

**SISTEM INFORMASI GEOGRAFIS BERBASIS WEB UNTUK
REKOMENDASI WISATA DI KABUPATEN BANYUMAS
MENGUNAKAN ALGORITMA DIJKSTRA**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan mendapatkan gelar Strata
Satu Program Studi Informatika**



**AKBAR DWI HERTANTO
212221012**

**PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU KOMPUTER
UNIVERSITAS NAHDLATUL ULAMA AL GHAZALI
CILACAP
2025**

**SISTEM INFORMASI GEOGRAFIS BERBASIS WEB UNTUK
REKOMENDASI WISATA DI KABUPATEN BANYUMAS
MENGUNAKAN ALGORITMA DIJKSTRA**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan mendapatkan gelar Strata
Satu Program Studi Informatika**



**AKBAR DWI HERTANTO
20212221012**

**PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU KOMPUTER
UNIVERSITAS NAHDLATUL ULAMA AL GHAZALI
CILACAP
2025**

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa penulisan skripsi dengan judul “Sistem Informasi Geografis Berbasis Web untuk Rekomendasi Wisata di Kabupaten Banyumas Menggunakan Algoritma Djikstra” adalah hasil karya saya dengan arahan dari pembimbing dan belum diajukan kepada pihak manapun. Sumber informasi yang dikutip dalam skripsi ini telah dicantumkan dalam Daftar Pustaka.

Demikian pernyataan ini dibuat dengan sebenarnya. Apabila di kemudian hari terdapat ketidaksesuaian dalam pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Cilacap, 10 Agustus 2026



Akbar Dwi Hertanto
212221012

LEMBAR PENGESAHAN

Skripsi Saudara

Nama : Akbar Dwi Hertanto
NIM : 212221012
Fakultas/Prodi : Fakultas MIKOM / Informatika
Judul : Sistem Informasi Geografis Berbasis Web untuk
Rekomendasi Wisata di Kabupaten Banyumas
menggunakan Algoritma Dijkstra

Telah disidangkan oleh Dewan Penguji Fakultas Matematika dan Ilmu Komputer
Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap pada hari / tanggal :
Jumat, 03 Juli 2026

Dan dapat diterima sebagai pemenuhan tugas akhir mahasiswa Program Strata satu
Program Studi Informatika (S.Kom) Fakultas Matematika dan Ilmu Komputer
(FMIKOM) pada Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap.

Cilacap, 10 Agustus 2026

Dewan Sidang

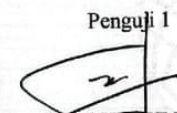
Ketua


Abdul Haq SE M.Cs
NIDN. 0606067701

Sekretaris


Tri Anggoro M.Kom
NIDN. 0601128103

Penguji 1


Abdul Haq SE M.Cs
NIDN. 0606067701

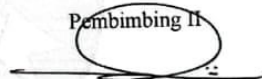
Penguji 2


Verry ST M.Kom
NIDN. 0628068001

Pembimbing I


Mochamad T. A. Aziz Zein S.Si M.Kom
NIDN. 2125098601

Pembimbing II


Tri Anggoro M.Kom
NIDN. 0601128103

Mengetahui,
Dekan Fakultas Matematika dan Ilmu Komputer




Mochamad T. A. Aziz Zein S.Si M.Kom
NIDN. 2125098601

HALAMAN NOTA KONSULTAN

Abdul Haq SE M.Cs

Dosen Fakultas Matematika dan Ilmu Komputer
Universitas Nahdlatul Ulama Al Ghazali Cilacap

NOTA KONSULTAN

Hal : Skripsi Saudara Akbar Dwi Hertanto
Lampiran : -

Kepada
Yth. Dekan Fakultas Matematika dan Ilmu Komputer
Universitas Nahdlatul Ulama Al Ghazali Cilacap
di Cilacap

Assalamu'alaikum Wr. Wb.

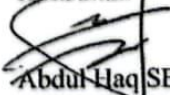
Setelah membaca, memeriksa dan melakukan perbaikan seperlunya maka skripsi saudara/i:

Nama : Akbar Dwi Hertanto
NIM : 212221012
Prodi : Informatika
Judul : Sistem Informasi Geografis Berbasis Web untuk Rekomendasi
Wisata di Kabupaten Banyumas menggunakan Algoritma Dijkstra

Dapat diajukan ke Fakultas Matematika dan Ilmu Komputer, Universitas Nahdlatul Ulama Al Ghazali Cilacap untuk memenuhi syarat memperoleh gelar Strata Satu (S1).

Wassalamu'alaikum Wr. Wb.

Cilacap, 10 Agustus 2026
Konsultan


Abdul Haq SE M.Cs
NIDN. 0606067701

NOTA PEMBIMBING

Cilacap, 05 Maret 2026

Kepada Yth :
Fakultas Matematika dan Komputer (FMIKOM)
UNUGHA Cilacap
di Cilacap

Assalamu 'alaikum Wr. Wb.

Setelah melakukan bimbingan, telaah, arahan dan koreksi tahap penulisan skripsi
saudari:

Nama : Akbar Dwi Hertanto
NIM : 212221012
Fakultas : Matematika dan Ilmu Komputer
Prodi : Informatika
Judul : Sistem Informasi Geografis Berbasis Web untuk Rekomendasi
Wisata di Kabupaten Banyumas Menggunakan Algoritma
Dijkstra

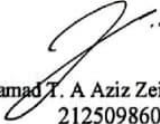
Kami berpendapat bahwa skripsi tersebut sudah dapat diajukan ke sidang skripsi.
Bersamaan ini kami kirimkan skripsi tersebut, semoga dapat segera disidangkan.

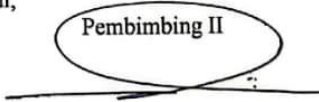
Atas perhatiannya kami ucapkan terima kasih.
Wassalamu 'alaikum Wr. Wb.

Mengetahui,

Pembimbing I

Pembimbing II


Mochamad T. A Aziz Zein S.Si M.Kom
2125098601


Tri Anggoro M.Kom
0601128103

HALAMAN MOTO

“Maka nikmat Tuhan yang manakah yang kamu dustakan?”

(QS. Ar-Rahman : 13)

“The world is a tragedy to those who feel, but a comedy to those who think.”

(Horace Walpole)

“Time is what we want the most but what we use worst.”

(William Penn)

HALAMAN PERSEMBAHAN

Segala puji bagi Allah SWT, Rabb semesta alam yang senantiasa memberikan karunia sehingga penulis mampu menyelesaikan penulisan skripsi ini. Karya ini saya persembahkan kepada :

1. Orang tua (Bapak Anton Murtanto dan Ibu Herin Rosana Yuningrum) yang selalu mendidik saya, memberikan do'a, dukungan, nasihat dan semangat yang tiada henti.
2. Keluarga FMIKOM Angkatan 2021 yang selalu memberikan keceriaan, kebersamaan dan motivasi.
3. Keluarga program studi Informatika, Sistem Informasi, Matematika yang saya banggakan
1. Seluruh teman UNUGHA yang telah memberikan do'a, dukungan, dan semangat.

KATA PENGANTAR

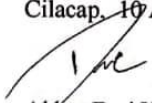
Puji syukur kepada Allah SWT atas segala limpahan nikmat dan karuniaNya, sehingga penulis dapat menyelesaikan penulisan skripsi. Sholawat dan salam senantiasa penulis haturkan kepada Nabi Muhammad SAW sebagai pembimbing seluruh umat manusia.

Skripsi ini tidak akan selesai tanpa adanya bantuan dari banyak pihak, karena itu penulis menyampaikan terima kasih kepada :

1. Bapak M T Abdul Aziz Zein, S.Si, M.Kom Dekan FMikom UNUGHA
2. Bapak M T Abdul Aziz Zein, S.Si, M.Kom pembimbing I yang telah dengan sabar memberikan bimbingan dan petunjuk dalam menyelesaikan skripsi ini.
3. Bapak Tri Anggoro M.Kom, pembimbing II yang telah dengan sabar memberikan bimbingan dan petunjuk dalam menyelesaikan skripsi ini.
4. Bapak Syafiq Rosyad M.Kom, sebagai Pembimbing Akademis
5. Bapak-Ibu dosen Program Studi Informatika FMikom UNUGHA
6. Semua pihak yang tidak dapat kami sebutkan satu persatu

Semoga Allah SWT membalas jerih payah dan pengorbanan yang telah diberikan dengan balasan yang lebih baik. Amiin. Penulis berharap semoga karya kecil ini bermanfaat bagi pembaca.

Cilacap, 10 Agustus 2026



Akbar Dwi Hertanto

ABSTRAK

AKBAR DWI HERTANTO. Sistem Informasi Geografis Berbasis Web Untuk Rekomendasi Wisata Di Kabupaten Banyumas Menggunakan Algoritma Dijkstra. Dibimbing oleh Mochamad Taufiqurrochman Abdul Aziz Zein M.Kom dan Tri Anggoro M.Kom.

Setiap wisatawan memiliki preferensi dan kebutuhan informasi yang berbeda saat merencanakan perjalanan. Namun, banyak daerah wisata masih belum memiliki sistem informasi yang efisien untuk memberikan panduan rute yang optimal ke lokasi wisata. Hal ini dapat menyebabkan kebingungan, ketidakefisienan waktu, serta distribusi kunjungan yang tidak merata antar destinasi. Permasalahan tersebut menjadi latar belakang dalam perancangan dan pembangunan sistem informasi geografis (SIG) berbasis web yang dapat memberikan rekomendasi rute wisata optimal secara interaktif.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan Web GIS yang membantu pengguna dalam menemukan jalur tercepat menuju lokasi wisata di Kabupaten Banyumas. Sistem ini menggunakan algoritma Dijkstra sebagai metode utama untuk menghitung rute terpendek berdasarkan jarak antar titik pada jaringan jalan. Data yang digunakan mencakup koordinat lokasi wisata dan jalur konektivitas antar titik, yang ditampilkan dalam peta interaktif berbasis Leaflet. Dengan adanya sistem ini, wisatawan dapat melihat visualisasi rute, estimasi jarak tempuh, dan informasi destinasi secara real-time.

Selain itu, sistem ini memberikan manfaat dalam mendukung digitalisasi sektor pariwisata lokal dan membantu pemerataan kunjungan ke berbagai destinasi wisata. Pemerintah daerah juga dapat memanfaatkan sistem ini untuk menyusun strategi promosi berbasis data spasial. Dengan demikian, diharapkan sistem ini mampu meningkatkan kualitas pengalaman wisatawan dan memperkuat pengelolaan pariwisata di Kabupaten Banyumas secara lebih terintegrasi dan modern.

Kata kunci: Web GIS, Algoritma Dijkstra, Rekomendasi Wisata, Sistem Informasi Geografis, Leaflet

ABSTRACT

AKBAR DWI HERTANTO. Web-Based Geographic Information System for Tourism Recommendations in Banyumas Regency Using the Dijkstra Algorithm. Supervised by Ninik Agustin M.Sc and Tri Anggoro M.Kom.

Every tourist has different preferences and information needs when planning a trip. However, many tourist areas still lack an efficient information system to provide optimal route guidance to tourist locations. This can lead to confusion, time inefficiencies, and uneven distribution of visits between destinations. These issues underlie the design and development of a web-based geographic information system (GIS) that can provide interactive recommendations for optimal tourist routes.

This research aims to design and implement a web GIS that helps users find the fastest routes to tourist locations in Banyumas Regency. The system uses the Dijkstra algorithm as the primary method for calculating the shortest route based on the distance between points on the road network. The data used includes the coordinates of tourist locations and connectivity paths between points, which are displayed on an interactive Leaflet-based map. With this system, tourists can view route visualizations, estimated travel distances, and destination information in real time.

Furthermore, this system provides benefits in supporting the digitalization of the local tourism sector and helping to distribute visits to various tourist destinations equitably. Local governments can also utilize this system to develop promotional strategies based on spatial data. Thus, it is hoped that this system will improve the quality of the tourist experience and strengthen tourism management in Banyumas Regency in a more integrated and modern way.

Keywords: Web GIS, Dijkstra Algorithm, Tourism Recommendations, Geographic Information System, Leaflet

DAFTAR ISI

HALAMAN PERNYATAAN KEASLIAN SKRIPSI	Error! Bookmark not defined.
LEMBAR PENGESAHAN	Error! Bookmark not defined.
HALAMAN NOTA KONSULTAN.....	Error! Bookmark not defined.
HALAMAN NOTA PEMBIMBING.....	v
HALAMAN MOTO	vii
HALAMAN PERSEMBAHAN	viii
KATA PENGANTAR	Error! Bookmark not defined.
ABSTRAK.....	x
<i>ABSTRACT</i>	xi
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah.....	2
C. Batasan Masalah	3
D. Tujuan Penelitian	3
E. Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	4
A. Penelitian Terkait	4
B. Landasan Teori.....	6
BAB III METODOLOGI.....	9
A. Waktu dan Tempat Penelitian.....	9
1. Waktu Penelitian	9
2. Tempat Penelitian.....	9
B. Prosedur Penelitian	9
C. Analisis Hasil	18
D. Jadwal Penelitian	18
BAB IV HASIL DAN PEMBAHASAN.....	20
A. Hasil	20

1. Pengumpulan Data	20
2. Implementasi Sistem	21
1) Halaman Login.....	31
3. Pengujian Sistem.....	37
B. Pembahasan.....	39
BAB V KESIMPULAN.....	43
A. Kesimpulan	43
B. Saran/Rekomendasi.....	44
DAFTAR PUSTAKA	46
LAMPIRAN.....	47

DAFTAR GAMBAR

Gambar 1 Mind Mapping Aplikasi Web GIS	6
Gambar 2 Diagram Alur Penelitian	10
Gambar 3 <i>use case diagram</i>	12
Gambar 4 <i>Activity diagram User</i>	13
Gambar 5 <i>Activity Diagram Admin</i>	14
Gambar 6 Diagram Sequence Login	15
Gambar 7 Diagram Sequence Input	15
Gambar 8 Diagram Sequence Update	16
Gambar 9 Diagram Sequence Delete	16
Gambar 10 Timeline dan Gantt Chart	19
Gambar 11 Halaman <i>Dashboard User</i>	22
Gambar 12 Halaman Explore Now	23
Gambar 13 Halaman Rekomendasi Wisata	27
Gambar 14 Halaman Login	31
Gambar 15 Halaman Dashboard Admin	32
Gambar 16 Halaman Data Wisata	33
Gambar 17 Halaman Tambah Data Wisata	34
Gambar 18 Halaman Edit Data Wisata	35
Gambar 19 Halaman Import CSV	36
Gambar 20 Halaman Statistik	37

DAFTAR TABEL

Tabel 1 Penelitian Terkait	4
Tabel 2 Spesifikasi Data.....	20
Tabel 3 Contoh Data yang diambil	21
Tabel 4 Uji Sistem Menggunakan <i>Black Box Testing</i>	38

BAB I PENDAHULUAN

A. Latar Belakang

Pariwisata merupakan salah satu sektor unggulan yang memberikan kontribusi signifikan terhadap pertumbuhan ekonomi daerah. Kabupaten Banyumas dengan keanekaragaman destinasi wisata seperti Baturaden, Curug Cipendok, dan Small World Purwokerto, memiliki potensi besar untuk terus dikembangkan. Namun, potensi ini belum sepenuhnya dioptimalkan karena keterbatasan dalam pengelolaan informasi dan penyediaan layanan navigasi wisata yang efektif. Transformasi digital di sektor pariwisata menjadi kebutuhan yang tidak bisa ditunda untuk meningkatkan kualitas pengalaman wisatawan. Seiring perkembangan teknologi informasi, Sistem Informasi Geografis (SIG) muncul sebagai inovatif dalam memetakan dan menganalisis data berbasis lokasi. SIG mampu memberikan informasi yang akurat tentang posisi, jarak, dan rute menuju lokasi tertentu, sehingga sangat relevan untuk diterapkan dalam konteks pariwisata[1]. Data dari Global Industry Analysts (2023) menunjukkan bahwa pasar global SIG diproyeksikan tumbuh hingga USD 20,7 miliar pada 2025, menandakan kebutuhan yang meningkat untuk sistem berbasis lokasi dalam berbagai sektor, termasuk pariwisata. Pemanfaatan WebGIS dalam sektor pariwisata telah terbukti efektif dalam mempermudah wisatawan mendapatkan informasi destinasi secara visual dan interaktif, sebagaimana dikembangkan dalam sistem informasi destinasi Kabupaten Kotabaru[2] serta pemetaan interaktif di wilayah Jawa Timur[3].

Meskipun Banyumas telah memiliki berbagai media promosi wisata seperti brosur dan portal online, belum tersedia sistem yang mampu merekomendasikan rute tercepat ke berbagai destinasi berdasarkan posisi aktual wisatawan. Kondisi ini menyulitkan wisatawan dalam menentukan destinasi yang optimal dan jalur terbaik, terlebih karena kondisi geografis Banyumas yang bervariasi. Selain itu, ketidaktersediaan sistem navigasi lokal berbasis data aktual turut menghambat distribusi wisatawan secara merata. Di sisi lain, munculnya tempat wisata baru yang menjadi tren di media sosial dan media cetak sering kali membuat wisatawan berkunjung hanya karena tren, bukan karena ketertarikan pribadi. Akibatnya, tempat wisata lama cenderung ditinggalkan. Oleh karena itu, dibutuhkan sistem yang tidak hanya menyediakan navigasi optimal, tetapi juga mampu merekomendasikan alternatif objek wisata berdasarkan preferensi pengguna, bukan sekadar tren sesaat[4]. Efisiensi perjalanan menjadi aspek krusial bagi wisatawan, terutama dalam menentukan rute terpendek menuju lokasi bersejarah atau wisata religi guna mengoptimalkan waktu kunjungan[5][6].

Dalam upaya meningkatkan kemudahan akses informasi rute pariwisata, penerapan algoritma graf seperti Dijkstra menjadi solusi yang sangat potensial. Algoritma Dijkstra telah terbukti efektif dalam menyelesaikan masalah pencarian jalur terpendek dalam berbagai aplikasi sistem navigasi berbasis SIG[7]. Dengan algoritma ini, sistem dapat menghitung rute tercepat dengan mempertimbangkan berbagai faktor seperti jarak dan konektivitas jalan, sehingga wisatawan dapat menghemat waktu dan menikmati lebih banyak destinasi. Berkat kemampuannya tersebut, algoritma Dijkstra menjadi salah satu metode yang tepat untuk digunakan dalam mengatasi permasalahan pencarian rute terpendek[8].

Beberapa penelitian sebelumnya telah mengaplikasikan algoritma Dijkstra untuk rekomendasi lokasi fasilitas umum, seperti SPBU dan bengkel, dengan hasil yang memuaskan[9]. Namun, penerapan algoritma ini dalam sistem informasi geografis khusus untuk pariwisata lokal, seperti di Banyumas, masih minim dieksplorasi. Hal ini menunjukkan adanya kesenjangan riset sekaligus peluang untuk mengembangkan sistem yang dapat membantu wisatawan menentukan destinasi wisata berdasarkan jarak optimal dari lokasi mereka saat ini.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan membangun sebuah sistem informasi geografis berbasis algoritma Dijkstra yang mampu memberikan rekomendasi rute wisata terbaik di Kabupaten Banyumas. Sistem ini akan memanfaatkan data spasial destinasi wisata, jaringan jalan, serta algoritma penelusuran jalur terpendek untuk memberikan pengalaman wisata yang lebih terencana, nyaman, dan efisien bagi para pengguna. Pengembangan sistem yang interaktif sangat diperlukan agar analisis spasial fasilitas umum dapat dilakukan dengan lebih akurat dan memudahkan pengguna dalam pengambilan keputusan[10].

Secara teoritis, penelitian ini akan berkontribusi dalam pengembangan aplikasi SIG berbasis algoritma graf untuk sektor pariwisata. Sementara itu, secara praktis, sistem ini diharapkan dapat menjadi solusi inovatif bagi pemerintah daerah dan pelaku wisata dalam mengelola arus wisatawan secara lebih efektif, meningkatkan tingkat kepuasan pengunjung, serta memperkuat daya saing pariwisata Banyumas di era digital.

B. Rumusan Masalah

Berdasarkan latar belakang, rumusan masalah penelitian ini adalah sebagai berikut:

1. Bagaimana penerapan algoritma djikstra pada rancang bangun SIG untuk rekomendasi rute wisata optimal?

2. Bagaimana penerapan algoritma djikstra dapat meningkatkan efisiensi waktu tempuh wisatawan?

C. Batasan Masalah

1. Penelitian hanya pada kawasan wisata yang berada di wilayah administratif Kabupaten Banyumas
2. Sistem ini hanya memuat data objek wisata yang berada di Kabupaten Banyumas dan memiliki akses jalan yang bisa dilalui oleh kendaraan roda empat ke atas
3. Algoritma yang digunakan dibatasi hanya pada algoritma djikstra
4. Data spasial bersumber dari data terbuka seperti OpenStreetMap
5. Aplikasi berbasis web

D. Tujuan Penelitian

1. Untuk menerapkan algoritma djikstra dalam merancang dan membangun sistem informasi geografis yang mampu merekomendasikan rute wisata optimal di Kabupaten Banyumas.
2. Untuk meningkatkan efisiensi waktu tempuh wisatawan melalui penerapan algoritma djikstra dalam sistem pencarian rute wisata.
3. Untuk mengembangkan sistem berbasis preferensi pengguna yang dapat mendistribusikan kunjungan wisatawan secara lebih merata di berbagai destinasi wisata.

