <i>Cosine Similarity</i> dalam satu sistem yang utuh dengan <i>dataset</i> 331 destinasi unik di Bandung Raya.

Berdasarkan Tabel 2.1, dapat diketahui bahwa penelitian terkait sistem rekomendasi telah banyak dilakukan dengan berbagai pendekatan. Penelitian Roy & Dutta (2022) membahas perkembangan sistem rekomendasi secara umum, sedangkan penelitian Huda *et al.* (2022), Yulita *et al.* (2023), Al Rasyid, R., & Ningsih (2024), serta Albab, J. F. U., & Rohman (2026) membuktikan bahwa teknik TF-IDF dan *Cosine Similarity* mampu dimanfaatkan untuk mengukur tingkat kesamaan antar dokumen teks dengan baik.

Meskipun demikian, masih terdapat celah dari penelitian-penelitian tersebut. Teknik TF-IDF dan *Cosine Similarity* pada banyak penelitian justru diterapkan pada objek di luar sistem rekomendasi pariwisata berbasis *website* yang utuh, seperti Huda *et al.* (2022) pada artikel berita, Yulita *et al.* (2023) pada penilaian teks otomatis, Setiawan & Adnyana (2023) pada *chatbot helpdesk*, serta Al Rasyid, R., & Ningsih (2024) pada *query* pencarian *dataset* destinasi wisata. Penelitian-penelitian tersebut menekankan proses perhitungan kemiripan, tetapi belum mengintegrasikan keseluruhan tahapan metode ke dalam satu sistem rekomendasi wisata berbasis *website* yang interaktif. Di samping itu, Lubis *et al.* (2023) menegaskan bahwa tahapan *Text Preprocessing* seperti *Case Folding*,

tokenization, *stopword removal*, dan *stemming* berpengaruh terhadap kualitas pengolahan teks, sehingga tahapan tersebut perlu diintegrasikan secara menyeluruh dalam sistem, bukan sebagai proses yang berdiri sendiri.

Dari sisi objek penelitian, sebagian besar sistem rekomendasi wisata di Indonesia masih difokuskan pada wilayah selain Bandung Raya. Faurina & Sitanggang (2023) mengembangkan sistem rekomendasi untuk wisata Bali, Albab, J. F. U., & Rohman (2026) untuk wisata Yogyakarta, Al Rasyid, R., & Ningsih (2024) untuk wisata Semarang, dan Salsabilla, C., & Utomo (2025) untuk wisata Kabupaten Pekalongan, sementara Saputra, D. M., Angelia, N., & Yusliani (2024) mengangkat destinasi wisata di Indonesia secara umum. Penelitian yang mengangkat destinasi wisata Bandung memang sudah ada, yaitu Pasaribu, Y. S., & Sitompul (2023), tetapi penelitian tersebut menggunakan pendekatan *Collaborative Filtering*, bukan *Content-Based Filtering*.

Berdasarkan celah tersebut, kebaruan dalam studi ini terletak pada pembangunan mesin rekomendasi destinasi wisata Bandung Raya berbasis *website* dengan pendekatan *Machine Learning* menggunakan metode *Content-Based Filtering*. Studi ini mengintegrasikan seluruh tahapan pemrosesan teks, mulai dari *Text Preprocessing* Bahasa Indonesia menggunakan Sastrawi, pembobotan kata menggunakan TF-IDF, hingga komputasi tingkat kesamaan menggunakan *Cosine Similarity*. Di samping itu, studi ini memanfaatkan dataset sebanyak 331 destinasi wisata yang meliputi wilayah Kota Bandung, Kabupaten Bandung, dan Kabupaten Bandung Barat, sehingga sistem yang dibangun mampu memberikan rekomendasi destinasi wisata yang lebih sesuai dengan preferensi pengguna.

2.2 Pariwisata

2.2.1 Definisi Pariwisata

Pariwisata termasuk salah satu bidang yang turut menopang perekonomian Indonesia. Menurut Undang-Undang Republik Indonesia Nomor 10 Tahun 2009 tentang Kepariwisata, pariwisata didefinisikan sebagai berbagai macam kegiatan wisata yang didukung oleh fasilitas serta layanan yang disediakan oleh masyarakat,

pengusaha, pemerintah, dan pemerintah daerah (Undang-Undang RI Nomor 10 Tahun 2009).

Definisi lain menyebutkan bahwa pariwisata merupakan aktivitas perpindahan seseorang dari tempat tinggalnya ke tempat lain untuk tujuan rekreasi, bisnis, maupun keperluan lain dengan masa tinggal tidak lebih dari satu tahun (Prasetyo *et al.*, 2021). Pariwisata juga dipandang sebagai kegiatan yang dilakukan wisatawan untuk mengisi waktu luang dengan mengunjungi destinasi tertentu (Saputra *et al.*, 2024).

2.2.2 Destinasi Wisata

Destinasi wisata adalah tempat atau kawasan geografis yang menjadi tujuan kunjungan wisatawan karena memiliki daya tarik dan fasilitas pendukung tertentu (Faurina & Sitanggang, 2023). Berdasarkan karakteristiknya, destinasi wisata dapat dikelompokkan menjadi beberapa jenis, antara lain wisata alam, budaya, buatan, religi, edukasi, dan kuliner. Tiap jenis destinasi memiliki ciri khas tersendiri yang menarik minat kelompok wisatawan tertentu (Albab, J. F. U., & Rohman, 2026).

2.2.3 Pariwisata Bandung Raya

Dalam penelitian ini, cakupan wilayah Bandung Raya difokuskan pada tiga daerah, yaitu Kota Bandung, Kabupaten Bandung, dan Kabupaten Bandung Barat. Sebagai bagian dari kawasan Bandung, ketiga wilayah tersebut menawarkan banyak pilihan destinasi, sehingga wisatawan kerap kesulitan menentukan tempat yang sesuai dengan keinginan mereka (Pasaribu, Y. S., & Sitompul, 2023).

Destinasi wisata di Bandung Raya cukup bervariasi, mulai dari wisata alam seperti Kawah Putih dan Tangkuban Perahu, wisata edukasi seperti Museum Geologi, hingga wisata kuliner dan pusat perbelanjaan. Berdasarkan data Dinas Kebudayaan dan Pariwisata Kota Bandung, jumlah kunjungan wisatawan ke Kota Bandung pada tahun 2023 mencapai 7,7 juta orang, meningkat sekitar 17 persen dari tahun sebelumnya (Sutrisno, 2025).

