

PENERAPAN ALGORITMA DIJKSTRA JALUR TERPENDEK ANTAR OBJEK WISATA DANAU TOBA

¹Ester Manalu*, ²Virzinia Napitupulu, ³Maria Afriyani, ⁴Hoth Badia Butar-butur.

^{1,2,3,4}Program Studi Teknik Informatika, Fakultas Ilmu Komputer.
Universitas Katolik Santo Thomas Medan¹²³⁴.
Jl. Setia Budi No.479 F, Tanjung Sari Medan, Sumatera Utara

*e-mail: estermanalu51@gmail.com¹, virzinianapitupulu5@gmail.com²,
mariaafriyani7@gmail.com³, hothbadia@gmail.com³.

Abstrak

Penentuan rute terpendek dalam sistem navigasi pariwisata memiliki peran penting dalam meningkatkan efisiensi perjalanan wisatawan, terutama di kawasan yang memiliki banyak destinasi seperti Danau Toba. Penelitian ini bertujuan untuk mengimplementasikan algoritma Dijkstra dalam pencarian rute tercepat antar lokasi wisata utama di kawasan Danau Toba. Metode yang digunakan adalah pendekatan kuantitatif berbasis pemodelan graf, di mana simpul-simpul (nodes) merepresentasikan titik-titik wisata dan sisi (edges) menunjukkan hubungan antar lokasi yang diukur berdasarkan jarak aktual dari Google Maps. Simulasi dilakukan menggunakan titik-titik wisata yang direpresentasikan sebagai node A hingga L, yaitu: Balige (A), Sitorang (B), Dolok Sanggul (C), Porsea (D), Hutagalung (E), Parapat (F), Parbuluan (G), Dolok Pardamean (H), Paropo (I), Tiga Rungu (J), Saribu Dolok (K), dan Merek (L). Salah satu contoh hasil simulasi menunjukkan bahwa lintasan A → B → D → F → H → K → L merupakan rute tercepat dengan jarak tempuh terpendek. Implementasi algoritma Dijkstra terbukti efektif dalam menyelesaikan persoalan penentuan rute optimal berdasarkan parameter jarak. Temuan ini berpotensi untuk diterapkan dalam pengembangan sistem navigasi lokal berbasis wisata guna menunjang kenyamanan dan efisiensi perjalanan. Penelitian selanjutnya disarankan untuk mengintegrasikan variabel waktu tempuh, kondisi lalu lintas secara real-time, dan preferensi pengguna untuk hasil rekomendasi rute yang lebih relevan.

Kata Kunci: Algoritma Dijkstra, Rute Terpendek, Danau Toba, Google Maps, Navigasi Wisata.

Abstract

Determining the shortest route in tourism navigation systems plays a crucial role in improving travel efficiency for tourists, especially in regions with numerous destinations such as Lake Toba. This study aims to implement Dijkstra's algorithm to identify the fastest routes between major tourist locations in the Lake Toba area. The method used is a quantitative approach based on graph modeling, where the nodes represent tourist points and the edges indicate connections between locations, measured using actual distances from Google Maps. The simulation was conducted using tourist points represented as nodes A through L, namely: Balige (A), Sitorang (B), Dolok Sanggul (C), Porsea (D), Hutagalung (E), Parapat (F), Parbuluan (G), Dolok Pardamean (H), Paropo (I), Tiga Rungu (J), Saribu Dolok (K), and Merek (L). One example from the simulation shows that the path A → B → D → F → H → K → L is the fastest route with the shortest travel distance. The implementation of Dijkstra's algorithm



licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

DOI: <https://doi.org/10.52362/jmijayakarta.v5i4.2002>

proved effective in solving the problem of determining the optimal route based on distance parameters. These findings have the potential to be applied in the development of local tourism-based navigation systems to enhance travel comfort and efficiency. Further research is recommended to integrate travel time variables, real-time traffic conditions, and user preferences to generate more relevant route recommendations.

Keywords: Dijkstra's Algorithm, Shortest Route, Lake Toba, Google Maps, Tourism Navigation.

1 Pendahuluan (or Introduction)

Danau Toba merupakan salah satu destinasi wisata unggulan di Indonesia, yang terletak di Provinsi Sumatera Utara. Keindahan alamnya yang unik, ditambah dengan kekayaan budaya Batak di sekitarnya, menjadikan kawasan ini menarik untuk dikunjungi oleh wisatawan domestik maupun mancanegara. Terdapat berbagai objek wisata di sekitar Danau Toba, seperti Pulau Samosir, Air Terjun Sipiso-piso, Bukit Holbung, dan lainnya yang tersebar di berbagai lokasi geografis.

