

Implementasi *Haversine Formula* dan *Algoritme Dijkstra* pada Aplikasi *Vehicle Help* Sebagai Layanan Konsultasi Pemilik Kendaraan

Ferawaty¹, Albert Cenderawan^{2*}

Program Studi Informatika, Universitas Pelita Harapan, Kampus Kota Medan, Medan, Indonesia

*E-mail *Corresponding Author*: albertcenderawan123@gmail.com

Abstract

Owners of motorized vehicles who still use conventional methods to find information on the location of refueling and vehicle service workshops, receive inaccurate and not up-to-date information. The purpose of this research is to develop the Vehicle Help application to make it easier for motorists to search for information on the location of the nearest fuel filling station and service shop. This study implements and tests the accuracy of the Haversine Formula and Dijkstra's algorithm in recommending the location of the nearest fuel filling station and service shop. The system development methodology used is the Waterfall model. The results of the study show that the mobile-based Vehicle Help application that was built can provide easy and complete information on where to fill up fuel and vehicle service workshops through digital mapping and available recommendation features. In addition, the results of the Haversine Formula test and Dijkstra's algorithm in recommending gas stations and service shops get an accuracy of 90% from 10 trials.

Keywords: *Location Search; Digital Map; Service Workshop; Vehicle Help; Mobile Application*

Abstrak

Pemilik Kendaraan bermotor yang masih memanfaatkan cara konvensional dalam mencari informasi lokasi tempat pengisian Bahan Bakar Minyak (BBM) dan bengkel servis kendaraan, memperoleh informasi yang tidak tepat dan tidak *up-to-date*. Tujuan penelitian ini adalah mengembangkan aplikasi *Vehicle Help* untuk mempermudah pengendara melakukan pencarian informasi lokasi tempat pengisian BBM dan bengkel servis terdekat. Penelitian ini mengimplementasikan dan menguji akurasi *Haversine Formula* dan algoritme *Dijkstra* dalam merekomendasikan lokasi tempat pengisian BBM dan bengkel servis terdekat. Metodologi pengembangan sistem yang digunakan adalah dengan model *Waterfall*. Hasil penelitian menunjukkan bahwa aplikasi *Vehicle Help* berbasis *mobile* yang dibangun dapat memberikan informasi tempat pengisian BBM dan bengkel servis kendaraan secara mudah dan lengkap melalui pemetaan digital dan fitur rekomendasi yang tersedia. Selain itu, hasil pengujian *Haversine Formula* dan algoritme *Dijkstra* dalam merekomendasikan SPBU dan bengkel *service* mendapatkan akurasi sebesar 90% dari 10 kali percobaan.

Kata kunci: *Pencarian Lokasi; Peta Digital; Bengkel Servis; Vehicle Help; Aplikasi Mobile*

1. Pendahuluan

Sistem transportasi nasional memegang peran yang sangat penting dalam mendukung pembangunan nasional, yakni sebagai jembatan utama sekaligus urat nadi ekonomi dalam sebuah wilayah maupun antarwilayah. Dengan sistem transportasi yang baik, mobilitas masyarakat semakin mudah dan semakin cepat antarwilayah [1]. Arus distribusi barang (terutama bahan-bahan pokok) semakin mudah dilakukan sehingga dapat meningkatkan nilai ekonomi suatu wilayah negara tersebut. Bagian dari sistem transportasi yang turut memberikan kontribusi dalam meningkatkan perekonomian di suatu wilayah adalah transportasi darat. Setiap transportasi darat memiliki mesin dan *spare part* di dalamnya sehingga setiap tahunnya memerlukan perawatan (*services*) agar tidak cepat rusak ataupun mogok ketika beroperasi. Apalagi khususnya kendaraan mobil yang jika secara insidental mengalami kehabisan bensin, kerusakan atau memerlukan perawatan tentunya lebih sulit untuk dipindahkan sehingga

membutuhkan informasi tempat pengisian bensin dan bengkel di sekitar lokasi agar dapat segera mendapatkan penindakan terhadap kendaraan tersebut [2].

Pada praktiknya, dalam melakukan pencarian lokasi tempat pengisian bensin dan bengkel *service* masih dilakukan secara konvensional dengan bertanya kepada masyarakat sekitar. Namun seringkali informasi yang diberikan kurang jelas dan sesuai seperti informasi alamat yang salah dan jam operasional yang tidak sesuai sehingga ketika sudah sampai lokasi ternyata tempat tersebut sudah tutup. Selain secara konvensional, mencari lokasi pengisian bensin dan bengkel juga dilakukan dengan pemanfaatan teknologi informasi. Meski demikian, aplikasi teknologi informasi sekarang masih belum menjamin ketersediaan informasi tempat perawatan kendaraan secara *real-time* [3][4]. Hal ini disebabkan oleh kurangnya pengawasan oleh pemilik perawatan kendaraan sehingga informasi-informasi di dalam menjadi tertinggal (kurang *up-to-date*). Beberapa tempat perawatan kendaraan juga tidak mencantumkan fasilitas dan jasa perawatan yang tersedia secara akurat. Di sisi lain, banyak pemilik/pengusaha yang lebih suka mencantumkan foto-foto (interior maupun eksterior) tempat perawatan kendaraan untuk dijadikan 'penanda' (supaya pengunjung tidak salah tempat) [5]. Permasalahan pada penelitian ini juga pernah dibahas di beberapa penelitian terdahulu, dimana penelitian terdahulu tersebut mengembangkan sebuah aplikasi pencarian lokasi tempat pengisian bensin dan bengkel *service* terdekat dengan mengimplementasikan algoritme *Dijkstra*, namun hasil penelitian terdahulu belum menguji secara pasti akurasi dari model penelitian yang diusulkan [5][6][7][8].