2.3 Sistem Rekomendasi

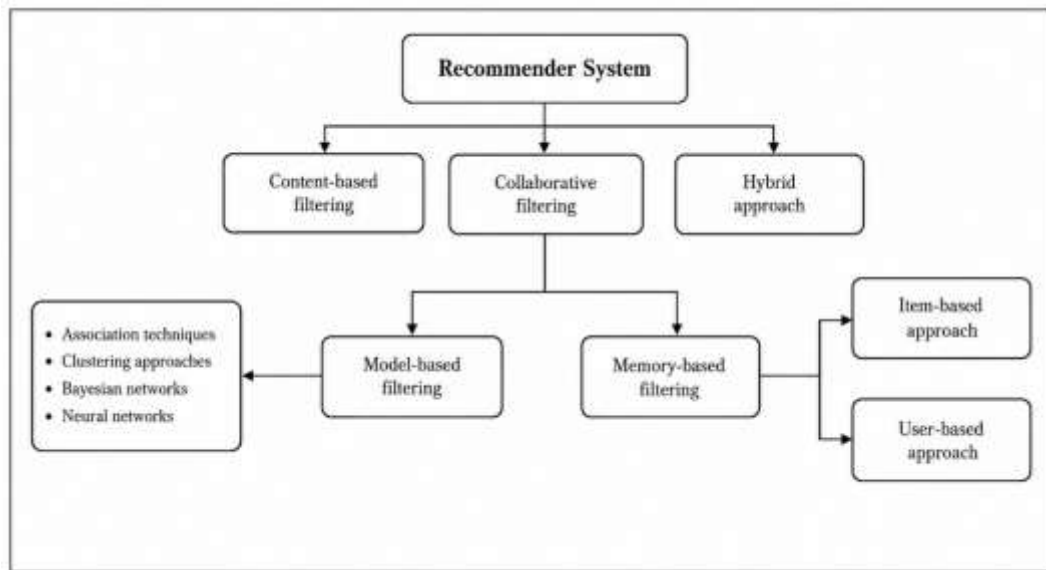
Sistem yang dirancang untuk menyaring informasi dari sejumlah besar data dan memberikan saran yang relevan kepada pengguna sesuai preferensinya. Penerapannya telah banyak ditemukan di berbagai bidang seperti *e-commerce*, hiburan, dan pariwisata, karena mampu membantu pengguna menemukan informasi yang dibutuhkan secara lebih cepat (Roy & Dutta, 2022). Tujuan utama dari sistem rekomendasi adalah memperkirakan item yang kemungkinan besar akan diminati oleh pengguna berdasarkan data yang telah tersedia, sehingga mengurangi waktu yang dihabiskan untuk mencari informasi secara manual (Saputra *et al.*, 2024).

Pada umumnya, sistem rekomendasi bekerja dengan cara menganalisis pola data, baik dari karakteristik item itu sendiri maupun dari interaksi pengguna, kemudian hasil analisis tersebut digunakan untuk menghasilkan daftar saran yang dipersonalisasi. Terdapat tiga jenis pendekatan utama dalam sistem rekomendasi,

yaitu *Content-Based Filtering*, *Collaborative Filtering*, dan *Hybrid Filtering*, yang masing-masing memiliki cara kerja berbeda tergantung pada jenis data yang tersedia dan kebutuhan sistem yang dibangun (Huda *et al.*, 2022).

Content-Based Filtering memberikan rekomendasi dari atribut di setiap item, sementara *Collaborative Filtering* lebih menekankan pada pola perilaku pengguna lain yang memiliki kemiripan preferensi, dan *Hybrid Filtering* menggabungkan kedua pendekatan tersebut untuk meningkatkan akurasi rekomendasi serta pemilihannya bergantung pada konteks penelitian dan ketersediaan data yang dimiliki (Faurina & Sitanggang, 2023).

Secara umum, sistem rekomendasi dapat diklasifikasikan ke dalam tiga pendekatan utama yaitu:



Gambar 2. 1 Klasifikasi Sistem Rekomendasi

Sumber (Roy & Dutta, 2022)

Untuk menegaskan alasan pemilihan *Content-Based Filtering* perlu dibandingkan dengan pendekatan rekomendasi yang umum digunakan, yaitu *Collaborative Filtering* dan *Hybrid Filtering*. Ketiga pendekatan tersebut memiliki dasar rekomendasi, kebutuhan data, serta kelebihan dan kekurangan yang berbeda-beda sesuai konteks penggunaannya (Faurina & Sitanggang, 2023). Perbandingan ketiga metode tersebut disajikan pada Tabel 2.3 berikut:

Tabel 2. 3 Perbandingan Metode Rekomendasi

No	Aspek	<i>Content-Based Filtering</i> (metode penelitian ini)	<i>Collaborative Filtering</i>	<i>Hybrid Filtering</i>
1	Dasar rekomendasi	Kemiripan atribut atau karakteristik antar item	Pola interaksi dan preferensi banyak pengguna	Gabungan atribut item dan interaksi pengguna
2	Kebutuhan data	Deskripsi atau atribut item tanpa data pengguna lain	Data rating atau riwayat banyak pengguna	Atribut item sekaligus data interaksi pengguna
3	Masalah <i>cold-start</i>	Relatif kecil yaitu item baru tetap bisa direkomendasikan	Besar yaitu lemah untuk pengguna/item baru	Lebih kecil yaitu kedua pendekatan saling menutupi
4	Kelebihan	Tetap berjalan walau data pengguna sedikit	Rekomendasi cenderung lebih beragam	Akurasi cenderung lebih tinggi

No	Aspek	<i>Content-Based Filtering</i> (metode penelitian ini)	<i>Collaborative Filtering</i>	<i>Hybrid Filtering</i>
5	Kekurangan	Variasi terbatas; bergantung kualitas deskripsi	Membutuhkan data interaksi dalam jumlah besar	Implementasi lebih kompleks
6	Contoh penerapan	Prasetyo, B., Atina, V., & Purwanto (2021); Wicaksono, A. W., Rohman, A. N., & Hartanto (2024); Salsabilla, C., & Utomo (2025)	Pasaribu, Y. S., & Sitompul (2023)	Faurina & Sitanggang (2023)

2.4 *Content-Based Filtering*

Adalah salah satu metode dalam sistem rekomendasi yang bekerja dengan memanfaatkan atribut item, yakni memberikan rekomendasi berdasarkan kemiripan antara satu item dengan item lainnya tanpa interaksi pengguna lain, sistem akan merekomendasikan item lain yang memiliki karakteristik serupa (Prasetyo, B., Atina, V., & Purwanto, 2021).

Dalam praktiknya, *Content-Based Filtering* cocok diterapkan ketika data interaksi pengguna terbatas karena rekomendasi sepenuhnya bergantung pada deskripsi atau atribut item itu sendiri, sehingga hal ini menjadi keunggulannya pada sistem rekomendasi yang baru dibangun dan belum memiliki banyak data pengguna (Christyawan, F., Rohman, A. N., & Hartanto, 2024).

Secara umum, *Content-Based Filtering* dibangun dari tiga komponen utama, yaitu representasi item (*item representation*), profil pengguna (*user profile*), dan komponen pencocokan. Representasi item mengubah setiap item menjadi sekumpulan fitur yang dapat diukur, misalnya kata-kata penting yang terdapat pada deskripsi sebuah destinasi wisata. Profil pengguna menggambarkan preferensi pengguna yang dibentuk dari item yang sebelumnya dipilih atau diminati. Selanjutnya, komponen pencocokan membandingkan profil pengguna dengan representasi tiap item untuk menentukan item yang paling relevan untuk direkomendasikan (Roy & Dutta, 2022).

Tahapan kerja *Content-Based Filtering* umumnya meliputi tiga proses utama, yaitu ekstraksi fitur dari setiap item pada *dataset*, representasi item dalam

bentuk vektor numerik agar dapat diolah secara matematis, dan perhitungan kemiripan antaritem menggunakan teknik tertentu seperti *Cosine Similarity*, dengan keluaran berupa daftar item yang paling mirip dengan item yang menjadi acuan pengguna (Wicaksono *et al.*, 2024).

Kelebihan *Content-Based Filtering* adalah tidak bergantung pada data interaksi pengguna lain sehingga tetap dapat berjalan meskipun jumlah pengguna masih sedikit, serta mampu merekomendasikan item baru selama deskripsi atau atributnya tersedia (Prasetyo *et al.*, 2021). Adapun keterbatasannya, kualitas rekomendasi sangat dipengaruhi oleh kelengkapan dan kualitas deskripsi item, serta cenderung menghasilkan rekomendasi yang mirip dengan item yang telah disukai sehingga keragamannya terbatas (Roy & Dutta, 2022).