Namun, jarak antar objek wisata yang berjauhan dan kondisi medan yang bervariasi seringkali menyulitkan wisatawan dalam menentukan rute perjalanan yang efisien. Hal ini dapat menyebabkan pemborosan waktu, tenaga, dan biaya selama berwisata. Oleh karena itu, diperlukan suatu sistem atau metode yang dapat membantu menentukan jalur terpendek antar objek wisata secara akurat dan cepat.

Algoritma Dijkstra merupakan salah satu algoritma pencarian jalur terpendek yang paling banyak digunakan dalam berbagai aplikasi, terutama dalam sistem pemetaan dan navigasi. Algoritma ini bekerja dengan menghitung jarak minimum dari satu titik ke titik lainnya dalam graf berbobot positif, sehingga sangat sesuai untuk diterapkan dalam konteks pemilihan rute wisata.

Penelitian ini bertujuan untuk menerapkan algoritma Dijkstra dalam menentukan rute terpendek antar objek wisata di sekitar Danau Toba. Dengan memodelkan objek wisata sebagai simpul (node) dan jarak antar objek sebagai sisi (edge), diharapkan hasil yang diperoleh dapat membantu pengembangan sistem navigasi atau aplikasi pemandu wisata yang lebih efisien dan bermanfaat bagi wisatawan maupun pengelola destinasi.

2 Tinjauan Literatur (or Literature Review)

Algoritma Dijkstra merupakan salah satu teknik populer dalam pencarian lintasan terpendek pada graf berbobot dengan nilai non-negatif. Konsep ini pertama kali diperkenalkan oleh Edsger Dijkstra pada tahun 1959 dan sejak saat itu telah banyak diterapkan dalam berbagai sistem navigasi serta perencanaan jalur karena keefektifannya dalam menentukan rute optimal.

Beragam studi telah menunjukkan penerapan algoritma ini dalam konteks pariwisata. Misalnya, [1] menggunakan pendekatan algoritma Dijkstra untuk menentukan rute perjalanan wisata di wilayah Yogyakarta berdasarkan data spasial. Selain itu, [2] mengembangkan sistem berbasis peta digital (GIS) yang mampu merekomendasikan jalur wisata terbaik melalui integrasi algoritma Dijkstra. Sementara itu, penelitian oleh [3] menunjukkan efektivitas algoritma tersebut dalam penentuan lintasan antar objek wisata di sekitar Danau Toba, dengan dukungan aplikasi berbasis Android.

Berdasarkan temuan dari studi-studi sebelumnya, dapat ditarik kesimpulan bahwa kombinasi algoritma Dijkstra dengan sistem pemetaan digital berpotensi besar dalam menghasilkan saran rute yang efisien dan tepat sasaran. Dengan mempertimbangkan hal tersebut, penelitian ini melanjutkan pendekatan serupa dengan fokus khusus pada kawasan wisata Danau Toba sebagai wilayah studi.

3 Metode Penelitian (or Research Method)

Metode penelitian yang digunakan dalam penulisan ini adalah Algoritma Dijkstra. Algoritma ini digunakan untuk mencari jalur terpendek dari satu titik ke titik lainnya berdasarkan bobot terkecil



licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

DOI: <https://doi.org/10.52362/jmijayakarta.v5i4.2002>

yang ada pada graf berbobot [4]. Algoritma ini sangat sesuai untuk menyelesaikan masalah pencarian rute optimal antar objek wisata di sekitar kawasan Danau Toba.

Adapun tahapan yang dilakukan dalam penerapan algoritma ini terdiri dari beberapa langkah sistematis sebagai berikut:

3.1 Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen, yang bertujuan untuk mengevaluasi efektivitas algoritma Dijkstra dalam menemukan rute tercepat antar objek wisata di sekitar Danau Toba berdasarkan kondisi spasial nyata. Pendekatan ini dipilih karena sesuai dengan karakteristik permasalahan yang melibatkan perhitungan jarak, pemodelan graf, serta optimasi jalur menggunakan data numerik [5]. Pendekatan eksperimental memungkinkan peneliti untuk melakukan pengujian langsung terhadap algoritma Dijkstra dengan input yang terkontrol dan hasil yang terukur, sehingga simpulan yang diperoleh dapat bersifat objektif dan terverifikasi.

