© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

A Selection of Bandung City Travel Route Using The FLOYD-WARSHALL Algorithm

Muhammad Ghifari Arfananda[1*], Surya Michrandi Nasution[1], Casi Setianingsih[1]

[1]Telkom University,
School of Electrical Engineering, Bandung, 40257, INDONESIA

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2020.12.07.010
Received 29 March 2020; Accepted 9 August 2020; Available online 30 August 2020

Abstract: The rapid development of information and technology, the city of Bandung tourism has also increased. However, tourists who visit the city of Bandung have problems with a limited time when visiting Bandung tourist attractions. Traffic congestion, distance, and the number of tourist destinations are the problems for tourists travel. The optimal route selection is the solution for those problems. Congestion and distance data are processed using the Simple Additive Weighting (SAW) method. Route selection uses the Floyd-Warshall Algorithm. In this study, the selection of the best route gets the smallest weight with a value of 5.127 from the Algorithm process. Based on testing, from two to five tourist attractions get an average calculation time of 3 to 5 seconds. This application is expected to provide optimal solutions for tourists in the selection of tourist travel routes.

Keywords: Shortest path, bandung travel, distance, traffic jams, floyd-warshall algorithm, simple additive weighting method.

1. Introduction

Travel is an important activity for the community with the development of technology. Bandung is a city that has many tourist attractions visited by domestic and foreign tourists. The number of tourists in the city of Bandung in 2016 reached 5,000,625 inhabitants [1]. With these data, the increasing number of populations causes the number of residents in the city of Bandung which causes higher traffic congestion. That condition is a problem for tourists enjoying the city because they must deal with the limited number of places with high traffic, limited distance, and time. This is an aspect that needs to be considered by the city authorities. Applications such as Google Maps, Waze, and others, still have shortcomings in terms of determining the lack of time with several destinations to be visited. The route selection becomes an important aspect of a tour.

The best route selection requires actual and accurate information. This will try to add functions with new implementation in route optimization. This application is a calculation of the time with a limited amount, the number of tourist attractions to be visited, and actual traffic conditions. With this information, traffic conditions can be estimated at that moment [2]. Tourists can choose the best route to the interesting place they want to visit. The route to be followed is the best route. The track will be traversed in only one track by passing through several destinations with the fastest time. In this case, the destination to be visited is not passed in the same destination one time [3].

In this study, the author uses the Floyd-Warshall Algorithm for route search. The Algorithm is a method that solves problems by looking at the solutions to be obtained and making the best decision. The advantage is the solution previously found compared with subsequent solutions that aim to find the most appropriate solution [4]

*Corresponding author: ghifariarf@student.telkomuniversity.ac.id

2. Literature Study

2.1 Graph

Graph is part of objects called vertices or points V = {$v1$, $v2$, ...} (vertices) connected by sides E = {$E1$, $E2$, ...} (edge). The graph is usually for each node and side with a limited number [5]. A side can connect a point with the same point. This side is called a loop (5).

In a graph, structure can be developed by giving weight or value to the path. This value is used to symbolize many different concepts, such as vertices, edges, distances between vertices, and so on. These values are added up to find out the effect on the path. The shortest path is the minimum value of the sum [3].

2.2 Paths and Circuit Euler's

Euler's Path is a graph with the trajectory of each side traversed just once. The path that if return to the starting point is a closed path called the Euler circuit [6]. With this theory, it can be said that the Euler circuit is the Euler path but, the Euler path is not necessarily the Euler circuit, as follows:

1. A graph which has an Eulerian path is called a semi- Eulerian graph.
2. Graphs that have Euler circuits are called Eulerian graphs

2.3 Traveling Salesman Problem (TSP)

Traveling Salesman Problem (TSP) is a problem in achieving multiple goals with exactly one of several goals to find the shortest path [7]. By seeing all the tracks must be traversed to get the smallest value and calculate with another point. Problems in finding the minimum distance in a closed-loop against various parameters without having to go to the same place more than once. TSP is usually used in the implementation of planning and shipping goods [7].

