PARAMETER
JURNAL MATEMATIKA, STATISTIKA DAN TERAPANNYA

e-ISSN. 2829-6303
Vol. 05, No. 2, September 2026
Pages 233-248

Check for updates

Implementation of Dijkstra Algorithm in Optimizing the Shortest Tourist Route

Rachmania Dwi Widyasti[1], **Wika Dianita Utami**[2*], **Dian Yuliati**[3]

[1,2,3]Department of Mathematics, Faculty of Science and Tachnology,
UIN Sunan Ampel Surabaya
Jl. Ahmad Yani No.117, Surabaya, 60237, Jawa Timur, Indonesia

E-mail Correspondence Author: *wikadianita@uinsa.ac.id*

Abstract

Suroboyo Bus and Wira-Wiri Feeder are public transportation services that support the mobility of Surabaya residents, both for daily activities and tourist visits. However, their use faces challenges in accessing tourist attractions in Surabaya, such as Kebun Binatang Surabaya (KBS), Tugu Pahlawan, Surabaya North Quay (SNQ), Kota Lama, Alun-alun Surabaya, Taman Flora, and THP Kenjeran, due to their distant locations and incomplete transportation access. This study aims to determine the shortest route for visiting tourist destinations by applying Dijkstra's algorithm. This method represents the transportation network as a graph, with bus stops as nodes and routes as weighted edges. In general, 5,040 possible route sequences can be generated from the seven selected tourist destinations. However, this study focuses on THP Kenjeran as the starting point and Alun-alun Surabaya as the destination, with the five remaining tourist destinations required to be visited. Under these conditions, 120 possible route sequences are generated through permutations of the five mandatory destinations. The results of the study show that the shortest path from THP Kenjeran to Alun-alun Surabaya covers a distance of 38.75 km, following the route THP Kenjeran → Kenpark → ITS → ITATS → Taman Flora → Terminal Bratang → Krukah Timur → Stasiun Wonokromo → Terminal Joyoboyo → KBS → Embong Malang → Pirngadi → Tugu Pahlawan → Indrapura → SNQ → Kota Lama → Alun-alun Contong → Tunjungan → Alun-alun Surabaya.

Keywords: *Dijkstra's algorithm, graph, shortest route, surabaya tourism.*

: https://doi.org/10.30598/parameterv5i2pp233-248

Email: jurnalparameter@gmail.com
Homepage: https://ojs3.unpatti.ac.id/index.php/parameter

Submitted: September 2025
Accepted: September 2026

1. INTRODUCTION

The city of Surabaya has an area of 326.81 km² with a population of approximately 3,018,022 as of 2024, giving city a population density of 8,995 people/km² [1]. This high density has led to an increased need for public transportation that can support community mobility. In an effort to address these issues, the City Government has introduced various public transportation innovations, starting with the launch of the Suroboyo Bus in 2018 with a payment system using plastic bottle waste [2], and 2023, a transportation system called wira-wiri feeder was launched [3]. The Suroboyo Bus is designed with 4 routes that only traverse major roads, while the Wira-Wiri Feeder has 11 routes that can reach residential streets.

In supporting an integrated transportation system, the Surabaya City Transportation Agency provides the GOBIS application to monitor fleet movements in real-time and purchase travel tickets [4]. In addition, bus stops have been built along the service route to facilitate access and transfers between modes of transportation [5]. These efforts have proven to increase public interest in public transportation. According to the head of the PTU UPTD of the Surabaya Transportation Department, the number of Suroboyo Bus passengers continues to increase, reaching 1,990,257 in 2024 [6]. Meanwhile, according to the head of the Transportation Team of the Surabaya City Transportation Agency, the number of Wira-Wiri Feeder passengers reached 121,250 in 2024 [7].

The city of Surabaya has great potential in the tourism sector, with its diverse range of tourist destinations, including historical, educational, marine, and culinary tourism [8]. Some popular tourist destinations include Kebun Binatang Surabaya (KBS), Tugu Pahlawan, Surabaya North Quay (SNQ), Kota Lama, Alun-alun Surabaya, Taman Flora, and THP Kenjeran. However, the diversity of tourism is not yet fully supported by the available public transportation system. According to a statement by the Head of the Regional Development Planning Agency (Bappeda Litbang) in Jatim Newsroom, the Old Town Surabaya tourist area can be accessed using public transportation, but only Suroboyo Bus has a direct route to visit the area [9]. Differences in service coverage and routes between Suroboyo Bus and Wira-Wiri Feeder require tourists to switch routes when traveling between tourist destinations. This situation becomes a problem when tourists plan to visit multiple destinations in a single trip. Therefore, an approach is needed to determine an efficient sequence of tourist destinations and transportation routes that can minimize the total distance traveled while still allowing all selected destinations to be visited.

Based on these issues, a solution is needed in the form of efficient routes that tourists can choose to visit multiple tourist attractions at once using public transportation. This issue can be resolved by applying mathematical theory, specifically Graph Theory. Graph theory is widely applied to represent complex relationships between objects in the form of graphs and has been utilized in transportation systems for route optimization and network analysis [10]. One of the algorithms commonly used in graph theory is Dijkstra's algorithm. This algorithm employs a greedy approach to determine the shortest path between two nodes in a directed, weighted graph by repeatedly selecting the node with the minimum distance and updating its distance value [11]. However, Dijkstra's algorithm is designed to determine the shortest path between a specific source and destination, and does not directly determine the optimal order for visiting multiple destinations. Therefore, using Dijkstra's algorithm alone is insufficient for the problem addressed in this study. To overcome this limitation, this study combines Dijkstra's algorithm with permutations. Permutations are used to systematically generate all possible sequences of tourist destinations to be visited, while Dijkstra's algorithm determines the shortest path between each pair of consecutive destinations in every sequence. The distances obtained for each segment are then summed to calculate the total distance of each possible route. By evaluating all generated sequences using the same shortest-path procedure, this approach compares all visit sequences based on the minimum total distance traveled. Thus, the combination of permutations and the Dijkstra algorithm is suitable for multi-destination tourism route planning problems.

