



{ MUDIMA }



Floyd-Warshall Algorithm for Web-Based Route Optimization for a Road Network Study in Medan City

Faridawaty Marpaung^{1*}, Arnita², Tommy³, Putri Harliana⁴

^{1,2,4}Universitas Negeri Medan

³Universitas Harapan Medan

Corresponding Author: Faridawaty faridawaty@unimed.ac.id

ARTICLE INFO

Keywords: Shortest Path, Network Model, Floyd-Warshall Algorithm, Optimization

Received : 2 October

Revised : 17 October

Accepted : 18 November

©2023 Marpaung, Tommy, Arnita, Harliana: This is an open-access article distributed under the terms of the [Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



ABSTRACT

The socio-demographic conditions of the region have an influence on the transportation performance in the region. This study aims to determine the shortest path between two specified points, namely the starting and the end point of several sub-districts in Medan city, especially the sub-district of Medan Amplas. To get the output, several steps are needed, namely collecting the coordinates of street names and road lengths, which is obtained from Google maps. Graphs are used to present the data. Data analysis is used by Floyd-Warshall Algorithm with HTML Web Application supported by PHP programming language, JavaScript and MySQL database. The test results using the Black Box method show that the system functionality is 100% running well. For testing, the results of the final W_{ij} and Z^* final matrix calculations generated by the system is same with the manual calculations. Finally, to test the reliability of the system made, the results of the reliability test show the truth of working well and it can be accepted. There are no path deviations, so that the results are as expected

INTRODUCTION

In this modern era, the whole world uses the road as one of the most useful media and it is often used for transportation and travel. Manipulation of the shortest path between various locations seems to be a major problem in the road network. Utilizing the shortest transportation route to arrive at a destination is the need of every individual and the most important thing is to be able to arrive in a short time. Various applications and products are introduced to solve or overcome difficulties by developing shortest path algorithms for road networks (Alam & Faruq, 2019). Indonesia is a vast country with a number of large cities that have a modern highway network that continues to develop. The city of Medan is a large city with rapid growth, so there may be new road construction or expansion projects taking place after 2022. For more detailed and up-to-date information about modern highways in Medan, you should contact your local government or other sources the latest news. Development of Indonesia's road network continues, with many toll road expansion and construction projects underway (Soemardi & Pribadi, 2021). The existence of this toll road helps improve connectivity between cities and facilitates the mobility of people and goods. This information is based on my knowledge as of 2022, and it is possible that there have been developments and changes since then.

The main function of roads is as transportation infrastructure that allows the movement of people and goods from one place to another (Surya et al., 2020). Highways are the backbone of land transportation systems throughout almost the world. Highways provide access for individuals to move from home to work, schools, shopping centers, and various personal destinations (Situmorang, Trilaksono, & Japutra, 2019) This provides individual mobility which is important in everyday life. Highways provide access for individuals to move from home to work, schools, shopping centers, and various personal destinations. It provides individual mobility which is important in everyday life.

Roads are the main routes for the transport of goods and supplies (Malisan, Marpaung, Hutapea, Puriningsih, & Arianto, 2023). Roads facilitate trade, distribution of goods, and efficient supply chains. Highways provide access to public services such as hospitals, schools, and health facilities. They are also important in case of emergency evacuation. Then it also supports economic growth by connecting production areas with markets and customers (Li, 2020). They allow access to rural and urban areas, which play an important role in economic activity. Highways help facilitate tourism by connecting tourist attractions with city centers and accommodation. This helps regional economies that depend on tourism. Highways provide access to educational institutions such as schools, colleges, and universities, which are important in developing human capacity and educational progress. Another impact is reducing congestion and greenhouse gas emissions through improving sustainable mobility, such as public transportation, bicycles and pedestrians (Pojani & Stead, 2015).

Roads help in the dissemination of information and communications by facilitating the installation of telecommunication networks and cables. Highway construction can trigger growth and development of certain areas. They can open up remote areas for investment and infrastructure development. Being a gathering place for road users, highways have an important role in promoting traffic safety and reducing the risk of accidents (Schepers, Twisk, Fishman, Fyhri, & Jensen, 2017). Highways play a role in connecting country to country, making them an important element in global connectivity and international trade. In the modern era, highways also increasingly involve technology, such as traffic management systems, GPS navigation, and smart sensors to increase efficiency and safety for road users (Putra & Warnars, 2018). In addition, the emergence of electric cars and autonomous vehicles will also bring significant changes in the function of roads in the future.

The network is a system of points with a distance between the two points. A common problem with networks involves finding the shortest path

between one point in a network and another. The search for the shortest path is a problem that has been discussed and studied the most since the late 1950s. The shortest path problem is generally explained using the concept of a graph, which can be either a directed graph or an undirected graph (Bernstein, 2013). The edges of an undirected graph can be considered to allow trips in both directions. Conversely, an edge in a directed graph can only be used for one-way travel. Usually in determining the shortest path using a weighted graph (Sadavare & Kulkarni, 2012).

In graph theory, the shortest path problem is to find a path between two vertices in weighted directed graph in order to minimize the weights of the constituent edges (Sadavare & Kulkarni, 2012). In this study the data used is data from the two-way road network observation which is divided based on several sub-districts in Medan city, where observations are carried out only for Medan city sub-district by using a google maps with the aim of knowing the roads in the Medan district and the length of a road traversed. Medan city is one of the big cities in Indonesia with a very dense population. In addition to many immigrants, especially many students who migrate. The people of Medan city often face problems in using public transportation.

The graph consists of three key point, namely vertices, edge, and weights. In this study, the vertices of graph are represented as street names in Medan Subdistrict, the edges represent roads which connect the street names in Medan Subdistrict and the edge weights represent the length of the road network.