2.4 Simple Additive Weighting (SAW)

The Simple Additive Weighting (SAW) method is known as the weighted sum method. The basic concept of the SAW method is to find a weighted sum of the performance ratings for each alternative on all attributes (Fishburn, 1967) (MacCrimmon, 1968). The SAW method requires the decision matrix normalization process (X) to a scale that can be compared with all available alternatives. The SAW method requires the decision-maker to determine the weight for each attribute. The total score for the alternative is obtained by adding up all of the multiplication results between and the weight of each attribute [8].

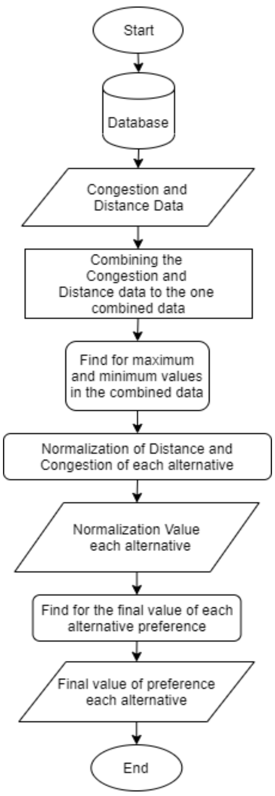

Fig. 1 - Simple Additive Weighting Process

Based on Fig. 1, the SAW method process in adding the weight. This method acts as calculate various types of data, in this research such as traffic and distance. The following is a complete explanation of the diagram above:

1. Retrieve data on the state of traffic and distance taken from the database.
2. The weight of the two data is entered into each column. The first column is the traffic criteria and the second one is the distance criteria.
3. Find the recommended maximum and minimum values in each column of the criteria. Determine the preference weights for each criterion in the final value.
4. Normalization with preference weights each criterion to get the weight for the next process.
5. The result of each normalization criterion will be an alternative to the merging process. This process will calculate the value obtained in the previous process with preference weights for each alternative.
6. The results from the merger get weights on the values that will be used in the Floyd-Warshall Algorithm.

2.5 Floyd-Warshall Algorithm

The Floyd-Warshall algorithm is a algorithm dynamic program about the problem of finding the shortest route [9]. By having graph input directed and weighted (V, E), in the form of a set of points (V) and sides (E). amount the value of the shortest path of each side on a path is the value of the path. This algorithm calculates the smallest value of all paths connects a pair of dots and does it all at once point pair [9].

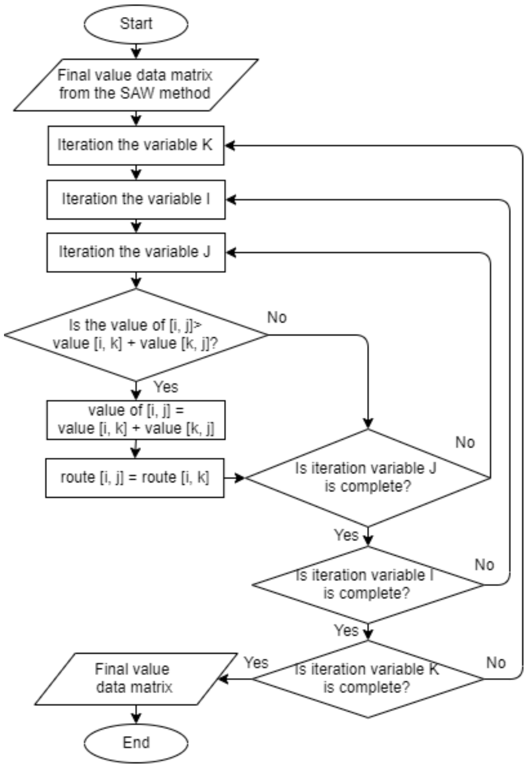

Fig. 2 - Floyd-Warshall Algorithm Process

Based on Figure. 2. explained how the algorithm works in this selection route. The design of the Floyd-Warshall algorithm is as follows:

1. Data that has been processed are analyzed for processing to the Floyd-Warshall Algorithm.
2. Give a zero value for the same point or place and not until the point or place is empty in the matrix.
3. Make a route matrix according to the value of each point or place.
4. Process the matrix according to the Floyd-Warshall Algorithm according to iteration K, I, J.
5. After the iteration is finished, the final result of the route matrix is the best route to get to the destination

3. Design and Testing

3.1 System Overview

This research will choose the route search menu or provide information about the traffic conditions in the city of Bandung. This system helps for choosing tourist attractions based on the opening and closing hours of tourist attractions. Tourists can choose one or all three tourist attractions. The map will show the route that will be taken to the final destination with a path. This route is based on using Floyd-Warshall Algorithm.

