International Journal Software Engineering and Computer Science (IJSECS)
6 (2), 2026, 600-614
Published Online August 2026 in IJSECS (http://www.journal.lembagakita.com/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v6i2.7604.

RESEARCH ARTICLE **Open Access**

Dijkstra Algorithm-Based Shortest Path Optimization for Multi-Destination Tourism Routes in Samosir Regency: A Google Maps-Driven Case Study

Yesy Simanjuntak [1*], Fadillah Amanah [2], Suvriadi Panggabean [3].

[1*,2,3] Department of Computer Science, Faculty of Mathematics and Natural Sciences, State University of Medan, Medan City, North Sumatra Province, Indonesia.

**Corresponding author*: yesysmjtk.4233550045@mhs.unimed.ac.id

Received: June 17, 2026; Accepted: June 28, 2026; Published: August 1, 2026.

Abstract: Samosir Regency offers various natural and cultural tourism destinations distributed across several sub-districts. The distribution of these destinations and their road connections can make it difficult for visitors to determine efficient travel routes. This study aimed to determine the shortest route from Tano Ponggol to Hadabuan Nasogo Waterfall using a weighted graph approach and Dijkstra's algorithm based on distance data obtained from Google Maps. Nine tourist attractions were modeled as vertices and thirteen connecting routes as weighted edges representing travel distances for four-wheeled vehicles. Distance data were collected from Google Maps in April 2025 and processed using Dijkstra's algorithm. The results showed that the shortest route was A→B→D→E→H→I, with a total distance of 100.1 km and six of the nine attractions included in the route. Compared with the sequential baseline route that passes through all nine destinations (A→B→C→D→E→F→G→H→I), the shortest route reduced the travel distance by 31.6 km (24%) and the estimated travel time by 1 hour and 3 minutes (27%). However, the shorter route bypassed three attractions (C, F, and G), showing that a shortest-path approach does not ensure coverage of all tourism destinations. Dijkstra's algorithm is therefore suitable for determining the shortest route between a selected source and destination, while tourists who intend to visit multiple destinations may require a multi-stop optimization approach, such as the Traveling Salesman Problem. The findings provide a route recommendation for tourism travel in Samosir Regency and illustrate the need to select an optimization method according to the intended travel objective.

Keywords: Dijkstra's Algorithm; Graph Theory; Shortest Path; Tourism Route Planning; Samosir Regency; Google Maps.

1. Introduction

The development of information technology has significantly transformed transportation and digital navigation systems. One of the most widely used technologies is Google Maps, which assists users in identifying locations, estimating travel distances, and selecting travel routes. In addition to its use for daily navigation, Google Maps can provide spatial and route information that can be used as input for route optimization studies (Agustiono & Hermanto, 2010). Samosir Regency is one of the tourism areas in the Lake Toba region of North Sumatra and offers various natural and cultural attractions, including Bukit Holbung, Pantai Pasir Putih Parbaba, Air Terjun Efrata, and Desa Wisata Tomok. The distribution of tourist attractions across different locations can make it difficult for visitors to determine an efficient route between selected destinations. This situation may result in longer travel distances, travel times, and transportation costs. Therefore, a route planning approach based on the connectivity and distance between tourist destinations can help identify more efficient travel routes (Cantona *et al.*, 2020).

Several studies have applied shortest-path algorithms to route optimization problems in different domains. Kurniawan and Atmojo (2025) implemented Dijkstra's algorithm to determine tourist routes in Jakarta using a weighted graph model. Angul *et al.* (2025) and Taneo *et al.* (2025) investigated the use of Dijkstra's algorithm and greedy approaches for shortest-route optimization. Their findings demonstrate the applicability of graph-based algorithms for determining efficient routes according to the characteristics of the routing problem. Dijkstra's algorithm has also been applied to healthcare navigation systems (Andriati *et al.*, 2025), tourism village route planning (Taneo *et al.*, 2025), and museum navigation applications (Andini *et al.*, 2025). These studies demonstrate the applicability of Dijkstra's algorithm to shortest-path problems in different routing contexts. Although Dijkstra's algorithm has been applied in various routing studies, research specifically representing interconnected tourist destinations in Samosir Regency as a weighted graph remains limited. In particular, there is limited research that uses route distance data obtained from Google Maps to represent the connections among selected tourist destinations in the region. This provides an opportunity to examine how a weighted graph can be used to model a tourism network and determine the shortest path between selected destinations.

The novelty of this study lies in the construction of a weighted graph representing selected tourist destinations in Samosir Regency, with the connecting routes assigned weights based on distance data obtained from Google Maps. The study focuses on a regional tourism network rather than a single tourism site or an urban transportation network. Dijkstra's algorithm is then applied to this graph to determine the shortest path between a selected starting point and destination. This approach provides a simple graph-based model for examining route distance within a network of tourism destinations in Samosir Regency. This study contributes to tourism route planning by providing a graph-based representation of selected tourism destinations in Samosir Regency and applying Dijkstra's algorithm to determine a shortest path within the network. The resulting route can serve as a reference for travel planning between the selected starting point and destination. The findings may also provide a basis for future development of digital tourism navigation applications that consider multiple tourism destinations and different travel objectives.

Therefore, this study aims to implement Dijkstra's algorithm to determine the shortest route between selected tourist destinations in Samosir Regency using distance data obtained from Google Maps. The objectives of this research are: (1) to model selected tourist destinations in Samosir Regency using graph theory; (2) to implement Dijkstra's algorithm to determine the shortest path between a selected source and destination; and (3) to evaluate the resulting route using distance data obtained from Google Maps. The remainder of this paper is organized as follows. Section 2 reviews related studies and theoretical foundations. Section 3 describes the research methodology. Section 4 presents the results and discussion. Finally, Section 5 concludes the study and provides recommendations for future research.

2. Related Work

2.1 Graph Theory

According to Safetra *et al.* (2026), graph theory provides a mathematical framework for representing relationships among objects through vertices and edges. In route-planning problems, tourist destinations can be represented as vertices, while road connections can be represented as weighted edges. The edge weights may represent measurable quantities such as distance or travel time. This representation enables graph-based algorithms to be applied to route optimization problems. The theory can be used to model connections among locations, making it suitable for route-related problems. A graph consists of a set of vertices representing objects or locations and a set of edges representing connections between them. In tourism route planning, tourist destinations can be represented as vertices, while the roads connecting the destinations can be represented as weighted edges. The edge weights may represent travel distance, travel time, or another measurable cost. A shortest-path algorithm can then be applied to the weighted graph to determine a route according to the selected optimization criterion. Graph theory has been applied in various information technology problems involving relationships, networks, and optimization (Safetra *et al.*, 2026). Its ability to represent interconnected objects makes it useful for route planning and other network-based applications. Graph-based approaches can also be applied to problems such as network coverage and monitoring. For example, Maulana (2025) applied the Kruskal minimum spanning tree approach to tourism route optimization in Bojong, Tegal Regency. This study illustrates another application of graph-based methods to tourism route optimization. Although the problem differs from shortest-path determination, the study illustrates the broader application of graph theory to network-based decision problems. Graph theory also provides the mathematical basis for shortest-path algorithms, including Dijkstra's algorithm. Agusnur (2025) discussed the use of graph theory in urban transportation network optimization. This illustrates the broader application of graph-based modeling to transportation and network optimization problems. These studies demonstrate the usefulness of graph-based models for representing relationships among locations and supporting optimization problems.

