



Department of Digital Business

Journal of Artificial Intelligence and Digital Business (RIGGS)

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 5 No. 1 (2026) pp: 8350-8369

P-ISSN: 2963-9298, e-ISSN: 2963-914X

Implementasi Algoritma Dijkstra Untuk Menentukan Rute Terpendek Perjalanan Wisata Di Kabupaten Muna

Rahmat Hidayat Sanusi¹, Audy A. Kenap², Efraim R. S. Moningkey³

^{1,2,3}Teknik Informatika, Fakultas Teknik, Universitas Negeri Manado

¹rahmatsanusi160303@gmail.com, ²audyakenap@unima.ac.id, ³fmoningkey@unima.ac.id

Abstrak

Penelitian ini bertujuan untuk mengimplementasikan algoritma Dijkstra dalam menentukan rute terpendek perjalanan wisata di Kabupaten Muna serta membangun sistem berbasis web yang mampu menampilkan visualisasi rute secara interaktif. Kabupaten Muna memiliki banyak destinasi wisata yang tersebar di berbagai wilayah sehingga wisatawan sering mengalami kesulitan dalam menentukan jalur perjalanan yang efisien. Oleh karena itu, diperlukan suatu sistem yang dapat membantu pengguna menemukan rute terpendek agar perjalanan menjadi lebih efektif, hemat waktu, dan biaya. Metode yang digunakan dalam penelitian ini adalah metode pengembangan perangkat lunak Waterfall yang meliputi tahap analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan sistem. Data penelitian berupa koordinat lokasi wisata dan jarak antar lokasi diperoleh melalui observasi, Google Maps, serta data dari instansi terkait. Sistem dibangun menggunakan bahasa pemrograman PHP dengan framework CodeIgniter 4, database MySQL, serta Leaflet.js untuk visualisasi peta, sedangkan perhitungan rute dilakukan menggunakan algoritma Dijkstra pada graf berbobot. Hasil penelitian menunjukkan bahwa algoritma Dijkstra mampu menentukan rute terpendek dari titik awal Kota Raha menuju berbagai destinasi wisata dengan akurat. Sistem yang dihasilkan dapat menampilkan jalur perjalanan pada peta digital serta memberikan informasi jarak tempuh secara otomatis. Dengan adanya sistem ini, wisatawan dapat merencanakan perjalanan dengan lebih efisien, sedangkan pemerintah daerah dapat mememanfaatkannya sebagai media pendukung pengembangan pariwisata berbasis teknologi di Kabupaten Muna.

Kata kunci: Algoritma Dijkstra, Rute Terpendek, Sistem Informasi Geografis, Pariwisata, Kabupaten Muna.

1. Latar Belakang

Dalam sektor pariwisata, Indonesia dikenal memiliki beragam destinasi unggulan yang mampu menarik wisatawan domestik maupun mancanegara. Kontribusi pariwisata terhadap pembangunan daerah dan nasional tidak hanya terlihat dari peningkatan pendapatan domestik bruto, tetapi juga melalui penciptaan lapangan kerja, pelestarian budaya, serta pembangunan infrastruktur. Pada era modern, perkembangan teknologi informasi semakin memperkuat peran sektor pariwisata. Kehadiran teknologi digital memungkinkan pengelolaan destinasi wisata dilakukan lebih efektif sekaligus memberikan pengalaman perjalanan yang lebih baik bagi wisatawan (Jaelani & Hanim, 2021).

Indonesia sebagai negara kepulauan dengan kekayaan alam dan budaya yang melimpah memiliki potensi besar dalam sektor pariwisata. Salah satu daerah yang menyimpan potensi tersebut adalah Kabupaten Muna di Provinsi Sulawesi Tenggara dibuktikan dengan kunjungan wisatawan 2021-2024 sebanyak (2.824.514) (Statistik, 2025). Kabupaten Muna memiliki destinasi wisata beragam, mulai dari wisata pantai hingga gua dan danau. Beberapa objek wisata yang cukup dikenal antara lain Pantai Meleura dengan pasir putih dan air laut birunya, Danau Ubur-Ubur yang unik karena dihuni ubur-ubur tidak menyengat, Gua Liangkabori dengan peninggalan lukisan prasejarah. Keberadaan destinasi wisata ini menjadi modal penting dalam meningkatkan daya tarik Kabupaten Muna sebagai tujuan wisata.

Karena banyaknya destinasi wisata yang tersebar di Kabupaten Muna, wisatawan kesulitan untuk menemukan informasi yang jelas mengenai lokasi objek wisata serta jalur yang harus ditempuh untuk mencapai tempat tujuan dengan waktu yang efisien. Informasi mengenai rute perjalanan menjadi sangat penting agar wisatawan dapat memaksimalkan pengalaman berwisata tanpa harus menghabiskan waktu dan biaya yang berlebihan (Suryana, 2020).

Dalam merencanakan perjalanan, pemilihan jalur terpendek menjadi faktor krusial karena dapat menghemat waktu tempuh, mengurangi konsumsi bahan bakar serta meningkatkan efisiensi biaya perjalanan wisatawan. Tanpa adanya informasi rute yang optimal, wisatawan berpotensi menempuh perjalanan yang lebih jauh, melebihi anggaran yang tersedia, dan mengurangi kenyamanan selama berwisata. Berdasarkan permasalahan tersebut, maka penulis membuat sistem pencarian jalur terpendek yang dapat dijadikan acuan wisatawan untuk menentukan alternatif rute menuju destinasi wisata di Kabupaten Muna. Dengan adanya sistem ini, wisatawan dapat memilih jalur yang searah atau lokasi yang berdekatan sehingga perjalanan menjadi lebih efisien.

Dalam penelitian ini digunakan Algoritma Dijkstra untuk menentukan rute terpendek antar lokasi wisata yang ada di Kabupaten Muna. Algoritma pada dasarnya merupakan proses komputasi yang mengubah input menjadi output dengan langkah-langkah sistematis. Sebuah algoritma dikatakan benar apabila menghasilkan keluaran yang sesuai untuk setiap masukan yang diberikan. Sederhananya, algoritma adalah formula untuk menemukan solusi dari masalah yang ada. Menurut (Anggarawati & Eka, 2024), algoritma dapat dipahami sebagai serangkaian langkah logis dan sistematis yang dirancang untuk menyelesaikan suatu permasalahan secara komprehensif.

Berbagai algoritma dapat digunakan untuk menyelesaikan permasalahan pencarian rute terpendek pada graf terarah, salah satunya adalah Algoritma Dijkstra. Algoritma Dijkstra dirancang untuk menentukan jalur optimal dari node awal menuju node tujuan dengan cara membandingkan bobot terkecil pada setiap lintasan yang tersedia. Melalui mekanisme tersebut, Algoritma Dijkstra mampu menghasilkan rute yang paling efisien di antara beberapa alternatif jalur yang ada (Purba et al., 2025).

Berdasarkan penelitian sebelumnya yang dilakukan oleh (Alif Pratama et al., 2025), Algoritma Dijkstra dapat menyelesaikan masalah pencarian jalur terpendek dari node awal ke node tujuan dalam graf berbobot. Dijkstra merupakan salah satu algoritma optimisasi yang digunakan untuk menemukan lintasan dengan bobot minimum. Pada graf berbobot positif, jalur terpendek selalu diperoleh melalui seleksi bobot terkecil dari node awal menuju node akhir. Dengan demikian, jumlah jalur yang dipilih akan menjadi lintasan dengan nilai total bobot paling rendah.

Berdasarkan uraian yang telah dipaparkan pada latar belakang diatas, penulis mencoba menerapkannya pada pencarian lintasan terpendek wisata di Kabupaten Muna. Dengan judul "Implementasi Algoritma Dijkstra Untuk Menentukan Rute Terpendek Perjalanan Wisata Di Kabupaten Muna" bisa menentukan jalur lokasi wisata yang efektif bagi wisatawan dengan jarak rute terkecil dan waktu tempuh yang sedikit untuk menuju lokasi wisata yang ada di Kabupaten Muna.

2. Metode Penelitian

Metodologi penelitian ini dilaksanakan selama enam bulan, yaitu dari Juli hingga Desember 2025, dengan lokasi penelitian di Kabupaten Muna, Provinsi Sulawesi Tenggara. Pemilihan lokasi dilakukan secara purposive karena daerah ini memiliki potensi pariwisata yang cukup besar sehingga relevan untuk pengembangan sistem penentuan rute wisata berbasis web. Proses penelitian meliputi pengumpulan data, perancangan, hingga pengembangan sistem yang dilakukan secara langsung di lapangan maupun secara daring. Perangkat yang digunakan terdiri dari perangkat keras berupa laptop dengan spesifikasi prosesor AMD A4 dan RAM 8 GB, sedangkan perangkat lunak meliputi bahasa pemrograman PHP, framework CodeIgniter 4, MySQL sebagai basis data, XAMPP sebagai server lokal, Leaflet.js sebagai pustaka pemetaan, OSRM API untuk perhitungan rute, serta Visual Studio Code sebagai editor kode. Data penelitian terdiri atas data spasial berupa koordinat destinasi wisata, data jarak antar lokasi yang diperoleh melalui Google Maps, data sekunder dari Badan Pusat Statistik dan Dinas Pariwisata Kabupaten Muna, serta literatur ilmiah yang berkaitan dengan algoritma graf dan optimasi rute.

Teknik pengumpulan data dilakukan melalui observasi, wawancara, dan studi literatur. Observasi dilakukan dengan mengunjungi langsung beberapa objek wisata untuk memperoleh data kondisi lapangan dan jarak antar lokasi. Wawancara dilakukan dengan pihak terkait, khususnya Dinas Pariwisata, untuk memperoleh informasi mengenai daftar destinasi dan jumlah kunjungan wisatawan. Studi literatur digunakan untuk memperkuat landasan teori mengenai sistem informasi geografis, algoritma graf, dan algoritma Dijkstra sebagai metode pencarian rute terpendek (Pressman, 2014; Sommerville, 2016).

Metode pengembangan sistem yang digunakan adalah model Waterfall karena memiliki tahapan yang sistematis dan berurutan sehingga memudahkan pengendalian proses pengembangan. Tahapan tersebut meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan (Pressman, 2014). Pada tahap analisis kebutuhan dilakukan identifikasi fungsi sistem untuk menampilkan lokasi wisata dan menentukan rute terpendek. Tahap desain meliputi perancangan basis data, flowchart, dan antarmuka pengguna. Implementasi dilakukan dengan pengkodean menggunakan PHP dan MySQL serta integrasi algoritma Dijkstra dan Leaflet untuk visualisasi peta. Pengujian sistem menggunakan Black Box Testing untuk memastikan seluruh fungsi berjalan

sesuai kebutuhan (Sommerville, 2016). Tahap terakhir adalah pemeliharaan, yaitu perbaikan kesalahan dan penyempurnaan sistem agar dapat digunakan secara stabil dan berkelanjutan.

3. Hasil dan Diskusi

A. Analisa

1. Objek Penelitian

Pada penelitian ini, objek yang diteliti adalah destinasi wisata alam dan sejarah di Kabupaten Muna. Salah satu pantai yang paling terkenal adalah Pantai Meleura, yang terletak di Desa Lakarinta, Kecamatan Lohia. Pantai ini dikenal dengan pasir putih dan air laut yang jernih kebiruan. Selain itu, terdapat pula Pantai Walengkabola di Kecamatan Tongkuno. Selain wisata pantai, Kabupaten Muna juga memiliki objek wisata alam lainnya seperti Danau Ubur-Ubur dan Gua Liangkabori di Kecamatan Lohia, yang terkenal karena lukisan prasejarah di dinding guanya dan masih ada wisata lainnya.