2.5 *Machine Learning*

Machine Learning (ML) merupakan salah satu cabang kecerdasan buatan yang membuat sistem belajar dari data tanpa diprogram secara eksplisit, sehingga mampu mengenali pola, membuat prediksi, dan mengambil keputusan berdasarkan data yang telah dipelajari, termasuk dalam proses analisis data pada sistem rekomendasi agar menghasilkan rekomendasi yang mengacu pada preferensi pengguna (Roy & Dutta, 2022).

Secara umum, *Machine Learning* dikelompokkan menjadi dua paradigma utama, yaitu *Supervised Learning* yang menggunakan data berlabel sebagai acuan dalam proses pelatihan untuk prediksi dan klasifikasi, serta *Unsupervised Learning* yang bekerja pada data tanpa label untuk menemukan pola atau pengelompokan tersembunyi (Nurhalizah *et al.*, 2024).

Penerapan *Machine Learning* pada sistem rekomendasi destinasi wisata umumnya menggunakan pendekatan *Unsupervised Learning* karena tidak memerlukan label sebagai acuan, sehingga sistem cukup mempelajari karakteristik deskripsi destinasi kemudian menghitung tingkat kemiripannya untuk menghasilkan rekomendasi, dan pendekatan ini sangat cocok diterapkan pada data teks berjumlah besar yang tidak memiliki label tertentu (Nandita *et al.*, 2025).

2.6 *Natural Language Processing (NLP)*

Natural Language Processing (NLP) merupakan salah satu cabang *artificial intelligence* yang berfokus pada interaksi antara komputer dengan bahasa manusia, dengan tujuan memungkinkan komputer memahami, menafsirkan, dan menghasilkan bahasa manusia dalam bentuk teks maupun suara secara bermakna melalui penggabungan linguistik komputasional dengan *Machine Learning* (Jurafsky *et al.*, 2025).

NLP terdiri atas beberapa komponen utama, yaitu *morphological analysis* (analisis bentuk kata), *syntactic analysis* (analisis struktur kalimat), *semantic analysis* (analisis makna), dan *pragmatic analysis* (analisis konteks), serta dalam praktiknya diterapkan pada berbagai aplikasi seperti penerjemahan otomatis, *chatbot*, analisis sentimen, *spam filtering*, *question answering*, peringkasan dokumen, hingga sistem rekomendasi berbasis konten (Hasan *et al.*, 2023).

Dalam sistem rekomendasi, NLP berperan mengolah deskripsi teks suatu item agar kemiripannya dapat dihitung secara matematis dengan item lain maupun dengan *query* pengguna, sebagaimana diterapkan pada sistem rekomendasi musik yang memproses teks masukan pengguna menggunakan teknik NLP sebelum dihitung kemiripannya dengan koleksi musik yang tersedia (Iqbal *et al.*, 2023).

Bahasa Indonesia memiliki tantangan tersendiri dalam pemrosesan NLP karena kaya dengan kata berimbuhan (afiksasi) seperti prefiks, sufiks, infiks, dan konfiks yang dapat mengubah makna kata, serta banyaknya *stopword* khas Bahasa Indonesia seperti "di", "yang", "dan", dan "dengan" yang perlu dihilangkan agar pemrosesan teks menjadi lebih efektif, sehingga dikembangkan *library* khusus seperti Sastrawi untuk menangani karakteristik morfologi Bahasa Indonesia tersebut (Amien, 2023).

Pada penelitian ini, NLP digunakan untuk mengolah deskripsi destinasi wisata Bandung sehingga dapat dihitung kemiripan antardestinasi maupun antara *query* pengguna dengan destinasi pada *dataset*. Tahapan NLP yang diterapkan meliputi *text Preprocessing*, pembobotan kata menggunakan TF-IDF, dan

perhitungan kemiripan menggunakan *Cosine Similarity* yang akan dijelaskan pada subbab berikutnya.

2.7 Text Preprocessing

Text Preprocessing merupakan tahapan awal dalam pemrosesan data teks sebelum dimanfaatkan dalam analisis lebih lanjut, yang bertujuan membersihkan dan menyederhanakan teks yang pada umumnya tidak terstruktur dan mengandung banyak noise seperti tanda baca, simbol, angka, atau huruf kapital yang tidak konsisten, agar lebih mudah diproses oleh algoritma dan menghasilkan analisis yang lebih tepat (Lubis *et al.*, 2023).

Pada penelitian ini, tahapan *Text Preprocessing* yang digunakan meliputi *Case Folding*, *cleaning*, *tokenization*, *stopword removal*, dan *stemming*. Kelima tahapan tersebut saling melengkapi dan menghasilkan teks yang lebih bersih, ringkas, serta siap diproses oleh algoritma TF-IDF.

2.7.1. Case Folding

Proses mengubah semua huruf dalam teks jadi bentuk yang sama, biasanya huruf kecil (*lowercase*), agar kata yang sama dengan penulisan berbeda (Lubis *et al.*, 2023). Sebagai contoh, kata "Bandung", "BANDUNG", dan "bandung" akan diubah menjadi "bandung". Dengan begitu, sistem tidak akan menganggap ketiganya sebagai kata yang berbeda.

2.7.2. Cleaning

Cleaning atau pembersihan teks adalah proses menghapus karakter yang tidak diperlukan dalam analisis seperti tanda baca, angka, simbol, dan karakter spesial, termasuk tautan, emoji, hingga spasi berlebih, sehingga menghasilkan teks yang lebih bersih dan hanya berisi huruf alfabet yang lebih mudah dianalisis pada tahap berikutnya (Huda *et al.*, 2022).

2.7.3. Tokenization

Membagi teks menjadi bagian kecil yaitu token, umumnya berupa kata, di mana setiap token merepresentasikan satu kata yang dianalisis secara terpisah dan menjadi dasar perhitungan frekuensi kata pada proses TF-IDF (Al Rasyid, R., &

Ningsih, 2024). Sebagai contoh, kalimat "wisata alam di bandung" akan dipecah menjadi token ["wisata", "alam", "di", "bandung"]. Token-token tersebut kemudian menjadi dasar perhitungan frekuensi kata.

2.7.4. *Stopword Removal*

Stopword removal adalah proses menghapus kata-kata yang sering muncul namun tidak memiliki makna penting dalam analisis yang umum disebut *stopword*, seperti "di", "yang", "dan", "atau", dan "ke" sehingga sistem lebih fokus pada kata bernilai informasi tinggi dan rekomendasi yang dihasilkan menjadi lebih relevan (Setiawan & Adnyana, 2023).

2.7.5. *Stemming (Sastrawi)*

Stemming adalah proses mengubah kata menjadi bentuk dasarnya dengan menghilangkan imbuhan, baik awalan maupun akhiran, agar kata-kata berakar sama dianggap sebagai satu entitas dalam analisis (Albab, J. F. U., & Rohman, 2026). Misalnya "berwisata", "diwisatakan", dan "wisatawan" diubah menjadi bentuk dasar "wisata"

Dalam penelitian ini, proses stemming dilakukan dengan memanfaatkan library Sastrawi. Sastrawi adalah sebuah library Python yang dikembangkan untuk *stemming* teks Bahasa Indonesia berdasarkan algoritma Nazief-Adriani. *Library* ini cukup populer di kalangan peneliti karena mampu menangani imbuhan Bahasa Indonesia yang kompleks, mulai dari awalan, sisipan, akhiran, hingga gabungan ketiganya.

