PEMILIHAN RUTE TERPENDEK PASIEN UNTUK PENANGANAN COVID – 19 DI JAKARTA MENGGUNAKAN ALGORITMA A*

*SHORTEST ROUTE SELECTION OF PATIENTS FOR HANDLING COVID - 19 IN JAKARTA USING ALGORITHM A**

Ida Bagus Raka Sastrakarmanjata[1], Purba Daru Kusuma[2], Casi Setianingsih[3]

[1,2,3] Universitas Telkom, Bandung
[1] rakasastra@student.telkomuniversity.ac.id, [2] purbodaru@telkomuniversity.ac.id, [3] setiacasie@telkomuniversity.ac.id

Abstrak

Kasus Covid-19 di DKI Jakarta masih terus bertambah, dimana setiap harinya selalu ada penambahan kasus positif Covid-19. Sehingga perlu adanya jalur respon cepat bagi masyarakat untuk menuju ke lokasi-lokasi yang menyediakan rapid test dan swab test di setiap wilayah di DKI Jakarta. Jika pasien semakin banyak, maka dibutuhkan lokasi rumah sakit yang menyediakan penanganan pasien Covid-19. Jika melihat skema lalu lintas di DKI Jakarta saat ini, penjemputan ambulans rumah sakit ke lokasi pasien atau sebaliknya bisa memakan waktu yang cukup lama. dengan kemacetan lalu lintas yang tersebar di wilayah DKI Jakarta. Aplikasi ini dirancang untuk memberikan informasi seperti titik-titik lokasi penanganan Covid-19 seperti Swab Test, tempat isolasi mandiri, dan rujukan rumah sakit. Aplikasi ini menggunakan algoritma A* untuk menentukan rute terpendek yang akan dilalui oleh masyarakat. Dari 30 pengujian didapatkan bahwa semua destinasi dapat dirutekan dengan SAW jika komposisi jarak 77% dan kemacetan 23%. Memori rata-rata yang dibutuhkan untuk menentukan rute menggunakan algoritma pendakian bukit paling curam adalah 84,76 MB dan waktu yang dibutuhkan untuk menentukan rute adalah 2126 milisecond.

Kata Kunci: Covid-19, jarak, aplikasi, A – Star.

Abstract

Covid-19 cases in DKI Jakarta are still increasing, where every day there are always additional positive cases of Covid-19. So, its need for a quick response route for the community to go to locations that provide rapid tests and swab tests in each area in DKI Jakarta. If there are more and more patients, hospital locations that provide handling of Covid-19 patients are needed. If you look at the current traffic scheme in DKI Jakarta, picking up hospital ambulances to the patient's location or vice versa can take quite a long time. with traffic jams scattered in the DKI Jakarta area. This application is designed to provide information such as location points for handling Covid-19 such as Rapid Swab Tests, self-isolation places, and hospital referrals. This application uses the A algorithm to determine the shortest route to be traversed by the community. From 30 tests, it was found that all destinations can be routed with a SAW if composition 77% distance and 23% congestion. The average memory required to determine the route using the steepest ascent hill climbing algorithm is 84.76 MB and the time required to determine the route is 2126 milliseconds.*

Keywords: *Covid-19, distance, application, A – Star*

1. Pendahuluan

Provinsi DKI Jakarta mulai melonggarkan Pembatasan Sosial Berskala Besar (PSBB) dengan jumlah kasus baru virus corona (Covid-19) yang terus meningkat. Berdasarkan data Kementerian Kesehatan (Kemenkes) terdapat penambahan 1.211 kasus baru pada Senin (12/10/2020) sehingga akumulasi menjadi 88.174 orang. Jumlah kematian bertambah 25 orang sehingga total menjadi 1.914 orang. Secara keseluruhan, kasus aktif atau pasien dalam perawatan Covid-19 di DKI Jakarta mencapai 13.720.

