

## **PENERAPAN ALGORITMA DIJKSTRA UNTUK MENENTUKAN RUTE TERPENDEK DARI RS.BINA KASIH MENUJU RS.ADAM MALIK**

**Anastasya Putri Wibowo , Khoiriyati Azmi**

[annastasyaputri17@gmail.com](mailto:annastasyaputri17@gmail.com) , [krytazmi@gmail.com](mailto:krytazmi@gmail.com)

Program Studi Matematika, Fakultas Matematika Dan Ilmu Pengetahuan Alam,  
Universitas Negeri Medan

**Yasmin Azzahra , Yohana Yulia Purba**

[yasminazzahra@gmail.com](mailto:yasminazzahra@gmail.com) , [yohanayulia08@gmail.com](mailto:yohanayulia08@gmail.com)

Program Studi Matematika, Fakultas Matematika Dan Ilmu Pengetahuan Alam,  
Universitas Negeri Medan

**Dinda Kartika**

[dindakartika@unimed.ac.id](mailto:dindakartika@unimed.ac.id)

Dosen Program Studi Matematika, Fakultas Matematika Dan Ilmu Pengetahuan Alam,  
Universitas Negeri Medan

### **Abstrak**

Permasalahan jarak dan pencarian rute merupakan tantangan umum ketika ingin mengunjungi tempat-tempat tertentu. Penelitian ini bertujuan untuk menentukan rute terpendek dari rumah sakit Bina Kasih menuju rumah sakit Adam Malik dengan menggunakan algoritma dijkstra di implementasikan dalam html dan user interface. Algoritma dijkstra digunakan untuk menguji jarak paling dekat dari satu titik ke titik lainnya dengan mengambil persimpangan atau jalan sebagai rute perjalanan. Penentuan rute terpendek bertujuan untuk mempersingkat waktu tempuh. Penelitian ini dilakukan untuk mengurangi waktu tempuh yang diperlukan untuk merujuk pasien dari Rumah Sakit Bina Kasih ke Rumah Sakit Adam Malik melalui rute yang paling efisien. Berdasarkan uji coba, Algoritma Dijkstra menunjukkan kemampuan yang baik dalam mencari rute terpendek. Algoritma ini bekerja dengan memilih sisi dengan bobot minimum yang menghubungkan node atau simpul terpilih dengan simpul lainnya. Berdasarkan pengujian yang telah dilakukan, penerapan algoritma dijkstra untuk mendapatkan rute terpendek dinilai efektif.

**Kata Kunci:** Algoritma Dijkstra, Rute Terpendek, Layanan Kesehatan.

### **Abstract**

Distance issues and route searching are common challenges when wanting to visit certain places. This study aims to determine the shortest route from Bina Kasih Hospital to Adam Malik Hospital using the dijkstra algorithm implemented in HTML and a user interface. The dijkstra algorithm is used to test the shortest distance from one point to another by taking intersections or roads as the travel route. The determination of the shortest route aims to shorten travel time. This study was conducted to reduce the travel time required to refer patients from Bina Kasih Hospital to Adam Malik Hospital through the most efficient route. Based on the trials, the dijkstra algorithm showed good ability in finding the shortest route. This algorithm works by selecting the edge with the minimum weight that connect the selected node with other nodes. Based on the test that have been conducted, the application of the dijkstra algorithm to obtain the shortest route is considered effective.

**Keywords:** Dijkstra Algorithm, Shortest Route, Healthcare Services.

## PENDAHULUAN

Rumah Sakit merupakan salah satu fasilitas penting dalam penyediaan layanan kesehatan kepada masyarakat. Dalam situasi darurat, kecepatan dan efisiensi dalam rujukan pasien dari satu rumah sakit ke rumah sakit lainnya sangatlah krusial. Salah satu rute yang sering kali diperlukan adalah dari Rumah Sakit Bina Kasih menuju Rumah Sakit Rujukan Adam Malik. Kedua rumah sakit ini berperan penting dalam sistem pelayanan kesehatan di wilayah Medan, dengan Rumah Sakit Adam Malik sebagai salah satu rumah sakit rujukan utama di Indonesia (Rachmawati & Gustin, 2020).

Rumah Sakit Bina Kasih terletak di Medan Helvetia dan dikenal dengan pelayanan medis yang komprehensif serta berbagai spesialisasi. Namun, dalam kasus-kasus yang memerlukan penanganan lebih lanjut atau spesialisasi tertentu yang tidak tersedia, pasien sering kali harus dirujuk ke Rumah Sakit Adam Malik. Rumah Sakit Adam Malik, yang juga terletak di Medan, merupakan rumah sakit umum pusat yang menyediakan layanan medis yang lebih lengkap dan canggih, termasuk fasilitas untuk penanganan penyakit kritis dan berbagai prosedur medis lanjutan.

Menentukan rute terpendek dan tercepat antara kedua rumah sakit ini sangatlah penting untuk mengurangi waktu perjalanan, yang pada gilirannya dapat meningkatkan peluang keselamatan dan kesembuhan pasien. Dalam konteks ini, penggunaan algoritma Dijkstra menjadi sangat relevan. Algoritma Dijkstra adalah salah satu metode yang efisien untuk menemukan rute terpendek dalam graf berarah yang memiliki bobot, yang dalam kasus ini dapat diterapkan untuk jaringan jalan di kota Medan. Jarak terpendek akan diperoleh dari dua atau lebih titik dari suatu graf dan nilai total yang didapat yakni yang bernilai paling kecil (Muharrom, 2020).

