DIJKSTRA ALGORITHM IMPLEMENTATION IN DETERMINING SHORTEST ROUTE TO MOSQUE IN RESIDENTIAL CITRA INDAH CITY

Siti Lestari1; Ardiansyah2; Angelina Puput Giovani3; Desy Dwijayanti4

Computer Science
STMIK Nusa Mandiri Jakarta, Indonesia
www.nusamandiri.ac.id
11171086@nusamandiri.ac.id; masardi@gmail.com; angelinapuputg@gmail.com; desy.dwijayanti512@gmail.com

Abstract—The application of artificial intelligence (Artificial Intelligence) for problem solving (problem solving) in the field of computer science has experienced rapid development from year to year as the development of artificial intelligence itself. Problems involving searching (searching) is one example of the use of artificial intelligence that is quite popular to solve various kinds of problems. In daily activities, the use of roads is always an unavoidable activity, so determining the shortest path from one point to another becomes a problem that is often encountered. This is also felt by residents who live in a large enough housing. Sometimes to be able to reach the destination they are often confused in deciding which way to go to get the shortest distance to the destination. Citra Indah City Housing is a residential area in the Jonggol District area, Bogor Regency, developed by the Ciputra Group. Within the Vignolia Hill Cluster there is a mosque located on the northwest corner of the Vignolia Hill cluster or at the western end of the AH.17 block. The large number of blocks raises problems regarding the shortest route that can be taken by residents to get to the mosque. So, the purpose of this study is to determine the shortest distance traveled by citizens to get to the mosque. The method used is to apply the Dijkstra algorithm which is able to produce the shortest route for residents to get to the mosque.

Keywords: Dijkstra's Algorithm, Shortest Path

INTRODUCTION

The application of Artificial Intelligence to problem-solving in the field of computer science has experienced rapid development from year to year along with the development of artificial intelligence itself. The problem involving searching is one example of the use of artificial intelligence which is quite popular to solve various problems. The application varies, starting from real-world problems, such as determining the route on a map, traveling salesman problem (TSP), determining the sequence of assembly (assembly sequencing) by robots, to the application in the game world, such as making computers able to play chess like humans or determining to take the path of character in a game (Budihartono, 2016).

In daily activities, the use of roads is always an unavoidable activity. In the process of moving from the initial location to the destination location, passing several roads. Roads are transportation facilities in which each route is a complex network (Hannats et al., 2012). In graph
theory, determining the optimal path from one point to another is a problem that is often encountered in everyday life (Rifanti, 2017). This is also felt by residents who live in a large enough housing. Sometimes to get to the destination quickly then the shortest distance from the initial location must be found to get to the desired destination without going through the road 2 (two) times (Hutapea, 2013).

Citra Indah City Housing is a residential area in the Jonggol District area, Bogor Regency, developed by the Ciputra group. The area of Citra Indah City housing area has now reached 1,200 hectares with 58 clusters. One of the clusters in the Citra Indah City housing complex is the Bukit Vignolia cluster, which has an area of 100,997.87 m² consisting of 30 blocks with 504 housing units. Within the Vignolia Hill Cluster, there is a mosque located on the northwest corner of the Vignolia Hill cluster or at the western end of the AH.17 block. A large number of blocks raise problems regarding the shortest route that can be taken by residents to get to the mosque. The problem of finding the shortest path in a graph is one of the optimization problems (Mardlootillah et al., 2015). In finding the shortest path, the more points and lines on a graph, the more complicated and time-consuming (Nggufron et al., 2019) when searching manually. Therefore, a support program is needed to search for the shortest path in the graph to speed up the search. The program designed is a simulation of the shortest path search on a graph. The simulation is designed to provide convenience and avoid calculation errors for ordinary people who want to implement the shortest path search in daily life.

One of the algorithm methods used in searching the shortest distance is the Dijkstra algorithm. Dijkstra's algorithm is the algorithm most often used in the search for the shortest route, simple to use by using simple nodes on an uncomplicated road network (Primadasa, 2015). The principle of Dijkstra's algorithm is to search for the two smallest paths. Dijkstra's algorithm has an iteration to find the point whose distance from the starting point is the shortest (Ardana & Saputra, 2016). Dijkstra's algorithm is named after its discoverer, Edsger Dijkstra. Dijkstra's algorithm uses the greedy principle, where at each step the side with the minimum weight chosen connects a selected node to another unselected node (Pugas et al., 2011).