3.2 Teknik Pengumpulan Data

Data yang digunakan dalam penelitian ini diperoleh dari dua sumber utama, yaitu data primer dan data sekunder. Kedua jenis data ini digunakan secara bersamaan untuk membentuk representasi graf jaringan antar objek wisata secara lebih akurat [6].

a. Data Primer

Data primer dikumpulkan melalui observasi langsung terhadap jalur-jalur yang menghubungkan sejumlah objek wisata di kawasan Danau Toba. Peneliti mencatat titik-titik penting seperti persimpangan jalan, lokasi wisata utama, tikungan, dan titik temu jalur. Pengukuran jarak antar titik dilakukan secara manual menggunakan alat ukur dan perangkat GPS untuk mendapatkan nilai jarak aktual [7]. Data ini digunakan sebagai dasar pembobotan pada setiap sisi dalam graf yang dibentuk.

b. Data Sekunder

Data sekunder diperoleh melalui sumber peta digital seperti Google Maps dan OpenStreetMap. Data ini digunakan sebagai referensi tambahan untuk menentukan struktur graf jalan secara keseluruhan serta memverifikasi letak simpul dan hubungan antar titik. Selain itu, data sekunder juga digunakan untuk mendukung proses visualisasi hasil perhitungan jalur terpendek dan melakukan perbandingan efektivitas hasil perhitungan algoritma [8].

3.3 Prosedur Penelitian

a. Penentuan Simpul dan Jaringan

Langkah awal dilakukan dengan mengidentifikasi dan menentukan simpul-simpul penting yang mewakili lokasi objek wisata serta jalur penghubung di kawasan Danau Toba. Setiap simpul diberi kode khusus agar dapat dengan mudah digunakan dalam proses pemodelan graf. Hubungan antar simpul dicatat untuk membentuk struktur jaringan.

b. Pengukuran dan Pembobotan Jalur

Setelah simpul ditentukan, dilakukan pengukuran jarak antar simpul untuk menentukan nilai bobot pada setiap sisi dalam graf. Pengukuran dilakukan dalam satuan kilometer, dan hasilnya disusun dalam bentuk matriks jarak atau daftar tetangga, sebagai representasi graf berbobot yang akan diproses oleh algoritma Dijkstra.

c. Penerapan Algoritma Dijkstra

Dengan struktur graf yang telah dibentuk, algoritma Dijkstra diterapkan untuk mencari jalur dengan jarak minimum antara dua titik objek wisata. Proses dilakukan melalui



DOI: <https://doi.org/10.52362/jmijayakarta.v5i4.2002>

tahapan seleksi simpul dengan bobot terkecil, pembaruan nilai jarak pada simpul-simpul tetangga, serta penandaan simpul yang telah dikunjungi. Proses berulang hingga ditemukan rute terpendek menuju simpul tujuan. Perhitungan dilakukan secara sistematis menggunakan bantuan tabel atau perangkat lunak spreadsheet.

d. Visualisasi Hasil

Hasil perhitungan algoritma kemudian divisualisasikan dalam bentuk peta berbasis digital, yang menunjukkan jalur terpendek dari simpul awal ke simpul tujuan. Visualisasi ini bertujuan untuk memudahkan proses interpretasi serta memungkinkan pengguna atau peneliti lain melakukan verifikasi visual terhadap hasil perhitungan yang diperoleh.

3.4 Evaluasi dan Pengujian Hasil

Evaluasi hasil dilakukan dengan cara membandingkan rute yang diperoleh dari algoritma Dijkstra dengan rute yang direkomendasikan oleh aplikasi pemetaan digital seperti Google Maps. Pengujian ini bertujuan untuk menilai efisiensi jalur yang dihasilkan oleh algoritma berdasarkan graf yang telah dimodelkan. Selain membandingkan total jarak tempuh, pengujian juga mencakup jumlah simpul yang dilalui dan tingkat kesesuaian jalur dengan kondisi aktual di lapangan. Evaluasi ini penting untuk memastikan bahwa algoritma mampu memberikan solusi optimal yang dapat digunakan secara praktis dalam konteks perjalanan wisata di kawasan Danau Toba.

Flowchart Penelitian



Gambar 1. Flowchart penelitian

4 Hasil dan Pembahasan (or Results and Analysis)

1. Pengumpulan Data

Pada tahap ini dilakukan proses pemetaan jalur antar objek wisata yang berada di sekitar kawasan Danau Toba, Sumatera Utara. Proses ini diawali dengan identifikasi titik awal (node awal) dan titik tujuan akhir (node akhir) yang menjadi acuan dalam pencarian