Berdasarkan uraian permasalahan tersebut, maka perlu dirancang dan dibangun sebuah aplikasi *mobile* yang menawarkan bantuan terbaik ketika kendaraan butuh perbaikan ataupun memerlukan informasi tempat pengisian bensin di situasi genting yang dinamakan "*Vehicle Help*". Pada penelitian ini akan diimplementasikan *Haversine Formula* dan algoritme *Dijkstra* dalam merekomendasikan lokasi tempat pengisian bensin dan bengkel *service* terdekat yang ditampilkan dalam peta digital *Google Maps* sehingga pengemudi segera mendapatkan penanganan terhadap kendaraannya [9]. Penelitian ini memilih algoritme *Dijkstra* dikarenakan berdasarkan penelitian yang dilakukan oleh Umar, dkk. pada tahun 2021 membahas mengenai perbandingan algoritme *Dijkstra*, *A-Star*, dan *Floyd Warshal* didapatkan hasil bahwa algoritme *Dijkstra* lebih unggul dalam proses pencarian rute [10]. Untuk meningkatkan hasil rekomendasi pencarian terdekat maka akan dikombinasikan algoritme *Dijkstra* dengan *Haversine Formula* sehingga diharapkan dapat meningkatkan akurasi hasil rekomendasi [11]. Tujuan penelitian ini yaitu mengembangkan sebuah aplikasi "*Vehicle Help*" sebagai layanan konsultasi pemilik kendaraan serta mengimplementasikan dan menguji akurasi dari *Haversine Formula* dan algoritme *Dijkstra* dalam merekomendasikan lokasi tempat pengisian bensin dan bengkel *service* terdekat. Dengan adanya penelitian ini diharapkan dapat bermanfaat dalam mempermudah masyarakat dalam mendapatkan informasi lokasi pengisian bensin dan bengkel *service* kendaraan secara cepat, tepat, akurat, dan *up to date*.

2. Tinjauan Pustaka

Penelitian terdahulu yang berkaitan dengan topik penelitian yang dilakukan penulis sudah pernah dilakukan oleh Sutriyono pada tahun 2019 yang membahas mengenai implementasi algoritme *Dijkstra* pada aplikasi pencarian bengkel tambal ban terdekat di kota Medan berbasis Android. Penelitian berfokus pada fitur pencarian rute terpendek menuju ke lokasi bengkel tambal ban terdekat dengan mengimplementasikan algoritme *Dijkstra*. Model penelitian ini memanfaatkan teknologi *Global Positioning System* (GPS) untuk mendeteksi posisi pengguna dan kemudian algoritme *Dijkstra* akan merekomendasikan jalur terpendek untuk pengguna menuju ke lokasi bengkel tambal ban terdekat. Hasil penelitian berupa dikembangkannya sebuah aplikasi berbasis *mobile* untuk pencarian bengkel tambal ban terdekat. Pada penelitian yang akan dilakukan penulis juga mengembangkan aplikasi pada teknologi berbasis *mobile* dan memanfaatkan GPS dalam mendeteksi lokasi pengguna. Kekurangan pada penelitian ini belum dilakukan pengujian akurasi dari algoritme *Dijkstra* dalam merekomendasi rute terpendek bengkel tambal ban terdekat. Kekurangan pada penelitian ini menjadi gap yang akan dilakukan penelitian ini dimana akan diuji dan dievaluasi akurasi *Haversine Formula* dan algoritme *Dijkstra* dalam merekomendasikan lokasi SPBU ataupun bengkel *service* terdekat.

Berikutnya, penelitian yang dilakukan oleh Arta dan Indrawan pada tahun 2021 yang membahas mengenai rancang bangun sistem informasi geografis pencarian fasilitas umum

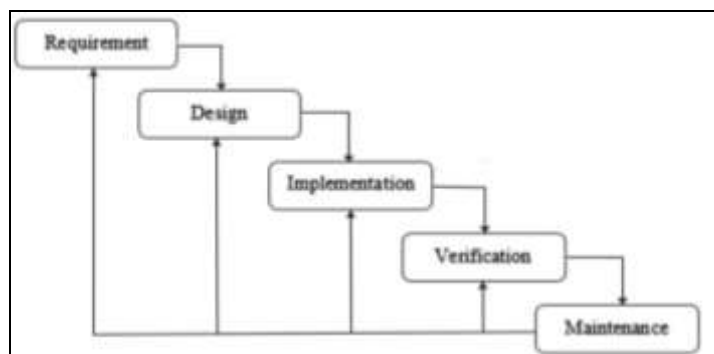
terdekat meliputi informasi Anjungan Tunai Mandiri (ATM), Stasiun Pengisian Bahan Bakar (SPBU), rumah makan, bengkel tambal ban dan apotek. Metode pengembangan aplikasi yang digunakan adalah *Waterfall Model*. Hasil yang didapatkan dengan adanya sebuah Sistem informasi geografis ini diharapkan dapat lebih memudahkan dalam memberikan informasi mengenai tata letak fasilitas umum yang berada di kota Denpasar. Informasi yang disajikan pada android ini meliputi informasi Anjungan Tunai Mandiri (ATM), Stasiun Pengisian Bahan Bakar (SPBU), rumah makan, bengkel tambal ban dan apotek [8]. Pada penelitian yang akan dilakukan penulis juga mengimplementasikan metodologi pengembangan sistem *Waterfall Model*, namun perbedaannya adalah penelitian terdahulu hanya berbasis pada fitur pencarian dan belum memiliki fitur rekomendasi lokasi SPBU ataupun bengkel *service* terdekat, sehingga *gap* yang ditemukan pada penelitian ini akan diimplementasikan metode *Haversine Formula* dan algoritme *Dijkstra* dalam merekomendasikan lokasi SPBU ataupun bengkel *service* terdekat secara cepat dan akurat sehingga memudahkan pengendara.

