Comparison of the Exhaustive Algorithm and Johnson's Algorithm in Determining the Shortest Path to Sipiso Piso Tourism

Indra Widiasto, Khairul

Abstract

Determining the shortest route to a tourist destination is an important aspect of travel planning and tourism management, especially for locations with many alternative routes like Sipiso-Piso Tourism. This study aims to compare the performance of two algorithms in determining the shortest path, namely the Exhaustive algorithm and Johnson's Algorithm. The Exhaustive method evaluates all possible routes to find the optimal path, while Johnson's Algorithm combines a weighted directed graph approach for efficient shortest path calculations on a complete graph. Research data were collected through route mapping and distance weighting between travel points to the Sipiso-Piso tourist destination, and then processed using both algorithms. The analysis results show that the Exhaustive Algorithm is capable of finding the shortest path with high accuracy, but its computation time significantly increases as the number of travel points increases. Meanwhile, Johnson's Algorithm offers better time efficiency while still being able to produce the shortest path with a competitive level of accuracy. These findings have practical implications for the development of algorithm-based tourism navigation systems that prioritise a balance between accuracy and computational efficiency.

Keywords: *Shortest Path, Exhaustive Algorithm, Johnson's Algorithm, Sipiso-Piso Tour, Computational Efficiency*

[1] Indra Widiasto
[1] Master of Information Technology, Universitas Pembangunan Panca Budi, Indonesia
e-mail: indrawidiasto@gmail.com

[2] Khairul
[2] Master of Information Technology, Universitas Pembangunan Panca Budi, Indonesia
e-mail: khairul@dosen.pancabudi.ac.id
3rd International Cofference on the Epicentrum of Economic Global Framework (ICEEGLOF)
Theme: Navigating The Future: Business and Social Paradigms in a Transformative Era.
https://proceeding.pancabudi.ac.id/index.php/ICEEGLOF

Introduction

Nature tourism has become one of the significant sectors in the development of tourism in Indonesia. One of the destinations that attracts tourists is the Sipiso-Piso Waterfall, located in Karo Regency, North Sumatra. This location offers exotic natural scenery and diverse access routes, making trip planning to the site important to enhance the tourist experience. The efficiency of travel routes not only affects the comfort of tourists but also impacts travel time, travel costs, and the management of visitor flow at tourist destinations. This study aims to investigate how natural resources can contribute to the growth of the tourism industry. in Indonesia that hasn't yet existed thanks to actions taken by the government, businesses, and locals to increase the standard of living of those attracted and to raise the caliber of tourism. This study was carried out through strategic planning and solution development using qualitative descriptive approaches [1].

In the context of travel route planning, the shortest path algorithm plays a crucial role in determining the optimal route among the many available alternatives. One of the classic methods is the Exhaustive Algorithm, which evaluates all possible routes to find the shortest path. This method has high accuracy but often encounters challenges with computational complexity as the number of travel points increases. As an alternative, Johnson's Algorithm was introduced to improve efficiency in calculating the shortest distance on weighted and complete graphs while maintaining calculation accuracy. The research conducted successfully developed an application for finding the best route (based on schedule departure, shortest distance, and number of fleet transfers) based on Google Maps on the Android operating system with a case study of Trans-J [2].

The purpose of this study is to identify the elements that influence young people's inclination to travel. Multiple linear regression analysis is used as a non-probability sampling strategy. Cost, time and distance traveled, natural attractions, shopping options, and education all have a significant impact on young people under 34's propensity to visit, according to research findings and previously reviewed discussions [3].

The performance of navigation systems and user experience are significantly impacted by algorithm selection, according to prior research in the field of route optimization. However, there is currently no study that explicitly compares Johnson's Algorithm to the Exhaustive Algorithm, particularly in relation to Indonesian tourist locations. With an emphasis on computing efficiency and route accuracy, this study attempts to assess and compare the two algorithms in order to find the quickest route to Sipiso-Piso Tourism. It is intended that the findings of this study will aid in the creation of an ideal algorithm-based tourism navigation system and offer travelers and tourism management helpful information for more efficient trip planning.