Several previous studies have explored the application of Dijkstra's algorithm, including research conducted by Patricia et al. on determining the shortest route for TransJakarta to tourist destinations in Jakarta, which found that the best route had a total distance of 46.41 km **[12]**. Another study, namely the determination of the shortest route on school bus routes conducted by I.P.W.Gautama, et al., obtained an optimal total distance of 95.5 km **[13]**. This method has been proven to be superior in determining the shortest route, as in a study conducted by Tri et al. on the comparison of ant and Dijkstra algorithms in determining the shortest route for goods distribution, which found that the Dijkstra algorithm provides the most optimal route for distributing goods from the origin to the destination **[14]**. Furthermore, research by Wahyadi M. Rhifky et al. on the comparison of Greedy, A-Star, and Dijkstra's algorithms in finding the shortest path yielded results indicating that the Dijkstra algorithm can find all nodes that are target destinations in complex graphs **[15]**. These studies indicate that Dijkstra's algorithm is capable of effectively determining shortest paths in various transportation and routing problems. However, its application in the present study is extended by combining it with permutations to address the additional problem of determining the sequence of visits among multiple tourist destinations.

Although various studies have successfully applied the Dijkstra algorithm in determining the shortest route, there are still studies that have not been explored much, such as determining an efficient visit sequence when multiple destinations must be visited in a single trip. Therefore, this study proposes an approach to optimize various visit routes by integrating permutation techniques with Dijkstra algorithm. This approach considers various possible sequences of tourist visit routes and evaluates each route based on the shortest travel distance within the transportation network. The objective of this study is to provide efficient tourist routes for visiting seven destinations in Surabaya using the Suroboyo Bus and Wira-Wiri Feeder services. The results of this study are expected to help tourists choose the right travel routes and transportation transfers, while also providing insights for improving public transportation connectivity to tourist areas in Surabaya.

2. RESEARCH METHODS

2.1. Graph

A graph is a diagram that can represent objects with points or vertices connected by lines or edges, making it easier to understand the relationships between objects. A graph G contains two finite sets (V, E), where V is a non-empty set of vertices (V(G)) and E is a set of edges connecting one vertex to another (E(G)) **[16]**.

The graph has the equation (1) **[17]**.

$$G = (V, E) \tag{1}$$

When $V = \{v_1, v_2, v_3, \dots, v_n\}$ and $E = \{e_1, e_2, e_3, \dots, e_n\}$

And the representation of the graph can be seen in **Figure 1** below.

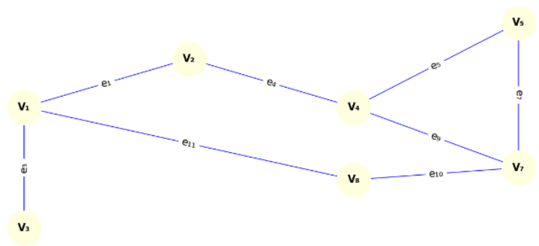


Figure 1. Graph

In graph theory, a graph is generally classified into two types: directed and undirected graphs [18]. In an undirected graph, each edge does not have a specific direction. As can be seen in **Figure 1**, V_7 and V_8 can be considered as two-way nodes, so they have a relationship ($V_7 \rightarrow V_8$) and ($V_8 \rightarrow V_7$). In contrast, a directed graph assigns a specific orientation to each edge, represented by an arrowhead. The orientation may be outgoing or incoming, showing the connection from the source node to the target node. The representation of a directed graph can be see in **Figure 2**.

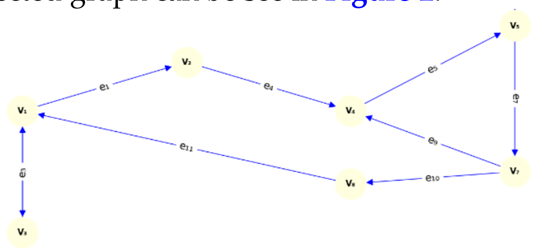


Figure 2. Directed Graph

In a graph, there are neighboring vertices, which are two vertices that are directly connected by an edge. For example, in **Figure 2**, the neighboring vertices of V_1 are V_2, V_3 and V_8. Furthermore, graphs can also be classified into two categories, namely weighted graphs and unweighted graphs. A weighted graph is a graph in which each edge has a value attached to the line connecting two nodes [18], a weighted graph is shown in **Figure 3**. In a graph, weights are usually determined according to the problem's objective—for example, representing the weight as the distance between locations to assist in identifying the shortest path.