Dengan masih masifnya penyebaran Covid-19 di DKI Jakarta, maka perlu adanya jalur respon cepat bagi masyarakat untuk menuju lokasi-lokasi yang menyediakan swab test di setiap wilayah di DKI Jakarta. Jika pasien semakin banyak, maka dibutuhkan lokasi rumah sakit yang menyediakan penanganan pasien Covid-19. Jika melihat skema lalu lintas di DKI Jakarta saat ini, penjemputan ambulans rumah sakit ke lokasi pasien atau sebaliknya bisa memakan waktu yang cukup lama. dengan kemacetan lalu lintas yang tersebar di wilayah DKI Jakarta.

Oleh karena itu, kami ingin membuat sebuah aplikasi yang akan berguna untuk membantu masyarakat atau pasien di wilayah DKI Jakarta untuk menuju lokasi-lokasi strategis seperti tempat-tempat yang menyediakan Rapid

and Swab test dan rumah sakit yang bersedia menangani pasien terinfeksi Covid-19 tanpa harus melewati berbagai titik jalan yang sedang mengalami kemacetan.

2. Dasar Teori

2.1 Lintasan Terpendek

Jalur terpendek (shortest path) adalah jalur yang membutuhkan waktu paling sedikit untuk mencapai node tujuan dari node sumber dalam waktu [10]. Graf yang digunakan dalam pencarian jalur terpendek adalah graf berbobot, yaitu graf yang setiap sisinya diberi nilai atau bobot. Pencarian jalur terpendek (shortest path problem), pada dasarnya merupakan pemecahan masalah untuk menentukan jalur mana yang akan dilalui sehingga ditemukan jalur terpendek dari titik awal hingga tujuan. Dalam mencari jalur terpendek, permasalahan yang dihadapi adalah mencari jalur mana yang akan dilalui sehingga didapatkan jalur terpendek dari satu simpul ke simpul lainnya: [11]

1. Jalur terpendek antara dua simpul tertentu.
2. Jalur terpendek antara semua pasangan simpul.
3. Lintasan terpendek dari suatu simpul tertentu ke semua simpul lainnya.
4. Lintasan terpendek antara dua simpul yang melewati simpul tertentu [11].

2.2 Haversine Formula

Rumus Haversine merupakan persamaan penting dalam navigasi, memberikan jarak lingkaran besar antara dua titik pada bidang bujur dan lintang[12].

$$D = 2 \times sin^{-1}\left(\sqrt{\sin\left(\frac{lat2 - lat1}{2}\right)^2 + \sin\left(\frac{lon2 - lon1}{2}\right)^2 \times \cos lat1 \times \cos lat2}\right) \times \mu E$$

Dengan,

lat1 = Latitude lokasi user
lat2 = Latitude Node
lon1 = Longitude lokasi user
lon2 = Longitude Node
µE = Mean R radius bumi (6,371 km)
D = Jarak antara lokasi user dengan node

2.3 Simple Additive Weighting (SAW)

Metode SAW sering juga dikenal dengan metode penjumlahan berbobot. Konsep dasar dari metode SAW adalah mencari penjumlahan terbobot dari penilaian kinerja pada setiap alternatif pada suatu atribut [13].

$$V_i = \sum_{j=1}^{n} W_j R_{ij}$$

Dengan,

Vi = Nilai Alternative

Wi = Bobot

Rij = Hasil Normalisasi

2.4 Algoritma A – Star

Algoritma A-Star menggunakan metode Best First Search dan mencari jalur optimal dari node awal ke node tujuan [14]. Best First Search Algorithm yang merupakan kombinasi dari Uniform Cost Search dan Greedy-Best First Search. Uniform Cost Search ini akan memilih jarak terkecil dari node awal ke node berikutnya ke node tujuan, sedangkan Greedy-Best First Search yang menggunakan fungsi heuristik akan mengestimasi biaya dari node awal ke node tujuan. Heuristik ini memiliki peran yang sangat penting untuk mengontrol pencarian pada algoritma A*, sehingga algoritma ini akan menemukan rute yang lengkap dan optimal (selalu menemukan solusi jika ada solusi) [15].

