Jurnal Minfo Polgan
Volume 14, Nomor 1, Juni 2025
DOI : https://doi.org/10.33395/jmp.v14i1.14892
e-ISSN : 2797-3298
p-ISSN : 2089-9424

Implementasi Algoritma Dijkstra dalam Menentukan Rute Terpendek dari Unika St. Thomas Medan ke Lapangan Merdeka

[1]Cesia Trisani Saragih Garingging, [2] Sasmita Lumban Gaol, [3]Maria Angelina Lubis, [4] Firman Torino Sianturi, [5]Boy Mountavani Sembiring, [6]Sardo Pardingotan Sipayung
Universitas Katolik Santo Thomas, Medan , Indonesia
[1]cesiasaragih@gmail.com, [2]sasmitalumbangaol468@gmail.com,
[3]mariaangelinalubis@gmail.com, [4]firmansianturi354@gmail.com,
[5]boysembiring583@gmail.com, [6]pinsarsiphom@gmail.com

Submit : 07 Jun 2025 | **Diterima** : 15 Jun 2025 | **Terbit** : 16 Jun 2025

ABSTRAK

Penentuan rute terpendek dalam sistem informasi geografis memiliki peran penting dalam mendukung efisiensi perjalanan, terutama di kawasan perkotaan dengan tingkat kemacetan tinggi. Penelitian ini bertujuan untuk mengimplementasikan algoritma Dijkstra dalam pencarian rute optimal dari Universitas Katolik Santo Thomas Medan menuju Balai Kota Lapangan Merdeka Medan. Metode yang digunakan adalah pendekatan kuantitatif berbasis pemodelan graf, di mana simpul-simpul (nodes) merepresentasikan titik-titik persimpangan jalan dan sisi (edges) menunjukkan hubungan antar jalan yang diukur berdasarkan jarak tempuh aktual dari Google Maps. Simulasi dilakukan menggunakan jalur dengan representasi node A hingga Q, yang masing-masing menunjukkan ruas jalan tertentu: mulai dari Unika (A), melalui Jalan Setia Budi (B), Dr. Mansyur (C), Jamin Ginting (F), Kapten Patimura (G), Jendral Sudirman (I), Letjend Suprapto (K), Pemuda (L), Jendral Ahmad Yani (N), hingga mencapai tujuan akhir di Lapangan Merdeka (Q). Hasil penelitian menunjukkan bahwa lintasan dari A ke B ke C ke F ke G ke I ke K ke L ke N dan ke Q memiliki jarak terpendek sebesar 13.450 m. Implementasi algoritma Dijkstra terbukti efektif dalam menyelesaikan persoalan pemilihan rute tercepat berdasarkan parameter jarak. Temuan ini dapat diadopsi dalam pengembangan aplikasi navigasi lokal berskala kecil untuk menunjang efisiensi transportasi. Penelitian lanjutan disarankan untuk mengintegrasikan variabel waktu tempuh, kondisi lalu lintas real-time, dan preferensi pengguna guna meningkatkan akurasi serta relevansi hasil pencarian rute.

Kata Kunci: Algoritma Dijkstra, Rute Terpendek, Sistem Informasi Geografis, Google Maps, Navigasi

PENDAHULUAN

Perkembangan teknologi digital saat ini telah memengaruhi berbagai bidang kehidupan, termasuk dalam hal perencanaan perjalanan di lingkungan perkotaan. Kota Medan sebagai salah satu kota metropolitan di Indonesia menghadapi tantangan serius terkait mobilitas masyarakat, terutama karena tingginya volume kendaraan dan kompleksitas sistem jalan. Dalam konteks tersebut, kebutuhan akan sistem penunjuk arah yang mampu memberikan informasi rute tercepat atau terpendek menjadi semakin penting..Salah satu rute strategis yang sering dilalui oleh masyarakat adalah jalur antara Universitas Katolik Santo Thomas Medan dan Lapangan Merdeka. Rute ini menghubungkan kawasan pendidikan dengan pusat kota, yang sering dijadikan lokasi kegiatan sosial, ekonomi, maupun budaya. Penentuan jalur tercepat dari kedua lokasi ini menjadi penting, khususnya bagi mahasiswa, pegawai, maupun pengunjung umum yang ingin menghemat waktu dan jarak tempuh.

Untuk menyelesaikan persoalan penentuan rute terpendek, dapat digunakan pendekatan matematis berbasis teori graf. Salah satu algoritma yang sering digunakan adalah algoritma Dijkstra, yang dirancang untuk mencari lintasan minimum dari satu titik awal ke titik lainnya dalam graf berbobot. Algoritma ini bersifat deterministik, efisien secara waktu, dan mampu menghasilkan

Jurnal Minfo Polgan
Volume 14, Nomor 1, Juni 2025 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v14i1.14892 p-ISSN : 2089-9424

solusi optimal dalam konteks graf terhubung. Oleh karena itu, penerapannya banyak ditemukan dalam sistem navigasi, aplikasi transportasi, serta perencanaan jalur logistik.

Penelitian ini tidak hanya berfokus pada penerapan algoritma Dijkstra secara teoritis, tetapi juga menggabungkan data aktual dari kondisi jalan di Kota Medan. Penulis secara langsung melakukan observasi dan pemetaan titik-titik penting di sepanjang rute yang diteliti. Setiap simpul dalam graf merepresentasikan titik seperti persimpangan atau ruas jalan utama, sementara sisi antar simpul diberi bobot berdasarkan jarak tempuh yang diukur atau diestimasi dari kondisi nyata.