2.7.6. *Contoh Tahapan Preprocessing*

Untuk memudahkan pemahaman, berikut contoh penerapan keseluruhan tahapan *Text Preprocessing* pada salah satu deskripsi destinasi wisata:

Tabel 2. 4 Contoh Tahapan *Text Preprocessing*

No	Tahap	Hasil
1	Teks Awal	"Kawah Putih adalah wisata alam di Bandung yang sangat indah dan populer!"
2	<i>Case Folding</i>	"kawah putih adalah wisata alam di bandung yang sangat indah dan populer!"

No	Tahap	Hasil
3	<i>Cleaning</i>	"kawah putih adalah wisata alam di bandung yang sangat indah dan populer"
4	<i>Tokenization</i>	["kawah", "putih", "adalah", "wisata", "alam", "di", "bandung", "yang", "sangat", "indah", "dan", "populer"]
5	<i>Stopword Removal</i>	["kawah", "putih", "wisata", "alam", "bandung", "indah", "populer"]
6	<i>Stemming</i>	["kawah", "putih", "wisata", "alam", "bandung", "indah", "populer"]

Selain itu, contoh penerapan *stemming* menggunakan *library* Sastrawi pada beberapa kata berimbuhan dapat dilihat pada tabel berikut:

Tabel 2. 5 Contoh Stemming Menggunakan Sastrawi

No	Kata Asli	Hasil Stemming
1	berwisata	wisata
2	wisatawan	wisata
3	pemandangannya	pandang
4	dikunjungi	kunjung
5	menarik	tarik
6	kepopulerannya	populer

Tabel di atas menunjukkan bagaimana Sastrawi memetakan kata berimbuhan ke bentuk dasarnya. Hal ini sangat membantu dalam perhitungan TF-IDF agar kata-kata dengan akar yang sama dihitung sebagai satu entitas.

2.8 Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan salah satu teknik pembobotan kata yang umum digunakan dalam pengolahan data teks. Metode ini berfungsi untuk memberikan bobot pada setiap kata berdasarkan tingkat kemunculan dan tingkat kepentingannya dalam suatu dokumen (Huda *et al.*, 2022). Semakin sering sebuah kata muncul dalam suatu dokumen namun jarang muncul di dokumen lain, maka bobotnya akan semakin tinggi. Hal inilah yang membuat TF-IDF mampu membedakan kata yang bersifat umum dengan kata yang benar-benar mewakili isi dokumen.

TF-IDF terdiri atas dua komponen utama, yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). Komponen TF menghitung frekuensi

kemunculan suatu kata dalam sebuah dokumen, IDF menghitung berapa langka kata di seluruh dokumen yang ada pada *dataset* (Yulita *et al.*, 2023).

2.8.1 *Term Frequency (TF)*

Term Frequency (TF) adalah jumlah kemunculan suatu kata dalam sebuah dokumen. Semakin sering kata tersebut muncul, semakin besar pula nilai TF-nya. Secara matematis, TF dapat dihitung dengan rumus berikut:

$$TF(t, d) = \frac{f(t, d)}{N(d)}$$

Keterangan:

- $TF(t, d)$ = nilai *Term Frequency* kata t dalam dokumen d
- $f(t, d)$ = frekuensi kemunculan kata t pada dokumen d
- $N(d)$ = total keseluruhan kata yang terdapat pada dokumen d

2.8.2 *Inverse Document Frequency (IDF)*

Inverse Document Frequency (IDF) digunakan untuk mengukur seberapa signifikansi suatu kata dalam keseluruhan dokumen. Kata yang hadir di banyak dokumen akan memperoleh nilai IDF yang rendah, sedangkan kata yang hanya muncul di sedikit dokumen akan memiliki nilai IDF yang besar. Rumus IDF adalah sebagai berikut:

$$IDF(t) = \log\left(\frac{N}{df(t)}\right)$$

Keterangan:

- $IDF(t)$ = nilai *Inverse Document Frequency* kata t
- N = total keseluruhan dokumen yang terdapat dalam *dataset*
- $df(t)$ = banyaknya dokumen yang memuat kata

2.8.3 **Pembobotan TF-IDF**

Nilai bobot akhir dari setiap kata diperoleh dari perkalian antara nilai TF dan IDF., dengan rumus sebagai berikut:

$$W(t, d) = TF(t, d) \times IDF(t)$$

Keterangan:

- $W(t,d)$ = bobot kata t pada dokumen d
- $TF(t,d)$ = nilai *Term Frequency*
- $IDF(t)$ = nilai *Inverse Document Frequency*

2.9 *Cosine Similarity*

Cosine Similarity adalah metode pengukuran kemiripan antara dua vektor dalam ruang multidimensi yang bekerja dengan menghitung sudut antara kedua vektor semakin kecil sudut yang terbentuk, semakin tinggi tingkat kemiripannya dengan nilai berada pada rentang 0 hingga 1, di mana nilai 1 menunjukkan kedua vektor identik dan nilai 0 menunjukkan keduanya tidak mirip sama sekali (Al Rasyid, R., & Ningsih, 2024).

Dalam sistem rekomendasi berbasis teks, *Cosine Similarity* sering dipakai dalam mengukur kemiripan antardokumen yang sebelumnya telah diubah menjadi bentuk vektor TF-IDF, dan kombinasi kedua metode ini cukup populer karena mampu menghasilkan rekomendasi yang akurat dan relevan, khususnya pada data teks berjumlah besar (Wicaksono *et al.*, 2024).

Cosine Similarity Selain pada sistem rekomendasi, *Cosine Similarity* yang dikombinasikan dengan TF-IDF juga terbukti efektif untuk menghitung kemiripan antarteks, misalnya pada pencocokan lirik lagu untuk menghasilkan rekomendasi musik yang relevan berdasarkan kemiripan kata (Timur, 2025).

2.9.1 Rumus *Cosine Similarity*

Secara matematis, *Cosine Similarity* dihitung dengan rumus berikut:

$$\cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \times \|\vec{B}\|} = \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

Keterangan:

- $\cos(\theta)$ = nilai kemiripan antara vektor A dan vektor B
- $A \cdot B$ = hasil perhitungan dot product dari kedua vektor

- $\|A\|$ = panjang vektor A
- $\|B\|$ = besar panjang vektor B
- A_i, B_i = elemen pada posisi ke-i dari vektor A dan B
- n = total dimensi vektor

2.9.2 Penerapan pada Sistem Rekomendasi

Dalam sistem rekomendasi destinasi wisata, *Cosine Similarity* digunakan untuk menghitung tingkat kemiripan antara satu destinasi dengan destinasi lainnya berdasarkan vektor TF-IDF dari deskripsi tiap destinasi, lalu destinasi dengan nilai *Cosine Similarity* tertinggi ditampilkan sebagai rekomendasi teratas kepada pengguna, sehingga sistem dapat memberikan saran yang sesuai tanpa memerlukan data rating atau ulasan dari pengguna lain (Christyawan *et al.*, 2024).

2.10 Website

Website adalah sekumpulan halaman yang memuat informasi berupa teks, gambar, audio, maupun video yang dapat dijangkau melalui jaringan internet menggunakan *web browser*, dan umumnya terdiri atas dua sisi pengembangan yang saling melengkapi, yaitu sisi *Frontend* yang berkaitan dengan tampilan yang dilihat pengguna serta sisi *backend* yang menangani logika pemrosesan data di balik layar (Prasetyo *et al.*, 2021).