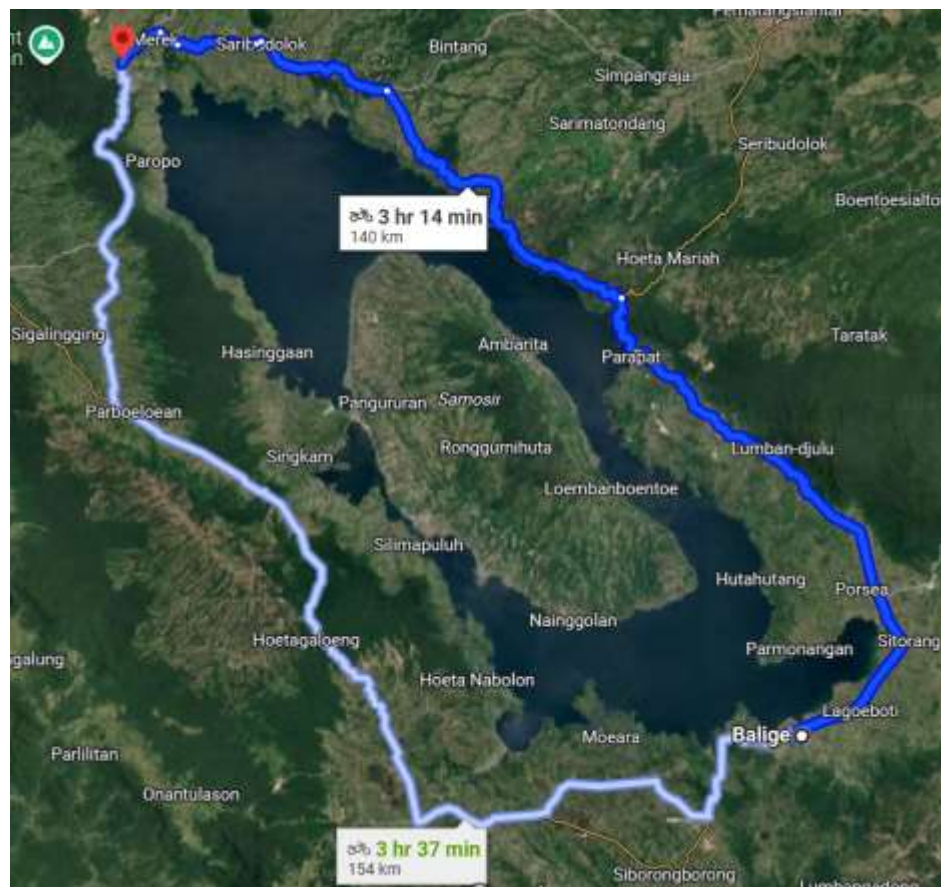


icensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

DOI: <https://doi.org/10.52362/jmijayakarta.v5i4.2002>

jalur terpendek. Dalam konteks penelitian ini, titik awal ditetapkan pada lokasi wisata populer seperti Pelabuhan Ajibata, sedangkan titik tujuan akhir dapat ditentukan pada lokasi strategis lain seperti Bukit Indah Simarjarunjung atau objek wisata lainnya yang menjadi destinasi. Setelah titik awal dan tujuan ditetapkan, dilakukan penentuan ruas-ruas jalan yang menghubungkan antar objek wisata tersebut. Ruas-ruas ini kemudian direpresentasikan sebagai sisi (edge) dalam graf yang akan dianalisis.

Setiap ruas jalan yang menghubungkan dua simpul dalam graf diberikan bobot berdasarkan estimasi jarak tempuh atau panjang jalur nyata antara dua lokasi tersebut [9] Informasi mengenai jarak antar objek wisata diperoleh dari peta digital seperti Google Maps serta observasi lapangan yang relevan. Pemetaan jalur dilakukan secara visual pada peta kawasan Danau Toba dengan menandai simpul-simpul penting, seperti lokasi wisata utama, persimpangan jalan, dan titik temu jalur penghubung. Visualisasi awal ini bertujuan untuk memberikan gambaran spasial dari lokasi yang akan dimodelkan dalam bentuk graf berbobot, yang kemudian digunakan dalam penerapan algoritma Dijkstra untuk menentukan rute wisata tercepat dan paling efisien di kawasan Danau Toba.



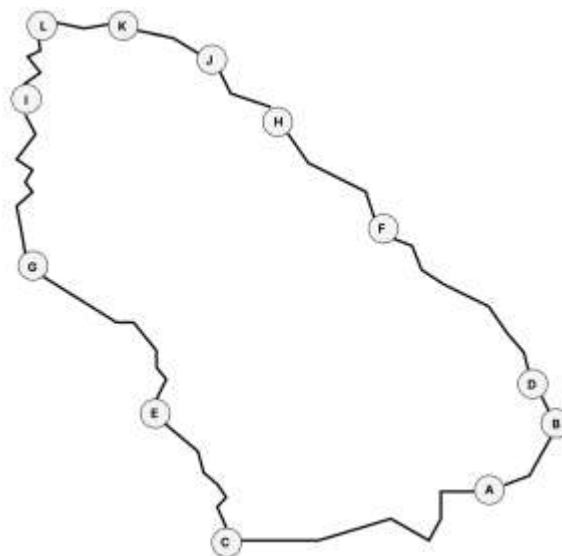
Gambar 2. Peta dari Balige ke Merek

Kemudian, peta ini ditransformasikan menjadi sebuah representasi graf berbobot. Dalam graf tersebut, setiap simpul (node) menggambarkan lokasi atau titik persimpangan, sedangkan setiap sisi (edge) menggambarkan koneksi antar simpul, dengan bobot berupa nilai jarak antar titik tersebut.