Tabel 1 Objek Penelitian

No	Nama	Lokasi
1.	Mata Air Jompi	Lagasa, Kec. Duruka, Kabupaten Muna
2.	Gua Liangkabori	Desa Liang Kabori, Kecamatan Lohia, Kabupaten Muna
3.	Danau Napabale	Desa Lohia, Kecamatan Lohia, Kabupaten Muna
4.	Pantai Meleura	Desa Lakarinta Kecamatan Lohia, Kabupaten Muna
5.	Danau Motonuno	Desa Lakarinta, Kecamatan Lohia, Kabupaten Muna
6.	Permandian Topa	Desa Labone, Kecamatan Lasalepa, Kabupaten Muna
7.	Mata Air Sangia	Kabangka, Kec. Kabangka, Kabupaten Muna
8.	Cagar Alam Napabalano	Napabalano, Kec. Napabalano, Kabupaten Muna
9.	Permandian Fotuno Rete	Desa Wakumoro, Kecamatan Parigi, Kabupaten Muna
10.	Pantai Walengkabola	Desa Walengkabola, Kecamatan Tongkuno, Kabupaten Muna
11.	Permandian Moko	Kecamatan Tongkuno, Kabupaten Muna
12.	Pantai Morano	Oempu, Kecamatan Tongkuno, Kabupaten Muna
13.	Danau Moko 3	Kecamatan Tongkuno, Kabupaten Muna
14.	Danau Moko 4	Kecamatan Tongkuno, Kabupaten Muna
15.	Puncak Masalili	Masalili, Kec. Kontunaga, Kabupaten Muna
16.	Puncak Wakila	Desa Kondongia, Kecamatan Lohia, Kabupaten Muna
17.	Danau Ubur-ubur	Desa Lohia, Kec. Lohia. Kab Muna.
18.	Masjid Tua Kerajaan Muna	Kecamatan Tongkuno, Kabupaten Muna
19.	Museum Bharugano Wuna	Raha, Kabupaten Muna
20.	Benteng Wasidakari	Desa Kasaka, Kecamatan Kabawo, Kabupaten Muna

2. Data Penelitian

Pada penelitian ini, data yang digunakan sebagai bahan penelitian adalah nama-nama destinasi wisata yang terdapat di wilayah Kabupaten Muna. Data yang dikumpulkan mencakup informasi mengenai setiap lokasi wisata, informasi rute perjalanan yang dimulai dari titik keberangkatan yaitu Kota Raha menuju berbagai lokasi wisata di Kabupaten Muna untuk keperluan penggambaran graf manual, jarak antar lokasi wisata yang digunakan untuk melakukan proses pengolahan data pencarian rute terpendek secara manual, serta titik koordinat geografis masing-masing lokasi yang dimanfaatkan untuk menandai posisi titik wisata di dalam sistem aplikasi pencarian rute terpendek. Seluruh data penelitian ini diperoleh melalui sumber Google Maps untuk mengukur jarak antar lokasi dan mendapatkan koordinat yang akurat.

Tabel 2 Koordinat Destinasi Wisata

No	Node	Nama Lokasi	Latitude	Longitude
1.	A	Museum Bharugano Wuna	-4.8330239	122.7247377
2.	B	Mata Air Jompi	-4.8462631	122.7168882
3.	C	Permandian Topa	-4.728976	122.7240131
4.	D	Puncak Wakila	-4.8781497	122.7054898
5.	E	Cagar Alam Napabalano	-4.6356443	122.7099599
6.	F	Puncak Masalili	-4.8674165	122.6778225
7.	G	Gua Liangkabhor	-4.883172	122.6882663
8.	H	Danau Napabale	-4.9047713	122.750682
9.	I	Danau Ubur-ubur	-4.8962779	122.7572336
10.	J	Danau Motonuno	-4.9254288	122.750284
11.	K	Pantai Meleura	-4.9293544	122.7558587
12.	L	Mata Air Sangia	-4.9188693	122.4085274
13.	M	Masjid Tua Kerajaan Muna	-4.974158	122.609148
14.	N	Benteng Wasidakari	-5.0231035	122.5711475
15.	O	Permandian Fotuno Rete	-5.0776073	122.5068243
16.	P	Permandian Moko	-5.1742711	122.5660095
17.	Q	Danau Moko 3	-5.1696265	122.5829782
18.	R	Danau Moko 4	-5.1636628	122.5839921
19.	S	Pantai Morano	-5.1621944	122.5883596
20.	T	Pantai Walengkabola	-5.193644	122.5882665
21.	X	Kota Raha	-4.8323968	122.7108002

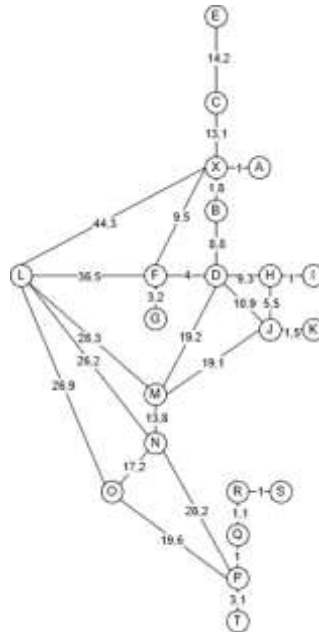
Pada Tabel 2 merupakan keterangan titik koordinat lokasi wisata yang terdapat di Kabupaten Muna dan digunakan sebagai data pendukung dalam penelitian ini. Titik koordinat setiap lokasi wisata tersebut berfungsi untuk memudahkan proses penandaan posisi masing-masing destinasi ke dalam sistem aplikasi pencarian rute terpendek berbasis algoritma Dijkstra, sehingga sistem dapat menampilkan letak geografis tiap objek wisata secara akurat di peta digital.

Tabel 3 Jarak Antar Setiap Node

No	Node Awal	Node Tujuan	Jarak (km)
1.	E	C	14,2
2.	C	X	13,1
3.	X	A	1,0
4.	X	B	1,8
5.	X	F	9,5
6.	X	L	44,3
7.	B	D	8,8
8.	D	F	4,0
9.	F	L	36,5
10	F	G	3,2
11.	D	H	9,3
12.	H	I	1,0
13.	H	J	5,5
14.	J	K	1,5
15.	D	J	10,9
16.	J	M	19,1
17.	D	M	19,2
18.	M	N	13,8
19.	N	O	17,2
20.	L	O	26,9
21.	L	N	26,2
22.	L	M	28,3
23.	O	P	19,6
24.	N	P	28,2
25.	P	Q	1,0

No	Node Awal	Node Tujuan	Jarak (km)
26.	Q	R	1,0
27.	R	S	1,0
28.	P	T	3,1

Pada Tabel 3 merupakan keterangan jarak antar setiap node dalam satuan kilometer, yang digunakan sebagai dasar dalam proses penggambaran graf jaringan wisata Kabupaten Muna. Data jarak antar node ini berfungsi untuk mempermudah proses pengolahan data serta menjadi acuan dalam perhitungan rute terpendek secara manual menggunakan Algoritma Dijkstra.



Gambar 1 Graf Rute Perjalanan wisata di Kabupaten Muna

Graf pada Gambar 1 dibuat berdasarkan data jarak antar setiap lokasi wisata yang telah ditampilkan pada Tabel 3. Setiap node pada graf mewakili satu titik lokasi wisata di Kabupaten Muna, sedangkan setiap sisi (edge) yang menghubungkan dua node menunjukkan adanya hubungan langsung antar lokasi dengan nilai jarak tertentu dalam satuan kilometer. Penyusunan graf ini bertujuan untuk memvisualisasikan jaringan hubungan antar lokasi wisata yang menjadi dasar dalam penerapan Algoritma Dijkstra. Dengan adanya graf ini, proses pencarian rute terpendek dapat dilakukan secara sistematis dengan mempertimbangkan seluruh jarak yang terhubung antar node.

Node X ditetapkan sebagai titik awal (node asal) yang merepresentasikan Kota Raha sebagai pusat aktivitas wisatawan. Dari titik ini, algoritma akan menghitung dan menentukan rute tercepat atau terpendek menuju setiap node lainnya berdasarkan bobot jarak terkecil.

3. Pengolahan Data Rute Perjalanan Secara Manual

Pada tahap ini dilakukan proses penentuan rute terpendek dari titik awal X (Kota Raha) menuju seluruh *node* tujuan menggunakan algoritma Dijkstra secara manual. Tujuan dari pengolahan data ini adalah untuk membuktikan bahwa hasil perhitungan sistem sesuai dengan prinsip dasar algoritma yang digunakan.

Algoritma Dijkstra bekerja dengan menentukan jarak terpendek antar *node* melalui proses pembaruan nilai bobot pada setiap iterasi. Pada setiap langkah, *node* dengan jarak sementara terkecil dari *node* awal akan dipilih menjadi *node* tetap (*permanent node*), kemudian digunakan untuk memperbaiki jarak *node* lainnya yang berdekatan (*adjacent nodes*). Proses ini terus berlanjut hingga seluruh *node* telah dikunjungi dan diperoleh nilai jarak minimum ke setiap *node* tujuan.

Langkah 1 Inisialisasi *Node* Awal

Pada tahap awal, *node* X (Kota Raha) dijadikan sebagai titik awal dengan nilai jarak 0 km, sedangkan seluruh *node* lainnya diberi nilai jarak tak hingga (∞). Berdasarkan data *graf*, *node* X memiliki lima *node* tetangga langsung, yaitu A, B, F, C, dan L, dengan jarak sebagai berikut:

- a. $X-A = 1,0$ km
- b. $X-B = 1,8$ km
- c. $X-F = 9,5$ km
- d. $X-C = 13,1$ km
- e. $X-L = 44,3$ km

Dari kelima *node* tersebut, *node* A memiliki jarak terkecil (1,0 km), sehingga A ditetapkan sebagai *node* tetap pertama.

Langkah 2 *Node* Aktif A

Node A hanya memiliki koneksi langsung dengan *node* X yang telah menjadi *node* tetap. Oleh karena itu, tidak terjadi pembaruan jarak pada langkah ini. *Node* berikutnya dengan jarak terkecil adalah B (1,8 km), sehingga B dijadikan *node* aktif berikutnya.

Langkah 3 *Node* Aktif B

Node B memiliki hubungan langsung dengan *node* D sejauh 8,8 km. Perhitungan jarak dilakukan sebagai berikut:

$$X-B-D = 1,8 + 8,8 = 10,6 \text{ km.}$$

Karena nilai tersebut lebih kecil dibandingkan jarak sebelumnya (∞), maka jarak ke *node* D diperbarui menjadi 10,6 km. *Node* berikutnya dengan jarak terkecil adalah F (9,5 km), sehingga F dipilih sebagai *node* aktif selanjutnya.

Langkah 4 *Node* Aktif F

Node F memiliki tiga tetangga, yaitu D, G, dan L. Hasil perhitungan adalah sebagai berikut:

- a. $X-F-D = 13,5$ km \rightarrow lebih besar dari 10,6 km (tidak diperbarui).
- b. $X-F-G = 12,7$ km \rightarrow jarak ke G diperbarui menjadi 12,7 km.
- c. $X-F-L = 46,0$ km \rightarrow lebih besar dari 44,3 km (tidak diperbarui).

Pada akhir langkah ini jarak minimum yang baru ditemukan adalah untuk *node* G (12,7 km). Namun, karena *node* D (10,6 km) memiliki jarak yang lebih kecil, maka D akan dijadikan *node* aktif berikutnya.

Langkah 5 *Node* Aktif D

Node D memiliki koneksi dengan *node* B, F, H, J, dan M. Berdasarkan hasil perhitungan:

- a. $X-B-D-H = 19,9$ km,
- b. $X-B-D-J = 21,5$ km,
- c. $X-B-D-M = 29,8$ km.

