

OPTIMASI RUTE PERJALANAN WISATA KOTA SURAKARTA DENGAN PENDEKATAN DIJKSTRA-HAVERSINE UNTUK REKOMENDASI ITINERARY SATU HARI BERBASIS KECERDASAN BUATAN

Angelia Putri Yuniar, Faulinda Ely Nastiti, Moh. Muhtarom

Teknik Informatika, Universitas Duta Bangsa Surakarta

Jl. Bhayangkara No.55, Tipes, Kec. Serengan, Kota Surakarta, Jawa Tengah 57154

putriiangelia27@gmail.com

ABSTRAK

Kota Surakarta sebagai salah satu destinasi wisata budaya memiliki berbagai pilihan objek wisata yang menarik. Namun, keterbatasan waktu kunjungan, khususnya wisata satu hari, menyebabkan wisatawan memerlukan perencanaan rute yang efisien agar dapat mengunjungi beberapa destinasi dalam waktu terbatas. Penelitian ini diorientasikan untuk mendesain dan membangun sebuah platform digital berbasis web yang dapat menyusun rekomendasi koridor perjalanan wisata secara optimal. Metode yang digunakan meliputi pengumpulan data destinasi wisata, perhitungan jarak antar lokasi menggunakan metode *Haversine*, pemodelan graf berbobot, serta penerapan algoritma Dijkstra untuk menentukan rute terpendek. Sistem dikembangkan menggunakan *PHP Native*, *MySQL*, dan *Leaflet Maps* melalui tahapan analisis, desain, implementasi, pengujian, dan *maintenance*. Hasil penelitian menunjukkan bahwa sistem berhasil menentukan rute perjalanan wisata dengan total jarak yang lebih efisien berdasarkan titik awal perjalanan dan kategori wisata yang dipilih pengguna. Tahap pengujian perangkat lunak yang menerapkan *Black Box Testing* memberikan bukti empiris bahwa seluruh modul fungsional di dalam sistem berjalan tanpa kendala (*error-free*) dan telah memenuhi parameter kebutuhan fungsional yang ditetapkan pada fase desain. Selain itu, sistem mampu menampilkan rekomendasi *Itinerary* perjalanan wisata beserta visualisasi rute pada peta digital secara interaktif sehingga membantu wisatawan dalam merencanakan perjalanan wisata satu hari secara lebih optimal.

Kata kunci : Sistem Rekomendasi, Algoritma Dijkstra, Rute Terpendek, Pariwisata, Kota Surakarta, Haversine.

1. PENDAHULUAN

Kota Surakarta merupakan salah satu kota budaya di Indonesia yang memiliki beragam potensi pariwisata, seperti wisata sejarah, budaya, dan kuliner. Letaknya yang strategis serta kedekatannya dengan kota-kota tujuan wisata lain seperti Yogyakarta, Magelang, Salatiga, Semarang, dan Gunungkidul menjadikan kota ini sebagai salah satu destinasi yang cukup diminati wisatawan. Tren kunjungan wisata ke kota ini menunjukkan trajektori yang konsisten positif, sebagaimana tercermin dari peningkatan volume kunjungan dari kisaran 5,4 juta pada tahun 2024 menuju 5,6 juta pada tahun 2025 [1]. Namun, rata-rata lama tinggal wisatawan yang relatif singkat, yaitu sekitar 1,42 hari [2], menyebabkan banyak wisatawan melakukan kunjungan dalam waktu terbatas.

Berbagai metode telah digunakan dalam pengelolaan rute perjalanan wisata, di antaranya penelitian yang dilakukan oleh [3] menggunakan algoritma *Support Vector Machine* (SVM) dan *Decision Tree* dalam sistem rekomendasi wisata, namun penelitian tersebut lebih berfokus pada klasifikasi rekomendasi destinasi dan belum membahas optimasi rute perjalanan wisata secara spesifik. Penelitian [4] menggunakan metode MABAC dan ROC pada sistem pendukung keputusan wisata, tetapi belum mampu menentukan jalur perjalanan terpendek antar destinasi wisata. Selain itu, penelitian yang dilakukan oleh [5] mengembangkan sistem rekomendasi perjalanan wisata berbasis *expert system* dan algoritma genetika, namun proses

perhitungannya relatif kompleks ketika jumlah destinasi bertambah. Berdasarkan beberapa penelitian tersebut, diperlukan pengembangan sistem rekomendasi wisata yang mampu menentukan *Itinerary* perjalanan secara efisien berdasarkan jarak antar destinasi wisata.

Algoritma Dijkstra merupakan salah satu solusi komputasional yang terbukti andal dalam permasalahan penetapan jalur terpendek pada struktur graf berbobot. Algoritma ini bekerja dengan meminimalkan akumulasi jarak tempuh antar destinasi sehingga efisiensi perjalanan dapat ditingkatkan secara substansial [6][7]. Pengelolaan rute perjalanan wisata dalam konteks kunjungan satu hari memerlukan metode yang mampu mengoptimalkan jarak antar destinasi. Algoritma Dijkstra bekerja dengan menentukan jalur terpendek dari satu titik ke titik lainnya berdasarkan bobot jarak [7][8]. Dalam penelitian ini, perhitungan jarak antar destinasi wisata dilakukan menggunakan metode *Haversine* berdasarkan koordinat geografis berupa *latitude* dan *longitude* [9]. Hasil perhitungan jarak tersebut digunakan sebagai bobot pada proses pencarian rute menggunakan algoritma Dijkstra.