Penelitian terbaru dilakukan oleh Devia, dkk. pada tahun 2022 yang membahas mengenai rancang bangun aplikasi pencarian bengkel motor berbasis Android di wilayah kota Bekasi. Hasil penelitian berupa sebuah aplikasi pencarian bengkel berbasis Android. Dalam aplikasi ini terdapat beberapa halaman (*page*) yang bisa diakses oleh pemakai, serta memiliki *database* yang diberi nama "bengkel" dengan isi 2 buah tabel yaitu: tabel *login* dan tabel bengkel. Dalam penelitian ini aplikasi yang dihasilkan telah melewati tahap uji coba dengan menggunakan metode pengujian *BlackBox* [5]. Penelitian terbaru ini cukup baik dimana sudah dilakukan pengujian terhadap aplikasi yang dibangun, namun kekurangannya dimana penelitian yang dilakukan oleh Devia, dkk. hanya mengembangkan aplikasi pencarian bengkel tanpa mengimplementasikan algoritme untuk meningkatkan akurasi hasil pencariannya. Dari *gap* tersebut, maka peneliti mengimplementasikan metode *Haversine Formula* dan algoritme *Dijkstra* agar dapat meningkatkan akurasi dari hasil pencarian.

Dapat disimpulkan bahwa perbedaan penelitian yang dilakukan dengan penelitian terdahulu yaitu penelitian yang dilakukan saat ini memiliki fitur yang lebih kompleks yang berfokus dalam mengatasi permasalahan dari para pengendara motor dalam mencari informasi bengkel *service* dan lokasi pengisian bensin. Aplikasi yang akan dibangun pada penelitian ini berfokus pada aspek kecepatan pencarian serta kemudahan dalam berkonsultasi antara pengendara dengan pihak *vendor* serta pengujian akurasi dari *Haversine Formula* dan algoritme *Dijkstra* dalam merekomendasikan lokasi SPBU ataupun bengkel *service* terdekat.

3. Metodologi

Metodologi pengembangan sistem yang digunakan pada penelitian ini yaitu *Waterfall Model*. Model Waterfall merupakan salah satu model SDLC yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model ini menggunakan pendekatan sistematis dan berurutan. Berikut ini, Gambar 1 menunjukkan alur tahapan dari metode *Waterfall Model*.



Gambar 1. Alur Tahapan Metode *Waterfall Model*

a. Analisis Kebutuhan (*Requirement*)

Pada tahap ini dilakukan pengumpulan informasi dengan melakukan observasi ke beberapa SPBU dan bengkel *service* di kota Medan untuk mengumpulkan informasi yang akan didaftarkan ke aplikasi yang dibangun serta gambaran proses bisnis yang diperlukan untuk

diimplementasikan pada sistem usulan yang akan dibangun. Informasi-informasi yang didapatkan kemudian akan dianalisis yang terbagi dalam beberapa tahapan yaitu analisis masalah, analisis kebutuhan sistem, dan pemodelan kebutuhan sistem menggunakan *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

b. Desain Sistem (*Design*)

Pada tahap ini dilakukan perancangan desain sistem dengan menggunakan *software Balsamic Wireframe Mockups 3*, perancangan *flowchart* menggunakan *Diagrams.net*, serta perancangan basis data menggunakan *Class Diagram*.

c. Implementasi (*Implementation*)

Pada tahap ini, seluruh hasil analisis dan perancangan akan diimplementasikan dalam bentuk sebuah aplikasi berbasis *mobile* yang dibangun dengan menggunakan *Framework Laravel* sebagai *Back End* serta bahasa pemrograman Java sebagai *Front End* dan basis data menggunakan MySQL.

d. Verifikasi (*Verification*)

Pada tahap ini akan dilakukan verifikasi terhadap sistem usulan yang telah dibangun dengan melakukan testing atau pengujian terhadap sistem yang telah dibuat untuk memastikan sistem berjalan sesuai dengan keinginan lalu mengevaluasi sistem. Selain itu, tahapan ini juga akan menguji akurasi dari *Haversine Formula* dan algoritme *Dijkstra* dalam merekomendasikan SPBU atau bengkel *service* terdekat dari GPS Pengguna.

4. Hasil dan Pembahasan

Hasil penelitian berupa dikembangkannya sebuah aplikasi yang dinamakan "*Vehicle Help*" sebagai platform bagi pemilik kendaraan untuk mencari informasi bengkel *service* dan lokasi pengisian bensin serta layanan konsultasi bagi pemilik kendaraan. Berikut ini diuraikan hasil penelitian yang telah dilakukan berdasarkan tahapan metode penelitian.

4.1. Analisis Kebutuhan

Analisis pada penelitian ini dibagi menjadi 3 tahapan yaitu analisis kebutuhan sistem, analisis proses, dan pemodelan sistem usulan.

a. Analisis kebutuhan sistem

Analisis kebutuhan sistem menguraikan sistem berjalan yang digunakan oleh pemilik kendaraan dalam mencari informasi bengkel *service* dan tempat pengisian bensin. Pada praktiknya, dalam melakukan pencarian lokasi tempat pengisian bensin dan bengkel *service* masih dilakukan secara konvensional dengan bertanya kepada masyarakat sekitar. Namun seringkali informasi yang diberikan kurang jelas dan sesuai seperti informasi alamat yang salah dan jam operasional yang tidak sesuai sehingga ketika sudah sampai lokasi ternyata tempat tersebut sudah tutup. Selain secara konvensional, mencari lokasi pengisian bensin dan bengkel juga dilakukan dengan pemanfaatan teknologi informasi. Meski demikian, *software* pencarian informasi dan lokasi SPBU dan bengkel *service* hingga saat ini masih belum menjamin ketersediaan informasi tempat perawatan kendaraan secara *real-time*. Hal ini disebabkan oleh kurangnya pengawasan oleh pemilik perawatan kendaraan (bengkel dan *service*) sehingga informasi-informasi bengkel *service* dalam aplikasi menjadi tertinggal (kurang *up-to-date*). Beberapa tempat perawatan kendaraan juga tidak mencantumkan ketersediaan fasilitas dan jasa perawatan secara akurat. Di sisi lain, banyak pemilik/pengusaha yang lebih menyukai kegiatan mencantumkan foto-foto (interior maupun eksterior) tempat perawatan kendaraan untuk dijadikan 'penanda' (supaya pengunjung tidak salah tempat).

b. Analisis proses

Analisis proses akan menguraikan proses kerja dari formula *Haversine* yaitu suatu formula dalam penentuan jarak dari titik koordinat berdasarkan posisi garis lintang dan garis bujur [12] dan algoritme *Dijkstra* dalam memberikan rekomendasi tempat pengisian bensin dan bengkel *service* terdekat.