2.2 Tourism Route Optimization

Tourism route optimization involves determining travel routes according to specific objectives, such as minimizing travel distance, travel time, or transportation costs. According to Angul *et al.* (2025), route optimization can help tourists plan their trips more efficiently by reducing unnecessary travel. An appropriate route can also support accessibility to tourism destinations and improve the efficiency of travel planning. Graph theory can be used to represent tourism networks, where tourist destinations are modeled as vertices and connections between destinations are represented as edges with weights such as distance or travel time (Lakutu *et al.*, 2023). Different graph algorithms can then be applied according to the characteristics and objectives of the routing problem. Dijkstra's algorithm is commonly used to determine the shortest path in a weighted graph, while other algorithms have different functions and assumptions.

Manalu *et al.* (2025) applied Dijkstra's algorithm to determine shortest paths between tourism objects in the Lake Toba area. Their study demonstrates the relevance of Dijkstra's algorithm for tourism route planning in the same broader geographical region as the present study. Their study reported the route A→B→D→F→H→J→L with a total distance of 147.9 km. This finding demonstrates the applicability of Dijkstra's algorithm to tourism route planning in the Lake Toba region. The present study focuses on selected interconnected tourist destinations in Samosir Regency and uses route distance data obtained from Google Maps to construct the weighted graph. Digital technology and digital marketing are also relevant to tourism promotion and destination development. Barus *et al.* (2026) discussed the use of digital marketing to increase tourism potential in Samosir Regency. Digital information systems can therefore support the dissemination of destination information and assist tourists in planning their travel. Digital information systems can provide tourists with information about destinations and support travel planning. Digital mapping platforms such as Google Maps can provide route distance and estimated travel-time information for route planning studies. Sabilla and Taufiq (2022) applied the A* algorithm in a WebGIS application for shortest-route searching. This study demonstrates the use of graph-based search algorithms and digital mapping systems to support route selection. Their study illustrates the use of mapping data and graph-based algorithms to support route selection. However, the specific data source and findings reported by Susanto *et al.* (2023) should be verified against the original publication before the citation is finalized.

Pramadjaya and Rohmawati (2025) investigated the implementation of Dijkstra's algorithm for determining the shortest route to Universitas Pamulang. Their study further demonstrates the application of Dijkstra's algorithm to shortest-route problems. Their study compared the characteristics of different graph-search approaches. Dijkstra's algorithm is suitable when the objective is to minimize the total weight of a path, while BFS is generally appropriate for finding paths with the minimum number of edges in an unweighted graph. DFS, in contrast, is primarily used for graph traversal and does not generally guarantee a shortest path. Previous studies demonstrate the application of graph-based algorithms to route-planning problems in tourism and other domains. The selection of an algorithm should correspond to the objective and structure of the routing problem. In the present study, travel distance is used as the edge weight. Therefore, Dijkstra's algorithm is applied to determine the shortest-distance path between the selected source and destination.

Table 1. Comparative Summary of Previous Route Optimization Studies

No.	Author(s)	Year	Study Context	Algorithm	Main Finding
1	Kurniawan & Atmojo	2025	Tourism route planning in Jakarta	Dijkstra	Applied Dijkstra to determine tourism routes using a weighted graph
2	Angul *et al.*	2025	Shortest-route optimization	Dijkstra & Greedy	Investigated Dijkstra and greedy approaches for shortest-route problems
3	Manalu *et al.*	2025	Lake Toba tourism	Dijkstra	Applied Dijkstra to determine shortest routes between tourism objects
4	Taneo *et al.*	2025	Tourism village route planning	Dijkstra & Greedy	Compared Dijkstra and greedy approaches for shortest-route optimization
5	Andini *et al.*	2025	Shortest-route search	Dijkstra	Applied Dijkstra for shortest-route optimization
6	Sabilla & Taufiq	2022	WebGIS route search	A*	Applied A* for shortest-route searching
7	Current Study	2026	Samosir tourism network	Dijkstra	Identified A→B→D→E→H→I with a total distance of 100.1 km

2.3 Dijkstra’s Algorithm

Dijkstra’s algorithm is a shortest-path algorithm used to determine the minimum-weight path from a selected source vertex to other vertices in a weighted graph with non-negative edge weights. The algorithm begins by assigning a tentative distance to each vertex, with zero assigned to the source vertex and an initially infinite distance assigned to the remaining vertices. At each iteration, the unvisited vertex with the smallest tentative distance is selected. The algorithm then evaluates the edges connected to the selected vertex by comparing the current tentative distance with the distance obtained through the selected vertex. If the new distance is smaller, the tentative distance and predecessor are updated. This process continues until the required destination is reached or all relevant vertices have been processed. Andriati *et al*. (2025) applied Dijkstra’s algorithm to determine a shortest route using a graph-based representation of locations and connecting roads. Their findings demonstrate the application of the algorithm to navigation problems. Dijkstra’s algorithm has also been applied in other routing contexts, including logistics and tourism. These applications demonstrate the use of Dijkstra’s algorithm for determining shortest paths in weighted networks. The specific findings and characteristics of each cited study should be verified against the original publications before the references are finalized. Dijkstra’s algorithm is particularly suitable for the present study because the tourism network is represented as a weighted graph, with travel distance assigned as the edge weight. Therefore, the algorithm is used to minimize the total travel distance between the selected source and destination. However, Dijkstra’s algorithm does not inherently optimize the number of tourist destinations visited. A routing problem that requires a traveler to visit multiple specified destinations would involve a different optimization objective and may require a multi-stop routing approach.

2.4 Google Maps

Google Maps is a digital mapping platform that provides location, route, distance, and estimated travel-time information. Such information can be used as input for route-planning and graph-based optimization studies. In the present research, Google Maps is used as the source of route distance data for constructing weighted connections between selected tourist destinations. Previous studies have also combined mapping data with graph-based algorithms for route planning. The specific findings concerning the comparison between Google Maps and Dijkstra’s algorithm should be verified against the original publication before the citation is finalized. For the present study, the roles of Google Maps and Dijkstra’s algorithm are distinct. Google Maps provides the route distance information used to define the edge weights, while Dijkstra’s algorithm processes the resulting weighted graph to determine the shortest path. Therefore, Google Maps serves as the data source for the route network, whereas Dijkstra’s algorithm is used to determine the shortest-distance path between the selected source and destination.