Penggabungan algoritma Dijkstra dan metode *Haversine* dilakukan menggunakan pendekatan *sequential* dalam proses pencarian rute perjalanan wisata. Metode *Haversine* digunakan terlebih dahulu untuk menghitung jarak antar koordinat destinasi wisata, kemudian hasil perhitungan tersebut diproses menggunakan algoritma Dijkstra untuk menentukan

rute perjalanan terpendek. Kombinasi kedua metode tersebut memungkinkan sistem menghasilkan rekomendasi *Itinerary* wisata satu hari secara otomatis dan lebih efisien. Pendekatan ini termasuk salah satu penerapan kecerdasan buatan dalam membantu pengguna menentukan keputusan perjalanan wisata berdasarkan proses komputasi dan optimasi rute [5][10].

Tujuan utama penelitian ini adalah merancang dan mewujudkan sistem rekomendasi perjalanan wisata satu hari di Kota Surakarta yang berbasis algoritma Dijkstra. Dengan adanya sistem yang mampu memberikan rekomendasi rute perjalanan yang optimal, wisatawan dapat memanfaatkan waktu secara lebih efisien dan mengunjungi lebih banyak destinasi wisata dalam waktu yang terbatas. Pada tataran yang lebih luas, penelitian ini diharapkan berkontribusi terhadap peningkatan mutu layanan informasi pariwisata berbasis teknologi digital.

2. TINJAUAN PUSTAKA

Sebelum pelaksanaan penelitian ini, penulis melakukan kajian kepustakaan secara komprehensif dengan menelusuri berbagai referensi yang relevan, baik berupa buku maupun jurnal ilmiah nasional dan internasional. Kajian ini bertujuan membangun landasan konseptual yang kokoh mengenai sistem rekomendasi serta metodologi penentuan rute perjalanan wisata.

Penelitian mengenai sistem rekomendasi wisata telah banyak dilakukan dengan berbagai pendekatan metode. Salah satu penelitian mengimplementasikan algoritma Dijkstra untuk menentukan rute terpendek menuju lokasi wisata dan hotel di Kota Sukabumi berbasis Android. Hasil penelitian membuktikan kapabilitas algoritma tersebut dalam menghasilkan rekomendasi perjalanan yang efisien [11].

Penelitian selanjutnya menerapkan metode *Content-Based Filtering* dan *Collaborative Filtering* dalam sistem rekomendasi wisata di Bali dengan memanfaatkan data atribut destinasi dan histori penilaian pengguna. Sistem yang dihasilkan memiliki performa yang baik, namun sangat bergantung pada ketersediaan data preferensi pengguna [12].

Penelitian lain mengoperasionalkan *Content-Based Filtering* berbasis TF-IDF dan *Cosine Similarity* untuk merekomendasikan objek wisata berdasarkan ulasan pengguna dengan hasil yang relevan, tetapi kualitasnya dipengaruhi oleh kelengkapan data ulasan [13].

Berdasarkan penelitian tersebut, metode berbasis preferensi pengguna memiliki keterbatasan pada ketersediaan data dan subjektivitas pengguna. Sementara itu, pendekatan berbasis rute seperti algoritma Dijkstra lebih menitikberatkan pada efisiensi jarak tempuh. Oleh karena itu, penelitian ini menggunakan algoritma Dijkstra sebagai fondasi komputasi dengan keunggulan berupa kemampuannya tidak hanya menentukan rute terpendek, tetapi juga

memberikan rekomendasi destinasi yang dapat dikunjungi secara optimal dalam satu hari.

2.1. Sistem Rekomendasi

Sistem rekomendasi merupakan teknologi informasi yang dirancang untuk menyampaikan saran kepada pengguna berdasarkan kebutuhan spesifik, pola preferensi, atau karakteristik data tertentu dalam rangka memfasilitasi proses pengambilan keputusan yang lebih terarah. Mekanisme kerja sistem ini bertumpu pada analisis korelasi antara data pengguna dan data item untuk memprediksi produk yang paling sesuai. Analisis tersebut dapat dilakukan berdasarkan riwayat interaksi, atribut item, maupun pola tertentu yang ditemukan dalam data [14]. Rekomendasi yang dihasilkan bersifat personal sehingga dapat mendukung pengambilan keputusan pengguna secara lebih tepat dan terarah [15].

2.2. Pariwisata

Pariwisata merupakan aktivitas perjalanan yang dilakukan untuk mengunjungi suatu daerah atau destinasi tertentu dengan tujuan rekreasi, hiburan, maupun menikmati daya tarik wisata yang tersedia. Pengembangan sektor pariwisata didukung oleh keberagaman destinasi yang dimiliki suatu daerah. Pariwisata kini menjadi salah satu sektor industri yang paling besar dan berperan penting di era globalisasi. Perkembangannya berlangsung sangat pesat seiring dengan menurunnya dominasi industri manufaktur dan berbagai sektor lainnya yang sebelumnya menguasai sistem ekonomi dan sosial dunia [16].

2.3. Algoritma Dijkstra

Algoritma Dijkstra merupakan representasi dari paradigma *Greedy* yang mendapat pengakuan luas dalam penyelesaian permasalahan pencarian jalur terpendek pada graf berarah maupun tidak berarah dengan nilai bobot sisi yang non-negatif. Algoritma ini bekerja dengan cara mengunjungi titik-titik pada graf, dimulai dari titik asal, lalu secara bertahap menentukan jarak terpendek ke setiap titik lainnya hingga titik tujuan tercapai. Prinsip utama algoritma ini adalah memilih simpul dengan bobot terkecil di setiap langkahnya. Dalam konteks pariwisata, simpul (*node*) merepresentasikan lokasi wisata, sementara bobot (*weight*) merepresentasikan jarak atau waktu tempuh antar destinasi.