DOI: <https://doi.org/10.52362/jmijayakarta.v5i4.2002>

Berikut adalah visualisasi dari graf yang dibangun berdasarkan pemetaan jalur yang telah dilakukan.



Gambar 3. Representasi graf berbobot dari jalur pada peta

Keterangan Simpul (Node) :

- A = Balige
- B = Sitorang
- C = Dolok Sanggul
- D = Porsea
- E = Hutagalung
- F = Parapat
- G = Parbuluan
- H = Dolok Pardamean
- I = Paropo
- J = Tiga Rungu
- K = Saribu Dolok
- L = Merek

Dengan demikian, simpul A ditetapkan sebagai node awal, dan simpul O sebagai node akhir dalam proses pencarian rute terpendek menggunakan algoritma Dijkstra.

2. Penentuan Bobot pada Graf

Setelah proses pemetaan lokasi objek wisata ke dalam bentuk graf dilakukan, tahap selanjutnya adalah menetapkan bobot pada setiap sisi graf yang merepresentasikan jarak antar simpul. Penetapan bobot ini merupakan komponen penting dalam penerapan algoritma Dijkstra, karena bobot berfungsi sebagai acuan untuk menentukan jalur terpendek antar lokasi wisata. Nilai bobot pada graf diberikan berdasarkan estimasi panjang jalan atau jarak tempuh antar dua titik simpul yang terhubung secara langsung dalam jaringan

Tabel 1. Rincian Bobot Sisi pada Graf

NO	Simpul	Total Jarak
1	A → B	15,5 KM



licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

DOI: <https://doi.org/10.52362/jmijayakarta.v5i4.2002>

2	A → C	52 KM
3	B → D	7,7 KM
4	C → E	27,3 KM
5	D → F	39,9 KM
6	E → G	33,3 KM
7	F → H	35,6 KM
8	G → I	43,1 KM
9	H → J	11,6 KM
10	I → L	24,6 KM
11	J → K	20,5 KM
12	K → L	17,1 KM

Berdasarkan data bobot yang telah ditentukan dalam tabel di atas, maka dapat dibentuk graf berbobot yang menggambarkan konektivitas antar simpul serta nilai bobot dari setiap sisi yang menghubungkannya [10].



Gambar 4. Representasi Graf Berbobot Jalur dari Balige ke Merek

3. Perhitungan Graph

Setelah bobot ditentukan pada setiap sisi graf, langkah selanjutnya adalah menetapkan simpul awal dan simpul tujuan untuk proses pencarian rute. Dalam penelitian ini, simpul A ditentukan sebagai titik awal, sedangkan simpul L sebagai titik tujuan. Proses perhitungan dimulai dengan melakukan inisialisasi jarak dari simpul awal ke seluruh simpul lainnya dalam graf.

Pada tahap ini, jarak dari simpul A ke dirinya sendiri diatur bernilai nol (0), karena tidak ada perpindahan. Sementara itu, jarak dari simpul A ke seluruh simpul lainnya dianggap belum diketahui, sehingga diinisialisasi dengan nilai tak hingga (∞) sebagai bentuk asumsi



DOI: <https://doi.org/10.52362/jmijayakarta.v5i4.2002>

bahwa belum ada jalur yang ditemukan. Nilai-nilai inisialisasi awal tersebut adalah sebagai berikut:

- a. Inisialisasi Jarak: Tetapkan nilai awal jarak dari simpul awal ke semua simpul lain dalam graf. Semua simpul, kecuali simpul awal, akan memiliki jarak awal tak hingga (∞). Simpul awal akan memiliki jarak awal 0 dari dirinya sendiri.

Jarak A = 0

Jarak B = ∞

Jarak C = ∞

Jarak D = ∞

Jarak E = ∞

Jarak F = ∞

Jarak G = ∞

Jarak H = ∞

Jarak I = ∞

Jarak J = ∞

Jarak K = ∞

Jarak L = ∞

Tentukan simpul terdekat dari simpul awal. Simpul A hanya terhubung dengan simpul.

