Implementasi Algoritma Dijkstra untuk Menentukan Rute Terpendek Menuju Pelayanan Kesehatan

Ika Arthalia Wulandari [(1)] Pristi Sukmasetyan [(2)]

Program Studi S1 Ilmu Komputer, Universitas Muhammadiyah Metro. Jl. Ki Hajar Dewantara No.116 Iringmulyo, Metro Timur, Kota Metro. Lampung [(1)]
Universitas Muhammadiyah Magelang, Jl. Mayjen Bambang Soegeng, Glagak, Sumberrejo, Kec. Mertoyudan, Kabupaten Magelang, Jawa Tengah 56172 [(2)]

ikaarthalia@gmail.com

Abstrak

Rute terpendek dari suatu perjalanan akan mempersingkat waktu tempuh. Begitu juga dalam hal pencarian tenaga ahli. Saat meminta rute dari satu titik (titik awal) ke lokasi lain (titik tujuan), biasanya hasil yang keluar adalah "jalur terpendek" dari titik awal ke titik tujuan. Jalur terpendek adalah masalah untuk menemukan jalur antara dua atau lebih simpul dalam graf berbobot minimum. Untuk mempermudah penyelesaian masalah jalur terpendek, maka diperlukan algoritma pencarian. Algoritma Dijkstra memecahkan masalah pencarian jalur terpendek antara dua simpul dalam graf berbobot dengan jumlah total terkecil, dengan mencari jarak terpendek antara simpul awal dan simpul lainnya, sehingga jalur yang terbentuk dari simpul awal ke simpul tujuan memiliki jumlah bobot terkecil. Pada penelitian ini, algoritma Dijkstra mencari jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya, sehingga dapat membantu memberikan pilihan jalur. Berdasarkan uji coba algoritma Dijkstra memiliki kemampuan untuk mencari jalur terpendek, karena pada algoritma tersebut setiap graf dipilih sisi dengan bobot minimum yang menghubungkan simpul terpilih dengan simpul lain yang belum terpilih.

Kata Kunci :Algoritma Dijkstra, Rute Terpendek.

ABSTRACT

The shortest journey route will shorten the travel time, likewise in terms of the search for experts. When requesting a route from one point (start point) to another location (destination point), it usually returns the "shortest path" from the starting point to the destination point. The shortest path is the problem of finding the path between two or more vertices in a minimum weighted graph. A search algorithm is needed to facilitate solving the shortest path problem. Dijkstra's algorithm solves the problem of finding the shortest path between two vertices in a weighted graph with the smallest total number by finding the shortest distance between the initial vertices and other vertices. The path formed from the initial to the destination vertex has the smallest number of weights. In this study, Dijkstra's algorithm looks for the shortest path based on the most negligible weight from one point to another so that it can help provide path choices. Based on the test of Dijkstra's algorithm, it can find the shortest path because the selected node is the node with the minimum weight connected to other nodes that have not been selected.

Keywords: *: Dijkstra's Algorithm, TheShortest path*

PENDAHULUAN

Teknologi telah banyak membantu kami, mulai dari mengirim email, pesan teks, media sosial, dan bahkan peta. Orang menggunakan peta untuk banyak hal, misalnya mencari tempat seperti mencari restoran terdekat, kafe, pom bensin, atau mencari arah dari rumah ke kantor. Saat meminta rute dari satu titik (titik awal) ke lokasi lain (titik tujuan), biasanya hasil yang keluar adalah "jalur terpendek" dari titik awal ke titik tujuan [1], [2]. Masalah penentuan jalur terpendek saat ini sangat menarik untuk dibahas karena sangat berkaitan dengan kehidupan sehari-hari. Untuk mempermudah penyelesaian masalah jalur terpendek, maka diperlukan algoritma pencarian.

Algoritma adalah urutan pemecahan masalah yang disusun secara sistematis dan logis untuk dipecahkan masalah. Algoritma yang biasa digunakan dalam menyelesaikan masalah pencarian jalur atau jalur terpendek adalah algoritma Dijkstra. Algoritma Dijkstra memecahkan masalah pencarian jalur terpendek antara dua simpul dalam graf berbobot dengan jumlah total terkecil, dengan mencari jarak terpendek antara simpul awal dan simpul lainnya, sehingga jalur yang terbentuk dari simpul awal ke simpul tujuan memiliki jumlah bobot terkecil [1]. Masalah ini terkait dengan pencarian rute terpendek dan biaya yang dianggarkan minimum. Jalur terpendek (*shortest path*) adalah masalah untuk menemukan jalur antara dua atau lebih simpul dalam graf berbobot yang bobot sisi gabungan dari graf yang dilalui adalah minimum [2]–[4].