Secara matematis, algoritma Dijkstra dapat dinyatakan dalam fungsi jarak. Misalkan $G = (V, E)$ adalah graf dengan set titik V dan set sisi E . Jika $d(v)$ adalah jarak terpendek dari titik awal s ke titik v , dan $w(u, v)$ adalah bobot dari sisi yang menghubungkan titik u ke v , maka persamaannya adalah:

$$d(v) = \min(d(v), d(u) + w(u, v)) \quad (1)$$

di mana $d(v)$ adalah jarak saat ini dari titik asal ke titik v , $d(u)$ merupakan jarak terpendek yang sudah diketahui dari titik asal ke titik u , dan $w(u, v)$ adalah

jarak atau bobot antara titik u dan titik v . Dengan demikian, algoritma Dijkstra dapat dimanfaatkan secara efektif dalam menentukan urutan kunjungan destinasi wisata yang paling efisien untuk mendukung penyusunan *itinerary* kunjungan satu hari [11].

2.4. Graf

Graf merupakan abstraksi struktur data yang terdiri atas kumpulan simpul (*node*) dan sisi (*edge*) yang berfungsi merepresentasikan keterhubungan antar entitas. Dalam konteks rute perjalanan, graf digunakan untuk memodelkan hubungan antar lokasi [17]. Dalam ilmu komputer, peta kota dapat direpresentasikan sebagai graf, di mana destinasi wisata dianggap sebagai simpul (V) dan jalan yang menghubungkan antar destinasi dianggap sebagai sisi (E).

$$G = (V, E) \quad (2)$$

Setiap sisi dilekati bobot (*weight*) w yang merepresentasikan jarak fisik dalam satuan kilometer (km) atau waktu tempuh dalam menit. Algoritma Dijkstra akan memproses graf tersebut untuk menentukan jalur dengan total bobot terkecil [18].

2.5. Rute Terpendek (Shortest Path)

Rute terpendek didefinisikan sebagai lintasan dengan jumlah bobot kumulatif terkecil yang menghubungkan dua simpul dalam suatu graf. Bobot pada graf dapat berupa jarak, waktu tempuh, atau biaya perjalanan, tergantung pada konteks permasalahan yang dihadapi. Dalam permasalahan rute perjalanan wisata, bobot yang digunakan umumnya berupa jarak antar destinasi atau waktu tempuh, sehingga jalur yang dihasilkan dapat meminimalkan waktu dan energi yang dibutuhkan oleh wisatawan [7].

2.6. Metode Haversine

Metode *Haversine* merupakan metode perhitungan jarak antara dua titik di permukaan bumi berdasarkan koordinat geografis berupa *latitude* dan *longitude*. Metode ini digunakan untuk menghitung jarak garis lurus antar lokasi dengan mempertimbangkan bentuk bumi yang mendekati bola [19].

Dalam penelitian ini, metode *Haversine* digunakan untuk menghitung jarak antar destinasi wisata di Kota Surakarta. Hasil kalkulasinya diposisikan sebagai nilai bobot (*weight*) dalam graf yang selanjutnya diproses oleh algoritma Dijkstra. Penggunaan metode ini membantu sistem menghasilkan estimasi jarak yang lebih akurat berdasarkan posisi geografis setiap lokasi wisata [20].

$$d = 2r \operatorname{asin} \sqrt{t \sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \left(\cos(\varphi_1) \cos(\varphi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right) + \right)} \quad (3)$$

Dimana d = jarak antar dua lokasi, r = jari-jari bumi, φ_1 dan φ_2 = *latitude* lokasi pertama dan kedua, sedangkan λ_1 dan λ_2 = *longitude* lokasi pertama dan kedua.

3. METODE PENELITIAN

Metodologi penelitian pada penelitian ini mencakup jenis dan sumber data, metode pengumpulan data, serta tahapan penelitian yang digunakan dalam pengembangan sistem.

3.1. Jenis dan Sumber Data

a. Jenis Data

Data yang menjadi basis penelitian ini berupa informasi destinasi wisata yang meliputi nama tempat, kategori, serta jarak antar destinasi. Data tersebut digunakan sebagai dasar dalam proses penentuan rekomendasi dan perhitungan rute kunjungan wisata menggunakan algoritma Dijkstra.

b. Sumber Data

Data diperoleh dari website pariwisata resmi Kota Surakarta yang menyediakan profil komprehensif destinasi wisata meliputi nama tempat dan kategori. Selain itu, data jarak antar destinasi diperoleh dari *Google Maps* berdasarkan jarak tempuh antar lokasi yang digunakan sebagai dasar dalam perhitungan rute menggunakan algoritma Dijkstra.

3.2. Metode Pengumpulan Data

Pengumpulan data didasarkan pada dua pilar sumber yaitu data primer yang bersumber dari dokumen pariwisata Kota Surakarta, serta data sekunder berupa bentangan jarak yang ditarik dari *Google Maps*. Parameter yang digunakan untuk mendukung fungsionalitas sistem ini meliputi nama objek wisata, kategori pengelompokan, letak koordinat spasial, serta informasi jarak relatif antar-destinasi [2].