- b. b. Simpul B memiliki jarak 15,5 KM dari simpul awal A. Karena simpul A hanya memiliki dua simpul tetangga, yaitu B (15,5 KM) dan C (52 KM), maka dipilih simpul dengan jarak terpendek, yaitu B. Maka ditetapkan bahwa jalur terpendek dimulai dari A ke B.
- c. c. Dari simpul B, satu-satunya jalur yang tersedia dalam tabel adalah menuju simpul D sejauh 7,7 KM. Maka jalur diperpanjang menjadi A ke B ke D.
- d. d. Simpul D memiliki satu simpul tetangga, yaitu F, dengan jarak 39,9 KM. Maka perhitungan dilanjutkan dari simpul D ke F, sehingga jalur terpendek menjadi A ke B ke D ke F.
- e. e. Simpul F memiliki satu simpul tetangga yaitu H sejauh 35,6 KM. Maka jalur dilanjutkan ke simpul H, menjadi A ke B ke D ke F ke H.
- f. f. Simpul H hanya terhubung ke simpul J dengan jarak 11,6 KM. Maka jalur diperpanjang menjadi A ke B ke D ke F ke H ke J.
- g. g. Simpul J memiliki jalur ke simpul L sejauh 24,6 KM. Maka jalur menjadi A ke B ke D ke F ke H ke J ke L.

maka ditetapkan bahwa jalur terpendek adalah:

A ke B ke D ke F ke H ke J ke L [11].

Tabel 2. Gambar tabel Perhitungan Algoritma Dijkstra

	A	B	C	D	E	F	G
A	0	15.5	52.0	23.2	79.3	63.1	112.6
	A	AB	AC	ABD	ACE	ABDF	ACEG



icensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

DOI: <https://doi.org/10.52362/jmijayakarta.v5i4.2002>

B	15.5 BA	0 B	67.5 BAC	7.7 BD	84.8 BACE	47.8 BDF	118.1 BACEG
C	52.0 CA	67.5 CBA	0 C	75.2 CBAD	27.3 CE	115.1 CBADF	60.6 CEG
D	23.2 DBA	7.7 DAB	75.2 DABC	0 D	92.5 DACE	39.9 DF	126.0 DACEG
E	79.3 ECA	84.8 ECBA	27.3 EDC	92.2 EDCB	0 E	132.4 EDF	33.3 EG
F	63.1 FDAB	47.6 FBD	115.1 FCBA	39.9 F	132.4 FE	0 F	159.3 FEG
G	112.6 GECBDA	118.1 GECB	60.6 GC	126.0 GCBD	33.3 GE	159.3 GFED	0 G
H	155.7 IGECBA	83.2 HFDB	150.7 HFCBA	75.5 HFD	167.5 HFEC	35.6 HF	192.9 HFG
I	110.3 JHFDBA	161.1 IGECB	103.7 IC	169.1 ICDB	76.4 IE	202.4 IFEDC	43.1 IG
J	130.8 KJHFDBA	94.8 JHFDB	162.3 JHFC	87.1 JHFD	179.1 JHFE	47.2 JHF	206.6 JHFG
K	148.3 LIHFDBA	115.3 KJHFDB	182.8 KJHFC	107.6 KJHDFD	199.6 KJHFE	67.7 KJHF	227.1 KJHFG
L	148.3 LIHFDBA	132.8 LIHFDB	200.3 LIHFC	125.1 LIHFD	217.1 LIHFE	85.2 LIHF	244.6 LIHFG

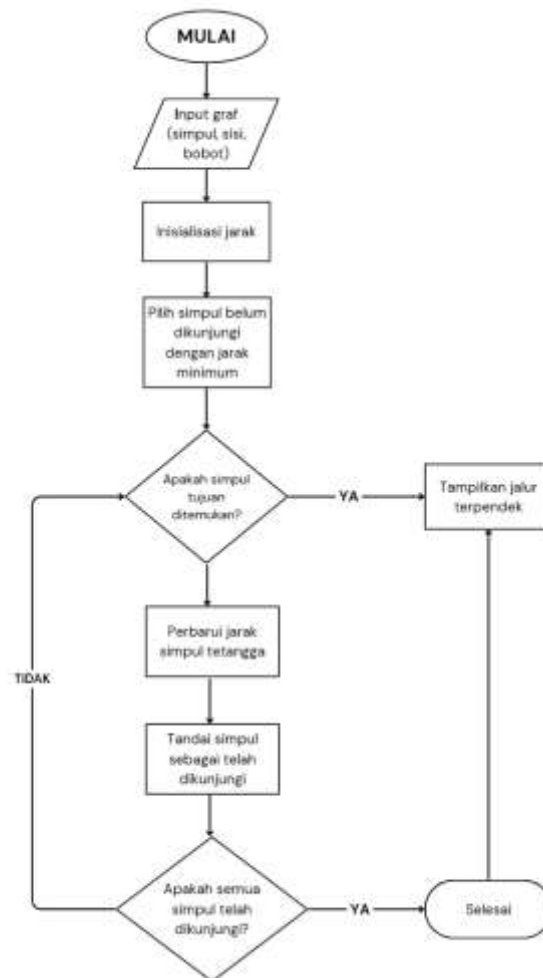
Dari tabel 2, dapat dilihat bahwa jarak terpendek diperoleh dari A ke B ke D ke F ke H ke J ke L dengan total jarak terpendek yaitu 147.9 KM.