Transportasi darurat medis memainkan peran penting dalam sistem pelayanan kesehatan, terutama dalam situasi kritis di mana kecepatan dan efisiensi rute dapat menentukan keselamatan pasien. Rumah Sakit Bina Kasih dan Rumah Sakit Rujukan Adam Malik di Medan sering kali perlu merujuk pasien dengan cepat. Namun, kurangnya informasi mengenai rute tercepat sering menjadi kendala, mengakibatkan waktu tempuh yang lebih lama dan

kualitas layanan medis yang menurun (Wang & Wu, 2021).

Untuk mengatasi masalah ini, teknologi algoritma dalam perencanaan rute menawarkan solusi yang efektif. Salah satu algoritma yang sangat berguna dalam konteks ini adalah algoritma Dijkstra. Algoritma ini dirancang untuk menemukan rute terpendek dari satu titik ke titik lainnya dalam sebuah graf dengan bobot non-negatif. Dengan menggunakan algoritma ini, diharapkan dapat ditemukan rute tercepat dan paling efisien dari Rumah Sakit Bina Kasih ke Rumah Sakit Rujukan Adam Malik (Li et al., 2020).

Penerapan algoritma Dijkstra tidak hanya berpotensi meningkatkan efisiensi transportasi medis, tetapi juga mengurangi waktu tempuh yang sangat penting dalam situasi darurat. Hal ini dapat meningkatkan responsivitas tenaga medis, sehingga pasien dapat menerima perawatan dengan lebih cepat. Selain itu, efisiensi yang diperoleh juga dapat mengurangi biaya operasional transportasi medis (Zhang et al., 2019).

Tujuan dari penelitian ini adalah untuk mengidentifikasi berbagai rute alternatif antara Rumah Sakit Bina Kasih dan Rumah Sakit Rujukan Adam Malik, menerapkan algoritma Dijkstra untuk menentukan rute terpendek, serta mengevaluasi efektivitas rute tersebut dalam konteks waktu tempuh dan kemudahan akses. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi nyata terhadap peningkatan efisiensi dan kualitas layanan kesehatan di Medan (Huang et al., 2022).

Algoritma ini diaplikasikan pada graf berarah dimana setiap simpul dihubungkan oleh sisi yang memiliki bobot. Dengan menghitung bobot pada setiap sisi, algoritma Dijkstra mampu menentukan jalur terpendek dari simpul awal ke simpul tujuan (Saputra & Rendi, 2021). Algoritma Dijkstra juga rangkaian dari langkah-langkah yang disusun secara sistematis dan logis untuk pemecahan masalah dalam pencarian jalur terpendek dalam suatu graf berbobot (Gunawan & Napianto, 2019).

Penelitian oleh Zhang et al. (2019) juga mendukung efektivitas algoritma Dijkstra dalam perencanaan rute untuk ambulans, dengan hasil yang menunjukkan pengurangan waktu tempuh yang signifikan. Implementasi algoritma ini di sektor medis telah membantu mempercepat proses rujukan

dan mengurangi beban kerja tenaga medis. Hal ini sangat penting dalam menjaga kualitas pelayanan kesehatan, terutama di kota besar seperti Medan (Zhang et al., 2019).

Huang et al. (2022) mengungkapkan bahwa peningkatan transportasi medis melalui penerapan algoritma canggih, termasuk Dijkstra, dapat memberikan hasil yang sangat menjanjikan dalam konteks pelayanan kesehatan. Studi mereka menunjukkan bahwa penggunaan algoritma ini dapat meningkatkan efisiensi dan keandalan rute transportasi medis, yang sangat penting dalam situasi darurat. Dengan demikian, implementasi algoritma Dijkstra dapat memberikan kontribusi signifikan dalam peningkatan kualitas layanan kesehatan (Huang et al., 2022).

Penelitian ini juga bertujuan untuk memberikan panduan praktis bagi rumah sakit dalam mengelola transportasi pasien secara lebih efisien. Dengan memanfaatkan algoritma Dijkstra, rumah sakit dapat merencanakan rute transportasi yang lebih cepat dan lebih efisien, sehingga dapat menghemat waktu dan sumber daya. Hal ini juga dapat meningkatkan kepuasan pasien karena mereka dapat menerima perawatan lebih cepat (Wang & Wu, 2021).

Selain itu, penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem transportasi medis yang lebih baik di masa depan. Dengan memahami dan menerapkan teknologi algoritma, rumah sakit dapat meningkatkan kualitas layanan mereka secara keseluruhan. Penelitian ini juga dapat menjadi dasar bagi penelitian lebih lanjut dalam bidang optimasi rute transportasi medis (Li et al., 2020).

Ada beberapa penelitian sebelumnya terkait dengan algoritma dijkstra ini yaitu membahas tentang penerapan algoritma dijkstra untuk mendapatkan rute terpendek dinilai efektif sebab dengan total 20 bobot hanya memerlukan 7 bobot, yang mana hal tersebut menyatakan hanya perlu 35% bobot untuk mencapai tujuan (Cantona, et al., 2020).

Secara keseluruhan, penelitian ini bertujuan untuk meningkatkan efisiensi dan efektivitas transportasi medis antara Rumah Sakit Bina Kasih dan Rumah Sakit Rujukan Adam Malik. Dengan menerapkan algoritma Dijkstra, diharapkan dapat ditemukan rute tercepat dan paling efisien, yang

dapat mengurangi waktu tempuh dan meningkatkan kualitas layanan medis. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam peningkatan kualitas pelayanan kesehatan darurat di wilayah Medan (Zhang et al., 2019).