Algoritma Dijkstra merupakan algoritma greedy yang biasa digunakan untuk pencarian jarak terpendek dimana masukan dari algoritma ini adalah graf berarah berbobot dengan titik asal dari sekumpulan garis [4]–[6]. Algoritma Dijkstra memiliki kemampuan yang efektif untuk mencari jalur terpendek, dimana pada setiap graf dipilih sisi dengan bobot minimum yang menghubungkan suatu simpul yang telah dipilih dengan simpul lain yang belum terpilih [7]. Cara kerja algoritma ini adalah dengan mengunjungi semua node dan membuat jaraknya. Jika terdapat dua jarak pada node yang sama, maka dipilih jarak dengan bobot terendah, sehingga semua node memiliki jarak yang optimal dan pencarian ini dilakukan sampai node tujuan ditemukan. Dengan kata lain, algoritma Dijkstra menghitung jalur berdasarkan jarak terpendek yang ditempuh dalam sebuah kota.[6][8].

KAJIAN PUSTAKA DAN LANDASAN TEORI

Algoritma Djikstra

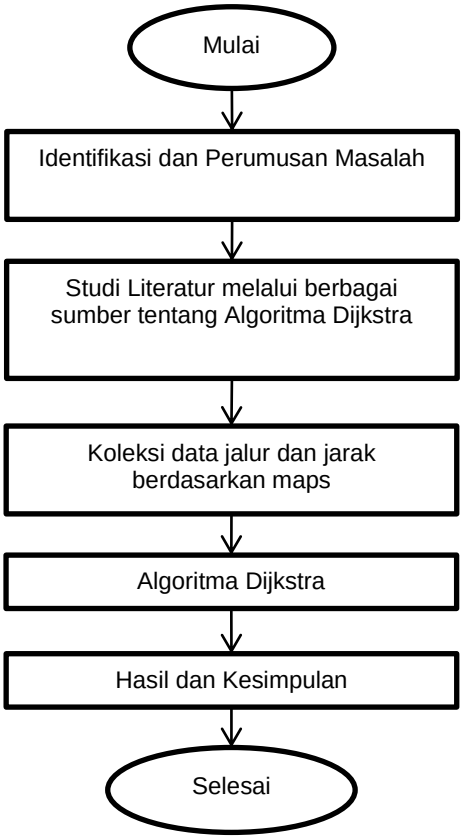
Algoritma Dijkstra merupakan algoritma *greedy* yang digunakan dalam menyelesaikan masalah jarak terpendek untuk graf berarah dengan bobot sisi non-negatif [9]. Cara kerja algoritma Dijkstra dalam mencari jarak terpendek adalah perhitungan dari titik asal ke titik terdekat, kemudian ke titik kedua, dan seterusnya. Ide dasar dari algoritma Dijkstra sendiri adalah untuk mencari nilai biaya yang paling dekat dengan tujuan yang berfungsi dalam graf berbobot, sehingga dapat membantu memberikan pilihan jalur. Misalkan titik mewakili bangunan dan garis mewakili jalan, algoritma Dijkstra menghitung semua bobot terkecil yang mungkin dari setiap titik. Secara garis besar algoritma ini bertujuan untuk mencari jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya[3], [6]. Kelebihan algoritma dijkstra adalah sebagai berikut [3], [9], [10].

1. Algortima Dijkstra merupakan jenis algoritma *Single Source Shortest*

2. Cara kerja Algortima Dijkstra menggunakan *Priority Queue*
3. Pemrograman menggunakan pemrograman Greedy
4. Bobot simpul untuk penentuan jarak terpendek graf berarah dengan bobot positif atau sisi non-negatif.

METODE

Metode penelitian merupakan suatu langkah yang dilakukan untuk mendapatkan solusi dari segala permasalahan. Pada masalah lintasan terpendek, untuk menentukan nilai optimum, diperlukan variabel-variabel seperti nilai bobot pada edge, jumlah simpul yang dilalui, bobot total semua simpul aktif dan kecepatan dalam notasi perhitungan. Hal ini sangat penting sebagai dasar pengambilan kesimpulan untuk mendapatkan keputusan yang optimum dalam menyelesaikan masalah jalur terpendek [5]. Dalam melakukan penelitian juga diperlukan adanya data mentah untuk dijadikan dasar pengujian. Untuk mendapatkan hasil tersebut perlu dilakukan pengolahan data terlebih dahulu. Tahap awal adalah mendesain graf, kemudian menentukan bobot masing – masing simpul berdasarkan jarak. Prosedur yang digunakan dalam penelitian ini dirangkum dalam Gambar 1.