E. Manfaat Penelitian

- 1) Manfaat teoritis dalam penelitian ini yaitu:
 - a. Penguatan model perhitungan rute optimal dalam SIG berbasis graf
 - b. Integrasi teori algoritma pencarian jalur dengan sistem informasi lokasi
- 2) Manfaat praktis dalam penelitian ini yaitu:
 - a. Membantu wisatawan memilih rute tercepat ke tempat wisata
 - b. Meningkatkan daya saing daerah wisata melalui inovasi digital

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

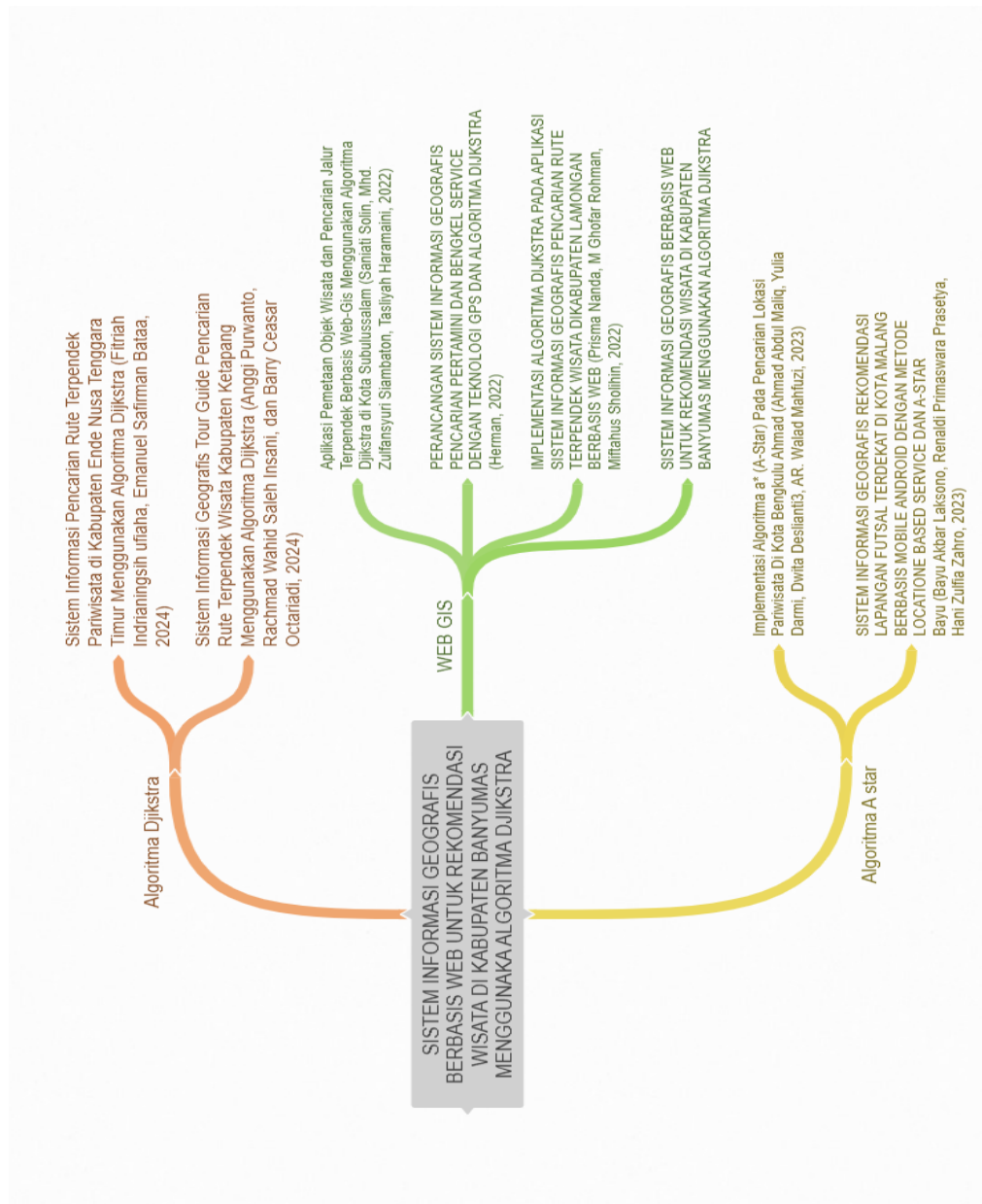
Beberapa penelitian terkait dengan Sistem Informasi Geografis (SIG) untuk rekomendasi wisata, penelitian tersebut antara lain :

Tabel 1 Penelitian Terkait

No.	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
1	Herman (2022)	Judul: Perancangan Sistem Informasi Geografis Pencarian Pertamina dan Bengkel Service Dengan Teknologi GPS dan Algoritma Dijkstra. Persamaan: Sama-sama menggunakan algoritma djikstra, sistem berbasis SIG Perbedaan: bukan objek wisata melainkan SPBU dan bengkel
2	Prisma Nanda B, M Ghofar Rohman, Miftahus Sholihin (2022)	Judul : Implementasi Algoritma Dijkstra pada Aplikasi Sistem Informasi Geografis Pencarian Rute terpendek Wisata Di Kabupaten Lamongan Berbasis Web. Persamaan: sistem berbasis SIG, menggunakan algoritma djikstra, aplikasi berbasis web Perbedaan: beda wilayah penelitian
3	Saniati Solin, Mhd. Zulfansyuri Siambaton, Tasliyah Haramaini (2022)	Judul : Aplikasi Pemetaan Objek Wisata dan Pencarian Jalur Terpendek Berbasis Web-GIS Menggunakan Algoritma Dijkstra di Kota Subulussalam. Persamaan: menggunakan algoritma djikstra, sistem berbasis SIG Perbedaan: beda wilayah dan tidak menggunakan preferensi pengguna
4	Fitriah Indianingsih Ufiah, Emanuel Safirman Bata (2024)	Judul : Sistem Informasi Pencarian Rute Terpendek Pariwisata di Kabupaten Ende Nusa Tenggara Timur Menggunakan Algoritma Dijkstra.

		Persamaan: menggunakan algoritma djikstra, aplikasi berbasis web GIS Perbedaan: beda lokasi penelitian dan tidak menggunakan preferensi pengguna
5	Anggi Purwanto, Rachmad Wahid Saleh Insani, Barry Caesar Octariadi (2024)	Judul : Sistem Informasi Geografis Tour Guide Pencarian Rute Terpendek Wisata Kabupaten ketapang Menggunakan Algoritma Dijkstra. Persamaan: fokus di wisata dan menggunakan algoritma djikstra Perbedaan: tidak menggunakan preferensi pengguna dan fungsi rekomendasi
6	Ahmad Abdul Maliq, Yulia Darmi, Dwita Deslianti, AR. Walad Mahfuzi (2023)	Judul : Implementasi Algoritma A* (A-Star) Pada Pencarian Lokasi Pariwisata Di Kota Bengkulu. Persamaan: fokus di wisata dan berbasis web GIS Perbedaan: berbeda algoritma
7	Bayu Akbar Laksono, Renaldi Primaswara Prasetya, Hani Zulfia Zahro (2023)	Judul : Sistem Informasi Geografis Rekomendasi Lapangan Futsal Terdekat Di Kota Malang Berbasis Mobile Android Dengan Metode Location Based Service dan A-Star. Persamaan: sama sama aplikasi berbasis lokasi Perbedaan: bukan objek wisata melainkan lokasi olahraga, beda platform, beda algoritma

Diagram *mind map* dicontohkan pada gambar berikut ini



Gambar 1 Mind Mapping Aplikasi Web GIS

B. Landasan Teori

1. Web GIS

Web GIS Merupakan bentuk sistem informasi geografis yang dapat diakses melalui web, di mana user dapat menampilkan serta berinteraksi langsung dengan data spasial yang telah tersedia[11]. Web GIS telah diterapkan secara luas untuk berbagai kebutuhan monitoring dan pemetaan, mulai dari persebaran pembangunan desa[12], monitoring lahan pertanian[13], hingga pengelolaan retribusi menara telekomunikasi[14], dan pemetaan fasilitas rumah ibadah[15]. Sistem ini dapat menggabungkan data spasial dan atribut wisata

dalam satu platform yang interaktif dan mudah diakses, dengan dukungan visualisasi peta menggunakan leaflet.js sebagai media pemetaan.

2. Algoritma Dijkstra

Algoritma djikstra dikenal sebagai metode yang efisien untuk menghitung jarak terdekat antara dua lokasi. Algoritma ini digunakan untuk menemukan jalur terpendek dalam struktur graf, baik yang berarah maupun tak berarah, dengan bobot tertentu, tanpa harus mengevaluasi semua kemungkinan jalur yang ada. Algoritma Dijkstra merupakan salah satu algoritma terbaik dalam pencarian rute terpendek karena mampu memberikan hasil yang optimal pada graf berbobot, baik untuk navigasi antar lokasi pendidikan[16] maupun optimasi rute pengiriman logistik[17]. Ini dari algoritma ini adalah memilih lintasan dengan jarak paling kecil secara bertahap hingga mencapai tujuan. Penerapannya umum digunakan dalam layanan berbasis lokasi pada platform web untuk mempermudah navigasi pengguna[18].

3. PHP

PHP merupakan bahasa pemrograman sisi server yang dirancang untuk berintegrasi dengan HTML dalam pembuatan halaman web yang bersifat dinamis. Salah satu keunggulannya adalah kemampuan menyembunyikan kode program dari pengguna, sehingga meningkatkan aspek keamanan situs. PHP memungkinkan konten halaman web berubah secara real-time berdasarkan permintaan, seperti menampilkan data dari basis data. Selain itu, PHP juga mendukung koneksi ke berbagai sistem manajemen basis data (DBMS), menjadikannya solusi yang fleksibel untuk pengembangan aplikasi web interaktif[19].

4. MySQL

MySQL merupakan sistem manajemen basis data relasional yang bersifat open-source, yang memungkinkan siapa saja untuk mengakses dan mengubah kode sumbernya. Awalnya dikembangkan oleh perusahaan asal Swedia, MySQL AB, sistem ini kemudian diakuisisi oleh Oracle Corporation. MySQL banyak dipilih dalam pengembangan berbagai jenis aplikasi, mulai dari situs web hingga sistem bisnis, karena menawarkan kecepatan, kestabilan, dan kemudahan dalam penggunaannya[20].

5. Website

Website merupakan kumpulan halaman yang menyajikan beragam jenis informasi, seperti gambar, video, teks, maupun gabungan dari semua itu. Konten yang ditampilkan bisa bersifat statis, yaitu jarang diperbarui dan hanya menyampaikan informasi satu arah dari pemilik website. Sementara itu, apabila

isi website sering diperbarui dan bersifat interaktif, maka website tersebut dikategorikan sebagai dinamis[21].

6. *React.js*

React.js merupakan *library* javascript open-source yang dikembangkan oleh facebook dan telah menjadi salah satu pilihan yang populer di kalangan web developer. *React.js* digunakan untuk membangun tampilan antarmuka pengguna yang responsif dan efisien dalam pengembangan aplikasi berbasis web[22].

7. *Leaflet.js*

Leaflet.js merupakan aplikasi yang mampu menampilkan informasi yang berbasis lokasi atau referensi geografis. *Leaflet.js* merupakan *library* dari javascript yang dirancang khusus untuk membangun peta interaktif di dalam halaman website. Dalam visualisasi peta, penggunaan *library* Leaflet dan *Web Mapping* berbasis *client-side* memberikan keunggulan berupa performa yang lebih ringan dan responsif saat diakses oleh pengguna[23]. Selain bersifat open-source, LeafletJS memiliki keunggulan berupa fitur yang extensible dan kompatibel dengan berbagai ekstensi tambahan yang memperluas fungsionalitasnya. Keefisienan LeafletJS juga terlihat dari kemampuannya untuk menampilkan lapisan peta yang dapat dikendalikan secara langsung oleh pengguna melalui fitur kontrol layer[24].

8. Laravel

Laravel merupakan salah satu framework PHP berbasis Model View Controller (MVC) yang dikembangkan oleh Taylor Otwell pada tahun 2011. Framework ini dirancang untuk mempermudah proses pengembangan website dengan sintaks yang jelas dan efisien, serta membantu mengurangi biaya pengembangan dan pemeliharaan aplikasi[25].

BAB III METODOLOGI

A. Waktu dan Tempat Penelitian

1. Waktu Penelitian

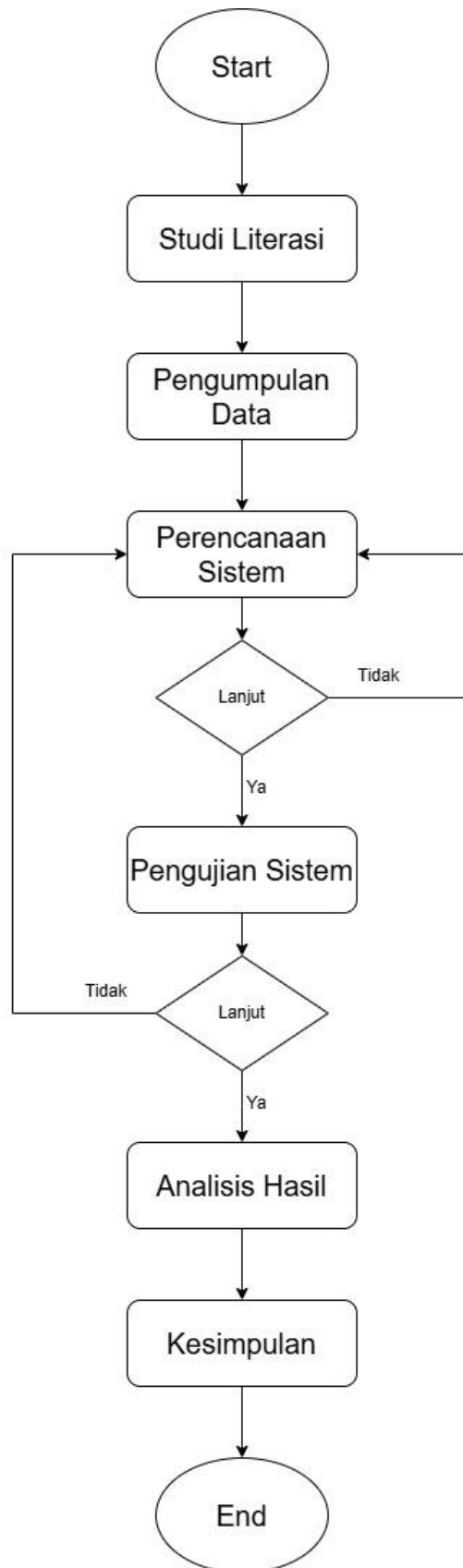
Penelitian ini dilakukan pada bulan April 2025 sampai dengan bulan Februari 2026. Tahapan penelitian akan dibagi menjadi beberapa proses yaitu tahap persiapan yang akan dilakukan selama bulan Mei 2025 sampai dengan bulan Juni 2025, proses kedua yaitu pengumpulan data dari bulan Mei hingga bulan Juni, proses ke 3 yaitu pengolahan data dan *deployment* aplikasi dari bulan Juni 2025 sampai dengan bulan Februari 2026.

2. Tempat Penelitian

Tempat Penelitian berada di Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Banyumas. Instansi ini dipilih karena memiliki keterkaitan langsung dengan sektor pariwisata dan menjadi lembaga yang menaungi berbagai kegiatan dan kebijakan terkait pengembangan destinasi wisata di wilayah Banyumas. Kabupaten Banyumas sendiri memiliki kekayaan objek wisata yang cukup beragam, sehingga sesuai dijadikan sebagai lokasi studi untuk mendukung sistem berbasis informasi geografis dalam bidang pariwisata.

B. Prosedur Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang sistematis untuk memastikan hasil yang akurat dan sesuai tujuan penelitian. Adapun tahapan yang akan dilakukan adalah sebagai berikut:



Gambar 2 Diagram Alur Penelitian

1. Studi Literasi

Tahap ini melibatkan proses pengumpulan berbagai sumber referensi seperti buku, artikel, jurnal ilmiah, makalah, dan sumber daring yang relevan dengan topik penelitian. Tujuannya adalah untuk memperoleh dasar teori yang kuat, memahami hasil-hasil penelitian sebelumnya, serta mengidentifikasi celah atau permasalahan yang belum banyak dikaji sebagai landasan dalam pengembangan penelitian selanjutnya.

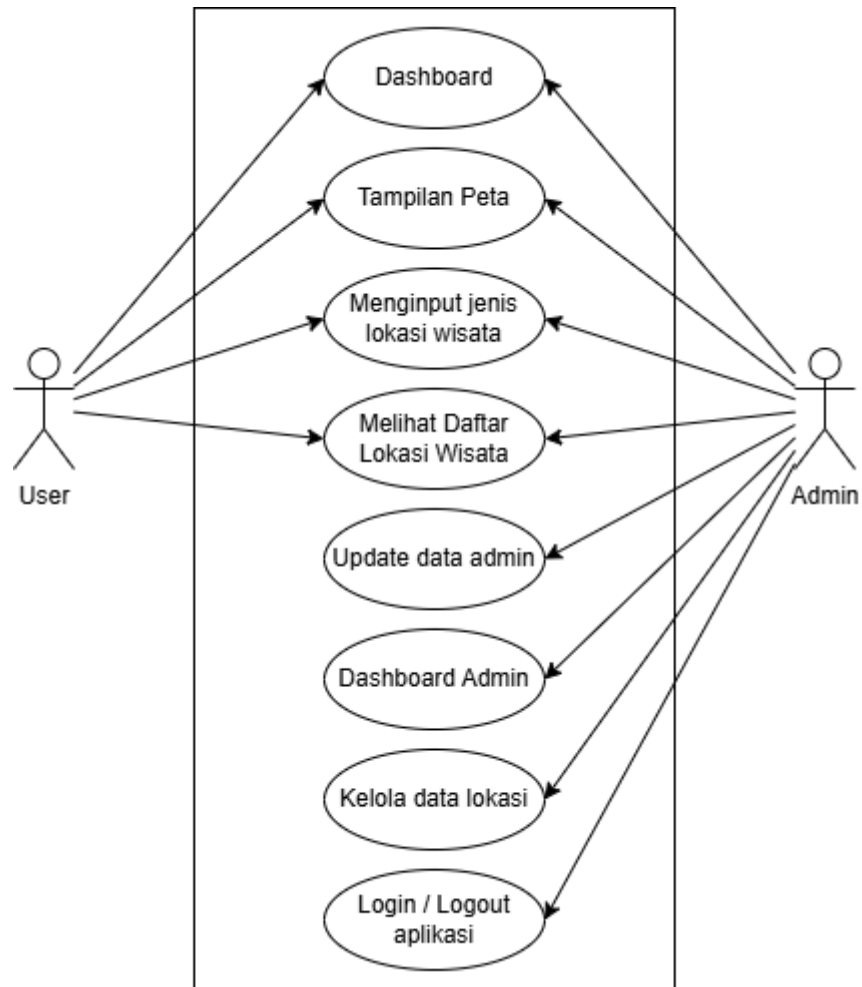
2. Pengumpulan Data

Data yang digunakan dalam penelitian ini diperoleh dari dua sumber utama, yaitu dari Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Banyumas sebagai sumber resmi, serta dari berbagai situs web pariwisata yang relevan. Pengambilan data dilakukan untuk mencakup informasi seputar objek wisata seperti nama lokasi, jenis wisata, alamat, dan koordinat geografis. Penambahan sumber dari website dilakukan untuk melengkapi data yang tidak tercantum secara resmi di dinas, sehingga cakupan destinasi wisata dalam sistem menjadi lebih luas dan representatif.

3. Perancangan sistem

a. Use Case Diagram

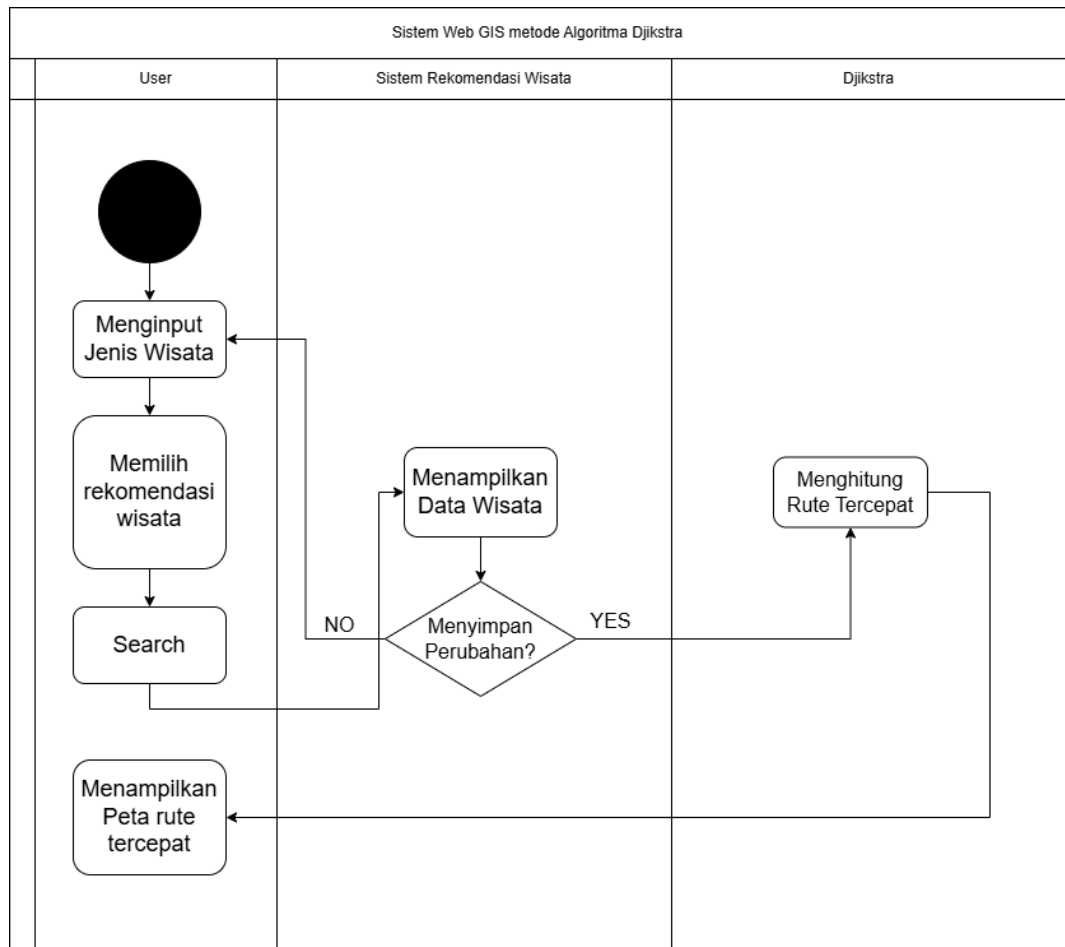
Use case diagram pada sistem ini menggambarkan alur proses dalam platform Web GIS untuk rekomendasi rute wisata menggunakan Algoritma Dijkstra. Sistem ini terdiri beberapa halaman, antara lain tampilan peta wisata, daftar destinasi, dan dashboard utama. User hanya memiliki akses untuk melihat tampilan peta interaktif, daftar objek wisata, serta menerima rekomendasi rute wisata terpendek. Sementara itu, admin memiliki akses tambahan ke halaman khusus seperti dashboard admin serta pengelolaan data wisata, yang mencakup fungsi untuk menambahkan, menghapus, dan memperbarui data destinasi dan jalur rute.



