

# Analisis Sistem Pencarian Jalur Terdekat Bengkel UMKM Kendaraan Roda Dua Dengan Algoritma A\* (*A Star*) dan Formula *Haversine*

Muhamad Femy Mulya<sup>1</sup>, Dedy Trisanto<sup>2</sup>, Nofita Rismawati<sup>3</sup>, Saipul Anwar<sup>4</sup>

<sup>1,4</sup>Program Studi Sistem Informasi, Tanri Abeng University, Jakarta, Indonesia

<sup>2</sup>Program Studi Sistem Informasi Industri Otomotif, Politeknik STMI, Jakarta, Indonesia

<sup>3</sup>Program Studi Teknik Informatika, Universitas Indraprasta PGRI, Jakarta, Indonesia

<sup>1</sup>femy.mulya@tau.ac.id, <sup>2</sup>dedymail2001@gmail.com, <sup>4</sup>saipul@tau.ac.id, <sup>3</sup>novi.9001@gmail.com

Diterima : 20 Februari 2022

Disetujui 27 Maret 2022

**Abstract**— Pertumbuhan kendaraan roda dua di DKI Jakarta dari tahun ke tahun semakin kian meningkat. Berdasarkan data Badan Pusat Statistik (2020) jumlah kendaraan bermotor roda dua di DKI Jakarta mencapai 16.141.380 unit. Faktor masalah yang sering terjadi pada pengguna roda dua apabila kendaraan tiba-tiba mengalami kerusakan motor di tengah perjalanan di DKI Jakarta tentu akan kesulitan mencari lokasi bengkel motor terdekat. Oleh karena itu, dibutuhkan suatu aplikasi layanan berbasis *mobile* untuk memberikan informasi layanan bengkel sepeda motor dan informasi mengenai layanan bengkel agar jaringan layanan bengkel UMKM (Usaha Mikro Kecil dan Menengah) menjadi semakin luas. Aplikasi ini memberi informasi jarak dan rute *customer* menuju lokasi layanan bengkel kendaraan bermotor roda dua UMKM. Pada penelitian ini menggunakan formula Haversine untuk mencari bengkel UMKM dengan nilai jarak terkecil dan algoritma A\* (*A Star*) untuk mencari rute dengan jarak tempuh terpendek. Untuk pengembangan aplikasi ini menggunakan *Android Studio* dengan bahasa pemrograman *Java* serta, untuk penyimpanan *database* aplikasi ini menggunakan layanan *Firebase*. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *incremental development*. Hasil perhitungan formula *Haversine* dapat dijadikan tolak ukur untuk menentukan lokasi bengkel UMKM yang akan dicari rute terdekatnya. Lalu pada penelitian ini juga, menggunakan metode pengujian *black box* dengan parameter *test case* untuk pencarian rute bengkel UMKM terdekat untuk wilayah Jakarta Timur.

**Keyword:** *Bengkel UMKM, Algoritma A\* (A Star), Formula Haversine*

## I. PENDAHULUAN

Pertumbuhan kendaraan di DKI Jakarta dari tahun ke tahun semakin meningkat. Hal tersebut dapat dilihat dari semakin banyaknya kendaraan bermotor roda dua yang beredar di jalanan Ibu Kota DKI Jakarta, sehingga banyak menimbulkan kemacetan khususnya di jalanan Ibu Kota. Untuk itu, bengkel kendaraan UMKM roda dua di Ibu

Kota DKI Jakarta sangat diperlukan untuk membantu para pengendara roda dua, jika sewaktu-waktu kendaraannya mengalami masalah dan butuh untuk proses perbaikan kendaraan dengan cepat.

Bengkel UMKM di DKI Jakarta saat ini sering ditemui di pinggir jalan ataupun di dekat rumah. Dengan jumlah bengkel yang banyak tersebut

tentu menjadi daya saing untuk setiap bengkel. Belum adanya aplikasi pelayanan *customer* bengkel UMKM berbasis *mobile* yang memiliki fitur pencarian bengkel terdekat dengan efisien. Oleh karena itu, dibutuhkan suatu aplikasi layanan berbasis *mobile* untuk memberikan informasi layanan bengkel UMKM agar jaringan layanan bengkel UMKM menjadi luas.