3. Methodology

This study employs a quantitative computational approach using weighted graph modeling and Dijkstra’s algorithm to determine the shortest route between selected tourist destinations in Samosir Regency. Tourist destination data were obtained from the official website of the Samosir Regency Government, while route distance and estimated travel-time data were collected from Google Maps. The tourism network is modeled as a weighted graph $G=(V,E,w)$, where V represents the set of vertices corresponding to tourist destinations, E represents the set of edges connecting destinations, and $w : E \rightarrow R +$ represents the weight function assigned to each edge. In this study, the edge weight represents travel distance in kilometers. The resulting graph is then processed using Dijkstra’s algorithm to determine the shortest-distance route from the selected source to the selected destination.

Figure 1. System Flowchart

The research process begins with the identification of tourist destinations in Samosir Regency. The selected destinations and route information are then collected from the relevant sources. Distance data obtained from Google Maps are converted into weighted edges and organized into an adjacency matrix. After the source and destination nodes are determined, Dijkstra's algorithm is applied to calculate the shortest path. The resulting route is then evaluated based on total travel distance and estimated travel time and compared with the predefined sequential route.

3.1 Data Collection

Data collection involved identifying tourist destinations and obtaining route information for the selected locations. Information on tourist attractions was collected from the official website of the Samosir Regency Government, while Google Maps was used to obtain route distances and estimated travel times between connected destinations. Nine tourist attractions were selected from three sub-districts in Samosir Regency: Pangururan, Simanindo, and Sianjur Mulamula. The selected destinations are represented as vertices in the graph, while the road connections between destinations are represented as weighted edges. The selected tourist attractions are presented in Table 2.

Table 2. Tourist Attraction Data

No.	Sub-district	Tourist Attraction
1	Pangururan	Tano Ponggol
2	Pangururan	Pantai Pasir Putih
3	Pangururan	Pemandian Air Panas Aek Rangat
4	Simanindo	Museum Tomok
5	Simanindo	Batu Kursi Siallagan
6	Simanindo	Museum Huta Bolon
7	Sianjur Mulamula	Aek Sipitu Dai
8	Sianjur Mulamula	Batu Hobon
9	Sianjur Mulamula	Air Terjun Hadabuan Nasogo

Based on Table 2, the nine tourist attractions are represented as nodes A through I to simplify the graph construction and route computation. Distance observations were conducted using Google Maps in April 2025. For each selected connection, the shortest available route for four-wheeled vehicles was recorded.

3.2 Distance Data

Distance and estimated travel-time data between tourist attractions were obtained from Google Maps in April 2025. The shortest available route for four-wheeled vehicles was selected for each connection. The distance in kilometers was used as the edge weight in the graph, while travel time was retained for the evaluation of the resulting route. The graph contains 13 edges consisting of eight main connections and five alternative connections. The main connections represent sequential links between neighboring destinations, while the alternative connections represent additional road connections between destinations that can be traversed directly. These connections were identified from the available routes in Google Maps and were included to represent alternative paths within the tourism network.

Table 3. Distance Data Between Tourist Attractions

No.	Route	Distance (km)	Travel Time	Type
1	Tano Ponggol (A) – Pantai Pasir Putih (B)	10.2	19 min	Main
2	Pantai Pasir Putih (B) – Pemandian Air Panas Aek Rangat (C)	12.2	22 min	Main
3	Pemandian Air Panas Aek Rangat (C) – Museum Tomok (D)	43.8	1 h 12 min	Main
4	Museum Tomok (D) – Batu Kursi Siallagan (E)	5.2	12 min	Main
5	Batu Kursi Siallagan (E) – Museum Huta Bolon (F)	16.4	25 min	Main
6	Museum Huta Bolon (F) – Aek Sipitu Dai (G)	35.5	56 min	Main
7	Aek Sipitu Dai (G) – Batu Hobon (H)	5.9	18 min	Main
8	Batu Hobon (H) – Air Terjun Hadabuan Nasogo (I)	2.5	9 min	Main
9	Tano Ponggol (A) – Pemandian Air Panas Aek Rangat (C)	2.0	2 min	Alternative
10	Pantai Pasir Putih (B) – Museum Tomok (D)	31.8	47 min	Alternative
11	Museum Tomok (D) – Aek Sipitu Dai (G)	56.2	1 h 29 min	Alternative
12	Batu Kursi Siallagan (E) – Batu Hobon (H)	50.4	1 h 23 min	Alternative
13	Museum Huta Bolon (F) – Batu Hobon (H)	34.1	59 min	Alternative

Based on Table 3, the network contains 13 route connections with different distances and estimated travel times. The shortest recorded connection is A–C, with a distance of 2.0 km and an estimated travel time of 2 minutes. The longest recorded connection is D–G, with a distance of 56.2 km and an estimated travel time of 1 hour 29 minutes. Distance and travel time do not necessarily have a constant relationship because travel time may vary according to the characteristics of the available routes and road conditions. For example, the E–H connection has a distance of 50.4 km and an estimated travel time of 1 hour 23 minutes, whereas the C–D connection has a distance of 43.8 km and an estimated travel time of 1 hour 12 minutes. In this study, distance is used as the edge weight in Dijkstra's algorithm, while travel time is retained as an additional variable for evaluating the resulting route. This distinction is necessary because the algorithm is designed to minimize the weight assigned to the graph; therefore, the calculated shortest path represents the minimum-distance route rather than a route directly optimized for travel time. To support verification of the collected data, the Google Maps sources for the 13 route connections are listed in Table 4.

Table 4. Google Maps Sources for Route Data

No	Subdistrict	Tourist Attraction	Distance (km)	Travel Time
1	Tano Ponggol (A)	Pantai Pasir Putih (B)	10,2	https://maps.app.goo.gl/agBhS59uRuUvjudF7
2	Pantai Pasir Putih (B)	Pemandian Air Panas Aek Rangat (C)	12,2	https://maps.app.goo.gl/EidBbzNo4qzcZcZV8
3	Pemandian Air Panas Aek Rangat (C)	Museum Tomok (D)	43,8	https://maps.app.goo.gl/mGNV8sfdt39tmtpY9
4	Museum Tomok (D)	Batu Kursi Siallagan (E)	5,2	https://maps.app.goo.gl/fsc5rbuzouGFC4cNA
5	Batu Kursi Siallagan (E)	Museum Huta Bolon (F)	16,4	https://maps.app.goo.gl/yV798EFov3AaVD1w7
6	Museum Huta Bolon (F)	Aek Sipitu Dai (G)	35,5	https://maps.app.goo.gl/jodNiu8MEGDcYSeK8
7	Aek Sipitu Dai (G)	Batu Hobon (H)	5,9	https://maps.app.goo.gl/G27zwUSpoAnhTqzk9
8	Batu Hobon (H)	Air Terjun Hadabuan Nasogo (I)	2,5	https://maps.app.goo.gl/ttbPNFJtszBbVRcp7
9	Tano Ponggol (A)	Pemandian Air Panas Aek Rangat (C)	2,0	https://maps.app.goo.gl/MUUZZrrN217bUcqs7

10	Pantai Pasir Putih (B)	Museum Tomok (D)	31,8	https://maps.app.goo.gl/q74Wnw1epSuriGhx5
11	Museum Tomok (D)	Aek Sipitu Dai (G)	56,2	https://maps.app.goo.gl/r67doRyQY7uYnfjs5
12	Batu Kursi Siallagan (E)	Batu Hobon (H)	50,4	https://maps.app.goo.gl/BW3Kmv37F9YncpRN9
13	Museum Huta Bolon (F)	Batu Hobon (H)	34,1	https://maps.app.goo.gl/j6GSvEkTiqfW5jDE8

All route data were collected in April 2025 using the shortest available route for four-wheeled vehicles. The original Google Maps links should be retained in the final manuscript or supplementary material so that the reported distances can be independently checked.

