Best Routes Selection Using Dijkstra And Floyd-Warshall Algorithm

Risald
Universitas Atma Jaya Yogyakarta
Yogyakarta, Indonesia
Risald.uajy93@gmail.com

Antonio E. Mirino
Universitas Atma Jaya Yogyakarta
Yogyakarta, Indonesia
anthonio_mrn@yahoo.com

Suyoto
Universitas Atma Jaya Yogyakarta
Yogyakarta, Indonesia
suyoto@staff.uajy.ac.id

Abstract— ***Road accidents are frequent and many cause casualties. Fast handling can minimize the number of deaths from traffic accidents. In addition to victims of traffic accidents, there are also patients who need emergency handling of the disease he suffered. One of the first help that can be given to the victim or patient is to use an ambulance equipped with medical personnel and equipment needed. The availability of ambulance and accurate information about victims and road conditions can help the first aid process for victims or patients. Supportive treatment can be done to deal with patients by determining the best route (nearest and fastest) to the nearest hospital. The best route can be known by utilizing the collaboration between the Dijkstra algorithm and the Floyd-warshall algorithm. This application applies Dijkstra's algorithm to determine the fastest travel time to the nearest hospital. The Floyd-warshall algorithm is implemented to determine the closest distance to the hospital. Data on some nearby hospitals will be collected by the system using Dijkstra's algorithm and then the system will calculate the fastest distance based on the last traffic condition using the Floyd-warshall algorithm to determine the best route to the nearest hospital recommended by the system. This application is built with the aim of providing support for the first handling process to the victim or the emergency patient by giving the ambulance calling report and determining the best route to the nearest hospital.***

Keywords— ***Health services; Dijkstra's Algorithm; Floyd-Warshall Intelligent System Algorithm.***

I. INTRODUCTION

Traffic accidents are one of the highest causes of death in the world. Therefore, victim handling becomes very important. One of the treatment by knowing the best route (fastest and closest), to the hospital [1]. Dijkstra's algorithm can be used to determine the fastest time [2]–[4]. Dijkstra's algorithm runs in O (N (mþn log n)) to solve the shortest path problem [5], [6]. In addition to determining the fastest time, it takes also the shortest route determination. Floyd-warshall algorithm can be used to determine the shortest path [7], [8]. The collaboration between Dijkstra and Floyd-warshall's algorithms is excellent for determining the best path for evacuation of accident victims. [9].

This evacuation must pass through a slightly obstructed path (distance and vehicle density)[10]. Therefore it takes a system that can determine the best route. There have been several systems related to the medical world [11], [12]. There are also mobile-based systems [13]-[15]. The system can sort the fastest distance based on the level of traffic flow [16].

Booking the route from the closest to the farthest is considered very helpful in the handling of traffic accident victims. It aims to reduce the number of victims who died from traffic accidents. Mobile-based systems are considered easier to use anywhere. Therefore, the handling of accident victims by using a mobile-based system that can sort routes based on the best distance to the hospital is very good to provide assistance to users of the system in helping the victims of accidents [17]–[19].

II. LITERATUR REVIEW

In 2013 Rafael Rodríguez-Puente and Manuel S Lazo-Cortés conducted a study entitled “Algorithm for shortest path search in geographic information systems by using reduced graphs”. The purpose of this study reduces the time run in the search space. This article proposes modification of Dijkstra's shortest path search algorithm. This study shows that the cost of the path found in this work, equal to the cost of the path found by using the Dijkstra algorithm in the original graph [19].

In 2014 Maria P. Scaparra, Richard L. Church and F. Antonio Medrano conducted a study entitled “Corridor location: the multi-gateway shortest path model”. The purpose of this study presents the application of this model to a real landscape and compares the results with past work. Overall, a new model called the multi-gateway shortest path problem can result in a variety of efficient alignments, which shake up what the work of the past can do [14].

III. METHOD AND MATERIALS

A. DIJKSTRA'S ALGORITHM

Dijkstra's algorithm is the most used algorithm in the shortest route search, simple to use with simple nodes on an uncomplicated road network. In searching for solutions Dijkstra's algorithm uses the principle of greedy, which is to find the optimum solution at every step, in order to get the optimum solution in the next step which will lead to the best solution. This makes the time complexity of Dijkstra's algorithm to be quite large, that is 0 (V * Log (v + e)), Where v and e are vertices and sides of the graph used.

