J. Sci. Inform. Soc. 2023, Vol. 1, No. 2
ISSN 3024-9074

Graph Model for Traffic Control in Surakarta

Rangga Ganzar Noegraha[1]*, Haris Nur Kuncoro[2], Tasmi[2], Ariana Yunita[2]

Department of Computer Science, Universitas Pertamina, Jakarta 12220, Indonesia
*Corresponding Author: rangga.gn@universitaspertamina.ac.id
Received: 26 January 2023
Accepted: 14 November 2023

Abstracts

Surakarta City is the most populous city in Central Java that has 64 traffic signals to regulate traffic. Drivers travel time in Surakarta City is heavily influenced by the amount of time they spend waiting for their turn to move at each traffic signals they pass. A time setting for traffic signal changes needs to be designed to optimize each drivers travel time on several routes in Surakarta City. Once the traffic has been researched and modeled, a time setting can be created. This study shows a graph model that represent roadways in Surakarta. The Dijkstra algorithm was used in this study to optimize the routes represented in the graph by finding all the shortest routes between two traffic signals points.

Keywords: Dijkstra, graph, optimization, travel time, traffic

Introduction

According to the Central Bureau of Statistics, Surakarta City is the most populous city in Central Java which has an area of around 44 km^2 with a population density of 12,391 people/km^2 [1]. Based on interviews with Transportation Agency in Surakarta City, the Surakarta City has installed 65 Traffic Signals (in Bahasa it is called “Alat Pemberi Isyarat Lalu Lintas” and acronymized as APILL), with 1 inactive Traffic Signal. When traveling in the city of Surakarta, in general, driver will go through several Traffic Signals. The amount of travel time required by the driver is strongly influenced by the time waiting for their turn to run at each traffic sign that is passed. The timing of the existing APILL causes drivers to often get their turn to stop when they are at a traffic signal.

In traveling, drivers often experience journeys that are not short, because drivers stop at several Traffic Signals. The drivers travel time will be much shorter when the driver always gets green signals. This research was conducted to construct a model of traffic flows in Surakarta city, so that a design of signs change time setting for each traffic signal can be constructed in further research to optimize the travel time of each driver. So that once the routes are modeled, a sign change time setting for each traffic signal can be designed to optimize the travel time of each driver on several routes that a driver may pass.

In modeling traffic flow in Surakarta City, a graph is used which shows the intersection of road routes that can be passed by drivers. Dijkstra algorithm is used to find all the shortest routes between two traffic signals points. Dijkstra algorithm was chosen because it is an effective algorithm in providing the shortest path from one location to another by doing iterations to find the point with the shortest distance from the starting point [2].

Dijkstra Algorithm is a form of Greedy Algorithm. This algorithm includes a graph search algorithm that is used to solve the shortest path problem with one source in a graph that has no negative weight. Dijkstra algorithm finds the shortest path in a number of steps [3]. Dijkstra algorithm works by creating a path to an optimal point at each step. So, in the n-th step, there are at least n points where the shortest path is known [4].

Several previous studies employ Dijkstra to find the shortest route. Rifanti [5] uses Dijkstra Algorithm to find the shortest route from one point to another in Purwokerto City. Rifanti provides recommendations for the shortest route that can be passed by drivers by using the highway or alternative roads. This recommendation is expected to reveal traffic jams in Purwokerto City by preventing drivers from roads that are experiencing congestion.

Other studies [2]–[4], [6] use Dijkstra algorithm as a solution for providing route recommendations for drivers. These five studies use road segments in a city as lines in a weighted graph optimized using Dijkstra Algorithm. In this study, the output of Dijkstra Algorithm is used to design a graph of the past flow model traffic in a city, namely the City of Surakarta. This graph model is used to optimize the travel time of some or all routes in Surakarta City.