3.3. Tahapan Penelitian

Alur penelitian dirancang secara hierarkis dan terstruktur, mencakup lima fase utama dari analisis awal hingga *maintenance*.

a. Tahap Analisis

Pada tahap analisis dilakukan identifikasi dan evaluasi terhadap permasalahan dalam penentuan rute perjalanan wisata yang efisien. Proses analisis dilakukan dengan mengumpulkan data destinasi wisata, kategori wisata, koordinat lokasi, serta kebutuhan pengguna dalam menentukan perjalanan wisata satu hari. Hasil analisis menjadi dasar pengembangan sistem rekomendasi wisata berbasis web menggunakan algoritma Dijkstra untuk menentukan rute yang lebih efisien.

b. Desain

Tahap desain merupakan perancangan sistem secara menyeluruh berdasarkan hasil analisis yang telah dilakukan. Pada tahap ini ditentukan struktur sistem, perancangan basis data, antarmuka pengguna, serta alur kerja perangkat lunak. Selain itu, algoritma yang digunakan juga dirancang secara detail agar sistem dapat berjalan sesuai dengan kebutuhan pengguna.

c. Implementasi

Pada fase ini, rancangan sistem diwujudkan menjadi aplikasi web fungsional menggunakan *PHP Native*, *MySQL*, dan *Leaflet Maps*. Pada tahap ini metode *Haversine* diimplementasikan untuk mengkalkulasi jarak antar destinasi berdasarkan nilai koordinat, sementara algoritma Dijkstra diterapkan untuk menentukan lintasan perjalanan paling efisien. Seluruh komponen sistem kemudian diintegrasikan sehingga membentuk sistem rekomendasi *Itinerary* wisata satu hari berbasis web.

d. Testing

Verifikasi sistem dilaksanakan untuk memastikan setiap komponen fungsional beroperasi sesuai kebutuhan pengguna. Proses ini meliputi penggabungan modul (*integration*) dan pengujian sistem secara keseluruhan. Proses ini mencakup pengujian integrasi antarmodul dan evaluasi sistem secara holistik, dengan tujuan mengidentifikasi anomali atau kesalahan serta mengonfirmasi kesesuaian sistem dengan desain dan spesifikasi yang telah ditetapkan sesuai kebutuhan pengguna.

e. Maintenance

Tahap *maintenance* mencakup kegiatan pemeliharaan dan perbaikan sistem setelah digunakan. Proses ini dilakukan untuk menjaga kinerja sistem tetap optimal, memperbaiki kesalahan yang ditemukan, serta melakukan penyesuaian jika terdapat kebutuhan atau pengembangan lebih lanjut.

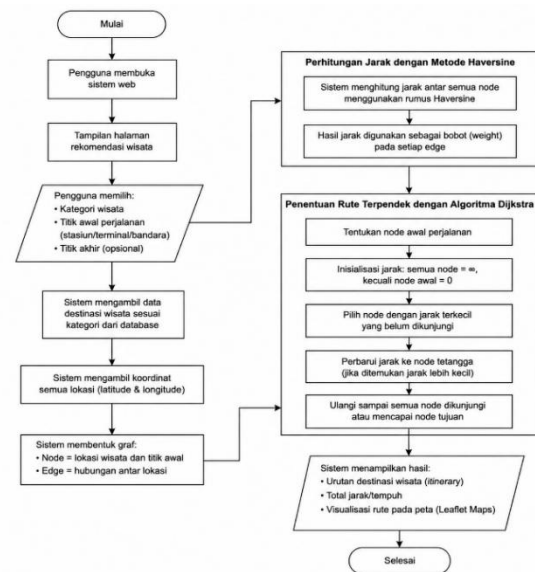
4. HASIL DAN PEMBAHASAN

Bagian ini memaparkan hasil implementasi sistem rekomendasi rute wisata Kota Surakarta berbasis web menggunakan algoritma Dijkstra.

4.1. Hasil Analisis

Tahap analisis dilaksanakan untuk memetakan permasalahan dan mendefinisikan kebutuhan sistem secara komprehensif dalam proses penentuan rute perjalanan wisata di Kota Surakarta. Berdasarkan hasil analisis, wisatawan yang melakukan perjalanan satu hari memerlukan sistem yang mampu membantu menentukan rute perjalanan secara lebih efisien agar dapat mengunjungi beberapa destinasi wisata dalam waktu terbatas. Selain itu, pengguna juga membutuhkan informasi lokasi wisata yang mudah diakses serta visualisasi rute perjalanan yang jelas.

Berdasarkan kebutuhan tersebut, sistem dirancang agar pengguna dapat memilih kategori wisata, menentukan titik awal perjalanan, melihat rekomendasi rute wisata, serta memperoleh *Itinerary* perjalanan secara otomatis. Data yang digunakan meliputi nama destinasi wisata, kategori wisata, deskripsi tempat, foto wisata, serta koordinat *latitude* dan *longitude*. Data koordinat tersebut menjadi basis kalkulasi jarak *Haversine* dan penentuan rute dengan algoritma Dijkstra.