The input of the Dijkstra algorithm is a weighted graph G (e, v), while the output is the shortest route from the initial node to each vertex in the graph thus the algorithm Dijkstra can find the best solution.

Dijkstra algorithm works similar to the workings of Breadth First Search algorithm (BFS) is to use the principle of queue, but the queue Dijkstra used algorithm is a priority queue. So only the nodes that have the highest priority will be searched. In determining the priority node, this algorithm compares each value (weight) of each node is stored to be compared to the value to be found from the next route, and so on until a node is found in the search.

B. FLOYD-WARSHALL ALGORITHM

Floyd-warshall's algorithm does not just look for the shortest path between two particular nodes, but the shortest path table between the nodes is created. Floyd-warshall's algorithm is one of the dynamic programming variants, which is a method that performs problems with the solution that will be generated as an interconnected decision. Solution solutions are formed from the previous and there is a possibility of more than one solution.

Floyd-warshall algorithm is one type of algorithm all pair shortest path, which is to find the shortest route for all pairs of nodes that exist on a graph. The input of this algorithm is a weighted and directed graph. The algorithm also calculates the negative-weighted side.

C. COLLABORATION OF DIJKSTRA AND FLOYD-WARSHALL'S ALGORITHMS

The thing that distinguishes solution search using dynamic programming and greedy algorithm is the decision taken at each stage of the greedy algorithm, based only on limited information so that the optimum value obtained at that time does not cover the overall optimum value. So in the greedy algorithm we do not think of the consequences that occur if we choose a decision at a stage.

In some cases, the greedy algorithm fails to provide the best solution because of the weakness it has. This is where the role of dynamic programming that tries to provide solutions that have thoughts on the consequences arising from decision-making at a stage. Dynamic programming is capable of reducing unsubstantiated resolution.

Dijkstra's algorithm is an algorithm that uses a greedy principle that needs to collaborate with the dynamic Floyd-warshall programming algorithm to find the best solution that can reduce decision making that is not a solution.

IV. RESULT AND DISCUSSION

A. DESIGN INTERFACE

Figure 1. Home Interface

In the picture above (Figure 1), this interface serves to display data of all nearby hospitals from the user's location. This map serves to provide an accurate picture of the location to users. In addition to maps, there is also a "get directions" button. This button serves to sort the location of the hospital based on distance and travel time. Orders from the hospital location will be displayed in the next interface.

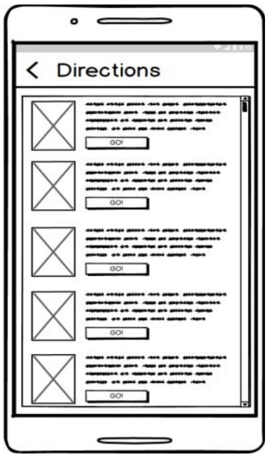

Figure 2. Directions Interface

In the picture above (Figure 2), serves to show the order of the location of the nearest hospital with distance and time using Dijkstra and Floyd-warshall algorithms, the information displayed is a picture of the hospital, hospital address, contact number, and "GO" button. This button serves to display a map that contains routes to the hospital. The route map will appear in the next interface.

Figure 3. Location Interface

In the picture above (Figure 3), serves to display a map that contains a route to the location of the hospital that has been selected in the previous interface. If the user wants to change the intended hospital, the user can return via the navigation button located in the top left corner of the screen and then select other hospital data.