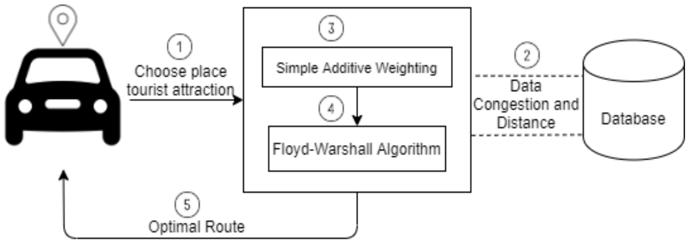

Fig. 3 - System Overview

The general description of this system explains the stages in the general flow of the system. The design of the system overview is as follows:

1. Tourists choose tourist attractions for travelers.
2. The system will retrieve traffic data and distance data to the database according to the tourist attractions they choose.
3. The results of place, congestion, and distance data will be processed using the Simple Additive Weighting Method (SAW) which then results from the SAW Method generate the final weight value of a path.
4. After getting the final value, the weight of a path then initialization to new processes using Floyd-Warshall Algorithm.
5. The output from Floyd-Warshall will display the order of routes to the destination. The route will be displayed to users.

3.2 System Implementation

The implementation of this system can be seen on Fig. 4 is the result of choosing 3 tourist attractions. The starting point is Telkom University with a GPS icon and destination point in the form of a red pin at the Museum of the Asia Africa Museum, Sri Baduga Museum, and Siliwangi Museum. The order of travel routes is the Telkom University to Sri Baduga Museum then Asia Africa Museum after that Siliwangi Museum and the last destination is Telkom University because it is a TSP.

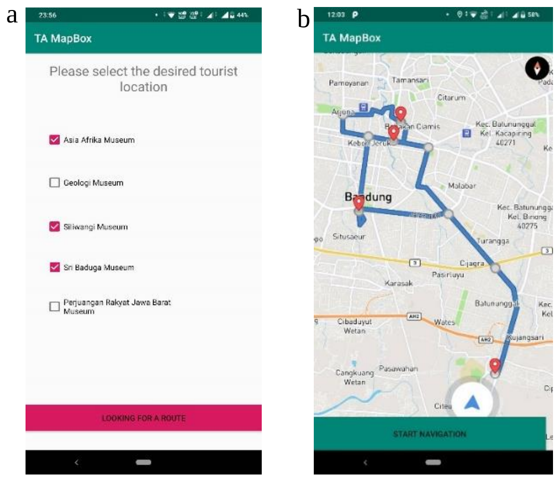

Fig. 4 – (a) Main Interface; (b) Display Route Interface

3.3 Simple Additive Weighting Processing

In this sample what was done on the application is from Telkom University to three tourist attractions. In this test takes sample Museum Sri Baduga, Museum Asia Afrika, and Museum Siliwangi. In this SAW method process is using random weights in congestion weights. The following is an initialization node and edge before calculate.

Table 1 - Initialization Node

Node	Node Name
A	Telkom University
B	Sri Baduga Museum
C	Asia Afrika Museum
D	Siliwangi Museum
1	Soekarno-Hatta Street
2	Soekarno-Hatta2 Street
3	BKR Street
4	Inhoftank Street
5	Inhoftank2 Street
6	Dalem Street
7	Otto Iskandar Dinata Street
8	Sumbawa Street

Based on Table 1, The Node with the alphabet is a place that will be visited by tourists. While the Node with numbers is the path selection node to be chosen by the Floyd-Warshall Algorithm. In this case, it is expected that the results to be obtained are route selection and TSP implementation.