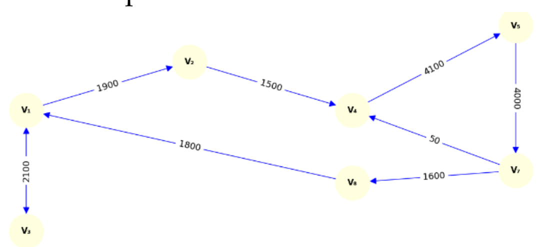


Figure 3. Wighted Graph

2.2. Djikstra Algorithm

The Dijkstra algorithm is one of the algorithms commonly used in solving problems related to transportation planning, logistics optimization, and route search on a graph/network [19]. This algorithm was first introduced by Dutch computer scientist Edsger W. Dijkstra in 1956 [20]. The purpose of the Dijkstra algorithm is to find the shortest path from a starting node to a destination, based on the minimum weight of each step on the available path [21]. The steps for calculating the Dijkstra algorithm [19] are as follows:

1. Perform initialization:
   - Determine the starting node and assign it a distance value of 0 and a permanent label(*)
   - Assign other nodes a distance value of ∞ and a temporary label
   - Take the starting node, for example "s" as the current node
2. Update the distance and set the current node:
   - Suppose V_i is the current node and node V_j is a neighbor node with a temporary label that is directly connected to node V_i
   - Update the distance value of node j using the formula

$$dV_j = min\{dV_j, dV_i + cV_{i,j}\} \tag{2}$$

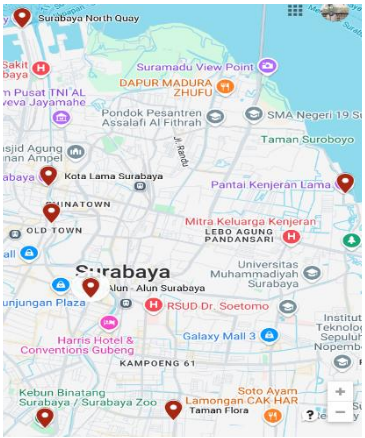
   With:
   dV_j = Temporary distance to node j
   dV_i = Permanent distance to current node i
   $cV_{i,j}$ = Weight of the edge between node i and j
   - Select the node j^* that has the smallest distance, namely

$$d_{j*} = min\ dV_j \tag{3}$$

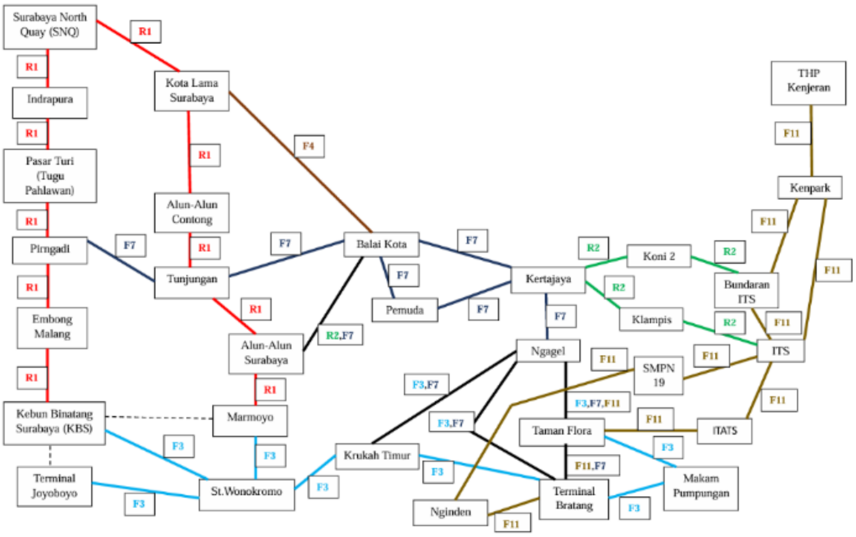
   Then change the j^* node to a permanent node and designate that node as the current node (for the next iteration)
3. The process stops, when all nodes reachable from the initial node s have been permanently labeled or when the destination node is reached. If there are still nodes with temporary labels, repeat step 2

2.3. Method

The data in this study consists of seven popular tourist attractions in Surabaya, namely Surabaya Zoo (KBS), Taman Flora, Tugu Pahlawan, Surabaya North Quay (SNQ), Kota Lama, Alun-alun Surabaya and Kenjeran Beach Park (THP). The distance-based weights between bus stops were derived from Google Maps, while the public transportation network map was sourced from the Surabaya Transportation Discussion Forum. Research data was collected on May 10, 2025. The modes of transportation used were Suroboyo Bus units R1 (Purabaya-Pelabuhan Tanjung Perak) and R2 (Kejawan Putih Tambak-UNESA), as well as Wira-Wiri Feeder units F3 (Terminal Joyoboyo-Pasar Gunung Anyar), F4 (Sier-Kota Lama), F7 (Terminal Bratang-Stasiun Pasar Turi), and F11 (Terminal Bratang-Shelter Bulak) that pass through these tourist locations. Seven tourist sites and the transportation network connecting them are illustrated in Figure 4.



(a)



(b)

Figure 4. (a)tourist attractions and (b)network map

The graph in **Figure 5** will be processed using the Dijkstra algorithm to determine the shortest route for visiting seven tourist attractions in Surabaya. The process can be seen in the following flowchart

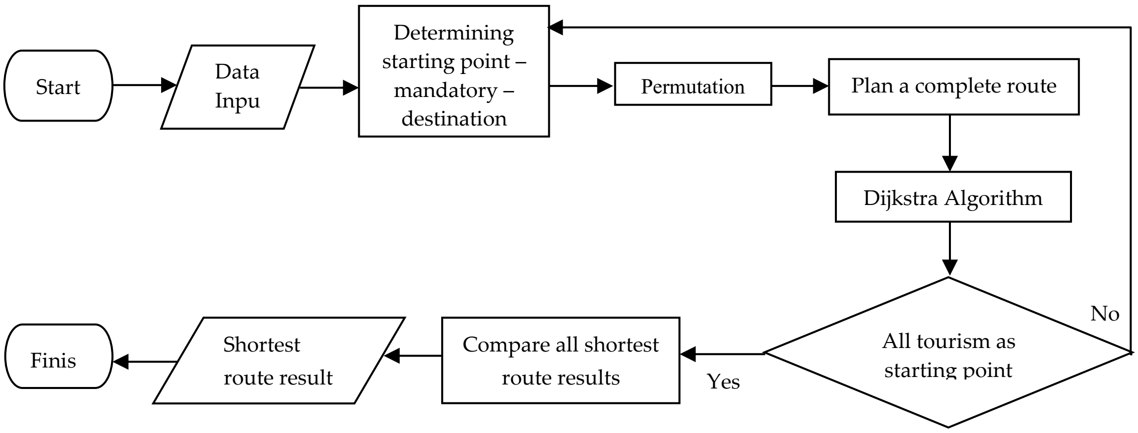


Figure 5. Flowchart for determining shortest route using permutations and Dijkstra algorithm