This necessitates the ability of network administrators to set up a network using specific routing protocols and algorithms. By comparing all of the network's nodes, Johnson's algorithm—which uses the all-pair shortest path to determine the packet forwarding route—can be used to SDN. The comparison of the Johnson algorithm's trial outcomes is still better, and the results of the Johnson algorithm's latency and throughput with the comparison algorithm are favorable [4]. The Floyd-Warshall method uses an average of 21,78% in CPU

use testing, while the Johnson algorithm uses an average of 23,84%. However, 1.3% of memory is used equally by both algorithms [5].

Previous research findings reveal, The shortest path, 1→3→5→6→ 4→2→7→1, was found by using the Branch and Bound method to the case study of JNE services in Balige District (Traveling Salesman Problem). With six branches, a total minimum travel time of 36 minutes was achieved. An increase of 57,25%, and a 33,2% increase in route spread [6] [7]. furthermoreIn comparison to the results of applying the Dijkstra algorithm, which yielded a total distance of 14 kilometers, and the A * (A-star) algorithm alone, which yielded a total distance of 13.1 kilometers, the Hybrid method was found to be a better and more effective option with a total distance of 12.4 kilometers [8].

In addition, the use of GIS for finding the nearest route also has broad development potential, ranging from the integration of real-time traffic data, interactive visualisation, to route optimisation based on additional criteria such as travel time, cost, and road conditions. This research aims to implement and compare the performance of the Exhaustive Search and Johnson's algorithms in the context of GIS, thereby contributing to the development of a more adaptive and effective transportation information system.

The integration of these two algorithms into GIS is expected to provide a clear performance comparison between accuracy and time efficiency in route searching. Thus, this research not only contributes to the development of GIS-based tourism technology but also provides practical guidance for tourists in planning their trips to the Sipiso-Piso tourist attraction quickly and effectively.

Literature Review

2.1. Geographic Information System

Before we discuss the definition of Geographic Information Systems, we should first understand what is meant by information systems. Information systems are a unity of elements that are dispersed and interact with each other, creating an information flow. The interaction process consists of data processes such as input, processing, integration, processing, computation or calculation, storage, and distribution of data or information.

It is necessary to distinguish between data and information. Data is a fact that exists and is inherent to an object, such as value, size, weight, area, and so on. Meanwhile, information is additional knowledge obtained after processing that data. The value of information greatly depends on the knowledge possessed by the user.

In other words, information is a collection of relevant and related data (according to its level of validity and reliability), which is processed and transformed into a form that is easy to understand, liked, and accessible. Users are free to utilise information as knowledge, a basis for planning, a foundation for decision-making, and even for simple things like entertainment.

According to Paryono (1994:1), a Geographic Information System (GIS) is a computer-based system used to store, manipulate, and analyse geographic information. This technology has rapidly developed in line with advancements in information technology or computer technology. According to Wahyu (2008:116), a Geographic Information System (GIS) is an information system used to input, store, retrieve, analyse, process, and produce geographically referenced data or geospatial data, to support decision-making in land use planning and management, natural resource management, transportation environment, urban facilities, and other public services.

A more comprehensive, effective, and sustainable spatial planning model may be produced by combining these two strategies. Additionally, the paper suggests creating a web-

based GIS platform to integrate technical evaluation accuracy and inclusivity in housing and spatial planning ([9]).

2.2. Teori Graph

A graph is defined as G = (V, E), where V is a non-empty set of vertices = { v1, v2, v3,..., vn} and E is a set of edges (or nodes) that connect pairs of vertices {e1, e2, e3,..., en}, or it can be briefly written as G = (V,E). This means that V cannot be empty, while E can be empty. So, a graph can have no edges at all, but its vertices must exist, at least one. Vertices in a graph can be numbered with letters, natural numbers, or a combination of both. Meanwhile, the edge connecting vertex vi and vertex vj is denoted by the pair (vi, vj) or by the symbols e1, e2, ... In other words, if e is an edge connecting vertex vi with vj, then e can be written as e = (vi, vj).