Table 2 - Initialization Edge With Weights

Edge	Distance	Congestion	SAW Weight
A → 1	0.315	0.600	0.915
1 → 3	0.189	0.600	0.789
1 → 4	0.366	0.000	0.366
2 → A	0.317	0.600	0.917
3 → C	0.316	0.3	0.616
3 → B	0.244	0.6	0.844
3 → 2	0.187	0.6	0.787
3 → 8	0.346	0.0	0.346
B → 3	0.350	0.0	0.350
B → 6	0.233	0.6	0.833
B → 5	0.123	0.3	0.423
4 → B	0.124	0.3	0.424

Edge	Distance	Congestion	SAW Weight
C → 3	0.700	0.0	0.700
C → 7	0.067	0.0	0.067
C → D	0.253	0.3	0.553
5 → 2	0.375	0.3	0.675
6 → C	0.057	0.6	0.657
6 → B	0.114	0.0	0.114
7 → B	0.201	0.3	0.501
8 → D	0.120	0.0	0.120
D → C	0.073	0.3	0.373
D → 3	0.389	0.0	0.389
D → 6	0.257	0.0	0.257

Based on Table 2, The result of the SAW method contains final weight value as the combination between Distance value and Congestion value. The final weight value is the sum of the Distance and Congestion values. The value of Distance is obtained based on the results of the division of each edge distance divided by the maximum distance from the entire edge distance. The value of Congestion is obtained based on the results of the division of each edge Congestion divided by the maximum Congestion from the entire edge Congestion. In this case, the Congestion value is initialized with a number starting from 0 to 1 where 0 traffic is smooth, 1 traffic is jammed and 0.5 traffic is moderate.

3.4 Floyd-Warshall Algorithm Processing

Final Weight from Table 2, then implemented into the Floyd-Warshall Algorithm. The weight of the Floyd-Warshall Algorithm is not the distance weight, but the Final Weight as the end result of the SAW method. The following is the weight of the Floyd-Warshall Algorithm before calculation.

Table 3 - Weight Floyd-Warshall Algorithm

Edge	Weight
A → 1	0.915
1 → 3	0.789
1 → 4	0.366
2 → A	0.917
3 → C	0.616
3 → B	0.844
3 → 2	0.787
3 → 8	0.346
B → 3	0.350
B → 6	0.833
B → 5	0.423
4 → B	0.424

Edge	Weight
C → 3	0.700
C → 7	0.067
C → D	0.553
5 → 2	0.675
6 → C	0.657
6 → B	0.114
7 → B	0.501
8 → D	0.120
D → C	0.373
D → 3	0.389
D → 6	0.257

Based on Table 3, each edge gets weighted values and connectedness between nodes. This value will be calculated by the Floyd-Warshall Algorithm. The final result will get the optimal route.

Table 4 - Final Route Results

No	Final Route Results	Final Weight
1	A > 1 > 3 > C > D > 6 > B > 5 > 2 > A	5.259
2	A > 1 > 3 > C > 7 > B > 3 > 8 > D > 3 > 2 > A	5.797
3	A > 1 > 3 > 8 > D > C > 7 > B > 5 > 2 > A	5.127
4	A > 1 > 3 > 8 > D > 6 > B > 3 > C > 3 > 2 > A	5.910
5	A > 1 > 4 > B > 3 > 8 > D > C > 3 > 2 > A	5.298
6	A > 1 > 4 > B > 3 > C > D > 3 > 2 > A	5.317

Based on Table 4, the result of calculation Floyd-Warshall with TSP solution there is Final Weight value. The smallest Final Weight value is the solution in selecting the route to be chosen with the Final Weight of 5.371. Then solution number 3 becomes the route to be chosen, to get the optimal route. This optimal route will be displayed to users.

3.5 Testing

The author tries to make a test by using different number of tourist attractions. Then get the computational time value according to the number of places. The number of tourist attractions starts from 2, 3, 4, and 5 places with the same congestion conditions. The number of tourist attractions show randomly up to 5 times.