Based on **Figure 5** above, the following research steps were obtained :

1. Input data in the form of a directed graph with nodes and weights.
2. Determine the starting node, destination node, and nodes that must be visited.
3. From the nodes that must be visited, permutations are performed to produce all possible route sequences.
4. Construct a complete route, starting from the initial node, the sequence of mandatory nodes to visit, and the final node.
5. From step 4 above, perform calculations using the Dijkstra algorithm. In each segment, the distance will be update using formula (2), then take the smallest distance value to be used as the current node (next iteration) based on equation (3), and the process stops when it has reached the destination node. Then, add up the shortest total distance obtained in each segment.
6. Analyze other routes by determining the starting node from the existing tourist attractions. If all tourist attractions have not been selected, repeat steps 2, 3, 4, and 5.
7. Compare all the results of the shortest route search for visiting seven tourist attractions to obtain the most optimal route.

3. RESULTS AND DISCUSSION

Based on **Figure 4.(b)**, the network map is illustrated in a graph. The graph represents bus stops as nodes, with routes between them forming edges that carry both direction and weight. The weight on each edge of the graph represents the distance between bus stops in meters. Thus, a graph is formed from the network map connecting seven tourist attractions in Surabaya in the form of a complex graph that has direction and weight. The naming of nodes in the graph is explained in Table 1, and the weighted and directed graph is show in **Figure 5** below.

Table 1. Node Identification

Node	Bus Stop (Halte)	Node	Bus Stop (Halte)
V_1	Kebun Binatang Surabaya (KBS)	V_{16}	Bundaran ITS
V_2	Embong Malang	V_{17}	Kenpark
V_3	Pirngadi	V_{18}	THP Kenjeran
V_4	Pasar Turi (Tugu Pahlawan)	V_{19}	Klampis
V_5	Indrapura	V_{20}	ITS
V_6	Surabaya North Quay (SNQ)	V_{21}	SMPN 19
V_7	Kota Lama Surabaya	V_{22}	ITATS
V_8	Alun-alun Contong	V_{23}	Ngagel
V_9	Tunjungan	V_{24}	Taman Flora
V_{10}	Balai Kota	V_{25}	Makam Pumpungan
V_{11}	Alun-alun Surabaya	V_{26}	Terminal Bratang
V_{12}	Marmoyo	V_{27}	Nginden
V_{13}	Kertajaya	V_{28}	Krukah Timur
V_{14}	Pemuda	V_{29}	St.Wonokromo
V_{15}	Koni 2	V_{30}	Terminal Joyoboyo

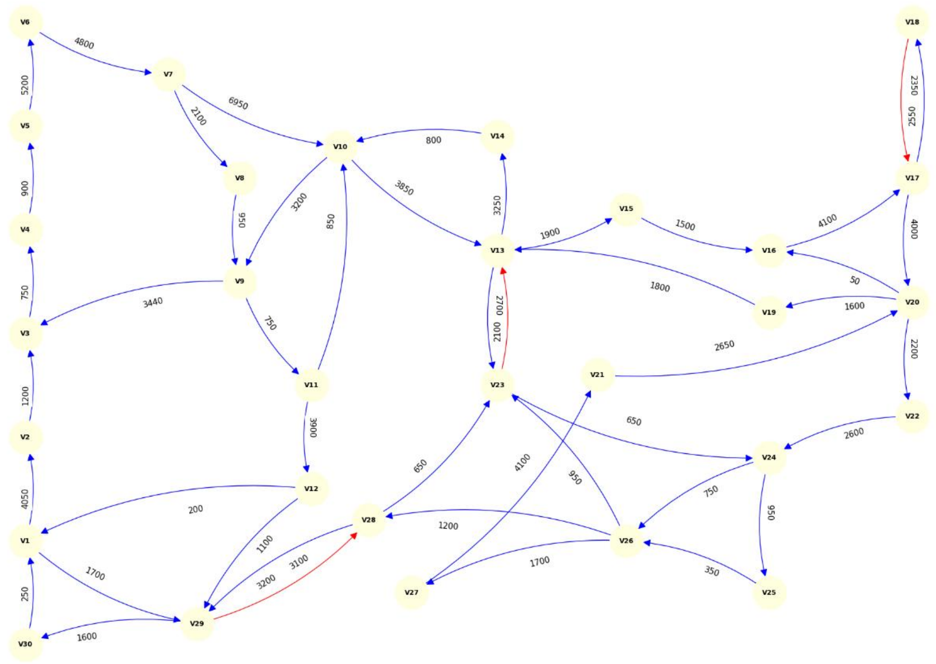


Figure 5. Graph representing tourism in Surabaya

To determine the shortest route to visit seven tourist attractions in Surabaya, it can be done by trying all possible tourist attractions as starting points for destinations. The first step is to determine the starting node, as well as the nodes that must be passed through to reach the destination node. Thus, a complete visit route is obtained and can be processed using the Dijkstra algorithm.

The first experiment determined the shortest route from THP Kenjeran to Alun-alun Surabaya, which must pass through Kebun Binatang Surabaya (KBS), Tugu Pahlawan, Surabaya North Quay (SNQ), Taman Flora, and Kota Lama. The Dijkstra algorithm was run on each segment to obtain the shortest distance on the route.