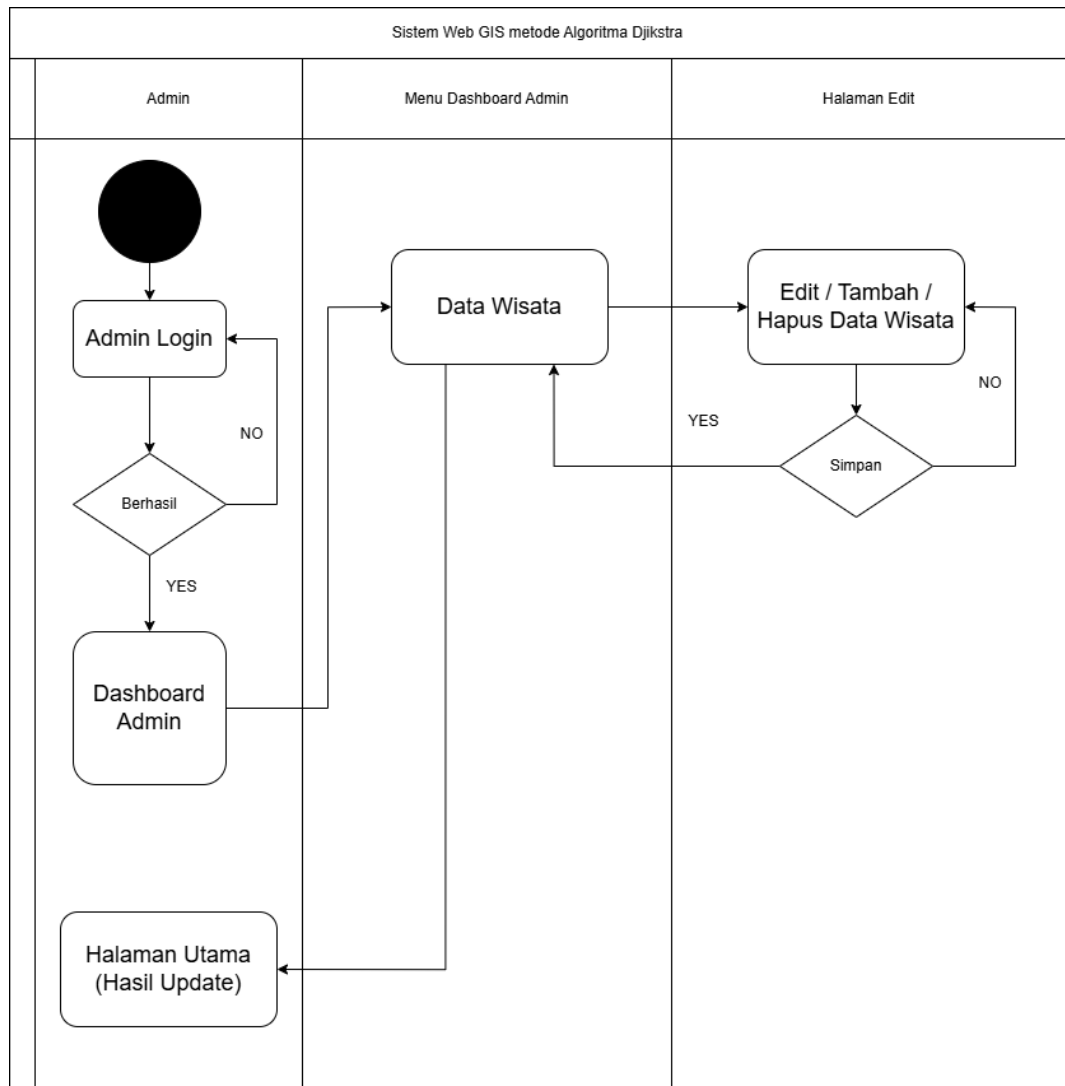
Gambar 3 use case diagram

b. *Activity Diagram Peta*

Pada *activity diagram* peta di bawah ini menggambarkan aktivitas yang hanya bisa dilakukan oleh user dalam pengelolaan sistem, bagaimana seorang user bisa memilih rekomendasi wisata dan mendapatkan rute tercepat.



Gambar 4 Activity diagram User



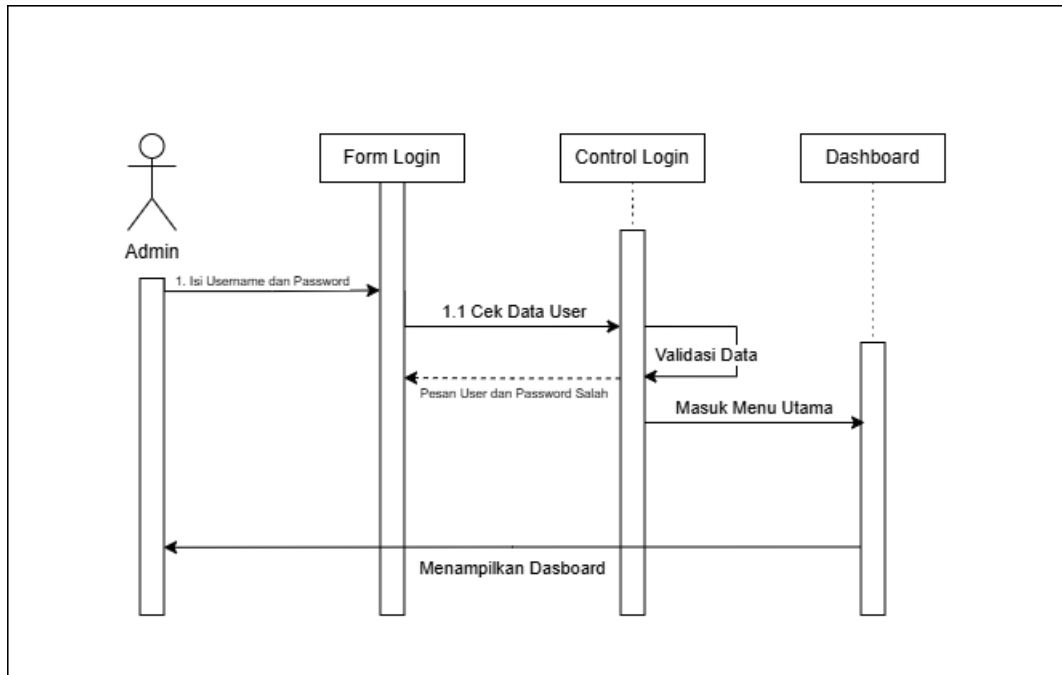
Gambar 5 Activity Diagram Admin

c. *Sequence Diagram*

Pada tahap ini *sequence diagram* dibuat agar dapat mempermudah melihat interaksi antara admin, sistem dan *database* secara dinamis

1) Diagram *Sequence Login*

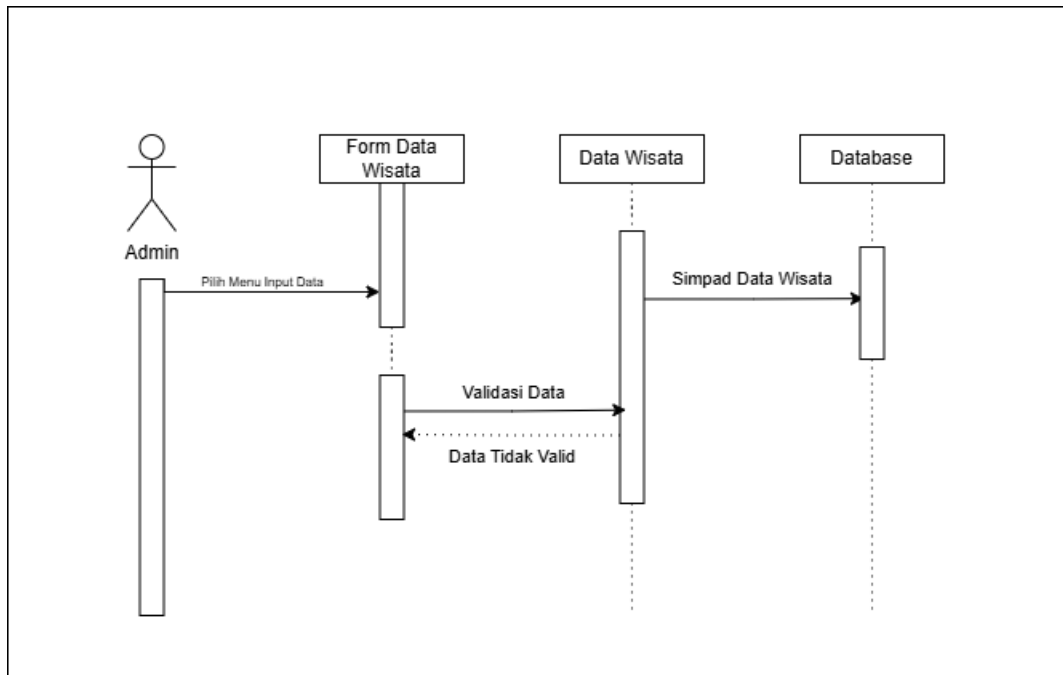
Untuk mengelola data dalam Web GIS, admin harus melalui tahapan login, admin harus melakukan login dengan mengisi username dan password. Tahapan login ini dimulai dengan masuk ke halaman login lalu mengisi *username* dan *password*, jika login berhasil, admin akan diarahkan ke halaman *dashboard* admin.



Gambar 6 Diagram Sequence Login

2) Diagram *Sequence* Input

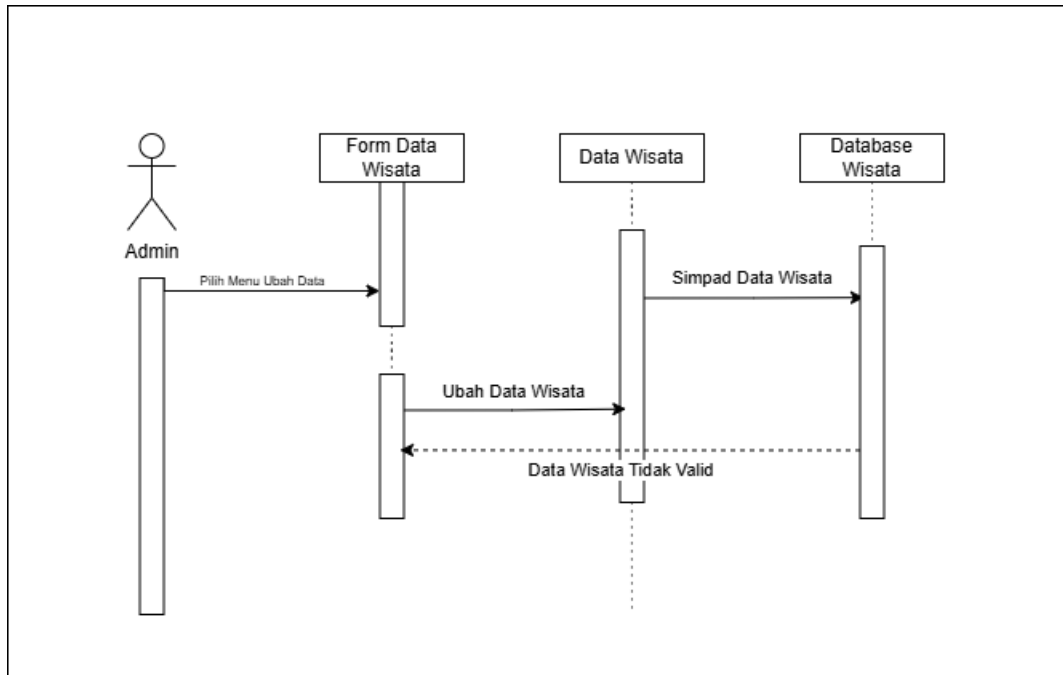
Setelah admin berhasil masuk, maka admin dapat menambah data lokasi wisata. Pilih menu tambah data, selanjutnya diarahkan untuk mengisi formulir klik simpan, sistem akan mengecek data yang dimasukkan. Jika data sudah benar maka akan disimpan di *database*



Gambar 7 Diagram Sequence Input

3) Diagram *Sequence Update*

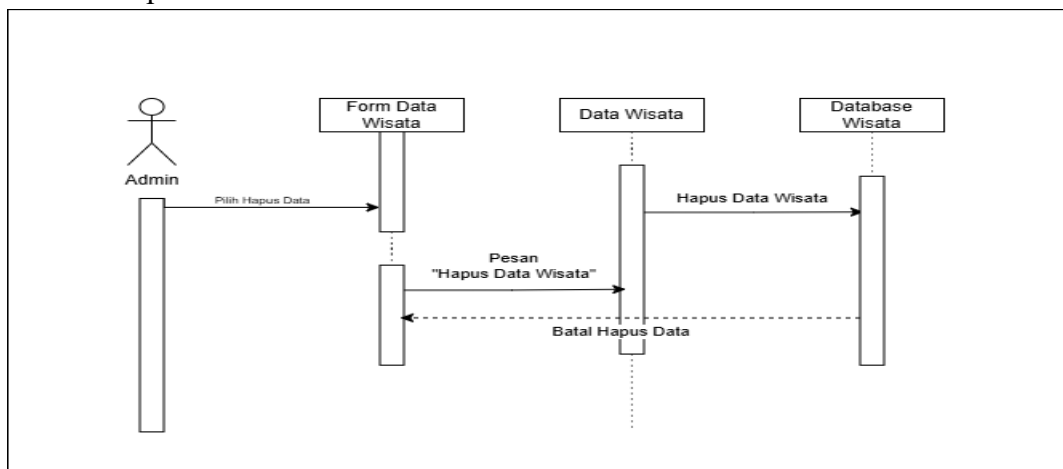
Setelah admin masuk, admin dapat mengubah data yang ada. admin dapat mengubah data dengan cara memilih menu ubah data, lalu isi *form* daftar wisata setelah itu mengklik simpan, sistem akan memvalidasi data benar lalu data tersebut akan disimpan ke *database*.



Gambar 8 Diagram Sequence Update

4) Diagram *Sequence Delete*

Setelah admin melakukan login admin dapat menghapus data yang tersimpan. Admin dapat menghapus data dengan cara pilih wilayah lalu ada konfirmasi “apakah yakin akan dihapus” maka data akan terhapus di *database*.



Gambar 9 Diagram Sequence Delete

d. Perancangan Metode Dijkstra

Sistem yang dibuat memiliki langkah atau cara kerja sebagai berikut:

- 1) Sistem menerima masukan data lokasi objek wisata dan data jaringan jalan yang membentuk graf berbobot, terdiri dari simpul (nodes) dan sisi (edges) yang menunjukkan jarak antar titik.
- 2) Data lokasi dan jalur diubah ke dalam bentuk graf berarah, dengan masing masing titik mewakili lokasi objek wisata dan lokasi pengguna, dan sisi yang menghubungkannya diberi bobot berdasarkan jarak tempuh.
- 3) Sistem menghitung rute terpendek dari lokasi pengguna menuju lokasi wisata menggunakan algoritma djikstra.
- 4) Untuk menghitung jalur terpendek, sistem menerapkan rumus perhitungan berbasis bobot antar simpul:

$$D(i) = \min(D(i), D(u) + w(u, i))$$

Keterangan:

- $D(i)$ = jarak terpendek dari awal ke simpul i
 - $D(u)$ = jarak dari titik awal ke simpul u
 - $W(u, i)$ = bobot / jarak dari simpul u ke simpul i
- 5) Setelah diperoleh rute terpendek, sistem menampilkan hasil berupa rute wisata pada peta yang interaktif, disertai estimasi jarak waktu tempuh.
 - 6) Jika pengguna memilih Lokasi wisata yang baru atau mengubah preferensi Lokasi, sistem akan mengulang proses perhitungan dari Langkah ke-3 hingga menghasilkan rute yang diperbarui sesuai tujuan baru.

4. Pengujian Data

Pengujian data pada sistem ini dilakukan untuk mengevaluasi keakuratan dan konsistensi hasil perhitungan rute terpendek yang dihasilkan oleh algoritma djikstra. Tujuan dari pengujian ini adalah untuk memastikan bahwa algoritma dapat menentukan jalur optimal dari lokasi pengguna menuju berbagai objek wisata dengan jarak tempuh paling efisien berdasarkan struktur graf jalan yang digunakan.

Pengujian dengan membandingkan hasil perhitungan rute dari sistem terhadap rute-rute alternatif yang mungkin tersedia, serta memverifikasi apakah jalur yang dihasilkan benar-benar memiliki bobot (jarak) paling kecil. Selain itu, pengujian juga mencakup validasi urutan simpul yang dilalui, serta perbandingan total jarak tempuh antar kombinasi jalur.

Berikut adalah rumus yang digunakan dalam proses perhitungan algoritma Dijkstra:

$$D(i) = \min(D(i), D(u) + w(u, i))$$

Keterangan:

- $D(i)$ = jarak terpendek dari awal ke simpul i
- $D(u)$ = jarak dari titik awal ke simpul u
- $W(u, i)$ = bobot / jarak dari simpul u ke simpul i

C. Analisis Hasil

Pada penelitian ini, sistem menghasilkan visualisasi peta wilayah Kabupaten Banyumas dalam bentuk Web GIS interaktif. Peta tersebut menampilkan lokasi-lokasi objek wisata yang terhubung melalui jalur jaringan jalan. Sistem memungkinkan pengguna untuk memilih lokasi awal dan tujuan wisata, kemudian menampilkan rute terpendek yang dihitung menggunakan algoritma Dijkstra.

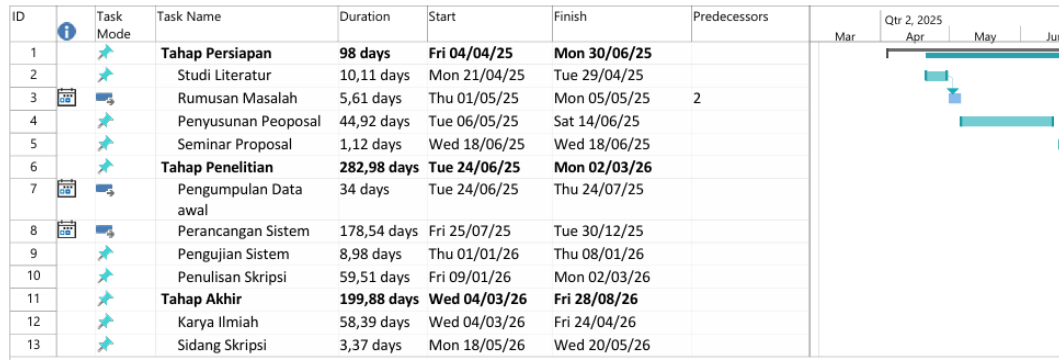
Setiap simpul pada graf mewakili titik lokasi (baik simpang maupun destinasi wisata), dan setiap sisi diberi bobot berdasarkan jarak antar titik. Algoritma Dijkstra digunakan untuk menentukan lintasan dengan jarak paling pendek dari titik awal ke tujuan yang dipilih, tanpa harus menghitung semua kemungkinan jalur yang ada. Hasil perhitungan secara otomatis divisualisasikan pada peta menggunakan garis penghubung berwarna, disertai dengan informasi jarak dan estimasi waktu tempuh.

Pengujian dilakukan melalui beberapa skenario lokasi awal dan tujuan yang berbeda. Hasil yang diperoleh menunjukkan bahwa sistem secara konsisten memberikan rute dengan total jarak paling kecil dibandingkan jalur alternatif lain. Selain itu, urutan simpul yang dilalui oleh algoritma juga sesuai dengan prinsip perhitungan jarak terpendek. Dengan demikian, dapat disimpulkan bahwa implementasi algoritma Dijkstra telah berjalan sesuai harapan dan dapat diterapkan secara efektif dalam sistem rekomendasi rute wisata.

Penggunaan peta interaktif dalam sistem ini juga memberikan kemudahan bagi pengguna dalam menelusuri rute yang tersedia, serta mempermudah pemahaman terhadap lokasi wisata yang ingin dikunjungi. Sistem ini tidak hanya mendukung efisiensi perjalanan, tetapi juga dapat mendorong distribusi kunjungan yang lebih merata ke berbagai destinasi wisata di Banyumas.

D. Jadwal Penelitian

Penelitian akan di mulai pada bulan maret sampai dengan bulan juni. Pembuatan jadwal digunakan untuk mengatur waktu agar penelitian terarah dan terstruktur, Dibawah ini rincian jadwal penelitian sebagai berikut.



Gambar 10 Timeline dan Gantt Chart

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dari pembuatan sistem informasi geografis berbasis web untuk rekomendasi wisata di daerah Banyumas menggunakan algoritma *dijkstra*. Sistem memproses data jaringan jalan dan koordinat lokasi wisata untuk menghitung rute terpendek dari posisi pengguna menuju tujuan yang dipilih. Hasil perhitungan rute kemudian ditampilkan pada peta interaktif, sehingga pengguna dapat melihat jalur tercepat beserta informasi jaraknya.

A. Hasil

1. Pengumpulan Data

Pada tahap ini data diperoleh dari Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Banyumas serta dilengkapi dengan informasi dari beberapa website yang relevan. Data yang dikumpulkan mencakup nama objek wisata, jenis wisata, alamat, dan deskripsi singkat. Sementara itu, titik koordinat setiap lokasi diperoleh melalui Google Maps untuk memastikan keakuratan titik koordinat. Seluruh data tersebut kemudian digunakan sebagai dasar pemetaan rute dan perhitungan jalur terpendek menggunakan algoritma *dijkstra*.

Dalam pengambilan data lokasi wisata, diperoleh daftar destinasi yang tersebar di seluruh wilayah Kabupaten Banyumas. Data tersebut mencakup objek wisata dari puluhan kecamatan dan desa/kelurahan yang menjadi cakupan wilayah pariwisata, sehingga sistem dapat memetakan persebaran lokasi wisata secara lebih menyeluruh.

a. Spesifikasi Data

Beberapa kriteria data yang bisa dimuat ialah sebagai berikut

Tabel 2 Spesifikasi Data

Elemen	Deskripsi yang diambil	Contoh
Sumber data	Instansi dan website yang relevan	Lokawisata Baturraden, - 7.3136309, 109.2291429
Jumlah data	Jumlah baris (jumlah wisata di kabupaten Banyumas)	75
Format data	Format file asli	CSV
Contoh data	1-4 baris pertama	(lihat tabel dibawah)

Tabel 3 Contoh Data yang diambil

No	Nama	Lattitude	Longitude
1	Lokawisata Baturraden	-7.3136309	109.2291429
2	Curug Cipendok	-7.3370898	109.1365451
3	Telaga Sunyi	-7.3072399	109.2431964
4	Baturraden Adventure Forest	-7.3070145	109.2418245

b. Proses Pengolahan Awal

Data dalam penelitian ini diperoleh dari Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Banyumas sebagai sumber utama, serta dilengkapi dengan data dari website wisata yang relevan. Penelitian melakukan proses ekstraksi dan validasi data untuk seluruh objek wisata yang berada di wilayah Kabupaten Banyumas. Parameter data yang digunakan dalam tabel penelitian meliputi nama objek wisata, nilai latitude, dan longitude yang diperoleh melalui Google Maps. Data tersebut digunakan sebagai input dalam sistem web GIS serta sebagai dasar perhitungan rute terpendek menggunakan algoritma Dijkstra.

c. Alasan Memilih Data

Metode pengumpulan data dalam penelitian ini menggunakan data sekunder yang diperoleh dari instansi pemerintah daerah serta website wisata yang relevan. Pemilihan parameter lokasi berupa koordinat latitude dan longitude dilakukan karena keduanya merupakan indikator utama dalam pemetaan spasial objek wisata. Koordinat geografis mencerminkan posisi aktual destinasi wisata di lapangan dan menjadi dasar dalam pembentukan jaringan graf rute terpendek secara objektif dan akurat melalui penerapan algoritma Dijkstra.