B. DESIGN SYSTEM

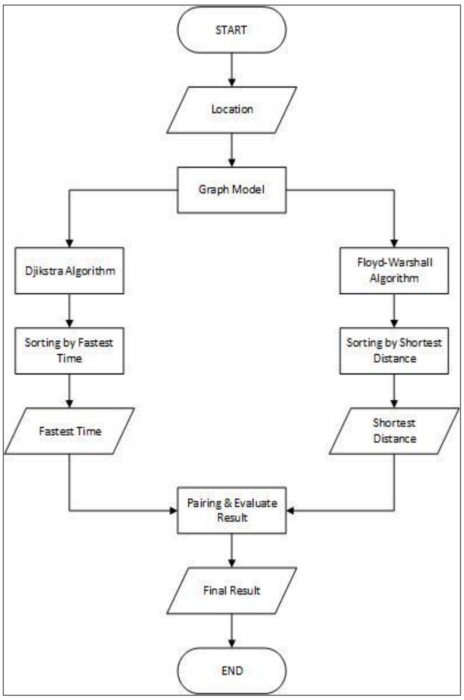

Figure 4. Flowchart System

In the picture above (Figure 4), initially the system will retrieve location data. Then the system will do a graph model, this model serves to determine the location based on 2 terms, namely mileage and travel time. To sort location data based on travel time will use the Dijkstra algorithm. To sort location data by distance will use Floyd-warshall algorithm. After the location data is sorted, the next process is to match the distance and location of the route based on the smallest number and then the system displays the final result.

C. DISCUSSION

We try to design the best route determination app based on the shortest distance and time. The system uses Dijkstra's algorithm to determine the fastest time and Floyd-warshall algorithm to determine the closest distance. Routing can be seen in the following example (Figure 5).

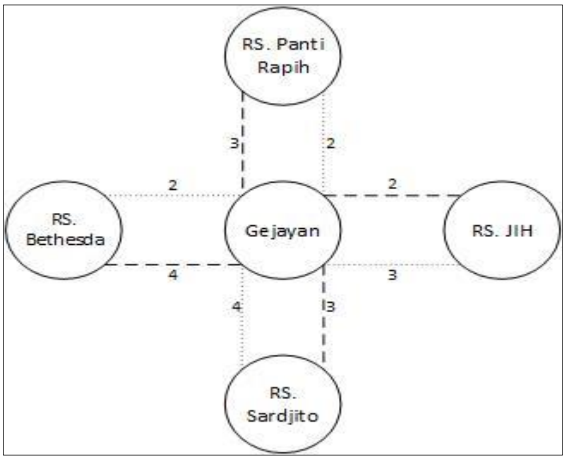

Figure 5. Graph Model

TABLE I. DATA DISTANCE AND TIME

Starting Direction	Distance	Time
Panti Rapih Hospital	3	2
JIH Hospital	2	3
Sardjito Hospital	3	4
Bethesda Hospital	4	2

TABLE II. FINAL SORTING

Point X	Distance	Time
JIH Hospital	2	3
Panti Rapih Hospital	3	2
Sardjito Hospital	3	4
Sardjito Hospital	4	2

In the table above (TABLE II), it can be seen that the system is sorted by the shortest distance and the shortest time. Initially, the system will determine the location data of several hospitals that are closest to the user's location. Hospital location data will be calculated using the Dijkstra algorithm to get the order of location data with the shortest distance to the longest. At the same time, the system will calculate the location data using the Floyd-warshall algorithm to obtain the location data sequence with the closest distance to the furthest distance. Once the distance data is known, then the system will re-sort the data distance as final sorting. The final sorting begins by determining the shortest distance and then continuing with the shortest time. For example, Panti Rapih Hospital and Sardjito Hospital have the same distance with 3. However, the system will prioritize Panti Rapih Hospital because it has a shorter time than 2 hospitals namely Sardjito Hospital which has a distance of 3. This result will be shown in the direction Interface.

This application is built with the aim to help users in determining the hospital to be the goal, it is useful to save the time required to deliver patients or casualties to the nearest hospital. This application provides hospital information based on the distance and the nearest travel time so that the user can deliver the patient or accident victim just in time to the nearest hospital to get more serious treatment from medical personnel. This application in the future will certainly be developed again in accordance with the needs and suggestions of users.

V. CONCLUSION

From the results of research that has been done there are some conclusions that can be taken are:

I. Factors taken into account in determining the shortest and fastest route are the distance to the nearest hospital and the density of the traffic flow.

II. If there are problems on the way to the hospital such as unexpected factors such as traffic accidents then the location of the hospital can be moved to other nearby hospitals.

III. Merging these two algorithms can mask the deficiencies of each algorithm. Dijkstra's algorithm, in collaboration with the Floyd-warshall algorithm, speeds up search time by reducing decision making that does not lead to solutions.

REFERENCES

[1] D. T. Wilson, G. I. Hawe, G. Coates, and R. S. Crouch, "A multi-objective combinatorial model of casualty processing in major incident response," *Eur. J. Oper. Res.*, vol. 230, no. 3, pp. 643–655, 2013.