Perbedaan utama antara penelitian ini dengan studi sebelumnya terletak pada pendekatan lokal yang digunakan. Sebagian besar penelitian terdahulu menerapkan algoritma Dijkstra dalam konteks umum, atau pada simulasi berbasis peta global tanpa memperhitungkan karakteristik unik suatu wilayah. Dalam penelitian ini, penulis secara khusus mengambil studi kasus di Kota Medan dengan rute yang spesifik dan mempertimbangkan data geografis serta struktur jalan yang nyata. Dengan demikian, hasil penelitian diharapkan dapat lebih aplikatif dan relevan untuk digunakan oleh masyarakat setempat.

Tujuan utama dari penelitian ini adalah merancang dan mengimplementasikan sistem sederhana yang mampu menghitung rute tercepat dari Unika Santo Thomas menuju Lapangan Merdeka dengan menggunakan algoritma Dijkstra. Selain itu, penelitian ini bertujuan untuk memberikan kontribusi ilmiah dalam bidang ilmu komputer terapan, khususnya terkait pemodelan graf dan pemrosesan data spasial. Hasil penelitian ini juga dapat dijadikan sebagai dasar pengembangan sistem informasi geografis berbasis lokal atau aplikasi navigasi yang lebih cerdas dan adaptif terhadap kondisi wilayah tertentu.

Melalui penelitian ini, diharapkan pengguna jalan dapat memperoleh rekomendasi rute yang lebih efisien, dan di sisi lain, mahasiswa maupun peneliti lain dapat memahami bagaimana algoritma pencarian jalur bekerja dalam konteks dunia nyata.

TINJAUAN PUSTAKA

Berbagai studi telah membuktikan bahwa pencarian jalur terpendek merupakan topik penting dalam pengembangan sistem navigasi berbasis graf. Salah satu penelitian yang cukup berpengaruh dilakukan oleh Prasetyo dan Wulandari pada tahun 2020, yang mengembangkan sistem navigasi digital dalam lingkup kampus dengan memanfaatkan algoritma Dijkstra. Dalam pendekatan mereka, peta kampus dimodelkan sebagai graf berbobot, dan algoritma digunakan untuk menentukan jalur dengan jarak terpendek antar lokasi. Sistem tersebut terbukti efisien dalam memberikan rekomendasi jalur, tetapi implementasinya terbatas pada ruang tertutup tanpa memperhitungkan faktor-faktor eksternal seperti kondisi lalu lintas atau variasi beban jalan. Hal ini memberikan peluang bagi penelitian lain untuk mengadopsi pendekatan serupa dalam ruang terbuka dan situasi yang lebih kompleks.

Selanjutnya, Sari dan Nugroho (2021) mengevaluasi kinerja algoritma Dijkstra dan A* dalam konteks pemetaan wilayah perkotaan. Dalam penelitian ini, data peta digital digunakan sebagai basis analisis, dan perhitungan jarak antar simpul didasarkan pada informasi yang diambil dari Google Maps API. Hasil yang diperoleh menunjukkan bahwa algoritma A* unggul dalam hal kecepatan karena memanfaatkan fungsi heuristik, sementara Dijkstra lebih stabil dalam menemukan solusi jalur terpendek, terutama pada kondisi graf yang tidak rumit. Temuan ini memperkuat relevansi Dijkstra untuk diterapkan dalam konteks jalur linear antar titik yang tidak memiliki struktur topologi kompleks, seperti jalur antara kampus ke pusat kota.

Sementara itu, penelitian yang dilakukan oleh Situmorang dan timnya pada tahun 2022 mengkaji pemanfaatan algoritma Dijkstra dalam sistem evakuasi bencana di Sumatera Utara. Penelitian ini menggunakan pendekatan graf untuk merepresentasikan jaringan jalan dan mengalokasikan bobot berdasarkan faktor-faktor geografis dan kondisi lingkungan setempat. Hasilnya menunjukkan bahwa algoritma ini efektif dalam menyediakan jalur evakuasi tercepat dari area rawan ke tempat perlindungan, menunjukkan adaptabilitas tinggi dalam konteks geografis dan skenario darurat. Penelitian ini memperluas cakupan penerapan Dijkstra tidak hanya untuk navigasi rutin, tetapi juga dalam mendukung sistem respon cepat terhadap keadaan krisis.

Berdasarkan ketiga penelitian tersebut, terlihat bahwa algoritma Dijkstra memiliki keunggulan

Jurnal Minfo Polgan
Volume 14, Nomor 1, Juni 2025 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v14i1.14892 p-ISSN : 2089-9424

dalam memberikan solusi jalur optimal pada berbagai kondisi dan kebutuhan. Meskipun begitu, kebanyakan penelitian sebelumnya dilakukan pada wilayah yang terbatas atau berbasis simulasi data. Penelitian ini mencoba mengisi celah tersebut dengan menerapkan algoritma Dijkstra pada jalur nyata di Kota Medan, yaitu dari Universitas Katolik Santo Thomas menuju Lapangan Merdeka. Dengan mempertimbangkan kondisi geografis aktual dan data jarak lapangan, penelitian ini diharapkan dapat memberikan kontribusi praktis dalam pengembangan sistem navigasi berbasis data lokal yang lebih akurat dan bermanfaat bagi masyarakat perkotaan.