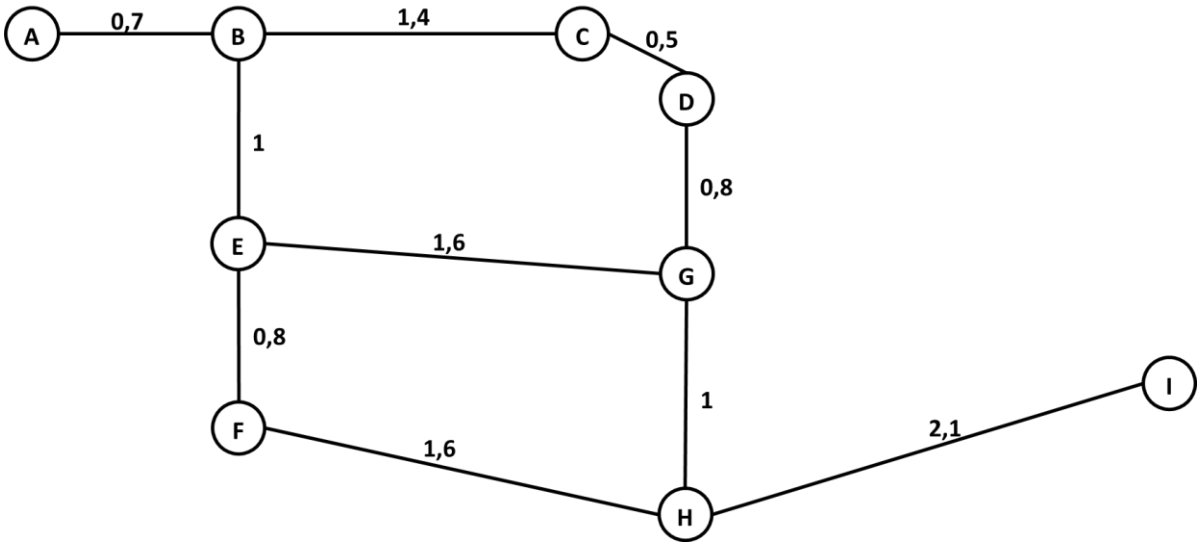


Gambar 1. Prosedur Penelitian

HASIL DAN PEMBAHASAN

A. Koleksi Data

Dalam implementasinya, penelitian ini menggunakan contoh dari Kampus 1 Universitas Muhammadiyah Metro (titik awal) yang disimbolkan dengan A menuju ke Rumah Sakit Umum (RSU) Muhammadiyah Metro sebagai pelayanan kesehatan (titik tujuan) yang disimbolkan dengan I. Antara titik awal sampai titik tujuan, secara manual dihasilkan sebanyak 9 titik atau node (simpul) yang merupakan persimpangan 2 jalan atau lebih. Nilai bobot ditentukan berdasarkan jarak yang ditunjukkan pada Peta *Google Maps* seperti ditunjukkan pada Gambar 2.



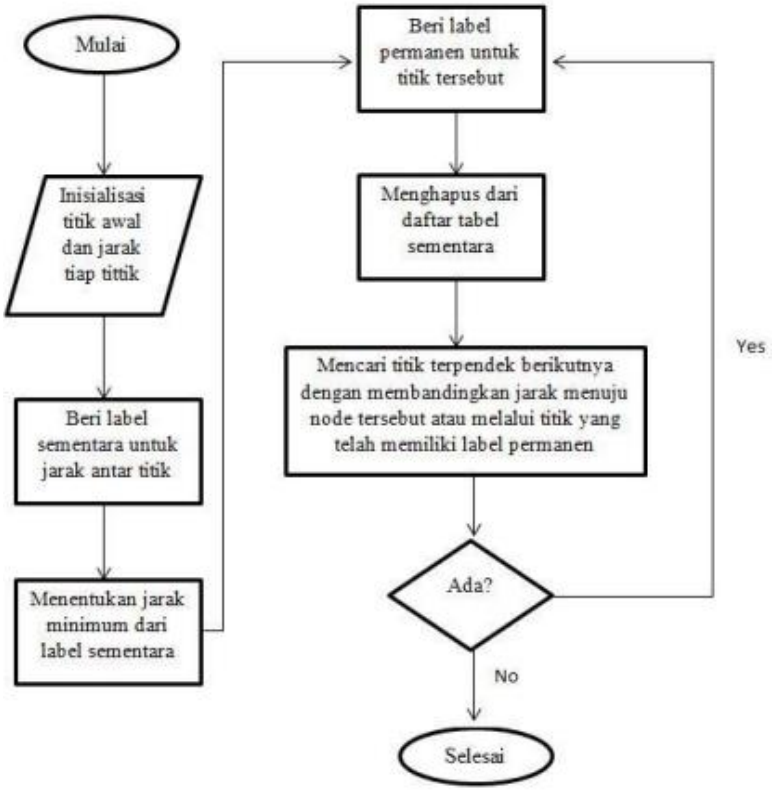
Gambar 2. Graf dengan 9 node dan 10 *edge*