Dalam membangun *website*, pada sisi *front-end* banyak dipilih *library* dan *framework* JavaScript seperti React dan Vue karena mampu membangun antarmuka pengguna yang dinamis dan responsif (Farro Axza Febsinatra Sofi'ie, 2023). Pendekatan berbasis komponen yang dimiliki React turut memudahkan pengembang mengelola tampilan yang kompleks dan dapat digunakan kembali pada bagian lain aplikasi (Naresvari, E., & Susetyo, 2025).

Sementara itu, pada sisi *back-end*, *framework* berbasis Python seperti Flask banyak digunakan karena strukturnya yang sederhana dan fleksibel, terutama untuk pengembangan sistem yang menggunakan REST API sebagai jembatan antara *front-end* dan *back-end* (Walingkas, H. L., & Saian, 2023). Flask juga menawarkan performa yang kompetitif dibandingkan *framework back-end* lain seperti Laravel

atau Express JS, khususnya untuk aplikasi berbasis Python yang melibatkan pengolahan data dan *Machine Learning* (Purwanto, 2023).

Pada penelitian ini, sistem rekomendasi destinasi wisata dibangun berbasis *website* agar dapat diakses lintas perangkat tanpa perlu instalasi tambahan. Pengembangan sisi *Frontend* menggunakan *framework* React, sisi *backend* menggunakan *framework* Flask berbasis bahasa Python, dengan dukungan teknologi tambahan seperti HTML, CSS, dan JavaScript.

2.10.1 React

React merupakan *library* JavaScript *open-source* yang telah dibangun oleh Facebook (sekarang Meta) untuk mengembangkan antarmuka pengguna, dan menggunakan konsep berbasis komponen sehingga setiap bagian halaman *website* dapat dipecah menjadi komponen kecil yang dapat digunakan kembali serta membuat pengembangan lebih terstruktur dan mudah dipelihara (Harahap *et al.*, 2025).

Salah satu kelebihan utama React adalah penggunaan *Virtual DOM* yang memungkinkan perubahan antarmuka dilakukan secara lebih cepat dan efisien tidak perlu *re-render* ulang seluruh halaman (Harahap *et al.*, 2025). Selain itu, React mendukung pengembangan *Single Page Application* (SPA), yaitu *website* yang berjalan pada satu halaman utama dengan navigasi antarkonten tanpa proses *reload*, sehingga menghadirkan pengalaman pengguna yang lebih cepat dan responsif (Jonathan, R. & Supriyadi., 2023).

Dalam penelitian ini, React dimanfaatkan untuk mengembangkan antarmuka *website* yang menampilkan daftar destinasi wisata, fitur pencarian berdasarkan kategori atau wilayah, halaman detail destinasi, serta hasil rekomendasi dari sistem.

2.10.2 Flask

Flask adalah sebuah *micro web framework* berbasis bahasa pemrograman Python yang dirancang ringan, sederhana, namun tetap fleksibel untuk membangun aplikasi *web* dari yang sederhana hingga kompleks, dan sering digunakan untuk

membangun *backend* yang menyediakan layanan API (*Application Programming Interface*) untuk berkomunikasi dengan sisi *Frontend* (Walingkas, H. L., & Saian, 2023).

Pada penelitian ini, Flask dipilih karena sifatnya yang ringan dan fleksibel sehingga memberi kebebasan menambahkan *library* sesuai kebutuhan, khususnya untuk integrasi dengan *library Machine Learning* dan *natural language processing*. Flask berfungsi sebagai penghubung antara model *Machine Learning* dengan antarmuka *website*, yaitu menyediakan *endpoint* API yang dapat dipanggil oleh sisi *Frontend* React untuk mengambil data destinasi, memproses permintaan rekomendasi, dan mengembalikan hasil perhitungan TF-IDF dan *Cosine Similarity* dalam bentuk respons JSON.

2.10.3 Python

Python termasuk bahasa pemrograman dengan sintaks sederhana dan mudah dipahami, serta banyak digunakan dalam berbagai bidang mulai dari pengembangan aplikasi *web*, analisis data, hingga implementasi *Machine Learning*, didukung oleh banyaknya *library* yang tersedia sehingga proses pengembangan menjadi lebih cepat (Surbakti *et al.*, 2024).

Dalam penelitian ini, Python digunakan sebagai bahasa utama untuk membangun *backend* sistem rekomendasi. Beberapa *library* Python yang digunakan antara lain:

- a. Flask
Sebagai *framework* untuk membangun *backend* dan menyediakan API.
- b. Scikit-learn
Library Machine Learning yang menyediakan fungsi TF-IDF *Vectorizer* dan *Cosine Similarity*.
- c. Pandas
Library untuk pengolahan data dalam bentuk *DataFrame*.
- d. NumPy
Library untuk perhitungan numerik.

e. Sastrawi

Library untuk *stemming* teks Bahasa Indonesia.

Kombinasi *library* tersebut memungkinkan sistem rekomendasi dibangun dengan efisien tanpa harus mengimplementasikan algoritma dari awal.

2.10.4 HTML, CSS, dan JavaScript

HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*), dan JavaScript merupakan tiga teknologi dasar dalam pengembangan *website*, di mana HTML berperan sebagai struktur halaman yang menyusun konten, CSS mengatur tampilan dan gaya visual halaman, sedangkan JavaScript memberikan interaktivitas pada halaman *website* (Rohmah *et al.*, 2024).

Dalam penelitian ini, ketiga teknologi tersebut digunakan melalui *framework* React yang menuliskan antarmuka dalam format JSX (*JavaScript XML*), yaitu kombinasi JavaScript dan HTML, untuk menghasilkan tampilan antarmuka pengguna yang dinamis.

2.10.5 JSON dan REST API

JSON (*JavaScript Object Notation*) adalah format data berbasis teks yang ringan dan mudah dibaca, baik oleh manusia maupun mesin, serta umum digunakan untuk pertukaran data antara *Frontend* dan *backend* pada aplikasi web, sedangkan REST API (*Representational State Transfer*) merupakan gaya arsitektur web service yang menggunakan protokol HTTP dengan format data yang pada umumnya berbentuk JSON (Muharom *et al.*, 2025).

Dalam penelitian ini, *backend* Flask menyediakan REST API yang dipanggil oleh *Frontend* React. Setiap kali pengguna meminta rekomendasi destinasi, React mengirimkan permintaan ke API Flask, kemudian Flask memproses permintaan tersebut menggunakan model *Machine Learning* dan mengembalikan hasil rekomendasi dalam format JSON untuk ditampilkan kepada pengguna.

2.11 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak merupakan pendekatan sistematis untuk merancang, membangun, dan menguji suatu sistem perangkat lunak, dan pemilihan metode yang tepat berpengaruh terhadap kualitas hasil akhir, efisiensi waktu, serta kemudahan pemeliharaan sistem, salah satunya metode *Waterfall* yang banyak digunakan dalam pengembangan sistem informasi (Anis *et al.*, 2024).

Metode *Waterfall* banyak dipakai pada pembuatan sistem informasi karena alurnya yang tersusun dan sistematis (Anis *et al.*, 2024). Beberapa penelitian terbaru juga menunjukkan bahwa metode *Waterfall* bagus diterapkan di pembangunan sistem informasi berbasis *website*, dimulai dari tahap analisis kebutuhan, perancangan sistem, implementasi, hingga tahap pengujian (Al Azfar, N. A., & Anggita, 2024). Hal inilah yang menjadi alasan metode *Waterfall* dipilih sebagai pendekatan pembangunan sistem dalam penelitian ini. Beberapa metode yang umum digunakan antara lain *Waterfall*, *Prototype*, *Agile*, dan *Rapid Application Development* (RAD) (Ramadhan, 2023).