Diketahui (Posisi diambil dari GPS aplikasi pemetaan *Google Maps*)

Rumah : 3.596072, 98.668233 (Rumah di Jalan Bungur No. 12B, Medan).

SPBU : *diambil dari jangkauan area kota Medan, Sumatera Utara.*

- Pertamina 14.201.1166 : 3.594362, 98.668547
- Shell Adam Malik : 3.606438, 98.671688
- Pertamina 14.201.123 : 3.592363, 98.669859
- Pertamina 14.201.110 : 3.594562, 98.671953

Akan ditentukan SPBU yang paling dekat dengan koordinat rumah, serta rute mana yang paling pendek untuk dilewati untuk sampai ke SPBU terdekat

Penjelasan:

CASE 01: RUMAH – PERTAMINA 14.201.1166

Tabel 1. Titik Koordinat antara Rumah dan SPBU Pertamina 14.201.1166

	<i>LATITUDE*</i>	<i>LONGITUDE*</i>
RUMAH	3.596072	98.668233
PERTAMINA 14.201.1166	3.594362	98.668547

*satuan derajat.

Konversi *latitude-longitude* dari navigasi GPS ke Radian.

Tabel 2. Tabel Konversi *Latitude-Longitude* dari Navigasi GPS ke Radian

	<i>LATITUDE*</i>	<i>LONGITUDE*</i>
RUMAH	0.062763297	1.722085533
PERTAMINA 14.201.1166	0.062733451	1.722091013

*satuan radian.

Penerapan metode *Haversine* untuk selisih *latitude*.

$$\text{Hav}(\varphi_1 - \varphi_2) = 0.5 * [1 - \cos(\varphi_1 - \varphi_2)]$$

$$\text{Hav}(\varphi_1 - \varphi_2) = 0.5 * [1 - \cos(0,062763297 - 0.062733451)]$$

$$\text{Hav}(\varphi_1 - \varphi_2) = 0.5 * [1 - \cos(0.000029846)]$$

$$\text{Hav}(\varphi_1 - \varphi_2) = 0.5 * [1 - 0.999999999985] = 0.000000000223$$

Penerapan metode *Haversine* untuk selisih *longitude*.

$$\text{Hav}(\lambda_1 - \lambda_2) = 0.5 * [1 - \cos(\lambda_1 - \lambda_2)]$$

$$\text{Hav}(\lambda_1 - \lambda_2) = 0.5 * [1 - \cos(1.722091013 - 1.722085533)]$$

$$\text{Hav}(\lambda_1 - \lambda_2) = 0.5 * [1 - \cos(0.00000548)]$$

$$\text{Hav}(\lambda_1 - \lambda_2) = 0.5 * [1 - 0.999999999985] = 0.000000000008$$

Penerapan metode *Haversine* dengan menggunakan fungsi *Haversine* di atas.

$$\text{Hav}(\Theta) = \text{Hav}(\varphi_1 - \varphi_2) + \cos(\varphi_1) * \cos(\varphi_2) * \text{Hav}(\lambda_1 - \lambda_2)$$

$$\text{Hav}(\Theta) = 0.000000000223 + \cos(0.062763297) * \cos(0.062733451) * 0.000000000008$$

$$\text{Hav}(\Theta) = 0.000000000223 + 0.998032902288 * 0.998031030783 * 0.000000000008 = 0.000000000230$$

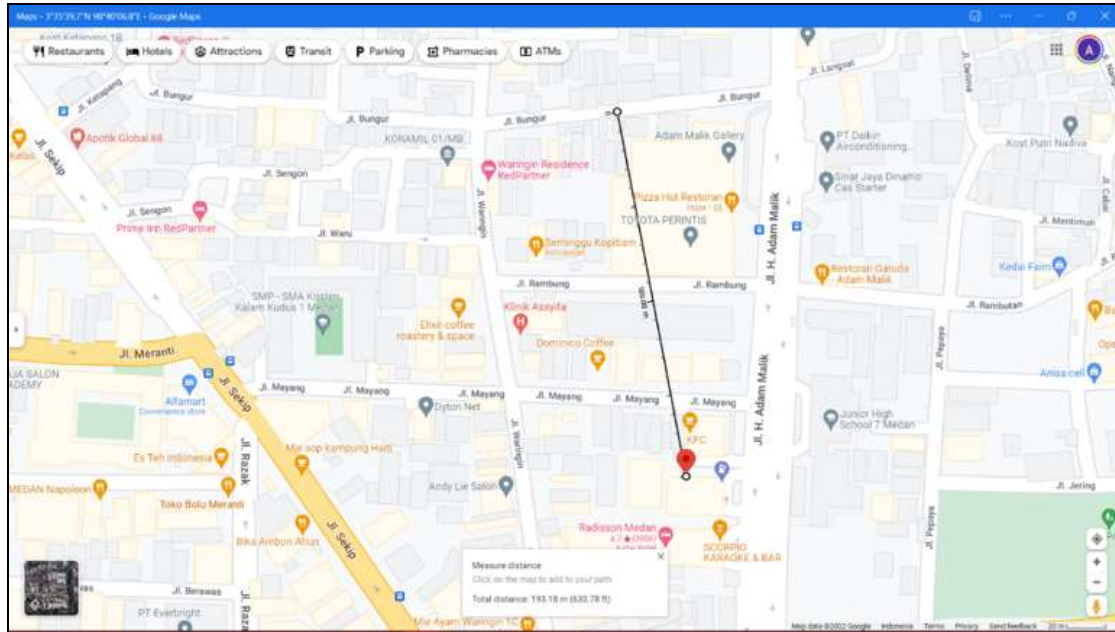
Penerapan metode *Haversine* untuk menentukan jarak antartitik koordinat.