The vertices in the image above can be any objects such as locations, electronic components, and so on, and the edges can represent any relationships such as highways, network connections, and so on. Vertex in the graph in this text represents cities and edges represent routes or roads connecting public facilities.

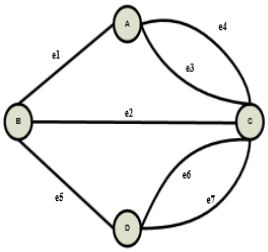


Figure 1. Example of a Graph with Four Vertices and Seven Edges

Caption: G is a graph with:
V = { a, b, c, d }
E = { (a, b), (b, c), (a, c), (a, c), (b, d), (c, d), (c,d) }
= {e1,e2,e3,e4,e5,e6,e7}

A graph is formally defined as an ordered pair G = (V, E) consisting of a non-empty set V as vertices or nodes and a set E as edges or links that connect pairs of nodes in the structure (Rosen, 2019). Classification of graphs based on edge orientation distinguishes a directed graph, which has a direction orientation on each edge each edge with an undirected graph or an undirected graph that does not have a specific orientation specific (Munir, 2022). The concept of graph connectivity explains that a connected graph is a structure where there is a path between every pair of vertices, while a disconnected graph consists of two or more separate components without connecting paths between the components (Akhsani & Jaelani, 2020). Graph representation theory includes the adjacency matrix, which is a square matrix of size n×n where the element in row i and column j is one if there is an edge between vertex i and j, and the incidence matrix, which represents the relationship between vertices and edges in the form of an n×m matrix.

2.3. Algoritma Exhaustive Search

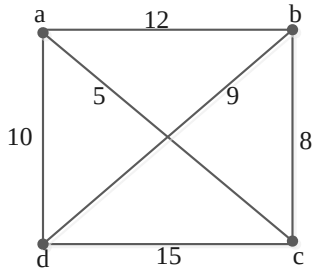
Another term closely related to brute force is exhaustive search. Both brute force and exhaustive search are often considered the same term, but there is a slight difference in the type of problems they solve. Exhaustive search is a brute force solution search technique for problems involving the search for elements with special properties, usually among combinatorial objects such as permutations, combinations, or subsets of a set. Based on this definition, exhaustive search is also brute force.

The exhaustive search algorithm for the Shortest Path problem is:

1. Enumerate (list) all Hamiltonian circuits of a complete graph with n nodes. List all Hamiltonian circuits of a complete graph with n vertices.

2. Calculate (evaluate) the weight of each Hamiltonian circuit found in step 1. Calculate (evaluate) the weight of each Hamiltonian circuit found in step 1.
3. Choose the Hamilton circuit with the smallest weight. Choose the Hamilton circuit with the smallest weight.

Let node a be the city where the journey begins (starting city). Enumerate all Hamiltonian circuits as follows:



No.	Rute Perjalanan	Bobot
1.	a → b → c → d → a	12+8+15+10=45
2.	a → b → d → c → a	12+9+15+5=41
3.	a → c → b → d → a	5+8+9+10=32
4.	a → c → d → b → a	5+15+9+12=41
5.	a → d → b → c → a	10+9+8+5=32
6.	a → d → c → b → a	10+15+8+12=45

The shortest travel route is
a→c→b→d→a
a→d→b→c→a
with a weight of 32.

For 4 cities, there are 6 possible travel routes (or Hamilton circuits). The shortest travel routes are a-c-b-d-a or a-d-b-c-a with a weight of 32. Because the journey starts and ends at the same node, for n nodes, all possible travel routes can be generated by permuting n – 1 nodes. The permutation of n – 1 nodes is (n – 1). In the above example, for n = 6, there will be (4 – 1)! = 3! = 6 travel routes.