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.

METODE PENELITIAN

Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen, yang bertujuan untuk mengevaluasi efektivitas algoritma Dijkstra dalam menemukan rute tercepat berdasarkan kondisi spasial nyata. Model ini dianggap tepat karena masalah yang diteliti melibatkan data numerik dan pengolahan graf untuk optimasi jalur. Pendekatan eksperimental memungkinkan pengujian langsung terhadap algoritma dengan input yang dikendalikan dan hasil yang terukur, sehingga dapat digunakan untuk menghasilkan simpulan yang objektif.

Teknik Pengumpulan Data

Data yang digunakan dalam penelitian ini diperoleh melalui dua sumber, yaitu data primer dan data sekunder, yang digabungkan untuk membentuk representasi graf kota secara akurat.

a.Data Primer
Data primer dikumpulkan melalui observasi langsung pada jalur yang menghubungkan Universitas Katolik Santo Thomas dan Lapangan Merdeka. Peneliti mencatat sejumlah simpul penting seperti persimpangan jalan, ujung tikungan, dan titik temu jalur utama. Pengukuran jarak antar titik dilakukan secara manual menggunakan alat ukur dan perangkat GPS, guna memastikan akurasi data jarak. Informasi ini menjadi dasar untuk menentukan bobot antar simpul dalam graf.
b.Data Sekunder
Sumber sekunder diperoleh dari peta digital seperti Google Maps dan OpenStreetMap. Data ini digunakan sebagai referensi dalam membentuk struktur jaringan jalan serta memverifikasi letak simpul dan jalur antar titik. Selain itu, data sekunder juga membantu dalam proses visualisasi akhir jalur optimal yang dihasilkan algoritma.

Prosedur Penelitian

Langkah-langkah dalam penelitian ini disusun secara sistematis agar proses pelaksanaan dapat direplikasi dan ditelusuri kembali:
a.Penentuan Simpul dan Jaringan
Identifikasi titik-titik penting dilakukan terlebih dahulu untuk menentukan simpul dalam jaringan. Setiap titik diberi kode khusus agar memudahkan dalam pemodelan graf.
b.Pengukuran dan Pembobotan Jalur
Setiap koneksi antar simpul diukur jaraknya dan dijadikan nilai bobot dalam graf. Semua data jarak dinyatakan dalam satuan meter dan disusun dalam bentuk matriks jarak.
c.Penerapan Algoritma Dijkstra
Setelah struktur graf terbentuk, algoritma Dijkstra digunakan untuk mencari rute dengan jarak minimum dari titik awal ke tujuan. Proses perhitungan dilakukan secara manual dengan bantuan tabel atau spreadsheet, meliputi proses seleksi simpul dengan jarak terkecil dan pembaruan nilai jarak hingga mencapai simpul tujuan.
d.Visualisasi Hasil
Jalur yang diperoleh divisualisasikan pada peta berbasis digital atau media cetak, sehingga

Jurnal Minfo Polgan
Volume 14, Nomor 1, Juni 2025 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v14i1.14892 p-ISSN : 2089-9424

memudahkan interpretasi dan verifikasi visual terhadap hasil perhitungan.

Evaluasi Dan Pengujian Hasil

Evaluasi dilakukan dengan cara membandingkan jalur hasil algoritma dengan rute yang direkomendasikan oleh aplikasi peta digital seperti Google Maps. Perbandingan dilakukan untuk melihat seberapa efisien hasil algoritma dalam menemukan jalur terpendek berdasarkan struktur graf yang dibentuk. Pengujian ini juga memperhatikan aspek seperti jumlah simpul yang dilalui serta total jarak tempuh.