$$\Theta = \cos^{-1}[1 - 2(\text{Hav}(\Theta))]$$

$$\Theta = \cos^{-1}[1 - 2(0.000000000230)] = 0.000030342175$$

$$d/r = 0.000030342175$$

$$d = \text{distance (dalam km)} = 0.000030342175 * 6371 = 0.19330999 \text{ km} = 193.30999 \text{ m.}$$



Gambar 2. Hasil Pengukuran Jarak “Rumah” ke “SPBU Pertamina 14.201.1166”

Bila dibandingkan dari fitur “*measure distance*” dari salah satu sistem aplikasi, ditemukan bahwa jarak yang dihimpun dari *Haversine method* mendekati jarak sebenarnya. Proses perhitungan dilanjutkan pada SPBU Pertamina lainnya yang terdaftar di aplikasi hingga mendapatkan hasil perhitungan jarak seperti Tabel 3.

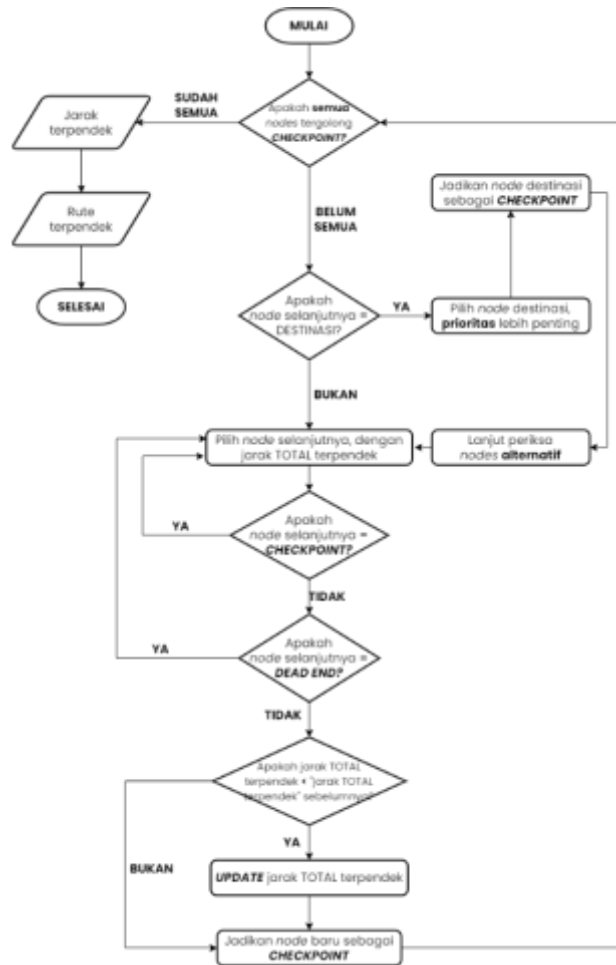
Tabel 3. Perbandingan Jarak antara Rumah ke Empat SPBU di Kota Medan

TITIK AWAL	TITIK AKHIR	JARAK*
Rumah	Pertamina 14.201.1166	193.3099
Rumah	Shell Adam Malik	1214.7448
Rumah	Pertamina 14.201.123	450.1701
Rumah	Pertamina 14.201.110	445.6695

*dalam satuan meter.

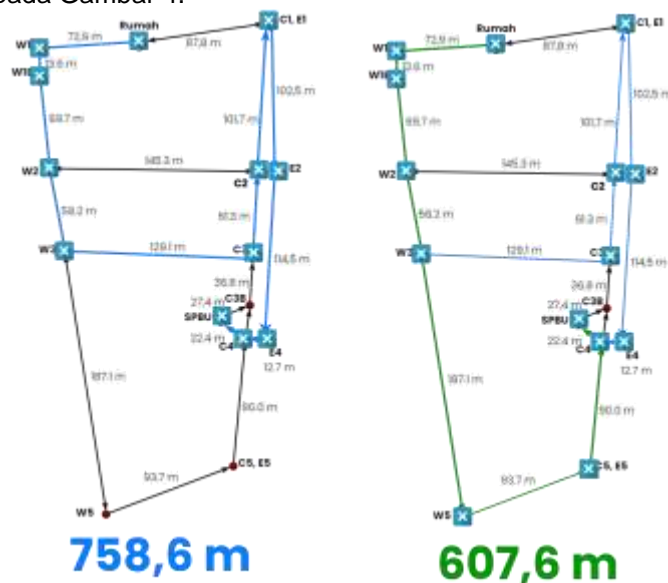
Algoritme Dijkstra: Menemukan Rekomendasi Rute Terpendek

Algoritme yang paling banyak digunakan untuk menentukan masalah pemilihan *route*, baik penentuan *route* terpendek maupun *route* yang paling optimal adalah algoritme *Dijkstra*. Algoritme Dijkstra menawarkan *node-node* yang dilalui untuk mencari rute terpendek dari *node* awal hingga *node* tujuan. Berdasarkan nilai minimum bobot bagian yang diberikan pada serangkaian tahapan-tahapan solusi. Setiap graf terdapat sumber node untuk kemudian secara strategi algoritme *greedy* dilakukan pencarian minimal bobot terkecil sehingga didapati nilai terpendek yang merupakan rute atau jarak terpendek [13]. Gambar 3 menunjukkan *Flowchart Diagram* penerapan algoritme *Dijkstra*.



Gambar 3. Flowchart Diagram dalam Menerapkan Algoritme Dijkstra

Setelah mengikuti setiap langkah/prosedur dalam *flowchart diagram* algoritme Dijkstra di atas, ditemukan dua rute yang menjadi rekomendasi rute terpendek menggunakan algoritme Dijkstra seperti pada Gambar 4.