The difficulty in determining the shortest path arises because there are many existing roads from one area to another, making it possible to choose an alternative path if there is an obstacle on the main shortest road. The need to find the shortest path and the fastest travel time is also taken into account to avoid losses. In finding the shortest path, the more points and lines on the graph, the more complicated it will be (Mardlootillah, Suyitno, & Arini, 2014). The transportation network in big cities like Medan in general has a complicated network. People will know the road that must be passed to get to the

destination they usually visit, but if the destination has never been visited on average, they are often difficult to determine the path that must be passed to reach it. So far, people will ask other people who know very well about the transportation network in the city. So that the trajectory they take will likely be longer and require more time and effort.

The preceding study about the selection of the best route using the Dijkstra algorithm for transportation network analysis have been studied by Ojekudo & Akpan (2017). This study addresses Dominion Paints Nig. Ltd's problem of transporting their products from their manufacturing plant to sales outlets by presenting an analysis of the shortest path using Dijkstra's Algorithm, and it was concluded that the best paths found from the analysis will save the company less distance in transporting the paints while also minimizing time and cost of fueling their vehicles (Ojekudo & Akpan, 2017). Another algorithm to determine the best route is Floyd-Warshall algorithm. To detect the occurrence of negative cycles, the Floyd-Warshall algorithm might be utilized. Some existing Floyd-Warshall algorithm implementations will fail because exponentially huge numbers may arise during execution (Hougardy, 2010). Conducted research titled Use of the Waterfall Method for the Development of a Rural Monitoring and Evaluation System. The rural monitoring and evaluation system is developed using modeling diagrams, specifically UML, and testing is performed after the system design step is done (Trisianto, 2018).

For each and every pair of pertinent nodes, the Floyd-Warshall method is more effective than a single-pair shortest path approach (Aziz, Farid, & Suryani, 2017). The Floyd-Warshall algorithm, also called the plum point method, is a dynamic programming strategy used to identify the shortest route between nodes in a given weighted graph. Floyd-Warshall in an effort to make computations simpler. Finding the shortest path can be made simpler by using it (Dela Cruz et al., 2016; Nova et al., 2023). For every side of every node, the Floyd-Warshall algorithm will compare all potential path trajectories. The distance is determined by adding the

weighted total of the shortest pathways between each pair of vertices during one algorithm iteration (Aziz et al., 2017).

This study aims to determine the shortest path between two specified points, namely the starting and the end point of several sub-districts in Medan city, especially the sub-district of Medan Amplas. The search for the shortest route is carried out using a web-based application based on the Floyd-Warshall algorithm. Three types of testing are carried out to measure system reliability, namely the Black Box, testing the calculation results, and the reliability testing of the system created.

This article consists of four section. Section 1 gives the introduction of the problem carried in this study. Section 2 explain about the dataset, the Waterfall algorithm, and the Floyd-Warshall algorithm. Section 3 gives the results and discussion of this research. And, Section 4 conclude all the results of this work.

METHODS

Various methods can be used to present the ideal path, in this case determining the shortest route. The appropriate strategy can assist in generating

information about the shortest path. An algorithm, such as the Floyd-Warshall algorithm (Khamami & Saputra, 2019), can be used to determine the shortest route. Floyd-Warshall is an algorithm that can calculate the optimal weight for connecting point pairs and do it for all point pairs at once. In other words, while determining the ideal route to be taken, first calculate all possible routes to be taken, and then compare each pair of paths (Aziz et al., 2017). At the stage of developing the shortest route system using the waterfall method.

The data collection method used in this study is observation, namely collecting secondary data in the form of street names in the Medan sub-district of Medan Amplas and determining the distance of each network point through Google Maps. Observational research via Google Maps is an approach that can be used to obtain information about the physical or geographic location of a particular place without having to physically be at that location. Google Maps is an online mapping service that provides maps and satellite images of various regions around the world. The dataset of the coordinates (latitude and longitude) and length of the Medan Amplas sub-district road consist of 212 data is given in Table 1. The location map of Medan Amplas district shown in Figure 1.



Figure 1. Location Map of Medan Amplas District

Table 1. Data on the Coordinates and Length of the Medan Amplas Sub-District Road

Id	Lat	Long	Label
1	3,549008	98,689384	Jl. <u>Perbatasan Barat</u> - Jl. <u>Perbatasan Bar.</u>
4	3,548889	98,691794	Jl. <u>Perbatasan Barat</u> - Jl. <u>STM</u>
5	3,54889	98,691862	Jl. <u>Perbatasan</u> - Jl. <u>STM</u>
6	3,549493	98,691865	Jl. <u>Pembangunan</u> - Jl. <u>STM</u>
.	.	.	.
165	3,55045	98,718837	Jl. <u>Selambo</u> - Jl. <u>Selambo V</u>
166	3,550761	98,720118	Jl. <u>Selambo</u> - Jl. <u>Selambo VI</u>
167	3,547578	98,716853	Jl. <u>Panglima Denai</u> - Jl. <u>Kerang</u>
.	.	.	.
212	3,532404	98,728116	Jl. <u>Medan - Pematang Siantar</u> - Jl. <u>Damai</u>
213	3,531782	98,744777	Jl. <u>Medan - Pematang Siantar</u> - Jl. <u>PerumTaman Riviera</u>

1. Waterfall Method

The Waterfall method is one of the Software Development Life Cycle Models (SDLC) which consists of: analysis, design, implementation, testing

and maintenance stages (Mariana, 2019). The stages of waterfall method are shown in Figure 2 (Trisianto, 2018).