## KAJIAN TEORI

### Algoritma Dijkstra

Algoritma Dijkstra adalah algoritma yang digunakan untuk menemukan rute terpendek dari satu titik ke titik lain dalam sebuah graf dengan bobot non-negatif. Algoritma ini diperkenalkan oleh Edsger W. Dijkstra pada tahun 1956. Prinsip dasar dari algoritma ini adalah mencari rute dengan bobot total paling kecil dari titik awal ke titik tujuan dengan cara memperbarui jarak terpendek secara bertahap melalui proses yang dikenal sebagai relaksasi. Algoritma ini bekerja secara efisien dalam graf dengan jumlah simpul dan tepi yang besar, dan sering diterapkan dalam sistem navigasi dan jaringan komputer (Dijkstra, 1959).

Algoritma Dijkstra dikira ditemukan oleh Edsger.Wybe Dijkstra pada tahun 1959. Algoritma ini merupakan algoritma yang dapat memecahkan masalah pencarian jalur terpendek dari suatu graf pada setiap simpul yang bernilai tidak negatif. Dijkstra merupakan algoritma yang termasuk dalam algoritma greedy, yaitu algoritma yang sering digunakan untuk memecahkan masalah yang berhubungan dengan suatu optimasi. Dalam pencarian jalur terpendeknya algoritma dijkstra bekerja dengan mencari bobot yang paling minimal dari suatu graf berbobot, jarak terpendek akan diperoleh dari dua atau lebih titik dari suatu graf dan nilai total yang didapat adalah yang bernilai paling kecil (Bunaen, et.al.,2022).

### Graf Berbobot

Dalam konteks algoritma Dijkstra, graf berbobot adalah struktur data yang terdiri dari simpul (*vertices*) dan tepi (*edges*) di mana setiap tepi memiliki nilai (bobot) yang mewakili biaya atau jarak antara dua simpul. Bobot pada tepi harus bernilai non-negatif agar algoritma Dijkstra dapat berfungsi dengan baik. Graf berbobot ini sering digunakan untuk memodelkan jaringan jalan, jaringan komputer, dan berbagai sistem lain di mana hubungan antara dua titik memiliki nilai tertentu yang perlu dioptimalkan (Cormen et al., 2009).

## Teori Kompleksitas

Kompleksitas waktu dari algoritma Dijkstra, ketika diimplementasikan dengan menggunakan struktur data seperti tumpukan prioritas (*priority queue*), adalah  $O(V^2)$  untuk graf dengan  $V$  simpul jika menggunakan matriks ketetanggaan. Namun, kompleksitas ini dapat ditingkatkan menjadi  $O((V + E) \log V)$  dengan menggunakan tumpukan prioritas berbasis binomial atau *Fibonacci heap*, di mana  $E$  adalah jumlah tepi dalam graf. Hal ini menjadikan algoritma Dijkstra sangat efisien untuk digunakan dalam graf yang besar dan kompleks (Cormen et al., 2009).

## Optimasi Rute dalam Transportasi Medis

Optimasi rute adalah proses menemukan rute paling efisien untuk mencapai tujuan tertentu. Dalam konteks transportasi medis, optimasi rute bertujuan untuk mengurangi waktu tempuh sehingga pasien dapat menerima perawatan dengan cepat. Algoritma Dijkstra dapat digunakan untuk menentukan rute terpendek dalam situasi darurat, seperti ambulans yang perlu mencapai rumah sakit dengan cepat. Penelitian menunjukkan bahwa penerapan algoritma ini dalam sistem transportasi medis dapat secara signifikan mengurangi waktu tempuh dan meningkatkan responsivitas layanan medis (Zhang et al., 2019; Wang & Wu, 2021).

## Sistem Informasi Geografis (SIG)

Sistem Informasi Geografis (SIG) adalah sistem yang dirancang untuk mengumpulkan, menyimpan, menganalisis, dan menampilkan data spasial atau geografis. Dalam konteks optimasi rute, SIG dapat digunakan untuk memvisualisasikan rute, memetakan jalan, dan menghitung jarak antara titik-titik tertentu. Integrasi algoritma Dijkstra dengan SIG dapat meningkatkan akurasi dan efisiensi dalam menentukan rute terpendek, terutama dalam lingkungan perkotaan yang kompleks (Huang et al., 2022).

## Aplikasi Nyata Algoritma Dijkstra

Algoritma Dijkstra telah digunakan secara luas dalam berbagai aplikasi nyata. Dalam transportasi, algoritma ini membantu dalam sistem navigasi kendaraan dan optimasi rute logistik. Misalnya, sistem GPS modern menggunakan algoritma Dijkstra untuk memberikan rute tercepat kepada pengguna. Dalam jaringan komputer, algoritma ini digunakan untuk menemukan jalur terpendek bagi pengiriman paket data. Penerapan algoritma ini

dalam sektor kesehatan, seperti perencanaan rute ambulans, menunjukkan hasil yang sangat menjanjikan dalam meningkatkan efisiensi dan responsivitas layanan (Li et al., 2020).

## Teori Pengambilan Keputusan

Teori pengambilan keputusan berkaitan dengan proses memilih tindakan terbaik di antara beberapa alternatif. Dalam konteks penelitian ini, pengambilan keputusan melibatkan pemilihan rute terpendek dan paling efisien untuk ambulans. Algoritma Dijkstra menyediakan kerangka kerja yang sistematis dan terstruktur untuk membantu pengambil keputusan dalam menentukan rute optimal berdasarkan bobot atau jarak dalam graf, yang selanjutnya dapat diterapkan dalam skenario nyata untuk meningkatkan efisiensi transportasi medis (Wang & Wu, 2021).