Gambar 4. Hasil Rekomendasi Rute Terpendek Menggunakan Algoritme Dijkstra

Bila disusun ke dalam tabel, berikut adalah dua rekomendasi rute terpendek menurut algoritme Dijkstra:

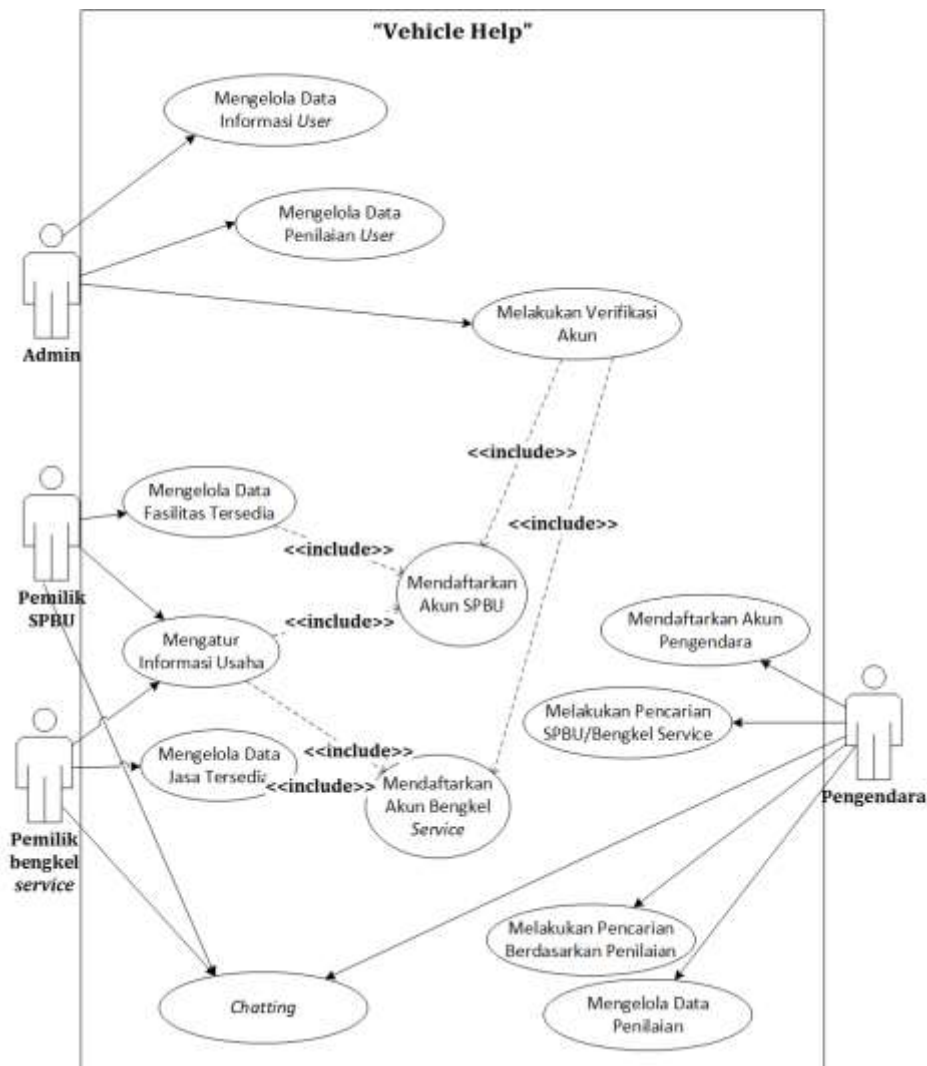
Tabel 4. Tabel Perbandingan Rute 1 dan Rute 2 menggunakan Algoritme *Dijkstra*

No	Rute	Daftar <i>Node</i> yang dilewati	Jarak (m)
1	Rumah – SPBU (Rute Biru)	Rumah, W1, W1B, W2, W3, C3, C2, C1E1, E2, E4, C4, SPBU	758,6
2	Rumah – SPBU (Rute Hijau)	Rumah, W1, W1B, W2, W3, W5, C5E5, C4, SPBU	607,6

Dengan demikian, bisa disimpulkan bahwa jarak Rumah ke SPBU Pertamina (melalui satu garis lurus) adalah 14.201.1166 adalah 193,18 meter (dari metode *Haversine*). Sedangkan rute terpendek yang bisa ditempuh dari Rumah ke SPBU Pertamina 14.201.1166 adalah 607,6 meter (menurut algoritme *Dijkstra*).

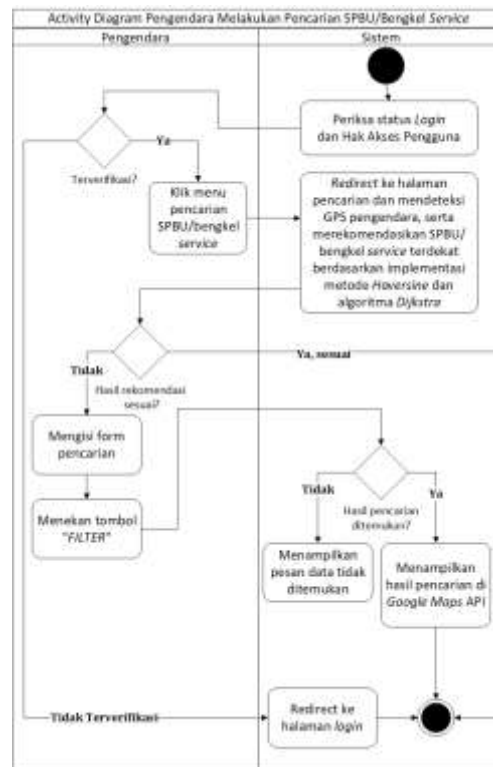
c. Pemodelan kebutuhan sistem

Pemodelan kebutuhan sistem digambarkan dengan UML berupa *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*. Gambar 5 menunjukkan model *Use Case Diagram* dari aplikasi “*Vehicle Help*”.