***Flowchart* Penelitian**

Gambar 1. Flowchart penelitian

HASIL DAN PEMBAHASAN

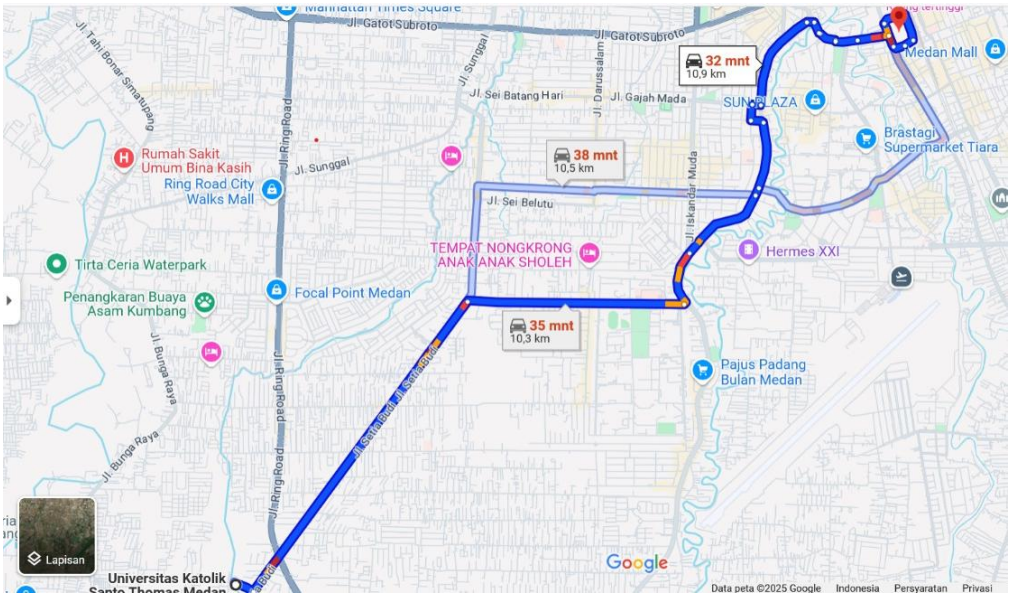
1. Pengumpulan Data

Data Lokasi

Pada tahap ini dilakukan proses pemetaan jalur dari Universitas Katolik Santo Thomas Medan menuju Lapangan Merdeka. Proses ini dimulai dengan identifikasi titik awal (node awal) dan titik tujuan akhir (node akhir), serta penentuan ruas-ruas jalan yang akan menjadi sisi (edge) penghubung antar simpul.

Node awal ditetapkan di lokasi Universitas Katolik Santo Thomas Medan, sementara node akhir ditetapkan di Lapangan Merdeka, yang menjadi titik destinasi. Setiap ruas jalan yang menghubungkan dua simpul direpresentasikan sebagai sisi pada graf dan diberi bobot berdasarkan jarak atau estimasi panjang jalan. Pemetaan jalur ini divisualisasikan terlebih dahulu pada peta kawasan Kota Medan dengan menandai simpul-simpul utama berdasarkan persimpangan atau titik temu jalan-jalan strategis.

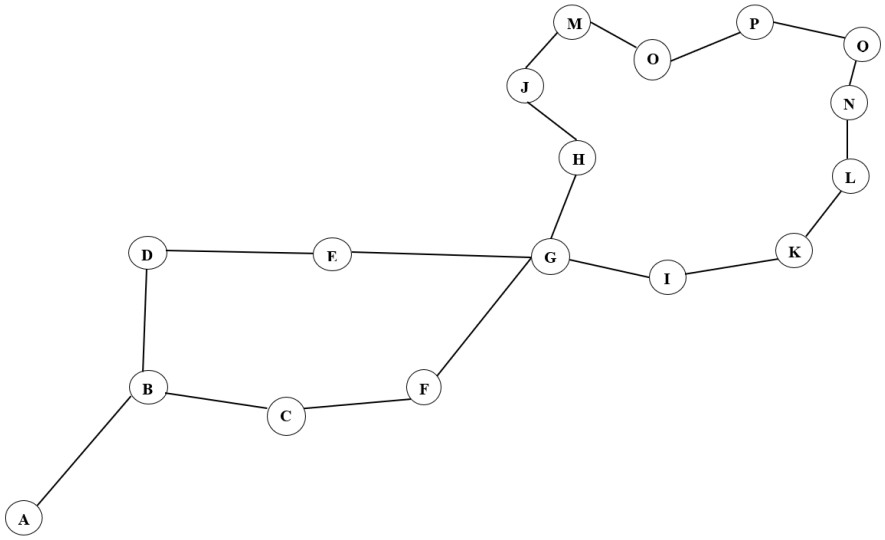
Jurnal Minfo Polgan
Volume 14, Nomor 1, Juni 2025 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v14i1.14892 p-ISSN : 2089-9424



Gambar 2. Peta dari Universitas Katolik Santo Thomas Medan ke Lapangan Merdeka

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.

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= Universitas Katolik Santo Thomas Medan
B= jl.Setia budi
C= jl.Dr.Mansyur
D= Jl.Sei Serayu
E= Jl.Abdullah Lubis
F= Jl.Jamin Ginting
G= Jl.Kapten Patimura

Jurnal Minfo Polgan
Volume 14, Nomor 1, Juni 2025 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v14i1.14892 p-ISSN : 2089-9424

H= Jl.S.Parman
I= Jl.Jendral Sudirman
J= Jl.MojoPahit
K= Jl.Letjend Suprapto
L= Jl.Pemuda
M= Jl.Gajah Mada
N= Jl.Jendral Ahmad Yani
O= Jl.kapten Maulana Lubis
P= Jl.Raden Saleh
Q= Lapangan Merdeka
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 ke dalam bentuk graf dilakukan, tahap selanjutnya adalah menetapkan bobot pada setiap sisi graf yang merepresentasikan jarak antar simpul. Bobot ini berfungsi sebagai parameter utama dalam penerapan algoritma Dijkstra untuk menentukan jalur terpendek. Penentuan bobot dilakukan dengan mengacu pada estimasi jarak nyata antar ruas jalan yang telah dipetakan sebelumnya.

Masing-masing sisi pada graf diberi nilai bobot berdasarkan panjang ruas jalan yang menghubungkan dua simpul. Sumber data jarak dapat berasal dari alat pengukur digital seperti Google Maps, aplikasi SIG (Sistem Informasi Geografis), atau perhitungan berdasarkan peta skala.