Segment 1 (THP Kenjeran → Taman Flora)

The initial node selected is the Kenjeran Beach bus stop V_{18} , which is assumed to be the current node with a distance value of 0 and a permanent label, while the other bus stops have a distance value ∞ with temporary labels.

Table 2. Shortest route results from THP Kenjeran → Taman Flora

Iteration		V_{18}	V_{17}	V_{20}	V_{16}	V_{19}	V_{22}	V_{13}	V_{24}
		0*	∞	∞	∞	∞	∞	∞	∞
1	V_{18}	0*	2350	∞	∞	∞	∞	∞	∞
2	V_{17}	0*	2350*	6350	∞	∞	∞	∞	∞
3	V_{20}	0*	2350*	6350*	6400	7950	8550	∞	∞
4	V_{16}	0*	2350*	6350*	6400*	7950	8550	∞	∞
5	V_{19}	0*	2350*	6350*	6400*	7950*	8550	9750	∞
6	V_{22}	0*	2350*	6350*	6400*	7950*	8550	9750	11150

Based on **Table 2** above, the first iteration shows that from node V_{18}, there is only one node that can be reached directly, namely the Kenpark V_{17} stop, so the distance value for that node will be updated

$$dV_{17} = min\{\infty, 0 + 2350\} = 2350$$

Next, node V_{17} is marked with (*), indicating that this node is permanently labeled and has been selected as the current node for the next iteration. Then, in the second iteration, node V_{17} can only reach ITS stop node V_{20}, so the distance value is updated

$$dV_{20} = min\{\infty, 2350 + 4000\} = 6350$$

And that node is marked with an asterisk (*), indicating that it has been selected as the current node for the next iteration.

The same steps are repeated in subsequent iterations by updating the distance and label until the destination node is reached. The implementation of the Dijkstra algorithm in this segment obtained the shortest route value of 11,150 meters with the nodes passed being THP Kenjeran → Kenpark → ITS → ITATS → Taman Flora.

Segment 2 (Taman Flora → Kebun Binatang Surabaya (KBS))

The initial node selected is Flora Park V_{24} as the current node with an initial value of 0, while the other stops have a value of ∞

Table 3. Shortest route results from Taman Flora → Kebun Binatang Surabaya (KBS)

Iteration		V_{24}	V_{25}	V_{26}	V_{23}	V_{28}	V_{27}	V_{13}	V_{29}	V_{21}	V_{14}	V_{15}	V_{30}	V_{16}	V_1
		0*	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	V_{24}	0*	950	750	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	V_{26}	0*	950	750*	1700	1950	2450	∞	∞	∞	∞	∞	∞	∞	∞
3	V_{25}	0*	950*	750*	1700	1950	2450	∞	∞	∞	∞	∞	∞	∞	∞
4	V_{23}	0*	950*	750*	1700*	1950	2450	3800	∞	∞	∞	∞	∞	∞	∞
5	V_{28}	0*	950*	750*	1700*	1950*	2450	3800	5050	∞	∞	∞	∞	∞	∞
6	V_{27}	0*	950*	750*	1700*	1950*	2450*	3800	5050	6550	∞	∞	∞	∞	∞
7	V_{13}	0*	950*	750*	1700*	1950*	2450*	3800*	5050	6550	7050	5700	∞	∞	∞
8	V_{29}	0*	950*	750*	1700*	1950*	2450*	3800*	5050*	6550	7050	5700	6650	∞	∞
9	V_{15}	0*	950*	750*	1700*	1950*	2450*	3800*	5050*	6550	7050	5700*	6650	7200	∞
10	V_{30}	0*	950*	750*	1700*	1950*	2450*	3800*	5050*	6550	7050	5700*	6650*	7200	6900

Table 3 above shows the results of the first iteration calculation, namely that node V_{24} has a direction that can directly reach two nodes, namely Makam Pumpungan V_{25} and Terminal Bratang V_{26}, so that the distance for this node is updated.

$$dV_{25} = min(\infty, 0 + 950) = 950$$

$$dV_{26} = min(\infty, 0 + 750) = 750$$

Of the two nodes above, V_{26} has the minimum value, so this node is changed to a permanent node (*) and becomes the current node in the next iteration.

The same steps are applied in subsequent iterations until the destination node, Surabaya Zoo, is reached. In this segment, the implementation of the Dijkstra algorithm finds the shortest value of 6,900 meters with a path starting from Taman Flora → Terminal Bratang → Krukah Timur → St.Wonokromo → Terminal Joyoboyo → Kebun Binatang Surabaya (KBS).

Segment 3 (Kebun Binatang Surabaya (KBS) → Tugu Pahlawan)

The initial node used is V_1 KBS, so this node is a permanently labeled node with a distance value of 0, and the other nodes have a distance value of ∞

Table 4. Shortest route results from Kebun Binatang Surabaya (KBS) →Tugu Pahlawan

Iteration		V_1	V_2	V_{29}	V_{28}	V_{30}	V_3	V_{23}	V_4
		0*	∞	∞	∞	∞	∞	∞	∞
1	V_1	0*	4050	1700	∞	∞	∞	∞	∞
2	V_{29}	0*	4050	1700*	4900	3300	∞	∞	∞
3	V_{30}	0*	4050	1700*	4900	3300*	∞	∞	∞
4	V_2	0*	4050*	1700*	4900	3300*	5250	∞	∞
5	V_{28}	0*	4050*	1700*	4900*	3300*	5250	5550	∞
6	V_3	0*	4050*	1700*	4900*	3300*	5250*	5550	6000

Based on **Table 4** above, the first iteration shows that node V_1 can directly reach nodes V_2 and V_{29}, so the distance between these nodes is updated.

$$dV_2 = min\{\infty, 0 + 4050\} = 4050$$

$$dV_{29} = min\{\infty, 0 + 1700\} = 1700$$

Among these two nodes, the smallest distance is found in node V_{29}, so this node is labeled permanently and selected as the current node in the next iteration.
The same steps are performed until the calculation reaches the destination node. In this segment, namely KBS to Tugu Pahlawan, the Dijkstra algorithm obtains the shortest distance value of 6,000 meters with the route taken by Kebun Binatang Surabaya (KBS) → Embong Malang → Pirngadi → Tugu Pahlawan.