## Efisiensi Operasional

Efisiensi operasional mengacu pada kemampuan sebuah sistem untuk memaksimalkan output dengan meminimalkan input, seperti waktu, biaya, dan sumber daya. Dalam transportasi medis, efisiensi operasional sangat penting untuk memastikan bahwa pasien dapat dirujuk dengan cepat dan aman. Algoritma Dijkstra membantu meningkatkan efisiensi operasional dengan menyediakan metode yang efektif untuk menemukan rute tercepat, yang mengurangi waktu tempuh dan biaya operasional serta meningkatkan kepuasan pasien dan keselamatan (Zhang et al., 2019).

## Keandalan Sistem

Keandalan sistem merujuk pada kemampuan sistem untuk berfungsi dengan baik di bawah kondisi yang berbeda. Dalam konteks transportasi medis, keandalan sistem sangat penting untuk memastikan bahwa rute yang dipilih tetap optimal meskipun ada perubahan kondisi lalu lintas atau kejadian tak terduga lainnya. Algoritma Dijkstra dapat meningkatkan keandalan sistem transportasi medis dengan menyediakan rute alternatif yang efisien dan adaptif terhadap perubahan kondisi jalan dan lalu lintas (Huang et al., 2022).

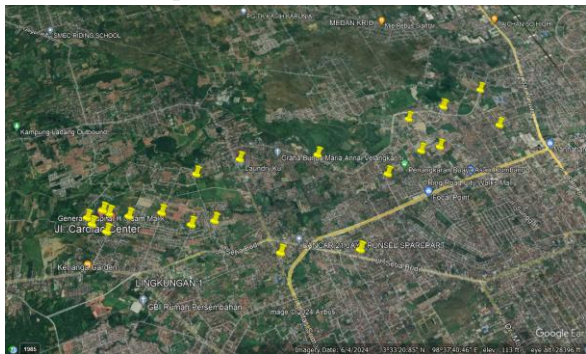
## Rute Terpendek

Rute terpendek (*shortest path*) dahulu hampir tidak dianggap. Jalur terpendek didefinisikan sebagai masalah kombinatorikal dalam grafik berbobot terbatas. Persoalan rute terpendek merupakan entitas yang mempresentasikan rute dari suatu

vertex ke vertex lainnya (Firwanda, Prianto, & dkk, 2021).

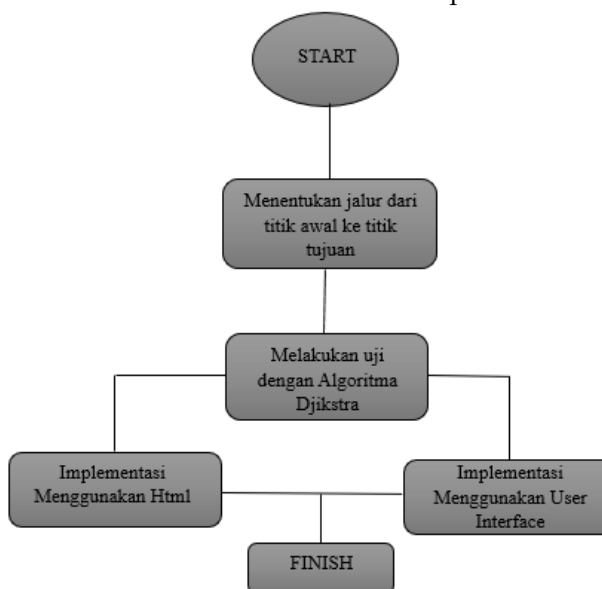
## METODE

Pada bagian ini dijelaskan bahwa metode pada penelitian ini menggunakan studi literatur untuk memperoleh data yang diperlukan melalui referensi seperti buku, jurnal, artikel, maupun pencarian disitus internet. Pengumpulan data dilakukan untuk mengumpulkan dan menganalisa data seperti informasi jarak menuju Rumah Sakit Adam Malik yang didapat dari Google Maps. Selanjutnya pengolahan data yang sesuai dengan pendekatan yang dilakukan yang akan menghasilkan informasi bagi para pembaca yang sesuai dengan tujuan penelitian. Berikut adalah titik lokasi yang ditunjukkan pada gambar 1. Menggunakan google earth dan merupakan data sekunder untuk mencari jarak antar setiap titik :



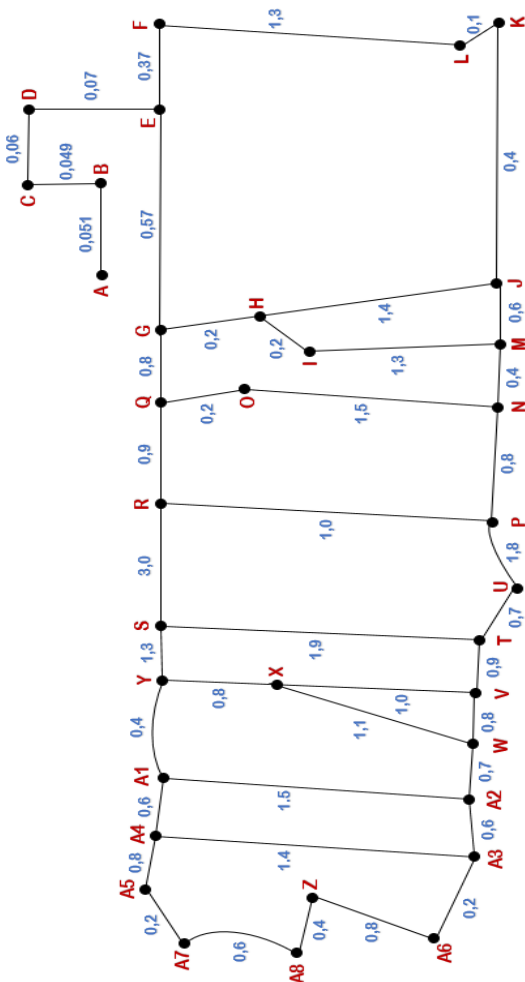
Gambar 1. Titik-Titik yang digunakan untuk penelitian.