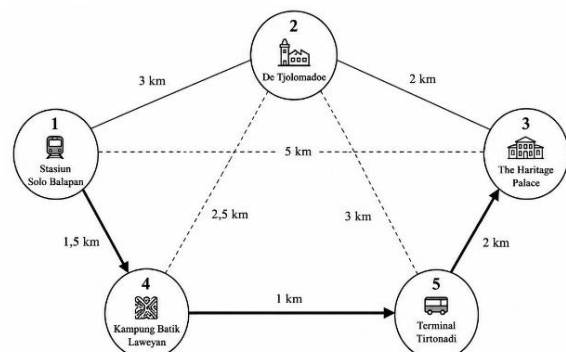


Gambar 1. Flowchart Sistem

Berdasarkan *flowchart* pada Gambar 1, proses sistem dimulai ketika pengguna memilih kategori wisata serta menentukan titik awal perjalanan. Sistem kemudian mengambil data koordinat lokasi wisata dari basis data dan menghitung jarak antar lokasi menggunakan metode *Haversine*. Selanjutnya, algoritma Dijkstra digunakan untuk menentukan rute perjalanan dengan jarak terpendek. Hasil proses ditampilkan dalam bentuk rekomendasi destinasi wisata, visualisasi rute pada peta, dan *Itinerary* perjalanan.

4.2. Perancangan Sistem

Tahap perancangan berfokus pada penyusunan logika pencarian rute dalam sistem rekomendasi. Pada tahap ini dibuat graf berbobot yang menggambarkan hubungan antar destinasi wisata dan titik transportasi. Setiap lokasi direpresentasikan sebagai simpul (*node*), sedangkan koneksi penghubung antar lokasi direpresentasikan sebagai sisi (*edge*) dengan bobot berupa jarak tempuh antar destinasi wisata.



Gambar 2. Diagram Graf Algoritma Dijkstra

Pada Gambar 2 ditunjukkan representasi graf yang digunakan dalam proses pencarian rute perjalanan wisata. *Node* merepresentasikan lokasi

wisata dan titik transportasi sementara itu, keterhubungan antar-titik spasial direpresentasikan oleh *edge*, dengan bobot yang merepresentasikan jarak geometris berbasis kalkulasi formula *Haversine*. Guna mengidentifikasi rute dengan akumulasi bobot paling minimum, penelitian ini mengimplementasikan algoritma Dijkstra. Berdasarkan hasil perhitungan, rute terpendek menuju The Heritage Palace diperoleh melalui jalur Stasiun Solo Balapan → Kampung Batik Laweyan → Terminal Tirtonadi → The Heritage Palace dengan total jarak 4,5 km.

4.3. Implementasi Sistem Rekomendasi

Proses realisasi sistem dilakukan melalui pengembangan platform rekomendasi pariwisata berbasis web dengan mengintegrasikan *PHP Native* dan *MySQL*. Arsitektur perangkat lunak ini memisahkan hak akses menjadi dua bagian utama, yaitu antarmuka administrator untuk manajemen data destinasi, serta antarmuka pengguna yang dirancang khusus untuk memfasilitasi pencarian rute perjalanan yang optimal.

4.4. Halaman Login Admin

Gambar 3. Halaman Login Admin

Dasbor administrator bertindak sebagai pusat kendali dalam memanipulasi informasi pada sistem rekomendasi pariwisata ini. Guna menjamin integritas data dan mencegah interferensi dari pihak yang tidak berwenang, mekanisme registrasi masuk (*login*) berbasis kredensial *username* dan *password* diwajibkan. Prosedur autentikasi tersebut diterapkan sebagai protokol proteksi untuk membatasi hak akses operasional secara ketat.

4.5. Halaman Input Data Wisata

Gambar 4. Form Input Data Destinasi Wisata

Halaman input data wisata digunakan admin untuk menambahkan informasi destinasi wisata ke dalam sistem. Data yang dimasukkan meliputi nama wisata, kategori wisata, deskripsi tempat, foto wisata, serta koordinat *latitude* dan *longitude*.

4.6. Implementasi Metode Haversine

Metode *Haversine* digunakan untuk menghitung jarak antar lokasi wisata berdasarkan koordinat geografis berupa *latitude* dan *longitude*. Metode ini digunakan karena mampu menghasilkan estimasi jarak yang cukup akurat berdasarkan posisi lokasi di permukaan bumi. Dalam sistem yang dikembangkan, setiap destinasi wisata memiliki data koordinat yang digunakan untuk proses perhitungan jarak antar lokasi wisata. Rumus *Haversine* :

$$d = 2r \arcsin(\sin^2(2\phi/2 - \phi_1) + \cos(\phi_1)\cos(\phi_2)\sin^2(2\lambda/2 - \lambda_1)) \quad (4)$$

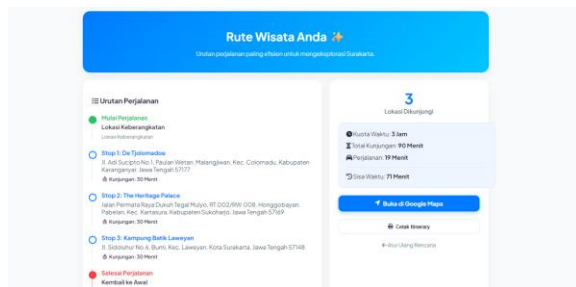
Dimana d = jarak antar lokasi, r = jari-jari bumi, ϕ = *latitude*, dan λ = *longitude*. Hasil perhitungan jarak menggunakan metode *Haversine* kemudian digunakan sebagai bobot (*weight*) pada graf yang diproses menggunakan algoritma Dijkstra untuk menentukan rute perjalanan terpendek. Dengan adanya metode ini, sistem mampu memberikan rekomendasi rute wisata yang lebih efisien berdasarkan lokasi awal dan destinasi wisata yang dipilih pengguna.