Although the exhaustive algorithm theoretically produces a solution, the time or resources required to find the solution are very large. In some literature on algorithmic strategies, an example of a problem often associated with exhaustive search or brute force is the Shortest Path Search problem.

2.4. Johnson's Algorithm

One of the requirements of the Johnson algorithm is that each job to be completed must pass through each machine. Each machine operates according to the production process schedule. The Johnson algorithm is a rule for minimising the makespan of a two-machine serial production process. The Johnson problem is formulated with j jobs processed on 2 machines, where tj,1 is the processing time on machine 1 and $t_{j,2}$ is the processing time on machine 2. The steps for calculation using the Johnson algorithm are:

1. Determine the values ($t_{j,1}$, $t_{j,2}$).
2. If the minimal processing time is on the first machine (e.g., $t_{j,1}$), place the job at the beginning of the scheduling sequence.
3. If the minimal processing time is on the second machine (e.g., $t_{j,2}$), place the job at the end of the scheduling sequence.
4. Remove the jobs from the list and arrange them in the scheduling sequence. The total time $t_{1,1}$ is the processing time of job 1 on machine 1. The total time $t_{1,2}$ is $t_{1,1} + t_{1,2}$. The total time $t_{2,1}$ is $t_{1,1} + t_{2,1}$. The total time $t_{2,2}$ is max { $t_{1,2}$,$t_{2,1}$ } + $t_{2,2}$ and so on. If there are still jobs remaining, repeat step 1; otherwise, if there are no more jobs left, the scheduling is complete.

In Johnson's Algorithm, there are several steps to calculate the shortest path as shown in Figure 1, which involves adding a new node q with a value of 0 to a graph with negative weights. The new

node q has a distance of 0 from the source node because node q is assumed to be the new source node. Then the Bellman-Ford algorithm will calculate the minimum weight of the graph with node q as the starting point. After the minimum weight is calculated, the updated graph weight will be obtained, resulting in positive weights. Then node q is removed, and the Dijkstra algorithm will calculate the shortest route from each node on the updated graph. The Bellman-Ford algorithm is used to reset the graph input to eliminate negative weights on each edge and detect negative cycles in the graph.

Methods

The proposed method is the selection of packet routes during data transmission, using the exhaustive search algorithm and Johnson's algorithm for routing to determine the shortest path during data transmission. To determine the shortest route, the Johnson Algorithm adds a new node to the graph. Next, it reconstructs the weights on all edge nodes using the Bellman-Ford Algorithm as a subroutine. Then, the exhaustive search algorithm is used to find the shortest route from each node to all other nodes.

The stages of designing the Johnson Algorithm can be seen in Figure 2. Based on Figure 2, the mechanism for determining the shortest route using the exhaustive search and Johnson algorithms begins by adding a new node connected to every other node with a weight of 0. Then, the minimum weight (with the new node as the starting point) to every other node is calculated. The next stage is to reconstruct the weight on each nearest node using Equation 1, where w is the weight of a node, u is the starting or source node, and v is the destination node. While h represents the hop.