Material and Methods

The initial step of this research was data collection which was carried out through an interview process with the Surakarta's Department of Transportation. The next process is to build a graph which is a representation of the traffic lanes in the city of Surakarta. Dijkstra algorithm is implemented in this graph to find the shortest route between two APILL points in Surakarta City. The output of Dijkstra Algorithm is used to build a new graph that represents traffic flow in Surakarta City. This second graph is analyzed to investigate whether or not all routes in Surakarta City can be optimized. The output generated from this research is the optimal path by looking this second graph. The flowchart of research methodology is shown in Figure 1.

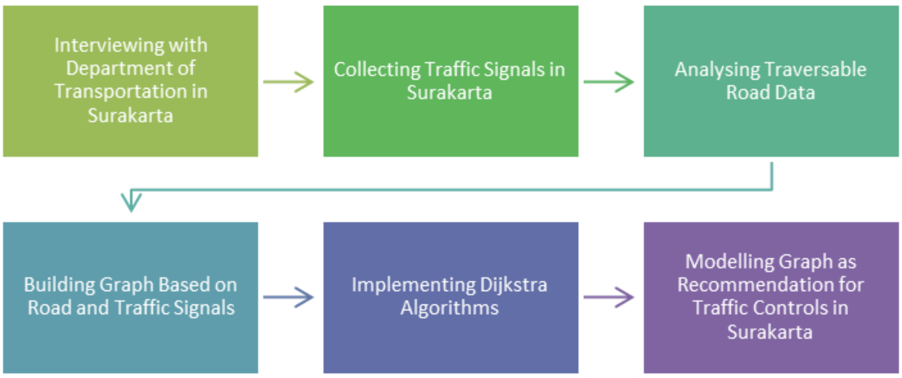

Figure 1. Research Stages

1. Interviews were conducted to find and explore data and information from the Surakarta City Transportation Agency.
2. Data for 64 APILL points spread throughout Surakarta City were obtained from the Surakarta City Transportation Agency, while the coordinates for APILL points were searched using Google Earth and Google Map.
3. Routes that may be passed are roads in the city of Surakarta. This data collection was carried out using the ArcGIS application.
4. APILL coordinates become points on Graph G. Neighborhood of two points on Graph G is determined by the existence of a highway that connects the two APILL points without going through another APILL.

5. Perform Dijkstra Algorithm calculations on Graph G to determine the shortest path between APILL points.
6. Route - the shortest route which is the output of Dijkstra Algorithm is defined as the points on the Traffic Flow Model Graph. While the neighbors of these points are determined by the presence of APILL through which both routes.

Results and Discussion

The location of traffic signals is shown in Figure 2a, while Figure 2b is the road segments in Surakarta City. The lines in bold are the main roads in the city of Surakarta. These road segments are used for modeling in graph form.

(a)

(b)

Figure 2. (a) Traffic Signals or APILL in Surakarta City (b) Surakarta City Roadway

The graph model built on the basis of road segments in Surakarta City is called Graph G. The points on Graph G are the APILL coordinate points in Surakarta City. On Graph G, point u is adjacent to point v if it is possible to travel from u to v using only the highway segment, without going through any other points. Graph G is a directed graph because the point u is adjacent to point v does not always coincide with the point v being adjacent to point u. The length of the road link connecting point u to point v is recorded as the weight for the uv line. With these conditions, the G-directed graph is obtained as follows:

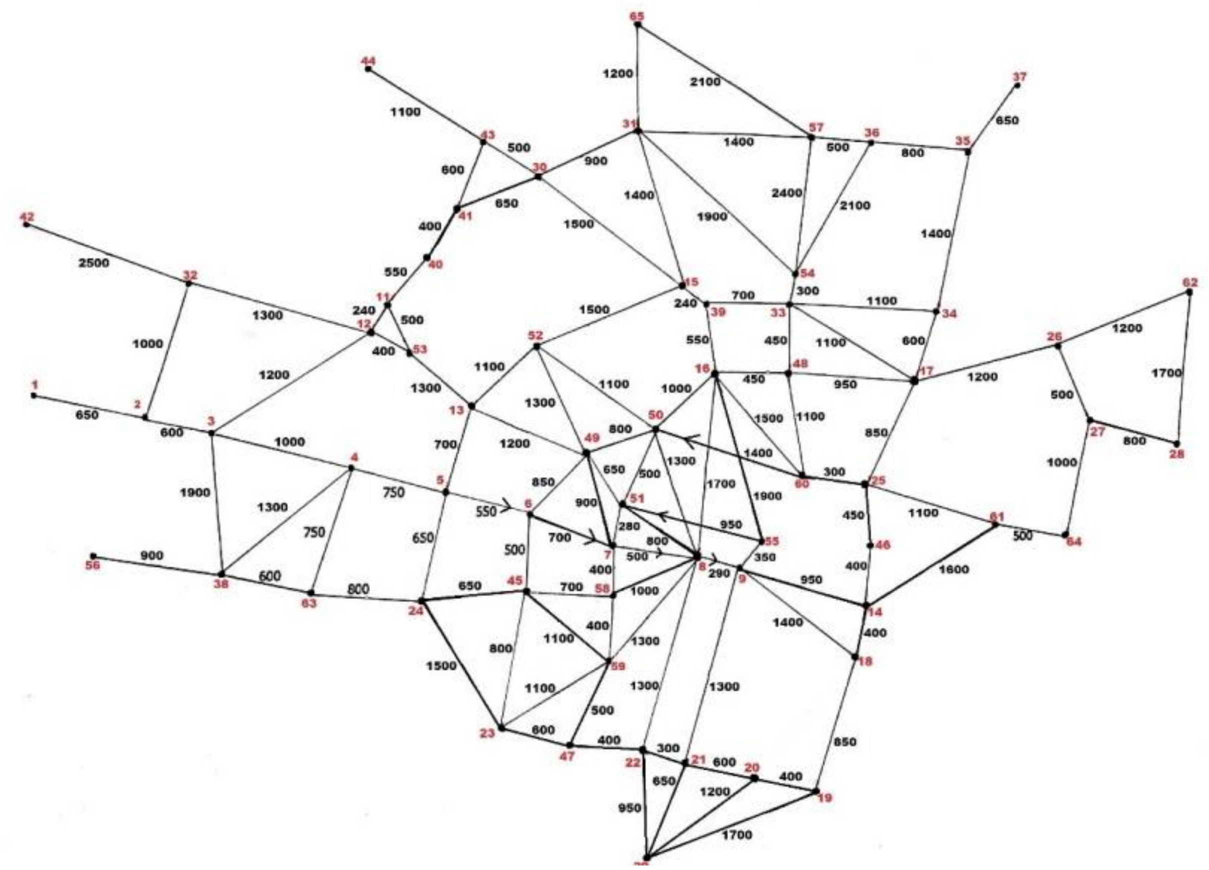

Figure 3. Directed Graph of road segments in Surakarta City

Dijkstra algorithm is used to find the route with the smallest weight for every two points in Graph G. From the output of Dijkstra Algorithm, there are routes that are contained within other routes. These routes are ignored because there are longer routes that represent them. By making this selection, 819 routes were found that do not contain each other, although many do overlap. The Traffic Flow Graph is built based on these 819 routes. In this paper, the Traffic Flow Graph 819 routes are written as Graph H. The points on Graph H are the points on Graph G. In Graph H, point u is next to point v if there is a route that uses road segments u to v. The number of routes through a line is recorded as a weight on that line. With these conditions, the H-weighted directed graph is obtained as follows:

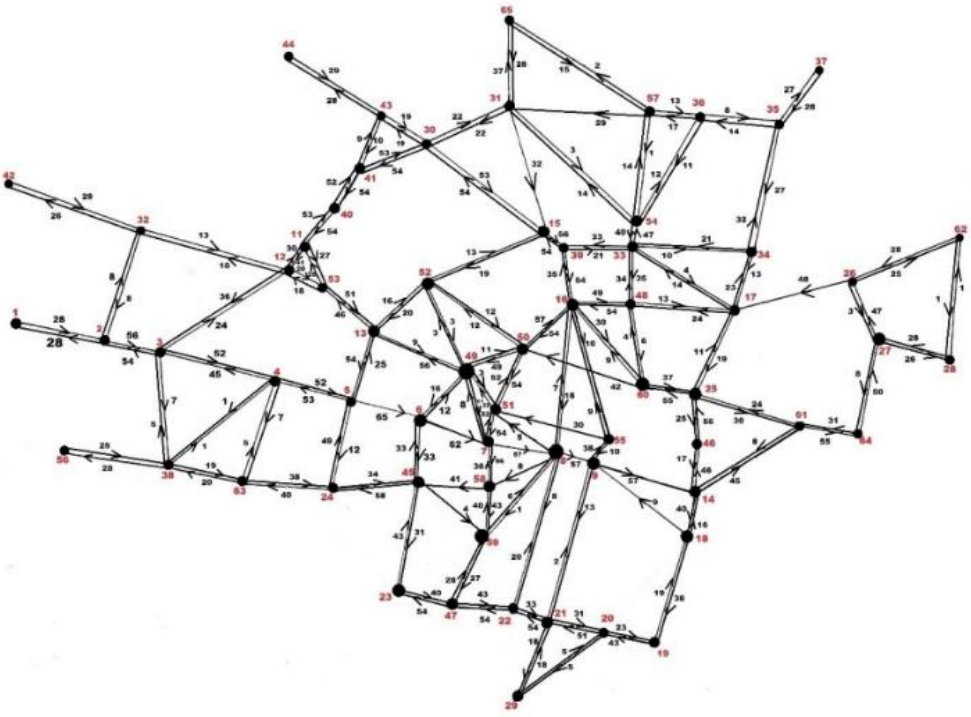

Figure 4. Traffic Flow Models in Surakarta Using A Graph

The graph above shows too many routes that follow the same line. By modeling the road segment that the route passes as a line, the resulting graph has too many edges. For example, there are 56 lines that connect point 2 to point 3. With too many routes passing through the same line, a number of prioritized routes are selected so that optimization of travel time for drivers can be carried out for these selected routes. In selecting prioritized routes, all routes in Surakarta City are first sorted from the longest to the shortest.

Conclusion

In conclusion, this study shows that a graph can be model traffic roadway in Surakarta. This study produces a model of traffic flow for Surakarta City in the form of a graph obtained from the output of Dijkstra Algorithm. The graph shows that there are number of road segments traversed by dozens of routes. In addition, Dijkstra algorithm is used to find the shortest route between any two points in the proposed graph. For future study, time savings can be calculated from a recent traffic flow model and a traffic model that has been optimized using Dijkstra. Further study also should investigate whether the optimization can be done on all routes in the city of Surakarta as well as choose as many routes as possible that can be optimized.

References

[1] BPS, "Provinsi Jawa Tengah Dalam Angka 2021," 2021.

[2] E. C. Galih and R. A. Krisdiawan, "Implementasi algoritma dijkstra pada aplikasi wisata kuningan berbasis android," *Nuansa Inform.*, vol. 12, no. 1, 2018.

[3] M. Qomaruddin, M. T. Alawiy, and S. Sugiono, "Perancangan Aplikasi Penentu Rute Terpendek Perjalanan Wisata di Kabupaten Jember Menggunakan Algoritma Dijkstra," *Sci. Electro*, vol. 6, no. 2, 2018.

[4] S. Sunardi, A. Yudhana, and A. A. Kadim, "Implementasi Algoritma Dijkstra Untuk Analisis Rute Transportasi Umum Transjogja Berbasis Android," *J. Sist. Inf. Bisnis*, vol. 9, no. 1, p. 32, 2019.

[5] U. M. Rifanti, "Pemilihan rute terbaik menggunakan Algoritma dijkstra untuk mengurangi Kemacetan lalu lintas di purwokerto," *JMPM J. Mat. dan Pendidik. Mat.*, vol. 2, no. 2, pp. 90–99, 2017.

[6] K. Anam and O. D. Hartono, "Aplikasi Pemandu Pencarian Wisata Terdekat Berbasis GIS Android Dengan Algoritma Dijkstra," *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 3, no. 1, pp. 91–99, 2019.