Jarak ke *node* H, J, dan M diperbarui sesuai hasil perhitungan di atas. *Node* dengan jarak terkecil berikutnya adalah G (12,7 km).

Langkah 6 *Node* Aktif G

Node G hanya memiliki satu hubungan langsung dengan *node* F yang telah dikunjungi. Karena tidak terdapat *node* baru yang bisa diperbarui dari G, maka tidak ada perubahan nilai jarak pada langkah ini. *Node* berikutnya yang memiliki jarak terkecil adalah C (13,1 km).

Langkah 7 *Node* Aktif C

Node C terhubung dengan *node* X dan *node* E. Perhitungan jarak menunjukkan bahwa jalur dari X ke E melalui C memiliki jarak $13,1 + 14,2 = 27,3$ km. Karena nilai ini lebih kecil dari ∞ , maka jarak ke *node* E diperbarui. Setelah semua tetangga C diperiksa, *node* aktif berikutnya adalah H (19,9 km) karena memiliki jarak terkecil di antara *node* yang belum dikunjungi.

Langkah 8 *Node* Aktif H

Node H memiliki hubungan dengan *node* I dan J.

- a. $X-B-D-H-I = 20,9$ km,
- b. $X-B-D-H-J = 25,4$ km (lebih besar dari 21,5 km, tidak diperbarui).

Maka jarak ke *node* I diperbarui menjadi 20,9 km.

Langkah 9 *Node* Aktif I

Node I hanya memiliki koneksi ke H, yang telah dikunjungi, sehingga tidak terjadi pembaruan jarak. Proses dilanjutkan ke *node* J (21,5 km).

Langkah 10 *Node* Aktif J

Node J memiliki dua hubungan langsung, yaitu ke *node* K dengan jarak 1,5 km dan ke *node* M dengan jarak 19,1 km.

- $X-B-D-J-K = 21,5 + 1,5 = 23,0$ km,
- $X-B-D-J-M = 40,6$ km (lebih besar dari 29,8 km, tidak diperbarui).

Hasil perhitungan ini memperbarui jarak ke *node* K (23,0 km). *Node* dengan jarak terkecil berikutnya adalah M (29,8 km).

Langkah 11 *Node* Aktif M

Node M memiliki hubungan dengan N (13,8 km) dan L (28,3 km).

- Jalur $X-B-D-M-N = 43,6$ km,
- Jalur $X-B-D-M-L = 58,1$ km (lebih besar dari 44,3 km, tidak diperbarui).

Maka jarak ke *node* N diperbarui menjadi 43,6 km. *Node* dengan jarak terkecil berikutnya adalah E (27,3 km).

Langkah 12 *Node* Aktif E

Node E hanya terhubung dengan C, yang telah diproses, sehingga tidak terjadi pembaruan jarak. *Node* berikutnya yang akan diproses adalah N (43,6 km).

Langkah 13 *Node* Aktif N

Node N memiliki tetangga M, L, O, dan P.

- Jalur $X-B-D-M-N-O = 43,6 + 17,2 = 60,8$ km,
- Jalur $X-B-D-M-N-P = 43,6 + 28,2 = 71,8$ km.

Jarak ke *node* P diperbarui menjadi 71,8 km.

Langkah 14 *Node* Aktif O

Node O memiliki hubungan dengan L dan P, namun jalur melalui O menghasilkan nilai yang lebih besar daripada jarak sebelumnya. Oleh karena itu, tidak terjadi pembaruan jarak pada langkah ini.

Langkah 15 *Node* Aktif P

Node P memiliki hubungan dengan Q dan T.

- Jalur $X-B-D-M-N-P-Q = 71,8 + 1,0 = 72,8$ km.
- Jalur $X-B-D-M-N-P-T = 71,8 + 3,1 = 74,9$ km.

Jarak ke *node* Q dan T diperbarui menjadi masing-masing 72,8 km dan 74,9 km.

Langkah 16 *Node* Aktif Q

Node Q memiliki hubungan dengan *node* R sejauh 1,0 km.

$X-B-D-M-N-P-Q-R = 72,8 + 1,0 = 73,8$ km, sehingga jarak ke R diperbarui menjadi 73,8 km.

Langkah 17 *Node* Aktif R

Node R memiliki hubungan dengan *node* S sejauh 1,0 km.

$X-B-D-M-N-P-Q-R-S = 73,8 + 1,0 = 74,8$ km, sehingga jarak ke *node* S diperbarui menjadi 74,8 km.

Langkah 18 *Node* Aktif S

Node S hanya terhubung dengan R yang telah dikunjungi, sehingga tidak terjadi pembaruan.

Langkah 19 *Node* T

Node T hanya terhubung dengan P yang telah dikunjungi, sehingga tidak terjadi pembaruan. Pada tahap ini seluruh *node* telah dikunjungi dan jarak terpendek dari *node* X ke setiap *node* telah diperoleh.

Tabel 4 Hasil Perhitungan Manual Rute Terpendek

<i>Node</i>	Jarak Terpendek (km)	Jalur Terpendek
A	1.0	X-A
B	1.8	X-B
F	9.5	X-F
D	10.6	X-B-D
G	12.7	X-F-G
C	13.1	X-C
H	19.9	X-B-D-H
I	20.9	X-B-D-H-I
J	21.5	X-B-D-J
K	23.0	X-B-D-J-K
E	27.3	X-C-E
M	29.8	X-B-D-M
N	43.6	X-B-D-M-N
L	44.3	X-L
O	60.8	X-B-D-M-N-O
P	71,8	X-B-D-M-N-P
Q	72,8	X-B-D-M-N-P-Q
R	73,8	X-B-D-M-N-P-Q-R
S	74,8	X-B-D-M-N-P-Q-R-S
T	74,9	X-B-D-M-N-P-T

B. Perancangan Sistem

Perancangan sistem merupakan tahap penting dalam proses pengembangan aplikasi karena berfungsi sebagai kerangka dasar yang menggambarkan bagaimana sistem akan bekerja sebelum tahap implementasi dilakukan. Pada penelitian ini, perancangan sistem difokuskan pada pembangunan aplikasi berbasis web yang mampu menentukan rute terpendek menuju lokasi wisata di Kabupaten Muna menggunakan algoritma Dijkstra. Tahapan perancangan sistem meliputi pembuatan model interaksi pengguna dengan sistem melalui use case diagram, perancangan alur proses sistem menggunakan flowchart, perancangan antarmuka pengguna (user interface), serta perancangan basis data yang mendukung proses komputasi rute terpendek.

1. Use Case Diagram

Use case diagram merupakan salah satu diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan hubungan interaksi antara pengguna dengan sistem. Diagram ini menunjukkan fungsi-fungsi utama yang dapat dilakukan oleh pengguna serta bagaimana sistem merespons setiap aktivitas tersebut. Dengan adanya use case diagram, pengembang dapat memahami kebutuhan sistem secara lebih jelas serta menentukan fitur yang akan disediakan bagi setiap jenis pengguna.

Pada sistem pencarian rute terpendek wisata di Kabupaten Muna, terdapat dua aktor utama yang berinteraksi dengan sistem, yaitu User dan Admin. Kedua aktor tersebut memiliki peran dan hak akses yang berbeda sesuai dengan kebutuhan operasional sistem.

User merupakan pengguna umum yang memanfaatkan sistem untuk memperoleh informasi wisata serta menentukan rute perjalanan yang paling efisien. User dapat mengakses berbagai fitur utama seperti halaman beranda, peta lokasi wisata, pencarian rute terpendek antar destinasi, melihat informasi detail lokasi wisata, serta mengakses halaman about yang berisi informasi umum mengenai aplikasi.

Sementara itu, Admin memiliki hak akses penuh terhadap sistem dan bertanggung jawab dalam pengelolaan data. Admin harus melakukan proses login terlebih dahulu sebelum dapat mengakses fitur manajemen data. Setelah berhasil masuk ke dalam sistem, admin dapat mengelola data node yang merepresentasikan titik lokasi pada graf, mengelola informasi wisata seperti nama lokasi, deskripsi, gambar, serta koordinat geografis, dan memperbarui konten halaman about. Setelah selesai melakukan pengelolaan data, admin dapat keluar dari sistem melalui fitur logout.

Secara keseluruhan, use case pada sistem ini terdiri dari beberapa fungsi utama, yaitu login admin, menampilkan halaman beranda, menampilkan peta lokasi wisata, melakukan pencarian rute terpendek menggunakan algoritma Dijkstra, menampilkan informasi detail lokasi wisata, menampilkan halaman about, mengelola data node,

mengelola data informasi wisata, mengelola halaman about, serta proses logout. Setiap fungsi tersebut dirancang untuk mendukung proses pencarian rute wisata secara efisien sekaligus memastikan pengelolaan data sistem dapat dilakukan dengan baik oleh administrator.

2. Flowchart Sistem Pencarian Rute

Flowchart sistem digunakan untuk menggambarkan alur kerja sistem dalam menentukan rute terpendek menuju lokasi wisata yang diinginkan oleh pengguna. Diagram ini menjelaskan tahapan proses secara berurutan mulai dari pengguna mengakses sistem hingga sistem menghasilkan rute perjalanan yang optimal.

Proses dimulai ketika pengguna membuka website dan sistem menampilkan menu utama aplikasi. Pada tahap ini, pengguna diminta untuk menentukan lokasi awal sebagai titik awal pencarian rute. Setelah itu, pengguna menentukan titik awal dan lokasi tujuan melalui peta yang tersedia dengan memanfaatkan fitur pemilihan titik pada peta.

Apabila proses pemilihan titik berhasil dilakukan, sistem kemudian menjalankan algoritma Dijkstra untuk menghitung jarak terpendek antara lokasi awal dan tujuan. Namun jika terjadi kesalahan dalam penentuan titik lokasi, sistem akan mengarahkan pengguna untuk mengulangi proses input hingga data yang dimasukkan valid. Setelah perhitungan algoritma selesai dilakukan, sistem menampilkan hasil berupa lintasan rute terpendek lengkap dengan informasi jarak tempuhnya. Proses pencarian rute kemudian berakhir ketika hasil rute berhasil ditampilkan kepada pengguna.

3. Flowchart Login Admin

Flowchart login admin menggambarkan proses autentikasi yang harus dilakukan oleh administrator sebelum dapat mengakses fitur pengelolaan data pada sistem. Proses ini dimulai ketika admin membuka halaman login pada website.

Pada halaman tersebut, admin diminta untuk memasukkan username dan password pada form yang telah disediakan. Data yang dimasukkan kemudian diverifikasi oleh sistem dengan cara mencocokkannya dengan data admin yang tersimpan dalam basis data. Apabila data yang dimasukkan sesuai, maka sistem akan memberikan akses kepada admin untuk masuk ke halaman utama administrator yang berisi berbagai fitur pengelolaan data.

Sebaliknya, apabila username atau password yang dimasukkan tidak sesuai, sistem akan menampilkan pesan kesalahan dan meminta admin untuk melakukan penginputan ulang hingga data yang dimasukkan benar. Mekanisme ini bertujuan untuk menjaga keamanan sistem sekaligus memastikan bahwa hanya pengguna yang memiliki hak akses yang dapat mengelola data aplikasi.

4. Flowchart Pengelolaan Data Informasi Wisata

Pengelolaan data informasi wisata dilakukan oleh admin untuk memastikan bahwa data destinasi wisata yang tersedia dalam sistem selalu akurat dan diperbarui secara berkala. Dalam proses ini, admin dapat melakukan tiga jenis operasi utama terhadap data, yaitu insert, update, dan delete.