Berdasarkan permasalahan diatas, maka dibutuhkan suatu aplikasi pencarian rute terpendek untuk menuju ke lokasi bengkel UMKM di wilayah DKI Jakarta menggunakan metode *Haversine* dan Algoritma A\* (*A Star*) sebagai penghitung jarak terpendek antar node dengan menggunakan layanan *Google Maps*. Metode *Haversine* dan Algoritma A\* (*A Star*) merupakan salah satu algoritma untuk menemukan jarak terpendek antar node pada suatu *graf* yang memiliki bobot [1]. Algoritma ini tepat untuk diimplementasikan dalam mencari rute terpendek karena proses waktu pencarian membutuhkan waktu lebih cepat dan langkah yang digunakan lebih sedikit [2]. Dengan didukung layanan *Google Maps* aplikasi ini diharapkan lebih memudahkan dan memberikan informasi lokasi bengkel UMKM yang lebih spesifik kepada para pengguna kendaraan bermotor roda dua khususnya di daerah Ibu Kota DKI Jakarta.

Tujuan dari penelitian ini adalah untuk menganalisa dan merancang suatu aplikasi berbasis *mobile android* untuk pencarian rute terpendek ke lokasi bengkel UMKM di wilayah DKI menggunakan metode *Haversine* dan Algoritma A\* (*A Star*) sebagai penghitung jarak terpendek dengan memanfaatkan layanan *Google Maps*, JSON (*Javascript Object Notation*) serta *Firestore*.

## II. LANDASAN TEORI

### A. Graf

*Graf* merupakan salah satu teori yang paling banyak diterapkan dalam berbagai bidang. Hal ini disebabkan karena *graf* dapat menjelaskan keterkaitan hubungan antar objek sehingga akan lebih mudah dipahami. *Graf*(G) biasa dinotasikan sebagai  $G = (V, E)$  [3]. Himpunan dari simpul-

simpul dilambangkan dengan V (*vertices* atau node), sementara E adalah himpunan objek penghubung antar simpul atau biasa disebut sebagai sisi (*edges* atau *arcs*) [4].

### B. Formula Haversine

Formula *Haversine* merupakan salah satu rumusan yang sering digunakan dalam bidang navigasi. Sistem navigasi seringkali membutuhkan jarak terpendek antara dua titik untuk keperluan sistemnya [5]. Hal tersebut dapat dihasilkan oleh formula *haversine*. Formula *haversine* menghitung jarak antara dua titik atau node berdasarkan koordinat lintang dan bujurnya [6].

$$\Delta\varphi = \frac{\pi}{180} * (\varphi_1 - \varphi_2) \quad (1)$$

$$\Delta\lambda = \frac{\pi}{180} * (\lambda_1 - \lambda_2) \quad (2)$$

$$a = \sin^2(\Delta\varphi/2) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2(\Delta\lambda/2) \quad (3)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (4)$$

$$d = R \cdot c \quad (5)$$

Dimana:

$\varphi$  = latitude

$\lambda$  = longitude

c = jarak angular (radian) d = jarak (km)

R = jari-jari bumi (6.371 km)

Penggunaan formula ini mengabaikan ketinggian bukit, kedalaman lembah dan bentuk bumi yang tidak sepenuhnya bulat sempurna. Jadi formula ini mengasumsikan bahwa bumi memiliki dataran yang rata dan berbentuk bulat sempurna [7].

### C. Algoritma A\* (*A Star*)

Algoritma *path finding* yang sering digunakan untuk mencari rute terpendek salah satunya adalah algoritma A\* (*A Star*). Algoritma A\* berhasil mencari rute terpendek dari titik awal menuju titik akhir dengan mempertimbangkan nilai dari F terkecil. Nilai F dapat dicari dengan menggunakan rumus [8]:

$$f(n) = h(n) + g(n) \quad (6)$$

Dimana:

$g(n)$  = nilai *movecost*

$h(n)$  = nilai heuristik

Nilai *movecost* pada algoritma A\* (A Star) merujuk pada biaya yang dikeluarkan jika terjadi perpindahan antar *node* [1]. Biaya tersebut dapat berupa jarak tempuh, waktu tempuh dan sebagainya. Sedangkan nilai heuristik mewakili biaya yang dikeluarkan dari *node* n ke *node* tujuan yang dihitung dengan fungsi heuristik.