Prinsip dari algoritma A* adalah melakukan traversal satu per satu pada setiap node untuk mendapatkan lintasan terpendek. Algoritma ini akan menghitung jarak satu jalur, kemudian menyimpannya dan kemudian menghitung jarak jalur lainnya [15]. Jarak yang dihitung diperoleh dari jarak sebenarnya (actual cost) ditambah dengan perkiraan jarak [16].

$$f(n) = g(n) + h(n)$$

Dimana,

a. f(n) = biaya yang dibutuhkan,

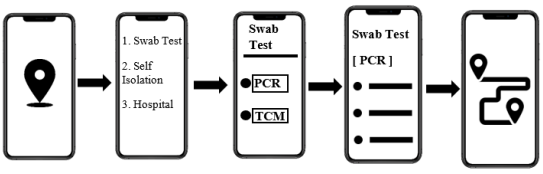
b. g(n) = biaya yang ditempuh dari node asal,

c. h(n) = nilai perkiraan dari node saat ini ke tujuan.

3. Pembahasan

3.1 Gambaran Umum Sistem

Berikut ini adalah gambaran aplikasi pemilihan rute terpendek pasien untuk penanganan Covid-19 di Jakarta. Aplikasi ini akan dipasang di smartphone yang nantinya akan menampilkan daftar pilihan penanganan Covid-19 seperti Rapid Swab Test, tempat isoman dan rumah sakit rujukan. Selanjutnya pengguna akan diarahkan untuk memilih salah satu dari ketiga treatment tersebut, kemudian pengguna akan diperlihatkan rute tujuan terdekat.



Gambar 1. System Overview

Secara umum aplikasi ini berbasis Android yang nantinya ketika pengguna membuka aplikasi ini bertujuan untuk membantu menemukan lokasi penanganan Covid-19 terdekat dari posisi pengguna. Di awal menu utama, pengguna diminta mengaktifkan GPS smartphone-nya dan memilih penanganan Covid-19 seperti gambar di atas. Selanjutnya aplikasi akan melakukan routing dan memberikan jalur terpendek sesuai dengan lokasi pengguna.

3.1 Flowchart Sistem

Setelah pengguna menjalankan aplikasi, pengguna akan diminta untuk mengaktifkan GPS smartphone-nya untuk menentukan node awal. Selanjutnya pengguna akan diminta untuk memilih kebutuhan atau penanganan Covid-19 yang diinginkan seperti terlihat pada gambar. Selanjutnya pengguna akan diberikan daftar pilihan mereka. Kemudian algoritma A – Star akan mencari jalur terpendek dari node awal lokasi pengguna ke node akhir yang ditentukan. Selanjutnya pengguna akan menerima output berupa rute terpendek pada peta.

4. Pengujian Dan Implementasi Sistem

Berdasarkan perancangan antarmuka, selanjutnya melakukan implementasi sistem dan untuk menguji aplikasi yang telah dibuat, dilakukan pengujian berupa pengujian alpha, pengujian beta dan pengujian aplikasi. Implementasinya telah dilakukan sebagai berikut

4.1 Implementasi Sistem

a. Splash Screen

Pada awal memulai aplikasi, layar splash akan muncul selama dua detik sebelum aplikasi masuk ke menu utama.

Gambar 2. Splash Screen

Gambar 2 menampilkan layar splash aplikasi. Layar splash menampilkan logo dan nama aplikasi selama 2 detik.

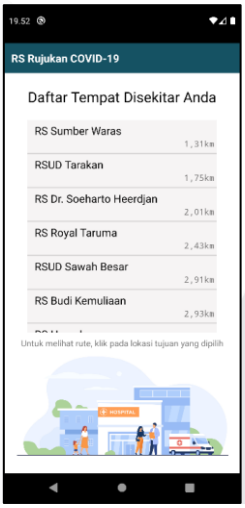
b. Main Menu

Gambar 3. Main Menu

Gambar 3 menunjukkan tampilan menu setelah splash screen selesai. Menu ini berisi tiga menu yang dapat dipilih oleh pengguna yaitu Tes Swab Covid-19 yang akan menampilkan daftar lokasi yang menyediakan Tes Swab Covid-19, menu Tempat Isolasi Mandiri akan menampilkan daftar hotel yang menyediakan Swab Test. fasilitas isolasi, sedangkan menu RS Rujukan Covid - 19 akan menampilkan daftar RS rujukan di DKI Jakarta.

c. Daftar Lokasi

Daftar tempat ditampilkan menggunakan tampilan recycler dimana data berupa nama lokasi didapatkan dari database yang sudah terkoneksi dengan aplikasi.