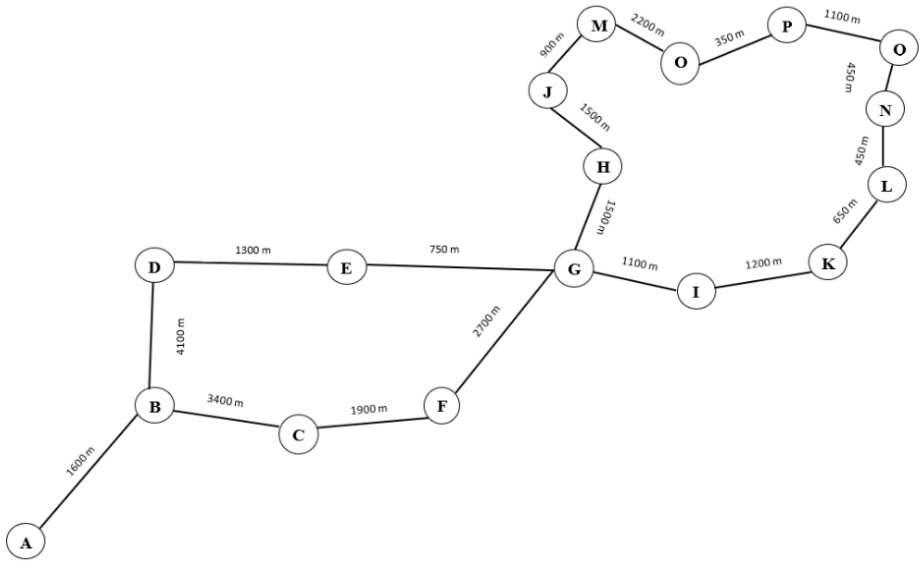
Tabel 1 berikut menyajikan rincian bobot sisi pada graf:

Tabel 1. Rincian Bobot Sisi pada Graf

No.	Simpul	Total jarak
1.	A →B	1600 m
2.	B →C	3400 m
3.	B→D	4100 m
4.	C →F	1900 m
5.	D →E	1300 m
6.	E →G	750 m
7.	F →G	2700 m
8.	G→H	1500 m
9.	G →I	1100 m
10.	H→J	1500 m
11.	I →K	1200 m
12.	J→M	900 m
13.	K →L	16000 m
14.	L →N	17000 m
15.	M→O	2200 m
16.	N→Q	450 m
17.	O→P	350 m
18.	P→Q	1100 m

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.

Jurnal Minfo Polgan
Volume 14, Nomor 1, Juni 2025 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v14i1.14892 p-ISSN : 2089-9424



Gambar 4. Representasi Graf Berbobot Jalur dari Universitas Katolik Santo Thomas Medan ke Lapangan Merdeka

3. Perhitungan Graph

Setelah penentuan bobot pada setiap sisi graf, langkah berikutnya adalah menetapkan simpul awal dan simpul tujuan. Dalam kasus ini, simpul A ditetapkan sebagai simpul awal dan simpul O sebagai simpul tujuan. Proses perhitungan dimulai dengan inisialisasi jarak awal untuk setiap simpul. Jarak dari simpul awal (A) ke dirinya sendiri diatur menjadi 0, sedangkan jarak ke semua simpul lainnya diinisialisasi dengan nilai tak hingga (∞). Berikut adalah langkah-langkah perhitungan yang akan dilakukan:

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 = ∞
Jarak M = ∞
Jarak N = ∞
Jarak O = ∞
Jarak P = ∞
Jarak Q = ∞

b. Tentukan simpul terdekat dari simpul awal.
Simpul A hanya terhubung dengan simpul B dengan jarak 1,500 m. Di sini kita melakukan update jarak dengan simpul terdekat dari simpul awal. Karena simpul awal (A) tidak memiliki simpul tetangga lain, maka ditetapkan bahwa jarak terpendek diambil dari simpul A ke B.

Jurnal Minfo Polgan
Volume 14, Nomor 1, Juni 2025 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v14i1.14892 p-ISSN : 2089-9424

c. Perhitungan selanjutnya dimulai dari titik yang memiliki jarak terpendek, yaitu simpul B. Simpul B memiliki dua simpul tetangga yang terhubung, yaitu simpul D dan simpul C.
Jarak dari simpul B ke D adalah 4,100 m.
Jarak dari simpul B ke C adalah 3,400 m.
Dari kedua simpul tersebut, diambil simpul dengan jarak paling pendek untuk dikunjungi. Dalam hal ini, simpul B ke C adalah yang memiliki jarak terpendek. Sehingga jarak terpendek selanjutnya adalah dari simpul A ke B ke C.

d. Perhitungan selanjutnya dimulai dari titik dengan jarak terpendek, yaitu simpul C.
Simpul C hanya terhubung dengan simpul F dengan jarak 1,900 m. Di sini kita melakukan update jarak dengan simpul terdekat dari simpul C. Karena simpul C tidak memiliki simpul tetangga lain, maka ditetapkan bahwa jarak terpendek diambil dari simpul A ke B ke C ke F.

e. Perhitungan selanjutnya dimulai dari titik dengan jarak terpendek, yaitu simpul F.