3.3 Graph Construction

The tourism network was modeled as a weighted graph in which each tourist attraction represents a vertex and each direct road connection represents an edge. The weight of each edge corresponds to the travel distance between the connected destinations. The graph is formally defined as: G=(V,E,w) where V denotes the set of vertices, E denotes the set of edges connecting the vertices, and w denotes the distance assigned to each edge. The nine tourist attractions were represented by nodes A through I, as shown in Table 5.

Table 5. Representation of Tourist Attraction Vertices

No.	Node	Tourist Attraction
1	A	Tano Ponggol
2	B	Pantai Pasir Putih
3	C	Pemandian Air Panas Aek Rangat
4	D	Museum Tomok
5	E	Batu Kursi Siallagan
6	F	Museum Huta Bolon
7	G	Aek Sipitu Dai
8	H	Batu Hobon
9	I	Air Terjun Hadabuan Nasogo

The vertex set is defined as: V = {A, B, C, D, E, F, G, H, I} and the edge set is defined as: E = {(A,B), (B,C), (C,D), (D,E), (E,F), (F,G), (G,H), (H,I), (A,C), (B,D), (D,G), (E,H), (F,H)}. The graph contains nine vertices and thirteen edges, with all edge weights having positive values. Therefore, the graph satisfies the non-negative edge-weight condition required for applying Dijkstra's algorithm. The tourism network is modeled as an undirected graph, meaning that each connection is assumed to have the same distance in both directions. This assumption should be confirmed against the Google Maps observations because road restrictions, one-way roads, or differences in available routes may result in different distances depending on the direction of travel. If directional differences exist, the graph should instead be modeled as a directed graph. The distance relationships among the vertices are represented by the adjacency matrix in Table 6.

Table 6. Adjacency Matrix of the Tourism Graph (km)

	A	B	C	D	E	F	G	H	I
A	0	10.2	2.0	–	–	–	–	–	–
B	10.2	0	12.2	31.8	–	–	–	–	–
C	2.0	12.2	0	43.8	–	–	–	–	–
D	–	31.8	43.8	0	5.2	–	56.2	–	–
E	–	–	–	5.2	0	16.4	–	50.4	–
F	–	–	–	–	16.4	0	35.5	34.1	–
G	–	–	–	56.2	–	35.5	0	5.9	–
H	–	–	–	–	50.4	34.1	5.9	0	2.5
I	–	–	–	–	–	–	–	2.5	0

A dash (–) indicates that no direct edge exists between the corresponding vertices. The diagonal elements are zero because they represent the distance from a vertex to itself. The adjacency matrix provides a numerical representation of the weighted graph and is used as an input representation for the shortest-path calculation.

3.4 Source and Destination Selection

To evaluate the shortest-path calculation, Tano Ponggol (A) was selected as the source node and Air Terjun Hadabuan Nasogo (I) as the destination node. These destinations were selected because they are located in different sub-districts of Samosir Regency and are connected through several possible routes in the constructed tourism network. This configuration allows the algorithm to evaluate alternative paths between the source and destination.

Table 7. Source and Destination Selection

Node	Tourist Attraction	Role
A	Tano Ponggol	Source
B	Pantai Pasir Putih	Intermediate node
C	Pemandian Air Panas Aek Rangat	Intermediate node
D	Museum Tomok	Intermediate node
E	Batu Kursi Siallagan	Intermediate node
F	Museum Huta Bolon	Intermediate node
G	Aek Sipitu Dai	Intermediate node
H	Batu Hobon	Intermediate node
I	Air Terjun Hadabuan Nasogo	Destination

Tano Ponggol (A) serves as the starting point, while Air Terjun Hadabuan Nasogo (I) serves as the final destination. The remaining nodes are potential intermediate nodes and are included in the shortest-path search according to the connectivity and edge weights of the graph.

3.5 Dijkstra's Algorithm

Dijkstra's algorithm was applied to determine the minimum-distance path from the source node to the destination node. The algorithm is appropriate for the constructed graph because all edge weights represent positive travel distances. The relaxation operation is defined as: $d(v)=\min(d(v),d(u)+w(u,v))$ where $d(v)$ represents the current tentative distance to vertex v, $d(u)$ represents the current shortest known distance to vertex u, and $w(u,v)$ represents the distance weight of the edge connecting vertices u and v. Initially, the source node is assigned a distance of zero, while all other nodes are assigned infinity. The algorithm then selects the unvisited node with the smallest tentative distance and evaluates the distances of its adjacent nodes through the relaxation process. If a shorter distance is obtained through the selected node, the corresponding distance and predecessor values are updated. This process continues until the destination node is reached or all relevant vertices have been processed. The pseudocode used in this study is presented below.

Pseudocode: Dijkstra's Algorithm

```
DIJKSTRA(G, source):

for each vertex v in G:
    d[v] ← ∞
    prev[v] ← null

d[source] ← 0

Q ← all vertices in a priority queue

while Q is not empty:
    u ← vertex with the smallest d[u]
    remove u from Q

    for each neighbor v of u:
        alt ← d[u] + w(u, v)

        if alt < d[v]:
            d[v] ← alt
            prev[v] ← u

return d[], prev[]
```

The algorithm produces the shortest distance from the source node to each reachable vertex and the predecessor information required to reconstruct the corresponding route. In this study, the predecessor information is used to reconstruct the path from A to I. The resulting route is then evaluated based on total travel distance and estimated travel time. For comparison, the shortest route obtained from Dijkstra's algorithm is compared with a predefined sequential route that passes through all nine selected tourist destinations:

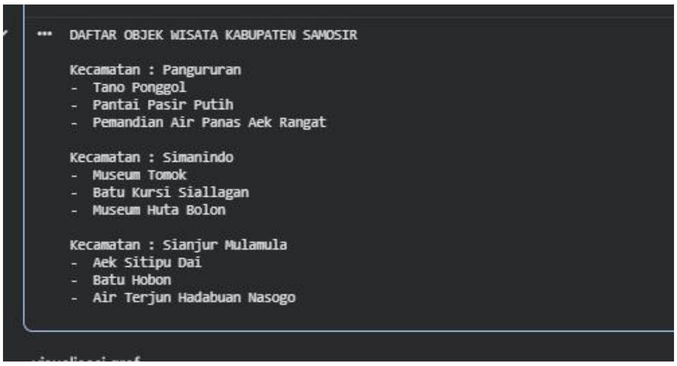
$$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow H \rightarrow I \quad (1)$$

This sequential route is treated as a baseline route, not as an independently optimized route. The purpose of the comparison is to quantify the difference in total distance and estimated travel time between (1) the shortest path from A to I and (2) a route that visits all nine selected destinations in the predefined sequence. This distinction is important because Dijkstra's algorithm solves a single-source shortest-path problem and does not seek to maximize the number of tourist destinations visited.