Segment 4 (Tugu Pahlawan → *Surabaya North Quay* (SNQ))

In this segment, the starting node is the Pasar Turi bus stop V_4, which is close to the Tugu Pahlawan tourist attraction, so this node becomes a permanently labeled node with a distance value of 0, and the other nodes have a value of ∞

Table 5. Shortest route results from Tugu Pahlawan → Surabaya North Quay (SNQ)

Iteration		V_4	V_5	V_6
		0*	∞	∞
1	V_4	0*	900	∞
2	V_5	0*	900*	6100

Based on the **table 5** above, the first iteration, node V_4, can only reach node V_5, so the updated distance value will be the minimum distance and become the current node in the next iteration.

$$dV_5 = min\{\infty, 0 + 900\} = 900$$

In the second iteration, it shows that node V_5 can only reach node V_6, which is the Pelabuhan Tanjung Perak bus stop, when visiting Surabaya North Quay, which is the destination node of this segment. Thus, the iteration is completed with the shortest distance from the implementation of the Dijkstra algorithm obtained as

$$dV_6 = min\{\infty, 900 + 5200\} = 6{,}100$$

And the path taken by this segment, starting from Tugu Pahlawan → Indrapura → SNQ.

Segment 5 (*Surabaya North Quay* (SNQ) → Kota Lama)

The initial node is Surabaya North Quay V_6, so the initial value is 0 for this sample and ∞ for the other samples.

Table 6. Shortest route results from Surabaya North Quay (SNQ) → Kota Lama

Iteration		V_6	V_7
		0*	∞
1	V_6	0*	4800

Table 6, shows that node V_6 can only reach node V_7, where node V_7 is the destination on this segment. Therefore, the shortest distance value on this segment path is obtained directly from the weight between nodes V_5 and V_6, which is 4,800 meters.

Segment 6 (Kota Lama → Alun-alun Surabaya)

The initial node used in this segment is V_7, which is the Jembatan Merah bus stop for access to Kota Lama tourist attractions. Therefore, node V_7 becomes the current node with a distance value of 0, while the other bus stops have a distance value of ∞.

Table 7. Shortest route results from Kota Lama → Alun-alun Surabaya

Iteration		$\boldsymbol{V_7}$	$\boldsymbol{V_8}$	$\boldsymbol{V_{10}}$	$\boldsymbol{V_9}$	$\boldsymbol{V_3}$	$\boldsymbol{V_{11}}$
		0*	∞	∞	∞	∞	∞
1	V_7	0*	2100	6950	∞	∞	∞
2	V_8	0*	2100*	6950	3050	∞	∞
3	V_9	0*	2100*	6950	3050*	6490	3800

Based on **table 7** above, in the first iteration, node V_7 is shown to have two directly reachable nodes, namely Balai Kota bus stop V_{10} and Alun-alun Contong V_8 , so the distance values for these nodes are updated.

$$dV_8 = min\{\infty, 0 + 2100\} = 2100$$

$$dV_{10} = min\{\infty, 0 + 6950\} = 6950$$

From the two nodes above, V_8 has the minimum distance value, so this node becomes a permanently labeled node and is used as the current node in the next iteration.

The same steps are repeated until the destination node is reached, resulting in the implementation of Dijkstra's algorithm on this segment being 3,800 meters with the route passing through Kota Lama → Alun-alun Contong → Tunjungan → Alun-alun Surabaya. After running Dijkstra's algorithm to find the shortest route from the complete route of Kenjeran Beach Park (THP) to Alun-alun Surabaya. Shortest values obtained for each segment are summed as follows: 11,150 + 6,900 + 6,000 + 6,100 + 4,800 + 3,800 = 38,750 meters. Therefore, the shortest route value obtained is 38.75 km with the route THP Kenjeran → Kenpark → ITS → Taman Flora → Terminal Bratang → Krukah Timur → St.Wonokromo → Terminal Joyoboyo → Kebun Binatang Surabaya (KBS) → Embong Malang → Pirngadi → Tugu Pahlawan → Indrapura → SNQ → Kota Lama → Alun-alun Contong → Tunjungan → Alun-alun Surabaya.

The same steps is find the shortest route using the Dijkstra algorithm were applied to five other complete routes, each of which began with a different tourist attraction as the starting node. The shortest routes for these five routes are shown in the following table.