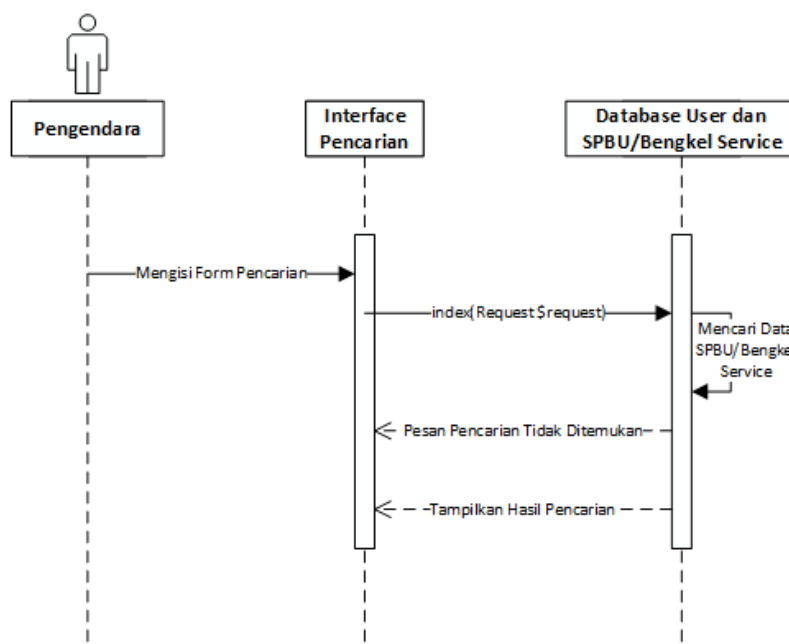
Gambar 5. Model *Use Case Diagram* Aplikasi “*Vehicle Help*”

Berikutnya, Gambar 6 menunjukkan model salah satu *Activity Diagram* dari aplikasi “*Vehicle Help*”.



Gambar 6. Model Salah Satu *Activity Diagram* Aplikasi “*Vehicle Help*” Melakukan Pencarian SPBU/Bengkel Service

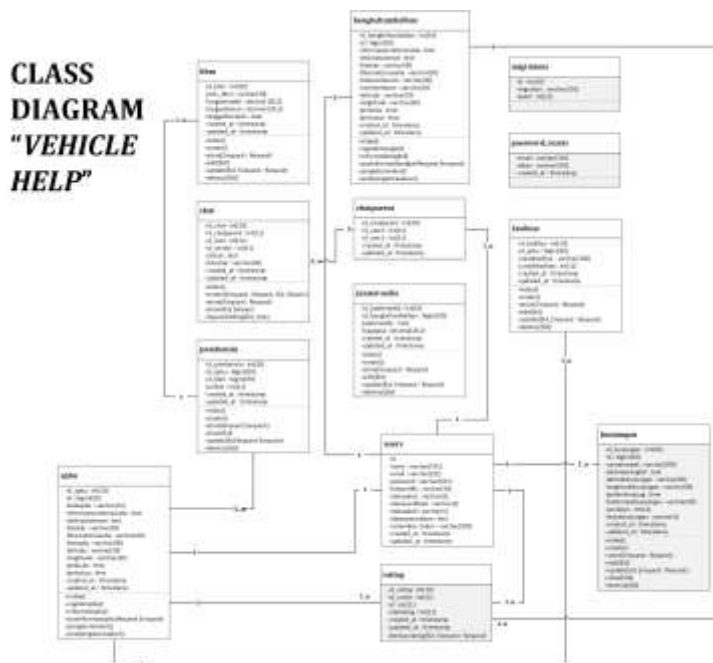
Kemudian, Gambar 7 menunjukkan model salah satu *Sequence Diagram* dari aplikasi “*Vehicle Help*”.



Gambar 7. Model Salah Satu *Sequence Diagram* Aplikasi “*Vehicle Help*” Melakukan Pencarian SPBU/Bengkel Service

4.2. Desain Sistem

Pada tahapan ini dilakukan perancangan *prototype* sistem dengan menggunakan *software* Balsamiq Mockup 3. Setelah perancangan *prototype* sistem selesai, maka berikutnya adalah merancang basis data yang akan digunakan pada aplikasi yang akan dibangun dengan menggunakan *Class Diagram*. *Class Diagram* aplikasi “*Vehicle Help*” dapat dilihat pada Gambar 8.



Gambar 8. Class Diagram Aplikasi “*Vehicle Help*”

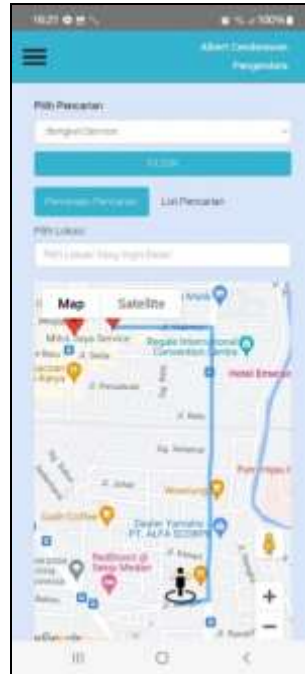
4.3. Implementasi

Pada tahap ini dilakukan implementasi sistem dimana aplikasi yang dibangun berupa sebuah *installer* dengan format .apk yang dapat diinstallasi pada *smartphone* pengendara. Ketika aplikasi berhasil diinstallasi, maka aplikasi dapat siap digunakan. Berikut ini, Gambar 9 menunjukkan tampilan dari aplikasi “*Vehicle Help*” yang dimulai dari tampilan *form login* sebagai autentikasi awal sebelum menggunakan aplikasi.