Gambar 4. Daftar Lokasi

Gambar 4 menunjukkan tampilan daftar tempat sesuai menu yang dipilih oleh pengguna pada menu utama. Pada halaman ini tampilan daftar lokasi dan jarak lokasi dari posisi pengguna sesuai menu yang dipilih pengguna pada menu utama.

d. Konfirmasi Tujuan

Setelah memilih tujuan dari halaman sebelumnya, pengguna akan masuk ke halaman konfirmasi tujuan.

Gambar 5. Konfirmasi Tujuan

Gambar 5 Menampilkan halaman konfirmasi destinasi yang berisi destinasi yang dipilih pada halaman sebelumnya dan himbauan tentang protokol kesehatan.

e. Rute Map

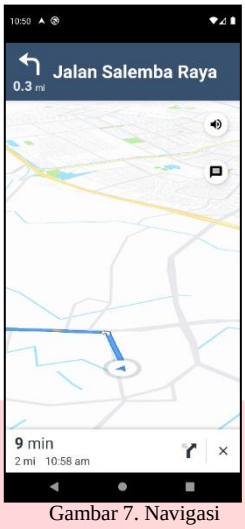
Tampilan peta akan muncul ketika pengguna telah memilih tujuan dan mengkonfirmasi tujuan. Tampilan peta pada aplikasi ini menggunakan Map Box API yang sekaligus juga menampilkan rute yang akan dilalui pengguna.

Gambar 6. Rute Map

Gambar 6 Menampilkan rute yang akan dilalui oleh pengguna mulai dari awal lokasi pengguna sampai dengan lokasi tujuan yang telah dipilih oleh pengguna sesuai dengan perhitungan algoritma.

f. Navigasi

Setelah menampilkan rute, pengguna dapat memulai navigasi dengan mengklik tombol "mulai navigasi".



Gambar 7. Navigasi

Gambar 7 adalah tampilan saat pengguna memulai navigasi. Aplikasi akan memandu pengguna dari awal pengguna berjalan hingga pengguna mencapai tujuan.

4.2 Hasil Pengujian dan Analisis

4.2.1 Skenario Pengujian

Pengujian aplikasi untuk mengetahui berapa banyak resource yang dibutuhkan untuk mencari rute berdasarkan algoritma yang digunakan. Dari pengujian ini, kita dapat mengetahui memori minimum yang diperlukan untuk menjalankan aplikasi ini di ponsel pintar android. Berikut adalah hasil pengujian aplikasi.

Tabel 4.1. Pengujian Aplikasi

Aspek Pengujian	Percobaan ke-					Rata -Rata
	1	2	3	4	5	
Memory (MB)	88,2	83	85,6	90,3	82,7	84,76
Waktu	00:02.186	00:01.603	00:02.068	00:01.836	00:02.370	00:02.126

Dari 5 kali percobaan pencarian rute, hasilnya seperti pada tabel di atas. memory – rata-rata memory yang digunakan untuk proses penentuan rute menggunakan algoritma A – Star adalah 84,76 MB. Sedangkan waktu yang dibutuhkan untuk memproses rute yang dipilih rata-rata adalah 00:02,126.

4.2.2 Pengujian SAW

Uji bobot SAW dilakukan untuk mengetahui rasio bobot terbaik antara jarak dan kemacetan. Pengujian ini menguji apakah bobot yang diterapkan pada kelangkaan dan kemacetan dapat berpengaruh besar pada pemilihan rute yang dipilih. Dalam uji SAW ini misalnya, pada hari Rabu yang berlokasi di kota Jakarta Barat, dekat Jl. Dri (Tss Raya) No.64, dengan koordinat (-6.156718781334539, 106.80523631686208) kemudian memilih tujuan untuk melakukan isolasi mandiri di Luminor Hotel. Berikut ini adalah hasil uji SAW.