Dimana.

Poin A : Posisi user di Universitas Muhammadiyah Metro (titik awal)
Poin B : Persimpangan Jalan Ki Hajar Dewantara - Jalan Mulia 1
Poin C : Persimpangan Jalan Ki Hajar Dewantara – Jalan Kunang
Poin D : Persimpangan Jalan Kunang – Jalan Letjen Alamsyah Ratu Prawira Negara
Poin E : Persimpangan Jalan Mulia 1 – Jalan Ahmad Yani
Poin F : Persimpangan Jalan Ahmad Yani – Jalan Sultan Sahrir
Poin G : Jalan Letjen Alamsyah Ratu Prawira Negara
Poin H : Persimpangan Jalan Letjen Alamsyah Ratu Prawira Negara - Jalan Sultan Sahrir – Jalan Soekarno Hatta
Poin I : Posisi alamat Pelayanan Kesehatan di RSU Muhammadiyah Metro (titik tujuan)

B. Implementasi Algoritm Djikstra

Pencarian algoritma Dijkstra berfokus pada pencarian jalur dengan *cost* terkecil antara satu titik dengan titik lainnya. Hasil akhir dari algroitma adalah mencari jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Proses algoritma dijkstra dapat dijelaskan secara singkat pada Gambar flowchat 3.



Gambar 3. Flowchart algoritma Dijkstra [5]

Pada studi kasus dalam penyelesaian algoritma Dijkstra yang telah ditentukan pada Gambar 2 penyelesaian algoritma Dijkstra terdapat 9 simpul untuk mendapatkan jarak terdekat untuk mencapai tujuan seperti pada Tabel 1.

Tabel 1. Jarak antara simpul yang terhubung

dari ↔ ke		jarak
A	B	0,7
B	C	1,4
B	E	1
C	D	0,5
D	F	0,8
E	G	0,8
E	F	1,6
F	H	1
G	H	1,6
H	I	2,1

Tabel 2 merupakan langkah perhitungan dari algoritma dijkstra untuk menentukan jarak terdekat dari titik awal (A) ke titik tujuan (I). Untuk setiap langkah dari simpul terpilih, jumlahkan angka terkecil sebelumnya dengan semua jarak dari simpul terpilih ke simpul terhubung yang kotaknya tidak berwarna biru, seperti penjelasan sebagai berikut.

Langkah 1. Perhitungan dimulai dari titik awal (Simpul A). Simpul A hanya terhubung dengan simpul B dengan jarak 0,7, sehingga jarak terpendek yang terpilih ada dari simpul **A ke B.**

Langkah 2. Perhitungan dimulai dari titik yang memiliki nilai terpendek pada langkah 1, yaitu simpul B. Simpul B terhubung dengan dua simpul yaitu simpul C dengan jumlah jarak 2,1 dan simpul E dengan jumlah jarak 1,7. Jumlah jarak diperoleh dengan menjumlahkan jarak simpul A ke simpul B dengan jarak simpul B ke simpul terhubung (simpul C dan E). Jarak terpendek dari langkah dua adalah jarak simpul B ke simpul E sehingga simpul terpilih untuk langkah kedua adalah **A ke B ke E.**