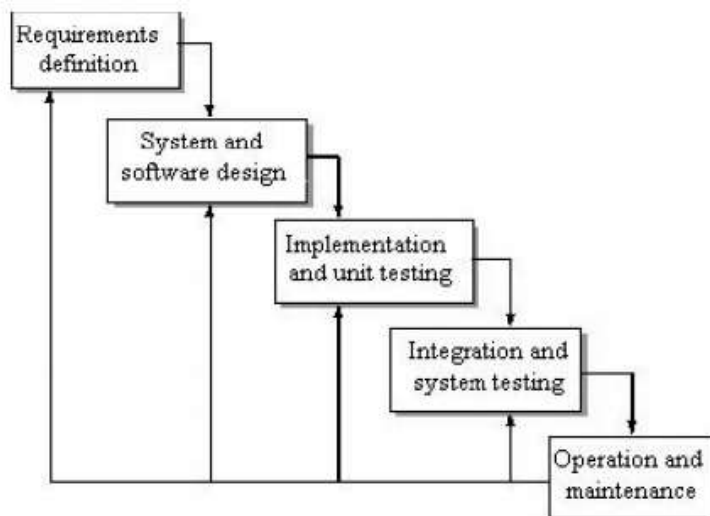


Figure 2. Stages of the Waterfall Method

In Figure 2, system development from the requirements analysis phase includes hardware and software. In the design phase, data flow diagram design is carried out which aims to facilitate understanding of the system being built. Implementation phase of activities to implement all designs have been made. At this stage, the programming process of the Floyd-Warshall algorithm is carried out using HTML Web Applications, supported by the PHP and JavaScript programming languages using the MySQL Database at each stage between the two vertices, until the approximation is known as the optimal value. The customer-delivered software will undoubtedly develop issues at some point, necessitating the maintenance phase. This modification may be necessary due to a software error that needs to be adjusted to the surroundings (new peripherals or operating system), or it may be required by customers who require development to be operational.

To find the shortest route is carried out at the testing stage. There are three tests carried out in this study (Trisianto, 2018), namely:

1. Black box testing, this test is fundamental aspects of the system without paying attention to the internal logical structure of software. Black Box testing tends to be concerned with the system. Functional testing is one kind of black box testing that is frequently utilized. To determine what the code is meant to do, functional testing makes use of high-level design testing approaches and customer-performed specifications to plan test cases. Functional testing guarantees that the features listed in the requirements specification function as intended.
2. Testing the calculation results, the test is carried out by comparing the calculation of the Initial W_{ij} matrix generated by the system with the Initial W_{ij} matrix which is calculated manually. If the final matrix results are obtained from the system with the

final matrix calculated manually, then the system is running as expected.

3. Testing of correctness results of the path and mileage generated by the system, is carried out by passing and observing the road network in the Medan sub-district and comparing the paths with the results of the system calculations.

2. Algorithm Floyd-Warshall

The Floyd-Warshall algorithm is an algorithm used to find the shortest path between all pairs of vertices in a weighted graph, where the weight on each edge (or edge) is a number that represents the distance or cost between two connected vertices. This algorithm can be used on directed or undirected graphs, with or without cycles, and can handle edges with negative weights, provided there are no negative cycles in the graph. The Floyd-Warshall algorithm has a time complexity of $O(n^3)$, where n is the number of nodes in the graph. This makes it efficient for medium to small sized graphs, but may be inefficient for very large graphs.

This algorithm is best known for its ability to handle graphs with negative weights if there are no negative cycles, but the Bellman-Ford algorithm is more commonly used if handling negative cycles is a primary requirement. The Floyd-Warshall algorithm is a type of dynamic programming, which solves problems by viewing the answer as a series of interconnected decisions. This signifies that the solution is derived from the preceding stage's solutions, and there may be more than one solution. To tackle the problem at hand in this study, the Floyd-Warshall method is utilized to find the shortest path between two or more points defined on a digital map.

The Floyd-Warshall algorithm is an algorithm that is often used in the context of roads and transportation to find the shortest route between all pairs of points (for example, cities or road intersections) in a road network. In this application, the Floyd-Warshall algorithm can assist in trip planning, vehicle routing, and transportation network optimization.

To find the shortest route between two specific locations, you can use the resulting shortest distance matrix. Find a route from the start node to the end node by following the graph steps based on the information in the distance matrix.

This algorithm can also be used for road network optimization. You can identify roads with heavy traffic or roads that require maintenance by checking the shortest distance matrix. Then, you can plan road repairs or modifications based on these findings. The Floyd-Warshall algorithm has a major advantage in modeling road networks, namely that it can handle directed graphs with negative weights (for example, roads with light traffic) and allows you to find the shortest route between all pairs of vertices in one iteration.

However, keep in mind that on a large scale, this algorithm can be expensive in terms of computing resources, especially if there are thousands of nodes in the road network. Therefore, for practical applications, other approaches are often used such as Dijkstra's algorithm to find the shortest route between two certain points or the A* algorithm for heuristic-based routing.

The Floyd-Warshall approach compares each route variation to the target point by adding the distance between nodes (crossing points) that are passed. The numerical value of the line connecting the nodes represents the weight of the distance. The Floyd-Warshall algorithm will employ each line as a calculation material to discover the shortest route (Kriswanto, Bendi, & Aliyanto, 2014). The flowchart algorithm Floyd-Warshall can be seen in Figure 3.