The input of this algorithm is a directed and weighted graph, G and a source vertex s in G. V is a set of all vertices in graph G. Each side of this graph is a pair of vertices (u, v) which symbolizes the relationship from vertex u to vertex v. The set of all edges called E. Weights of edges is calculated by the function w: E → [0, ∞]; so w (u, v) is the non-negative distance from vertex u to vertex v. The cost of an edge can be considered as the distance between two vertices, which is the sum of the distances of all edges in the path. For a pair of vertices s and t in V, this algorithm calculates the shortest distance from s to t (Pugas et al., 2011). Dijkstra's algorithm has been widely used in studies to solve shortest path problems, including Sodik Kirono, et. al, using graph models and Dijkstra's algorithm in determining vehicle routes on toll roads (Kirono et al., 2018).

Bilqis Amalyyah uses a combination of nodes based on the Dijkstra algorithm to determine the shortest distance between cities on the island of Java (Amalyyah et al., 2016). Antonio Gusmão, et. al, using Dijkstra's algorithm in designing a Web-based Tourism Geographic Information System for searching the shortest path in Timor Leste with an accuracy of an average distance value of 0.03% of measurements with Google Earth (Gusmão et al., 2013). Based on studies conducted by previous researchers, using the Dijkstra algorithm is the right choice in solving problems in determining the shortest route to the mosque in Citra Indah City housing complex, particularly the Bukit Vignolia cluster.

**MATERIALS AND METHODS**

In this study, the authors used the Siteplan Hill Vignolia Cluster image data from the Citra Indah City residential marketing website as a base map and identification of the block code. Then, at each intersection labeled as a node (v1, v2, v3 ... vn). The path that connects nodes is identified as an edge. The size of each edge is calculated using Google Maps.
The process of determining the shortest distance with the Dijkstra algorithm in this study can be illustrated with the following diagram:

1. Determine the Basic Map
2. Establish Nodes and Labels
3. Measure the distance between nodes
4. Testing the Dijkstra Algorithm
5. Analysis of test results

Source: (Lestari et al., 2020)  
Figure 2 Research Steps

The stages of the experiment carried out by the author in conducting this research were taken in a number of steps below:

1. Determine the Basic Map  
The first step is to determine the basic map as research material. In this study the authors took the Siteplan Hill Vignolia Cluster, Citra Indah Housing contained on the official website of Citra Indah Housing, with the URL address as follows: http://www.citraindah.com/vignolia-55658/134-vignolia-siteplan

2. Establish Nodes and Labels  
The second step is to assign nodes and label each node. Based on a predetermined base map, which is considered as a node is every housing intersection, and given the label of each node.

3. Measure the distance between nodes  
The next process is to measure the distance between nodes, with the following steps:
   a. Set the distance of all nodes as "infinite", then set the distance of the initial node (source) to the value 0.
   b. Mark the distance of the initial node as a permanent value, while the distance of the other nodes as a temporary value.
   c. Sets the initial node as the active node.
   d. Calculate the temporary distance of all the nodes that are neighbors of the active node by adding them to the length of the edge.
   e. If the distance calculation result is smaller than the temporary value, an update of the distance value at that node is performed.
   f. Set the node with the smallest temporary distance value as the active node and mark the distance value on that node as a permanent value.
   g. Repeat steps 4 through 7 until there are no more nodes with permanent values that have neighbors with temporary distance values.