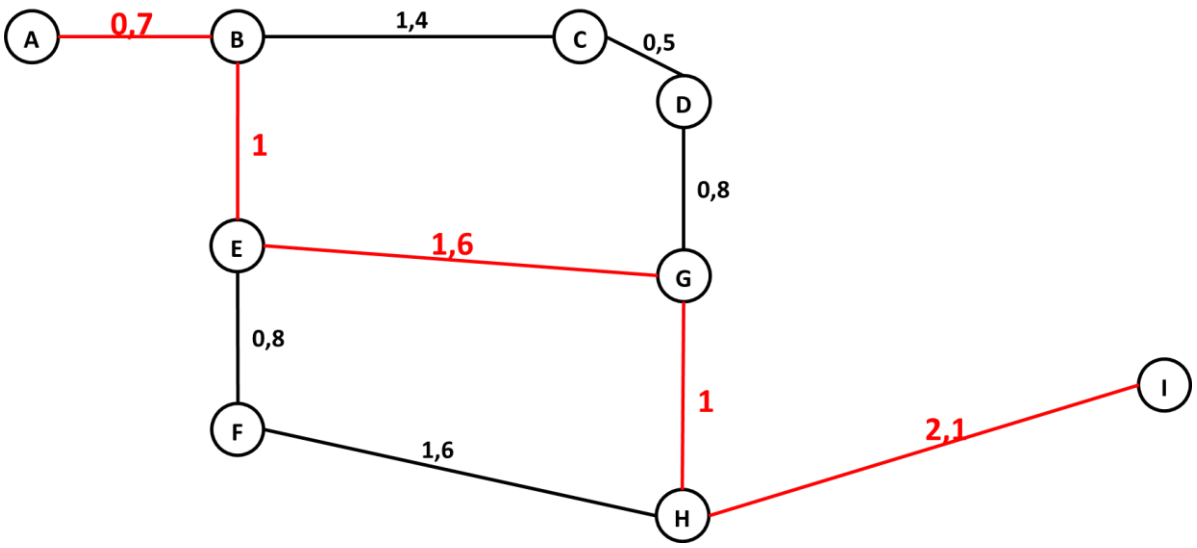
Langkah 3. Perhitungan dimulai dari titik yang memiliki nilai terpendek pada langkah 2 dan belum terpilih, yaitu simpul E. Simpul E terhubung dengan tiga simpul yaitu simpul B, G dengan jumlah jarak 2,5 dan simpul F dengan jumlah jarak 3,3. Jumlah jarak diperoleh dengan menjumlahkan jarak simpul A ke simpul B, simpul B ke simpul E dan simpul E ke simpul terhubung (simpul B, G dan F). Jarak terpendek dari langkah tiga adalah jarak simpul E ke simpul G yaitu sehingga simpul terpilih untuk langkah ketiga adalah **A ke B ke E ke G.**

Langkah 4. Perhitungan dimulai dari titik yang memiliki nilai terpendek pada langkah 3 dan belum terpilih, yaitu simpul G. Simpul G terhubung dengan tiga simpul yaitu simpul D, E, dan simpul H dengan jumlah jarak 4,1. Jumlah jarak diperoleh dengan menjumlahkan jarak simpul A ke simpul B, simpul B ke simpul E, simpul E ke simpul G dan simpul G ke simpul terhubung (simpul D, E dan H). Jarak terpendek dari langkah empat adalah jarak simpul G ke simpul H yaitu sehingga simpul terpilih untuk langkah keempat adalah **A ke B ke E ke G. ke H.** Dan langkah berikutnya sampai dengan tidak ada lagi simpul yang dapat terpilih

Tabel 2. Perhitungan Algoritma Dijkstra

Langkah	Simpul	A	B	C	D	E	F	G	H	I	Jarak Terpendek Untuk Simpul langkah berikutnya	
1	A	0	0,7	~	~	~	~	~	~	~	0,7	B
2	B	0	0,7	2,1	~	1,7	~	~	~	~	1,7	E
3	E	0	0,7	2,1	~	1,7	3,3	2,5	~	~	2,1	C
4	C	0	0,7	2,1	2,6	1,7	3,3	2,5	~	~	2,5	G
5	G	0	0,7	2,1	2,6	1,7	3,3	2,5	4,1	~	2,6	D
6	D	0	0,7	2,1	2,6	1,7	3,3	2,5	4,1	~	3,3	F
7	F	0	0,7	2,1	2,6	1,7	3,3	2,5	4,1	~	4,1	H
8	H	0	0,7	2,1	2,6	1,7	3,3	2,5	4,1	6,2	6,2	I
9	I	0	0,7	2,1	2,6	1,7	3,3	2,5	4,1	6,2	6,2	
	stop	A	A	B	C	B	E	E	G	H	0,7	

Dari hasil perhitungan algoritma dijkstra, diperoleh bahwa jarak lintasan terpendek adalah dimulai dari simpul A – B – E – G – H – I dengan jumlah jarak adalah 6,2 km seperti ditunjukkan pada Gambar 4.