#### D. JSON (JavaScript Object Notation)

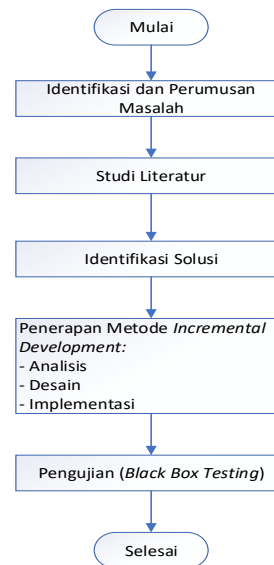
JSON (JavaScript Object Notation) adalah bahasa pertukaran data yang tidak bergantung dengan bahasa pemrograman apapun. Programmer banyak menggunakan JSON sebagai bahasa pertukaran data dikarenakan format ini dapat dengan mudah dibuat dan dipahami manusia [5]. JSON juga dapat di-generate dan dipahami dengan mudah oleh komputer sehingga prosesnya akan ringan dan cepat. Oleh karena hal-hal tersebut, JSON sangat baik digunakan untuk media pertukaran data.

#### E. Firebase

Firestore Realtime Database merupakan jenis database yang di letakkan pada hosting layanan cloud [9]. Data pada firestore disimpan dalam format standar dari JSON, kemudian data tersebut akan disinkronisasi secara realtime ke semua user yang terhubung ke aplikasi. Ketika membuat aplikasi antar-platform yang berbeda baik itu dengan SDK Android, iOS, dan JavaScript, maka semua user akan melakukan proses sharing informasi dari instance realtime database dan user juga akan menerima update data atau informasi terbaru secara otomatis [9].

### III. METODE PENELITIAN

Untuk menganalisa dan merancang sistem pencarian jalur terdekat Bengkel UMKM kendaraan roda dua dengan Algoritma A\* (A Star) dan Formula Haversine, maka pada penelitian ini akan menggunakan alur tahapan penelitian sebagai berikut:



Gambar 1. Alur Tahapan Penelitian

Pada gambar 1 terlihat alur tahapan penelitian yang dimulai dengan tahapan atau proses identifikasi dan perumusan masalah yang banyak dialami pengendara kendaraan bermotor roda dua yang mengalami kesulitan dalam mencari bengkel terdekat, ketika mengalami permasalahan terkait kondisi kendaraan, serta belum adanya aplikasi pelayanan customer bengkel UMKM berbasis mobile yang memiliki fitur pencarian rute terdekat menuju bengkel UMKM. Selanjutnya, tahapan berikutnya dilakukan studi literatur terkait sistem pencarian jalur terdekat dengan Algoritma A\* (A Star) dan Formula Haversine dari beberapa referensi penelitian sejenis. Lalu pada alur tahapan berikutnya dilakukan identifikasi solusi dengan mengidentifikasi maksud dan tujuan penelitian, serta menerapkan Algoritma A\* (A Star) dan Formula Haversine. Kemudian menerapkan Metode Incremental Development yang terdiri dari 3 tahapan yaitu proses Analisis, Desain dan Implementasi.

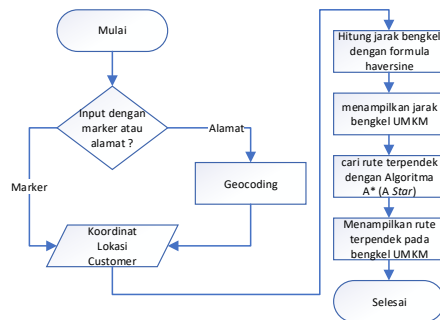
Pada tahapan Analisis dilakukan proses identifikasi kebutuhan sistem baru dengan melakukan analisis permasalahan dan analisis proses bisnis yang diusulkan, lalu pada tahapan desain dilakukan proses pemodelan sistem menggunakan UML (Unified Modelling Language) serta perancangan antar muka sistem yang dibangun. Selanjutnya pada tahapan implementasi dilakukan proses pembuatan aplikasi dengan Android Studio dan firebase.