4. Result and Discussion

4.1 Results

The study identified nine tourist attractions in Samosir Regency as the destinations included in the tourism route network. The selected attractions are located in three sub-districts: Pangururan, Simanindo, and Sianjur Mulamula. The destinations include cultural, natural, and recreational attractions that are connected through road networks accessible to four-wheeled vehicles. The tourist attraction data were obtained from the official website of the Samosir Regency Government and Google Maps. Each destination was assigned a node label from A to I to simplify the graph representation and the implementation of Dijkstra's algorithm.



```
DAFTAR OBJEK WISATA KABUPATEN SAMOSIR

Kecamatan : Pangururan
-  Tano Ponggol
-  Pantai Pasir Putih
-  Pemandian Air Panas Aek Rangat

Kecamatan : Simanindo
-  Museum Tomok
-  Batu Kursi Siallagan
-  Museum Huta Bolon

Kecamatan : Sianjur Mulamula
-  Aek Sitipu Dai
-  Batu Hobon
-  Air Terjun Hadabuan Nasogo
```

Figure 2. Tourism Data Collection Using Google Colab

Figure 2 shows the data collection and processing stage used to prepare the tourism destination data for the graph model. The collected information included destination names, locations, route distances, and estimated travel times. The tourism network was modeled as a weighted graph consisting of nine nodes and thirteen edges. Each node represents a tourist attraction, while each edge represents a direct road connection between two destinations. The weight of each edge corresponds to the travel distance obtained from Google Maps.

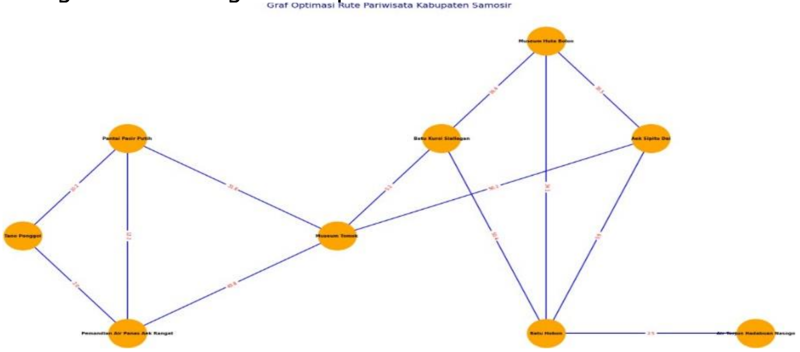


Figure 3. Tourism Graph Visualization

The graph contains both sequential and alternative connections between destinations. The alternative connections allow Dijkstra's algorithm to evaluate routes that bypass intermediate tourist attractions and may produce a shorter total distance. Figure 4 presents the weighted tourism graph using a circular layout. The nodes A–I represent the nine selected tourist attractions, while the edges represent direct road connections between the destinations. The distance and estimated travel time are displayed for each route segment based on the Google Maps observations.

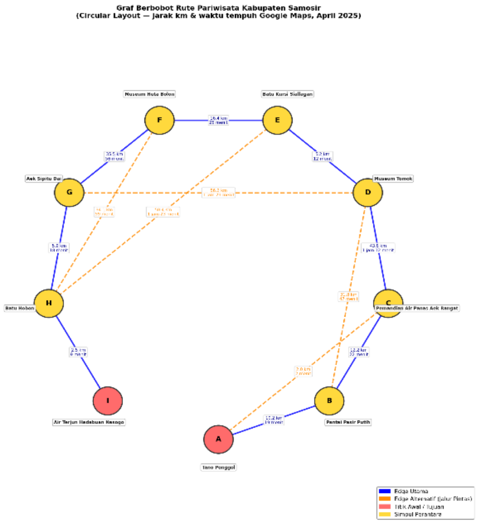


Figure 4. Weighted Graph of Tourism Routes in Samosir Regency

The graph contains eight sequential edges and five alternative edges. The alternative edges provide direct connections between several destinations that are not adjacent in the predefined sequential route. For example, the B–D connection has a distance of 31.8 km and an estimated travel time of 47 minutes. If tourists travel from B through C to D, the total distance becomes: B→C→D = 12.2 + 43.8 = 56.0 km with a combined travel time of: 22 + 72 = 94 minutes. Similarly, the E – H connection has a distance of 50.4 km and an estimated travel time of 1 hour 23 minutes. The sequential route E→F→G→H has a total distance of: 16.4 + 35.5 + 5.9 = 57.8 km and a combined travel time of: 25 + 56 + 18 = 99 minutes. These alternative connections provide multiple possible paths between the source and destination and allow Dijkstra's algorithm to evaluate the cumulative distance of different routes. Dijkstra's algorithm was applied to determine the shortest-distance path from Tano Ponggol (A) to Air Terjun Hadabuan Nasogo (I). The edge weights used in the calculation represent travel distances in kilometers.

```
============================================================
HASIL PENCARIAN RUTE TERPENDEK
============================================================
Titik Awal  : Tano Ponggol
Titik Tujuan: Air Terjun Hadabuan Nasogo

Rute Terpendek:
  A -> B: Tano Ponggol -> Pantai Pasir Putih = 10.2 km
  B -> D: Pantai Pasir Putih -> Museum Tomok = 31.8 km
  D -> E: Museum Tomok -> Batu Kursi Siallagan = 5.2 km
  E -> H: Batu Kursi Siallagan -> Batu Hobon = 50.4 km
  H -> I: Batu Hobon -> Air Terjun Hadabuan Nasogo = 2.5 km

Total Jarak : 100.1 km
Rute Kode   : A -> B -> D -> E -> H -> I
```

Figure 5. Dijkstra's Algorithm Results

The calculation began with node A as the source node, with an initial distance of 0 km. All other nodes were initially assigned infinite distance values. At each iteration, the unvisited node with the smallest tentative distance was selected and its neighboring nodes were evaluated.