Berikut adalah flowchart dari metode penelitian :



## HASIL DAN PEMBAHASAN

Penelitian ini memanfaatkan algoritma dijkstra yang dapat digunakan dalam perencanaan rute pada 34 jalan/persimpangan jalan dan menunjukkan hasil yang memuaskan berupa koefisien keoptimalannya. Algoritma dijkstra mampu menemukan rute terpendek dengan baik dan membantu masyarakat dengan meminimalkan waktu di perjalanan pada saat waktu rujukan. Dalam penelitian ini titik-titik dianggap sebagai persimpangan antar jalan/jalan itu sendiri yang digrafikan di google maps. Berikut merupakan graf dari hasil pemetaan melalui google maps.



Gambar 2. Graf dari titik google maps.

Pada Tabel 1. Berikut berisikan tujuan dari perjalanan beserta titik awal dan keterangan simpang jalan/jalan yang digunakan sebagai acuan dari penelitian ini

Tabel 1. Keterangan Alur Perjalanan

Simbol	Representasi
A	RSU. Bina Kasih
B	Jl. Pinang Baris Medan
C	Simp. Jl. Pinang Baris Medan
D	Simp. Jl. Pinang Baris Medan
E	Simp. Jl. Tahi Bonar Simatupang
F	Simp. Jl. Amal/ Jl. Tahi Bonar Simatupang
G	Jl. Sunggal/Jl. PDAM Tirtanadi
H	Simp. Jl. Tapian Nauli Ps. I/ Jl. Sunggal
I	Simp. Jl. Tanah Tinggi
J	Simp. Jl. Sunggal/Jl. Ringroad
K	Jl. Ringroad
L	Simp. Jl. Amal/ Jl. Ringroad
M	Jl. Tapian Nauli Ps. I / Jl. Ringroad
N	Jl. Ps. III Tapian Nauli
O	Simp. Jl. Ps. III Tapian Nauli
P	Simp. Jl. Kenanga Raya
Q	Jl. PDAM Tirtanadi
R	Simp. Jl. Bunga Asoka/ Jl. Bunga Raya
S	Jl. Flamboyan Raya
T	Jl. Melati Raya
U	Simp. Jl. Setia Budi
V	Simp. Jl. Lizadri Putra
W	Simp. Jln. Stella Raya/ Jl. Setia Budi
X	Simp. Jln. Stella Raya/ Jl. Lizadri Putra
Y	Jl. Seroja VII
A1	Seroja VIII
A2	Jl. Bunga Rinte
A3	Simp. Jl. Bunga Ncole Raya/ Jl. Jamin Ginting
A4	Jl. Petunia Raya
A5	Simp. Jl. Petunia Raya
A6	Sip. Jl. Bunga Lau/Jl. Jamin Ginting
A7	Jl. Petunia IV/ Jl. Petunia Raya
A8	Jl. Cardiac Centre
Z	RSU. Adam Malik/Titik Tujuan

Dalam menentukan jalur terpendek dari titik awal ke titik lainnya dapat ditentukan dengan perhitungan titik dari jarak terpendek yang dimulai dari bobot terendah.

Jarak yang diperoleh pada saat menentukan jarak terpendek dari Rumah Sakit Bina Kasih menuju Rumah Sakit Adam Malik begitupun sebaliknya.

A → B : 0.051	T → V : 0.9
B → C : 0.049	V → W : 0.8
C → D : 0.06	W → A2 : 0.7
D → E : 0.07	Y → A1 : 0.4
E → G : 0.57	A1 → A4 : 0.6
G → Q : 0.8	A2 → A3 : 0.6
Q → R : 0.9	A3 → A6 : 0.2
R → S : 3.0	A4 → A5 : 0.8
R → P : 1.0	A5 → A7 : 0.2
P → U : 1.8	A7 → A8 : 0.6
S → Y : 1.3	A6 → Z : 0.8
U → T : 0.7	A8 → Z : 0.4

Pada gambar dibawah ini berisikan perhitungan dijkstra dengan excel jarak dari graf yang terhubung, jarak suatu titik ke titik lainnya (KM) dan ~ menunjukkan titik tersebut tidak ada kaitannya karena tidak ada sisi yang terhubung ke titik lainnya.