Lalu pada tahapan akhir alur penelitian, akan dilakukan proses pengujian sistem dengan *Black Box Testing* menggunakan beberapa parameter *test case*.

#### IV. HASIL DAN PEMBAHASAN

##### A. Alur Perancangan Sistem Pencarian Jalur Terdekat Bengkel UMKM

Penelitian ini berusaha untuk membantu *customer*/pengguna aplikasi dalam mencari bengkel UMKM kendaraan roda dua dengan jarak terdekat. Untuk dapat mewujudkan hal tersebut, penulis menggunakan formula *Haversine* dan algoritma *A\** (*A Star*) dalam membangun sistem ini. Berikut ini adalah *flow chart* dari sistem yang dibuat sebagai berikut:

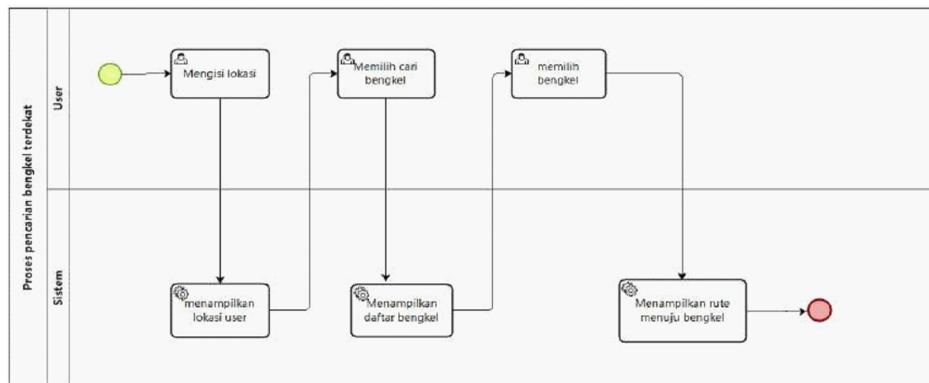


Gambar 2. Alur Perancangan Sistem Pencarian Jalur Terdekat Bengkel UMKM

##### B. BPMN (Business Process Model and Notation) Usulan Sistem Pencarian Bengkel UMKM Terdekat

Berdasarkan hasil analisis dari perumusan masalah yang ada, maka diusulkan suatu proses bisnis sebagai berikut:

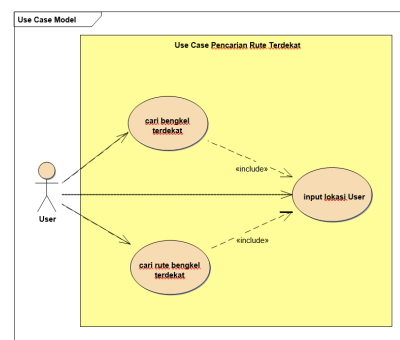
1. User mengisi lokasi keberadaan ke dalam *form* yang telah disediakan.
2. Sistem akan menampilkan lokasi *user*.
3. *User* akan memilih menu cari bengkel UMKM di dalam aplikasi.
4. Setelah *user* memilih menu cari bengkel UMKM, kemudian sistem akan menampilkan bengkel yang telah terdaftar di dalam aplikasi.
5. *User* akan memilih bengkel UMKM sesuai dengan keinginan dan kebutuhan.
6. Sistem akan mencari rute menuju lokasi bengkel UMKM yang telah dipilih.
7. Sistem akan menampilkan rute terdekat dari lokasi *user*.