1. Initialization the test node

Table 4 – Initialization of The Test Node

Node	Node Name
A	Telkom University
B	Sri Baduga Museum
C	Asia Afrika Museum
D	Siliwangi Museum
E	Perjuangan Rakyat Jawa Barat Monument
F	Geologi Museum

2. Condition of 2 tourist attractions

Table 4 – Computational Time Value of 2 Tourist Attractions

No	Tourist Attractions	Computational Time
1	A > C > B > A	4.071
2	A > B > C > A	3.897
3	A > E > F > A	3.656
4	A > C > D > A	3.881
5	A > F > E > A	3.990
	Average Time	3.899

Based on Table 4, the process with the condition of 2 tourist attractions in the table consists of 5 times testing. Each test has a different computational time by the order of the tourist attractions themselves. Number 3 has very small computational time with value 3.656 seconds value. The selection of places is taken randomly then in 5 times this test takes an average of computing time at 2 tourist attractions. With 5 times the average computing time taken at 2 tourist attractions. Testing these 2 places the average computation time is 3.899 seconds.

3. Condition of 3 tourist attractions

Table 5 - Computational Time Value of 3 Tourist Attractions

No	Tourist Attractions	Computational Time
1	A > C > D > B > A	4.399
2	A > C > B > D > A	4.350
3	A > B > C > D > A	4.441
4	A > B > D > C > A	4.377
5	A > D > C > B > A	4.491
	Average Time	4.412

Based on Table 5, the process with the condition of 3 tourist attractions in the table consists of 5 times testing. Each test has a different computational time by the order of the tourist attractions themselves. Number 2 has very small computational time with 4.350 seconds value. The selection of places is taken randomly then in 5 times this test is takes an average of computing time at 3 tourist attractions. With 5 times the average computing time taken at 3 tourist attractions. Testing these 3 places the average computation time is 4.412 seconds.

4. Condition of 4 tourist attractions

Table 6 - Computational Time Value of 4 Tourist Attractions

No	Tourist Attractions	Computational Time
1	A > B > C > F > D > A	4.512
2	A > B > F > D > C > A	4.719
3	A > B > F > C > D > A	4.554
4	A > B > D > C > F > A	4.620
5	A > B > C > D > F > A	4.588
	Average Time	4.599

Based on Table 6, the process with the condition of 4 tourist attractions in the table consists of 5 times testing. Each test has a different computational time by the order of the tourist attractions themselves. Number 1 has very small computational time with 4.512 seconds value. The selection of places is taken randomly then in 5 times this test takes an average of computing time at 4 tourist attractions. With 5 times the average computing time taken at 4 tourist attractions. Testing these 4 places the average computation time is 4.599 seconds.

5. Condition of 5 tourist attractions

Table 7 - Computational Time Value of 5 Tourist Attractions

No	Tourist Attractions	Computational Time (s)
1	A > B > D > C > F > E > A	4.773
2	A > D > B > C > F > E > A	4.906
3	A > E > F > D > C > B > A	4.835
4	A > B > C > D > F > E > A	4.912
5	A > F > E > D > C > B > A	4.880
	Average Time	4.861

Based on Table 7, the process with the condition of 5 tourist attractions in the table consists of 5 times testing. Each test has a different computational time with the order of the tourist attractions themselves. Number 1 has very small computational time with 4.773 seconds value. The selection of places is taken randomly then in 5 times this test takes an average of computing time at 5 tourist attractions. With 5 times the average computing time taken at 5 tourist attractions. Testing these 5 places the average computation time is 4.861 seconds.