Pada proses insert, sistem menampilkan form input yang memungkinkan admin memasukkan data baru berupa nama lokasi wisata, deskripsi tempat wisata, titik koordinat lokasi, serta gambar yang merepresentasikan destinasi tersebut. Setelah seluruh data diisi dengan benar, admin menekan tombol simpan untuk menyimpan data tersebut ke dalam database.

Pada proses update, admin dapat memperbarui data wisata yang sudah ada dengan cara memilih data yang ingin diubah, kemudian melakukan pengeditan pada informasi yang diperlukan seperti nama lokasi, deskripsi, koordinat, atau gambar. Setelah proses perubahan selesai dilakukan, data disimpan kembali ke dalam basis data.

Sedangkan pada proses delete, admin terlebih dahulu memilih data wisata yang akan dihapus dari sistem. Setelah data dipilih, sistem akan menghapus data tersebut dari database. Setiap proses penyimpanan, pembaruan, maupun penghapusan data akan diikuti dengan pesan konfirmasi sebagai indikator bahwa proses pengelolaan data berhasil dilakukan.

5. Flowchart Pengelolaan Data Node

Node merupakan representasi titik lokasi dalam graf yang digunakan oleh algoritma Dijkstra untuk menghitung rute terpendek. Oleh karena itu, pengelolaan data node menjadi bagian penting dalam sistem ini.

Proses pengelolaan data node juga meliputi operasi insert, update, dan delete. Pada proses insert, admin memasukkan data node baru berupa nama node serta koordinat lokasi yang akan disimpan dalam database. Pada

proses update, admin dapat memperbarui data node yang sudah ada dengan cara memilih node tertentu kemudian melakukan perubahan pada informasi yang diperlukan.

Sementara itu, pada proses delete, admin memilih data node yang akan dihapus dari sistem. Setelah data dipilih, sistem akan menghapus node tersebut dari basis data. Setiap perubahan data yang berhasil dilakukan akan ditampilkan dalam bentuk pesan konfirmasi sebagai tanda bahwa proses pengelolaan data telah selesai dilakukan.

6. Flowchart Pengelolaan Data Admin

Selain mengelola data wisata dan node, admin juga dapat mengelola data akun administrator yang memiliki hak akses terhadap sistem. Pengelolaan data admin meliputi penambahan akun baru, pembaruan data akun, serta penghapusan akun yang tidak lagi digunakan.

Pada proses penambahan akun, admin memasukkan username dan password baru yang kemudian disimpan ke dalam basis data. Pada proses pembaruan data, admin dapat mengubah informasi akun yang sudah ada, misalnya mengganti username atau password. Sedangkan pada proses penghapusan akun, admin memilih data admin yang ingin dihapus dari sistem sehingga akun tersebut tidak lagi memiliki akses terhadap aplikasi.

Setiap perubahan yang dilakukan akan disimpan ke dalam database dan sistem akan menampilkan pesan konfirmasi sebagai indikator bahwa proses pengelolaan data berhasil dilakukan.

7. Perancangan User Interface

Perancangan antarmuka pengguna dilakukan untuk mempermudah interaksi antara pengguna dengan sistem. Antarmuka yang dirancang harus bersifat sederhana, informatif, serta mudah digunakan baik oleh pengguna umum maupun administrator.

Pada halaman utama website ditampilkan nama aplikasi serta beberapa menu navigasi seperti beranda, daftar wisata, pencarian rute, halaman tentang aplikasi, dan login admin. Menu pencarian rute menjadi fitur utama yang memungkinkan pengguna menentukan lintasan terpendek menuju lokasi wisata yang diinginkan.

Halaman login admin dirancang dengan tampilan sederhana yang terdiri dari kolom input username dan password serta tombol login. Antarmuka ini berfungsi sebagai mekanisme autentikasi sebelum admin dapat mengakses fitur pengelolaan data.

Setelah berhasil login, admin akan diarahkan ke halaman utama administrator yang berisi berbagai menu pengelolaan data seperti data wisata, data node, data admin, data jarak, serta pengelolaan halaman about. Struktur tampilan halaman admin terdiri dari header, sidebar, area konten utama, dan footer yang dirancang agar proses manajemen data dapat dilakukan secara efisien.

8. Perancangan Basis Data

Basis data merupakan komponen penting dalam sistem karena berfungsi sebagai tempat penyimpanan seluruh data yang digunakan dalam proses pencarian rute terpendek. Sistem ini menggunakan database MySQL karena bersifat open source, mudah diintegrasikan dengan bahasa pemrograman PHP, serta memiliki kemampuan pengelolaan data yang baik.

Struktur basis data terdiri dari beberapa tabel utama yaitu tabel admin, tabel wisata, tabel node, tabel jarak, dan tabel about. Tabel admin digunakan untuk menyimpan data akun administrator yang memiliki hak akses terhadap sistem. Tabel wisata menyimpan informasi mengenai destinasi wisata seperti nama lokasi, deskripsi, koordinat geografis, serta gambar tempat wisata.

Tabel node berfungsi untuk menyimpan data titik lokasi yang digunakan sebagai simpul dalam graf jaringan. Sementara itu, tabel jarak menyimpan informasi jarak antar node yang saling terhubung dan berperan sebagai sisi (edge) dalam graf yang akan diproses oleh algoritma Dijkstra.

Hubungan antar tabel dirancang sedemikian rupa agar mendukung proses perhitungan rute terpendek. Tabel node memiliki hubungan one-to-many dengan tabel jarak karena satu node dapat terhubung dengan beberapa node lainnya. Tabel jarak memiliki dua foreign key yaitu `id_node_asal` dan `id_node_tujuan` yang merujuk ke tabel node. Selain itu, tabel wisata memiliki hubungan one-to-one dengan tabel node karena setiap lokasi wisata direpresentasikan oleh satu node dalam graf jaringan.

Melalui perancangan sistem yang terstruktur ini, aplikasi pencarian rute terpendek wisata di Kabupaten Muna diharapkan mampu memberikan solusi yang efektif bagi wisatawan dalam menentukan jalur perjalanan yang optimal. Desain sistem yang mencakup pemodelan interaksi pengguna, alur proses sistem, perancangan antarmuka,

serta struktur basis data yang terorganisasi dengan baik menjadi dasar penting dalam keberhasilan implementasi algoritma Dijkstra pada sistem tersebut.

C. Implementasi Algoritma Dijkstra

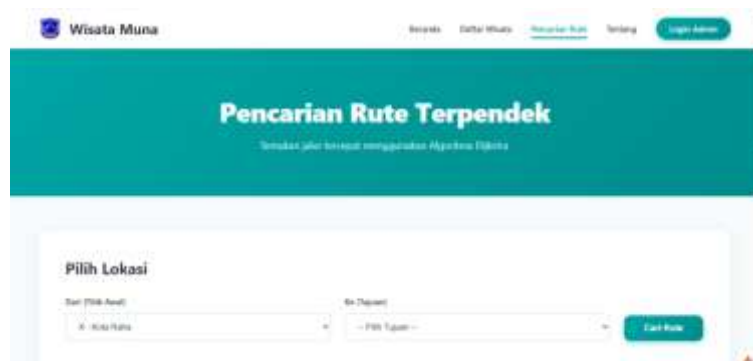
Implementasi Algoritma Dijkstra dalam sistem penentuan rute terpendek wisata Kabupaten Muna merupakan bagian fundamental yang menghubungkan data spasial (*node* dan *edge*) dengan proses komputasi jalur optimal yang dihasilkan secara otomatis oleh aplikasi. Algoritma Dijkstra diintegrasikan secara penuh ke dalam arsitektur sistem berbasis CodeIgniter 4, sehingga mampu memberikan hasil perhitungan yang dapat divisualisasikan melalui peta interaktif Leaflet.js.

Penerapan Algoritma Dijkstra tidak hanya berfokus pada aspek perhitungan matematis, tetapi juga memperhatikan bagaimana algoritma tersebut berperan dalam aliran kerja sistem secara keseluruhan, mulai dari interaksi pengguna, pengolahan data, pemilihan jalur, hingga penyajian hasil.

1. Antarmuka Input Pencarian Rute

Implementasi dimulai dari antarmuka pencarian rute yang dirancang sebagai titik awal interaksi antara pengguna dan sistem. Pada halaman ini, pengguna dapat memilih titik awal dan titik tujuan dari daftar lokasi wisata yang tersedia. Setiap lokasi direpresentasikan melalui kode *node* dan nama lokasi, sehingga memudahkan pengguna melakukan pemilihan.

Antarmuka pencarian ini tidak hanya berfungsi sebagai form input sederhana, namun menjadi *trigger* utama yang mengaktifkan seluruh rangkaian proses komputasi rute. Ketika pengguna menekan tombol “Cari Rute”, sistem melakukan validasi input dan meneruskan permintaan tersebut ke *controller*, tempat Algoritma Dijkstra akan dijalankan.



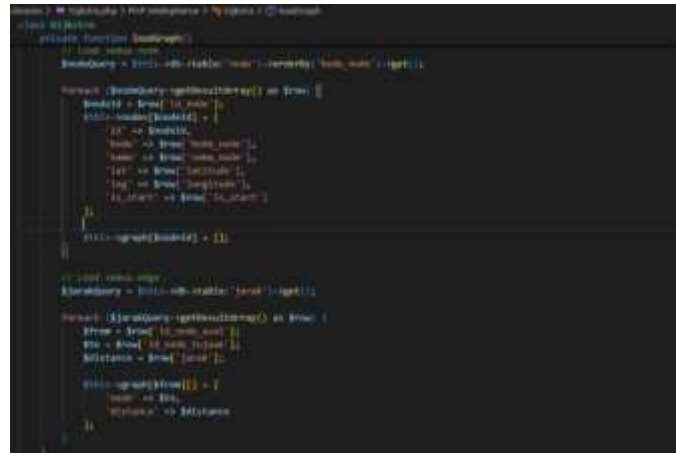
Gambar 21 Antarmuka Form Pencarian Rute Terpendek

Pada gambar 2, antarmuka dirancang intuitif dan ramah pengguna, serta memastikan bahwa proses penentuan rute dapat dimulai dengan mudah tanpa memerlukan pengetahuan teknis dari pengguna.

2. Pembentukan Graf dan Inisialisasi Algoritma

Tahap berikutnya adalah konversi data mentah dari *database* menjadi struktur graf yang dapat diproses oleh Algoritma Dijkstra. Sistem memuat seluruh *node*, termasuk koordinat geografisnya, kemudian memuat data jarak antar *node* dari tabel jarak. Data tersebut diolah menjadi *adjacency list*, yaitu struktur data yang paling umum digunakan untuk merepresentasikan graf berarah dan berbobot. Menurut Kenap dan Moningkey (2024), penerapan algoritma graf dapat digunakan untuk menentukan rute terpendek pada sistem navigasi digital.