Gambar 3. BPMN (Business Process Model and Notation) Usulan Sistem Pencarian Bengkel UMKM Terdekat

##### C. Use Case Diagram Sistem Pencarian Bengkel UMKM Terdekat

*Use Case Diagram* merupakan deskripsi sebuah interaksi antara satu atau lebih aktor dengan sebuah sistem informasi yang dibuat. Berikut merupakan *use case* diagram sistem pencarian bengkel UMKM kendaraan roda dua terdekat menuju lokasi *customer* pada aplikasi bengkel UMKM yang diusulkan, sebagai berikut:

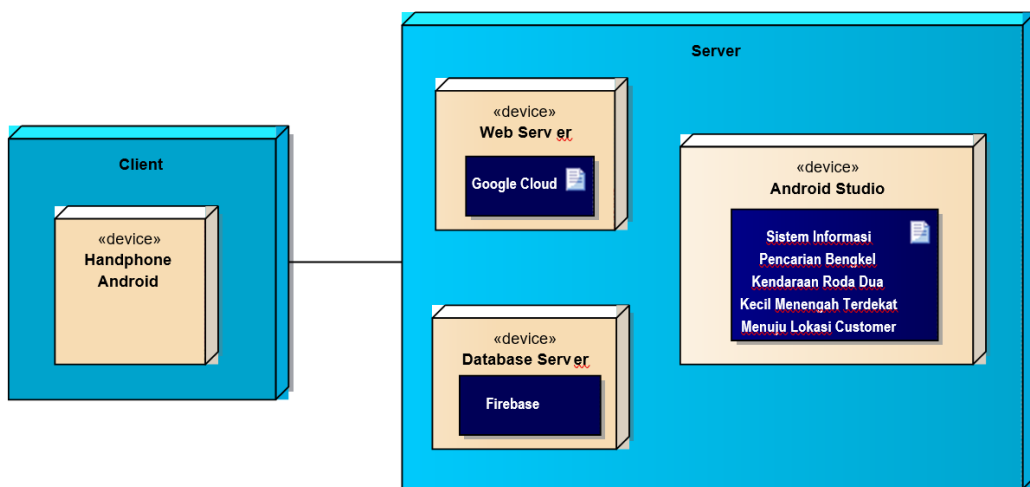


Gambar 4. Use Case Diagram Sistem Pencarian Bengkel UMKM Terdekat

Pada *Use Case Diagram* yang terlihat pada gambar 4 terdiri dari 1 aktor user yang terhubung dengan 3 *use case*. Pada *use case Input lokasi user* ini menggambarkan setiap aktor dapat memasukkan lokasi sesuai dengan titik keberadaan, lalu pada *use case mencari bengkel terdekat* ini menggambarkan setiap aktor dapat mencari bengkel UMKM yang berada dari lokasi terdekat, dan yang terakhir pada *use case mencari rute bengkel UMKM Terdekat* ini menggambarkan setiap aktor dapat mencari rute bengkel UMKM terdekat dari lokasi user berada.

*D. Deployment Diagram Sistem Pencarian Bengkel UMKM Terdekat*

*Deployment diagram* pada usulan sistem informasi pencarian bengkel UMKM kendaraan roda terdekat menuju lokasi *customer* digunakan untuk mewakili setiap komponen dari *software* dan bagaimana cara *software* tersebut ditempatkan pada infrastruktur atau arsitektur fisik sistem informasi. Berikut terlihat pada gambar 5 merupakan *deployment diagram* sistem informasi pencarian bengkel UMKM kendaraan roda dua terdekat menuju ke lokasi *customer*.