4. Implementasi Algoritma Dijkstra

Pada bagian ini ditampilkan visualisasi berupa *flowchart* untuk menggambarkan proses kerja algoritma Dijkstra. Tujuan penyajian *flowchart* ini adalah untuk memberikan pemahaman yang lebih jelas dan sistematis mengenai tahapan utama dalam algoritma pencarian jalur terpendek. Perlu ditekankan bahwa tahap ini hanya menampilkan alur logika secara konseptual dan belum mencakup implementasi dalam kode program.



DOI: <https://doi.org/10.52362/jmijayakarta.v5i4.2002>



Gambar 5 Diagram Alur

Diagram alur pada gambar menunjukkan runtutan proses dari awal hingga akhir dalam menentukan rute terpendek pada graf. Visualisasi ini menjadi alat bantu penting untuk memahami cara kerja algoritma secara bertahap. Penjelasan tiap langkah dalam *flowchart* adalah sebagai berikut:

- Mulai
Proses dimulai untuk menjalankan algoritma pencarian jalur terpendek pada graf.
- Masukkan data graf (simpul, sisi, bobot).
Graf yang berisi titik (simpul), koneksi antar titik (sisi), dan nilai bobot atau jarak antar titik dimasukkan sebagai data input.
- Inisiatif jarak awal
Set jarak dari simpul asal ke dirinya sendiri sebagai 0, sedangkan jarak ke simpul lainnya diatur ke tak hingga (∞). Pilih simpul belum dikunjungi dengan jarak terkecil.
- Apakah simpul tujuan sudah ditemukan
Jika simpul tujuan telah tercapai, tampilkan hasil jalur terpendek dan proses selesai. Jika belum, lanjutkan ke proses berikutnya.
- Perbaharui jarak
Hitung ulang jarak ke simpul-simpul tetangga melalui jalur yang sedang dianalisis. Jika ditemukan jalur lebih pendek, lakukan pembaruan nilai jarak.



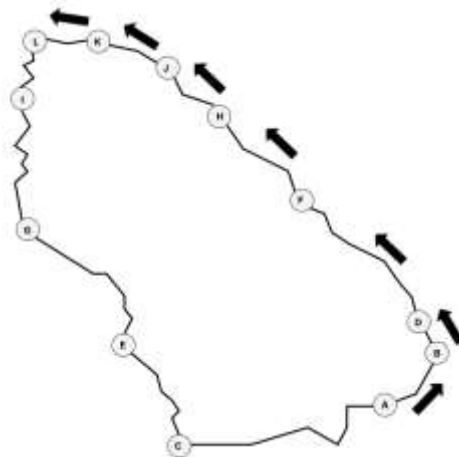
DOI: <https://doi.org/10.52362/jmijayakarta.v5i4.2002>

- f. Tandai simpul sebagai dikunjungi
Simpul yang telah diproses ditandai agar tidak diproses kembali.
- g. Apakah seluruh simpul sudah dikunjungi?
Jika semua simpul telah diperiksa, maka algoritma selesai. Jika belum, kembali ke pemilihan simpul dengan jarak minimum.
- h. Selesai
Proses algoritma selesai ketika semua simpul telah diproses atau simpul tujuan telah ditemukan. Pendekatan visual seperti ini didukung oleh [12], yang menunjukkan bahwa representasi flowchart meningkatkan pemahaman pengguna terhadap logika Dijkstra, terutama dalam konteks navigasi bergraf interaktif.

5 Kesimpulan (or Conclusion)

Berdasarkan hasil analisis, algoritma Dijkstra terbukti mampu memberikan solusi yang tepat dalam menentukan rute terpendek dari simpul awal ke simpul tujuan pada suatu jaringan graf berbobot. Dalam konteks penelitian ini, simpul A sebagai titik awal (Balige) dan simpul L sebagai titik tujuan (Merek) berhasil dihubungkan melalui rangkaian simpul dengan total jarak tempuh paling efisien.

Proses penelusuran dilakukan dengan mengikuti prinsip dasar algoritma, yaitu pemilihan simpul dengan jarak terpendek secara berurutan dan pembaruan jarak ke simpul-simpul tetangga.