Tabel 2. 6 Perbandingan Metode Pengembangan Perangkat Lunak

No	Metode	Karakteristik / Alur	Kelebihan	Kekurangan	Cocok untuk
1	<i>Waterfall</i>	Berurutan (<i>linear</i>) yaitu tiap tahap diselesaikan sebelum lanjut	Terstruktur, mudah dipahami, dokumentasi jelas, progres mudah dipantau	Kaku terhadap perubahan karena perubahan kebutuhan menuntut pengulangan dari tahap awal	Proyek dengan kebutuhan jelas dan stabil sejak awal
2	<i>Prototype</i>	Membuat purwarupa awal yang disempurnakan dari umpan balik pengguna	Kebutuhan pengguna cepat terlihat, mengurangi miskomunikasi	Iterasi berlebih boros waktu atau biaya serta dokumentasi kurang rapi	Kebutuhan belum jelas atau perlu validasi pengguna
3	<i>Agile</i>	Iteratif dan inkremental dalam siklus pendek (<i>sprint</i>)	Fleksibel terhadap perubahan, kolaborasi tinggi, rilis bertahap cepat	Dokumentasi minim, butuh tim berpengalaman dan komunikasi intensif	Proyek dinamis dengan kebutuhan yang sering berubah

No	Metode	Karakteristik / Alur	Kelebihan	Kekurangan	Cocok untuk
4	RAD	Pengembangan cepat memanfaatkan komponen siap pakai dan <i>prototyping</i>	Waktu pengembangan singkat, keterlibatan pengguna tinggi	Butuh tim ahli dan sumber daya memadai serta kurang sesuai untuk sistem kompleks	Proyek skala kecil menengah dengan tenggat singkat

2.11.1 Pengertian Metode *Waterfall*

Metode *Waterfall* atau *Linear Sequential Model* merupakan salah satu metode klasik dalam pengembangan perangkat lunak yang dilakukan secara berurutan, di mana setiap tahap harus diselesaikan terlebih dahulu sebelum berpindah ke tahap selanjutnya dan memberikan struktur yang jelas serta mudah dikelola untuk proyek dengan ruang lingkup yang sudah didefinisikan sejak awal (Mustakim *et al.*, 2024).

2.11.2 Tahapan Metode *Waterfall*

Metode *Waterfall* ada beberapa tahapan pokok yang dilaksanakan secara berurutan, yakni analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan (Al Azfar, N. A., & Anggita, 2024):

a. Analisis Kebutuhan (*Requirement Analysis*)

Bertujuan mengumpulkan dan menganalisis seluruh kebutuhan sistem yang hendak dibangun, baik kebutuhan fungsional maupun non-fungsional, yang menjadi dasar pada tahap perancangan.

b. Desain Sistem (*System Design*)

Mulai dari perencanaan arsitektur sistem, antarmuka pengguna, serta alur kerja sistem yang umumnya digambarkan menggunakan diagram perancangan seperti UML dan *flowchart*. Hasil dari tahap ini menjadi acuan dalam tahap implementasi.

c. Implementasi (*Implementation*)

Pada tahap ini, hasil rancangan diimplementasikan ke dalam kode program menggunakan bahasa pemrograman dan teknologi yang sesuai dengan kebutuhan sistem.

d. Pengujian (*Testing*)

Tahap pengujian dilakukan untuk memastikan sistem berjalan sesuai dengan kebutuhan yang telah ditentukan serta mengidentifikasi kesalahan sebelum sistem digunakan oleh pengguna.

e. Pemeliharaan (*Maintenance*)

Dilakukan setelah sistem selesai dibangun dan dipakai pengguna. Tahap ini mencakup perbaikan *bug*, penambahan fitur baru, atau penyesuaian sistem terhadap kebutuhan yang berkembang.

2.11.3 Kelebihan dan Kekurangan Metode *Waterfall*

Metode *Waterfall* memiliki beberapa kelebihan, di antaranya struktur tahapan yang jelas, mudah dipahami, dan memudahkan dokumentasi proyek, serta setiap tahap memiliki *deliverable* spesifik sehingga progres pengerjaan dapat dipantau dengan mudah (Anis *et al.*, 2024).

Di sisi lain, metode ini memiliki kekurangan terutama dalam hal fleksibilitas, karena apabila terdapat perubahan kebutuhan setelah suatu tahap selesai maka pengerjaan harus dimulai kembali dari tahap awal yang relevan (Mustakim *et al.*, 2024). Oleh karena itu, metode *Waterfall* lebih sesuai digunakan pada proyek dengan kebutuhan yang telah terdefinisi dengan jelas sejak awal, seperti pada penelitian ini.

2.12 UML (*Unified Modeling Language*)

Adalah bahasa pemodelan grafis untuk merancang dan mendokumentasikan sistem perangkat lunak melalui berbagai macam diagram untuk menggambarkan struktur, perilaku, serta interaksi sistem dari berbagai sudut pandang, sehingga membantu pengembang memahami sistem secara menyeluruh sebelum tahap implementasi (Narulita *et al.*, 2024).

UML banyak digunakan dalam perancangan sistem informasi karena mampu menggambarkan struktur dan interaksi sistem secara visual sebelum tahap implementasi (Narulita *et al.*, 2024). Beberapa diagram UML yang biasa digunakan mencakup *Use Case Diagram* untuk mengilustrasikan interaksi antara pengguna

dan sistem, *Activity Diagram* untuk menggambarkan alur aktivitas dalam sistem, *Sequence Diagram* untuk menggambarkan urutan proses yang terjadi, serta *Class Diagram* untuk menggambarkan struktur kelas dan relasi antar objek (Wayahdi, M. R., & Ruziq, 2023). Pemodelan dengan UML membantu pengembang dalam memahami kebutuhan sistem secara menyeluruh, sehingga rancangan yang dihasilkan lebih terstruktur dan mudah dikomunikasikan kepada pihak lain (Elis, E., & Voutama, 2022).

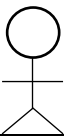
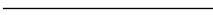
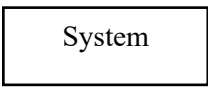
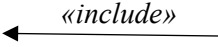
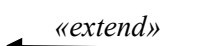
Pada penelitian ini, UML digunakan untuk merancang sistem rekomendasi destinasi wisata. Empat jenis diagram UML yang digunakan yaitu:

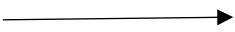
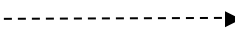
2.12.1 Use Case Diagram

Use Case Diagram adalah yang menyajikan interaksi *actor* dengan *website* dan memperlihatkan yang dapat dilakukan pengguna, dengan komponen utama berupa *actor*, *use case*, serta hubungan di antara keduanya (Wayahdi, M. R., & Ruziq, 2023). Berikut adalah simbol-simbol yang umum digunakan pada *Use Case Diagram*:

Tabel 2. 7 Simbol Use Case Diagram

Sumber (Wayahdi, M. R., & Ruziq, 2023)

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Yang berinteraksi dengan sistem
2		<i>Use Case</i>	aktivitas yang dilakukan dalam <i>website</i>
3		<i>Association</i>	yang menghubungkan <i>actor</i> dengan <i>use case</i>
4		<i>System Boundary</i>	Batas ruang lingkup sistem yang dirancang
5		<i>Include</i>	Hubungan di mana satu <i>use case</i> wajib menjalankan <i>use case</i> lain
6		<i>Extend</i>	Hubungan di mana satu <i>use case</i> memperluas <i>use case</i> lain pada kondisi tertentu

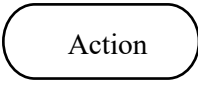
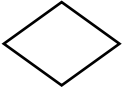
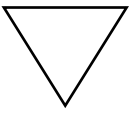
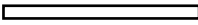
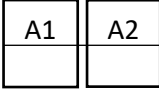
No	Simbol	Nama	Keterangan
7		Hubungan	Representasi hubungan antara aktor dan <i>use case</i> .
8		<i>Generalization</i>	Menunjukkan hubungan turunan aktor.