4.7. Implementasi Algoritma Dijkstra

Algoritma Dijkstra dioperasikan untuk menentukan jalur perjalanan wisata dengan jarak terpendek berdasarkan bobot hasil perhitungan metode *Haversine*. Sistem akan memproses titik awal perjalanan dan destinasi wisata yang dipilih pengguna untuk menghasilkan rekomendasi *Itinerary* perjalanan wisata satu hari secara lebih efisien.

Gambar 5. Halaman Input Rekomendasi Wisata

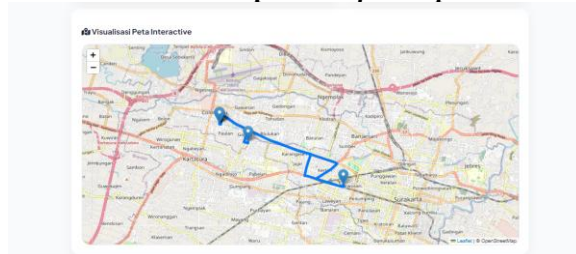
Pada halaman rekomendasi wisata, pengguna dapat memilih kategori wisata, menentukan titik awal perjalanan, serta mengatur durasi perjalanan yang diinginkan sebelum sistem melakukan proses pencarian rute.



Gambar 6. Halaman Rekomendasi Wisata

Setelah data dipilih, sistem akan memproses pencarian rute wisata menggunakan algoritma Dijkstra. Hasil rekomendasi kemudian ditampilkan dalam bentuk daftar destinasi wisata, total jarak perjalanan, serta visualisasi rute pada *Leaflet Maps*. Sistem juga menampilkan *Itinerary* perjalanan wisata sehingga pengguna dapat mengetahui urutan kunjungan destinasi secara lebih terstruktur.

4.8. Visualisasi Rute pada *Leaflet Maps*

Gambar 7. Visualisasi Hasil Rekomendasi Rute Wisata pada *Leaflet Maps*

Guna menyajikan representasi geografis yang dinamis, platform ini mengintegrasikan pustaka *Leaflet Maps* untuk memetakan titik-titik destinasi beserta lintasan perjalanan secara interaktif. Implementasi pemetaan digital tersebut bertujuan untuk mempermudah pengguna dalam menginterpretasikan dan memahami orientasi ruang dari rute yang telah direkomendasikan oleh sistem.

Sistem menampilkan visualisasi rute perjalanan wisata menggunakan *Leaflet Maps*. Garis berwarna biru menunjukkan jalur perjalanan hasil perhitungan algoritma Dijkstra, sedangkan marker menunjukkan lokasi destinasi wisata yang direkomendasikan. Visualisasi tersebut membantu pengguna memahami arah perjalanan dan urutan lokasi wisata secara lebih jelas. Selain itu, sistem juga menampilkan hasil rekomendasi rute yang lebih efisien berdasarkan lokasi awal perjalanan dan destinasi wisata yang dipilih pengguna.

4.9. Pengujian Sistem Menggunakan *Black Box Testing*

Verifikasi fungsionalitas sistem dilaksanakan menggunakan metode *Black Box Testing* untuk memastikan setiap fitur beroperasi sesuai spesifikasi. Pengujian dilakukan dengan memberikan masukan tertentu dan mengobservasi keluaran yang dihasilkan tanpa mempertimbangkan struktur internal kode program. Proses pengujian mencakup fitur *login* admin, pengelolaan data wisata, pencarian rute menggunakan algoritma Dijkstra, hingga visualisasi rute perjalanan pada *Leaflet Maps*.

Tabel 1 Hasil Pengujian *Black Box Testing*

No	Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Login admin	Admin memasukkan username dan password yang benar	Sistem menampilkan dashboard admin	Dashboard admin berhasil ditampilkan	Valid
2	Logout admin	Admin menekan tombol <i>logout</i>	Sistem keluar dari halaman admin	Admin berhasil keluar dari sistem	Valid
3	Menampilkan data wisata	Admin membuka halaman data wisata	Data wisata tampil pada tabel	Data wisata berhasil ditampilkan	Valid
4	Menambahkan data wisata	Admin menginput data wisata baru	Data wisata tersimpan ke database	Data berhasil ditambahkan	Valid
5	Mengedit data wisata	Admin mengubah informasi wisata	Data wisata diperbarui	Data berhasil diperbarui	Valid
6	Menghapus data wisata	Admin menghapus salah satu data wisata	Data wisata terhapus dari sistem	Data berhasil dihapus	Valid
7	Upload foto wisata	Admin mengunggah file gambar wisata	Foto wisata tersimpan dan tampil	Foto berhasil diunggah	Valid
8	Input latitude dan longitude	Admin memasukkan koordinat lokasi	Koordinat tersimpan pada database	Koordinat berhasil disimpan	Valid
9	Memilih kategori wisata	Pengguna memilih kategori wisata	Sistem menampilkan wisata sesuai kategori	Data wisata sesuai kategori tampil	Valid
10	Memilih titik awal perjalanan	Pengguna memilih titik awal	Sistem menerima titik awal perjalanan	Titik awal berhasil diproses	Valid
11	Memilih titik akhir perjalanan	Pengguna memilih titik akhir	Sistem menerima titik akhir perjalanan	Titik akhir berhasil diproses	Valid
12	Input durasi perjalanan	Pengguna memasukkan durasi perjalanan	Sistem memproses durasi perjalanan	Durasi berhasil diproses	Valid
13	Proses pencarian rute menggunakan algoritma Dijkstra	Pengguna menekan tombol cari rute	Sistem menampilkan rute terpendek	Rute berhasil ditampilkan	Valid