Proses ini sangat penting karena kualitas graf menentukan ketepatan hasil yang dikeluarkan oleh algoritma. Setiap *node* dihubungkan dengan *node* tujuan berikut bobot jaraknya. Sistem tidak hanya menyimpan hubungan antar lokasi, tetapi juga mampu memperhitungkan rute-rute alternatif berdasarkan jarak yang tersedia. Inisialisasi nilai *distance*, *visited*, dan *previous* dilakukan secara sistematis. Setiap *node* diberikan nilai awal jarak ∞ (tak hingga), kecuali *node* awal yang diset menjadi 0. Tahapan ini sepenuhnya mengikuti teori algoritma Dijkstra, sehingga memastikan algoritma berjalan sesuai prinsip dasarnya.



```
function graphFromDatabase() {
  // Mendapatkan data dari database
  const query = `SELECT * FROM nodes`;
  const result = db.query(query);

  // Membuat array untuk menyimpan node
  const nodes = [];

  // Menjalankan query ke database
  db.query(query, (err, rows) => {
    if (err) {
      console.error('Error fetching nodes:', err);
    } else {
      // Menjalankan loop untuk setiap baris data
      rows.forEach((row) => {
        // Membuat objek node
        const node = {
          id: row.id,
          name: row.name,
          x: row.x,
          y: row.y,
          weight: row.weight,
        };
        // Menambahkan node ke array
        nodes.push(node);
      });
    }
  });

  // Mengembalikan array node
  return nodes;
}

function graphFromEdges() {
  // Mendapatkan data dari database
  const query = `SELECT * FROM edges`;
  const result = db.query(query);

  // Membuat array untuk menyimpan edge
  const edges = [];

  // Menjalankan query ke database
  db.query(query, (err, rows) => {
    if (err) {
      console.error('Error fetching edges:', err);
    } else {
      // Menjalankan loop untuk setiap baris data
      rows.forEach((row) => {
        // Membuat objek edge
        const edge = {
          source: row.source,
          target: row.target,
          weight: row.weight,
        };
        // Menambahkan edge ke array
        edges.push(edge);
      });
    }
  });

  // Mengembalikan array edge
  return edges;
}
```

Gambar 3 Kode Pembentukan Graf dari Database

Potongan kode pada Gambar 3 menunjukkan proses pembentukan graf dari database, di mana sistem memuat seluruh node beserta koordinat geografisnya, lalu membentuk adjacency list berdasarkan tabel jarak. Setiap hubungan antar node direpresentasikan sebagai sebuah edge berbobot.



```
function initializeDijkstra() {
  // Inisialisasi variabel-variabel algoritma Dijkstra
  const nodes = graphFromDatabase();
  const edges = graphFromEdges();

  // Membuat array untuk menyimpan jarak
  const distance = new Array(nodes.length).fill(Infinity);
  // Membuat array untuk menyimpan node sebelumnya
  const previous = new Array(nodes.length).fill(-1);
  // Membuat array untuk menyimpan node yang sudah dikunjungi
  const visited = new Array(nodes.length).fill(false);

  // Menentukan node awal
  const startNode = nodes[0].id;

  // Menjalankan algoritma Dijkstra
  dijkstra(nodes, edges, distance, previous, visited, startNode);
}
```

Gambar 42 Kode Inisialisasi Awal Algoritma Dijkstra

Potongan kode pada Gambar 4 menunjukkan inisialisasi variabel algoritma Dijkstra, seperti distance, previous, dan visited, dengan memberikan nilai awal ∞ untuk seluruh node kecuali node awal yang diberi nilai 0. Tahapan ini merupakan fondasi utama yang memungkinkan algoritma menelusuri graf dan menghitung rute terpendek secara akurat.

3. Proses Perhitungan Jalur Terpendek

Inti dari implementasi ini terletak pada mekanisme kerja algoritma Dijkstra dalam menentukan rute dengan bobot paling rendah. Sistem melakukan proses perhitungan berikut:

- a. Pemilihan node dengan jarak minimum
Pada setiap iterasi, sistem memilih node yang memiliki jarak terkecil dan belum dikunjungi. Pemilihan dilakukan menggunakan metode scanning melalui struktur queue internal.
- b. Relaksasi jarak
Untuk setiap node tetangga, algoritma menghitung jarak alternatif. Jika jarak baru lebih kecil dari jarak sebelumnya, nilai tersebut diperbarui. Proses ini merupakan inti dari mekanisme pencarian jalur optimal.
- c. Pencatatan node predecessor
Sistem menyimpan node sebelumnya (predecessor) untuk keperluan rekonstruksi jalur setelah proses perhitungan selesai.
- d. Pemberhentian proses
Proses dihentikan ketika node tujuan telah diproses atau seluruh node telah diperiksa.



Gambar 53 Potongan Kode Proses Inti Algoritma Dijkstra

Potongan kode pada Gambar 5 menunjukkan inti eksekusi Algoritma Dijkstra. Pada setiap iterasi, sistem memilih *node* dengan jarak minimum dari struktur queue, kemudian melakukan relaksasi jarak ke setiap *node* tetangga. Bila ditemukan jarak yang lebih optimal, sistem memperbarui nilai jarak dan menyimpan predecessor untuk keperluan rekonstruksi jalur. Proses ini terus berulang hingga *node* tujuan ditemukan atau seluruh *node* selesai diproses.

4. Rekonstruksi Jalur Terpendek

Setelah perhitungan jarak selesai, algoritma melanjutkan ke tahap rekonstruksi jalur. Pada tahap ini, sistem menelusuri array previous dari *node* tujuan sampai *node* awal, kemudian mengurutkan kembali hasilnya. Proses ini menghasilkan jalur final yang ditampilkan kepada pengguna dalam bentuk kode *node*, nama lokasi, dan rincian jarak. Rekonstruksi ini menjamin bahwa pengguna memperoleh jalur yang tidak hanya optimal secara matematis, tetapi juga runtut secara logis.



Gambar 64 Potongan Kode Rekonstruksi Jalur Terpendek

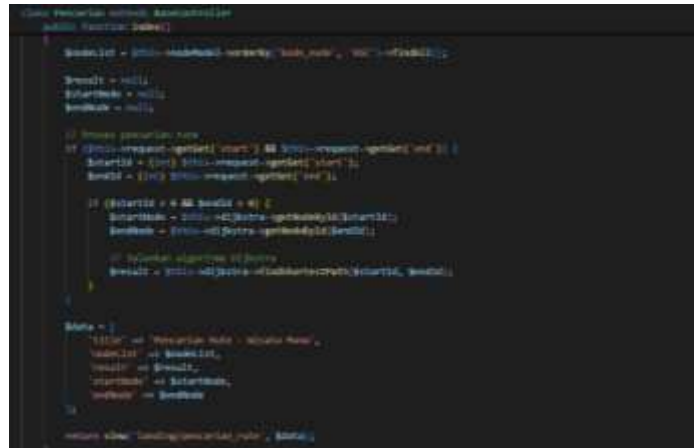
Potongan kode pada Gambar 6 menunjukkan tahap rekonstruksi jalur terpendek setelah proses relaksasi dan perhitungan jarak selesai dilakukan. Sistem menelusuri array previous mulai dari *node* tujuan menuju *node* awal. Setiap *node* yang dilewati disisipkan ke dalam array path menggunakan array_unshift(), sehingga urutan jalur terbentuk mulai dari titik awal hingga titik tujuan.

Tahap ini memastikan bahwa jalur yang dihasilkan tidak hanya merupakan jalur dengan bobot minimum, tetapi juga tersusun secara logis sesuai lintasan sebenarnya. Hasil rekonstruksi ini kemudian digunakan untuk menampilkan:

- Urutan *node* dalam bentuk kode.
- Nama lokasi wisata.
- Jarak per segmen.
- Serta visualisasi rute pada peta Leaflet.

5. Integrasi Algoritma dalam Controller

Integrasi algoritma dalam *controller* memastikan algoritma berjalan secara otomatis begitu pengguna melakukan permintaan pencarian rute. *Controller* memuat *node* awal dan *node* tujuan, memanggil fungsi findShortestPath(), lalu mengirimkan hasilnya ke *view*.



Gambar 7 Integrasi Algoritma Dijkstra dalam *Controller*

Potongan kode pada Gambar 7 menunjukkan proses integrasi antara algoritma Dijkstra dengan *controller* pada *framework* CodeIgniter 4. *Controller* bertugas:

- Menerima input dari pengguna, yaitu titik awal dan titik tujuan pencarian rute.
- Melakukan validasi input, memastikan nilai *node* yang dipilih valid.
- Memanggil fungsi `findShortestPath()`, yang menjadi inti dari eksekusi algoritma Dijkstra.
- Mengambil detail *node* awal dan *node* tujuan untuk ditampilkan di antarmuka pengguna.
- Mengirimkan hasil perhitungan ke *view*, termasuk jarak terpendek, waktu tempuh, dan urutan *node* yang dilalui.

Integrasi ini memastikan bahwa algoritma berjalan otomatis setiap kali pengguna melakukan pencarian rute. Selain itu, implementasi ini juga menegaskan hubungan erat antara komponen *MVC*:

- Model digunakan untuk memuat data *node*.
- Library* Dijkstra digunakan sebagai komponen logika algoritmik.
- Controller* bertindak sebagai penghubung antara frontend dan backend.
- View* menerima data jalur untuk divisualisasikan.

6. Visualisasi Hasil pada Peta Interaktif

Hasil implementasi Dijkstra tidak hanya ditampilkan dalam bentuk tabel, tetapi juga divisualisasikan secara geografis melalui peta digital. Visualisasi ini memanfaatkan *library* Leaflet.js yang mampu menampilkan marker pada setiap *node* yang dilalui, serta polyline rute berwarna biru yang menghubungkan seluruh *node* hasil perhitungan.

Visualisasi ini memberikan pemahaman spasial yang lebih jelas kepada pengguna, terutama wisatawan yang ingin mengetahui jalur perjalanan secara geografis. Selain itu, sistem juga menampilkan jarak antar segmen pada peta sehingga pengguna dapat mengetahui jarak setiap perpindahan *node*.



Gambar 85 Visualisasi Rute Terpendek Menggunakan Leaflet.js

Pada Gambar 8 menampilkan hasil visualisasi rute terpendek yang dihitung menggunakan Algoritma Dijkstra dan divisualisasikan melalui peta interaktif Leaflet.js. Pada tampilan ini, sistem menggambarkan jalur perjalanan

dalam bentuk garis biru (polyline) yang menghubungkan seluruh *node* yang harus dilalui berdasarkan perhitungan jarak minimum.

Setiap *node* pada rute ditandai menggunakan marker berwarna khusus, di mana marker hijau menandakan titik awal (*start*), marker orange menandakan titik akhir (*finish*), dan marker hijau menunjukkan *node-node* yang dilalui di tengah perjalanan. Selain itu, jarak antar segmen ditampilkan dalam bentuk label kecil pada peta, sehingga pengguna dapat mengetahui jarak setiap perpindahan antar dua *node*.

Visualisasi ini memberikan gambaran spasial yang jelas mengenai rute perjalanan wisata di Kabupaten Muna. Adanya tampilan peta interaktif, pengguna tidak hanya memperoleh informasi numerik berupa jarak dan waktu tempuh, tetapi juga dapat melihat posisi geografis dari jalur perjalanan yang direkomendasikan oleh sistem.

D. Implementasi Antarmuka Sistem

Implementasi antarmuka sistem pada aplikasi pencarian rute terpendek wisata Kabupaten Muna bertujuan memastikan seluruh fitur sistem dapat digunakan secara efektif oleh pengguna umum maupun administrator. Antarmuka dirancang dengan prinsip konsistensi tampilan, kemudahan navigasi, serta responsivitas agar proses eksplorasi destinasi wisata, pencarian rute, hingga pengelolaan data dapat dilakukan secara efisien. Desain ini juga mempertimbangkan aspek pengalaman pengguna (*user experience*) sehingga sistem dapat dipahami oleh pengguna tanpa memerlukan pengetahuan teknis khusus.

Halaman pertama yang ditampilkan ketika pengguna membuka aplikasi adalah Splash Screen. Halaman ini berfungsi sebagai tampilan pengenalan yang menampilkan identitas sistem “Jelajahi Muna” beserta tagline pariwisata daerah. Pada halaman ini juga tersedia fitur pencarian cepat yang memungkinkan pengguna mencari destinasi wisata secara langsung. Selain itu, terdapat tombol navigasi seperti “Lewati” dan “Mulai Jelajah” yang memberikan akses cepat menuju halaman utama sistem. Keberadaan elemen visual serta *call-to-action* pada splash screen membantu meningkatkan ketertarikan pengguna sekaligus memberikan gambaran awal mengenai fungsi utama aplikasi.