Simpul F hanya terhubung dengan simpul G dengan jarak 2,700 m. Di sini kita melakukan update jarak dengan simpul terdekat dari simpul F. Karena simpul F tidak memiliki simpul tetangga lain, maka ditetapkan bahwa jarak terpendek diambil dari simpul A ke B ke C ke F ke G.

f. Perhitungan selanjutnya dimulai dari titik dengan jarak terpendek, yaitu simpul G.
Simpul G memiliki dua simpul tetangga yang terhubung, yaitu simpul H dan simpul I.
Jarak dari G ke H adalah 1,500 m
Jarak dari G ke I adalah 1,100 m
Dari kedua simpul tersebut, diambil simpul dengan jarak paling pendek untuk dikunjungi. Dalam hal ini, simpul G ke I adalah yang memiliki jarak terpendek. Sehingga jalur terpendek selanjutnya adalah A ke B ke C ke F ke G ke I.

g. Perhitungan selanjutnya dimulai dari titik dengan jarak terpendek, yaitu simpul I.
Simpul I hanya terhubung dengan simpul K dengan jarak 1,200 m. Di sini kita melakukan update jarak dengan simpul terdekat dari simpul I. Karena simpul I tidak memiliki simpul tetangga lain, maka ditetapkan bahwa jalur terpendek adalah A ke B ke C ke F ke G ke I ke K.

h. Perhitungan selanjutnya dimulai dari titik dengan jarak terpendek, yaitu simpul K.
Simpul K hanya terhubung dengan simpul L dengan jarak 650 m. Di sini kita melakukan update jarak dengan simpul terdekat dari simpul K. Karena simpul K tidak memiliki simpul tetangga lain, maka ditetapkan bahwa jalur terpendek adalah A ke B ke C ke F ke G ke I ke K ke L.

i. Perhitungan selanjutnya dimulai dari titik dengan jarak terpendek, yaitu simpul L.
Simpul L hanya terhubung dengan simpul N dengan jarak 450 m. Di sini kita melakukan update jarak dengan simpul terdekat dari simpul L. Karena simpul L tidak memiliki simpul tetangga lain, maka ditetapkan bahwa jalur terpendek adalah A ke B ke C ke F ke G ke I ke K ke L ke N.

j. Perhitungan selanjutnya dimulai dari titik dengan jarak terpendek, yaitu simpul N.
Simpul N hanya terhubung dengan simpul Q dengan jarak 450 m. Di sini kita melakukan update jarak dengan simpul terdekat dari simpul N. Karena simpul N tidak memiliki simpul tetangga lain dan Q merupakan simpul tujuan, maka ditetapkan bahwa jalur terpendek adalah:

A ke B ke C ke F ke G ke I ke K ke L ke N ke Q.

Jurnal Minfo Polgan
Volume 14, Nomor 1, Juni 2025 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v14i1.14892 p-ISSN : 2089-9424

Tabel 2. Gambar tabel Perhitungan Algoritma Dijkstra

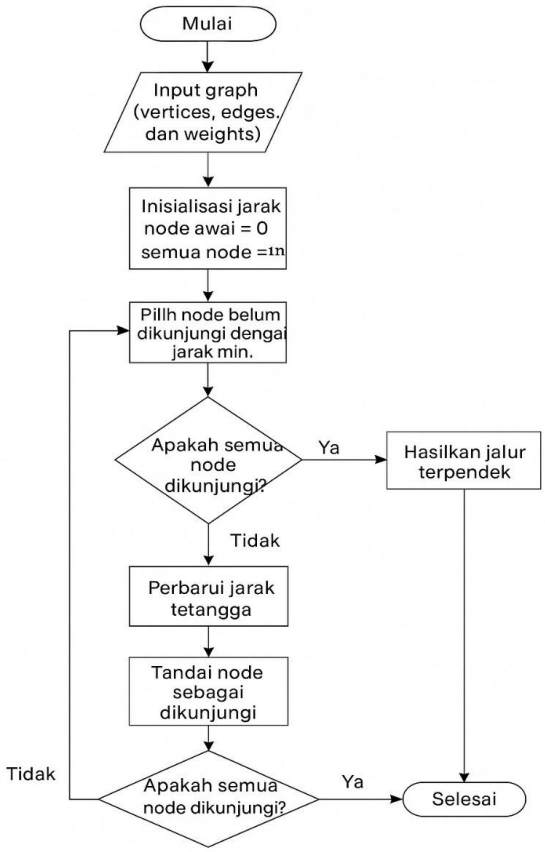
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
A	0	1600 AB	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
B	0	1600 AB	5000 ABC	5700 ABD	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
C	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
D	0	1600 AB	5000 ABC	5700 ABD	-	6900 ABCF	9600 ABCFG	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
E	0	1600 AB	5000 ABC	5700 ABD	-	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	∞	∞	∞	∞	∞	∞	∞
F	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	∞	∞	∞	∞	∞	∞
G	0	1600 AB	5000 ABC	5700 ABD	-	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	12550 ABCFGIKL	∞	∞	∞	∞	∞
H	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	12550 ABCFGIKL	∞	13000 ABCFGIKLN	∞	∞	∞
I	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	12550 ABCFGIKL	∞	13000 ABCFGIKLN	∞	∞	13450 ABCFGIKLNQ
J	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	12550 ABCFGIKL	∞	13000 ABCFGIKLN	∞	∞	13450 ABCFGIKLNQ
K	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	12550 ABCFGIKL	∞	13000 ABCFGIKLN	∞	∞	13450 ABCFGIKLNQ
L	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	12550 ABCFGIKL	∞	13000 ABCFGIKLN	∞	∞	13450 ABCFGIKLNQ
M	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	12550 ABCFGIKL	∞	13000 ABCFGIKLN	∞	∞	13450 ABCFGIKLNQ
N	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	12550 ABCFGIKL	∞	13000 ABCFGIKLN	∞	∞	13450 ABCFGIKLNQ
O	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	12550 ABCFGIKL	∞	13000 ABCFGIKLN	∞	∞	13450 ABCFGIKLNQ
P	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	12550 ABCFGIKL	∞	13000 ABCFGIKLN	∞	∞	13450 ABCFGIKLNQ
Q	0	1600 AB	5000 ABC	5700 ABD	∞	6900 ABCF	9600 ABCFG	11100 ABCFGH	10700 ABCFGI	∞	11900 ABCFGIK	12550 ABCFGIKL	∞	13000 ABCFGIKLN	∞	∞	13450 ABCFGIKLNQ