No	Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
14	Perhitungan jarak menggunakan metode <i>Haversine</i>	Sistem menghitung jarak antar lokasi	Sistem menghasilkan jarak antar lokasi	Jarak berhasil dihitung	Valid
15	Menampilkan <i>Itinerary</i> wisata	Pengguna melihat hasil <i>Itinerary</i>	Sistem menampilkan urutan perjalanan	<i>Itinerary</i> berhasil ditampilkan	Valid
16	Menampilkan visualisasi rute pada <i>Leaflet Maps</i>	Pengguna membuka hasil rekomendasi	Peta dan jalur perjalanan tampil	Visualisasi peta berhasil ditampilkan	Valid
17	Tombol navigasi <i>Google Maps</i>	Pengguna menekan tombol navigasi	Sistem mengarahkan ke <i>Google Maps</i>	Halaman <i>Google Maps</i> berhasil dibuka	Valid
18	Validasi form input kosong	Pengguna mengosongkan form input	Sistem menampilkan pesan kesalahan	Validasi berhasil ditampilkan	Valid
19	Penyimpanan data ke <i>database</i>	Sistem menyimpan data wisata	Data tersimpan pada <i>database MySQL</i>	Data berhasil disimpan	Valid
20	Responsivitas tampilan halaman	Sistem dibuka pada berbagai ukuran layar	Tampilan menyesuaikan ukuran layar	Tampilan responsif berjalan baik	Valid

Berdasarkan hasil pengujian sebagaimana tercantum pada Tabel 1, keseluruhan 20 skenario pengujian yang mencakup fitur-fitur utama sistem menunjukkan status valid dan beroperasi sesuai ekspektasi. Sistem mampu menjalankan proses pengelolaan data wisata, pencarian rute menggunakan algoritma Dijkstra, perhitungan jarak menggunakan metode *Haversine*, serta visualisasi rute pada *Leaflet Maps* dengan baik. Selain itu, proses validasi input dan penyimpanan data ke *database* juga berjalan sesuai kebutuhan sistem. Dengan demikian, sistem telah memenuhi seluruh persyaratan fungsional dan layak dioperasikan sebagai platform rekomendasi rute wisata berbasis web di Kota Surakarta.

4.10. Maintenance

Tahap *maintenance* dilakukan untuk menjaga sistem rekomendasi wisata tetap berjalan dengan baik setelah implementasi. Proses *maintenance* meliputi pembaruan data destinasi wisata, perbaikan kesalahan sistem, serta penyesuaian fitur apabila terdapat kebutuhan pengembangan lebih lanjut. Selain itu, pemeliharaan basis data juga dilakukan secara berkala untuk menjaga konsistensi dan keamanan data pada sistem rekomendasi wisata berbasis web.

5. KESIMPULAN DAN SARAN

Berdasarkan serangkaian proses penelitian yang telah dilaksanakan, sistem rekomendasi rute wisata Kota Surakarta berbasis web berhasil dikonstruksi dengan mengintegrasikan algoritma Dijkstra dan formula *Haversine* sebagai fondasi komputasi. Sistem terbukti mampu membantu pengguna dalam menyusun rute perjalanan yang teroptimasi berdasarkan preferensi kategori wisata, titik keberangkatan, dan estimasi jarak antarlokasi. Algoritma Dijkstra berhasil mengidentifikasi lintasan dengan akumulasi jarak terpendek, sementara formula *Haversine* menghasilkan estimasi jarak yang presisi berdasarkan koordinat geografis destinasi. Evaluasi melalui *Black Box Testing* mengonfirmasi bahwa seluruh komponen fungsional sistem beroperasi sesuai spesifikasi rancangan. Meskipun demikian, sistem yang dikembangkan masih memiliki sejumlah limitasi,

di antaranya ketidakmampuan mengakomodasi kondisi lalu lintas secara *real-time* dan belum mengintegrasikan preferensi pengguna secara granular. Oleh karena itu, agenda penelitian selanjutnya disarankan untuk mengembangkan sistem dengan menambahkan modul informasi kemacetan, kalkulasi estimasi waktu tempuh dinamis, serta penerapan metode rekomendasi berbasis profil preferensi pengguna agar kualitas rekomendasi yang dihasilkan menjadi lebih adaptif dan komprehensif.