2. Implementasi Sistem

Sistem informasi geografis berbasis web ini telah berhasil diimplementasikan menggunakan *framework Laravel* sebagai pengembangan sisi backend dan *Leaflet.js* untuk visualisasi peta interaktif. Sistem ini dirancang untuk membantu pengguna dalam menemukan dan menentukan rute tercepat menuju objek wisata di Kabupaten Banyumas melalui penerapan algoritma Dijkstra

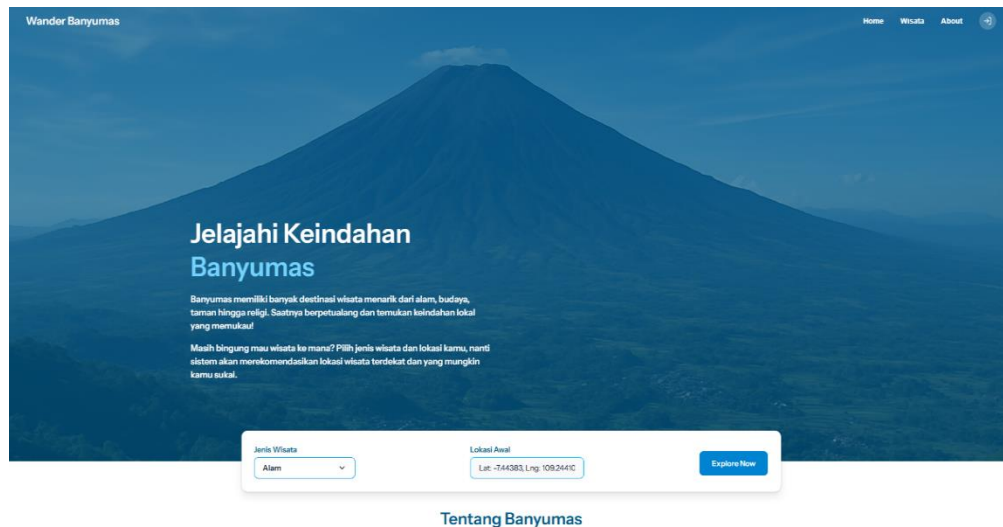
Implementasi sistem mencakup berbagai fitur yang dapat diakses oleh pengguna dan administrator. Pengguna umum dapat melihat peta interaktif, memilih lokasi wisata, serta memperoleh rekomendasi rute terpendek dari posisi awal menuju tujuan. Sementara itu, administrator memiliki akses untuk mengelola data objek wisata, memperbarui informasi lokasi, dan mengatur data koordinat yang digunakan dalam proses perhitungan rute.

Seluruh antarmuka sistem dirancang secara responsif dan informatif guna meningkatkan kemudahan penggunaan serta mendukung pengambilan keputusan perjalanan wisata berbasis data spasial.

Berikut ini halaman yang dibuat di dalam sistem informasi geografis:

a. User

1) Halaman Dashboard User

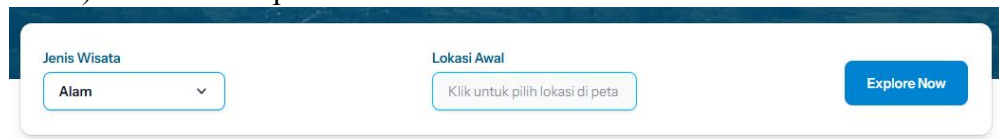


Gambar 11 Halaman *Dashboard User*

Halaman dashboard merupakan tampilan awal yang memberikan informasi mengenai tujuan dan fungsi dari sistem Web GIS. Pada halaman ini dijelaskan bahwa sistem bertujuan untuk memberikan rekomendasi rute wisata di Kabupaten Banyumas dengan memanfaatkan algoritma Dijkstra dalam menentukan jalur terpendek menuju lokasi wisata. Dashboard berfungsi sebagai pengantar agar pengguna memahami konteks dan manfaat sistem sebelum mengakses fitur lainnya. Selain itu, halaman ini dilengkapi dengan fitur *Explore Now* yang membantu pengguna yang belum menentukan tujuan wisata, sehingga sistem dapat memberikan rekomendasi lokasi wisata dan rute optimal secara lebih mudah dan interaktif. Untuk kode halaman *dashboard user* ada di lampiran *sources code 1*

Kode ini menampilkan halaman utama sistem menampilkan gambar latar destinasi wisata banyumas sebagai elemen visual utama pada dashboard. Penggunaan gambar tersebut bertujuan untuk memperkuat identitas pariwisata daerah sekaligus memberikan kesan awal yang informatif dan menarik bagi pengguna. Elemen visual ini mendukung pengalaman pengguna dengan menghadirkan nuansa eksplorasi wisata sejak interaksi, serta memperjelas konteks sistem sebagai media rekomendasi wisata berbasis informasi geografis.

2) Halaman Explore Now



Gambar 12 Halaman Explore Now

Halaman *Explore Now* merupakan fitur utama dalam sistem yang menampilkan visualisasi peta interaktif Kabupaten Banyumas dan berfungsi sebagai titik awal proses rekomendasi wisata. Pada halaman ini, user terlebih dahulu memilih jenis wisata yang diinginkan, seperti wisata alam, budaya dan sejarah, taman dan buatan, atau religi, kemudian menentukan lokasi awal dengan memilih titik koordinat pada peta yang merepresentasikan posisi awal user. Informasi preferensi wisata dan lokasi awal tersebut selanjutnya digunakan sebagai input utama dalam proses perhitungan jarak terpendek menggunakan algoritma Dijkstra untuk menghasilkan rekomendasi wisata terdekat.

Fitur-fitur yang tersedia pada halaman peta meliputi:

- Pemilihan jenis wisata, yang memungkinkan pengguna menentukan kategori destinasi sesuai minat.
- Penentuan lokasi awal, yang dilakukan dengan memilih titik koordinat pada peta sebagai posisi pengguna.

Melalui tahapan ini, sistem dapat memproses data spasial secara akurat sebelum memberikan rekomendasi objek wisata terdekat beserta rute terpendek dari lokasi pengguna.

Berikut potongan kode dari fitur *Explore Now*

```
<div className="absolute bottom-0 left-1/2 z-10 w-full max-w-5xl
-translate-x-1/2 translate-y-1/2 rounded-xl bg-white shadow-lg
ring-1 ring-black/10">
  <div className="flex flex-col justify-between
px-6 py-6 sm:flex-row sm:items-center sm:gap-6">
    { /* Jenis Wisata */ }
    <div className="relative mb-4 flex flex-
col sm:mb-0">
      <label className="mb-1 text-sm font-
extrabold text-sky-800">Jenis Wisata</label>
      <div className="relative">
        <button
          type="button"
          onClick={() => setOpen(!open)}
          className="flex w-48 items-
center justify-between rounded-md border border-sky-500 bg-white
px-4 py-2 text-sm font-semibold text-gray-900 shadow-sm
transition focus:ring-2 focus:ring-sky-600 focus:outline-none"
        >
          <span>{selected}</span>
        </div>
      </div>
    </div>
  </div>
```

```

<ChevronDown className={`h-4
w-4 transition-transform ${open ? 'rotate-180' : ''}`} />
</button>

{open && (
  <div className="absolute z-
20 mt-1 w-48 rounded-md border border-sky-500 bg-white font-black
shadow-lg">
    {[ 'Alam', 'Buatan dan
Taman', 'Budaya dan Sejarah', 'Religi' ].map((opt) => (
      <button
        key={opt}
        onClick={() => {
          setSelected(opt);
          setOpen(false);
        }}
        className={`block
w-full px-4 py-2 text-left text-sm font-medium ${
          selected ===
opt ? 'bg-sky-100 text-sky-800' : 'font-black text-gray-900
hover:bg-sky-50'
        }}`
      >
        {opt}
      </button>
    ))}
  </div>
)}
</div>
</div>

{/* Lokasi Awal */}
<div className="relative mb-4 flex flex-
col sm:mb-0">
  <label className="mb-1 text-sm font-
extrabold text-sky-800">Lokasi Awal</label>
  <input
    type="text"
    placeholder="Klik untuk pilih
lokasi di peta"
    value={locationText}
    readOnly
    onClick={() =>
      setShowMapModal(true)}
    className="cursor-pointer rounded-
md border border-sky-400 bg-gray-50 px-4 py-2 text-sm text-gray-
900 shadow-sm focus:ring-2 focus:ring-sky-600 focus:outline-none"
  />
</div>

{/* Modal Map Fullscreen */}
{showMapModal && (
  <div className="fixed inset-0 z-50
flex items-center justify-center bg-black/60">
    <div className="w-full max-w-5xl
overflow-hidden rounded-lg bg-white shadow-2xl">
      {/* Header Modal */}
      <div className="flex items-
center justify-between bg-white px-6 py-4 text-center">

```

```

<h2 className="text-lg
font-semibold text-sky-600">Pilih Lokasi Anda</h2>
<button onClick={() =>
setShowMapModal(false)} className="text-xl font-bold text-sky-
600 hover:text-sky-800">
X
</button>
</div>

{/* Map Container */}
<div className="relative h-
[80vh] w-full">
<MapContainer
center={[-7.4314,
109.2478]}
zoom={12}
scrollWheelZoom={true}
className="z-0 h-full
w-full border border-black"
>
<FixMapSize />
<TileLayer
attribution='&copy;
<a href="https://osm.org/copyright">OpenStreetMap</a>'
url="https://{s}
.tile.openstreetmap.org/{z}/{x}/{y}.png"
/>
{marker} && <Marker
position={marker} />
<LocationMarker
setMarker={setMarker} setLocationText={setLocationText} />
</MapContainer>

{/* Tombol Konfirmasi
Lokasi */}
<div className="absolute
bottom-4 left-1/2 -translate-x-1/2">
<button
onClick={() =>
setShowMapModal(false)}
className="rounded-
md bg-sky-800 px-6 py-3 text-sm font-semibold text-white shadow-
md hover:bg-sky-700"
>
Konfirmasi Lokasi
</button>
</div>
</div>
</div>
)}

{/* Tombol Explore */}
<div className="flex justify-center
sm:justify-end">
<button
onClick={handleExplore}

```

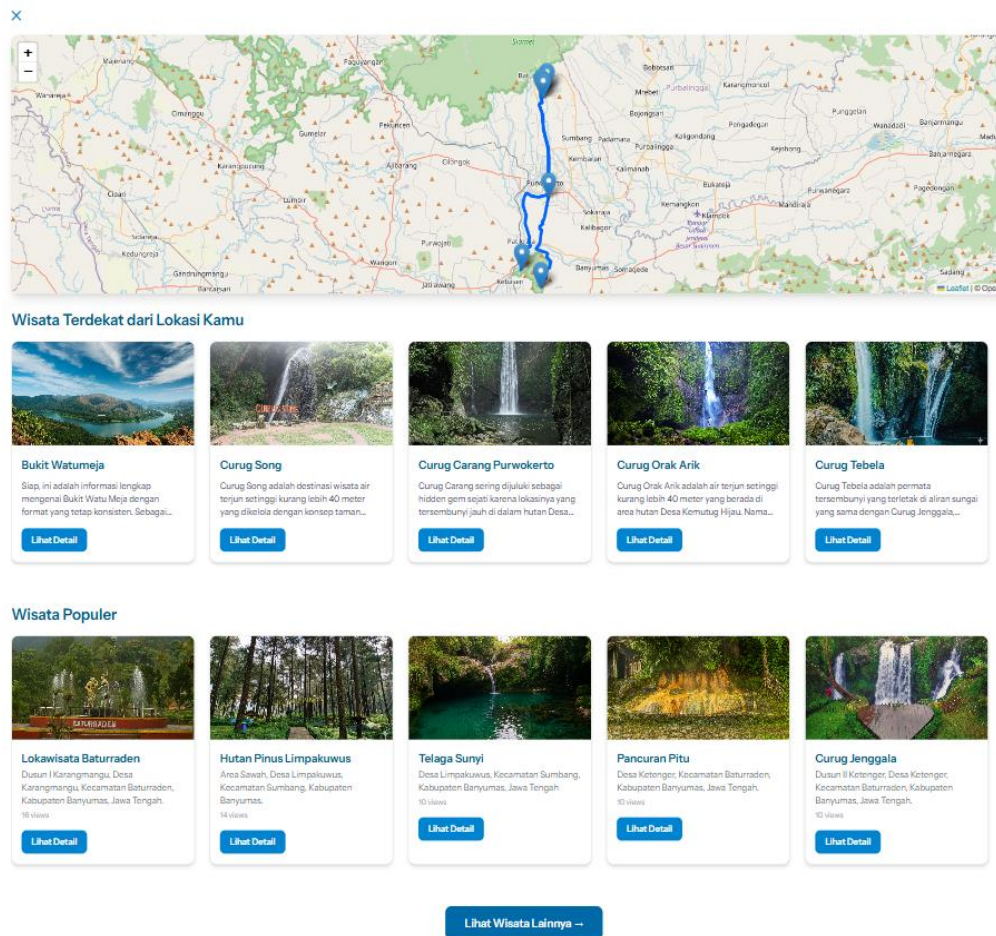
```

                                className="mt-2 rounded-md bg-
sky-600 px-6 py-3 text-sm font-semibold text-white transition
hover:bg-sky-800"
                                >
                                Explore Now
                                </button>
                                </div>
                                </div>
                                </div>

```

Kode pada sistem berfungsi untuk menangani interaksi pengguna dalam memilih jenis wisata, menentukan lokasi awal, serta mengaktifkan fitur *Explore Now*. Pemilihan jenis wisata memungkinkan pengguna menentukan kategori destinasi sesuai preferensi, yang digunakan sebagai salah satu parameter dalam proses rekomendasi. Fitur pemilihan lokasi awal memungkinkan pengguna menentukan titik koordinat posisi secara langsung melalui peta interaktif, yang selanjutnya digunakan sebagai titik awal perhitungan rute. Sementara itu, fitur *Explore Now* dirancang untuk membantu pengguna yang belum menentukan tujuan wisata, dengan memanfaatkan kombinasi jenis wisata yang dipilih dan lokasi awal pengguna untuk merekomendasikan objek wisata terdekat. Seluruh proses ini menjadi tahap awal sebelum sistem menghitung rute terpendek menggunakan algoritma Dijkstra.

3) Halaman rekomendasi wisata



Gambar 13 Halaman Rekomendasi Wisata

Halaman ini menampilkan hasil rekomendasi wisata setelah pengguna memilih jenis wisata dan menentukan lokasi awal melalui fitur *Explore Now*. Sistem menampilkan peta interaktif yang memvisualisasikan lokasi objek wisata beserta rute terpendek dari lokasi pengguna menuju destinasi yang direkomendasikan berdasarkan perhitungan algoritma Dijkstra.

Sebagai pelengkap dari tampilan peta yang bersifat spasial, informasi rekomendasi wisata disajikan dalam bentuk *list card*. Pada bagian *Wisata Terdekat dari Lokasi Pengguna*, setiap kartu menampilkan informasi ringkas mengenai objek wisata yang memiliki jarak terdekat dari posisi awal. Selain itu, sistem juga menyediakan bagian *Rekomendasi Wisata Populer* yang disajikan dalam bentuk *list card* untuk menampilkan destinasi wisata dengan tingkat popularitas tinggi. Penyajian informasi dalam bentuk kartu ini bertujuan untuk memudahkan pengguna dalam membandingkan pilihan wisata secara informatif dan visual sebelum menentukan tujuan perjalanan.

Berikut potongan kode dari Halaman Rekomendasi Wisata:

```
{showRecommendations && (
  <section className="shadow-in bg-white px-6 py-
25 pt-23 pb-2" data-aos="fade-up">
    {/* Header */}
    <div className="mb-4 flex items-center justify-
between">
      <button onClick={() =>
setShowRecommendations(false)} className="text-xl font-bold
text-sky-600 hover:text-sky-800">
        X
      </button>
    </div>
    {/* JIKA DATA KOSONG */}
    {recommendations.length === 0 ? (
      <div className="flex flex-col items-center
justify-center py-20 text-center">
        {/* Icon atau ilustrasi */}
        <svg
          xmlns="http://www.w3.org/2000/svg"
          className="mb-6 h-24 w-24 text-
gray-400"
          fill="none"
          viewBox="0 0 24 24"
          stroke="currentColor"
          strokeWidth={1.5}
        >
          <path
            strokeLinecap="round"
            strokeLinejoin="round"
            d="M3 17a4 4 0 014-4h10a4 4
0 014 4M7 7a5 5 0 010 10" />
        </svg>

        <h3 className="mb-2 text-2xl font-
bold text-gray-700">Wisata Belum Ada</h3>

        <p className="max-w-md text-gray-
500">
          Maaf, data wisata untuk kategori
          ini belum tersedia. Coba pilih jenis wisata lain untuk melihat
          rekomendasi terdekat.
        </p>
      </div>
    ) : (
      <>
        {/* MAP */}
        <div className="mb-6 h-[400px] w-full
overflow-hidden rounded-xl shadow-lg">
          <MapContainer center={[-7.4314,
109.2478]} zoom={12} scrollWheelZoom={true} className="h-full w-
full">
            <TileLayer attribution="&copy;
OpenStreetMap"
              url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png" />
          </MapContainer>
        </div>
      </>
    )
  )
}
```

```

position={marker}}></Marker>}}
{marker && <Marker
position={marker}}></Marker>}}
{recommendations.map((rec) =>
(
<Marker key={rec.id}
position={[rec.latitude, rec.longitude]}>
<Popup>
<strong>{rec.nam
a}</strong>
<br />
{rec.deskripsi}
</Popup>
</Marker>
)}}
recommendations.map((rec) => <RouteLine start={marker}
end={[rec.latitude, rec.longitude]} />)}
</MapContainer>
</div>
{ /* CARD */}
<div className="relative">
<h2 className="mb-4 text-2xl font-
extrabold text-sky-800">Wisata Terdekat dari Lokasi Kamu</h2>
<div className="scrollbar-hide
flex gap-6 overflow-x-auto pb-4">
{recommendations.map((rec) =>
(
<div
key={rec.id}
className="max-w-
[300px] min-w-[300px] flex-shrink-0 overflow-hidden rounded-xl
border border-gray-200 bg-white shadow-md transition
hover:shadow-lg"
>
{ /* Gambar */}
<img src={rec.gambar}
alt={rec.nama} className="h-40 w-full object-cover" />
{ /* Isi Card */}
<div className="p-4">
<h3 className="text-
lg font-bold text-sky-800">{rec.nama}</h3>
<p className="mt-
2 line-clamp-3 text-sm text-gray-700">{rec.deskripsi}</p>
{ /* Tombol Detail
*/}
<a
href={`/${wisa
ta/${rec.id}}`}
className="mt-
4 inline-block rounded-md bg-sky-600 px-4 py-2 text-sm font-
semibold text-white hover:bg-sky-800"
>
Lihat Detail
</a>

```

```

        </div>
    </div>
    )))
</div>
</div>
</>
))
{/* WISATA POPULER */}
<section className="mt-12">
    <div className="relative">
        <h2 className="mb-4 text-2xl font-
extrabold text-sky-800">Wisata Populer</h2>

        <div className="scrollbar-hide flex
gap-6 overflow-x-auto pb-4">
            {wisataPopuler.map((item) => (
                <div
                    key={item.id}
                    className="max-w-[300px]
min-w-[300px] flex-shrink-0 overflow-hidden rounded-xl border
border-gray-200 bg-white shadow-md transition hover:shadow-lg"
                >
                    {/* Gambar */}
                    <img
                        src={item.gambar ?
`/storage/${item.gambar}` : '/images/default.jpg'}
                        alt={item.nama}
                        className="h-40 w-
full object-cover"
                    />

                    {/* Isi Card */}
                    <div className="p-4">
                        <h3 className="text-
lg font-bold text-sky-800">{item.nama}</h3>

                        <p className="text-
sm text-gray-500">{item.lokasi}</p>

                        <p className="mt-1
text-xs text-gray-400">{item.views} views</p>

                        {/* Tombol Detail */}
                        <a
                            href={`/${wisata/$
{item.id}`}
                            className="mt-4
inline-block rounded-md bg-sky-600 px-4 py-2 text-sm font-
semibold text-white hover:bg-sky-800"
                        >
                            Lihat Detail
                        </a>
                    </div>
                </div>
            ))
        </div>
    </div>
</section>

```

```

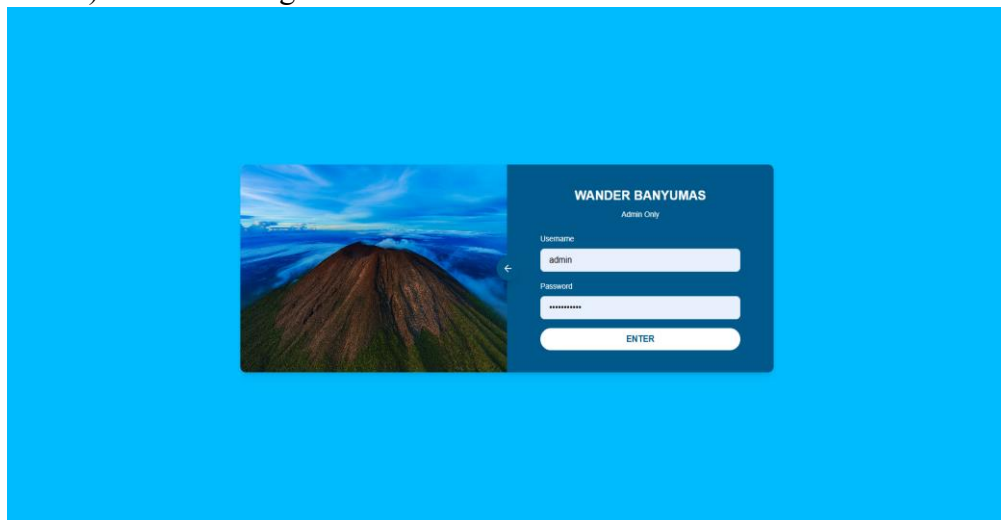
<div className="mt-12 text-center">
  <a
    href={routeByJenis(selected)}
    className="inline-block rounded-lg
bg-sky-700 px-8 py-3 text-lg font-bold text-white hover:bg-sky-
800"
    >
    Lihat Wisata Lainnya →
  </a>
</div>
</section>
)}

```

Potongan kode tersebut digunakan untuk menampilkan data rekomendasi objek wisata dalam bentuk *list card* yang bersifat interaktif. Setiap kartu menyajikan informasi ringkas mengenai objek wisata, seperti nama lokasi dan jarak dari posisi pengguna. Data disusun berdasarkan hasil perhitungan jarak terdekat, sehingga memudahkan pengguna dalam membandingkan pilihan wisata. Penyajian dalam bentuk *list card* ini juga mendukung pengalaman pengguna yang lebih informatif dan intuitif sebagai pelengkap visualisasi peta.

b. Admin

1) Halaman Login

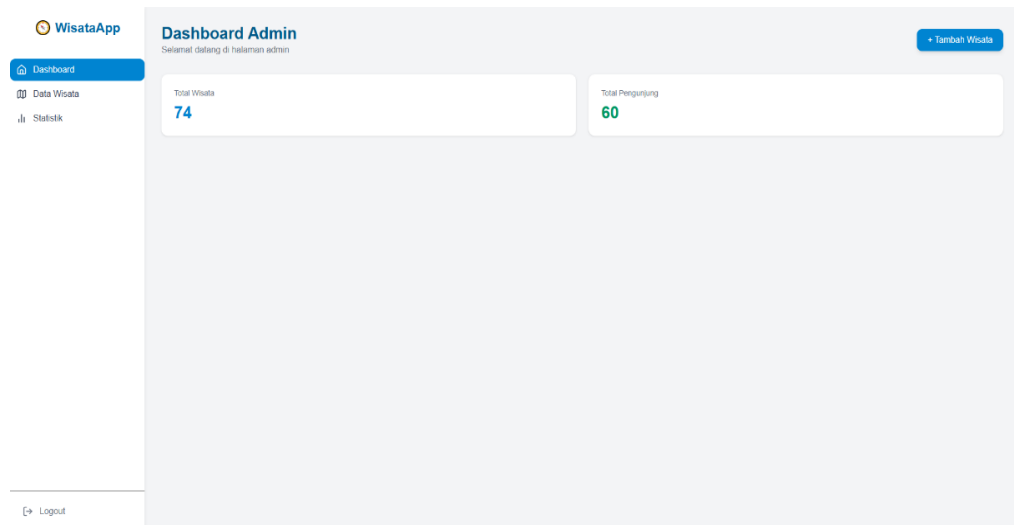


Gambar 14 Halaman Login

Pengguna umum dapat mengakses seluruh fitur utama sistem tanpa melalui proses autentikasi. Sistem hanya menyediakan halaman login khusus bagi administrator yang digunakan untuk mengelola data objek wisata dan informasi pendukung lainnya. Form login admin terdiri dari kolom username dan password yang berfungsi sebagai kredensial akses ke sistem. Setelah proses autentikasi berhasil, administrator akan diarahkan ke halaman dashboard admin untuk melakukan pengelolaan data pada sistem. Untuk kode halaman *login* ada di lampiran *sources code 2*

Potongan kode tersebut berfungsi untuk menangani proses autentikasi administrator pada sistem. Sistem melakukan pemeriksaan kecocokan antara username dan password yang dimasukkan dengan data yang tersimpan. Apabila proses autentikasi gagal, sistem akan menampilkan pesan kesalahan kepada administrator. Sebaliknya, jika autentikasi berhasil, administrator akan diarahkan ke halaman dashboard pengelolaan data pada sistem Web GIS.