Dari tabel 2, dapat dilihat bahwa jarak terpendek diperoleh dari A ke B ke C ke F ke G ke I ke K ke L ke N ke Q dengan total jarak terpendek yaitu 13.450

4. Implementasi Algoritma Dijkstra

Pada bagian ini, akan disajikan gambaran umum mengenai penerapan algoritma Dijkstra dalam bentuk *flowchart*. Penyusunan *flowchart* ini bertujuan untuk memperlihatkan langkah-langkah utama dalam algoritma secara visual dan sistematis, sehingga memudahkan pemahaman alur logika yang digunakan untuk menentukan rute terpendek. Perlu dicatat bahwa pada tahap ini belum dilakukan implementasi dalam bahasa pemrograman apapun; fokus utama hanyalah pada penyajian skema proses algoritma secara konseptual.

Jurnal Minfo Polgan
Volume 14, Nomor 1, Juni 2025 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v14i1.14892 p-ISSN : 2089-9424



Gambar 5. Diagram alur

Diagram alur yang ditampilkan pada gambar di atas menggambarkan proses langkah demi langkah dari algoritma Dijkstra dalam menentukan jalur terpendek pada suatu graf. Visualisasi ini membantu untuk memahami mekanisme kerja algoritma secara lebih terstruktur. Adapun uraian dari setiap tahapan dalam *flowchart* tersebut akan dijelaskan sebagai berikut:

a. Mulai
   Langkah pertama dalam algoritma ini adalah menjalankan proses untuk mencari jalur terpendek dalam graf.
b. Masukkan data graf (simpul, sisi, dan bobot):
   Graf dimasukkan ke dalam sistem, terdiri dari simpul-simpul (vertices), hubungan antar simpul (edges), serta bobot atau jarak antar simpul (weights).
c. Inisialisasi nilai jarak:
   Jarak dari simpul awal ke dirinya sendiri ditetapkan sebesar 0.
   Sementara itu, jarak ke semua simpul lainnya diatur ke nilai tak hingga (∞) karena belum diketahui.
   Selanjutnya, pilih simpul yang belum dikunjungi dengan jarak paling kecil.
   Apakah simpul tujuan telah tercapai?
   Jika simpul tujuan sudah ditemukan, maka proses dilanjutkan dengan menampilkan jalur terpendek, dan algoritma pun berakhir.
   Jika belum, proses diteruskan ke langkah berikutnya.
   Perbarui jarak simpul tetangga:
   Untuk setiap simpul yang bertetangga dengan simpul saat ini, hitung ulang jaraknya dengan mempertimbangkan jalur melalui simpul yang sedang diproses. Jika jarak baru lebih pendek dibanding sebelumnya, maka jaraknya diperbarui.
   Tandai simpul sebagai telah dikunjungi:

Jurnal Minfo Polgan
Volume 14, Nomor 1, Juni 2025 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v14i1.14892 p-ISSN : 2089-9424

Simpul yang telah diproses ditandai agar tidak dipilih lagi pada iterasi selanjutnya.

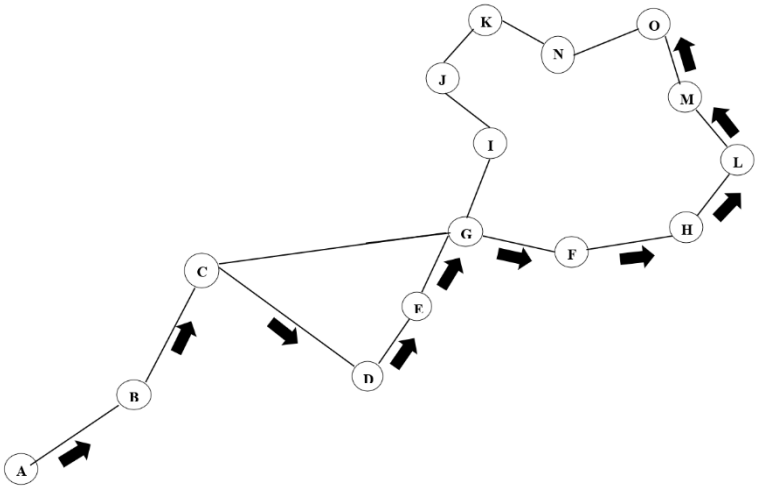
d. Apakah semua simpul telah dikunjungi?
Jika semua simpul sudah diperiksa, maka algoritma berakhir.
Jika belum, kembali ke proses pemilihan simpul dengan jarak minimum.

e. Selesai
Algoritma berakhir ketika semua simpul telah diproses atau ketika jalur terpendek menuju tujuan telah ditemukan.