4. Dijkstra Algorithm Testing  
The fifth step is to apply the Dijkstra algorithm in this study, which is to make a simple program using the PHP programming language. The steps that have been described in the third process, are applied in the Dijkstra algorithm, where the pseudocode can be written as follows:

```
function Dijkstra(Graph, source):
    // Initialization
    for each vertex v in Graph:
        // initial distance from source to vertex v is set to infinite
        dist[v] := infinity
        // Previous node in optimal path from source
        previous[v] := undefined
        // Distance from source to source
        dist[source] := 0
    // all nodes in the graph are unoptimized - thus are in Q
    Q := the set of all nodes in Graph
    // main loop
    while Q is not empty:
        u := node in Q with smallest dist[ ]
        remove u from Q
        // where v has not yet been removed from Q.
        for each neighbor v of u:
            alt := dist[u] + dist_between(u, v)
            if alt < dist[v]
                // Relax (u,v)
                dist[v] := alt
                previous[v] := u
        return previous[ ]
```

In the pseudocode, there is a "while" function that has $O(n)$ time complexity, and in the "while" function there are other functions that have $O(n)$ time complexity. So overall Dijkstra's algorithm has a time complexity of $O(n^2)$.

5. Analysis of test results  
After all, processes have been carried out, the final step is to analyze the results of the shortest route measurement test using the Dijkstra algorithm, and draw conclusions from the results of the analysis.

**RESULTS AND DISCUSSION**

**Nodes / Vertex Identification Process**

Based on the image of the Bukit Vignolia cluster plan obtained, a node is determined and labeled for
each node. From the identification process, 40 nodes were obtained.

Source: (Lestari et al., 2020)

Figure 3. Identification of Nodes / Vertex

**Distance Measurement Process between Nodes**

Measurement of the distance to each node is done using Google Maps so that data obtained from the node distance as follows:

| Node | Jarak (meter) |
|------|--------------|
| v8   | v10 40       |
| v9   | v11 40       |
| v10  | v11 80       |
| v10  | v14 40       |
| v11  | v12 40       |
| v12  | v13 40       |
| v12  | v18 60       |
| v13  | v14 40       |
| v13  | v16 60       |
| v14  | v15 60       |
| v15  | v16 40       |
| v15  | v20 40       |
| v17  | v18 20       |

Source: (Lestari et al., 2020)

**Test result**

To do the testing, a simple application with the PHP programming language was created. Tests were carried out on 12 nodes selected by dividing the graph into 4 quadrants, then from each quadrant, 3 random nodes were taken.

Source: (Lestari et al., 2020)

Figure 4 Quadrant Distribution
The selected nodes to be tested are as follows:

| No | Kuadran | Node |
|----|---------|------|
| 1  | I       | V12  |
| 2  | I       | V14  |
| 3  | I       | V16  |
| 4  | II      | V25  |
| 5  | II      | V30  |
| 6  | II      | V27  |
| 7  | III     | V34  |
| 8  | III     | V37  |
| 9  | III     | V33  |
| 10 | IV      | V4   |
| 11 | IV      | V8   |
| 12 | IV      | V11  |

Source: (Lestari et al., 2020)

The results of testing the 12 nodes to go to the mosque located at node v19 are as follows:

| No | Node Awal | Jarak (meter) | Rute |
|----|-----------|---------------|------|
| 1  | V12       | 120           | V12-V18-V17-V19 |
| 2  | V14       | 160           | V14-V15-V20-V19 |
| 3  | V16       | 60            | V16-V17-V19    |
| 4  | V25       | 260           | V25-V25-V21-V20-V19 |
| 5  | V30       | 320           | V30-V25-V24-V23-V21-V20-V19 |
| 6  | V27       | 180           | V27-V23-V21-V20-V19 |
| 7  | V34       | 400           | V34-V32-V22-V10-V14-V15-V20-V19 |
| 8  | V37       | 400           | V37-V36-V33-V22-V10-V14-V15-V20-V19 |
| 9  | V33       | 320           | V33-V32-V22-V10-V14-V15-V20-V19 |
| 10 | V4        | 280           | V4-V7-V9-V11-V12-V18-V17-V19 |
| 11 | V8        | 240           | V8-V10-V14-V15-V20-V19 |
| 12 | V11       | 160           | V11-V12-V18-V17-V19 |

Source: (Lestari et al., 2020)

**CONCLUSION**

Based on the methodology and testing that has been done, it can be concluded that the Dijkstra algorithm can help residents of the Bukit Vignolia Cluster, Citra Indah to determine the nearest route to the mosque. The results of this study can still be developed by adding the estimated travel time needed so that it can provide suggestions for residents to go to the mosque in order to arrive at the mosque on time for prayer.