The Floyd-Warshall algorithm, which is referred in (Khamami & Saputra, 2019), is as follows:

```

1)  $W = W_0$ 
2) For  $k = 1$  to  $n$ , do :
For  $i = 1$  to  $n$ , do :
For  $j = 1$  to  $n$ , do :
If  $W_{[i,j]} > W_{[i,k]} + W_{[k,j]}$ 
Then replace  $W_{[i,j]}$  with  $W_{[i,k]} + W_{[k,j]}$ 
3)  $W^* = W$ 
Declarations :
 $W = \text{matrix}$ 
 $W_0 = \text{first adjacency matrix graph}$ 
 $k = \text{iteration 1 to } n$ 
 $i = \text{first point } v_i$ 
 $j = \text{first point } v_j$ 
 $W^* = \text{result of matrix after comparison}$ 

```

The Floyd Warshall algorithm's minimum value determination procedure can be expressed mathematically as follows (Sahoo, Sen, Tiwary, Samanta, & Senapati, 2022):

1. Each matrix cell is examined in the 1st iteration to see whether the distance between the two beginning points exceeds the total of the distances from the origin to the destination (the destination point represents the first iteration) and the distance from the original point (the origin represents the first iteration). Alternatively, ask yourself if $W_{ij} > W_{ik} + W_{kj}$.
2. If so, then starting from the origin (origin = 1st iteration) to the destination ($W_{ik} + W_{kj}$), the distance between the two starting locations is replaced by the total of the distance from the origin to the destination (destination point = 1 iteration).
3. If not, the distance between the two beginning places (W_{ij}) is used.
4. Up until the last iteration, the iteration procedure is carried out (the number of iterations equals the total number of points).

The Floyd-Warshall method forms a matrix during its iteration to determine the shortest path, which corresponds to the k-iteration. This will slow down the procedure, especially for high values. Despite its slow processing time, the Floyd-Warshall algorithm is frequently employed to find the shortest path due to its simplicity.

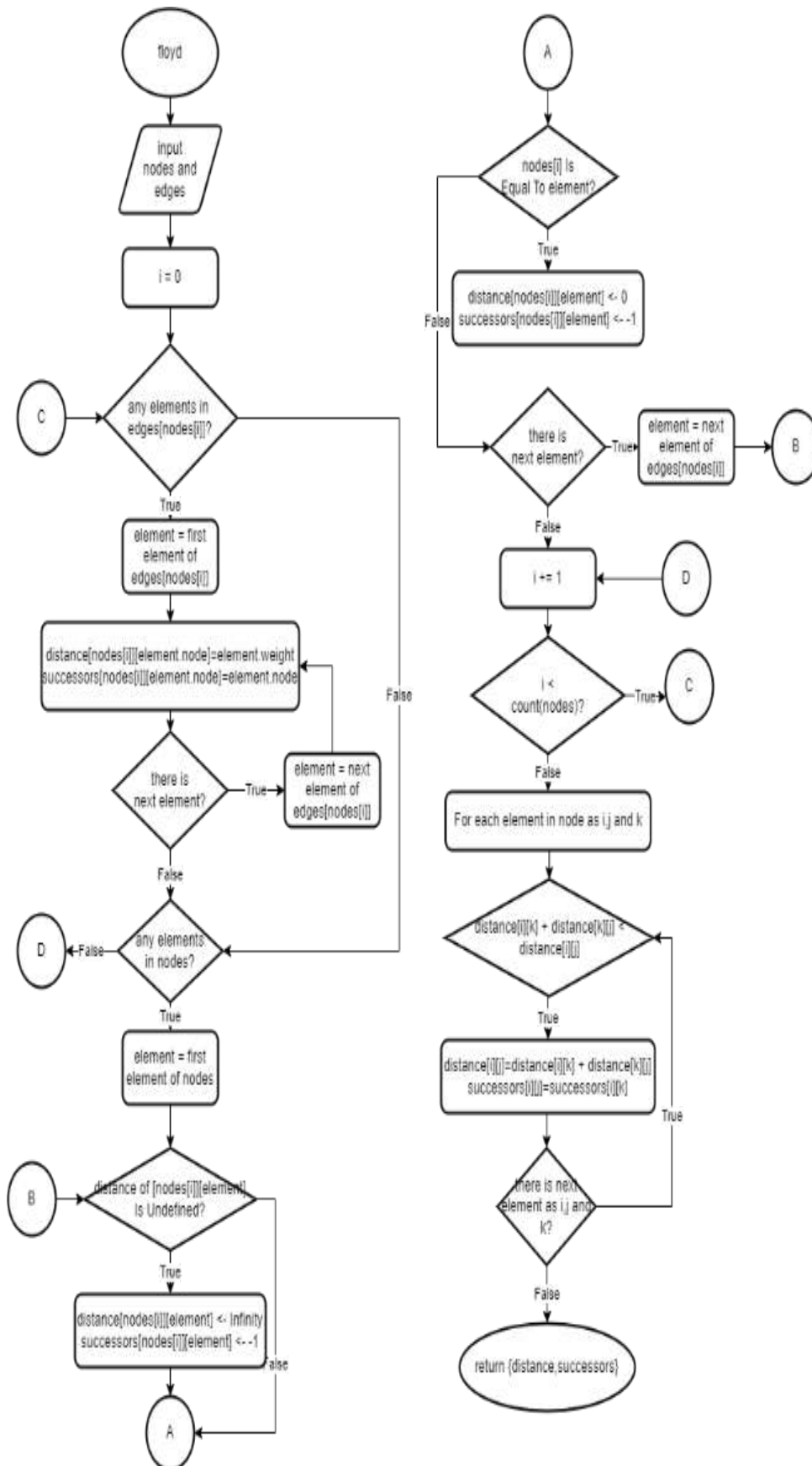


Figure 3. Flowchart Function Floyd-Warshall

RESULTS AND DISCUSSION

Medan Amplas is one of the 21 sub-districts located in Medan City, North Sumatra Province, Indonesia. This sub-district is bordered by Medan Johor to the west, Deli Serdang Regency to the east and south, and Medan City and Medan Denai to the north. Medan Amplas District has a diverse population with different ethnic, religious and cultural backgrounds. The population consists of various social groups, creating social and cultural diversity. This subdistrict provides various facilities and services for its residents, such as government centers, education, health care, commerce, and places of worship. There are various shops, traditional markets, hospitals, schools and