Tabel 4.2. Pengujian SAW

No.	Jam	60% jarak + 40% kemacetan	65% jarak + 35% kemacetan	70% jarak + 30% kemacetan	77% jarak + 23% kemacetan
1	8	Berhasil	Berhasil	Berhasil	Berhasil
2	12	Gagal	Berhasil	Berhasil	Berhasil
3	18	Gagal	Berhasil	Berhasil	Berhasil
4	21	Gagal	Berhasil	Berhasil	Berhasil

Berdasarkan 4 kali uji coba untuk memilih bobot SAW dengan jam yang berbeda pada hari yang sama, menunjukkan bahwa dari 12 kali uji coba algoritma A – Star berhasil menemukan rute sebanyak 12 kali. Rute yang ditemukan sama yaitu 13 titik dengan titik yang dilalui adalah B202 - B199 - B204 - B205 - B234 - B233 - B207 - B208 - B209 - B214 - B213 - B211 - B212. 00 untuk setiap bobot yang berbeda memiliki rentang perbandingan yang sangat minimal untuk nilai alternatif akhir.

Tabel 4.3. Perbandingan SAW

Jam	Curent State	Node	Saw 60%:40%	SAW 65%:35%	SAW 70%:30%	SAW 77%:23%
21:00 WIB	B233	B207	3,256327778	3,359487553	3,346569595	3,328484455
		B229	3,186268446	3,346401102	3,37253993	3,409134289

Hal ini dapat dilihat dari tabel 4.14 SAW dengan bobot 77% pada jarak dan 23% pada kemacetan memiliki rentang nilai antara node B207 dengan nilai 3.328484455 dan node B229 dengan nilai 3.409134289 memiliki rentang nilai 0.080649834 pada merasakan bahwa ia memiliki kisaran nilai tertinggi. lebih baik dari bobot lainnya. Oleh karena itu, penulis memilih bobot SAW sebesar 77% pada jarak dan 23% pada kemacetan sebagai bobot SAW pada aplikasi ini.

4.2.3 Pengujian Algoritma

Pada pengujian Algoritma, beberapa pengujian dilakukan dengan lokasi pengguna yang berbeda, lokasi tujuan yang berbeda, dan waktu yang berbeda untuk melihat perbedaan pada setiap rute.

Tabel 4.4. Pengujian Algoritma.

No	Lokasi User	Tujuan	Waktu	Rute Terpilih	Ket.
1	SMK Cengkareng	RS Mitra Keluarga Kalideres	Senin 12:00 WIB	B28 - B29 - B30 - B31 - B32 - B54	Berhasil
2	SMK Cengkareng	Laboratorium Klinik Bioprima	Selasa 12:00 WIB	B28 - B29 - B30	Berhasil
3	SMK Cengkareng	RSUD Kalideres	Rabu 12:00 WIB	B28 – B19 - B20 - B10	Berhasil
4	SMK Cengkareng	RS Ciputra	Kamis 12:00 WIB	B28 - B19 - B20 - B10 - B7	Berhasil
5	TPU Tegal Alur	RS Ciputra	Jumat 12:00 WIB	B1 - B3 -B4 -B5 -B7	Berhasil
6	Klenteng Kiu Lie Tong	Hotel Luminor Mangga Besar	Senin 21:00 WIB	B202 - B199 - B204 - B205 - B234 - B233 - B207 - B208 - B209 - B214 - B213 - B211 - B212	Berhasil
7	Klenteng Kiu Lie Tong	Laboratorium Klinik Utama Satria Medika	Selasa 21:00 WIB	B202 - B199 - B204 - B205 - B234 - B233 - B207 - B208 - B209	Berhasil
8	Klenteng Kiu Lie Tong	Laboratorium Intibios	Rabu 21:00 WIB	B202 - 199 - B204 - B205 - B234 - B233 - B207 - B208 - B209 - B211	Berhasil
9	Klenteng Kiu Lie Tong	Klinik Utama Rosela Indah	Kamis 21:00 WIB	B202 – B200 - B198 - B197 - B196 - B179 - B182 - B185 - B194 - B183 - B184 -B195 - B192 - B193 - B186 - B187 - B168 - B169	Berhasil
10	Casa Jardin Residen	Laboratorium RS Cendana	Jumat 21:00 WIB	B167 - B163 - B149 - B87 - B86 - B88 - B98 - B99 - B146 - B145	Berhasil