	A	B	C	D	E	F	G	H	I
A	0	0.051 <sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>
B	A*	A*	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>
C		A*	B*	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>
D			B*	C*	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>
E				C*	D*	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>	<sup>N</sup>
F					D*	E*	E*	<sup>N</sup>	<sup>N</sup>
G						E*	E*	<sup>N</sup>	<sup>N</sup>
H							E*	G*	<sup>N</sup>
I								G*	H*
Q									H*
O									
L									



Gambar 3. Perhitungan Dijkstra

Berikut merupakan link Excel Dijkstra Lengkap dalam google drive :

<https://drive.google.com/drive/folders/1qwyqAFCy9nE0DLrNUJ1WJ8fCqsZFvJN?usp=sharing>

Dari jarak-jarak yang diperoleh diatas dapat ditentukan jarak yang memiliki bobot terkecil, lalu dijumlahkan.

$$\begin{aligned}
 A \rightarrow Z &= A \rightarrow B + B \rightarrow C + C \rightarrow D + C \rightarrow D + D \rightarrow E \\
 &+ E \rightarrow G + G \rightarrow Q + Q \rightarrow R + R \rightarrow S \\
 &+ S \rightarrow Y + Y \rightarrow A1 + A1 \rightarrow A4 + A4 \rightarrow A5 \\
 &+ A5 \rightarrow A7 + A7 \rightarrow A8 + A7 \rightarrow A8 + A8 \rightarrow Z = 9.80
 \end{aligned}$$

Setelah dilakukan perhitungan untuk menentukan setiap rute terpendek dimana jarak rute terpendek dari graf tersebut adalah 9.80, selanjutnya dibuat tabel penentuan rute terpendek pada tabel 2 untuk setiap perjalanan dari titik awal yaitu Rumah Sakit Bina Kasih.

Tabel 2. Penentuan Rute Terpendek

No	Rute	Jarak (kilometer)	Titik
1.	RSU. Bina Kasih - Jl. Pinang Baris	0.051	$A \rightarrow B$
2	Jl. Pinang Baris - Simp. I Jl. Pinang Baris Medan	0.049	$B \rightarrow C$

3	Simp. I Jl. Pinang Baris Medan - Simp. II Jl. Pinang Baris Medan	0.06	$C \rightarrow D$
4	Simp. II Jl. Pinang Baris Medan - Simp. Jl. Tahi Bonar Simatupang	0.07	$D \rightarrow E$
5	Simp. Jl. Tahi Bonar Simatupang - Jl. Sunggal/Jl. PDAM Tirtanadi	0.57	$E \rightarrow G$
6	Jl. Sunggal/Jl. PDAM Tirtanadi - Jl. PDAM Tirtanadi	0.8	$G \rightarrow Q$
7	Jl. PDAM Tirtanadi - Simp. Jl. Bunga Asoka/ Jl. Bunga Raya	0.9	$Q \rightarrow R$
8	Simp. Jl. Bunga Asoka/ Jl. Bunga Raya - Jl. Flamboyan Raya	3	$R \rightarrow S$
9	Jl. Flamboyan Raya - Jl. Seroja VII	1.3	$S \rightarrow Y$
10	Jl. Seroja VII - Seroja VIII	0.4	$Y \rightarrow A1$
11	Seroja VIII - Jl. Petunia Raya	0.6	$A1 \rightarrow A4$
12	Jl. Petunia Raya - Simp. Jl. Petunia Raya	0.8	$A4 \rightarrow A5$
13	Simp. Jl. Petunia Raya - Jl. Petunia IV/ Jl. Petunia Raya	0.2	$A5 \rightarrow A7$
14	Jl. Petunia IV/ Jl. Petunia Raya - Jl. Cardiac Centre	0.6	$A7 \rightarrow A8$
15	Jl. Cardiac Centre - RSU. Adam Malik/Titik Tujuan	0.4	$A8 \rightarrow Z$

Berdasarkan hasil yang didapat dari pencarian rute terpendek, dapat ditemukan 15 rute terpendek yang dilalui untuk menuju titik-titik yang diasumsikan sebagai perjalanan dengan titik awal Rumah Sakit Bina Kasih. Setelah mendapatkan hasil di atas maka akan di implementasikan hasil perancangan menggunakan bahasa pemrograman. Bahasa pemrograman yang dipakai yaitu bahasa HTML. Berikut merupakan kode hasil implementasi dari algoritma dijkstra dalam bahasa pemrograman Html.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Algoritma Dijkstra</title>
  <link rel="stylesheet" type="text/css" href="css/style.css">

```

```

<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
/>
</head>
<body>
  <!--loader-->
  <div class="bg-loader">
    <div class="loader"></div>
  </div>
  <!--header-->
  <div class="medsos">
    <div class="container">
      <ul>
        <li><a href="#"><i class="fa-brands fa-square-instagram"></i></a></li>
        <li><a href="#"><i class="fa-brands fa-youtube"></i></a></li>
        <li><a href="#"><i class="fa-brands fa-twitter"></i></a></li>
      </ul>
    </div>
  </div>
  <header>
    <div class="container">
      <h1><a href="index.html">MENCARI RUTE TERPENDEK DENGAN ALGORITMA DIJKSTRA</a></h1>
      <ul>
        <li><a href="index.html">HOME</a></li>
        <li><a href="about.html">ABOUT</a></li>
        <li class="active"><a href="dijkstra.html">DIJKSTRA ALGORITHM</a></li>
        <li><a href="contact.html">CONTACT</a></li>
      </ul>
    </div>
  </header>
  <!-- label -->
  <section class="label">
    <div class="container">
      <p>Home / Dijkstra Algorithm</p>
    </div>
  </section>
  <div id="pembungkus">
    <div class="wrapper">
      <h1 class="text-center">Mencari Rute Terpendek</h1>
      <div class="centering">
        <form method="post" name="formDijkstra"
onsubmit="findShortestPath(); return false;">
          <label>Titik Awal</label><br>
          <input type="text" name="startNode"
oninput="this.value = this.value.toUpperCase()"
placeholder="Start Node" required="1"
class="input-text"><br>
          <label>Titik Tujuan</label><br>
          <input type="text" name="endNode"
oninput="this.value = this.value.toUpperCase()"

```