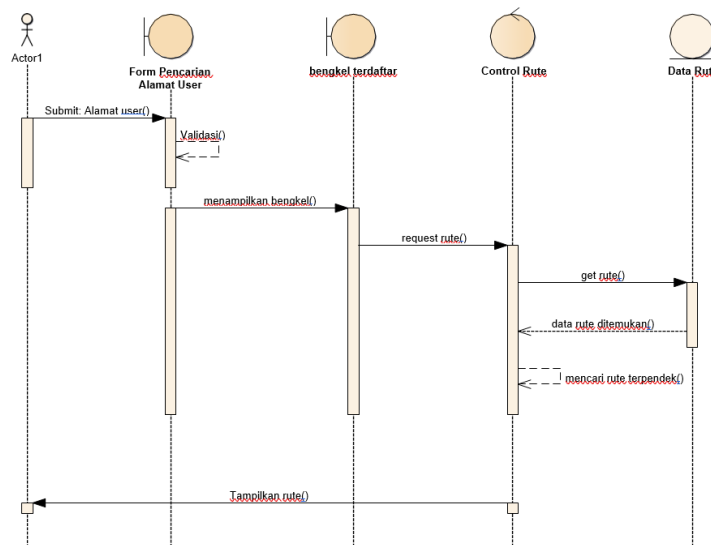


Gambar 5. *Deployment Diagram Sistem Pencarian Bengkel UMKM Terdekat*

*E. Sequence Diagram Sistem Pencarian Bengkel UMKM Terdekat*

*Sequence Diagram* ini menjelaskan alur aktivitas yang dilakukan oleh *user* untuk mencari

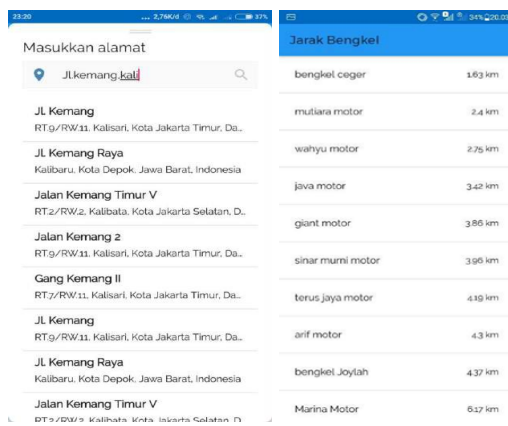
rute terdekat menuju bengkel dari lokasi *user*. *Sequence diagram* mencari rute terdekat dapat dilihat pada gambar 6 sebagai berikut:



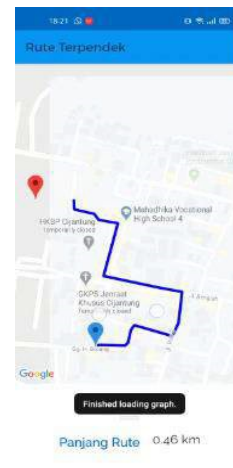
Gambar 6. *Sequence Diagram Sistem Pencarian Bengkel UMKM Terdekat*

**F. Rancangan Antarmuka Sistem Pencarian Bengkel UMKM Terdekat**

Implementasi *User Interface* berisi tampilan dari tiap-tiap proses yang ada pada sistem pencarian rute terdekat menggunakan Formula *Haversine* dan Algoritma *A\** (*A Star*). Pada bagian ini ada tiga tampilan yang akan ditampilkan, yaitu tampilan input lokasi *user*, tampilan daftar bengkel UMKM terdekat, dan tampilan rute terdekat bengkel UMKM, adapun tampilan antarmuka pada aplikasi ini sebagai berikut:



Gambar 7. Antarmuka Input Lokasi User dan Antarmuka Daftar Bengkel UMKM Terdekat



Gambar 9. Antarmuka Rute Terdekat Bengkel UMKM

**G. Pengujian Mencari Rute Terdekat**

Pengujian ini dilakukan untuk menguji apakah daftar urutan bengkel terdekat hasil dari formula *Haversine* berbanding lurus dengan jarak rute hasil Algoritma *A\** (*A Star*). Urutan awal pengujian akan menggunakan bengkel UMKM dengan nilai jarak terkecil sementara bengkel UMKM dengan jarak terbesar akan di uji coba pada urutan akhir. Berikut merupakan table uji coba untuk mencari rute terdekat menggunakan *Graph Hopper*:

Tabel 1. Hasil Pengujian Mencari Rute Terdekat

No.	Bengkel	Jumlah node	Node yang dikunjungi	Waktu proses	Jarak rute
1.	Condet Motor	85	1301	1.1	2.8 Km
2.	Bengkel Motor Papa Bro	84	1404	0.37	3.8 Km
3.	Wijaya Motor	140	4289	0.29	5.9 Km
4.	Sahabat Motor	115	5418	0.004	6.8 Km
5.	Bengkel Motor Joylah	218	10570	0.08	9.9 Km