**REFERENSI**

Amaliah, B., Fatichah, C., & Riptianingdyah, O. (2016). FINDING THE SHORTEST PATHS AMONG CITIES IN JAVA ISLAND USING NODE COMBINATION BASED ON DIJKSTRA ALGORITHM. *International Journal on Smart Sensing and Intelligent Systems*, 9(4), 2219–2236.

Ardana, D., & Saputra, R. (2016). *Penerapan Algoritma Dijkstra pada Aplikasi Pencarian Rute Bus Trans Semarang*. Snik, 299–306.

Budihartono, E. (2016). PENERAPAN ALGORITMA DIJKSTRA UNTUK SISTEM PENDUKUNG KEPUTUSAN BAGI PENENTUAN JALUR TERPENDEK PENGIRIMAN PAKET BARANG PADA TRAVEL. *SENIT*, 69–78.

Gusmão, A., Pramono, S. H., & Sunaryo. (2013). Sistem Informasi Geografis Pariwisata Berbasis Web Dan Pencarian Jalur Terpendek Dengan Algoritma Dijkstra. *Jurnal EECIS*, 7(2), 125–130. https://jurnaleccis.ub.ac.id/index.php/eeccis/article/view/214

Hannats, M., Ichsan, H., Yudaningtyas, E., & Muslim, M. A. (2012). Solusi Optimal Pencarian Jalur Tercepat dengan Algoritma Hybrid Fuzzy-Dijkstra. *Jurnal EECIS*, 6(2), 155–160. https://jurnaleccis.ub.ac.id/index.php/eeccis/article/view/185

Hutapea, M. I. (2013). IMPLEMENTASI ALGORITMA GREEDY UNTUK MENCARI RUTE TERPENDEK ANTAR KOTA. *MAJALAH ILMIAH METHODA*, 3(3), 23–30. http://www.ojs.ippmethodistmedan.net/index.php/METHO DA/article/view/68

Kirono, S., Arifianto, M. I., Putra, R. E., & Musoleh, A. (2018). GRAPH-BASED MODELING AND DIJKSTRA ALGORITHM FOR SEARCHING VEHICLE ROUTES ON HIGHWAYS. *International Journal on Smart Sensing and Intelligent Systems* (IJMET), 9(8), 1273–1280.

Lestari, S., Ardiensyah, Giovani, A. P., & Jayanti, D. D. (2020). *Independent Research Final Report*.

Mardlootillah, H. I., Suyitno, A., & Arini, F. Y. (2015). SIMULASI ALGORITMA DIJKSTRA DALAM MENANGANI MASALAH LINTASAN TERPENDEK PADA GRAF MENGGUNAKAN VISUAL BASIC. *Unnes Journal of Mathematics*,...
Ngufiron, N., Rochmad, R., & Mashuri, M. (2019). Pencarian Rute Terbaik Pemadam Kebakaran Kota Semarang Menggunakan Algoritma Dijkstra dengan Logika Fuzzy sebagai Penentu Bobot pada Graf. *Unnes Journal of Mathematics*, 8(1), 40–49. https://journal.unnes.ac.id/sju/index.php/ujm/article/view/19461/

Primadasa, Y. (2015). PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA DIJKSTRA PADA SIG BERBASIS WEB UNTUK DISTRIBUSI MINUMAN (STUDI KASUS PT. COCA-COLA KOTA PADANG. *Jurnal KomTekInfo Fakultas Ilmu Komputer*, 2(2), 47–54.

Pugas, D. O., Somantri, M., Satoto, K. I., & Nilsson, N. (2011). Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra dan Astar (A *) pada SIG Berbasis Web untuk Pemetaan Pariwisata Kota Sawahlunto. *13*(1), 27–32.

Rifanti, U. M. (2017). PEMILIHAN RUTE TERBAIK MENGGUNAKAN ALGORITMA DIJKSTRA UNTUK MENGURANGI KEMACETAN LALU LINTAS DI PURWOKERTO. *Jurnal Matematika Dan Pendidikan Matematika*, 2(2), 90–99.