2.12.2 Activity Diagram

Activity Diagram adalah diagram alur kerja suatu sistem secara berurutan, mulai dari awal sampai akhir, termasuk percabangan yang terjadi, sehingga menggambarkan alur kerja sistem dari sudut pandang aktivitas (Elis, E., & Voutama, 2022).

Tabel 2. 8 Simbol Activity Diagram

Sumber (Elis, E., & Voutama, 2022)

No	Simbol	Nama	Keterangan
1		<i>Initial Node</i>	Titik permulaan dari suatu alur aktivitas
2		<i>Final Node</i>	Titik pengakhiran dari suatu alur aktivitas
3		<i>Action</i>	Aktivitas yang dilakukan dalam sistem
4		<i>Decision</i>	Percabangan kondisi (ya/tidak)
5		<i>Merge</i>	Penggabungan dua atau lebih jalur aktivitas menjadi satu
6		<i>Fork / Join</i>	Pemisahan atau penggabungan aliran aktivitas paralel
7		<i>Control Flow</i>	Arah aliran aktivitas dari satu <i>action</i> ke <i>action</i> lainnya
8		<i>Swimlane</i>	Pembagian aktivitas berdasarkan <i>actor</i> yang bertanggung jawab

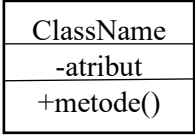
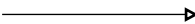
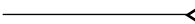
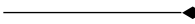
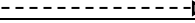
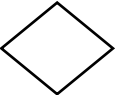
2.12.3 Class Diagram

Diagram yang membentuk struktur statis sistem, terutama kelas-kelas beserta atribut, metode, dan relasinya, yang menjadi dasar pembuatan struktur kode pada pemrograman berorientasi objek (Narulita *et al.*, 2024).

Sebuah kelas pada *Class Diagram* biasanya digambarkan sebagai persegi panjang yang dibagi menjadi tiga bagian, yaitu nama kelas di bagian paling atas, atribut di bagian tengah, dan metode di bagian bawah. Selain itu, hubungan antar kelas pada *Class Diagram* memiliki simbol yang berbeda-beda sesuai jenis relasinya.

Tabel 2. 9 Simbol *Class Diagram*

Sumber (Narulita, S., Nugroho, A., & Abdillah, 2024)

No	Simbol	Nama	Keterangan
1		<i>Class</i>	Representasi kelas yang terdiri atas nama, atribut, dan metode
2		<i>Association</i>	Hubungan antar kelas yang menunjukkan kedua kelas saling berkaitan
3		<i>Generalization / Inheritance</i>	Hubungan pewarisan antar kelas (kelas induk dan kelas anak)
4		<i>Aggregation</i>	Hubungan kepemilikan, di mana objek dapat berdiri sendiri
5		<i>Composition</i>	Hubungan kepemilikan yang lebih kuat, di mana objek tidak dapat berdiri sendiri
6		<i>Dependency</i>	Hubungan ketergantungan satu kelas terhadap kelas lain
7		<i>Multiplicity</i>	Jumlah objek yang terlibat dalam hubungan antar kelas
8		<i>Nary Association</i>	Cara agar menghindari asosiasi lebih dari dua objek

Selain simbol di atas, *Class Diagram* juga menggunakan tanda untuk menunjukkan tingkat akses dari atribut atau metode, yaitu:

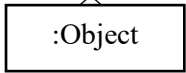
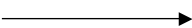
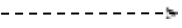
- Tanda + (*public*) yaitu dapat diakses dari kelas mana pun.
- Tanda – (*private*) yaitu hanya dapat diakses dari dalam kelas tersebut.
- Tanda # (*protected*) yaitu dapat diakses dari kelas itu dan kelas turunannya.

2.12.4 Sequence Diagram

Merupakan diagram yang membuat visual interaksi antarobjek berdasarkan urutan waktu, memperlihatkan objek saling mengirim pesan selama eksekusi suatu skenario, sehingga membantu memvisualisasikan alur komunikasi antarkomponen sistem secara temporal (Wayahdi, M. R., & Ruziq, 2023). Berikut adalah simbol-simbol yang umum digunakan pada *Sequence Diagram*:

Tabel 2. 10 Simbol Sequence Diagram

Sumber (Wayahdi, M. R., & Ruziq, 2023)

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Pelaku yang berkomunikasi ke sistem
2		<i>Object /</i>	Yang terlibat dalam interaksi
3		<i>Lifeline</i>	Garis waktu hidup dari sebuah objek
4		<i>Activation Bar</i>	Periode aktivitas suatu objek saat menjalankan operasi
5		<i>Synchronous Message</i>	Pesan yang menunggu balasan dari objek penerima
6		<i>Asynchronous Message</i>	Pesan yang tidak menunggu balasan dari objek penerima
7		<i>Return Message</i>	Pesan balasan dari objek penerima ke objek pengirim
8		<i>Destruction</i>	Penghancuran objek yang sudah tidak digunakan

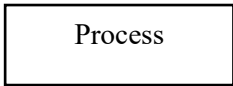
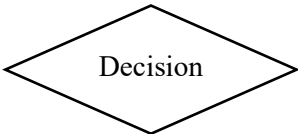
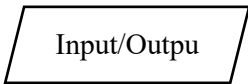
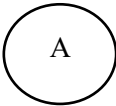
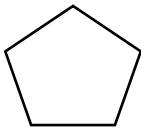
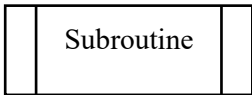
2.13 Flowchart

Adalah bentuk penyajian grafis yang menggambarkan langkah-langkah atau alur suatu proses secara berurutan untuk menyajikan logika sebuah algoritma atau prosedur, menggunakan simbol-simbol yang dihubungkan garis penghubung untuk menunjukkan arah aliran, sehingga membantu memahami cara kerja sistem sebelum diterjemahkan ke dalam kode program (Yewi, Y. K., & Manullang, 2024).