Path diambil dari node yang dihasilkan dari algoritma perhitungan kemudian node tersebut akan dicocokkan dengan data koordinat berupa garis lintang dan garis bujur masing-masing node yang ada di database kemudian diolah oleh mapbox.

Dari hasil pengujian pada tabel di atas dengan jarak 77%:23% kemacetan terdapat 10 pengujian pencarian rute, terdapat 5 rute yang diuji pada siang hari pukul 12:00 WIB dan 5 rute lainnya diuji pada malam hari pukul 21:00 WIB . Dari 10 rute yang diuji, semuanya berhasil menentukan rute terpendek menggunakan Algoritma A – Star. Pemilihan rute dari pengujian di atas tetap dalam kondisi dimana rute yang diambil berada di bawah 25 node. Dalam arti algoritma A – Star dapat memaksimalkan semua perhitungan dalam menentukan jalur terdekat pada aplikasi ini.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan pada bab sebelumnya, dapat membuka kunci aplikasi untuk membuka rute terdekat dari lokasi pasien ke fasilitas terkait covid-19 dengan aplikasi A - Star dapat menentukan rute terpendek rute penanganan kasus Covid-19 di DKI Jakarta dengan menggunakan bobot optimal sebesar 77% pada jarak dan 23% pada kemacetan. Rata-rata memori yang dibutuhkan untuk menentukan rute dengan algoritma A - Star pada aplikasi ini adalah 84,76 MB dan waktu yang dibutuhkan untuk menentukan rute dengan algoritma A - Star memiliki rata-rata 00:02,126 (2126 ms).

5.2 Saran

Saran penulis untuk penelitian selanjutnya adalah sebagai berikut:

1. Dalam pengembangan aplikasi selanjutnya diharapkan dapat dijalankan pada sistem operasi IOS.
2. Dalam pengembangan selanjutnya untuk menambahkan fitur info tentang status rumah sakit dan call center dari rumah sakit.

REFERENSI

[1] S. Ye, "Research on Urban Road Traffic Congestion Charging Based," in Physics Procedia 24, China, 2012.

[2] V. Lomendra, P. Sharmila, D. Ganess and N. Vandisha, "Assessing The Causes & Impacts of Traffic Congestion On The Society, Economy, and Individual: A Case of Mauritius As An Emerging Economy," Sciendo, vol. 3, no. 13, pp. 230 - 242 , 2018.

[3] B. Aditya, "Penyebab Kemacetan Lalu Lintas yang Perlu Diketahui," Merdeka.com, 15 Juni 2020. [Online]. Available: https://www.merdeka.com/trending/penyebab-kemacetan-lalu-lintas-yang-perlu-diketahui-kln.html. [Accessed 10 November 2020].

[4] Dob, "Nasional Melandai, Tapi Kasus Covid-19 di Jakarta Naik Terus," CNBCIndonesia.com, 12 Oktober 2020. [Online]. Available: https://www.cnbcindonesia.com/news/20201012161544-4-193730/nasional-melandai-tapi-kasus-covid-19-di-jakarta-naik-terus. [Accessed 10 November 2020].

[5] M. Ling, S. Fajar, H. Nur, W. Romi, "Pemanfaatan Metode Heuristik Dalam Pencarian Jalur Terpendek Dengan Algoritma Semut Dan Algoritma Genetika". Presented at Seminar Nasional Aplikasi Teknologi Informasi 2007 (SNATI 2007), Yogyakarta, 2007.