Gambar 4. Graph dengan jalur lintasan terpendek

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan terhadap algoritma Dijkstra dalam menentukan jalur terpendek (*shortest path*), menunjukkan bahwa algoritma Dijkstra menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Langkah-langkah yang dilakukan dalam algoritma Dijkstra dimulai dari penentuan titik awal, kemudian pembobotan jarak dari node pertama ke node terdekat satu per satu, algoritma Dijkstra akan mengembangkan pencarian dari satu titik ke titik lain dan ke titik berikutnya secara bertahap. Berdasarkan uji coba menggunakan algoritma Dijkstra, peneliti mengambil sampel uji yang memiliki lokasi tujuan berbeda, hasil pengujian pada aplikasi dapat menampilkan tujuan

jalur dari titik koordinat posisi pengguna. Algoritma Dijkstra memiliki kemampuan yang efektif untuk mencari jalur terpendek, karena dalam algoritma tersebut setiap graf dipilih dengan bobot minimum yang menghubungkan sebuah node terpilih dengan node lain yang tidak terpilih..

REFERENSI

[1] R. D. Gunawan and R. Napianto, “Implementation of Dijkstra ’ S Algorithm in Determining the Shortest Path (Case Study : Specialist Doctor Search in Bandar Lampung),” *Int. J. Inf. Syst. Comput. Sci.*, vol. 3, no. 3, pp. 98–106, 2019.

[2] D. Rachmawati and L. Gustin, “Analysis of Dijkstra’s Algorithm and A∗ Algorithm in Shortest Path Problem,” *J. Phys. Conf. Ser.*, vol. 1566, no. 1, 2020, doi: 10.1088/1742-6596/1566/1/012061.

[3] D. B. Johnson, “A Note on Dijkstra’s Shortest Path Algorithm,” *J. ACM*, vol. 20, no. 3, pp. 385–388, 1973, doi: DOI:https://doi.org/10.1145/321765.321768.

[4] N. Makariye, “Towards shortest path computation using Dijkstra algorithm,” in *2017 International Conference on IoT and Application (ICIOT)*, 2017, pp. 1–3, doi: doi: 10.1109/ICIOTA.2017.8073641.

[5] I. P. Sari, M. F. Fahroza, M. I. Mufit, and I. F. Qathrunad, “Implementation of Dijkstra’s Algorithm to Determine the Shortest Route in a City,” *J. Comput. Sci. Inf. Technol. Telecommun. Eng.*, vol. 2, no. 1, pp. 134–138, 2021, doi: 10.30596/jcositte.v2i1.6503.

[6] B. Golden, “Technical Note — Shortest-Path Algorithms : A Comparison,” no. March 2022, 1976.

[7] P. Sembiring, A. S. Harahap, and K. S. Zalukhu, “Implementation of Dijkstra’s algorithm to find an effective route to avoid traffic jam on a busy hour,” *J. Phys. Conf. Ser.*, vol. 1116, no. 2, pp. 4–8, 2018, doi: 10.1088/1742-6596/1116/2/022042.

[8] V. N. C. Sebayang and I. Rosyida, “Implementations of Dijkstra Algorithm for Searching the Shortest Route of Ojek Online and a Fuzzy Inference System for Setting the Fare Based on Distance and Difficulty of Terrain (Case Study: in Semarang City, Indonesia),” *Proc. Int. Conf. Math. Geom. Stat. Comput. (IC-MaGeStiC 2021)*, vol. 96, pp. 76–84, 2022, doi: 10.2991/acsr.k.220202.016.

[9] S. Broumi, A. Bakal, M. Talea, F. Smarandache, and L. Vladareanu, “Applying Dijkstra algorithm for solving neutrosophic shortest path problem,” *Int. Conf. Adv. Mechatron. Syst. (ICAMechS*, pp. 412–416, 2016, doi: doi: 10.1109/ICAMechS.2016.7813483.

[10] D. Wahyuningsih and E. Syahreza, “Shortest Path Search Futsal Field Location With Dijkstra Algorithm,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 12, no. 2, p. 161, 2018, doi: 10.22146/ijccs.34513.