Pada penelitian ini, *flowchart* dipakai untuk alat bantu merancang alur kerja sistem rekomendasi destinasi wisata mulai dari pengguna memasukkan kata kunci pencarian hingga sistem menampilkan rekomendasi destinasi yang sesuai. Dengan adanya *flowchart*, alur sistem dapat dipahami, oleh pengembang maupun ataupun yang membaca dokumentasi penelitian. Berikut adalah simbol-simbol yang umum digunakan dalam *flowchart*:

Tabel 2. 11 Simbol *Flowchart*

Sumber (Yewi, Y. K., & Manullang, 2024)

No	Simbol	Nama	Keterangan
1		<i>Terminator</i>	Menyatakan awal atau akhir dari sebuah proses
2		<i>Process</i>	Tahapan proses dalam alur kerja.
3		<i>Decision</i>	Menyatakan kondisi atau percabangan yang menghasilkan dua kemungkinan (ya atau tidak)
4		<i>Input/Output</i>	memasukkan data atau menampilkan data
5		<i>Connector</i>	Menyatakan penghubung antar bagian <i>flowchart</i> pada halaman yang sama
6		<i>Off-page Connector</i>	Menyatakan penghubung antar bagian <i>flowchart</i> pada halaman yang berbeda
7		<i>Predefined Process</i>	Menyatakan proses yang sudah didefinisikan sebelumnya, biasanya berupa subrutin atau modul terpisah
8		<i>Flow Line</i>	Menyatakan arah aliran proses dari satu simbol ke simbol lainnya
9		<i>Database Storage</i>	Menyatakan media penyimpanan data

No	Simbol	Nama	Keterangan
10		<i>Document</i>	Menyatakan dokumen sebagai <i>input</i> atau <i>output</i> dari proses

2.14 Pengujian Sistem

Pengujian sistem merupakan fase krusial dalam pembangunan perangkat lunak yang bertujuan memastikan sistem berfungsi sesuai kebutuhan yang telah ditentukan, dengan mengidentifikasi *bug*, kesalahan logika, atau ketidaksesuaian fitur sebelum sistem digunakan oleh pengguna akhir, sekaligus mengukur kualitas hasil yang dihasilkan sistem, terutama pada sistem rekomendasi yang memerlukan evaluasi terhadap relevansi rekomendasi (Faurina & Sitanggang, 2023).

Black Box Testing banyak digunakan untuk menguji aplikasi berbasis *website* karena berfokus pada fungsionalitas eksternal sistem tanpa harus mengetahui struktur kode internalnya, sehingga efektif memastikan setiap fitur berjalan sesuai spesifikasi yang ditetapkan (Zen *et al.*, 2024). Penguji memberikan masukan tertentu, kemudian membandingkan keluaran yang dihasilkan dengan keluaran yang diharapkan, sehingga penguji dapat mengevaluasi sistem dari sudut pandang pengguna akhir dan lebih mudah mengidentifikasi kesalahan fungsionalitas pada antarmuka maupun alur kerja sistem (Jibril *et al.*, 2024).

Pada penelitian ini, pengujian dilakukan menggunakan metode *Black-Box Testing* dan metrik *Precision@K* serta *Mean Average Precision* (MAP). Pendekatan tersebut digunakan untuk mengevaluasi sistem dari dua sudut pandang yang berbeda, yaitu kelayakan fitur dan ketepatan hasil rekomendasi.

2.14.1 *Black-Box Testing*

Black-Box Testing merupakan metode pengujian perangkat lunak yang dilaksanakan dengan mengevaluasi fungsi-fungsi sistem tanpa memperhatikan struktur kode di dalamnya, dengan menitikberatkan pada masukan yang diberikan dan keluaran yang dihasilkan, sehingga cocok memverifikasi sistem berjalan sesuai spesifikasi fungsional yang ditetapkan (Albab, J. F. U., & Rohman, 2026).

Pada penelitian ini, *Black-Box Testing* digunakan untuk menguji setiap fitur sistem rekomendasi destinasi wisata, mulai dari halaman beranda, pencarian destinasi, penampilan daftar destinasi, halaman detail, hingga fitur rekomendasi destinasi serupa. Pengujian dilakukan dengan mencocokkan hasil yang muncul pada sistem dengan hasil yang diharapkan, kemudian dikategorikan sebagai berhasil atau gagal. Berikut contoh format tabel pengujian *Black-Box Testing*:

Tabel 2. 12 Contoh Format Pengujian *Black-Box Testing*

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
1	Mengakses halaman beranda	Halaman beranda berhasil ditampilkan	Halaman beranda berhasil ditampilkan	Berhasil
2	Menginputkan kata kunci pada kolom pencarian	Sistem menampilkan destinasi yang relevan	Sistem menampilkan destinasi yang relevan	Berhasil
3	Memilih filter kategori wisata	Daftar destinasi terfilter sesuai kategori	Daftar destinasi terfilter sesuai kategori	Berhasil

Dari hasil pengujian tersebut, sistem dikatakan layak digunakan apabila seluruh skenario pengujian menghasilkan kesimpulan "Berhasil".

2.14.2 *Precision@K*

Precision@K adalah metrik evaluasi untuk mengukur tingkat ketepatan sistem rekomendasi pada *Top-K* hasil yang ditampilkan, yaitu dengan menghitung proporsi rekomendasi relevan dari total rekomendasi yang diberikan sistem, di mana semakin tinggi nilainya semakin baik kualitas rekomendasi (Yulita *et al.*, 2023). Secara matematis, *Precision@K* dihitung dengan rumus berikut:

$$\text{Precision@K} = \frac{\text{Jumlah rekomendasi relevan pada top-K}}{K}$$

Keterangan

- *Precision@K* = nilai *precision* pada K rekomendasi paling atas
- *K* = banyaknya rekomendasi yang ditampilkan (contohnya 5 atau 10)
- Rekomendasi relevan = rekomendasi yang sesuai dengan preferensi atau ekspektasi pengguna

Contoh perhitungan: jika sistem menampilkan 5 rekomendasi destinasi ($K = 5$) dan 4 di antaranya relevan dengan preferensi pengguna, maka nilai $Precision@5$ adalah:

$$Precision@5 = \frac{4}{5} = 0,80 = 80$$

Nilai tersebut menunjukkan bahwa 80% dari rekomendasi yang ditampilkan sistem termasuk kategori relevan.

2.14.3 Mean Average Precision (MAP)

Merupakan metrik evaluasi menghitung nilai rata-rata dari *Average Precision* (AP) dari beberapa skenario pencarian atau *query*, dengan demikian memberikan gambaran performa sistem pada berbagai kondisi pencarian dan lebih representatif dibandingkan jika hanya menggunakan satu *query*, dengan nilai pada rentang 0 sampai 1 di mana nilai yang mendekati 1 menunjukkan performa sistem baik (Sibuea & Widodo, 2024).

Sebelum menentukan nilai MAP, terlebih dahulu dihitung nilai *Average Precision* (AP) bagi *query* menggunakan rumus berikut:

$$AP = \frac{1}{R} \sum_{k=1}^n Precision@k \times rel(k)$$

Keterangan:

- AP = nilai *Average Precision* untuk satu *query*
- R = jumlah dokumen relevan pada hasil
- n = jumlah total dokumen yang ditampilkan
- $Precision@K$ = nilai *precision* pada posisi ke- k
- $rel(k)$ = bernilai 1 apabila dokumen pada posisi ke- k dinyatakan relevan, dan 0 apabila tidak relevan

Setelah memperoleh nilai AP dari setiap *query*, MAP dihitung dengan merata-ratakan seluruh nilai AP menggunakan rumus berikut:

$$MAP = \frac{1}{Q} \sum_{q=1}^Q AP_q$$

Keterangan:

- MAP = nilai *Mean Average Precision*
- Q = keseluruhan jumlah query yang diujikan
- AP_q = nilai AP pada *query* ke- q

Contoh Perhitungan MAP yaitu misalkan sistem diuji dengan 3 *query* yang masing-masing menghasilkan nilai AP sebagai berikut:

- *Query* 1: nilai AP = 0,80
- *Query* 2: nilai AP = 0,75
- *Query* 3: nilai AP = 0,85

Maka nilai MAP-nya adalah:

$$MAP = \frac{0,80 + 0,75 + 0,85}{3} = \frac{2,40}{3} = 0,80$$

Nilai MAP sebesar 0,80 menunjukkan bahwa secara rata-rata sistem rekomendasi memberikan hasil yang relevan dengan tingkat akurasi 80%. Dengan demikian, sistem dapat dikategorikan memiliki performa yang baik dalam menghasilkan rekomendasi destinasi wisata.