Table 7. Dijkstra's Algorithm Iteration

Iteration	Selected Node	A	B	C	D	E	F	G	H	I
0	–	0	∞	∞	∞	∞	∞	∞	∞	∞
1	A	0	10.2	2.0	∞	∞	∞	∞	∞	∞
2	C	0	10.2	2.0	45.8	∞	∞	∞	∞	∞
3	B	0	10.2	2.0	42.0	∞	∞	∞	∞	∞
4	D	0	10.2	2.0	42.0	47.2	∞	98.2	∞	∞
5	E	0	10.2	2.0	42.0	47.2	63.6	98.2	97.6	∞
6	F	0	10.2	2.0	42.0	47.2	63.6	98.2	97.6	∞
7	H	0	10.2	2.0	42.0	47.2	63.6	98.2	97.6	100.1

8	G	0	10.2	2.0	42.0	47.2	63.6	98.2	97.6	100.1
9	I	0	10.2	2.0	42.0	47.2	63.6	98.2	97.6	100.1

The table shows the step-by-step distance updates during the Dijkstra process. In the first iteration, node A is selected as the source, resulting in tentative distances of 10.2 km to B and 2.0 km to C. Node C is selected in the second iteration because its tentative distance of 2.0 km is smaller than the distance of 10.2 km to B. Processing C produces a tentative distance to D of: A→C→D = 2.0 + 43.8 = 45.8 km When node B is processed, the distance to D is updated through the alternative edge B–D: A→B→D = 10.2 + 31.8 = 42.0 km. Therefore, the tentative distance to D is reduced from 45.8 km to 42.0 km. The subsequent iterations continue the same relaxation process until the destination node I is reached. The final distance to I is 100.1 km.

Table 8. Dijkstra's Algorithm Predecessor and Final Distance

Node	Predecessor	Final Distance (km)
A	–	0.0
B	A	10.2
C	A	2.0
D	B	42.0
E	D	47.2
F	E	63.6
G	D	98.2
H	E	97.6
I	H	100.1

The predecessor information is used to reconstruct the shortest path from the destination node to the source node. Starting from node I, the predecessor sequence is: I ← H ← E ← D ← B ← A Reversing the sequence gives: A → B → D → E → H → I Therefore, Dijkstra's algorithm identifies A→B→D→E→H→I as the shortest-distance route from Tano Ponggol to Air Terjun Hadabuan Nasogo, with a total distance of 100.1 km. The result also shows that nodes C, F, and G are not included in the final shortest path. The shortest route identified by Dijkstra's algorithm consists of five travel segments: A→B→D→E→H→I The distance and travel-time details are presented in Table 9.

Table 9. Edge Details of the Shortest Route A→B→D→E→H→I

Edge	From	To	Distance (km)	Travel Time
A–B	Tano Ponggol	Pantai Pasir Putih	10.2	19 minutes
B–D	Pantai Pasir Putih	Museum Tomok	31.8	47 minutes
D–E	Museum Tomok	Batu Kursi Siallagan	5.2	12 minutes
E–H	Batu Kursi Siallagan	Batu Hobon	50.4	1 hour 23 minutes
H–I	Batu Hobon	Air Terjun Hadabuan Nasogo	2.5	9 minutes
		Total	100.1	2 hours 50 minutes

The total travel time obtained by summing the Google Maps estimates is 2 hours and 50 minutes. For comparison, the predefined sequential baseline route is: A→B→C→D→E→F→G→H→I This route passes through all nine selected tourist attractions and has a total distance of 131.7 km and an estimated travel time of 3 hours and 53 minutes.

Table 10. Comparison of Dijkstra's Route and Sequential Baseline Route

Comparison Aspect	Dijkstra's Route	Sequential Baseline Route
Route	A→B→D→E→H→I	A→B→C→D→E→F→G→H→I
Number of nodes	6	9
Total distance	100.1 km	131.7 km
Total travel time	2 hours 50 minutes	3 hours 53 minutes
Distance difference	31.6 km shorter	–
Time difference	1 hour 3 minutes faster	–

The Dijkstra route reduces the total distance by 31.6 km, equivalent to approximately 24.0% of the sequential baseline distance. Based on the recorded travel-time estimates, it also reduces the estimated travel time by 63 minutes, or approximately 27.0% of the sequential baseline travel time.

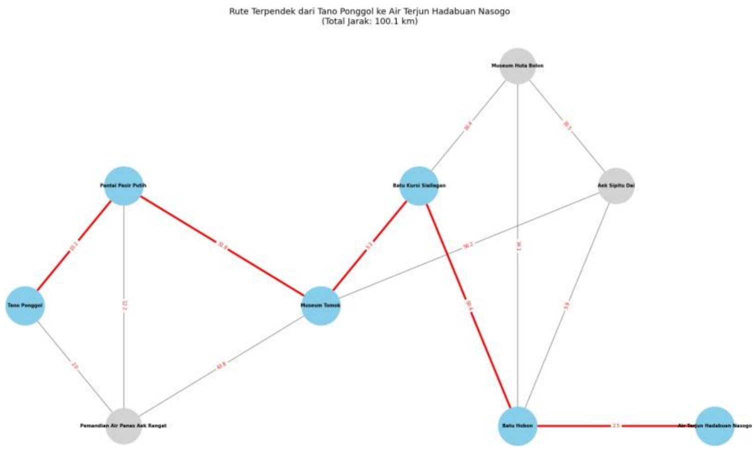


Figure 6. Visualization of Dijkstra's Shortest Route: A→B→D→E→H→I

Figure 6 illustrates the shortest route obtained from the Dijkstra calculation. The route passes through A, B, D, E, H, and I, while C, F, and G are bypassed.

4.2 Discussion

The results demonstrate that Dijkstra's algorithm successfully identified the shortest-distance route from Tano Ponggol (A) to Air Terjun Hadabuan Nasogo (I) as A→B→D→E→H→I, with a total distance of 100.1 km. The result illustrates how Dijkstra's algorithm determines a shortest path based on cumulative distance rather than simply selecting the geographically closest destination at each stage. For example, the initial route from A through C to D produces a cumulative distance of 45.8 km, calculated as A–C (2.0 km) plus C–D (43.8 km). However, when node B is evaluated, the alternative route A–B–D produces a shorter cumulative distance of 42.0 km, calculated as A–B (10.2 km) plus B–D (31.8 km). Consequently, the tentative distance to node D is updated from 45.8 km to 42.0 km. This process demonstrates the relaxation mechanism that enables Dijkstra's algorithm to identify the minimum-distance path within the weighted tourism network. The analysis also shows that the presence of alternative edges plays an important role in determining the final route. In particular, the B–D and E–H connections are included in the shortest path and allow the algorithm to bypass several intermediate destinations. The B–D connection has a distance of 31.8 km, whereas traveling from B to D through C requires 56.0 km, consisting of B–C (12.2 km) and C–D (43.8 km). The corresponding estimated travel time is also lower for the direct B–D connection, at 47 minutes compared with 94 minutes through B–C–D. Similarly, the E–H connection has a distance of 50.4 km, while the sequential route E–F–G–H covers 57.8 km. These findings indicate that alternative road connections can substantially influence the resulting shortest path because they provide opportunities to bypass intermediate nodes when the cumulative distance is lower.