[6] K. Setiawan, Supriyadin, I. Santoso, R. Buana, "Menghitung Rute Terpendek Menggunakan Algoritma A* Dengan Fungsi Euclidean Distance". Presented at Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA 2018), Yogyakarta, 2018.

[7] M. H. Fadhulrrahman, "Perencanaan dan Prototipe Aplikasi Pemilihan Rute Kendaraan Roda Dua: Pencarian Rute Alternatif Terpendek Berdasarkan Data Kemacetan Lalu Lintas Berbasis Android," Telkom University, Januari 2015. [Online]. Available: https://openlibrary.telkomuniversity.ac.id/home/catalog/id/100050/slug/perancangan-dan-prototipe-aplikasi-pemilihan-rute-kendaraan-roda-dua-pencarian-rute-alternatif-terpendek-berdasarkan-data-kemacetan-lalu-lintas-berbasis-android.html. [Accessed 11 November 2020].

[8] C. Lin, D. He, X. Huang, M. K. Khan and K.-K. R. Choo, "A New Transitively Closed Undirected Graph Authentication Scheme for Blockchain-based Identity Management," IEEE Access, pp. 28203 - 28212, 17 Mei 2018.

[9] S. Bhowmick and S. Panja, "Leader-follower Bipartite Consensus of Linear Multiagent Systems over a Signed Directed Graph," IEEE Transactions on Circuits and Systems II: Express Briefs, pp. 1436 - 1440, 28 November 2018.

[10] D. Zhang and L. Xiong, "The Research of Dynamic Shortest Path based on cloud computing," in International Conference on Computational Intelligence and Security, China, 2016.

[11] R. D. Putra, "Aplikasi Pencarian Rute Terpendek Menggunakan Algoritma Djikstra Berbasis Android Untuk Pemetaan Rumah Sakit di Kota Bandung," Telkom University, 2 Juli 2013. [Online]. Available: https://openlibrary.telkomuniversity.ac.id/pustaka/93087/aplikasi-pencari-rute-terpendek-menggunakan-algoritma-dijkstra-berbasis-android-untuk-pemetaan-rumah-sakit-di-kota-bandung.html. [Accessed 11 November 2020].DOZIER, Gerry, et al. Artificial potential field based robot navigation, dynamic constrained optimization and simple genetic hill-climbing. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360). IEEE, 1998. p. 189-194.

[12] CHOPDE, Nitin R.; NICHAT, Mangesh. Landmark based shortest path detection by using A* and Haversine formula. International Journal of Innovative Research in Computer and Communication Engineering, 2013, 1.2: 298-302.

[13] Frieyadie, "Penerapan Metode Simple Additive Weight (Saw)Dalam Sistem Pendukung Keputusan Promosi Kenaikan Jabatan". Jurnal Pilar Nusa Mandiri Vol.XII, No. 1, Jakarta, 2016

[14] D. Yang, B. Xu, K. Rao and W. Sheng, "Passive Infrared (PIR)-Based Indoor Position Tracking for Smart Homes Using Accessibility Maps and A-Star Algorithm," MDPI, vol. 2, no. 18, 2018.

[15] D. Hermanto and S. Dermawan, "Penerapan Algoritma A-Star Sebagai Pencari," Jurnal Nasional Teknik Elektro, vol. 7, no. 2, p. 2302 2949, 2018.

[16] T. P. Nugroho, E. Rohadi, A. Prasetyo, "Aplikasi Pencarian Jalur Terpendek Untuk Menemukan Lokasi Atm Di Kota Malang," Jurnal Informatika Polinema, vol. 3, no. 4, 2018.

[17] Suharjanti. "Analisis Validitas Dan Reliabilitas Dengan Skala Likert Terhadap Pengembangan Si/Ti Dalam Penentuan Pengambilan Keputusan Penerapan Strategic Planning Pada Industri Garmen", Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST), Yogyakarta,2014.

[18] Mapbox Route generation [online]. Available: https://docs.mapbox.com/android/navigation/guides/route-generation/,[Accessed 28 July 2021].