Table 8. Shortest route from other possible routes

Route	Shortest Route	Path Followed
Kebun Binatang Surabaya (KBS) → Kota Lama	56,940 meters (56.94 Km)	Kebun Binatang Surabaya (KBS) → St.Wonokromo → Krukah Timur → Ngagel → Taman Flora → Terminal Bratang → Ngagel → Kertajaya → Koni 2 → Bundaran ITS → Kenpark → THP Kenjeran → Kenpark → ITS → Klampis → Kertajaya → Pemuda → Balai Kota → Tunjungan → Alun-alun Surabaya → Balai Kota → Tunjungan → Pirngadi → Tugu Pahlawan → Indrapura → SNQ → Kota Lama
Tugu Pahlawan → Alun-alun Surabaya	56,600 meters (56.6 Km)	Tugu Pahlawan → Indrapura → SNQ → Kota Lama → Alun-alun contong → Tunjungan → Alun-alun Surabaya → Marmoyo → KBS → St.Wonokromo → Krukah Timur → Ngagel → Taman Flora → Terminal Bratang → Ngagel → Kertajaya → Koni 2 → Bundaran ITS → Kenpark → THP Kenjeran → Kenpark → ITS → Klampis → Kertajaya → Pemuda → Balai Kota → Tunjungan → Alun-alun Surabaya
Kota Lama → *Surabaya North Quay* (SNQ)	48,700 meters (48.7 Km)	Kota Lama → Alun-alun contong → Tunjungan → Alun-alun Surabaya → Balai Kota → Kertajaya → Koni 2 → Bundaran ITS → Kenpark → THP Kenjeran → Kenpark → ITS → ITATS → Taman Flora → Terminal Bratang → Krukah Timur → St.Wonokromo → Terminal Joyoboyo → KBS → Embong Malang → Pirngadi → Tugu Pahlawan → Indrapura → SNQ
Alun-alun Surabaya → Kota Lama	49,700 meters (49.7 Km)	Alun-alun Surabaya → Balai Kota → Kertajaya → Koni 2 → Bundaran ITS → Kenpark → THP Kenjeran → Kenpark → ITS → ITATS → Taman Flora→ Terminal Bratang → Krukah Timur → St.Wonokromo → Terminal Joyoboyo → KBS → Embong Malang → Pirngadi → Tugu Pahlawan → Indrapura → SNQ → Kota Lama
Taman Flora → Alun-alun Surabaya	52,600 meters (52.6 Km)	Taman Flora → Terminal Bratang → Ngagel → Kertajaya → Koni 2 → Bundaran ITS → Kenpark → THP Kenjeran → Kenpark → ITS → ITATS → Taman Flora → Terminal Bratang → Krukah Timur → St.Wonokromo → Terminal Joyoboyo → Kebun Binatang Surabaya (KBS) → Embong Malang → Pirngadi → Tugu Pahlawan → Indrapura → SNQ → Kota Lama → Alun-alun contong → Tunjungan → Alun-alun Surabaya

In a previous study conducted by Hendra et al., the determination of the shortest route focused only on trips from Gubeng Station as a fixed starting point with several tourist attractions in Surabaya as destinations. The results of this study showed that the shortest route obtained was from Gubeng Station to one tourist destination, namely Jalan Taman Apsari **[22]**. Unlike that study, this study analyzes the shortest route with various possible starting points and destinations, and integrates five must-visit tourist spots into one trip. This approach results in a more complete route for visiting various destinations in one trip using public transportation in Surabaya.

4. CONCLUSION

Based on research on determining the shortest route for visiting seven tourist attractions in Surabaya, namely Kebun Binatang Surabaya (KBS), Tugu Pahlawan, Surabaya North Quay (SNQ), Kota Lama, Alun-Alun Surabaya, Taman Flora, and THP Kenjeran, by implementing the Dijkstra algorithm. Six experiments were conducted on various possible visit routes, and the shortest route was obtained, starting from a visit to Kenjeran Beach Park (THP) to Surabaya Square. Using the Suroboyo Bus and Wira-Wiri feeder transportation services, the route taken was THP Kenjeran → Kenpark → ITS → Taman Flor**a** → Terminal Bratang → Krukah Timur → St.Wonokromo → Terminal Joyoboyo → KBS → Embong Malang → Pirngad**i** → Tugu Pahlawan → Indrapura → SNQ → Kota Lama → Alun-alun **C**ontong → Tunjungan → Alun-**a**lun Surabaya dengan jarak tempuh sebesar 38,750 meters (38.75 Km). This study is limited to using travel distance as the main consideration and does not include dynamic factors such as traffic conditions. Future research can incorporate real-time transportation data and other travel-time factors to improve route optimization.

Acknowledgments
Special thanks go to the supervisor for his guidance and direction.

Funding information
Authors state no funding involved.

Conflict of interest statement
Authors state no conflict of interest.

Data Availability
Public transportation network map can see at @tourism.surabaya

REFERENCES

[1] Badan Pusat Statistika, *Kota Surabaya Dalam Angka 2025*, vol. 37. 2025.

[2] M. Kibthiah, R. N. Chamida, U. N. Surabaya, and K. Khotimah, "Suroboyo Bus Sebagai Sistem Transportasi Berkelanjutan di Kota Surabaya," *Jurnal Transportasi, FSTPT*, vol. 23, no. 1, pp. 11–18, Apr. 2023.

[3] M. Zahro Rumana, A. Puspaningtyas, and Y. Hariyoko, "Evaluasi Kebutuhan Wira Wiri Suroboyo Sebagai Transportasi Publik Kota Surabaya," *PRAJA Observer: Jurnal Penelitian Administrasi Publik*, vol. 4, no. 04, 2024.

[4] Z. A. Haqie, P. Studi, I. Pemerintahan, and O. P. Ariyani, "Inovasi Pelayanan Publik Suroboyo Bus di Kota Surabaya," *Journal of Public Sector Innovations*, vol. 5(1), pp. 23–30, 2020.

[5] R. E. Wibisono and R. F. Putri, "Evaluasi Kualitas Pelayanan Halte Terminal Purabaya Pada Transportasi Umum Suroboyo Bus," *Ge-STRAM: Jurnal Perencanaan dan Rekayasa Sipil*, vol. 5, no. 2, pp. 72–78, Sep. 2022, doi: 10.25139/jprs.v5i2.4636.