In terms of overall efficiency, the route identified by Dijkstra's algorithm covers 100.1 km, compared with 131.7 km for the predefined sequential baseline route A→B→C→D→E→F→G→H→I. The difference of 31.6 km represents approximately 24.0% of the baseline distance. The estimated travel time also decreases from 3 hours 53 minutes for the sequential baseline route to 2 hours 50 minutes for the Dijkstra route, representing a reduction of 1 hour 3 minutes, or approximately 27.0%. These results indicate that the selected shortest path can provide a more efficient travel option when the primary objective is to minimize travel distance between the specified source and destination.

Nevertheless, the reduction in travel time should be interpreted as a secondary outcome rather than as the direct optimization objective of the algorithm. In this study, Dijkstra's algorithm uses travel distance as the edge weight, not travel time. The reported 27.0% reduction in travel time is therefore derived from the Google Maps travel-time estimates associated with the route segments selected by the algorithm. Consequently, the study does not demonstrate that Dijkstra's algorithm directly optimized travel time. Rather, it demonstrates that the minimum-distance route identified from the constructed graph also has a lower aggregated travel-time estimate than the predefined sequential baseline route.

An important implication of the findings is the trade-off between distance minimization and destination coverage. The shortest route contains six destinations, namely A, B, D, E, H, and I, while three destinations—C, F, and G—are bypassed. This outcome is consistent with the objective of Dijkstra's algorithm, which is to determine the minimum-weight path between a specified source and destination. The algorithm does not attempt to maximize the number of tourist attractions visited. Therefore, the route A→B→D→E→H→I should be interpreted as the shortest-distance connection between Tano Ponggol and Air Terjun Hadabuan Nasogo within the constructed tourism network, rather than as an optimal itinerary for tourists who intend to visit all nine attractions. From a practical perspective, the findings demonstrate the potential of weighted graph modeling for supporting tourism route planning in Samosir Regency. The tourist attractions can be represented as vertices, the road connections as edges, and the Google Maps distance data as edge weights. Dijkstra's algorithm can then process this network to identify a minimum-distance route between a selected origin and destination. For tourists whose primary priority is reducing travel distance between Tano Ponggol and Air Terjun Hadabuan Nasogo, the route A→B→D→E→H→I provides a shorter alternative to the predefined sequential route. The approach can also serve as a basic computational framework for future digital tourism navigation systems.

However, the findings should be considered within the limitations of the study. First, the distance and travel-time data were obtained from Google Maps in April 2025. Such information may change over time because of traffic conditions, road maintenance, construction, weather, or changes in road accessibility. Therefore, the reported route distances and travel-time estimates represent the conditions recorded during the data-collection period rather than permanent values. Second, the tourism network was modeled as an undirected graph, meaning that each connection was assumed to have the same distance in both directions. This assumption is appropriate only if the corresponding road connections are effectively symmetrical for the selected travel mode. If one-way restrictions, different road conditions, or directional route differences exist, a directed graph would provide a more appropriate representation.

Another limitation concerns the choice of edge weight. The present study uses travel distance as the primary optimization criterion and therefore does not directly incorporate other factors that may affect tourists' route decisions, such as traffic conditions, road quality, fuel consumption, attraction opening hours, accessibility, or individual tourist preferences. In addition, the selected shortest path does not consider the requirement to visit multiple or all tourist attractions. As a result, three of the nine destinations are excluded from the resulting route. Future research could address this limitation by developing a multi-objective or multi-stop optimization model. For example, approaches related to the Traveling Salesman Problem (TSP) could be considered when the objective is to determine an efficient route that visits multiple specified tourist destinations, while other optimization approaches could incorporate travel time, destination preferences, or other tourism-related constraints. The findings confirm that Dijkstra's algorithm is appropriate for determining a shortest-distance route in the weighted tourism graph constructed in this study. The algorithm identified a route of 100.1 km, which is 31.6 km shorter than the predefined sequential baseline. At the same time, the results reveal that shortest-path optimization and tourism itinerary planning are not necessarily equivalent objectives. A route that minimizes distance may bypass several attractions that tourists may wish to visit. Therefore, the effectiveness of a routing algorithm should be evaluated according to the specific objective of the trip. For distance-oriented travel between a defined source and destination, Dijkstra's algorithm provides an appropriate solution; for tourism planning that emphasizes destination coverage, additional multi-stop or multi-objective optimization methods are required.

5. Conclusion and Recommendations

This study applied weighted graph theory and Dijkstra's algorithm to determine the shortest-distance route between selected tourist destinations in Samosir Regency using Google Maps distance data. The tourism network consisted of nine tourist attractions and thirteen connecting edges. The shortest route from Tano Ponggol (A) to Air Terjun Hadabuan Nasogo (I) was A→B→D→E→H→I, with a total distance of 100.1 km. Compared with the predefined sequential route A→B→C→D→E→F→G→H→I, which covers 131.7 km, the Dijkstra route reduces the distance by 31.6 km (24.0%). The results demonstrate that alternative connections, particularly B–D and E–H, enable the shortest-path calculation to bypass three destinations: Pemandian Air Panas Aek Rangat (C), Museum Huta Bolon (F), and Aek Sipitu Dai (G). This finding highlights the distinction between shortest-path optimization and tourism itinerary planning. Dijkstra's algorithm minimizes the edge-weighted distance between a specified source and destination but does not maximize destination coverage. Therefore, the resulting route should be interpreted as an efficient shortest-distance route rather than an optimal itinerary for tourists intending to visit all selected attractions.

The study also demonstrates the applicability of weighted graph modeling for tourism route planning in Samosir Regency. However, the findings are subject to several limitations. The route data were collected from Google Maps in April 2025 and may change due to traffic, road conditions, or accessibility. The analysis also focuses on a single source–destination pair and a network of only nine attractions and thirteen edges. In addition, distance was used as the edge weight, without directly incorporating real-time traffic, road conditions, attraction opening hours, or tourist preferences.

Future research should expand the tourism network, incorporate dynamic traffic and road-condition data, and compare Dijkstra with other shortest-path algorithms such as A*. For multi-destination tourism planning, future studies could apply TSP or other multi-stop optimization methods and consider multiple objectives, including distance, travel time, destination preferences, and tourism constraints.

References

Agusnur, A. (2025). Penggunaan teori graf dalam optimasi jaringan transportasi kota. *Jurnal Matematika dan Aplikasi, 1*(1), 29–35.