**H. Hasil Pengujian Black Box**

*Black box testing* dilakukan untuk menguji perangkat lunak atau aplikasi yang berfokus dalam spesifikasi fungsional dari software atau perangkat lunak. *Black box testing* yang dilakukan pada penelitian ini, yaitu pengujian untuk Mencari Bengkel UMKM Terdekat dan

pengujian untuk Mencari Rute Bengkel UMKM Terdekat. Adapun untuk hasil pengujian *Black Box* pada penelitian ini sebagai berikut:

Tabel 2. Hasil Pengujian Black Box

No	Test Case (Kelas Uji)	Skenario Uji	Hasil yang Diharapkan	Valid (✓) atau Tidak Valid (x)
1	Perhitungan formula Haversine	Tidak <i>input</i> lokasi user di form yang telah disediakan, lalu klik cari bengkel	Tetap dapat menampilkan jarak bengkel terdekat karena <i>maps</i> membaca <i>marker user</i>	✓
2	Perhitungan formula Haversine	Input lokasi <i>user</i> dengan benar	Sistem akan menampilkan jarak bengkel terdekat dari lokasi yang telah dimasukkan <i>user</i>	✓
3	Perhitungan Algoritma *	Memilih bengkel yang telah ditampilkan sistem	Sistem akan menampilkan rute terdekat dari lokasi <i>user</i>	✓

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Setelah dilakukan serangkaian penelitian dan pengujian mengenai penggunaan Algoritma A\* (A Star) dan Formula Haversine pada aplikasi mobile yang telah dibangun untuk mencari jarak dan rute terdekat bengkel UMKM, maka dapat disimpulkan sebagai berikut:

1. Implementasi formula Haversine pada aplikasi *mobile* pencarian bengkel UMKM terdekat berhasil menghitung jarak antara lokasi *customer*/pelanggan dan beberapa sampel bengkel UMKM yang ada di wilayah Jakarta Timur. Hasil dari formula Haversine tersebut diurutkan berdasarkan nilai terkecil untuk mengetahui lokasi bengkel UMKM mana yang paling dekat dengan lokasi *customer*.
2. Hasil perhitungan formula Haversine dapat dijadikan tolak ukur untuk menentukan lokasi bengkel UMKM mana yang akan dicari rute terdekatnya. Namun pada beberapa kasus,

urutan daftar bengkel terdekat yang dihasilkan Haversine tidak selalu sama dengan daftar jarak rute yang dihasilkan oleh Algoritma A\* (A Star), hal ini disebabkan oleh rute yang dihasilkan Algoritma A\* (A Star) terdapat banyak tikungan dan jalan yang memutar arah sehingga jarak rute akan semakin besar.

3. Pada aplikasi *mobile* pencarian bengkel UMKM terdekat yang dirancang dan dibangun, rute terdekat dari lokasi bengkel UMKM menuju lokasi *customer* dapat berhasil ditemukan menggunakan Algoritma A\* (A Star). Algoritma A\* (A Star) berjalan dengan menggunakan *graf* sebagai media pencarian rute. Sistem menggunakan koordinat lokasi bengkel dan koordinat lokasi user sebagai input untuk Algoritma A\* (A Star).

### B. Saran

Berdasarkan hasil dari penelitian dan pengujian yang telah dilakukan. Berikut ini adalah beberapa saran yang dapat dipergunakan untuk pengembangan penelitian terkait permasalahan yang terjadi sebagai berikut:

1. Perlu dilakukan penelitian lebih lanjut untuk mengatasi kasus-kasus tertentu terkait urutan daftar bengkel UMKM terdekat yang dihasilkan Formula Haversine tidak selalu sama dengan daftar jarak rute yang dihasilkan oleh Algoritma A\* (A Star).
2. Menambahkan fitur rute alternatif sehingga bila ada permasalahan di rute yang dilalui, *customer* dapat mengganti ke rute perjalanan yang lebih baik.
3. Memperluas cakupan wilayah penelitian meliputi wilayah Nasional, sehingga dapat mencakup kepentingan masyarakat yang lebih luas.