DAFTAR PUSTAKA

- [1] Solopos, "Wisatawan Solo naik jadi 5,6 juta pada 2025, Taman Balekambang salip Solo Safari," Solopos. [Online]. Available: <https://solopos.espos.id/wisatawan-solo-naik-jadi-56-juta-pada-2025-taman-balekambang-salip-solo-safari-2178705>
- [2] Dinas Pariwisata Kota Surakarta, "Profil pariwisata Kota Surakarta 2024," 2024. [Online]. Available: https://pariwisatasolo.surakarta.go.id/wp-content/uploads/2025/07/Buku-Profil-2024_compressed.pdf
- [3] R. Oktafiani and R. Rianto, "Perbandingan Algoritma Support Vector Machine (SVM) dan Decision Tree untuk Sistem Rekomendasi Tempat Wisata," *Jurnal Nasional Sistem Informasi*, vol. 2, pp. 113–121, 2023.
- [4] F. Nugroho and A. Triayudi, "Sistem Pendukung Keputusan Rekomendasi Objek Wisata Menerapkan Metode MABAC dan Pembobotan ROC," vol. 5, no. September, 2023, doi: 10.30865/json.v5i1.6822.
- [5] I. P. G. H. Suputra and L. A. A. R. Putri, "Sistem Rekomendasi Perjalanan Wisata Berbasis Expert System dan Algoritma Genetika," *Jurnal RESISTOR*, vol. 5, no. 1, pp. 47–54, 2022.
- [6] A. Amin and B. Hendrik, "Analisis Penerapan Algoritma Dijkstra dalam Optimasi Penentuan Rute Sebuah Kajian Literatur Sistematis," *J. Educ. Res.*, vol. 6, no. 1, pp. 100–106, 2025, doi: 10.37985/jer.v6i1.2155.
- [7] R. S. A. Wibowo and B. Rahadjeng, "Aplikasi

- Penggunaan Algoritma Dijkstra dalam Masalah Lintasan Terpendek Wisata di Yogyakarta,” *MATHunesa*, vol. 11, no. 3, pp. 468–477, 2023.
- [8] N. Hidayah and U. Khaira, “Implementasi Algoritma Dijkstra untuk Penentuan Rute Terpendek Menuju Fakultas Sains dan Teknologi Universitas Jambi,” *COMSERVA J. Penelit. dan Pengabd. Masy.*, vol. 5, no. 7, pp. 1487–1498, 2025, doi: 10.59141/comserva.v5i7.3411.
- [9] V. Anggraini, P. Sari, A. Sifaunajah, S. Informasi, U. Kh, and A. W. Hasbullah, “Penggunaan Formula *Haversine* Pada Sistem Monitoring Pengiriman Produk Marketplace Sikmajo,” vol. 7, no. 1, pp. 45–52, 2024.
- [10] P. Br Barus, W. Naibaho, R. G. Sitanggang, E. L. Sinaga, and S. P. Sipayung, “Penerapan Algoritma Dijkstra untuk Menentukan Rute Terpendek dari Universitas Katolik Santo Thomas Medan Menuju Perpustakaan Universitas Sumatera Utara,” *J. Pendidik. Tambusai*, vol. 9, no. 2, pp. 20654–20661, 2025.
- [11] A. Sudibyo, R. Sastra, and N. Musyaffa, “Implementasi Algoritma Dijkstra Untuk Menentukan Tempat Wisata dan Hotel Di Sukabumi Berbasis Android,” *Reputasi J. Rekayasa Perangkat Lunak*, vol. 5, no. 1, pp. 34–43, 2024, doi: 10.31294/reputasi.v5i1.3198.
- [12] R. Faurina, E. Sitanggang, P. S. Informatika, F. Teknik, U. Bengkulu, and K. Limun, “Implementasi Metode Content-Based Filtering dan Collaborative Filtering pada Sistem Rekomendasi Wisata di Bali,” vol. 22, no. 4, pp. 870–881, 2023.
- [13] D. Pratiwi and L. Rosnita, “Penerapan Metode Content-Based Filtering dalam Sistem Rekomendasi Objek Wisata di Aceh Tamiang,” vol. 4, no. 2, pp. 85–96, 2024.
- [14] A. Wildana and A. Kautsar, “Sistem Rekomendasi Pemilihan Produk Skincare,” vol. 4, no. 1, pp. 534–543, 2025.
- [15] A. H. Khotimah, “SISTEM REKOMENDASI PENJUALAN KAIN DAN BAJU BATIK DI TOKO BATIK TRESNO DHARMA KARANGANYAR DENGAN MENGGUNAKAN METODE COLLABORATIVE FILTERING,” 2024, *Universitas Duta Bangsa Surakarta*.
- [16] I Made Adi Kampana, *Pengantar Pariwisata*. Penerbit Lakeisha, 2023. [Online]. Available: https://books.google.co.id/books?hl=en&lr=&id=oGrFEAAQAQBAJ&oi=fnd&pg=PA35&dq=pariwisata&ots=5fco0y0CSX&sig=xshIP88RUlcYzCV3K8WFcZrlTII&redir_esc=y#v=onepage&q&f=false
- [17] R. Perayoga, P. Hendradi, and A. Setiawan, “Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek Objek Wisata,” *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 3, pp. 1471–1482, 2023, doi: 10.30865/klik.v4i3.1495.
- [18] E. Christian Rufus, R. Rizkyaka Riyadi, D. Nugraha Hasibuan, E. Christian, and V. Handrianus Pranatawijaya, “Penerapan Algoritma Dijkstra Dalam Menentukan Rute Terpendek Untuk Jasa Pengiriman Barang Di Palangka Raya,” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 3, pp. 3387–3391, 2024, doi: 10.36040/jati.v8i3.9683.
- [19] M. Hasanah and A. Suharso, “Algoritma *Haversine* pada Sistem Informasi Geografis : Tinjauan Literatur Sistematis,” vol. 17, no. July, pp. 135–143, 2023.
- [20] N. Made, A. Esta, D. Wirastuti, L. Verlin, I. Mkwawa, and K. G. Samarah, “Implementation of Geographic Information System Based on Google Maps API to Map Waste Collection Point Using the *Haversine* Formula Method,” vol. 9, no. 3, pp. 731–745, 2023, doi: 10.26555/jiteki.v9i3.26588.