government institutions in this sub-district. Like many areas in Medan, this sub-district has diverse economic sectors, including trade, industry, agriculture and services. Medan Amplas District also has various small and medium businesses that support the local economy. This sub-district may not be known as a major tourist destination, but it has several interesting places and restaurants that visitors and locals can enjoy. This sub-district has 7 sub-districts, namely Amplas, Bangun Mulia, Harjosari I, Harjosari II, Sitirejo II, Sitirejo III and Tambang Deli. The dataset consists of 211 data, is given in Table 1.



Figure 4. Road Data in Graph Form

The initial stage of searching for the shortest route is transforming road coordinate data in Medan sub-district of Medan Amplas in the form of a graph, which can be seen in Figure 4. In Figure 4, forms a road network in Medan sub-district of Medan Amplas which consists of 211 points. The graph consists of three key point, namely vertices, edge, and weights. In this study, the vertices of graph are represented as street names in Medan Subdistrict, the edges represent roads which connect the street names in Medan Subdistrict and the edge

weights represent the length of the road network. The point in this case represents the coordinates of the road. The existing points are connected to each other according to the road coordinate, data obtained from Google Maps in Medan sub-district of Medan Amplas. There are 553 sides formed into a road network.

The Programming of Floyd-Warshall Algorithm

The Floyd-Warshall algorithm is a search algorithm shortest path using a directed weighted graph with the weight of the positive or negative side, but does not have a cycle with negative weight. This algorithm calculates the shortest distance from all pairs of vertices in a directed weighted graph. One example of the application of the Floyd-Warshall algorithm is in road maps. This algorithm returns the smallest weight for each pair of nodes. Basically, this algorithm does not return the sequence of paths taken to get the smallest weight from one node to another. However, with certain modifications a sequence can be obtained the path taken to get the shortest path from the required pair of nodes or for each pair of nodes.

The Floyd-Warshall algorithm is an algorithm in computer science that is used to find the shortest distance between all pairs of vertices in a weighted graph. This algorithm works on directed or undirected graphs with weights on each edge. The main goal of this algorithm is to find a shortest distance matrix that contains the shortest distances

between all pairs of vertices in the graph. Applications can be run through web-based applications that have been designed using programming of the Floyd-Warshall algorithm uses HTML Web Application, supported by the PHP and JavaScript programming languages using the MySQL and database obtained. The main algorithm to run the web-based applications given in Figure 6. The web-based application is displayed in Figure 5 and consists of several menus:

- Floyd-Warshall menu, which shows that the algorithm used in this route search is the Floyd-Warshall algorithm.
- Node menu, contains a list of points that have been entered into the database.
- Connection menu, to input the starting point and destination point where you want to find the shortest route.
- Graph menu,
- User account menu.

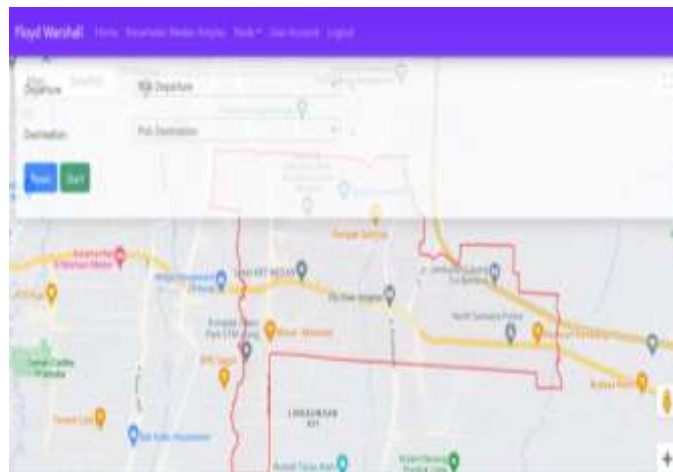


Figure 5. Web Application Display



Figure 6. Main Program

The calculation process with the Floyd-Warshall algorithm in finding the shortest path between all points in the network is carried out after getting nodes and paths, then the initial W_{ij} matrix and Initial Z_{ij} matrix are obtained. Initial W_{ij}

matrix is a matrix that is useful for finding the shortest distance of each pair of points, while the Initial Z_{ij} matrix is useful for finding the shortest path between points. The flowchart to obtain the Matrix is shown in Figure 7.