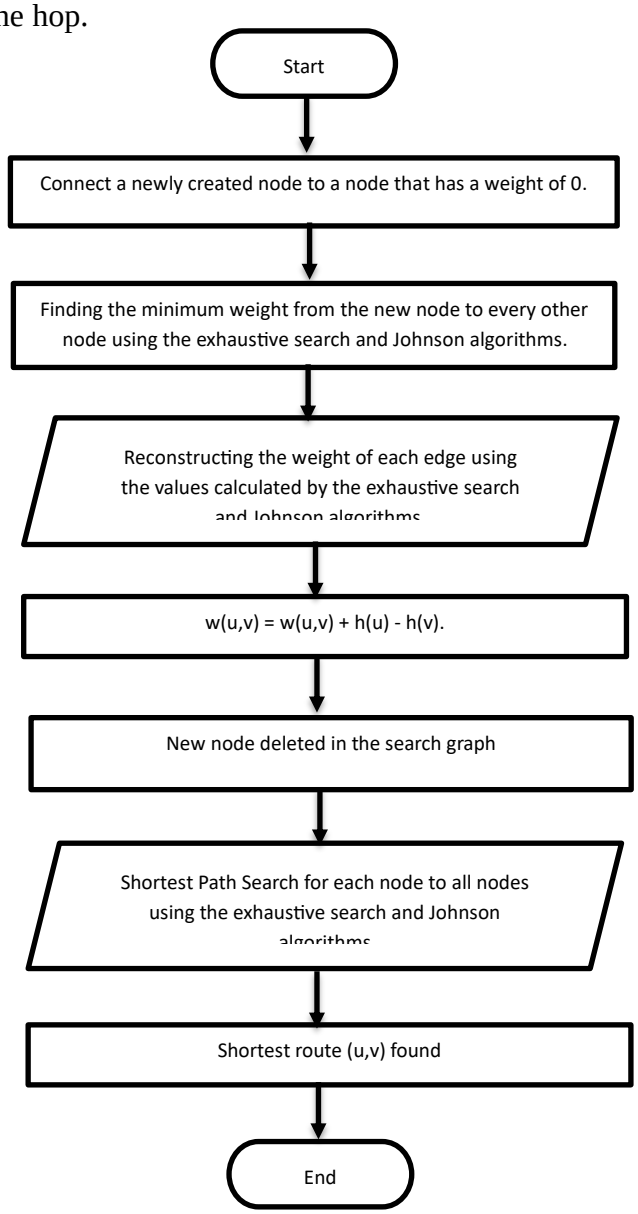


Figure 2. Design of routing mechanisms.

Result and Discussion

The algorithm and method used to find the optimum route between the origin node and the destination node, where the parameter values associated with each arc are known, are called the shortest path algorithm and method. There are four categories of the shortest path problem, namely:

1. The shortest path between two distinct nodes
2. The shortest path between all pairs of nodes.
3. The shortest path from a node to all nodes.
4. The shortest path from one node to another passing through certain specified nodes.

The shortest route that will be discussed is the first category, which is between two different nodes. There are methods and algorithms used for the first category, including the method known as Exhaustive search.

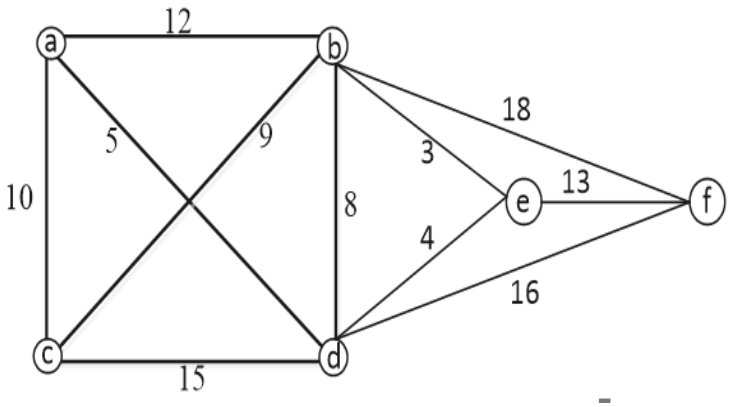
3.1 Analisis Metode Exhaustive search

In the process of calculating the shortest route from the starting point to the destination, several stages can be performed using an algorithm.

1. Finding the shortest route to the Sipiso-piso tourist attraction
   a. Problem: Given 5 roads and the distances between each location, find the shortest tour that visits each destination. Find the shortest tour that passes through each destination.
   b. he shortest route problem is none other than finding the sipiso-piso tourist attractions with the minimum weight.