Setelah melewati splash screen, pengguna diarahkan ke Landing Page yang berfungsi sebagai pusat navigasi utama sistem. Halaman ini menampilkan header berisi identitas aplikasi serta menu navigasi seperti Beranda, Daftar Wisata, Pencarian Rute, Tentang, dan Login Admin. Bagian utama halaman menampilkan hero section dengan gambar destinasi wisata Kabupaten Muna yang dilengkapi judul dan deskripsi singkat mengenai penggunaan algoritma Dijkstra untuk menentukan rute terpendek. Selain itu, halaman ini juga menampilkan statistik sistem yang menunjukkan jumlah destinasi wisata, *node* lokasi, dan rute yang tersedia dalam basis data. Informasi tersebut memberikan gambaran awal kepada pengguna mengenai cakupan data yang dimiliki sistem serta meningkatkan kredibilitas aplikasi sebagai sumber informasi wisata.

Pada bagian selanjutnya terdapat modul pencarian rute terpendek, yang merupakan fitur utama aplikasi. Modul ini menyediakan pilihan lokasi awal dan lokasi tujuan melalui menu dropdown. Setelah pengguna menentukan titik awal dan tujuan, sistem akan memproses permintaan menggunakan layanan routing dan algoritma pencarian jalur terpendek. Hasil perhitungan kemudian ditampilkan dalam bentuk informasi jarak dan jalur perjalanan yang optimal. Integrasi antara antarmuka pengguna dan proses komputasi ini memungkinkan pengguna memperoleh rekomendasi rute perjalanan secara cepat dan akurat.

Selain itu, halaman utama juga dilengkapi peta wisata interaktif berbasis Leaflet.js yang menampilkan seluruh lokasi wisata dalam bentuk marker pada peta digital. Pengguna dapat mengklik marker untuk melihat informasi singkat mengenai destinasi yang dipilih. Visualisasi spasial ini memberikan pemahaman geografis yang lebih jelas mengenai persebaran lokasi wisata serta membantu pengguna merencanakan perjalanan secara lebih efektif.

Fitur eksplorasi destinasi juga disediakan melalui Halaman Daftar Wisata. Halaman ini menampilkan seluruh objek wisata dalam bentuk katalog visual menggunakan desain kartu (*card*). Setiap kartu memuat gambar destinasi, nama wisata, deskripsi singkat, serta koordinat lokasi. Desain katalog ini disusun menggunakan sistem grid responsif sehingga tampilan dapat menyesuaikan berbagai ukuran layar perangkat. Selain itu, terdapat fitur pencarian yang memungkinkan pengguna menemukan destinasi wisata tertentu secara cepat. Data yang ditampilkan berasal dari basis data MySQL yang diolah melalui framework CodeIgniter sebelum ditampilkan pada halaman web.

Fitur inti sistem yaitu halaman pencarian rute terpendek menampilkan form input untuk menentukan titik awal dan tujuan perjalanan. Secara default, titik awal ditetapkan pada Kota Raha, namun pengguna dapat memilih lokasi lain sesuai kebutuhan. Setelah proses pencarian dilakukan, sistem menampilkan hasil dalam dua bentuk, yaitu informasi tekstual dan visualisasi peta. Informasi tekstual meliputi urutan *node* yang dilalui, jarak antar segmen, total jarak perjalanan, serta estimasi waktu tempuh. Sementara itu, visualisasi peta menampilkan jalur perjalanan

dalam bentuk polyline serta marker yang menandai titik awal, tujuan, dan node perantara sehingga pengguna dapat memahami rute perjalanan secara spasial.

Selain antarmuka pengguna umum, sistem juga menyediakan halaman administrator untuk pengelolaan data. Halaman login admin digunakan sebagai mekanisme autentikasi agar hanya pengguna yang berwenang yang dapat mengakses sistem pengelolaan. Setelah berhasil login, administrator akan diarahkan ke halaman dashboard yang menampilkan ringkasan data sistem seperti jumlah destinasi wisata, jumlah node, koneksi jarak, serta jumlah akun admin.

Melalui dashboard tersebut, administrator dapat mengelola berbagai data penting seperti data wisata, node lokasi, jarak antar node, akun admin, serta konten informasi pada halaman about. Pengelolaan ini dilakukan melalui antarmuka berbasis tabel yang dilengkapi fitur tambah, ubah, dan hapus data. Dengan adanya fitur manajemen ini, sistem dapat diperbarui secara berkala sehingga informasi yang ditampilkan kepada pengguna tetap akurat dan relevan.

Secara keseluruhan, implementasi antarmuka sistem dalam aplikasi ini menunjukkan integrasi yang baik antara desain antarmuka, pengolahan data, serta algoritma pencarian rute. Antarmuka yang informatif dan mudah digunakan membantu pengguna dalam menemukan destinasi wisata serta menentukan rute perjalanan yang optimal di Kabupaten Muna.

E. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa aplikasi pencarian rute terpendek lokasi wisata di Kabupaten Muna berjalan sesuai dengan kebutuhan fungsional yang telah ditetapkan. Metode pengujian yang digunakan adalah *Black Box Testing*, yaitu teknik pengujian yang berfokus pada pemeriksaan fungsi sistem berdasarkan masukan (*input*) dan keluaran (*output*) tanpa melihat proses internal atau kode program.

Pengujian dilakukan terhadap seluruh fitur utama, baik pada sisi pengguna (*user*) maupun administrator (*admin*), untuk memastikan bahwa sistem dapat beroperasi dengan baik serta mampu menampilkan rute terpendek secara akurat sesuai dengan hasil perhitungan algoritma Dijkstra. Selain itu, pengujian juga bertujuan memverifikasi bahwa proses perhitungan rute yang dilakukan oleh sistem menghasilkan keluaran yang konsisten dengan hasil perhitungan manual.

Hasil pengujian sistem menggunakan metode *Black Box Testing* disajikan dalam tabel berikut:

Tabel 5 Hasil Pengujian Sistem (*Black Box Testing*)

No.	Fitur yang Diuji	Skenario Pengujian	Input	Output yang Diharapkan	Hasil
1	Login Admin	Memasukkan <i>username</i> dan <i>password</i> yang benar	<i>Username</i> dan <i>password</i> valid	Sistem berhasil menampilkan halaman utama admin	Berhasil
2	Login Admin	Memasukkan <i>username</i> atau <i>password</i> salah	<i>Username</i> atau <i>password</i> tidak valid	Sistem menampilkan pesan kesalahan <i>login</i>	Berhasil
3	Tampilan Peta	Mengakses halaman beranda	Klik menu "Beranda"	Peta tampil dengan marker seluruh lokasi wisata	Berhasil
4	Pencarian Rute Terpendek	Menentukan titik awal dan tujuan	Pemilihan <i>node</i> awal dan <i>node</i> tujuan	Sistem menampilkan rute terpendek sesuai algoritma Dijkstra	Berhasil
5	Validasi Pencarian Rute	Input titik awal atau tujuan tidak lengkap	Hanya satu titik dipilih	Sistem menampilkan pesan bahwa titik belum lengkap	Berhasil
6	Informasi Lokasi	Mengakses halaman detail lokasi wisata	Klik lokasi wisata	Sistem menampilkan deskripsi, koordinat, dan gambar lokasi wisata	Berhasil
7	Kelola Data Wisata	Menambah data wisata baru	Form lengkap data wisata	Data tersimpan dan tampil pada daftar wisata	Berhasil
8	Kelola Data Wisata	Mengedit data wisata	Form edit data wisata	Data berhasil diperbarui di <i>database</i>	Berhasil
9	Kelola Data Wisata	Menghapus data wisata	Klik tombol hapus	Data terhapus dari <i>database</i> dan daftar	Berhasil

No.	Fitur yang Diuji	Skenario Pengujian	Input	Output yang Diharapkan	Hasil
10	Kelola Data <i>Node</i>	Menambah <i>node</i> baru	Nama <i>node</i> dan titik koordinat	<i>Node</i> tersimpan dalam <i>database</i>	Berhasil
11	Kelola Data <i>Node</i>	Mengedit <i>node</i>	Form edit data <i>node</i>	Data <i>node</i> diperbarui dalam <i>database</i>	Berhasil
12	Kelola Data <i>Node</i>	Menghapus <i>node</i>	Klik hapus	<i>Node</i> terhapus dari <i>database</i>	Berhasil
13	Kelola Halaman About	Mengubah informasi halaman About	Update konten text	Informasi About berhasil diperbarui	Berhasil
14	<i>Logout</i> Admin	Menekan tombol <i>logout</i>	Klik “ <i>Logout</i> ”	Sistem kembali ke halaman <i>login</i>	Berhasil

F. Pemeliharaan dan Evaluasi Sistem

Pemeliharaan dan evaluasi sistem dilakukan untuk memastikan bahwa aplikasi pencarian rute terpendek wisata Kabupaten Muna selalu berada dalam kondisi optimal, aman, dan sesuai kebutuhan pengguna. Kegiatan pemeliharaan ini mencakup pemantauan berkala, perbaikan kesalahan (bug fixing), pembaruan data spasial, peningkatan performa algoritma, serta pengembangan fitur tambahan jika diperlukan. Evaluasi sistem dilakukan untuk menilai kesesuaian fungsi sistem terhadap tujuan awal, kemudahan penggunaan, keakuratan hasil algoritma Dijkstra, serta stabilitas aplikasi selama digunakan.

1. Pemeliharaan Sistem (*Maintenance*)

Proses pemeliharaan meliputi beberapa jenis aktivitas berikut:

- a. Pemeliharaan Korektif (*Corrective Maintenance*).
 Pemeliharaan ini dilakukan apabila ditemukan kesalahan sistem, baik dari sisi tampilan maupun fungsi. Meliputi perbaikan error pada tampilan peta ketika marker tidak terbaca dengan benar, perbaikan bug saat admin mengelola *node* dan jarak (CRUD) yang menyebabkan data tidak tersimpan, perbaikan ketidaksesuaian respons rute apabila sistem gagal menerima input titik awal dan titik tujuan.
- b. Pemeliharaan Preventif (*Preventive Maintenance*).
- c. Pemeliharaan ini dilakukan untuk mencegah kerusakan di masa depan dan meningkatkan stabilitas sistem. Kegiatan preventif meliputi optimalisasi performa algoritma Dijkstra serta pengecekan akurasi graf jarak, pemeriksaan struktur *database* untuk menghindari duplikasi *node* atau hubungan jarak yang tidak valid, pembaruan pustaka frontend dan backend untuk meningkatkan keamanan aplikasi, backup data berkala untuk menjaga integritas dan keamanan data.
- d. Pemeliharaan Adaptif (*Adaptive Maintenance*).
- e. Pemeliharaan adaptif dilakukan untuk memastikan sistem tetap relevan dengan perkembangan lingkungan, teknologi, dan kebutuhan pengguna. Kegiatan meliputi penyesuaian koordinat dan penambahan lokasi wisata baru sesuai pedoman terbaru Dinas Pariwisata Kabupaten Muna, penyesuaian tampilan peta apabila terdapat pembaruan pada layanan map provider atau *framework* frontend, pengembangan opsi tampilan rute alternatif jika diminta dalam evaluasi pengguna.
- f. Pemeliharaan Perfektif (*Perfective Maintenance*).
- g. Pemeliharaan jenis ini dilakukan untuk meningkatkan kualitas sistem dari sisi fungsional maupun pengalaman pengguna. Kegiatan meliputi penyempurnaan antarmuka peta agar lebih responsif pada perangkat mobile, penambahan deskripsi lokasi, foto beresolusi tinggi, dan fitur zoom rute, penambahan fitur pencarian lokasi berdasarkan kata kunci (search bar), peningkatan tampilan visual rute dengan warna dan ikon yang lebih jelas.