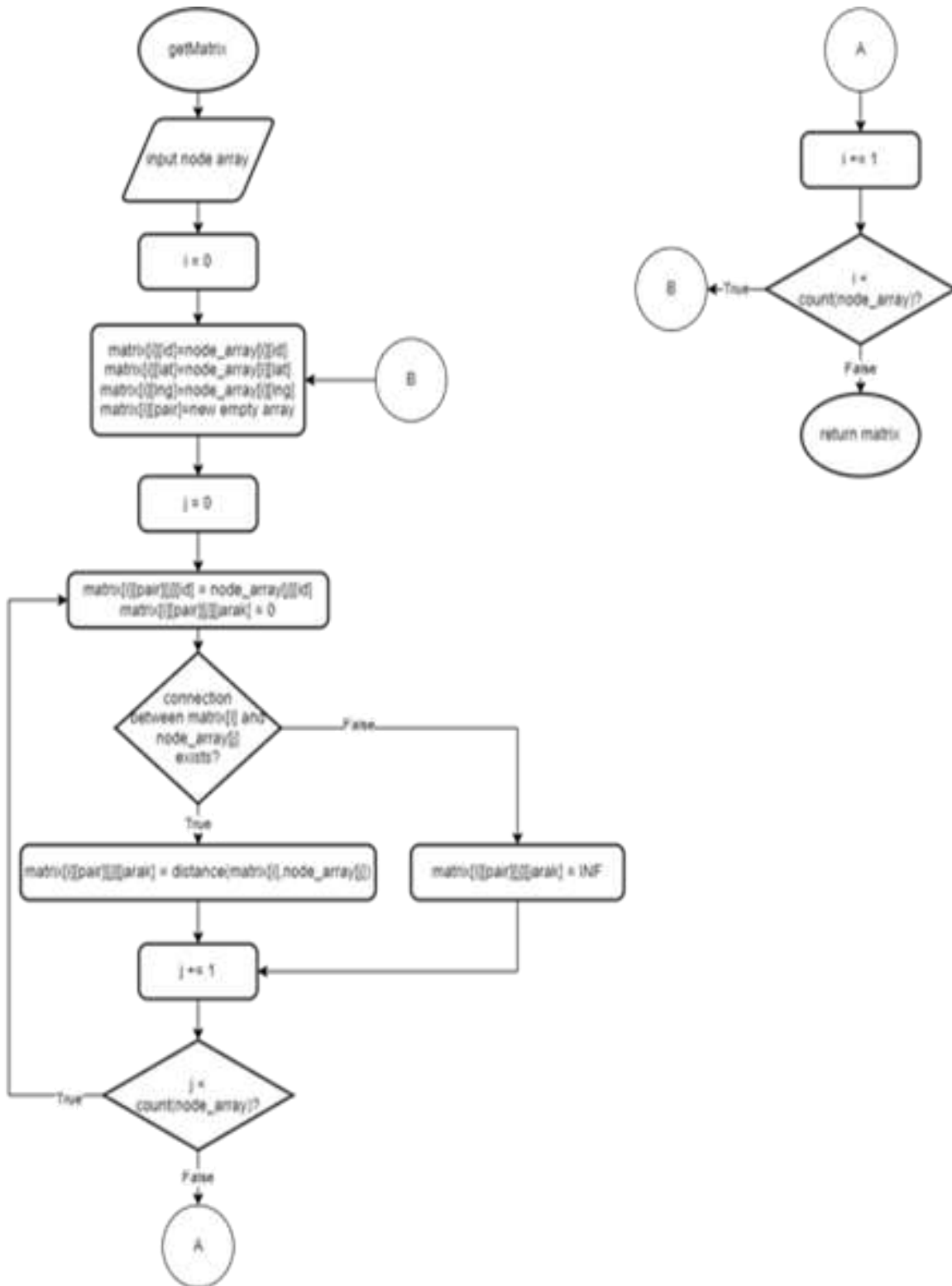


Figure 7. Flowchart Function Get Matrix()

Shortest Route Search

Through the system built, it can find out the shortest path taken on the way from the initial position to the destination position in Medan sub-district. The results obtained from the results of the Floyd-Warshall calculation provide the optimal choice of travel routes in terms of distance. Running the Floyd-Warshall algorithm and calculating the shortest distance matrix, you can use the matrix to find the shortest route between two specific points in the graph. The `find_shortest_path` function will find the shortest route between two points in the graph using the shortest distance matrix generated

by the Floyd-Warshall algorithm. The result is a list of nodes that form the shortest route. If no routes are available, it will return the message "No routes available". In this example, the shortest route between vertex 0 and vertex 3 is searched and printed.

Figure 8 provides a view of a web application looking for the shortest path. For example, the shortest route from Jl. Development towards Jl. Dwikora. After pressing the start button, the results of the shortest route solution are obtained in detail that needs to be taken based on the input departure and destination as shown in Figure 8.