[2] A. Sedeño-Noda and A. Raith, "A Dijkstra-like method computing all extreme supported non-dominated solutions of the biobjective shortest path problem," *Comput. Oper. Res.*, vol. 57, pp. 83–94, 2015.

[3] M. S. Bazaraa and R. W. Langley, "A Dual Shortest Path Algorithm," *SIAM J. Appl Math*, vol. 26, no. 3, pp. 496–502, 2011.

[4] W. Peng, X. Hu, F. Zhao, and J. Su, "A fast algorithm to find all-pairs shortest paths in complex networks," *Procedia Comput. Sci.*, vol. 9, pp. 557–566, 2012.

[5] Y. Zhang, G. Xie, and J. Chen, "A New Distribution Path Selection Method for Biomedical Cold Chain : Bidirectional Dijkstra Algorithm," *Sensors & Transducers*, vol. 158, no. 11, pp. 219–224, 2013.

[6] M. J. Henchey, R. Batta, A. Blatt, M. Flanigan, and K. Majka, "A study of situationally aware routing for emergency responders.," *J. Oper. Res. Soc.*, vol. 66, no. 4, pp. 570–578, 2015.

[7] P. ofner and B. Moller, "Dijkstra, floyd and warshall meet kleene," *Form. Asp. Comput.*, vol. 24, no. 4–6, pp. 459–476, 2012.

[8] A. Pradhan and G. Mahinthakumar, "Finding all-pairs shortest path for a large-scale transportation network using parallel floyd-warshall and parallel Dijkstra algorithms," *J. Comput. Civ. Eng.*, vol. 27, no. June, pp. 263–273, 2012.

[9] Y. Z. Chen, S. F. Shen, T. Chen, and R. Yang, "Path optimization study for vehicles evacuation based on Dijkstra algorithm," *Procedia Eng.*, vol. 71, pp. 159–165, 2014.

[10] H. Dezani, R. D. S. Bassi, N. Marranghello, L. Gomes, F. Damiani, and I. Nunes da Silva, "Optimizing urban traffic flow using Genetic Algorithm with Petri net analysis as fitness function," *Neurocomputing*, vol. 124, pp. 162–167, 2014.

[11] D.-W. Li, H. Yang, L. Han, and S. Huo, "Predicting the folding pathway of engrailed homeodomain with a probabilistic roadmap enhanced reaction-path algorithm.," *Biophys. J.*, vol. 94, no. March, pp. 1622–1629, 2008.

[12] T. Teijeiro, P. Felix, J. Presedo, and C. Zamarron, "An open platform for the protocolization of home medical supervision," *Expert Syst. Appl.*, vol. 40, no. 7, pp. 2607–2614, 2013.

[13] M. Wang, Y. Qian, and X. Guang, "Improved calculation method of shortest path with cellular automata model," *Emerald Sight*, no. 10, pp. 508–517, 2012.

[14] M. P. Scaparra, R. L. Church, and F. A. Medrano, "Corridor location: The multi-gateway shortest path model," *J. Geogr. Syst.*, vol. 16, no. 3, pp. 287–309, 2014.

[15] N. B. Nugraha, Suyoto, and Pranowo, "Mobile application development for smart tourist guide," Adv. Sci. Lett., pp. 2475–2477, 2017.

[16] T. Darwish and K. Abu Bakar, "Traffic aware routing in vehicular ad hoc networks: characteristics and challenges," *Telecommun. Syst.*, vol. 61, no. 3, pp. 489–513, 2016.

[17] A. E. IORDAN, "Development of an Interactive Environment Used for Simulation of Shortest Paths Algorithms.," *Ann. Fac. Eng. Hunedoara - Int. J. Eng.*, vol. 10, no. 3, pp. 97–102, 2012.

[18] I. García, J. Pacheco, and A. Alvarez, "Optimizing routes and stock," *J. Heuristics*, vol. 19, no. 2, pp. 157–177, 2013.

[19] R. Rodríguez-Puente and M. S. Lazo-Cortés, "Algorithm for shortest path search in Geographic Information Systems by using reduced graphs.," *Springerplus*, vol. 2, p. 291, 2013.