2. Evaluasi Sistem

Proses evaluasi dilakukan peninjauan secara menyeluruh terhadap kinerja sistem serta tingkat kesesuaian fungsionalitas yang telah diimplementasikan dengan kebutuhan pengguna. Evaluasi ini tidak hanya berfokus pada aspek teknis, tetapi juga pada efektivitas sistem dalam memberikan pengalaman penggunaan yang optimal. Pengujian kembali dilakukan terhadap keakuratan hasil perhitungan rute terpendek yang dihasilkan oleh algoritma Dijkstra, termasuk memastikan bahwa jarak dan lintasan yang ditampilkan tetap konsisten dengan data graf yang digunakan. Selain itu, konsistensi tampilan peta, marker lokasi wisata, serta garis polyline yang menggambarkan

rute perjalanan juga turut ditinjau untuk memastikan bahwa visualisasi peta dapat dipahami dengan jelas oleh pengguna.

Evaluasi juga mencakup pemeriksaan kelancaran fungsi pengelolaan data oleh administrator, seperti proses penambahan, pengubahan, dan penghapusan data wisata maupun *node*. Hal ini penting untuk memastikan bahwa administrator dapat melakukan pembaruan data secara efisien tanpa mengalami kendala teknis.

Proses evaluasi turut melibatkan pengujian ulang terhadap berbagai skenario penggunaan utama, baik pada sisi pengguna maupun admin, guna memastikan bahwa sistem tetap dapat berfungsi dengan baik setelah melalui beberapa siklus pemeliharaan dan perbaikan.

Implementasi algoritma Dijkstra pada sistem penentuan rute wisata Kabupaten Muna menunjukkan bahwa metode ini mampu menghasilkan jalur dengan jarak paling minimum berdasarkan bobot pada setiap sisi graf. Dalam prosesnya, sistem melakukan perhitungan jarak secara iteratif dengan memilih node yang memiliki nilai jarak terkecil dari titik awal hingga node tujuan. Mekanisme tersebut memastikan bahwa jalur yang dihasilkan merupakan jalur optimal dari berbagai kemungkinan lintasan yang tersedia dalam jaringan graf wisata. Proses ini sejalan dengan konsep dasar teori graf yang menyatakan bahwa algoritma Dijkstra efektif digunakan untuk menyelesaikan permasalahan pencarian jalur terpendek pada graf berbobot positif (Moningkey, 2023).

Tahapan implementasi dimulai dari pembentukan graf yang merepresentasikan hubungan antar lokasi wisata di Kabupaten Muna. Setiap node dalam graf mewakili satu titik lokasi wisata, sedangkan setiap edge menunjukkan hubungan antar lokasi dengan bobot berupa jarak tempuh. Representasi graf ini sangat penting karena menjadi dasar dalam proses komputasi algoritma. Pengolahan data spasial yang baik akan mempengaruhi tingkat akurasi jalur yang dihasilkan oleh sistem. Oleh karena itu, struktur graf yang dibangun harus mampu menggambarkan hubungan jarak antar lokasi secara tepat agar proses pencarian rute dapat berjalan secara optimal (Rorimpandey, 2022).

Selain itu, integrasi algoritma dalam sistem berbasis web memungkinkan proses pencarian rute dilakukan secara otomatis melalui interaksi pengguna dengan sistem. Pengguna hanya perlu menentukan titik awal dan tujuan perjalanan, kemudian sistem akan melakukan proses komputasi secara otomatis menggunakan algoritma Dijkstra. Hasil perhitungan tersebut selanjutnya ditampilkan dalam bentuk jalur rute pada peta digital sehingga memudahkan pengguna dalam memahami arah perjalanan. Penggunaan teknologi sistem informasi geografis berbasis web juga meningkatkan efisiensi dalam penyajian informasi lokasi wisata kepada pengguna (Tangkawarow, 2023).

Implementasi sistem ini tidak hanya menitikberatkan pada aspek algoritmik, tetapi juga pada desain antarmuka yang memudahkan pengguna dalam mengakses informasi rute wisata. Antarmuka yang intuitif dan sederhana memungkinkan wisatawan untuk menentukan rute perjalanan tanpa memerlukan pengetahuan teknis mengenai sistem yang digunakan. Desain tampilan yang jelas serta visualisasi peta interaktif membantu pengguna memahami jalur perjalanan yang direkomendasikan oleh sistem secara lebih mudah dan cepat (Kumajas, 2022).

Hasil pengujian sistem menunjukkan bahwa algoritma Dijkstra mampu menghasilkan rute terpendek secara akurat dari titik awal menuju berbagai destinasi wisata yang tersedia. Sistem juga mampu menampilkan visualisasi jalur pada peta digital serta memberikan informasi jarak tempuh secara otomatis kepada pengguna. Dengan adanya sistem ini, wisatawan dapat merencanakan perjalanan dengan lebih efisien karena sistem menyediakan jalur perjalanan yang optimal berdasarkan data jarak antar lokasi yang tersedia dalam basis data sistem. Hal ini menunjukkan bahwa penerapan algoritma graf dalam sistem informasi geografis dapat memberikan kontribusi nyata dalam pengembangan teknologi informasi di sektor pariwisata.

4. Kesimpulan

Berdasarkan hasil analisis dan pengujian yang telah dilakukan terhadap sistem yang dikembangkan, diperoleh beberapa temuan utama yang menjadi dasar dalam penarikan kesimpulan penelitian ini. Temuan-temuan tersebut mencakup keberhasilan penerapan Algoritma Dijkstra dalam menentukan rute terpendek secara akurat, serta keberhasilan pembangunan sistem informasi berbasis web yang mampu menyajikan visualisasi rute wisata secara interaktif dan mudah digunakan oleh pengguna. 1.) Hasil penelitian menunjukkan bahwa penerapan Algoritma Dijkstra pada sistem informasi rute wisata Kabupaten Muna telah berhasil dilakukan dengan baik dan memberikan hasil yang akurat. Algoritma Dijkstra dapat diterapkan secara efektif dengan membangun representasi graf berbobot yang terdiri dari node-node lokasi wisata dan jarak antar node. Proses perhitungan jalur terpendek berjalan optimal, dan seluruh hasil rute yang dihasilkan sistem memiliki kesesuaian 100% dengan perhitungan manual. Hal ini membuktikan bahwa Algoritma Dijkstra mampu menentukan jalur tercepat dan paling efisien menuju destinasi wisata di Kabupaten Muna. 2.) Penelitian ini berhasil membangun sebuah sistem berbasis web

yang dapat menampilkan visualisasi rute terpendek secara interaktif kepada wisatawan. Sistem menyediakan peta dinamis yang menampilkan marker lokasi wisata, garis lintasan rute terpendek, serta informasi jarak tempuh secara langsung. Fitur-fitur pendukung seperti pengelolaan data wisata, pengaturan node dan jarak, serta tampilan antarmuka yang responsif juga bekerja dengan baik berdasarkan hasil pengujian Black Box Testing. Seluruh fungsi utama berjalan stabil tanpa ditemukan bug kritis, dan pengguna dapat menentukan titik awal serta tujuan dengan mudah melalui peta interaktif.

Referensi

1. Afdhaluzzikri, M., Santriawan, L. D., Sapni, E., Nugroho, M. S., & Romdhini, M. U. (2024). Application of Dijkstra Algorithm in Determining Transportation Costs for Tourist Attractions in Lombok Island Based on the Shortest Path. *Jurnal Pariwisata Nusantara (JUWITA)*, 3(2), 67–73. <https://doi.org/10.20414/juwita.v3i2.11017>
2. Alif Pratama, F., Sagit Sahay, A., & Nugrahaningsih, N. (2025). Perbandingan Algoritma Dijkstra dan A-Star dalam Pencarian Rute Terpendek Sekolah Menengah Atas (SMA) di Kota Palangka Raya Berbasis Website. *Journal of Information Technology and Computer Science*, 5(1), 29–48. <https://doi.org/10.47111/jointecoms.v5i1.19805>
3. alifiani, abdillah, dan saliha. (2021). Alifiani, 2021, jalur tercepat dengan algoritma dijkstra. *Jurnal Derivat, Volume 8 No. 2 Desember 2021 ISSN: 2549-2616*, 8(2), 140–148.
4. Amelia Pertiwi, A., Putri Syaifullah Nst, S., & Yandra Niska, D. (2025). Penentuan Rute Terpendek Untuk Jaringan Rel Kereta Api Di Sumatera Utara Menggunakan Matlab. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(4), 6567–6571. <https://doi.org/10.36040/jati.v9i4.14008>
5. Amin, A., & Hendrik, B. (2025). Analisis Penerapan Algoritma Dijkstra dalam Optimasi Penentuan Rute: Sebuah Kajian Literatur Sistematis. *Journal of Education Research*, 6(1), 100–106. <https://doi.org/10.37985/jer.v6i1.2155>
6. Andriati, D. A., Patria, M., & Ramadhan, N. A. (2025). *MENGGUNAKAN TEORI GRAF UNTUK PEMERINGKATAN SITUS WEB*. 81–87.
7. Anggarawati, T., & Eka, R. (2024). Menapaki Jejak Komputasi Modern: Fondasi Matematis yang Membentuk Era Digital. *Aljabar : Jurnal Ilmuan Pendidikan, Matematika dan Kebumihan*, 2(1), 07–12.
8. Anindito Wisnu Susanto, L., Anastasya Ursia, A., Kurnianing Tyas, A., Omega Putri Usdinoari, C., Arif Budi Prasetyo, D., & Surya Nugraha, A. (2022). Implementasi Algoritma Dijkstra Dalam Menentukan Rute Tependek Objek Wisata Menggunakan Transportasi Transjakarta. *Seminar Nasional Pendidikan Matematika*, 2(1), 280–290.
9. Audil, R., Wardhana, I. G. A. W., Abdurahim, A., Alfian, M. R., & Aini, Q. (2022). *Indeks hyper-wiener, harrary, szeged, dan padmakar-ivan dari graf pangkat pada grup bilangan bulat modulo hyper-wiener, harrary, szeged, and padmakar-ivan indices of power graphs on integer group modulo*.
10. Bhekti, N. K. A. dan C. (2021). *WebGIS Berbasis Leaflet JS untuk Pemetaan*. 10(2), 510–515.
11. Daniel, F., & Taneo, P. N. L. (2020). *Teori Graf*. Deepublish.
12. Endra, R. Y., Aprilinda, Y., Dharmawan, Y. Y., & Ramadhan, W. (2022). Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website. *Expert*, 11(1), 346061.
13. Giraud, T. (2022). osrm : Interface Between R and the OpenStreetMap-Based Routing Service OSRM. *Journal of Open Source Software*, 7, 1–7. <https://doi.org/10.21105/joss.04574>
14. Grujic, Z. (2025). *Optimal Routing in Urban Road Networks : A Graph-Based Approach Using Dijkstra 's Algorithm*.
15. Hadi, H. M., & Ibrahim, I. M. (2025). A Comprehensive Review of Shortest Path Algorithms for Network Routing. *Asian Journal of Research in Computer Science*, 18(3), 152–175. <https://doi.org/10.9734/ajrcos/2025/v18i3584>
16. Herdiansah, A., Purnamasari, M., Handayani, T., Sulaeman, A., Studi, P., Informatika, T., Teknik, F., Tangerang, U. M., Tangerang, K., Tangerang, K., Studi, P., Informatika, T., Informatika, F. T., Raya, U. S., Studi, P., & Informatika, T. (2024). *IMPLEMENTASI FRAMEWORK CODEIGNITER PADA PENGEMBANGAN SISTEM INFORMASI DASHBOARD TRACKING*. 8(2), 213–220.
17. Irawan Chandra, Y., Ruri Irawati, D., Riastuti, M., Jakarta STI, S., BRI No, J., Dalam, R., Baru, K., & Selatan, J. (2025). Perancangan UI/UX Aplikasi Layanan Kesehatan Hewan Menggunakan Design Thinking dan Model Agile Berbasis Web. *Jurnal Ilmiah Teknik Informatika*, 26(1), 184–197.
18. Ismawati, I., & Maulani, A. (2024). *Implementasi Algoritma Dijkstra Menggunakan Adjacency Matrix*. 14(1), 1–10. <https://doi.org/10.24843/JMAT.2024.v14.i01.p167>
19. Jaelani, A., & Hanim, T. F. (2021). Teknologi Digital, Keberlanjutan Lingkungan, Dan Desa Wisata Di Indonesia. *Al-Mustashfa: Jurnal Penelitian Hukum Ekonomi Syariah*, 6(2), 237. <https://doi.org/10.24235/jm.v6i2.9613>
20. Jonathan Steven Iskandar, & Yosefina Finsensia Riti. (2022). Perbandingan Algoritma Greedy dan Algoritma Dijkstra dalam Pencarian Rute Terpendek dari Kabupaten Tuban ke Kota Surabaya. *Jurnal PETIK*, 8(2), 96–106.
21. Kafando, H., Ouedraogo, B., Ojeh, V. N., Rienow, A., Gadiaga, A., Elh, I., & Garba, M. (2024). *Development of a Web-Based GIS of Flood Zones in the Municipality of Ouagadougou in Burkina Faso*. 32–43. <https://doi.org/10.4236/jgis.2024.161003> \
22. Kenap, A. A., Moningkey, E. R. S., Tangkawarow, I. R. H. T., Rorimpandey, G. C., & Kumajas, S. C. (2024). Implementasi algoritma graf untuk sistem penentuan rute wisata. *Journal of Information Systems*, 10(2), 120–130.
23. Kansha, W. M. (2023). *Analisis Perbandingan Struktur dan Performa Framework Codeigniter dan Laravel dalam Pengembangan Web Application*. 09(01), 25–31.
24. Komarullah, H. (2025). Penggunaan Algoritma Dijkstra Dalam Menentukan Rute Terpendek Wisata Di Kabupaten Jember. *As-Sunninyah*, 4(02), 96–101. <https://doi.org/10.62097/assunninyah.v4i02.2139>
25. Kothimbire, M. D. K., Shelke, D. S., Gaikwad, M. S. V., Yelpale, A. P., & Shinde, R. N. (2025). A Comprehensive Review of Graph Theory Applications in Network Analysis. *International Journal of Mathematics And Computer Research*, 13(03), 4956–4967. <https://doi.org/10.47191/ijmcr/v13i3.08>
26. Kumajas, S. C. (2022). User interface design for web-based information systems. Manado: Universitas Negeri Manado.
27. Lestari, S., Giovani, A. P., Dwijayanti, D., & Komputer, I. (2020). *Penerapan Algoritma Dijkstra Dalam Penentuan Rute*. 7(September), 1–6.
28. Manalu, W. U. S., Hakim, L., & Wulandari, C. (2023). Sistem Informasi Pengaduan Siswa Berbasis Website Dengan Framework Laravel. *Journal of Information System Research (JOSH)*, 4(3), 1005–1013. <https://doi.org/10.47065/josh.v4i3.3368>
29. Mardiani, D., & Maulani, D. A. (2023). Efektifitas Perkuliahan Teori Graf Menggunakan Media Pembelajaran Multiplatform. *Jurnal PETIK*, 9(1), 35.