2) Halaman Dashboard Admin

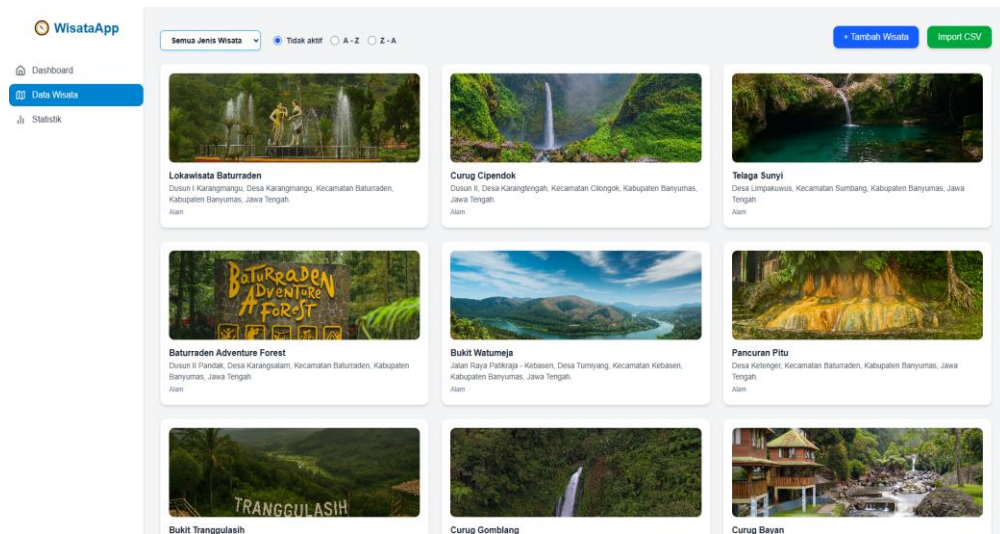


Gambar 15 Halaman Dashboard Admin

Setelah berhasil melakukan login, administrator akan diarahkan ke halaman Dashboard Admin yang berfungsi sebagai pusat kontrol dalam pengelolaan data sistem. Pada halaman ini ditampilkan informasi ringkas berupa jumlah total objek wisata dan jumlah pengunjung sebagai gambaran umum kondisi data yang tersimpan dalam sistem. Dashboard juga menyediakan menu navigasi untuk mengakses fitur pengelolaan data wisata dan statistik, serta tombol *Tambah Wisata* yang digunakan untuk menambahkan data objek wisata baru. Dengan adanya dashboard ini, administrator dapat memantau dan mengelola data wisata secara terstruktur dan efisien sebelum digunakan dalam proses rekomendasi rute wisata. Untuk kode halaman *Dashboard Admin* ada di lampiran *sources code 3*

Potongan kode tersebut digunakan untuk menampilkan halaman dashboard admin pada sistem. Halaman ini menyajikan informasi ringkas berupa total objek wisata dan total pengunjung sebagai gambaran umum kondisi data yang tersimpan. Selain itu, dashboard juga menyediakan tombol *Tambah Wisata* yang digunakan oleh administrator untuk menambahkan data objek wisata baru ke dalam sistem. Tampilan ini berfungsi sebagai pusat kontrol awal bagi admin sebelum mengakses fitur pengelolaan data lainnya.

3) Halaman Data Wisata



Gambar 16 Halaman Data Wisata

Halaman Data Wisata merupakan bagian dari sistem yang digunakan oleh administrator untuk mengelola seluruh data objek wisata yang tersimpan. Pada halaman ini, data wisata disajikan dalam bentuk kartu (card) yang menampilkan informasi utama setiap objek wisata, meliputi nama wisata, lokasi administratif, serta kategori jenis wisata. Penyajian dalam bentuk visual ini bertujuan untuk memudahkan admin dalam melakukan identifikasi dan pengelolaan data secara cepat dan intuitif.

Selain itu, halaman ini dilengkapi dengan fitur filter jenis wisata, yang memungkinkan admin menampilkan data berdasarkan kategori tertentu, serta fitur pengurutan data (A–Z dan Z–A) untuk membantu pencarian objek wisata secara alfabetis. Sistem juga menyediakan tombol Tambah Wisata untuk menambahkan data objek wisata baru secara manual, serta fitur Import CSV yang digunakan untuk memasukkan data wisata dalam jumlah besar secara efisien.

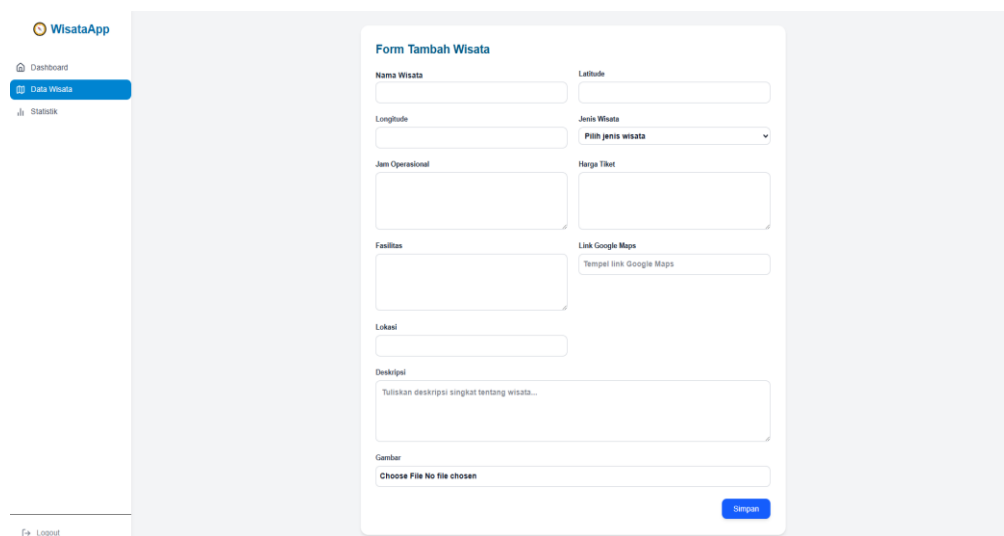
Melalui halaman ini, administrator dapat mengelola data wisata secara terpusat, terstruktur, dan mudah diperbarui, sehingga mendukung keakuratan informasi yang akan ditampilkan kepada pengguna umum pada sistem rekomendasi wisata berbasis WebGIS. Untuk kode halaman data wisata ada di lampiran *sources code 4*

Potongan kode tersebut berfungsi untuk menampilkan halaman Data Wisata pada sisi administrator. Halaman ini menyajikan daftar objek wisata dalam bentuk list card, di mana setiap kartu memuat informasi singkat mengenai destinasi wisata seperti nama, lokasi, dan kategori wisata.

Pada halaman ini disediakan beberapa fitur pendukung pengelolaan data, antara lain tombol tambah wisata yang digunakan untuk memasukkan data objek wisata baru, serta tombol import CSV yang memungkinkan admin melakukan penambahan data secara massal. Selain itu, sistem juga menyediakan tombol edit wisata yang akan muncul ketika kursor diarahkan ke salah satu kartu wisata, sehingga admin dapat melakukan pembaruan data dengan lebih cepat dan efisien.

Dengan tampilan berbasis list card dan dukungan fitur manajemen data tersebut, halaman Data Wisata berperan sebagai pusat pengelolaan informasi destinasi wisata yang digunakan oleh sistem rekomendasi wisata berbasis WebGIS.

4) Halaman Tambah Data Wisata



Gambar 17 Halaman Tambah Data Wisata

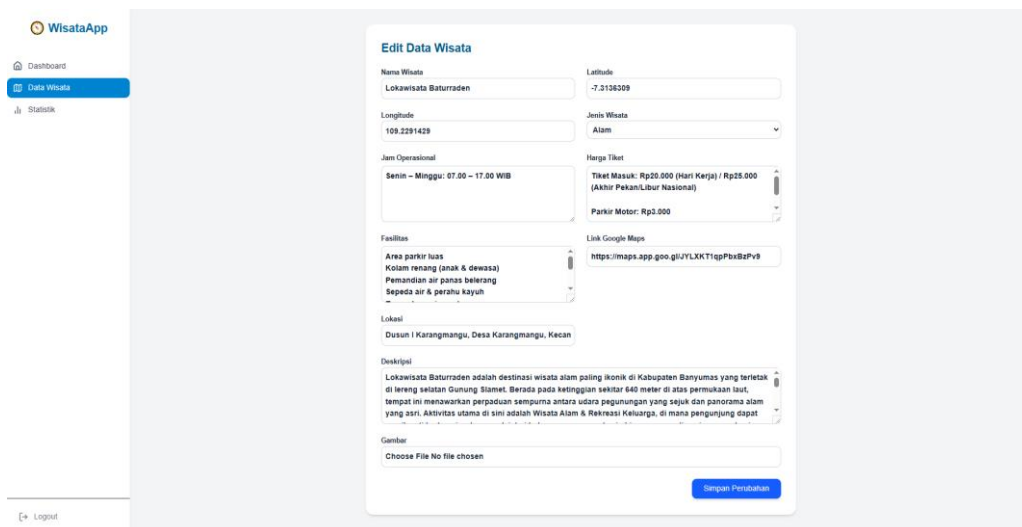
Halaman Tambah Data Wisata digunakan oleh administrator untuk melakukan input data objek wisata baru ke dalam sistem. Pada halaman ini disediakan sebuah formulir yang memuat beberapa field isian, antara lain nama wisata, koordinat geografis (latitude dan longitude), jenis wisata, jam operasional, harga tiket, fasilitas, lokasi, deskripsi singkat, serta unggahan gambar wisata. Selain itu, tersedia kolom tautan Google Maps yang digunakan untuk mengarahkan pengguna ke aplikasi atau situs Google Maps apabila pengguna ingin melihat lokasi objek wisata tersebut secara langsung di Google Maps.

Data yang diinputkan melalui formulir ini akan diproses oleh sistem dan disimpan ke dalam database, sehingga dapat digunakan dalam proses visualisasi peta, rekomendasi destinasi wisata, serta perhitungan jarak terdekat menggunakan algoritma Dijkstra. Halaman ini berperan penting dalam mendukung kelengkapan informasi wisata dan kemudahan akses lokasi bagi

pengguna. Untuk kode halaman tambah data wisata ada di lampiran *sources code 5*

Potongan kode tersebut merupakan formulir input data wisata yang digunakan oleh administrator untuk menambahkan data objek wisata baru ke dalam sistem. Data yang dimasukkan melalui formulir ini akan diproses oleh sistem dan disimpan ke dalam database, sehingga dapat digunakan pada fitur visualisasi peta, halaman detail wisata, serta proses rekomendasi dan penentuan jarak terdekat menggunakan algoritma Dijkstra.

5) Halaman Edit Data Wisata



Gambar 18 Halaman Edit Data Wisata

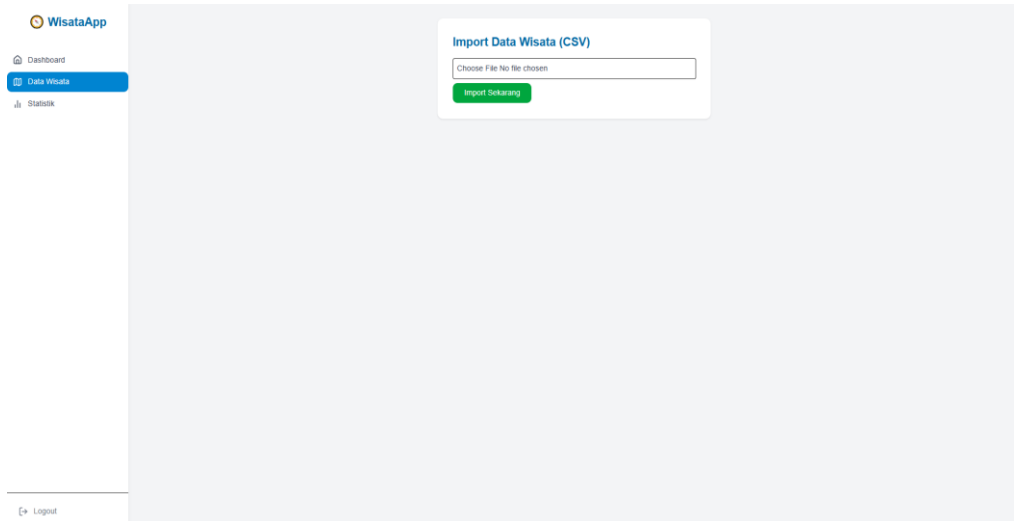
Halaman Edit Data Wisata digunakan oleh administrator untuk melakukan pembaruan terhadap data objek wisata yang telah tersimpan di dalam sistem. Struktur dan kolom isian pada halaman ini sama dengan halaman tambah data wisata, meliputi nama wisata, koordinat geografis (latitude dan longitude), jenis wisata, jam operasional, harga tiket, fasilitas, lokasi, deskripsi singkat, unggahan gambar wisata, serta tautan Google Maps.

Data yang ditampilkan pada formulir edit merupakan data yang telah tersimpan sebelumnya, sehingga administrator dapat melakukan perubahan atau penyesuaian informasi sesuai kebutuhan. Setelah proses pengeditan dilakukan, data yang diperbarui akan diproses oleh sistem dan disimpan kembali ke dalam database, sehingga informasi wisata yang ditampilkan kepada pengguna tetap akurat dan terkini. Untuk kode halaman edit data wisata ada di lampiran *sources code 6*

Potongan kode tersebut merupakan formulir edit data wisata yang digunakan oleh administrator untuk memperbarui informasi objek wisata yang telah tersimpan di dalam sistem. Data yang diubah melalui formulir ini akan diproses oleh sistem dan disimpan kembali ke dalam database, sehingga pembaruan informasi dapat ditampilkan pada fitur visualisasi peta, halaman

detail wisata, serta digunakan dalam proses rekomendasi dan penentuan jarak terdekat menggunakan algoritma Dijkstra.

6) Halaman Import CSV



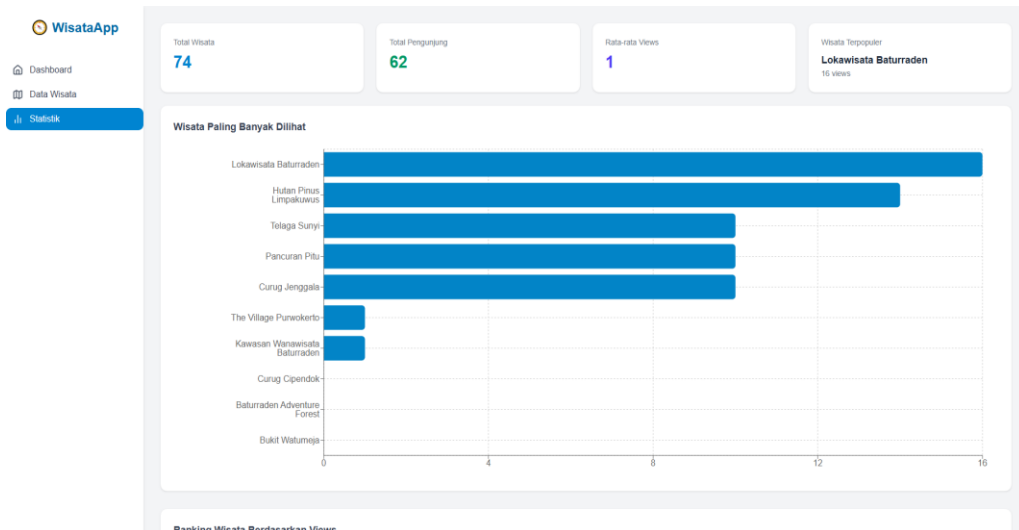
Gambar 19 Halaman Import CSV

Halaman Import Data Wisata (CSV) digunakan oleh administrator untuk melakukan penambahan data objek wisata secara massal ke dalam sistem. Pada halaman ini, admin dapat mengunggah berkas berformat CSV yang berisi data wisata melalui fitur unggah file yang telah disediakan. Setelah file dipilih, proses impor dapat dijalankan dengan menekan tombol *Import Sekarang*.

Data yang berhasil diimpor akan diproses oleh sistem dan disimpan ke dalam database, sehingga dapat langsung digunakan pada fitur visualisasi peta, halaman detail wisata, serta mendukung proses rekomendasi dan perhitungan jarak terdekat menggunakan algoritma Dijkstra. Fitur ini bertujuan untuk meningkatkan efisiensi pengelolaan data wisata, terutama ketika jumlah data yang dimasukkan cukup besar. Untuk kode halaman *import* CSV ada di lampiran *sources code 7*

Potongan kode tersebut digunakan untuk menangani proses import data wisata melalui file CSV oleh administrator. Data yang diimpor akan disimpan ke dalam database dan selanjutnya dimanfaatkan dalam penyajian informasi wisata, visualisasi peta, serta proses rekomendasi destinasi menggunakan algoritma Dijkstra.

7) Halaman Statistik



Gambar 20 Halaman Statistik

Halaman Statistik digunakan untuk menampilkan ringkasan dan analisis data kunjungan objek wisata yang terdapat dalam sistem. Pada bagian atas halaman ditampilkan informasi statistik utama, meliputi total data wisata, total pengunjung, rata-rata jumlah tampilan (views), serta wisata terpopuler berdasarkan jumlah kunjungan tertinggi.

Selain itu, halaman ini juga menyajikan visualisasi data dalam bentuk grafik batang yang menunjukkan daftar objek wisata yang paling banyak dilihat oleh pengguna. Grafik tersebut memudahkan administrator dalam membandingkan tingkat popularitas masing-masing destinasi wisata secara visual. Dengan adanya halaman statistik ini, admin dapat memantau performa dan minat pengguna terhadap objek wisata yang tersedia sebagai bahan evaluasi dan pengambilan keputusan pengelolaan data wisata. Untuk kode halaman *import* CSV ada di lampiran *sources code* 8.

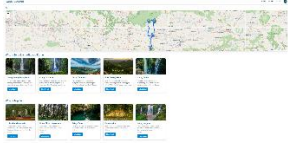
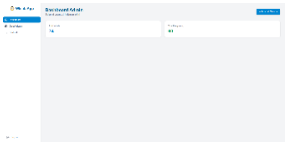
Potongan kode tersebut digunakan untuk menampilkan data statistik kunjungan wisata dalam bentuk ringkasan dan visualisasi grafik. Sistem mengolah data jumlah kunjungan dari setiap objek wisata untuk ditampilkan sebagai grafik batang, sehingga administrator dapat melihat perbandingan tingkat popularitas wisata secara visual. Informasi statistik ini membantu dalam pemantauan performa destinasi wisata yang tersimpan di dalam sistem.

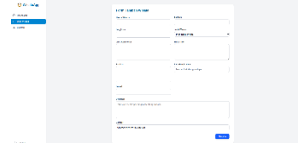
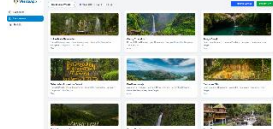
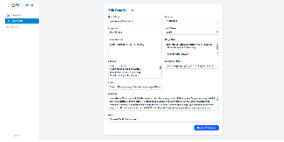
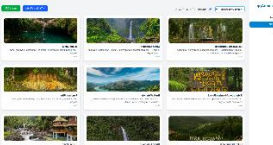
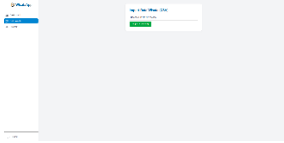
3. Pengujian Sistem

Berdasarkan hasil perancangan dan implementasi sistem, telah berhasil dibangun sebuah aplikasi sistem informasi geografis berbasis web untuk rekomendasi wisata di Kabupaten Banyumas menggunakan *React.js* sebagai

pengembangan antarmuka dan *Leaflet.js* untuk visualisasi peta interaktif. Sistem ini menerapkan algoritma djikstra untuk menghitung jarak terpendek dari lokasi awal pengguna menuju lokasi objek wisata sehingga memberikan rekomendasi destinasi wisata berdasarkan parameter jarak. Untuk memastikan sistem berjalan sesuai dengan fungsi yang dirancang, dilakukan pengujian menggunakan metode *Black Box Testing* terhadap fitur-fitur utama, meliputi pemilihan jenis wisata, penentuan lokasi awal pengguna, perhitungan jarak, visualisasi rute pada peta, serta penampilan rekomendasi wisata terdekat, dan hasil pengujian menunjukkan bahwa seluruh fitur sistem dapat berfungsi dengan baik sesuai dengan kebutuhan pengguna.