```

placeholder="End Node" required="1"
class="input-text"><br>
          <button type="submit"
class="button">Jalankan</button>
        </form>
      <div>
        Hasil Jalur dan Jarak Terpendek<br>
        <span><textarea id="hasilJalurDijkstra" readonly="1"
class="hasil-input"></textarea></span></div>
      </div>
    </div>
    <script>
      class Dijkstra {
        constructor(simpul, graf) {
          this(simpul, graf);
        }
        mencari_rute(start, end) {
          let unvisitedsimpul = {};
          for (let n of this(simpul)) {
            unvisitedsimpul[n] = Infinity;
          }
          unvisitedsimpul[start] = 0;
          let visitedsimpul = {};
          let parents = {};
          while (Object.keys(unvisitedsimpul).length > 0) {
            let min_vertex =
Object.keys(unvisitedsimpul).reduce((a, b) =>
unvisitedsimpul[a] < unvisitedsimpul[b] ? a : b);
            if (unvisitedsimpul[min_vertex] ===
Infinity) {
              break; // Tidak ada simpul yang dapat
dijangkau
            }
            if (min_vertex === end) {
              visitedsimpul[min_vertex] =
unvisitedsimpul[min_vertex];
              break;
            }
            for (let [neighbour, weight] of
Object.entries(this.graf[min_vertex] || {})) {
              if (neighbour in visitedsimpul) {
                continue;
              }
              let jalur_baru =
unvisitedsimpul[min_vertex] + weight;
              if (jalur_baru <
unvisitedsimpul[neighbour]) {
                unvisitedsimpul[neighbour] =
jalur_baru;
                parents[neighbour] = min_vertex;
              }
            }
          }
        }
      }
    </script>
  </div>

```



```

        visited_simpul[min_vertex] =
unvisited_simpul[min_vertex];
        delete unvisited_simpul[min_vertex];
    }
    return [parents, visited_simpul];
}
static jalur_dilewati(parents, start, end) {
    let jalur = [end];
    while (true) {
        let key = parents[jalur[0]];
        jalur.unshift(key);
        if (key === start) {
            break;
        }
    }
    return jalur;
}
static cetak_jalur_dan_jarak(jalur, graf) {
    let total_jarak = 0;
    let hasil = [];
    for (let i = 0; i < jalur.length - 1; i++) {
        let simpul_asal = jalur[i];
        let simpul_tujuan = jalur[i + 1];
        let jarak =
graf[simpul_asal][simpul_tujuan];
        hasil.push(`${simpul_asal}
        ${simpul_tujuan} : ${jarak}`);
        total_jarak += jarak;
    }
    hasil.push(Total jarak:
    `${total_jarak.toFixed(2)}`);
    return hasil.join("\n");
}
const input_simpul = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H',
'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W',
'X', 'Y', 'A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'Z'];
const input_graf = {
    'A': { 'B': 0.051 },
    'B': { 'A': 0.051, 'C': 0.049 },
    'C': { 'B': 0.049, 'D': 0.06 },
    'D': { 'C': 0.06, 'E': 0.07 },
    'E': { 'D': 0.07, 'F': 0.37, 'G': 0.57 },
    'F': { 'E': 0.37, 'L': 1.3 },
    'G': { 'E': 0.57, 'H': 0.2, 'Q': 0.8 },
    'H': { 'G': 0.2, 'I': 0.2, 'J': 1.4 },
    'I': { 'H': 0.2, 'M': 1.3 },
    'J': { 'H': 1.4, 'M': 0.4, 'K': 0.4 },
    'K': { 'J': 0.4, 'L': 0.1 },
    'L': { 'F': 1.3, 'K': 0.1 },
    'M': { 'I': 1.3, 'J': 0.6, 'N': 0.4 },
    'N': { 'M': 0.4, 'O': 1.5, 'P': 0.8 },
    'O': { 'N': 1.5, 'Q': 0.2 },
    'P': { 'N': 0.8, 'R': 1.0, 'U': 1.8 },
    'Q': { 'G': 0.8, 'O': 0.2, 'R': 0.9 },
    'R': { 'P': 1.0, 'Q': 0.9, 'S': 3.0 },
    'S': { 'R': 3.0, 'T': 1.9, 'Y': 1.3 },

```

```

    'T': { 'S': 1.9, 'U': 0.7, 'V': 0.9 },
    'U': { 'P': 1.8, 'T': 0.7 },
    'V': { 'T': 0.9, 'X': 1.0, 'W': 0.8 },
    'W': { 'X': 1.1, 'V': 0.8, 'A2': 0.7 },
    'X': { 'V': 1.0, 'W': 1.1, 'Y': 0.8 },
    'Y': { 'S': 1.3, 'A1': 0.4 },
    'A1': { 'Y': 0.4, 'A2': 1.5, 'A4': 0.6 },
    'A2': { 'W': 0.7, 'A1': 1.5, 'A3': 0.6 },
    'A3': { 'A2': 0.6, 'A4': 1.4, 'A6': 0.2 },
    'A4': { 'A1': 0.6, 'A3': 1.4, 'A5': 0.8 },
    'A5': { 'A4': 0.8, 'A7': 0.2 },
    'A6': { 'A3': 0.2, 'Z': 0.8 },
    'A7': { 'A5': 0.2, 'A8': 0.6 },
    'A8': { 'A7': 0.6, 'Z': 0.4 },
    'Z': { 'A6': 0.8, 'A8': 0.4 }
};
// Menambahkan koneksi dua arah
function addTwoWayConnections(graph) {
    for (let node in graph) {
        for (let neighbor in graph[node]) {
            if (!graph[neighbor]) {
                graph[neighbor] = {};
            }
            if (!graph[neighbor][node]) {
                graph[neighbor][node] =
graph[node][neighbor];
            }
        }
    }
    addTwoWayConnections(input_graf);
    function findShortestPath() {
        const startNode =
document.forms["formDijkstra"]["startNode"].value;
        const endNode =
document.forms["formDijkstra"]["endNode"].value;
        const dijkstra = new Dijkstra(input_simpul,
input_graf);
        const [parents, visited_simpul] =
dijkstra.mencari_rute(startNode, endNode);
        const jalur = Dijkstra.jalur_dilewati(parents,
startNode, endNode);
        let hasil = "";
        if (endNode in visited_simpul) {
            hasil += 'Jalur terpendek dari simpul ' +
startNode + ' ke ' + endNode + ' adalah ' + jalur.join(' -
> ') + '\n';
            hasil += 'Dengan bobot = ' +
visited_simpul[endNode] + '\n\n';
            hasil += Dijkstra.cetak_jalur_dan_jarak(jalur,
input_graf);
        } else {
            hasil += 'Tidak ada jalur dari ' + startNode + '
ke ' + endNode;
        }
    }
}

```