## DAFTAR PUSTAKA

- [1] A. B. W. Putra, A. A. Rachman, A. Santoso, and M. Mulyanto, "Perbandingan Hasil Rute Terdekat Antar Rumah Sakit di Samarinda Menggunakan Algoritma A\*(star) dan Floyd-Warshall," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 1, pp. 59–68, Mar. 2020, doi: 10.32736/sisfokom.v9.i1.685.
- [2] R. Chandra Sani, J. Rekayasa Sistem Komputer, J. Sistem Informasi, and J. H. Hadari Nawawi Pontianak, "PROTOTIPE PERANGKAT PENDETEKSI LOKASI

- MOBIL PELAYANAN UMUM BERGERAK BERBASIS GEOLOCATION MENGGUNAKAN ALGORITMA PENCARIAN A\*,” *Coding J. Komput. dan Apl.*, vol. 6, no. 3, pp. 23–32, Aug. 2018, doi: 10.26418/CODING.V6I3.27438.
- [3] T. D. PUTRI, W. SUGENG, and E. SAFITRI, “Algoritma Dijkstra untuk Penentuan Jarak Tempuh Terpendek Pengantaran Katering Pabrik,” *MIND (Multimedia Artif. Intell. Netw. Database) J.*, vol. 5, no. 2, pp. 108–120, Jul. 2020, doi: 10.26760/MINDJOURNAL.V5I2.108-120.
- [4] W. Bismi, W. Gata, T. Asra, I. Komputer, and U. Nusa Mandiri, “Application of Hybrid Algorithm in Determining the Shortest Route Between Campus Branches,” *Ultim. Comput. J. Sist. Komput.*, vol. 13, no. 1, pp. 1–9, Jun. 2021, doi: 10.31937/SK.V13I1.1856.
- [5] S. Kartika, S. Suendri, and R. A. Putri, “Sistem Pencarian Lokasi dan Rute Terdekat Menggunakan Metode Haversine Formula Pada Aplikasi Donatur Pakaian Berbasis Android,” *AL-ULUM J. SAINS DAN Teknol.*, vol. 7, no. 1, Feb. 2022, doi: 10.31602/AJST.V7I1.5678.
- [6] M. Arifin and E. Mailoa, “APPLICATION OF HAVERSINE FORMULA IN NEAREST SALATIGA REPAIR SHOP SEARCH GEOGRAPHIC INFORMATION SYSTEM USING FLUTTER FRAMEWORK,” *Cybersp. J. Pendidik. Teknol. Inf.*, vol. 5, no. 2, pp. 157–171, Oct. 2021, doi: 10.22373/CJ.V5I2.9350.
- [7] W. Dari, L. Indah Prahartiwi, N. Mandiri, J. Damai No, and W. Jati Barat Jakarta Selatan DKI Jakarta, “Metode Haversine Formula Dalam Sistem Informasi Pariwisata Kabupaten Banyuwangi (SIPBANGI) Berbasis Android,” *BINA Insa. ICT J.*, vol. 6, no. 1, pp. 11–22, Jun. 2019, Accessed: Feb. 14, 2022. [Online]. Available: <http://ejournal-binainsani.ac.id/index.php/BIICT/article/view/1096>.
- [8] S. Kasus, D. Daerah, K. Baru, J. Selatan, and A. Kurniawati, “Penerapan Algoritma A\*(STAR) Untuk Mencari Rute Tercepat Pada Suatu Bengkel.”
- [9] S. P. E. S. G.S, “Perancangan Sistem Informasi Geografis Pariwisata Pasaman Barat Berbasis Android Menggunakan Metode Haversine,” *J. Komput. Ter.*, vol. 7, no. 2, pp. 240–250, Dec. 2021, doi: 10.35143/JKT.V7I2.4876.