Gambar 6. Visualisasi rute terdekat dari A ke L

Jalur optimal yang diperoleh adalah:

$A \rightarrow B \rightarrow D \rightarrow F \rightarrow H \rightarrow J \rightarrow L$, yang mencerminkan lintasan dengan bobot terkecil, yaitu 147,9 KM, di antara semua kemungkinan rute yang tersedia.

Hasil ini menunjukkan bahwa algoritma Dijkstra dapat digunakan secara efektif untuk menyelesaikan permasalahan pencarian jalur terpendek dalam sistem jaringan jalan wisata, dan memiliki relevansi tinggi dalam perencanaan perjalanan yang efisien. Keakuratan hasil perhitungan menunjukkan bahwa pendekatan ini mampu merepresentasikan kondisi rute secara realistis dan optimal berdasarkan struktur graf yang telah dibangun.

Dengan demikian, algoritma Dijkstra merupakan pendekatan yang andal dan aplikatif dalam menyelesaikan permasalahan optimasi jalur, khususnya dalam pengembangan sistem navigasi wisata di kawasan Danau Toba dan wilayah lainnya yang memiliki distribusi lokasi geografis yang luas dan kompleks.

Referensi (Reference)



licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

DOI: <https://doi.org/10.52362/jmijayakarta.v5i4.2002>

- [1] L. Y. Sipayung, C. R. Sinaga, and A. C. Sagala, "Application of Dijkstra's Algorithm to Determine the Shortest Route from City Center to Medan City Tourist Attractions," *J. Comput. Networks, Archit. High Perform. Comput.*, vol. 5, no. 2, pp. 648–655, 2023, doi: 10.47709/cnahpc.v5i2.2699.
- [2] S. Yuliani, M. Rozahi Istambul, and E. Angga Laksana, "Dijkstra's Algorithm to Find Shortest Path of Tourist Destination in Bandung," *Artic. Turkish J. Comput. Math. Educ.*, vol. 12, no. 8, pp. 1163–1168, 2021, [Online].
- [3] A. Lusiani, E. Sartika, A. Binarto, E. Habinuddin, and I. Azis, "Determination of the Fastest Path on Logistics Distribution by Using Dijkstra Algorithm," *Proc. 2nd Int. Semin. Sci. Appl. Technol. (ISSAT 2021)*, vol. 207, no. Issat, pp. 246–250, 2021, doi: 10.2991/aer.k.211106.039.
- [4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms 2nd version*. 2001. [Online].
- [5] V. Tiara, "Optimizing Urban Transportation Networks : Comparative Analysis of Dijkstra ' s Algorithm in Graph-Based Shortest Path Solutions," 2025.
- [6] N. M. Yalçinkaya, D. Arca, and H. Keskin Citiroglu, "Producing alternative tourism routes using network analysis to promote Seferihisar cittaslow concept," *Inf. Technol. Tour.*, 2025, doi: 10.1007/s40558-025-00319-1.
- [7] C. M. Muriuki and B. Kenduiyo, "A Multimedia Web GIS Portal for Promotion of Tourism in Kenya," *J. Geogr. Inf. Syst.*, vol. 13, no. 01, pp. 19–35, 2021, doi: 10.4236/jgis.2021.131002.
- [8] A. Kim, G. Valiant, M. Wootters, and A. Rubinstein, "CS 161 (Stanford , Winter 2024) Lecture 11 Dijkstra and Bellman-Ford Dijkstra ' s Algorithm," vol. 161, pp. 1–9, 2024.
- [9] G. Gunawan, S. Handayani, and S. Anandianskha, "Application of ant colony algorithm to optimize waste transport distribution routes in Tegal," *J. Mantik*, vol. 8, no. 1, pp. 798–807, 2024, doi: 10.35335/mantik.v8i1.5223.
- [10] T. Kien Hua and N. Abdullah, "Weighted Sum-Dijkstra's Algorithm in Best Path Identification based on Multiple Criteria," *J. Comput. Sci. Comput. Math.*, vol. 8, no. 3, pp. 107–113, 2018, doi: 10.20967/jcscm.2018.04.008.
- [11] E. Iskandar, E. Prayitno, D. Kurniawati, and A. A. A. Imron, "Development of The Nearest Tourism Determination Application using Dijkstra Algorithm," *J. Int. Conf. Proc.*, vol. 2, no. 1, 2019, doi: 10.32535/jicp.v2i1.421.
- [12] B. Marussig and U. Reif, "Surface patches with rounded corners," *Comput. Aided Geom. Des.*, vol. 97, pp. 1–15, 2022, doi: 10.1016/j.cagd.2022.102134.