[6] B. Constantine Koloway, "Alasan Meningkatnya Jumlah Penumpang Suroboyo Bus pada 2024, Ada Tarif Integrasi Sampai Rute Baru," TribunJatim.com. Accessed: Apr. 25, 2025. [Online]. Available: https://jatim.tribunnews.com/2025/01/15/alasan-meningkatnya-jumlah-penumpang-suroboyo-bus-pada-2024-ada-tarif-integrasi-sampai-rute-baru

[7] Dinas Komunikasi dan Informatika Provinsi Jawa Timur, "Penumpang Feeder WiraWiri Terus Bertambah, Dishub Surabaya Segera Tambah Rute di Akhir 2024," Dinas Kominfo Provinsi Jawa Timur. Accessed: May 02, 2025. [Online]. Available: https://kominfo.jatimprov.go.id/berita/penumpang-feeder-wirawiri-terus-bertambah-dishub-surabaya-segera-tambah-rute-di-akhir-2024

[8] T. Akwila, V. C. Nurmanto, and A. Parantika, "Wisata Gastronomi Sebagai Daya Tarik Destinasi di Surabaya," *Jurnal Ilmiah Wahana Pendidikan*, vol. 9, no. 7, pp. 467–476, 2023, doi: 10.5281/zenodo.7826868.

[9] Dinas Komunikasi dan Informatika Provinsi Jawa Timur, "Inilah Beberapa Aksesibilitas Menuju Wisata Kota Lama Surabaya." Accessed: May 28, 2025. [Online]. Available: https://kominfo.jatimprov.go.id/berita/launching-23-juni-2024-inilah-beberapa-aksesibilitas-menuju-wisata-kota-lama-surabaya

[10] R. Gore, T. Gore, N. Rokade, and Y. Mandlik, "Applications of Graph Theory," *International Journal Of Latest Technology In Engineering, Management & Applied Scince (IJLTEMAS)*, vol. XIV, pp. 148–150, 2025, doi: 10.51583/IJLTEMAS.

[11] Jason, M. Siever, A. Valentino, K. M. Suryaningrum, and R. Yunanda, "Dijkstra's algorithm to find the nearest vaccine location," in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 5–12. doi: 10.1016/j.procs.2022.12.105.

[12] P. J. B. B. Tapobali and C. N. Krisnamurti, "Implementasi Algoritma Dijkstra Dalam Menentukan Rute Terpendek Bus Transjakarta Dalam Mengunjungi 5 Destinasi Wisata Populer Di Jakarta," *E-Jurnal Matematika*, vol. 9, no. 4, p. 265, Dec. 2020, doi: 10.24843/mtk.2020.v09.i04.p307.

[13] I. P. W. Gautama and K. Hermanto, "Penentuan Rute Terpendek dengan Menggunakan Algoritma Dijkstra pada Jalur Bus Sekolah," *Jurnal Matematika*, vol. 10, no. 2, p. 116, Dec. 2020, doi: 10.24843/jmat.2020.v10.i02.p128.

[14] T. Azisah Nurman, R. Ibnas, and A. Nurul Hidayat, "Penerapan Algoritma Dijkstra dan Algoritma Semut pada Pendistribusian Barang."

[15] M. R. Wayahdi, S. H. N. Ginting, and D. Syahputra, "Greedy, A-Star, and Dijkstra's Algorithms in Finding Shortest Path," *International Journal of Advances in Data and Information Systems*, vol. 2, no. 1, pp. 45–52, Feb. 2021, doi: 10.25008/ijadis.v2i1.1206.

[16] A. Rafi' Addani, T. Turmudi, and I. Sujarwo, "Penerapan Graf Berarah dan Berbobot untuk Mengetahui Inluencer yang Paling Berpengaruh dalam Penyebaran Informasi pada Twitter," *Jurnal Riset Mahasiswa Matematika*, vol. 2, no. 5, pp. 186–194, Jun. 2023, doi: 10.18860/jrmm.v2i5.16810.

[17] P. Buhaerah, Z. Busrah, and H. Sanjaya, *Teori Graf Dan Aplikasinya*, 1st ed. Makassar: Living Spiritual Quotient.

[18] A. Majeed and I. Rauf, "Graph theory: A comprehensive survey about graph theory applications in computer science and social networks," Mar. 01, 2020, *MDPI Multidisciplinary Digital Publishing Institute*. doi: 10.3390/inventions5010010.

[19] Md. Mehedi Hassan, Md. Asadujjaman, and Md. Golam Robbani, "A Modified Approach of Dijkstra's Method for Finding Shortest Path in a Weighted Directed Graph," *American Journal of Science, Engineering and Technology*, Jul. 2023, doi: 10.11648/j.ajset.20230803.14.

[20] A. Durand, T. Watteau, G. Ghazi, and R. M. Botez, "Generalized Shortest Path Problem: An Innovative Approach for Non-Additive Problems in Conditional Weighted Graphs," *Mathematics*, vol. 12, no. 19, Oct. 2024, doi: 10.3390/math12192995.

[21] A. Cantona, Fauziah, and Winarsih, "Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta," *JTMI Jurnal Teknologi dan Manajemen Informatika*, vol. 6, no. 1, 2020, [Online]. Available: http://http://jurnal.unmer.ac.id/index.php/jtmi

[22] H. Hendra and Y. F. Riti, "Perbandingan Algortima Dijkstra dan Floyd-Warshall dalam Menentukan Rute Terpendek Stasiun Gubeng Menuju Wisata Surabaya," *JIKA (Jurnal Informatika)*, vol. 6, no. 3, p. 297, Oct. 2022, doi: 10.31000/jika.v6i3.6528.