Tabel 4 Uji Sistem Menggunakan *Black Box Testing*

No	Skenario	Test Case	Hipotesis	Valid
1.	Halaman Dashboard User	Akses halaman <i>Dashboard User</i> 	Informasi pnegantar WEB GIS 	Valid
2.	Fitur Explore Now	Masukkan jenis wisata dan titik koordinat awal user 	Kolom lokasi awal akan memunculkan peta 	Valid
3.	Halaman Rekomendasi Wisata	Hasil fitur explore now 	Mucnul beberapa rekomendasi wisata dan rute map 	Valid
4.	Halaman Login Admin	Masukkan username dan password 	Admin dapat masuk ke dashboard admin apabila benar 	Valid

5.	Halaman Tambah Data Wisata	Masukkan nama wisata, latitude, longitude, jenis wisata, jam operasional, harga tiket, fasilitas, link gmaps, lokasi, deskripsi, gambar 	Data akan tersimpan dan muncul di halaman data wisata 	Valid.
6.	Halaman Edit Data Wisata	Pilih data wisata yang ingin diedit lalu pilih kolom yang mau di edit 	Data akan diperbarui dengan kolom yang diedit 	Valid.
7.	Halaman Import CSV	Masukkan file csv yang berisi data wisata 	Data akan tersimpan dan muncul di halaman data wisata 	Valid.
8.	Halaman Statistik	Berisi informasi tentang total wisata, total pengunjung, rata-rata views, wisata populer, dan bar views 	Memperlihatkan wisata mana yang sering user kunjungi 	Valid.

B. Pembahasan

Hal ini menunjukkan bahwa sistem informasi geografis berbasis web yang telah dirancang dan diimplementasikan mampu berjalan dengan baik secara fungsional, baik dari sisi pengguna (user) maupun administrator (admin), serta

mendukung penyajian dan analisis informasi lokasi wisata secara spasial di Kabupaten Banyumas. Sistem ini tidak hanya menyajikan informasi wisata, tetapi juga memberikan rekomendasi destinasi berdasarkan kebutuhan dan lokasi pengguna.

Penerapan algoritma Dijkstra dalam sistem ini digunakan untuk menentukan objek wisata dengan jarak terpendek dari posisi pengguna. Implementasi ini dilakukan dengan tahapan sebagai berikut:

1. Penentuan Bobot Sisi Dalam graf yang dibangun, setiap sisi yang menghubungkan pengguna ke lokasi wisata harus memiliki bobot. Kode di bawah ini berfungsi untuk menghitung bobot jarak tersebut berdasarkan titik koordinat:

```
function getDistance(lat1: number, lon1: number, lat2: number, lon2: number) {  
  
  const R = 6371; // km  
  
  const dLat = (lat2 - lat1) * (Math.PI / 180);  
  
  const dLon = (lon2 - lon1) * (Math.PI / 180);  
  
  const a =  
  
    Math.sin(dLat / 2) * Math.sin(dLat / 2) +  
  
    Math.cos(lat1 * (Math.PI / 180)) * Math.cos(lat2 * (Math.PI / 180)) * Math.sin(dLon / 2) * Math.sin(dLon / 2);  
  
  const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));  
  
  return R * c; // jarak dalam km  
  
}
```

2. Algoritma Dijkstra diterapkan ke dalam kode program untuk menghasilkan rute terpendek bagi wisatawan. Proses eksekusi mengikuti tahapan standar algoritma Dijkstra sebagai berikut:

- a) Inisialisasi dan Penandaan (*Marking*): Sistem memulai dengan mengambil data koordinat pengguna (sebagai titik awal) dan memetakan seluruh data objek wisata dari database. Setiap data wisata diberikan properti tambahan untuk menampung nilai jarak yang akan dihitung.

```
// Kode untuk Inisialisasi: Memetakan data dari database  
(mapped)  
  
// dan mempersiapkan variabel untuk menampung jarak  
(withDistance)
```

```

const withDistance = mapped.map((rec) => ({
    ...rec,

    // Status awal: data siap diproses untuk evaluasi

}));

```

- b) *Distance Evaluation*: pada tahap ini, sistem menghitung “bobot” atau jarak antara lokasi pengguna (node saat ini) ke setiap objek wisata (node tetangga) menggunakan fungsi `getDistance`. Ini adalah inti dari pencarian jarak dalam graf

```

// Kode untuk Evaluasi Distance: Menghitung bobot/jarak dari
user

// ke masing-masing objek wisata menggunakan koordinat Lat
& Lng

distance:    getDistance(marker[0],    marker[1],    rec.lat,
rec.lng),

```

- c) Pilih selanjutnya dan ulangi *Next & Repeat*: setelah semua jarak dihitung, sistem memilih lokasi dengan akumulasi jarak terkecil (paling optimal). Proses ini dilakukan secara berulang (iterasi) terhadap seluruh data hingga didapatkan urutan lokasi dari yang terdekat hingga terjauh.

```

// Kode untuk Next (Pilih Selanjutnya): Mengurutkan
berdasarkan bobot terkecil

const sorted = withDistance.sort((a, b) => a.distance -
b.distance);

// Kode untuk Repeat: Mengambil 5 hasil terbaik dari seluruh
iterasi data

const nearestFive = sorted.slice(0, 5);

```

Melalui potongan kode di atas, sistem secara otomatis menjalankan tahapan Dijkstra. Fungsi `.map` melakukan inisialisasi dan *distance evaluation* secara stimulant, fungsi `getDistance` menentukan bobot jarak, dan fungsi `.sort` dan `.slice` menjalankan tahapan pilih selanjutnya untuk menentukan hasil akhir yang paling efisien bagi pengguna.

Dalam penelitian ini, parameter utama yang digunakan sebagai dasar rekomendasi adalah jarak, karena jarak merupakan faktor penting dalam pengambilan keputusan wisata, khususnya terkait efisiensi waktu dan kemudahan akses. Setelah pengguna menentukan jenis wisata yang diinginkan dan memilih lokasi awal, sistem akan menghitung jarak menuju setiap destinasi wisata yang tersedia, kemudian mengurutkannya berdasarkan jarak terdekat.

Hasil perhitungan menggunakan algoritma Dijkstra menghasilkan daftar destinasi wisata yang direkomendasikan berdasarkan kedekatan lokasi dengan pengguna. Destinasi dengan jarak paling dekat akan diprioritaskan sebagai rekomendasi utama. Selain itu, sistem juga menampilkan jalur perjalanan pada peta interaktif untuk memudahkan pengguna dalam memahami rute yang dapat ditempuh menuju lokasi wisata.

Penerapan algoritma Dijkstra dalam sistem ini memberikan hasil yang akurat dalam menentukan jarak terpendek, sehingga rekomendasi yang dihasilkan bersifat relevan dan mudah dipahami oleh pengguna. Integrasi hasil perhitungan jarak dengan visualisasi peta dalam sistem WebGIS membantu meningkatkan efektivitas penyampaian informasi spasial serta mendukung pengambilan keputusan wisata secara lebih tepat.

Dengan demikian, sistem WebGIS rekomendasi wisata yang dikembangkan mampu memberikan pemetaan dan rekomendasi destinasi wisata di Kabupaten Banyumas secara lebih terstruktur, informatif, dan berbasis spasial, serta mendukung kebutuhan pengguna dalam merencanakan kunjungan wisata secara efisien.

BAB V KESIMPULAN

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan sebagai berikut:

1. Penerapan algoritma Dijkstra pada rancang bangun SIG telah berhasil diimplementasikan ke dalam sistem WebGIS untuk memberikan rekomendasi rute wisata optimal. Algoritma ini bekerja dengan memproses data koordinat geografis objek wisata sebagai titik (*node*) dan menghitung jarak antar titik untuk menentukan jalur dengan akumulasi jarak terpendek. Sistem ini berhasil memvisualisasikan rute tersebut secara interaktif kepada pengguna melalui peta berbasis Leaflet.js.
2. Untuk mengukur efisiensi sistem yang dibangun, dilakukan pengujian dengan mengambil sampel 10 destinasi wisata di Kabupaten Banyumas. Pengujian ini membandingkan antara rute yang biasa digunakan secara umum (jalur utama/konvensional) dengan rute hasil optimasi algoritma Dijkstra yang diimplementasikan pada sistem WebGIS.

Tabel 5 Data Hasil Pengujian Perbandingan Jarak Rute

No	Titik Awal	Destinasi Wisata	Rute Dijkstra (Km)	Rute Konvensional (Km)	Selisih/Efisiensi (Km)
1.	Alun-Alun Purwokerto	Curug Tebela	12	13.8	1.8
2.	Alun-Alun Purwokerto	Curug Orak Arik	10.8	12.5	1.7
3.	Alun-Alun Purwokerto	Curug Carang	11.3	13	1.7
4.	Alun-Alun Purwokerto	Bukit Tranggulasih	14.2	16	1.8
5.	Alun-Alun Purwokerto	Bukit Watumeja	12.5	16.9	4.4
6.	Alun-Alun Purwokerto	Museum Jendral Soedirman	4.8	7.8	3
7.	Alun-Alun Purwokerto	Makam R Djoko K	14.9	17.8	2.9
8.	Alun-Alun Purwokerto	Museum Wayang	17.8	20.7	2,9

9.	Alun-Alun Purwokerto	Masjid Saka Tunggal	28.3	29.8	1.5
10.	Alun-Alun Purwokerto	Goa Maria Kaliori	14.2	18.3	4.1
Jarak Rata-Rata			14.08 Km	16.66 Km	2.58 Km

Analisis indikator efisiensi: berdasarkan tabel di atas, diperoleh indikator efisiensi sebagai berikut:

1. Selisih Jarak: Penerapan algoritma Dijkstra memberikan rute yang lebih pendek dengan rata-rata selisih 2.58 Km per perjalanan.
2. Presentasi Efektivitas: secara perhitungan, rute yang dihasilkan sistem memiliki tingkat efisiensi sebesar 15.49% lebih baik dibandingkan jalur konvensional.
3. Optimasi Waktu: dengan berkurangnya jarak tempuh secara otomatis, maka waktu perjalanan yang diperlukan wisatawan menjadi lebih singkat dan meminimalkan risiko kesalahan pengambilan jalur di persimpangan jalan.

Perbandingan hasil perhitungan rute: Hasil perhitungan rute terpendek ini diperoleh karena algoritma Dijkstra bekerja dengan memeriksa seluruh titik (*node*) persimpangan yang tersedia dalam database spasial dan memilih bobot terkecil (jarak terpendek) secara akumulatif. Sistem tidak hanya mengandalkan jalan protokol, tetapi juga mempertimbangkan jalan alternatif yang secara matematis lebih dekat menuju lokasi tujuan. Hal ini membuktikan bahwa sistem tidak hanya berfungsi secara teknis, tetapi memberikan nilai guna praktis bagi wisatawan dalam menghemat waktu dan bahan bakar.

B. Saran/Rekomendasi

Untuk pengembangan lebih lanjut, sistem informasi geografis berbasis web yang telah dibangun dapat ditingkatkan dengan menambahkan parameter lain dalam proses rekomendasi wisata, seperti estimasi waktu tempuh, kondisi lalu lintas, tingkat kepadatan pengunjung, atau preferensi pengguna, sehingga hasil rekomendasi yang dihasilkan menjadi lebih personal dan akurat. Selain itu, sistem juga dapat dikembangkan dengan integrasi data real-time serta pemanfaatan algoritma pencarian rute yang lebih kompleks atau berbasis pembobotan multi-kriteria guna meningkatkan kualitas penentuan rute dan destinasi wisata.

Pengembangan pada aspek antarmuka juga dapat dilakukan dengan menambahkan visualisasi rute yang lebih interaktif, statistik kunjungan wisata secara real-time, serta fitur ulasan atau penilaian pengguna terhadap objek wisata. Dengan pengembangan tersebut, sistem diharapkan mampu menyajikan informasi yang lebih informatif, adaptif, dan responsif terhadap

kebutuhan pengguna, serta memberikan kontribusi yang lebih optimal dalam pengelolaan dan promosi pariwisata di Kabupaten Banyumas.

DAFTAR PUSTAKA

- [1] P. Nanda B, M. Ghofar Rohman, and M. Sholihin, "Implementasi Algoritma Dijkstra Pada Aplikasi Sistem Informasi Geografis Pencarian Rute Terpendek Wisata Dikabupaten Lamongan Berbasis Web," *Joutica J. Inform. Unisla*, vol. 7, no. 1, pp. 546–550, 2022.
- [2] P. Application, "Aplikasi Penyedia Informasi Destinasi Wisata Kabupaten," vol. 2, no. 2, 2024.
- [3] F. Apriyad, M. I. Askar, and M. R. H, "Pemetaan Interaktif Destinasi Wisata Jawa Timur Menggunakan WebGIS Berbasis React dan Leaflet . js," vol. 1, no. 2, pp. 59–64, 2025.
- [4] E. Putri and H. Februariyanti, "Sistem Rekomendasi Tempat Wisata Kota Padang Dengan Haversine," *Proceeding SENDIU (Seminar Nas. Multi Disiplin Ilmu) 2020 Univ. Stikubank Semarang*, pp. 106–114, 2020, [Online]. Available: <https://www.unisbank.ac.id/ojs/index.php/sendiu/article/view/7968>
- [5] T. Dari, P. Kota, S. Ke, and T. Bersejarah, "Penerapan algoritma dijkstra untuk menentukan rute terpendek dari pusat kota surabaya ke tempat bersejarah," vol. 4, no. 1, pp. 213–223, 2022.
- [6] T. W. Religi, "Universitas Jember, Jl. Kalimantan 37, Jember 68121, Indonesia Email:," vol. 7, no. 2, pp. 162–170, 2023, doi: 10.32505/qalasadi.v7i2.6227.
- [7] S. Olpah, K. Kunci, R. Terpendek, A. Dijkstra, and A. Star, "Literatur Review : Perbandingan Perbandingan Algoritma Dijkstra dan A Star dalam Pencarian Rute Terdekat," no. May, 2021.
- [8] G. Stevens, A. Eleazar Rangin, J. Timothy Souk, J. Pratama, and V. Handrianus Pranatawijaya, "Implementasi Algoritma Dijkstra Dalam Menentukan Rute Wisata Di Kota Palangka Raya," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3518–3523, 2024, doi: 10.36040/jati.v8i3.9701.
- [9] Herman, "Perancangan Sistem Informasi Geografis Pencarian Pertamina Dan Bengkel Service Dengan Teknologi Gps Dan Algoritma Dijkstra," *J. TIMES*, vol. 11, no. 2, pp. 9–16, 2022, doi: 10.51351/jtm.11.2.2022668.
- [10] A. Al Ambiyah, L. Ode, P. Aqsan, and J. Nangi, "Dan Analisis Spasial Fasilitas Umum Di Kota Kendari Menggunakan Leaflet . js," vol. 9, no. 5, pp. 7682–7689, 2025.
- [11] E. Okyusmariantio, E. Sadjati, H. Ikhsani, and U. L. Kuning, "Pembangunan Sistem Informasi WebGIS untuk Evaluasi Rehabilitasi Hutan dan Lahan," vol. 2, no. 2, pp. 74–87, 2024.
- [12] I. I. Journal, "IJIS Indonesian Journal on Information System e- ISSN 2548-6438 p-ISSN 2614-7173," vol. 6, no. September, pp. 169–180, 2021.
- [13] B. Riswanto, P. Gymnastiar, A. Sudarmanto, S. Akbar, and R. Mubarak, "WEB GIS untuk Monitoring Lahan Pertanian Berkelanjutan," pp. 20–29,

2025.

- [14] I. Engineering *et al.*, “Rancang Bangun Sistem Informasi Geografis,” vol. 7, no. 2, pp. 375–380, 2023.
- [15] S. Gis, “Nusantara Computer and Design Review Pengembangan Sistem Pemetaan Masjid Berbasis Web Geographic Information,” pp. 31–36, 2024.
- [16] T. D. Arsyad *et al.*, “Implementasi Algoritma Dijkstra dalam mencari rute terpendek dari Universitas Negeri Medan,” vol. 9, no. 1, pp. 235–242, 2025.
- [17] N. F. Lakutu, M. R. Katili, S. L. Mahmud, and N. Imansyah, “Algoritma Dijkstra dan Algoritma Greedy Untuk Optimasi Rute Pengiriman Barang Pada Kantor Pos Gorontalo,” vol. 11, no. 1, pp. 55–65, 2023.
- [18] W. Alif, S. Riad, and R. H. Cian, “Implementasi Dijkstra Pada Aplikasi Pemesanan Tiket Wisata Untuk Rekomendasi Lokasi Wisata Terdekat,” *J. Manaj. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 52–61, 2024.
- [19] A. A. Maliq, Y. Darmi, D. Deslianti, and A. W. Mahfuzi, “Implementation Of The A* (A-Star) Algorithm For Website-Based Search Of Tourism Sites In The City Of Bengkulu,” *J. Komputer, Inf. dan Teknol.*, vol. 3, no. 2, pp. 337–346, 2023, doi: 10.53697/jkomitek.v3i2.1213.
- [20] L. H. Al Hadi and Sularno, “Perancangan Aplikasi Sistem Informasi Geografis Tempat Wisata Di Kota Payakumbuh Berbasis Web,” *J. Informatics Bus.*, vol. 1, no. 04, pp. 327–330, 2024, [Online]. Available: file:///C:/Users/T410/Downloads/MySQL sumber 8.pdf
- [21] A. Pratama, C. Amalia, and R. Putra Fhonna, “Sistem Informasi Geografis Pemetaan Wilayah Kelayakan Tanaman Cabai Menggunakan Metode K-Means,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 2548–2554, 2024, doi: 10.36040/jati.v8i3.9155.
- [22] I. L. Kharisma and A. K. Arrahman, “Kerusakan Jalan di Wilayah Kecamatan Bayongbong,” *J. Restikom Ris. Tek. Inform. dan Komput.*, vol. 3, no. 3, pp. 150–159, 2021.
- [23] M. Rizal, W. Aji, and T. D. Wismarini, “JTIM : Jurnal Teknologi Informasi dan Multimedia Webgis Fasilitas Kesehatan di Kabupaten Demak dengan Leaf-let dan Web Mapping Interaktif Client-Side,” vol. 7, no. 3, pp. 587–600, 2025.
- [24] O. Arifin and A. R. Supriyatna, “Menggunakan Leaflet Js Dan Geojson,” vol. 17, pp. 364–371, 2023.
- [25] U. U. Indonesia *et al.*, “Rancang Bangun Sistem Informasi Geografis Layanan Kesehatan Di Kecamatan Kota Sigli Dengan Framework Laravel Design Of Geographic Information System For Health Services In Kota Sigli District With Laravel,” vol. 10, no. 2, 2024.

LAMPIRAN

Lampiran Source Code 1

```
import RouteLine from '@components/RouteLine';
import AOS from 'aos';
import 'aos/dist/aos.css';
import { LeafletMouseEvent } from 'leaflet';
import 'leaflet/dist/leaflet.css';
import { ChevronDown } from 'lucide-react';
import { useEffect, useState } from 'react';
import { MapContainer, Marker, Popup, TileLayer, useMap,
useMapEvents } from 'react-leaflet';
import Footer from '../components/Footer';
import MainLayout from '../layouts/MainLayout';

interface Wisata {
  id: number;
  nama: string;
  lokasi: string;
  gambar: string;
  latitude: number;
  longitude: number;
  deskripsi?: string;
  views?: number;
  distance?: number;
}

interface Props {
  wisataPopuler: Wisata[];
}

const Main = ({ wisataPopuler = [] }: Props) => {
  const [showRecommendations, setShowRecommendations] =
  useState(false);
  const [showMapModal, setShowMapModal] = useState(false);
  const [marker, setMarker] = useState<[number, number] |
  null>(null);
  const [locationText, setLocationText] = useState('');
  const [open, setOpen] = useState(false);
  const [selected, setSelected] = useState('Alam');
  const [, setShowEmptyMessage] = useState(false);
  const [recommendations, setRecommendations] =
  useState<Wisata[]>([]);

  interface Wisata {
    id: number;
    nama: string;
    lokasi: string;
    gambar: string;
    latitude: number;
    longitude: number;
    deskripsi?: string;
    views?: number;
    distance?: number;
  }

  useEffect(() => {
```

```

    AOS.init({ duration: 1000, once: false });
    const handleScroll = () => {
      const scrollTop = window.scrollY;
      const docHeight = document.body.scrollHeight;
      const winHeight = window.innerHeight;

      const scrolled = scrollTop / (docHeight - winHeight);

      if (scrolled > 0.7) {
        setShowEmptyMessage(true);
      }
    };

    window.addEventListener('scroll', handleScroll);
    return () => window.removeEventListener('scroll',
handleScroll);
  }, []);

  function FixMapSize() {
    const map = useMap();
    useEffect(() => {
      // Pastikan map dirender setelah modal benar-benar
tampil
      const timeout = setTimeout(() => {
        map.invalidateSize();
      }, 500);
      return () => clearTimeout(timeout);
    }, [map]);
    return null;
  }

  function LocationMarker({
    setMarker,
    setLocationText,
  }: {
    setMarker: React.Dispatch<React.SetStateAction<[number,
number] | null>>;
    setLocationText:
React.Dispatch<React.SetStateAction<string>>;
  }) {
    useMapEvents({
      click(e: LeafletMouseEvent) {
        const { lat, lng } = e.latlng;
        setMarker([lat, lng]);
        setLocationText(`Lat: ${lat.toFixed(5)}, Lng:
${lng.toFixed(5)}`);
      },
    });
    return null;
  }

  function getDistance(lat1: number, lon1: number, lat2: number,
lon2: number) {
    const R = 6371; // km
    const dLat = (lat2 - lat1) * (Math.PI / 180);
    const dLon = (lon2 - lon1) * (Math.PI / 180);
    const a =
      Math.sin(dLat / 2) * Math.sin(dLon / 2) +