5. Analisis Hasil

Penentuan rute tercepat dari **Unika St. Thomas Medan** menuju **Lapangan Merdeka** dilakukan dengan pendekatan **manual menggunakan algoritma Dijkstra**. Proses ini diawali dengan pembuatan graf berdasarkan peta jalan yang merujuk pada informasi geografis aktual, di mana titik-titik (simpul) mewakili lokasi strategis dan garis antar titik (sisi) menggambarkan jalan yang menghubungkannya, lengkap dengan jarak tempuhnya dalam satuan meter atau kilometer.

Dalam pelaksanaannya, simpul **A** ditetapkan sebagai titik awal (Unika St. Thomas Medan) dan simpul **Q** sebagai titik akhir (Lapangan Merdeka). Melalui langkah-langkah sistematis dalam algoritma Dijkstra, seperti inisialisasi jarak awal, pemilihan simpul dengan jarak terpendek, dan pembaruan jarak ke simpul tetangga, akhirnya ditemukan rute tercepat sebagai berikut:



Gambar 6. Visualisasi rute terdekat dari A ke Q

A → B → C → F → G → I → K → L → N → Q

Rangkaian simpul ini menunjukkan lintasan dengan total jarak terpendek yang dapat ditempuh dari titik awal ke tujuan. Jalur ini diperoleh berdasarkan pembandingan jarak antar simpul dan pemilihan secara berurutan terhadap simpul dengan bobot terkecil.

Hasil akhir menunjukkan bahwa jalur yang diperoleh melalui proses perhitungan ini **sejalan dengan rute tercepat yang biasanya direkomendasikan oleh aplikasi peta digital seperti Google Maps**, memperkuat validitas metode Dijkstra dalam penerapannya pada masalah pencarian rute optimal di jaringan jalan nyata.

Sebagai tambahan, visualisasi graf dan peta jalan yang digunakan juga membantu memperjelas hubungan antar titik dan mendukung proses analisis. Rute tersebut secara nyata mencerminkan efisiensi dari pendekatan algoritma dalam membantu pemecahan masalah pencarian jalur terpendek.

KESIMPULAN

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. Dalam konteks penelitian ini, simpul **A** sebagai titik awal (Unika St. Thomas Medan) dan simpul **Q** sebagai titik tujuan (Lapangan Merdeka) berhasil dihubungkan melalui rangkaian simpul dengan

total jarak 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. Jalur optimal yang diperoleh adalah **A → B → C → F → G → I → K → L → N → Q**, yang mencerminkan lintasan dengan bobot terkecil 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 transportasi atau pemetaan, dan memiliki relevansi tinggi dalam perencanaan perjalanan di dunia nyata. Keakuratan hasil menunjukkan bahwa pendekatan ini mampu menggambarkan kondisi rute yang realistis dan efisien berdasarkan struktur graf yang telah dibangun.

Dengan demikian, algoritma Dijkstra merupakan pendekatan yang andal dan aplikatif dalam menyelesaikan permasalahan optimasi jalur, khususnya dalam sistem jaringan jalan di lingkungan perkotaan.

DAFTAR PUSTAKA

Arthalia, W., & Sukmasetyan, P. (n.d.). *Implementasi algoritma Dijkstra untuk menentukan rute terpendek menuju pelayanan kesehatan*. Universitas Muhammadiyah Magelang.

Behún, M., Knežo, D., Cehlár, M., Knapčíková, L., & Behúnová, A. (2022). Recent application of Dijkstra's algorithm in the process of production planning. *Applied Sciences*, *12*(14). https://doi.org/10.3390/app12147088

Khan, A. J., & Dewangan, N. (2025). An evaluation of the Dijkstra's algorithm, Floyd's algorithm and Ant Colony Optimization. *Indian Journal of Advanced Mathematics*, *5*(1), 38–42. https://doi.org/10.54105/ijam.A1198.05010425

Jason, M. S., Valentino, A., Suryaningrum, K. M., & Yunanda, R. (2022). Dijkstra's algorithm to find the nearest vaccine location. In *Procedia Computer Science* (Vol. 216, pp. 5–12). Elsevier. https://doi.org/10.1016/j.procs.2022.12.002

Liu, J., Fu, M., Zhang, W., Chen, B., Prakapovich, R., & Sychou, U. (2023). CDT-Dijkstra: Fast planning of globally optimal paths for all points in 2D continuous space

Nubatonis, E. R. (n.d.). Analysis and design of the shortest route search application using the Dijkstra algorithm to visit the lurah office in Kupang City. *International Journal of Research in Advanced Engineering and Technology*. Retrieved from http://www.allengineeringjournal.in

Panggabean, S., Gata, W., Syarif, A. R., Rahmadani, S., & Widianto, T. (2021). *Implementasi algoritma Dijkstra untuk menentukan jalur terpendek wilayah Pasar Minggu dan STMIK Nusa Mandiri Jakarta* (Vol. 9). STMIK Nusa Mandiri Jakarta.

Stefansson, E., Biggar, O., & Johansson, K. H. (2025). Faster shortest-path algorithms using the acyclic-connected tree

Šumak, B., & Pušnik, M. (2023). Analysis of the shortest path method application in social networks. In *Frontiers in Artificial Intelligence and Applications* (Vol. 364, pp. 169–182). IOS Press.

Yosua, S., Sigalingging, C., Jipesya, J., & Jumaryadi, Y. (2021). Implementasi algoritma Dijkstra dalam pencarian klinik hewan terdekat. *Jurnal Ilmiah FIFO*, *13*(1), 85. https://doi.org/10.22441/fifo.2021.v13i1.009