   Here are the steps to solve example 1 above:
2. Enumerate (list) all sipiso-piso tourist attractions from the complete graph with n nodes.
3. Calculate (evaluate) the weight of each sipiso-piso tourist attraction found in step 1. 3. Choose the path to the sipiso-piso tourist attraction with the smallest weight. The following will illustrate the solution to the problem of finding the shortest route to the Sipiso-piso tourist attraction:

TSP (Travelling Salesperson Problem) with n = 5, starting node = a (Medan) and the destination node = F (Sipiso-piso tourist attraction).



The solution is as follows: 1. For 5 nodes, all possible travel routes can be generated by permuting n – 1 nodes. 2. The permutation of n – 1 nodes is (n – 1)! 3. In the above example, for n = 5, there will be (5 – 1)! = 4! = 24 travel routes.

1. a→b→f = 12 + 18 = 30
2. a→b→e→f = 12+ 3 +13 =28
3. a→b→d→f =12 + 8 + 16 =36
4. a→b→d→e→f = 12 + 8 + 4 +13 = 37
5. a→b→c→d→f =12 + 9 +15 + 16 = 52
6. a→b→d→e→f = 12 + 8 + 4 + 13 = 37
7. a→b→e→d→f = 12 + 3 + 4 + 16 = 35
8. a→b→c→d→e→f = 12 + 9 + 15 + 4 + 13 =53
9. a→c→d→f = 10 + 15 + 16 = 41
10. a→c→b→f = 10 + 9 + 18 = 37
11. a→c→d→e→f = 10 + 15 + 4 + 13 = 42
12. a→c→b→e→f = 10 + 9 + 3 + 13 = 34
13. a→c→b→e→f = 10 + 9 + 3 + 13 = 34
14. a→c→b→d→f = 10 + 9 +8 + 16 = 43
15. a→c→b→e→d→f = 10 + 9 + 3 + 4 + 16 = 42
16. a→c→b→d→e→f =10 + 9 + 8 + 4 + 13 = 44
17. a→d→f = 5 + 16 = 21
18. a→d→e→f = 5 + 4 + 13 = 22
19. a→d→b→f = 5 + 8 + 18 = 31
20. a→d→b→e→f = 5 + 8 + 3 + 13 = 29
21. a→d→c→b→f = 5 + 15 + 9 + 18 = 47
22. a→d→e→b→f = 5 + 4 +3 + 18 = 30
23. a→d→c→b→e→f = 5 + 15 + 9 + 3 + 13 = 45
24. a→d→b→e→f = 5 + 8 + 3 + 13 = 29

and the shortest path from node = a (medan) to node = f (sipiso piso) is a→d→f with a weight of 21.

If solved using the exhaustive search method, we must enumerate (n – 1)! sipiso-piso tourist attractions, calculate the weight of each, and select the sipiso-piso tourist attraction with the smallest weight.

The time complexity of the exhaustive search algorithm for the shortest route problem is proportional to (n – 1)! multiplied by the time to calculate the weight of each sipiso-piso tourist attraction.

Calculating the weight of each Sipiso-piso tourist attraction takes O(n) time, so the time complexity of the exhaustive search algorithm for the shortest route problem is O (n · n!).

3.2 Analisis Algoritma Johnson

The Johnson algorithm consists of several steps Add a new node q and connect it to all original nodes with a weight of 0. Calculate the distance h(v) from q to all nodes using Bellman-Ford to handle negative weights. Change the edge weights:

$$w'(u,v) = w(u,v) + h(u) - h(v)$$

so that all weights are non-negative. Calculate the shortest distance from each node using Dijkstra with weight w'. Distance correction:

$$d(u,v) = d'(u,v) - h(u) + h(v)$$

so that the actual distance is obtained.
Example from A → all nodes:
Starting from A: distance(A) = 0
Neighbour nodes:

A→B = 5 → distance(B) = 5
A→C = 2 → distance(C) = 2
Choose the node with the minimum distance from unvisited → C (distance=2)
Neighbour nodes of C:
C→D = 7 → distance(D) = 2+7=9
C→E = 10 → distance(E) = 2+10=12
Next node: B (distance=5)
B→C = 1 → distance(C) = min (2,5+1) =2
B→D = 3 → distance(D) = min (9,5+3=8) → distance (D)=8
Node D (distance=8
D→E=2 → distance(E)=min (12,8+2=10) → distance(E)=10 Node E finished

Table 1. The result of the shortest distance from A

Goal	Distance (km)	Rute
B	5	A→B
C	2	A→C
D	8	A→B→D
E	10	A→B→D→E

Conclusion

Geographic Information System (GIS) has proven effective in modelling the road network to the Sipiso-Piso tourist attraction, providing interactive map visualisations that make it easier for tourists to choose the nearest route. GIS is capable of integrating geographic data (coordinates, road distances) and displaying route calculation results accurately. Exhaustive Search works by evaluating all possible routes from the starting point to the destination. The result is accurate but less efficient for large graphs because the computational complexity increases exponentially as the number of nodes increases, whereas Johnson's Algorithm can compute the shortest paths between all pairs of points with higher efficiency, combining Bellman-Ford and Dijkstra. This algorithm is more suitable for complex and dense road networks, with varying road weights. The implementation of both algorithms shows that the optimal route can be clearly determined: Johnson provides a faster and more stable solution, while Exhaustive Search is valid for small graphs or as a comparison method. The distance and route data generated can be used for tourist guidance and travel optimisation. The final results of this research affirm that the combination of GIS with route search algorithms (specifically Johnson) can serve as a decision support tool for both tourists and tourism managers. This system allows for distance estimation, identification of alternative routes, and more efficient trip planning.

References

[1] [1] A. A. Rahma and S. Pariwisata, “Jurnal Nasional Pariwisata,” vol. 12, no. April, pp. 1–8, 2020.
[2] [2] A. Priyantoro, K. Mustofa, and U. G. Mada, “Pengembangan Aplikasi Pencarian Rute Terbaik Pada Sistem Operasi Android (Studi Kasus Rute Trans-Jogja),” pp. 72–88.

[3] [3] R. Hudiono, “Wisatawan Muda dan Destinasi Wisata : Sebuah Kajian Kuantitatif,” vol. 5, no. 3, pp. 688–696, 2024.
[4] [4] A. P. Segara, R. M. Ijtihadie, A. P. Segara, R. M. Ijtihadie, and T. Ahmad, “Implementasi algoritma shortest path johnson untuk mekanisme route discovery pada software defined network,” vol. 19, pp. 28–36, 2021.
[5] [5] M. I. Firdaus, R. Primananda, M. Hannats, and H. Ichsan, “Analisis Perbandingan Performansi Algoritme Floyd-Warshall dan Algoritme Johnson untuk Penentuan Rute Terpendek pada Software Defined Network,” vol. 2, no. 9, pp. 2469–2475, 2018.
[6] [6] O. Rute, D. Kurir, M. Metode, and T. Salesman, “G-Tech : Jurnal Teknologi Terapan,” vol. 6, no. 2, pp. 159–165, 2022.
[7] [7] S. M. Algoritma, M. H. Setiawan, M. Imrona, and D. T. Murdiansyah, “Optimasi Rute Angkutan Kota Secara Exhaustive Search,” vol. 2, pp. 47–54, 2017, doi: 10.21108/indojc.2017.22.178.
[8] [8] W. Bismi, W. Gata, and T. Asra, “Penerapan Algoritma Hybrid Dalam Menentukan Rute Terpendek Antara Cabang Kampus,” vol. 13, no. 1, pp. 1–9, 2021.
[9] [9] I. Media, I. Media, and D. A. N. Perumahan, “GOVERNANCE : Jurnal Ilmiah Kajian Politik Lokal dan Pembangunan,” vol. 11, pp. 8–15, 2025.