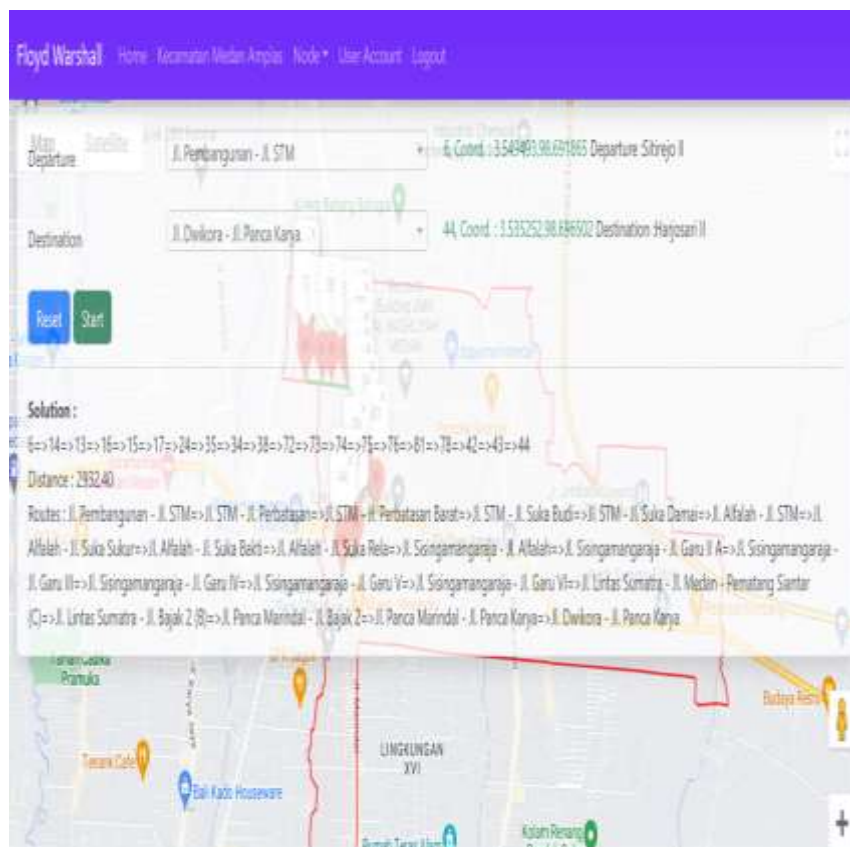


Figure 8. The Web Application Displays Looking for the Shortest Path

Next, three types testing are carried out to measure system reliability. The first test results using the Black Box method showed that the system functionality was 100% running well. For the second test, its function is to check the calculation results of the Final W_{ij} and Final Z^* matrices produced by the system using the application that was created.

And the last stage is testing the reliability of the system created, the results of the system reliability test on the web-based application show that it works properly and is acceptable and there are no path deviations. Based on the results of the three tests, it can be concluded that the results obtained are in accordance with expectations.

getMatrix function

Function getMatrix()

node array <- input nodes array or read from nodes database.

Matrix <- empty array, output matrix

For I = 0 to count(node array) Do

matrix[i][id]<-node array[i][id]

matrix[i][lat]<-node array[i][lat]

matrix[i][lng]<-node array[i][lng]

matrix[i][pair]<-new empty array

For j=0 to count(node array) Do

matrix[i][pair][j][id] <- node array[j][id]

If matrix[i][pair][j][id] =

node array[j][id] Then

matrix[i][pair][j][jarak] = 0

Else

If connection between matrix[i] and

node array[j] exists Then

matrix[i][pair][j][jarak] <- distance

between matrix[i] and node array[j]

Else

matrix[i][pair][j][jarak] <- INF

Endif

Endif

Endfor

Endfor

return matrix

endfunction

floyd function

function Floyd()

nodes <- input nodes array

edges <- input nodes connection array of

object

successors <- successor of node array

distance <- output nodes distance array

For i=0 to count(nodes) Then

For Each element in edges[nodes[i]] Do

distance[nodes[i]][element node] <-

element weight

successors[nodes[i]][element node] <-

element node

Endfor

For Each element in nodes Do

If distance[nodes[i]][element] Is

Undefined Then

distance[nodes[i]][element] <- Infinity

successors[nodes[i]][element] <- -1

Endif

If nodes[i] Is Equal To element Then

distance[nodes[i]][element] <- 0

successors[nodes[i]][element] <- -1

Endif

Endfor

Endfor

For Each I in nodes

For Each j in nodes

For each k in nodes

If distance[i][k] + distance[k][j] <

distance[i][j] Then

distance[i][j] <- distance[i][k] +

distance[k][j]

successors[i][j] <- successors[i][k]

Endif

Endfor

Endfor

Endfor

return distance

Endfunction

Main Prosedur

```
Sub FloydRouteSearch()
  node1 <- source node
  node2 <- destination node
  matrix <- Call getMatrix()
  nodes <- empty array
  edges <- empty array
  floydModel <- empty Floyd model

  FinalRute <- empty results route array
  NextRute <- empty node

  For i=0 to count(matrix) Do
    nodes[i] <- matrix[i][id]
    For j=0 to count(matrix[i][pair]) Do
      edges[nodes[i]] <-
      {matrix[i][pair][j][id],matrix[i][pair][j][distance]}
    Endfor
  Endfor

  floydModel <- Call Floyd(nodes,edges)
  FinalRute.Push(node1)
  NextRute <-
  floydModel.successors[node1][node2]
  If NextRute != -1 Then
    While True
      If floydModel.successors[NextRute][node2]
      != -1 Then
        NextRute <-
        floydModel.successors[NextRute][node2]
        FinalRute.Push(NextRute)
        If NextRute = node2 Then
          Exit While
        Endif
      Else
        Exit While /* routes cannot be found.
      Endif
    Endwhile
  Endif
  write(FinalRute)
Endsub
```

CONCLUSION

Based on the findings of this research, the shortest path search application based on the Floyd-Warshall algorithm can be deployed in the Medan sub-district of Medan Amplas. The search for the shortest route from the starting point to the destination point begins with converting the road coordinate data from Google Map into the form of a road network. The network is then transformed into a neighboring matrix and the shortest path choice is determined using the Floyd-Warshall method.

ACKNOWLEDGMENTS

Thank are conveyed to Universitas Negeri Medan through the Research Institute UNIMED which has funded this research. Thanks also to the Dean of MIPA UNIMED for supporting facilities and infrastructure during this research.

REFERENCES

- Alam, M. A., & Faruq, M. O. (2019). Finding Shortest Path for Road Network Using Dijkstra's Algorithm. *Bangladesh Journal of Multidisciplinary Scientific Research*, 1(2), 41–45. <https://doi.org/10.46281/bjmsr.v1i2.366>
- Aziz, A., Farid, M. M., & Suryani, E. (2017). Floyd warshall algorithm with FIS Sugeno for search evacuation route optimization. *2017 International Seminar on Application for Technology of Information and Communication (ISEMANTIC)*, 147–151. IEEE. <https://doi.org/10.1109/ISEMANTIC.2017.8251860>
- Bernstein, A. (2013). Maintaining shortest paths under deletions in weighted directed graphs. *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, 725–734. New York, NY, USA: ACM. <https://doi.org/10.1145/2488608.2488701>
- Dela Cruz, J. C., Magwili, G. V., Mundo, J. P. E., Gregorio, G. P. B., Lamoca, M. L. L., & Villasenor, J. A. (2016). Items-mapping and route optimization in a grocery store using Dijkstra's, Bellman-Ford and Floyd-Warshall Algorithms. *2016 IEEE Region 10 Conference (TENCON)*, 243–246. IEEE. <https://doi.org/10.1109/TENCON.2016.7847998>
- Hougardy, S. (2010). The Floyd–Warshall algorithm on graphs with negative cycles. *Information Processing Letters*, 110(8–9), 279–281. <https://doi.org/10.1016/j.ipl.2010.02.001>