Agustiono, W., & Hermanto, K. D. (2010). Pemanfaatan teknologi Google Maps untuk penyelesaian online travelling salesman problem (TSP) berbasis geografis dengan menggunakan algoritma heuristik. *JUTI: Jurnal Ilmiah Teknologi Informasi, 8*(2), 38–42. https://doi.org/10.12962/j24068535.v8i2.a315

Andini, M., Kultsum, R. U., Raihan, M. H. R., & Lestari, S. (2025). Optimasi pencarian rute terpendek menggunakan algoritma Dijkstra. *Journal of Information System, Applied, Management, Accounting and Research, 9*(1), 290–302. https://doi.org/10.52362/jisamar.v9i1.1746

Andriati, D. A., Dariato, E., & Hafizh, R. (2025). Implementasi teori graf dan optimisasi algoritma Dijkstra, BFS, dan DFS dalam menentukan rute terpendek jaringan bengkel di Jakarta berbasis Google Maps. *Jurnal Multimedia dan Teknologi Informasi (Jatilima), 7*(3), 674–684. https://doi.org/10.54209/jatilima.v7i03.1654

Angul, A., Fallo, D., Tanggo, K. V., Belo, I. N. A., & Hoar, F. (2025). Implementasi algoritma Dijkstra dan greedy dalam penyelesaian masalah rute terpendek. *Jurnal Kridatama Sains dan Teknologi, 7*(1), 489–496. https://doi.org/10.53863/kst.v7i01.1654

Barus, Y. L. B., Harianja, R. P., Sinaga, F., Nababan, A., & Sihombing, N. E. (2026). Pemanfaatan digital marketing dalam meningkatkan potensi pariwisata pada Dinas Kebudayaan dan Pariwisata Kabupaten Samosir. *Jurnal Pengabdian Kepada Masyarakat "PARHOBAS", 1*(1), 126–133.

Cantona, A., Fauziah, & Winarsih. (2020). Implementasi algoritma Dijkstra pada pencarian rute terpendek ke museum di Jakarta. *Jurnal Teknologi dan Manajemen Informatika, 6*(1), 27–34. https://doi.org/10.26905/jtmi.v6i1.3837

Fitriya B, W. A., Sumardi, S. R., Paranoan, N. R., & Allo, C. B. G. (2023). Penentuan rute di aplikasi Google Maps dengan menggunakan graf dan algoritma Prim. *KOLONI: Jurnal Multidisiplin Ilmu, 2*(1), 220–225. https://doi.org/10.31004/koloni.v2i1.434

Hambali, F. U., & Wismarini, T. D. (2022). Metode network analisis pencarian rute optimal dengan algoritma Dijkstra kunjungan Desa Wisata Wonolopo dengan jarak tempuh berdasar metode Euclidean distance. *Jurnal Mahajana Informasi, 7*(1), 1–8. https://doi.org/10.51544/jurnalmi.v7i1.2949

Kurniawan, E. K., & Atmojo, W. T. (2025). Penerapan algoritma Dijkstra dalam penentuan rute terpendek kunjungan wisata Jakarta. *ZONAsi: Jurnal Sistem Informasi, 7*(2), 445–454. https://doi.org/10.31849/zn.v7i2.26342

Lakutu, N. F., Katili, M. R., Mahmud, S. L., & Yahya, N. I. (2023). Algoritma Dijkstra dan algoritma greedy untuk optimasi rute pengiriman barang pada Kantor Pos Gorontalo. *EULER: Jurnal Ilmiah Matematika, Sains dan Teknologi, 11*(1), 55–65. https://doi.org/10.34312/euler.v11i1.18244

Manalu, E., Napitupulu, V., Afriyani, M., & Butar-butar, H. B. (2025). Penerapan algoritma Dijkstra jalur terpendek antar objek wisata Danau Toba. *Jurnal Manajemen Informatika Jayakarta, 5*(4), 375–386. https://doi.org/10.52362/jmijayakarta.v5i4.2002

Maulana, A. (2025). Penerapan Kruskal minimum spanning tree pada optimasi rute wisata Bojong Kabupaten Tegal. *Jurnal Rekayasa Informatika (JRIT), 2*(1), 36–45.

Muharrom, M. (2020). Implementasi algoritma Dijkstra dalam penentuan jalur terpendek: Studi kasus jarak tempat kuliah terdekat. *Indonesian Journal of Business Intelligence, 3*(1), 25–30. https://doi.org/10.21927/ijubi.v3i1.1229

Musabbikhah, L., & Yuliantari, R. V. (2022). Review artikel: Analisis penggunaan algoritma Dijkstra untuk mencari rute terpendek di rumah sakit. *Edu Elektrika Journal, 11*(1), 1–5. https://doi.org/10.15294/eej.v11i1.53460

Prakoso, A. B., Ariyanto, Y., & Hayati Ririd, A. R. T. (2017). Optimasi rute lokasi wisata Kota Malang menggunakan metode algoritma genetika. *Jurnal Informatika Polinema, 3*(3), 48–52. https://doi.org/10.33795/jip.v3i3.34

Pramadjaya, A., & Rohmawati, I. (2025). Analisis rute terpendek menuju Universitas Pamulang dengan implementasi algoritma Dijkstra: Analysis of the shortest route to Universitas Pamulang using the implementation of Dijkstra's algorithm. *MALCOM: Indonesian Journal of Machine Learning and Computer Science, 5*(3), 953–961. https://doi.org/10.57152/malcom.v5i3.1983

Rahmawati, D., Putri, M., Rosmawati, & Hakim, F. (2023). Analysis of the fastest cost and route using the graph theory and network analysis. *Enthusiastic: International Journal of Applied Statistics and Data Science, 3*(2), 202–211. https://doi.org/10.20885/enthusiastic.vol3.iss2.art7

Sabilla, A. D., & Taufiq, A. (2022). Penerapan algoritma A* pada WebGIS pencarian rute terpendek. *Journal of Information System and Computer, 2*(2), 32–35. https://doi.org/10.34001/jister.v2i2.395

Safetra, M. F. C., Desviona, N., Helmina, Rianti, A., & Prayogi, M. R. (2026). Penerapan teori graf dalam kehidupan sehari-hari. *Algoritma: Jurnal Matematika, Ilmu Pengetahuan Alam, Kebumian dan Angkasa, 4*(1), 52–68. https://doi.org/10.62383/algoritma.v4i1.923

Susanto, B. M., & Hariyanto, A. (2023). Aplikasi penentuan rute terpendek pengambilan sampah berbasis Internet of Things terintegrasi dengan Google Maps API. *Angkasa: Jurnal Ilmiah Bidang Teknologi, 15*(1), 133–140. https://doi.org/10.28989/angkasa.v15i1.1651

Taneo, R. E., Ndun, R., Fallo, D. Y. A., & Do'o, F. (2025). Optimasi jalur terpendek menggunakan algoritma Dijkstra dan greedy pada sistem informasi geografis. *Jurnal Kridatama Sains dan Teknologi, 7*(1), 572–582. https://doi.org/10.53863/kst.v7i01.1664

Rande, P., Witin, S. S. G., Permatasari, W. D., Aminah, N., Amijaya, F. D. T., & Putri, D. F. (2025). Penerapan algoritma branch and bound untuk optimasi rute wisata di Kalimantan Timur berdasarkan traveling salesman problem. *MATHunesa: Jurnal Ilmiah Matematika, 13*(2), 197–205.