```

```

        Math.cos(lat1 * (Math.PI / 180)) * Math.cos(lat2 *
(Math.PI / 180)) * Math.sin(dLon / 2) * Math.sin(dLon / 2);
        const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
        return R * c; // jarak dalam km
    }

    const handleExplore = async () => {
        if (!marker) {
            alert('Silakan pilih lokasi terlebih dahulu.');
```

```

            return;
        }

        type WisataAPI = {
            id: number;
            nama: string;
            latitude: string;
            longitude: string;
            deskripsi?: string;
            gambar?: string;
        };

        try {
            const res = await
fetch(`http://127.0.0.1:8000/api/wisata/jenis/${selected}`);
            const json = await res.json();

            const data: WisataAPI[] = json.data || [];

            // Mapping data
            const mapped = data.map((item) => ({
                id: item.id,
                name: item.nama,
                lat: parseFloat(item.latitude),
                lng: parseFloat(item.longitude),
                desc: item.deskripsi || 'Tidak ada deskripsi',
                imageUrl: item.gambar ?
`http://127.0.0.1:8000/storage/${item.gambar}` :
'/images/noimage.jpg',
            }));

            // Hitung jarak dari user ke masing-masing wisata
            const withDistance = mapped.map((rec) => ({
                ...rec,
                distance: getDistance(marker[0], marker[1], rec.lat,
rec.lng),
            }));

            // Urutkan dari jarak terdekat ke terjauh
            // Urutkan dari jarak terdekat
            const sorted = withDistance.sort((a, b) => a.distance
- b.distance);

            // Ambil 5 terdekat
            const nearestFive = sorted.slice(0, 5);

            // Mapping ke format Wisata
            const mappedNearest: Wisata[] = nearestFive.map((item)
=> ({
                id: item.id,
```

```

        nama: item.name,
        lokasi: item.desc,
        gambar: item.imageUrl,
        latitude: item.lat,
        longitude: item.lng,
        deskripsi: item.desc,
        distance: item.distance,
    )));

    setRecommendations(mappedNearest);
} catch (error) {
    console.error('Gagal mengambil data wisata:', error);
    setRecommendations([]);
}

setShowRecommendations(true);
};

const routeByJenis = (jenis: string) => {
    switch (jenis) {
        case 'Alam':
            return '/wisata-alam';
        case 'Buatan dan Taman':
            return '/wisata-buatan';
        case 'Budaya dan Sejarah':
            return '/wisata-budaya';
        case 'Religi':
            return '/wisata-religi';
        default:
            return '/';
    }
};

return (
    <MainLayout>
        { /* Hero Section */ }
        <section className="relative -mt-14 bg-cover bg-center
pb-36" style={{ backgroundImage: "url('/images/gslamet.png')" }}>
            <div className="absolute inset-0 bg-sky-800 opacity-
70"></div>
                <div className="mx-auto -mt-6.5 flex max-w-7xl
items-center justify-between px-8 pt-95">
                    { /* Left Content */ }
                    <div className="max-w-xl">
                        <h1 className="mb-4 text-5xl leading-tight
font-extrabold text-white" data-aos="fade-up">
                            Jelajahi Keindahan <span className="text-
sky-300">Banyumas</span>
                        </h1>
                        <p className="mb-4 font-semibold text-white"
data-aos="fade-up">
                            Banyumas memiliki banyak destinasi
wisata menarik dari alam, budaya, taman hingga religi. Saatnya
berpetualang dan temukan
                            keindahan lokal yang memukau!
                        </p>
                        <p className="font-semibold text-white"
data-aos="fade-up">

```

```

        Masih bingung mau wisata ke mana? Pilih
jenis wisata dan lokasi kamu, nanti sistem akan merekomendasikan
lokasi wisata

        terdekat dan yang mungkin kamu sukai.
    </p>
</div>
</div>

    {/* Explore Form - diposisikan di perbatasan biru-
putih */}
    <div className="absolute bottom-0 left-1/2 z-10 w-
full max-w-5xl -translate-x-1/2 translate-y-1/2 rounded-xl bg-white
shadow-lg ring-1 ring-black/10">
        <div className="flex flex-col justify-between
px-6 py-6 sm:flex-row sm:items-center sm:gap-6">
            {/* Jenis Wisata */}
            <div className="relative mb-4 flex flex-col
sm:mb-0">
                <label className="mb-1 text-sm font-
extrabold text-sky-800">Jenis Wisata</label>
                <div className="relative">
                    <button
                        type="button"
                        onClick={() => setOpen(!open)}
                        className="flex w-48 items-
center justify-between rounded-md border border-sky-500 bg-white
px-4 py-2 text-sm font-semibold text-gray-900 shadow-sm transition
focus:ring-2 focus:ring-sky-600 focus:outline-none"
                    >
                        <span>{selected}</span>
                        <ChevronDown className={`h-4 w-
4 transition-transform ${open ? 'rotate-180' : ''}`} />
                    </button>

                    {open && (
                        <div className="absolute z-20
mt-1 w-48 rounded-md border border-sky-500 bg-white font-black
shadow-lg">
                            {[ 'Alam', 'Buatan dan
Taman', 'Budaya dan Sejarah', 'Religi' ].map((opt) => (
                                <button
                                    key={opt}
                                    onClick={() => {
                                        setSelected(opt);
                                        setOpen(false);
                                    }}
                                    className={`block
w-full px-4 py-2 text-left text-sm font-medium ${
                                        selected ===
opt ? 'bg-sky-100 text-sky-800' : 'font-black text-gray-900
hover:bg-sky-50'
                                    }}
                                >
                                    {opt}
                                </button>
                            ))}
                        </div>
                    )}
                </div>
            </div>
        </div>
    </div>

```

```

        </div>

        {/ * Lokasi Awal */}
        <div className="relative mb-4 flex flex-col
sm:mb-0">
            <label className="mb-1 text-sm font-
extrabold text-sky-800">Lokasi Awal</label>
            <input
                type="text"
                placeholder="Klik untuk pilih lokasi
di peta"
                value={locationText}
                readOnly
                onClick={() => setShowMapModal(true)}
                className="cursor-pointer rounded-
md border border-sky-400 bg-gray-50 px-4 py-2 text-sm text-gray-900
shadow-sm focus:ring-2 focus:ring-sky-600 focus:outline-none"
            />
        </div>

        {/ * Modal Map Fullscreen */}
        {showMapModal && (
            <div className="fixed inset-0 z-50 flex
items-center justify-center bg-black/60">
                <div className="w-full max-w-5xl
overflow-hidden rounded-lg bg-white shadow-2xl">
                    {/ * Header Modal */}
                    <div className="flex items-
center justify-between bg-white px-6 py-4 text-center">
                        <h2 className="text-lg font-
semibold text-sky-600">Pilih Lokasi Anda</h2>
                        <button onClick={() =>
setShowMapModal(false)} className="text-xl font-bold text-sky-600
hover:text-sky-800">
                            X
                        </button>
                    </div>

                    {/ * Map Container */}
                    <div className="relative h-
[80vh] w-full">
                        <MapContainer
                            center={[-7.4314,
109.2478]}
                            zoom={12}
                            scrollWheelZoom={true}
                            className="z-0 h-full
w-full border border-black"
                        >
                            <FixMapSize />
                            <TileLayer
                                attribution='&copy;
<a href="https://osm.org/copyright">OpenStreetMap</a>'
                                url="https://{s}.t
ile.openstreetmap.org/{z}/{x}/{y}.png"
                            />
                            {marker && <Marker
position={marker} />}
                    </div>
                </div>
            </div>
        )}

```

```

                                <LocationMarker
setMarker={setMarker} setLocationText={setLocationText} />
                                </MapContainer>

                                {/* Tombol Konfirmasi Lokasi
*/}

                                <div className="absolute
bottom-4 left-1/2 -translate-x-1/2">
                                    <button
                                        onClick={() =>
setShowMapModal(false)}
                                        className="rounded-
md bg-sky-800 px-6 py-3 text-sm font-semibold text-white shadow-md
hover:bg-sky-700"
                                    >
                                        Konfirmasi Lokasi
                                    </button>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
            {/* Tombol Explore */}
            <div className="flex justify-center
sm:justify-end">
                <button
                    onClick={handleExplore}
                    className="mt-2 rounded-md bg-sky-
600 px-6 py-3 text-sm font-semibold text-white transition hover:bg-
sky-800"
                >
                    Explore Now
                </button>
            </div>
        </div>
    </div>
</section>

    {/* Rekomendasi Wisata */}
    {showRecommendations && (
        <section className="shadow-in bg-white px-6 py-25
pt-23 pb-2" data-aos="fade-up">
            {/* Header */}
            <div className="mb-4 flex items-center justify-
between">
                <button onClick={() =>
setShowRecommendations(false)} className="text-xl font-bold text-
sky-600 hover:text-sky-800">
                    X
                </button>
            </div>
            {/* JIKA DATA KOSONG */}
            {recommendations.length === 0 ? (
                <div className="flex flex-col items-center
justify-center py-20 text-center">
                    {/* Icon atau ilustrasi */}
                    <svg
                        xmlns="http://www.w3.org/2000/svg"

```

```

                                className="mb-6 h-24 w-24 text-
gray-400"

                                fill="none"
                                viewBox="0 0 24 24"
                                stroke="currentColor"
                                strokeWidth={1.5}
                                >
                                <path
                                    strokeLinecap="round"
                                    strokeLinejoin="round"
                                    d="M3 17a4 4 0 014-4h10a4 4 0
014 4M7 7a5 5 0 010 10 0 5 5 0 00-10 0z"
                                />
                                </svg>

                                <h3 className="mb-2 text-2xl font-bold
text-gray-700">Wisata Belum Ada</h3>

                                <p className="max-w-md text-gray-500">
                                    Maaf, data wisata untuk kategori
ini belum tersedia. Coba pilih jenis wisata lain untuk melihat
rekomendasi terdekat.
                                </p>
                                </div>
                                ) : (
                                <>
                                    { /* MAP */ }
                                    <div className="mb-6 h-[400px] w-full
overflow-hidden rounded-xl shadow-lg">
                                        <MapContainer center={[-7.4314,
109.2478]} zoom={12} scrollWheelZoom={true} className="h-full w-
full">
                                            <TileLayer attribution="&copy;
OpenStreetMap"
url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png" />

                                                {marker && <Marker
position={marker}></Marker>

                                                    {recommendations.map((rec) =>
(
                                                        <Marker key={rec.id}
position={[rec.latitude, rec.longitude]}>
                                                            <Popup>
                                                                <strong>{rec.nama}
</strong>
                                                                    <br />
                                                                    {rec.deskripsi}
                                                                </Popup>
                                                            </Marker>
                                                        )))
                                                    </div>
                                                    { /* CARD */ }

```

```

<div className="relative">
  <h2 className="mb-4 text-2xl font-
extrabold text-sky-800">Wisata Terdekat dari Lokasi Kamu</h2>
  <div className="scrollbar-hide flex
gap-6 overflow-x-auto pb-4">
    {recommendations.map((rec) =>
      (
        <div
          key={rec.id}
          className="max-w-[300px]
min-w-[300px] flex-shrink-0 overflow-hidden rounded-xl border
border-gray-200 bg-white shadow-md transition hover:shadow-lg"
        >
          {/* Gambar */}
          <img src={rec.gambar}
alt={rec.nama} className="h-40 w-full object-cover" />

          {/* Isi Card */}
          <div className="p-4">
            <h3 className="text-
lg font-bold text-sky-800">{rec.nama}</h3>
            <p className="mt-2
line-clamp-3 text-sm text-gray-700">{rec.deskripsi}</p>

            {/* Tombol Detail
*/}

            <a
              href={`\wisata
/${rec.id}`}
              className="mt-
4 inline-block rounded-md bg-sky-600 px-4 py-2 text-sm font-
semibold text-white hover:bg-sky-800"
            >
              Lihat Detail
            </a>
          </div>
        </div>
      )
    )
  </div>
</div>
)}
{/* WISATA POPULER */}
<section className="mt-12">
  <div className="relative">
    <h2 className="mb-4 text-2xl font-
extrabold text-sky-800">Wisata Populer</h2>

    <div className="scrollbar-hide flex
gap-6 overflow-x-auto pb-4">
      {wisataPopuler.map((item) => (
        <div
          key={item.id}
          className="max-w-[300px]
min-w-[300px] flex-shrink-0 overflow-hidden rounded-xl border
border-gray-200 bg-white shadow-md transition hover:shadow-lg"
        >
          {/* Gambar */}
          <img

```

```

src={item.gambar ?
`/storage/${item.gambar}` : '/images/default.jpg'}
alt={item.nama}
className="h-40 w-full
object-cover"
/>

{/* Isi Card */}
<div className="p-4">
  <h3 className="text-lg
font-bold text-sky-800">{item.nama}</h3>

  <p className="text-sm
text-gray-500">{item.lokasi}</p>

  <p className="mt-1 text-
xs text-gray-400">{item.views} views</p>

  {/* Tombol Detail */}
  <a
    href={`\wisata/${i
tem.id}`}
    className="mt-4
inline-block rounded-md bg-sky-600 px-4 py-2 text-sm font-semibold
text-white hover:bg-sky-800"
    >
    Lihat Detail
  </a>
</div>
</div>
)}}
</div>
</section>

<div className="mt-12 text-center">
  <a
    href={routeByJenis(selected)}
    className="inline-block rounded-lg bg-
sky-700 px-8 py-3 text-lg font-bold text-white hover:bg-sky-800"
    >
    Lihat Wisata Lainnya →
  </a>
</div>
</section>
)}

{/* About Section */}
<section className="mt-[-10px] bg-white px-6 py-24">
  <div className="mx-auto max-w-5xl text-center">
    <h2 className="mb-6 text-3xl font-extrabold
text-sky-800">Tentang Banyumas</h2>
    <p className="text-justify text-lg leading-
relaxed font-black text-gray-600">
      Banyumas adalah permata tersembunyi di Jawa
Tengah yang kaya akan pesona alam, budaya, dan sejarah. Terletak di
kaki Gunung

```

```

        Slamet, daerah ini memiliki udara sejuk dan
pemandangan hijau yang memanjakan mata. Dari curug-curug yang
memukau seperti
        Curug Cipendok, hingga keindahan Lokawisata
Baturraden yang menjadi magnet wisata keluarga dan pecinta alam,
Banyumas adalah
        destinasi yang layak dijelajahi oleh siapa
pun yang mendambakan liburan penuh petualangan dan relaksasi.
        <br />
        <br />
        Tidak hanya menyajikan keindahan alam,
Banyumas juga menyimpan kekayaan budaya seperti tradisi lengger dan
batik Banyumasan
        yang unik. Kuliner khas seperti mendoan,
sroto, dan gethuk goreng siap memanjakan lidah pengunjung. Banyumas
bukan hanya
        tempat untuk berwisata, melainkan pengalaman
yang menyatu antara keindahan alam, kearifan lokal, dan keramahan
masyarakatnya.
        <br />
        <br />
        Jadi, tunggu apa lagi? Mari temukan cerita
dan keajaiban yang hanya bisa kamu rasakan langsung di Banyumas!
    </p>
</div>
</section>
<Footer />
</MainLayout>
    );
};

export default Main;

```

Lampiran Source Code 2

```

import { router, useForm } from '@inertiajs/react';
import { ArrowLeft } from 'lucide-react';
import React from 'react';

export default function LoginPage() {
  const { data, setData, processing, errors } = useForm({
    nama: '',
    password: '',
  });

  const handleSubmit = (e: React.FormEvent) => {
    e.preventDefault();
    router.post('/login', data, {
      onSuccess: () => {
        router.visit('/admindash/dashadmin');
      },
    });
  };

  const goBack = () => {
    router.visit('/');
  };
}

```

```

    return (
      <div className="flex h-screen items-center justify-center
bg-sky-400">
        <div className="flex w-4/5 max-w-5xl overflow-hidden
rounded-lg bg-white shadow-lg">
          <div className="relative w-1/2">
            

            <button
              onClick={goBack}
              className="absolute top-1/2 right-[-24px]
-translate-y-1/2 rounded-full bg-sky-800 p-3 text-white shadow-
lg transition hover:bg-sky-700"
            >
              <ArrowLeft size={20} />
            </button>
          </div>

          <div className="flex w-1/2 flex-col items-center
justify-center bg-sky-800 p-10">
            <h2 className="mb-2 text-2xl font-bold text-
white">WANDER BANYUMAS</h2>
            <p className="mb-6 text-sm text-white">Admin
Only</p>

            <form onSubmit={handleSubmit} className="w-
full max-w-sm">
              <div className="mb-4">
                <label className="mb-2 block text-sm
text-white">Username</label>
                <input
                  type="text"
                  value={data.nama}
                  onChange={(e) => setData('nama',
e.target.value)}
                  placeholder="Enter your username"
                  className="w-full rounded-md
border border-gray-300 px-4 py-2 text-black placeholder-gray-400
focus:ring-2 focus:ring-white focus:outline-none"
                />
                {errors.nama && <p className="mt-1
text-sm text-red-400">{errors.nama}</p>}
              </div>

              <div className="mb-4">
                <label className="mb-2 block text-sm
text-white">Password</label>
                <input
                  type="password"
                  value={data.password}
                  onChange={(e) => setData('password',
e.target.value)}
                  placeholder="Enter your password"
                  className="w-full rounded-md
border border-gray-300 px-4 py-2 text-black placeholder-gray-400
focus:ring-2 focus:ring-white focus:outline-none"
                />
              </div>
            </form>
          </div>
        </div>
      </div>
    )
  )
}

```

```

                                {errors.password && <p className="mt-
1 text-sm text-red-400">{errors.password}</p>}
                                </div>

                                <button
                                  type="submit"
                                  disabled={processing}
                                  className="w-full rounded-full bg-
white py-2 font-semibold text-sky-800 shadow hover:bg-gray-100"
                                >
                                  {processing ? 'Loading...' : 'ENTER'}
                                </button>
                              </form>
        </div>
      </div>
    </div>
  );
}

```

Lampiran Source Code 3

```

import AdminLayout from '@layouts/AdminLayout';
import { Link } from '@inertiajs/react';

interface DashboardProps {
  totalWisata: number;
  totalViews: number;
}

export default function DashAdmin({ totalWisata, totalViews }:
DashboardProps) {
  return (
    <AdminLayout>
      {/* Header */}
      <div className="mb-8 flex items-center justify-
between">
        <div>
          <h1 className="text-3xl font-bold text-sky-
800">Dashboard Admin</h1>
          <p className="text-gray-500">Selamat datang
di halaman admin</p>
        </div>

        <Link href={route('datawisata.create')}
className="rounded-lg bg-sky-600 px-5 py-2 text-white shadow-md
transition hover:bg-sky-700">
          + Tambah Wisata
        </Link>
      </div>

      {/* Konten Dashboard */}
      <div className="grid grid-cols-1 gap-6 md:grid-cols-
2">
        <div className="rounded-xl bg-white p-6 shadow-
sm">

```

```

        <h3 className="text-sm font-medium text-gray-500">Total Wisata</h3>
        <p className="mt-2 text-3xl font-bold text-sky-600">{totalWisata}</p>
      </div>

      <div className="rounded-xl bg-white p-6 shadow-sm">
        <h3 className="text-sm font-medium text-gray-500">Total Pengunjung</h3>
        <p className="mt-2 text-3xl font-bold text-emerald-600">{totalViews}</p>
      </div>
    </div>
  </AdminLayout>
);
}

```

Lampiran Source Code 4

```

import AdminLayout from '@layouts/AdminLayout';
import { Link, router, usePage } from '@inertiajs/react';
import { useState } from 'react';

interface Wisata {
  id: number;
  nama: string;
  lokasi: string;
  gambar: string | null;
  deskripsi?: string;
  jenis_wisata?: string;
}

export default function DataWisata({ wisata }: { wisata: Wisata[] }) {
  const { url } = usePage();

  // Ambil query filter jenis wisata
  const currentQuery = new URLSearchParams(url.split('?')[1]);
  const selectedJenis = currentQuery.get('jenis') || '';

  // Sorting: none | az | za
  const [sortOrder, setSortOrder] = useState<string>('none');

  // Hapus data
  const handleDelete = (id: number) => {
    if (confirm('Yakin ingin menghapus data ini?')) {
      router.delete(route('admin.wisata.destroy', id), {
        onSuccess: () => alert('Data wisata berhasil dihapus!'),
        onError: () => alert('Gagal menghapus data wisata.'),
      });
    }
  };
}

```

```

// Filter jenis wisata
const handleFilterChange = (value: string) => {
  const query = value ? `?jenis=${value}` : '';
  router.visit(`/admindash/datawisata${query}`);
};

// Sorting final, hanya berjalan jika sortOrder bukan 'none'
const finalData = [...wisata];

if (sortOrder === 'az') {
  finalData.sort((a, b) => a.nama.localeCompare(b.nama));
} else if (sortOrder === 'za') {
  finalData.sort((a, b) => b.nama.localeCompare(a.nama));
}

return (
  <AdminLayout>
    {/* FILTER DAN SORT */}
    <div className="mb-6 flex flex-col gap-4 md:flex-row
md:items-center md:justify-between">
      <div className="mt-3 flex flex-wrap items-center
gap-6">
        {/* FILTER JENIS WISATA */}
        <select
          className="flex w-48 items-center justify-
between rounded-md border border-sky-500 bg-white px-4 py-2 text-
sm font-semibold text-gray-900 shadow-sm transition focus:ring-2
focus:ring-sky-600 focus:outline-none"
          value={selectedJenis}
          onChange={ (e) =>
handleFilterChange(e.target.value)}
        >
          <option value="">Semua Jenis
Wisata</option>
          <option value="Alam">Alam</option>
          <option value="Buatan dan Taman">Buatan
dan Taman</option>
          <option value="Budaya dan Sejarah">Budaya
dan Sejarah</option>
          <option value="Religi">Religi</option>
        </select>
        {/* SORTING RADIO */}
        <div className="flex items-center gap-4 text-
sm font-semibold text-gray-700">
          {/* NONE */}
          <label className="flex cursor-pointer
items-center gap-2">
            <input
              type="radio"
              name="sort"
              value="none"
              checked={sortOrder === 'none'}
              onChange={ () =>
setSortOrder('none')}
              className="h-4 w-4 accent-blue-
600"
            />
            Tidak aktif

```

```

        </label>

        {/ * A-Z */}
        <label className="flex cursor-pointer
items-center gap-2">

            <input
                type="radio"
                name="sort"
                value="az"
                checked={sortOrder === 'az'}
                onChange={() => setSortOrder('az')}
                className="h-4 w-4 accent-blue-
600"
            />
            A - Z
        </label>

        {/ * Z-A */}
        <label className="flex cursor-pointer
items-center gap-2">

            <input
                type="radio"
                name="sort"
                value="za"
                checked={sortOrder === 'za'}
                onChange={() => setSortOrder('za')}
                className="h-4 w-4 accent-blue-
600"
            />
            Z - A
        </label>
    </div>
</div>

    {/ * TOMBOL AKSI */}
    <div className="flex items-center gap-4">
        <Link href={route('datawisata.create')}
className="rounded-lg bg-blue-600 px-5 py-2 text-white shadow-md
hover:bg-blue-700">
            + Tambah Wisata
        </Link>

        <Link href={route('datawisata.import')}
className="rounded-lg bg-green-600 px-5 py-2 text-white shadow-
md hover:bg-green-700">
            Import CSV
        </Link>
    </div>
</div>

    {/ * GRID DATA */}
    <div className="grid grid-cols-1 gap-6 sm:grid-cols-
2 md:grid-cols-3">
        {finalData.map((item: Wisata) => (
            <div key={item.id} className="group relative
rounded-xl border border-gray-100 bg-white p-4 shadow-md">
                <img src={`/storage/${item.gambar}`}
alt={item.nama} className="h-40 w-full rounded-lg object-cover"
/>