- Khamami, A., & Saputra, R. (2019). The shortest path search application based on the city transport route in Semarang using the Floyd-warshall algorithm. *Journal of Physics: Conference Series*, 1217(1), 012116. <https://doi.org/10.1088/1742-6596/1217/1/012116>
- Kriswanto, Y. R., Bendi, R. K. J., & Aliyanto, A. (2014). Penentuan Jarak Terpendek Rute Transmisi dengan Algoritma Floyd-Warshall. *Semantik 2014*, 4(1), 209–216. Retrieved from <https://publikasi.dinus.ac.id/index.php/semantik/article/view/865>
- Li, L. (2020). Education supply chain in the era of Industry 4.0. *Systems Research and Behavioral Science*, 37(4), 579–592. <https://doi.org/10.1002/sres.2702>
- Malisan, J., Marpaung, E., Hutapea, G., Puriningsih, F. S., & Arianto, D. (2023). Development of short sea shipping in the north coast of Java Island, Indonesia as a potential market. *Transportation Research Interdisciplinary Perspectives*, 18, 100760. <https://doi.org/10.1016/j.trip.2023.100760>
- Mardlootillah, H. I., Suyitno, A., & Arini, F. Y. (2014). Simulasi Algoritma Dijkstra Dalam Menangani Masalah Lintasan Terpendek Pada Graf Menggunakan Visual Basic. *Unnes Journal of Mathematics*, 3(1), 56–61. <https://doi.org/https://doi.org/10.15294/ujm.v3i1.3285>
- Mariana, A. N. (2019). Implementation of the Waterfall Model for the Development of the DKI Jakarta PKK Current Mail Information System. *JURIKOM (Jurnal Riset Komputer)*, 6(5), 453–459. <https://doi.org/http://dx.doi.org/10.30865/jurikom.v6i5.1392>
- Nova, K., A, U., Jacob, S. S., Banu, G., Balaji, M. S. P., & S, S. (2023). Floyd–Warshalls algorithm and modified advanced encryption standard for secured communication in VANET. *Measurement: Sensors*, 27, 100796. <https://doi.org/10.1016/j.measen.2023.100796>
- Ojekudo, N. A., & Akpan, N. P. (2017). Anapplication of Dijkstra’s Algorithm to shortest route problem. *IOSR Journal of Mathematics*, 13(1), 20–32. <https://doi.org/10.9790/5728-1303012032>
- Pojani, D., & Stead, D. (2015). Sustainable Urban Transport in the Developing World: Beyond Megacities. *Sustainability*, 7(6), 7784–7805. <https://doi.org/10.3390/su7067784>
- Putra, A. S., & Warnars, H. L. H. S. (2018). Intelligent Traffic Monitoring System (ITMS) for Smart City Based on IoT Monitoring. *2018 Indonesian Association for Pattern Recognition International Conference (INAPR)*, 161–165. IEEE. <https://doi.org/10.1109/INAPR.2018.8626855>
- Sadavare, A. B., & Kulkarni, R. V. (2012). A review of application of graph theory for network. *International Journal of Computer Science and Information Technologies*, 3(6), 5296–5300. Retrieved from <https://ijcsit.com/docs/Volume3/vol3Issue6/ijcsit2012030601.pdf>
- Sahoo, L., Sen, S., Tiwary, K., Samanta, S., & Senapati, T. (2022). Modified Floyd–Warshall’s Algorithm for Maximum Connectivity in Wireless Sensor Networks under Uncertainty. *Discrete Dynamics in Nature and Society*, 2022, 1–11. <https://doi.org/10.1155/2022/5973433>
- Schepers, P., Twisk, D., Fishman, E., Fyhri, A., & Jensen, A. (2017). The Dutch road to a high level of cycling safety. *Safety Science*, 92, 264–273. <https://doi.org/10.1016/j.ssci.2015.06.005>
- Situmorang, R., Trilaksono, T., & Japutra, A. (2019).

Friend or Foe? The complex relationship between indigenous people and policymakers regarding rural tourism in Indonesia. *Journal of Hospitality and Tourism Management*, 39, 20–29.

<https://doi.org/10.1016/j.jhtm.2019.02.001>

Soemardi, B. W., & Pribadi, K. S. (2021). The Indonesian Construction Industry, 1995–2019. In *Construction Industry Advance and Change: Progress in Eight Asian Economies Since 1995* (pp. 63–87). Emerald Publishing Limited. <https://doi.org/10.1108/978-1-80043-504-920211004>

Surya, B., Hamsina, H., Ridwan, R., Baharuddin, B., Menne, F., Fitriyah, A. T., & Rasyidi, E. S. (2020). The Complexity of Space Utilization and Environmental Pollution Control in the Main Corridor of Makassar City, South Sulawesi, Indonesia. *Sustainability*, 12(21), 9244. <https://doi.org/10.3390/su12219244>

Trisianto, C. (2018). Penggunaan metode waterfall untuk pengembangan sistem monitoring dan evaluasi pembangunan pedesaan. *ESIT*, 12(1), 8–22.