```

document.getElementById("hasilJalurDijkstra").value
= hasil;
}
</script>
<footer>
  <div class="container">
    <small>Copyright &copy; 2024 - Kelompok 1. All
Rights Reserved.</small>
  </div>
</footer>
<script                                type="text/javascript"
src="js/script.js"></script>
</body>
</html>

```

Dibawah ini merupakan hasil dari program yang sudah dibuat dapat dilihat pada gambar 4. Program ini sudah di buat dalam program user interface dimana program ini bisa menghasilkan jarak terpendek hanya dengan memanggil titik awal dan titik tujuan. Setelah itu program akan menghasilkan keluaran berupa titik mana saja yang dilalui dan jarak terpendek dari titik awal ke titik tujuan.

Gambar 4. Hasil Program User Interface

Berikut Link dari program diatas :

<https://penentuanruteterpendek-algoritmadijkstra.000webhostapp.com/>

## PENUTUP

### SIMPULAN

Penelitian ini berhasil menentukan rute terpendek dari Rumah Sakit Bina Kasih menuju Rumah Sakit Rujukan Adam Malik dengan menggunakan algoritma dijkstra. Algoritma ini terbukti efektif dalam mengoptimalkan waktu tempuh melalui pemilihan sisi dengan bobot minimum yang menghubungkan simpul-simpul pada jaringan jalan.

Penggunaan algoritma dijkstra dapat secara signifikan mengurangi waktu yang diperlukan untuk merujuk pasien, menunjukkan bahwa algoritma ini merupakan solusi yang efisien untuk permasalahan jarak dan pencarian rute dalam konteks layanan kesehatan. Hasil penelitian ini mendukung penggunaan algoritma dijkstra sebagai alat yang handal untuk menentukan rute terpendek dalam situasi darurat medis.

## DAFTAR PUSTAKA

- Aldy Cantona, F. . (2020). Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta. *Jurnal Teknologi dan Manajemen Informatika*, 27-34.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms (3rd ed.)*. MIT Press.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269-271.
- Firwanda, a. y., Prianto, c., & dkk. (2021). Penentuan Rute Terpendek Lokasi Badan Pusat Statistik Kota Bandung Dengan Algoritma Dijkstra. *Jurnal Teknik Informatika*, 37-45.
- Gunawan, R. D., & Napianto, R. (2019). Implementation of Dijkstra's Algorithm in Determining the Shortest Path (Case Study: Specialist Doctor Search in Bandar Lampung). *International Journal of Information Systems and Computer Science*, 3(3), 98-106.
- Huang, R., Lee, D., & Chen, Y. (2022). Enhancing Medical Transportation through Advanced Algorithms. *International Journal of Healthcare Management*, 15(1), 32-41.
- Li, X., Tang, L., Zhao, Y., & Ma, H. (2020). Applications of Dijkstra's Algorithm in Transportation and Logistics. *Journal of Transportation Systems*, 7(3), 102-114.
- Muharrom, M. (2020). Implementasi Algoritma Djistrak dalam Penentuan Jalur Terpendek Studi Kasus Jarak Tempat Kuliah Terdekat. *ndonesian Journal of Business Intelligence*, 25-30.
- Maria Chatrin Bunaen, H. P. (213-223). Penerapan Algoritma Djistrak Untuk Menentukan Rute Terpendek Dari Pusat Kota Surabaya ke

Tempat Bersejarah. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 2020.

- Rachmawati, D., & Gustin, L. (2020). Analysis of Dijkstra's Algorithm and A Algorithm in Shortest Path Problem. *Journal of Physics: Conference Series*, 1566(1). doi:10.1088/1742-6596/1566/1/012061.
- SAPUTRAMA, R., & HARTATIANA, H. (2021). APLIKASI ALGORITMA DIJKSTRA UNTUK MENENTUKAN RUTE TERPENDEK DARI KAMPUS A KE B UIN RADEN FATAH. *E-JURNAL MATEMATIKA*.
- Wang, S., & Wu, X. (2021). Optimization of Emergency Medical Services Using Dijkstra's Algorithm. *Health Informatics Journal*, 27(4), 544-558.
- Zhang, Y., Wang, H., & Liu, Z. (2019). Efficient Route Planning for Ambulances: A Case Study Using Dijkstra's Algorithm. *Journal of Emergency Medical Services*, 35(2), 89-98.