DOI: <https://doi.org/10.31004/riggs.v5i1.7352>

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

30. Maro, L., & Purab, L. K. S. (2021). Penerapan Konsep Pewarnaan Graf dalam Penyusunan Jadwal Perkuliahan Menggunakan Metode Algoritma Welch-Powell pada Program Studi Teknik Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Tribuana Kalabahi. *Jurnal Ilmiah Wahana Pendidikan*, 7(6), 193–197. <https://doi.org/10.5281/zenodo.5574383>
31. Musliyana, Z., & Helinda, A. (2022). Analisis Performansi Query Mysql Menggunakan Query Builder Pada Framework Codeigniter 4. *Journal of Informatics and Computer Science*, 8(1), 36–40.
32. Musridho, R. J., Rusnedy, H., Febri, W., & Sudirman, R. (2025). *Analisis Komparatif Algoritma Dijkstra dan Google Maps API dalam Penentuan Rute Tercepat*. 4(3), 5017–5022.
33. Moningkey, E. R. S. (2023). Graph theory and shortest path algorithms in geographic information systems. Manado: Universitas Negeri Manado.
34. Niarmann, A., Iswandi, & Candri, A. K. (2023). Comparative Analysis of PHP Frameworks for Development of Academic Information System Using Load and Stress Testing. *International Journal Software Engineering and Computer Science (IJSECS)*, 3(3), 424–436. <https://doi.org/10.35870/ijseecs.v3i3.1850>
35. Parmanand, P., Sahdev, S., & Dwivedi, A. (2024). Study the optimization of Dijkstra's Algorithm. *Journal of Ravishankar University (PART-B)*, 37(2), 255–267. <https://doi.org/10.52228/jrub.2024-37-2-18>
36. Permana, N. P., Triase, T., & Harahap, A. M. (2024). SISTEM INFORMASI MANAJEMEN PENDIDIKAN MADRASAH (SISPEMAD) KEMENTERIAN AGAMA KABUPATEN SERDANG BEDAGAI MENGGUNAKAN METODE EXECUTIVE INFORMATION SYSTEM (EIS). *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(6), 11399–11405.
37. Pramadjaya, A., & Rohmawati, I. (2025). *Analysis of the Shortest Route to Universitas Pamulang Using the Implementation of Dijkstra's Algorithm Analisis Rute Terpendek Menuju Universitas Pamulang dengan Implementasi Algoritma Dijkstra*. 5(July), 953–961.
38. Probrini, E. (2025). ... with Dijkstra's Algorithm Shortest Route Search with Dijkstra's algorithm: Pencarian Rute Terpendek dengan Algoritma Dijkstra Shortest Route Search with Dijkstra's *Jurnal/ Eduelektromatika*, 6(1), 50–57.
39. Purba, J., Manurung, S., Girsang, J., Gulo, J. A., & Sipayung, S. P. (2025). Implementasi Algoritma Dijkstra Untuk Menentukan Rute Terpendek dari SMA 17 Medan Ke Unika St.Thomas. *Jurnal Minfo Polgan*, 14(1), 1045–1052. <https://doi.org/10.33395/jmp.v14i1.14963>
40. Rahmadi Asri, S., Kom, M., Uilly Muzakir, M. T., Choirina, P., Aufar, Y., Kom, M., Nugraha, D., S ST, M. I. T., Ira Zulfa, S. T., & Lase, D. C. (2025). *Pengantar Teknik Informatika*. Azzia Karya Bersama.
41. Ramadhan, F., Aulienda, A., & Nurrobbi, F. (2024). *Perencanaan Rute Optimal Kunjungan Destinasi Wisata Bandung dengan Algoritma Dijkstra Pada C++*. 6(2), 275–281.
42. Riti, Y. F., Iskandar, J. S., & Hendra, H. (2023). Comparison Analysis of Graph Theory Algorithms for Shortest Path Problem. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 12(3), 415–424. <https://doi.org/10.32736/sisfokom.v12i3.1756>
43. Rorimpandey, G. C. (2022). Spatial data management in geographic information systems. Manado: Universitas Negeri Manado.
44. Sari Siregar, Y., Oktaviana Sembiring, B., Rahayu, E., & Franchitika, R. (2024). Pemanfaatan Aplikasi MySQL untuk Membantu Siswa SMK Swasta Nur Azizi dalam Pengolahan Data Utilization of the MySQL Application to Assist Nur Azizi Private Vocational School Students in Data Processing. *Desember*, 2024(2), 229–240.
45. Statistik, B. P. (2025). *Jumlah Perjalanan Wisatawan Nusantara Menurut Kabupaten/Kota Asal (Perjalanan)*. <https://sultra.bps.go.id/id/statistics-table/2/NjE5ZlI=/jumlah-perjalanan-wisatawan-nusantara-menurut-kabupaten-kota-asal.html>
46. Suryana, F. (2020). *Analisis Perbandingan Algoritma Dijkstra dan Ant Colony Optimization (ACO) dalam Optimalisasi Rute Jalur Wisata di Takengon*. 1–5.
47. Syah, M. F., Wardhono, W. S., & Afirianto, T. (2025). *Pengembangan Aplikasi Monitoring Lesson Study (Edvisor) Berbasis Website Menggunakan Model SDLC Waterfall*. 9(6), 1–8.
48. Syefudin, S., Zain, A. M., & Gunawan, G. (2023). Implementasi Algoritma Dijkstra Dalam Penentuan Jalur Terpendek Menuju Objek Wisata Di Kabupaten Tegal. *Jurnal Technopreneur (JTech)*, 11(2), 70–75. <https://doi.org/10.30869/jtech.v11i2.1233>
49. **Tangkawarow, I. R. H. T. (2023). Web-based geographic information systems for tourism route planning. Manado: Universitas Negeri Manado.**
50. TAPOBALL, P. J. B. B., & KRISNAMURTI, C. N. (2020). Implementasi Algoritma Dijkstra Dalam Menentukan Rute Terpendek Bis Transjakarta Dalam Mengunjungi 5 Destinasi Wisata Populer Di Jakarta. *E-Jurnal Matematika*, 9(4), 265. <https://doi.org/10.24843/mtk.2020.v09.i04.p307>
51. Trisani, C., Garingging, S., Gaol, S. L., Lubis, M. A., Sianturi, T., Sembiring, B. M., Sipayung, S. P., Katolik, U., & Thomas, S. (2025). *Implementasi Algoritma Dijkstra dalam Menentukan Rute Terpendek dari Unika St. Thomas Medan ke Lapangan Merdeka*. 14, 789–800.
52. Trisna, P., Permana, H., & Primakara, U. (2025). 3 1,2,3. 11(2), 225–233.
53. Ubaidillah Irganata Putra Mamba'ul Ulum, Indyah Hartami Santi, & Mukh. Taofik Chulkamdi. (2025). Implementation of Dijkstra Algorithm in Determining the Fastest Route for Goods Delivery. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, 4(2), 1443–1449. <https://doi.org/10.59934/jaiea.v4i2.923>
54. Uktolseja, L. J., Manuhutu, M. A., Leuwol, N. V., & Manuhutu, A. (2025). *Cakrawala : Jurnal Pengabdian Masyarakat Global Available Online at : https://jurnaluniv45sby.ac.id/index.php/Cakrawala Penerapan Teknologi Odoo dalam Prespektif Perubahan Sosial pada Layanan Kesehatan Implementation of Odoo Technology in the Perspective . 4.*
55. Waruwu, J., Ofelius Laia, Yusmei Paskah Harefa, Abdul Majid Tanjung, & Michael Laowo. (2024). Optimalisasi Manajemen Pegawai Dan Sistem Informasi Berbasis Web Dengan Menggunakan Database Mysql (Studi Kasus Pt. Xyz). *Jurnal Informatika Teknologi dan Sains (Jinteks)*, 6(1), 33–38. <https://doi.org/10.51401/jinteks.v6i1.3703>