```

```

                                <h3 className="mt-3 font-semibold text-
gray-800">{item.nama}</h3>
                                <p className="text-sm text-gray-
600">{item.lokasi}</p>
                                <span className="mt-1 inline-block text-
xs text-slate-500">{item.jenis_wisata}</span>

                                { /* TOMBOL EDIT & HAPUS */ }
                                <div className="absolute top-2 right-2
opacity-0 transition group-hover:opacity-100">
                                    <button
                                        onClick={() =>
router.visit(route('admin.wisata.edit', item.id))}
                                        className="mx-1 rounded bg-blue-
500 px-2 py-1 text-xs text-white hover:bg-blue-600"
                                    >
                                        Edit
                                    </button>

                                    <button
                                        onClick={() =>
handleDelete(item.id)}
                                        className="mx-1 rounded bg-red-
500 px-2 py-1 text-xs text-white hover:bg-red-600"
                                    >
                                        Hapus
                                    </button>
                                </div>
                            </div>
                        )))}
                    </div>
                </AdminLayout>
            );
        }

```

Lampiran Source Code 5

```

import AdminLayout from '@layouts/AdminLayout';
import { useForm } from '@inertiajs/react';
import React, { useState } from 'react';

export default function DataWisataCreate() {
    const { data, setData, post, reset, errors } = useForm({
        nama: '',
        jenis_wisata: '',
        deskripsi: '',
        lokasi: '',
        latitude: '',
        longitude: '',
        gambar: null as File | null,
        jam_operasional: '',
        harga_tiket: '',
        fasilitas: '',
        maps_url: '',
    });
}

```

```

const [preview, setPreview] = useState<string | null>(null);

const handleSubmit = (e: React.FormEvent) => {
  e.preventDefault();

  const formData = new FormData();
  Object.entries(data).forEach(([key, value]) => {
    if (value !== null && value !== undefined) {
      if (value instanceof File) {
        formData.append(key, value);
      } else {
        formData.append(key, String(value));
      }
    }
  });

  post(route('admin.wisata.store'), {
    forceFormData: true,
    onSuccess: () => {
      reset();
      setPreview(null);
      alert('Data wisata berhasil disimpan!');
    },
    onError: () => alert('Gagal menyimpan data wisata.'),
  });
};

return (
  <AdminLayout>
    {/* FORM TAMBAH DATA */}
    <div className="mx-auto max-w-4xl rounded-2xl bg-white p-8 shadow-md">
      <h1 className="mb-6 text-2xl font-semibold text-sky-800">Form Tambah Wisata</h1>

      <form onSubmit={handleSubmit}
        encType="multipart/form-data" className="grid grid-cols-1 gap-6 md:grid-cols-2">
        {/* Nama Wisata */}
        <div className="flex flex-col">
          <label htmlFor="nama" className="text-sm font-bold text-gray-800">
            Nama Wisata
          </label>
          <input
            id="nama"
            type="text"
            value={data.nama}
            onChange={(e) => setData('nama', e.target.value)}
            className="rounded-lg border border-gray-300 p-2 font-bold text-gray-800 focus:ring-2 focus:ring-blue-400 focus:outline-none"
            required
          />
          {errors.nama && <p className="text-sm text-red-500">{errors.nama}</p>}
        </div>

```

```

        { /* Latitude */}
        <div className="flex flex-col">
            <label htmlFor="latitude" className="mb-
1 text-sm font-bold text-gray-800">
                Latitude
            </label>
            <input
                id="latitude"
                type="text"
                value={data.latitude}
                onChange={(e) => setData('latitude',
e.target.value)}
                className="rounded-lg border border-
gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-
blue-400 focus:outline-none"
            />
        </div>

        { /* Longitude */}
        <div className="flex flex-col">
            <label htmlFor="longitude" className="mb-
1 text-sm font-bold text-gray-700">
                Longitude
            </label>
            <input
                id="longitude"
                type="text"
                value={data.longitude}
                onChange={(e) => setData('longitude',
e.target.value)}
                className="rounded-lg border border-
gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-
blue-400 focus:outline-none"
            />
        </div>

        { /* Jenis Wisata */}
        <div className="flex flex-col">
            <label htmlFor="jenis_wisata"
className="mb-1 text-sm font-bold text-gray-800">
                Jenis Wisata
            </label>
            <select
                id="jenis_wisata"
                value={data.jenis_wisata || ''}
                onChange={(e) => setData('jenis_wisata',
e.target.value)}
                className="rounded-lg border border-
gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-
blue-400 focus:outline-none"
            >

                <option value="">Pilih jenis
wisata</option>
                <option value="Alam">Alam</option>
                <option value="Buatan">Buatan dan
Taman</option>
                <option value="Budaya">Budaya dan
Taman</option>

```

```

        <option value="Religi">Religi</option>
      </select>
    </div>

    {/ * Jam Buka */}
    <div className="flex flex-col">
      <label className="mb-1 text-sm font-bold text-gray-800">Jam Operasional</label>
      <textarea
        rows={4}
        className="rounded-lg border border-gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-blue-400 focus:outline-none"
        value={data.jam_operasional}
        onChange={(e) =>
          setData('jam_operasional', e.target.value)}
      ></textarea>
    </div>

    {/ * Harga Tiket */}
    <div className="flex flex-col">
      <label className="mb-1 text-sm font-bold text-gray-800">Harga Tiket</label>
      <textarea
        className="rounded-lg border border-gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-blue-400 focus:outline-none"
        rows={4}
        value={data.harga_tiket}
        onChange={(e) => setData('harga_tiket',
          e.target.value)}
      ></textarea>
    </div>

    {/ * Fasilitas */}
    <div className="flex flex-col">
      <label className="mb-1 text-sm font-bold text-gray-800">Fasilitas</label>
      <textarea
        className="rounded-lg border border-gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-blue-400 focus:outline-none"
        rows={4}
        value={data.fasilitas}
        onChange={(e) => setData('fasilitas',
          e.target.value)}
      ></textarea>
    </div>

    {/ * Google Maps URL */}
    <div className="flex flex-col">
      <label htmlFor="maps_url" className="mb-1 text-sm font-bold text-gray-800">
        Link Google Maps
      </label>
      <input
        id="maps_url"
        type="text"
        value={data.maps_url}

```

```

                                onChange={ (e) => setData('maps_url',
e.target.value)}
                                className="rounded-lg border border-
gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-
blue-400 focus:outline-none"
                                placeholder="Tempel link Google Maps"
                                />
                                {errors.maps_url && <p className="text-
sm text-red-500">{errors.maps_url}</p>}
                                </div>

                                {/* Lokasi */}
                                <div className="flex flex-col">
                                    <label htmlFor="lokasi" className="mb-1
text-sm font-bold text-gray-800">
                                        Lokasi
                                    </label>
                                    <input
                                        id="lokasi"
                                        type="text"
                                        value={data.lokasi}
                                        onChange={ (e) => setData('lokasi',
e.target.value)}
                                        className="rounded-lg border border-
gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-
blue-400 focus:outline-none"
                                    />
                                </div>

                                {/* Deskripsi */}
                                <div className="flex flex-col md:col-span-2">
                                    <label htmlFor="deskripsi" className="mb-
1 text-sm font-bold text-gray-700">
                                        Deskripsi
                                    </label>
                                    <textarea
                                        id="deskripsi"
                                        value={data.deskripsi}
                                        onChange={ (e) => setData('deskripsi',
e.target.value)}
                                        className="rounded-lg border border-
gray-300 p-3 font-bold text-gray-900 focus:ring-2 focus:ring-
blue-400 focus:outline-none"
                                        placeholder="Tuliskan deskripsi singkat
tentang wisata..."
                                        rows={4}
                                    ></textarea>
                                </div>

                                {/* Gambar */}
                                <div className="flex flex-col md:col-span-2">
                                    <label htmlFor="gambar" className="mb-1
text-sm font-bold text-gray-700">
                                        Gambar
                                    </label>
                                    <input
                                        id="gambar"
                                        type="file"
                                        onChange={ (e) => {

```

```

const file = e.target.files?.[0];
setData('gambar', file || null);
if (file)
setPreview(URL.createObjectURL(file));
    }
    className="rounded-lg border border-
gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-
blue-400 focus:outline-none"
    />
    {preview && <img src={preview} alt="Preview"
className="mt-3 w-48 rounded-lg border border-gray-200" />}
    </div>

    {/* Tombol Simpan */}
    <div className="flex justify-end md:col-span-
2">
        <button type="submit" className="rounded-
lg bg-blue-600 px-6 py-2 text-white shadow-md transition
hover:bg-blue-700">
            Simpan
        </button>
    </div>
</form>
</div>
</AdminLayout>
    );
}

```

Lampiran Source Code 6

```

import AdminLayout from '@layouts/AdminLayout';
import { router, useForm, usePage } from '@inertiajs/react';
import React, { useEffect } from 'react';

interface Wisata {
  id: number;
  nama: string;
  jenis_wisata: string;
  deskripsi: string;
  lokasi: string;
  latitude: string;
  longitude: string;
  gambar: string;
  jam_operasional: string;
  harga_tiket: string;
  fasilitas: string;
  maps_url: string;
}

export default function DataWisataEdit() {
  const { props } = usePage<{ wisata?: Wisata }>();
  const wisata = props.wisata;
  console.log('Props dari Laravel:', props);

  const { data, setData, errors } = useForm({

```

```

        nama: '',
        jenis_wisata: '',
        deskripsi: '',
        lokasi: '',
        latitude: '',
        longitude: '',
        gambar: null as File | null,
        jam_operasional: '',
        harga_tiket: '',
        fasilitas: '',
        maps_url: '',
    });

    useEffect(() => {
        if (wisata) {
            setData({
                nama: wisata.nama || '',
                jenis_wisata: wisata.jenis_wisata || '',
                deskripsi: wisata.deskripsi || '',
                lokasi: wisata.lokasi || '',
                latitude: wisata.latitude || '',
                longitude: wisata.longitude || '',
                gambar: null,
                jam_operasional: wisata.jam_operasional || '',
                harga_tiket: wisata.harga_tiket || '',
                fasilitas: wisata.fasilitas || '',
                maps_url: wisata.maps_url || '',
            });
        }
    }, [wisata, setData]);

    const handleSubmit = (e: React.FormEvent) => {
        e.preventDefault();

        if (!wisata?.id) {
            alert('Data wisata belum siap.');
```

return;

```

        }

        formData.append('_method', 'PUT');

        // Kirim hanya field yang tidak kosong
        Object.entries(data).forEach(([key, value]) => {
            if (value !== '' && value !== null && value !==
undefined) {
                formData.append(key, value as Blob | string);
            }
        });

        console.log('Data yang dikirim ke server:',
Object.fromEntries(formData.entries()));

        // Tetap pakai POST, tapi Laravel akan anggap ini PUT
        (karena ada _method)
        router.post(route('admin.wisata.update', wisata.id),
formData, {
            onStart: () => console.log('Mengirim data ke
server...'),

```

```

        onSuccess: () => alert('Data wisata berhasil
diperbarui!'),
        onError: (errors) => {
            console.error('Gagal update:', errors);
            alert('Terjadi kesalahan saat memperbarui data.');
```

},

```
    });
};

if (!wisata) {
    return (
        <AdminLayout>
            <div className="flex min-h-screen items-center
justify-center text-gray-600">
                <p>Memuat data wisata...</p>
            </div>
        </AdminLayout>
    );
}

return (
    <AdminLayout>
        <div className="mx-auto max-w-4xl rounded-2xl bg-
white p-8 shadow-md">
            <h1 className="mb-6 text-2xl font-semibold text-
sky-800">Edit Data Wisata</h1>

            <form    onSubmit={handleSubmit}
encType="multipart/form-data"    className="grid grid-cols-1 gap-6
md:grid-cols-2">

                {/* Nama Wisata */}
                <div className="flex flex-col">
                    <label className="text-sm font-bold text-
gray-800">Nama Wisata</label>
                    <input
                        type="text"
                        value={data.nama}
                        onChange={(e) => setData('nama',
e.target.value)}
                        className="rounded-lg border border-
gray-300 p-2 font-semibold text-gray-800"
                        required
                    />
                    {errors.nama && <p className="text-sm
text-red-500">{errors.nama}</p>}
                </div>

                {/* Latitude */}
                <div className="flex flex-col">
                    <label className="text-sm font-bold text-
gray-800">Latitude</label>
                    <input
                        type="text"
                        value={data.latitude}
                        onChange={(e) => setData('latitude',
e.target.value)}
                        className="rounded-lg border border-
gray-300 p-2 font-semibold text-gray-800"
                    />
                </div>
            </form>
        </div>
    </AdminLayout>
);

```

```

</div>

    { /* Longitude */}
    <div className="flex flex-col">
        <label className="text-sm font-bold text-gray-800">Longitude</label>
        <input
            type="text"
            value={data.longitude}
            onChange={(e) => setData('longitude',
e.target.value)}
            className="rounded-lg border border-gray-300 p-2 font-semibold text-gray-800"
        />
    </div>

    { /* Jenis Wisata */}
    <div className="flex flex-col">
        <label className="text-sm font-bold text-gray-800">Jenis Wisata</label>
        <select
            value={data.jenis_wisata}
            onChange={(e) => setData('jenis_wisata',
e.target.value)}
            className="rounded-lg border border-gray-300 p-2 font-semibold text-gray-800"
        >
            <option value="">Pilih jenis
wisata</option>
            <option value="Alam">Alam</option>
            <option value="Buatan">Buatan dan
Taman</option>
            <option value="Budaya">Budaya dan
Sejarah</option>
            <option value="Religi">Religi</option>
        </select>
    </div>

    { /* Jam Buka */}
    <div className="flex flex-col">
        <label className="mb-1 text-sm font-bold text-gray-800">Jam Operasional</label>
        <textarea
            rows={4}
            className="rounded-lg border border-gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-blue-400 focus:outline-none"
            value={data.jam_operasional}
            onChange={(e) =>
setData('jam_operasional', e.target.value)}
        ></textarea>
    </div>

    { /* Harga Tiket */}
    <div className="flex flex-col">
        <label className="mb-1 text-sm font-bold text-gray-800">Harga Tiket</label>
        <textarea

```

```

                className="rounded-lg border border-
gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-
blue-400 focus:outline-none"
                rows={4}
                value={data.harga_tiket}
                onChange={(e) => setData('harga_tiket',
e.target.value)}
            ></textarea>
        </div>

        {/* Fasilitas */}
        <div className="flex flex-col">
            <label className="mb-1 text-sm font-bold
text-gray-800">Fasilitas</label>
            <textarea
                className="rounded-lg border border-
gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-
blue-400 focus:outline-none"
                rows={4}
                value={data.fasilitas}
                onChange={(e) => setData('fasilitas',
e.target.value)}
            ></textarea>
        </div>

        {/* Google Maps URL */}
        <div className="flex flex-col">
            <label htmlFor="maps_url" className="mb-
1 text-sm font-bold text-gray-800">
                Link Google Maps
            </label>
            <input
                id="maps_url"
                type="text"
                value={data.maps_url}
                onChange={(e) => setData('maps_url',
e.target.value)}
                className="rounded-lg border border-
gray-300 p-2 font-bold text-gray-900 focus:ring-2 focus:ring-
blue-400 focus:outline-none"
                placeholder="Tempel link Google Maps
(optional)"
            />
            {errors.maps_url && <p className="text-
sm text-red-500">{errors.maps_url}</p>}
        </div>

        {/* Lokasi */}
        <div className="flex flex-col">
            <label className="text-sm font-bold text-
gray-800">Lokasi</label>
            <input
                type="text"
                value={data.lokasi}
                onChange={(e) => setData('lokasi',
e.target.value)}
                className="rounded-lg border border-
gray-300 p-2 font-semibold text-gray-800"
            />

```

```

        </div>

        { /* Deskripsi */}
        <div className="flex flex-col md:col-span-2">
            <label className="text-sm font-bold text-gray-800">Deskripsi</label>
            <textarea
                value={data.deskripsi}
                onChange={ (e) => setData('deskripsi',
e.target.value)}
                className="rounded-lg border border-gray-300 p-2 font-semibold text-gray-800"
                rows={4}
            ></textarea>
        </div>

        { /* Gambar */}
        <div className="flex flex-col md:col-span-2">
            <label className="text-sm font-bold text-gray-800">Gambar</label>
            <input
                type="file"
                onChange={ (e) => setData('gambar',
e.target.files?.[0] || null)}
                className="rounded-lg border border-gray-300 p-2 font-semibold text-gray-800"
            />
        </div>

        <div className="flex justify-end md:col-span-2">
            <button type="submit" className="rounded-lg bg-blue-600 px-6 py-2 text-white shadow-md transition hover:bg-blue-700">
                Simpan Perubahan
            </button>
        </div>
    </form>
</div>
</AdminLayout>
);
}

```

Lampiran Source Code 7

```

import AdminLayout from '@layouts/AdminLayout';
import { router } from '@inertiajs/react';
import React, { useState } from 'react';

export default function ImportWisata() {
    const [file, setFile] = useState<File | null>(null);

    function handleSubmit(e: React.FormEvent) {
        e.preventDefault();
    }
}

```

```

        if (!file) {
            alert('Pilih file CSV terlebih dahulu.');
```

```
            return;
        }

        const formData = new FormData();
        formData.append('file', file);

        router.post(route('datawisata.import'), formData, {
            onSuccess: () => alert('Berhasil import CSV!'),
            onError: () => alert('Gagal import CSV.'),
        });
    }

    return (
        <AdminLayout>
            <div className="mx-auto max-w-xl rounded-xl bg-white
p-8 shadow">
                <h1 className="mb-4 text-2xl font-bold text-sky-
700">Import Data Wisata (CSV)</h1>

                <form onSubmit={handleSubmit}>
                    <input
                        type="file"
                        accept=".csv"
                        className="mb-2 w-full rounded border p-
2 text-gray-700"
                        onChange={ (e) =>
setFile(e.target.files?.[0] || null)}
                    />

                    {file && <p className="mb-4 text-sm font-
medium text-green-700">File dipilih: {file.name}</p>}

                    <button type="submit" className="rounded-lg
bg-green-600 px-6 py-2 text-white hover:bg-green-700">
                        Import Sekarang
                    </button>
                </form>
            </div>
        </AdminLayout>
    );
}

```

Lampiran Source Code 8

```

import AdminLayout from '@layouts/AdminLayout';
import { Head } from '@inertiajs/react';
import { Bar, BarChart, CartesianGrid, ResponsiveContainer,
Tooltip, XAxis, YAxis } from 'recharts';

interface WisataStat {
    id: number;
    nama: string;
    views: number;
}

```

```

interface StatistikProps {
  topWisata: WisataStat[];
  totalWisata: number;
  totalViews: number;
}

export default function Statistik({ topWisata, totalWisata,
totalViews }: StatistikProps) {
  const rataRataViews = totalWisata > 0 ? Math.round(totalViews
/ totalWisata) : 0;

  const wisataTerpopuler = topWisata.length > 0 ? topWisata[0]
: null;

  return (
    <AdminLayout>
      <Head title="Statistik Wisata" />

      {/* SUMMARY (contoh, boleh kamu pisahkan nanti) */}
      <div className="mb-6 grid grid-cols-1 gap-6 md:grid-
cols-4">
        <div className="rounded-xl bg-white p-6 shadow">
          <p className="text-sm text-gray-500">Total
Wisata</p>
          <p className="mt-2 text-3xl font-bold text-
sky-600">{totalWisata}</p>
        </div>

        <div className="rounded-xl bg-white p-6 shadow">
          <p className="text-sm text-gray-500">Total
Pengunjung</p>
          <p className="mt-2 text-3xl font-bold text-
emerald-600">{totalViews.toLocaleString('id-ID')}</p>
        </div>

        <div className="rounded-xl bg-white p-6 shadow">
          <p className="text-sm text-gray-500">Rata-
rata Views</p>
          <p className="mt-2 text-3xl font-bold text-
indigo-600">{rataRataViews.toLocaleString('id-ID')}</p>
        </div>
        <div className="rounded-xl bg-white p-6 shadow">
          <p className="text-sm text-gray-500">Wisata
Terpopuler</p>

          {wisataTerpopuler ? (
            <>
              <p className="mt-2 text-lg font-bold
text-gray-800">{wisataTerpopuler.nama}</p>
              <p className="text-sm text-gray-
500">{wisataTerpopuler.views.toLocaleString('id-ID')} views</p>
            </>
          ) : (
            <p className="mt-2 text-sm text-gray-
400">Belum ada data</p>
          )}
        </div>
      </div>
    </AdminLayout>
  )
}

```

```

        { /* CHART */ }
        <div className="rounded-xl bg-white p-6 shadow">
            <h2 className="mb-4 text-lg font-bold text-gray-
700">Wisata Paling Banyak Dilihat</h2>

            <div className="h-[600px]">
                <ResponsiveContainer width="100%"
height="100%">
                    <BarChart data={topWisata}
layout="vertical" margin={{ top: 10, right: 30, left: 100, bottom:
10 }}>
                        <CartesianGrid strokeDasharray="3 3"
/>
                            <XAxis type="number" />
                            <YAxis type="category" dataKey="nama"
width={180} />
                                <Tooltip />
                                <Bar dataKey="views" fill="#0284c7"
radius={[0, 6, 6, 0]} />
                                    </BarChart>
                                </ResponsiveContainer>
                            </div>
                        </div>
                    { /* TABLE RANKING */ }
                    <div className="mt-8 rounded-xl bg-white p-6 shadow">
                        <h2 className="mb-4 text-lg font-bold text-gray-
700">Ranking Wisata Berdasarkan Views</h2>

                        <div className="overflow-x-auto">
                            <table className="w-full text-sm">
                                <thead>
                                    <tr className="border-b text-left
text-gray-500">
                                        <th className="py-3">No</th>
                                        <th>Nama Wisata</th>
                                        <th className="text-
right">Views</th>
                                        <th className="text-
right">Kontribusi</th>
                                    </tr>
                                </thead>

                                <tbody>
                                    {topWisata.map((item, index) => {
                                        const persen = totalViews > 0 ?
((item.views / totalViews) * 100).toFixed(1) : '0';

                                        return (
                                            <tr key={item.id}
className="border-b last:border-0">
                                                <td className="py-3 text-
gray-800">{index + 1}</td>
                                                <td className="font-medium
text-gray-800">{item.nama}</td>
                                                <td className="text-right
text-gray-800">{item.views.toLocaleString('id-ID')}</td>

```

```

                                <td className="text-right
text-gray-800">{persen}%</td>
                                </tr>
                                );
                                }}}
                                </tbody>
                                </table>
                                </div>
                                </div>
                                </AdminLayout>
                                );
}

```