3.6 Result

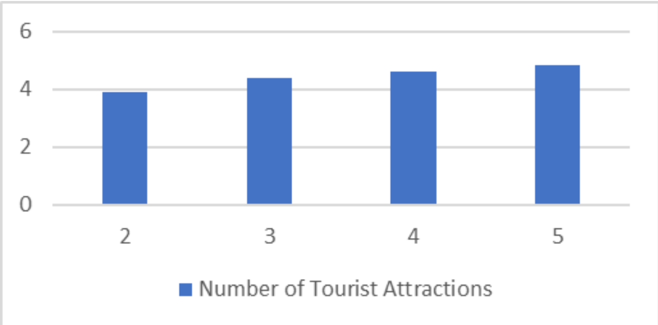

Fig. 5 - Test Result

In Fig. 5 describes that the X-axis is for the number of attractions and the vertical axis is for computational time. The results in this test appears that the graph always increases with the addition of nodes. This testing has the lowest computing time with 3.899 seconds value on 2 number of attractions. While the highest computing time is 4.861 seconds at 5 number of attractions. Therefore, the more tourist attractions there will be the longer computing time to take.

4. Conclusion

Based on the results of the Floyd-Warshall Algorithm test that the algorithm can determine the best route to reach several tourist attractions in the city of Bandung. By using the Floyd-Warshall Algorithm, to reach several tourist attractions on one track can be applied with a weight of 5.127. Testing uses computational time comparisons at each change in tourist attractions as a node. The change in computational time is getting higher with the number of nodes. Based on the results of the Floyd-Warshall Algorithm, test on the number of tourist attractions that count the number of tourist attractions from 2 to 5 produces an average time calculation of 3 to 5 seconds. The more tourist attractions there is the longer computing time to take. The best route in order to provide solutions for tourists to choose the best routes. Therefore, weights and nodes affect the results of the Floyd-Warshall algorithm. This research is successful in displaying the TSP route in Android.

Acknowledgment

This research was partly supported by Telkom University and the Computer Engineering major which provided opportunities, platforms, and places that greatly assisted research. The author also thanks the thesis supervisor who always accompanies and provides assistance and comments that greatly improve the manuscript.

References

[1] Badan Pusat Statistik Kota Bandung. (2018, April 20). Jumlah Wisatawan Mancanegara dan Domestik di Kota Bandung, 2016. Retrieved: 5 Februari 2019, Dinas Kebudayaan dan Pariwisata Kota Bandung, https://bandungkota.bps.go.id/statictable/2017/08/29/120/jumlah-wisatawan-mancanegara-dan-domestik-di-kota-bandung-2016.html

[2] Aziz, Abdul, Muhammad Maarif Farid, Esti Suryani. (2017). Floyd Warshall. Algorithm with FIS Sugeno for Search Evacuation Route Optimization. International Seminar on Application for Technology of Information and Communication (iSemantic)

[3] Escribano, Arturo Gonzalez, Hector Ortega-Arranz, and Diego R. Llanos. (2014). Shortest-Path Problem: Analysis and Comparison of Methods

[4] Meng, Xianghu, Jun Li dan Xianzhong Dai. (2017). A Dynamic Colored Traveling Salesman Problem and Its Solution. 36th Chinese Control Conference (CCC)

[5] Rodríguez-Puente, R., & Lazo-Cortés, M. S. (2013). Algorithm for shortest path search in Geographic Information Systems by using reduced graphs. SpringerPlus, 2(1), 291

[6] Fleischner, H. (2001). Some of the many uses of Eulerian graphs in graph theory (plus some applications). Discrete Mathematics, Elsevier Science B.V

[7] Xuan, Wang, and Yuanxiang Li. (2005). Solving Traveling Salesman Problem by Using A Local Evolutionary Algorithm. Vol 1, IEEE International Conference on Granular Computing

[8] Damping, Ria A. T, Addy S, Dyna M. K. (2017). Penerapan Metode Simple Additive Weight (SAW) Pada Aplikasi Penentuan Pilot Pada Rute Penerbangan. Jurnal Informatika Mulawarman Vol. 12, No. 2 November 2017

[9] Singh, Akanksha, and Pramod K. M. (2014). Performance Analysis of Floyd Warshall Algorithm vs Rectangular Algorithm. International Journal of Computer Applications, Vol. 107

[10] Risald, Antonio E. Mirino, Suyoto. (2017). Best Routes Selection Using Dijkstra And Floyd-Warshall Algorithm. International Conference on Information & amp; Communication Technology and System (ICTS)