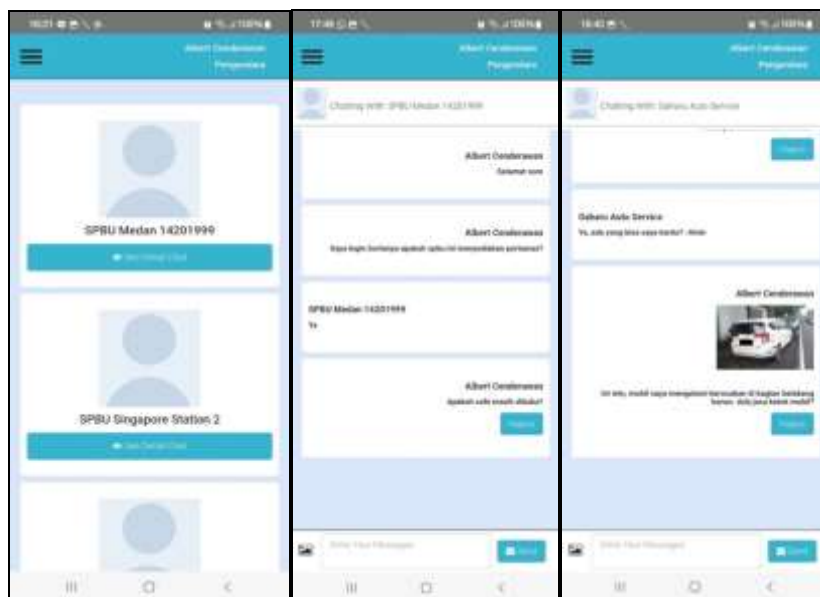
Gambar 9. Tampilan Form Login

Selanjutnya, Gambar 10 menunjukkan tampilan dari fitur pencarian lokasi tempat pengisian bensin atau bengkel *service* yang secara otomatis akan direkomendasikan lokasi terdekat dari kedua tempat tersebut menggunakan metode *Haversine Formula* yang mengukur jarak dari pengguna terhadap seluruh lokasi layanan konsultasi pemilik kendaraan yang terdaftar dan kemudian algoritme *Dijkstra* merekomendasikan rute terpendek menuju lokasi tersebut.



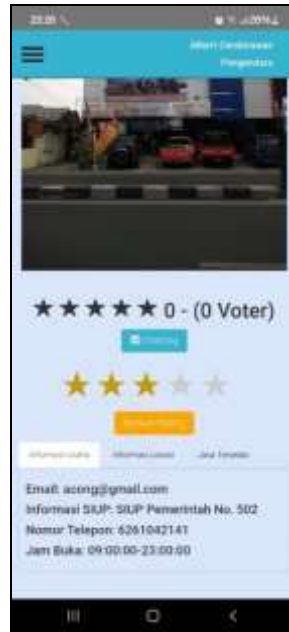
Gambar 10. Tampilan Fitur Pencarian Dengan Implementasi Metode *Haversine Formula* dan algoritme *Dijkstra*

Kemudian, Gambar 11 menunjukkan beberapa fitur-fitur yang tersedia pada aplikasi “*Vehicle Help*” yang dibangun seperti salah satunya yaitu fitur *chatting*.



Gambar 11. Tampilan Fitur *Chatting*

Terakhir, Gambar 12 menunjukkan fitur penilaian layanan konsultasi pemilik kendaraan yang tersedia. Pada fitur ini, pengendara dapat memberikan penilaian terhadap kualitas layanan dari pemilik layanan konsultasi tersebut.



Gambar 12. Tampilan Fitur Pemberian *Rating*

4.3. Verifikasi

Proses verifikasi pada penelitian ini dibagi menjadi dua yaitu pengujian *Black Box Testing* dan pengujian mengenai *Haversine Formula* dan algoritme *Dijkstra*.

- a. Pengujian *Black Box Testing* dilakukan oleh penulis/peneliti sebanyak 3 kali dan kemudian dicatat *output* yang keluar. Berikut ini akan ditunjukkan hasil dari evaluasi *Black Box Testing* yang telah dilakukan pada Tabel 5.

Tabel 5. Hasil Pengujian *Black Box Testing*

No.	Skenario Uji		Test Case	Hasil Yang Diharapkan	Output	Diterima /Ditolak
1	Mendaftarkan SPBU	Akun	Melakukan pengisian <i>form</i> pendaftaran dan menekan tombol <i>sign up</i> .	Muncul pesan validasi pendaftaran berhasil dilakukan dan menunggu admin untuk melakukan verifikasi.	Muncul pesan validasi pendaftaran berhasil dilakukan dan menunggu admin untuk melakukan verifikasi.	Diterima
2	Diterima
13	Mengelola Penilaian	Data	Melakukan percobaan penambahan, perubahan, dan penghapusan data kunjungan penilaian.	Pesan validasi data bertambah, berubah, dan terhapus.	Pesan validasi data bertambah, berubah, dan terhapus.	Diterima

- b. Proses pengujian dilakukan terhadap implementasi dari *Haversine Formula* dan algoritme *Dijkstra* dalam merekomendasikan bengkel *service* atau SPBU terdekat. Pengujian dilakukan sebanyak 10 kali dengan titik posisi yang berbeda [14]. Tabel 6 menunjukkan hasil pengujian yang dilakukan.

Tabel 6. Hasil Pengujian *Haversine Formula* dan Algoritme *Dijkstra*

Pengujian	Hasil Rekomendasi Yang Benar	Hasil Rekomendasi Sistem	Keterangan
1	SPBU Pertamina 14-202-104	SPBU Pertamina 14-202-104	Sesuai
2	Bengkel Delta Mulia	Bengkel Mobil Cuann Jaya	Tidak Sesuai
3	Medan <i>Service</i>	Medan <i>Service</i>	Sesuai
4	SPBU Medan Yos Sudarso	SPBU Medan Yos Sudarso	Sesuai
5	SPBU Medan Merak Jingga	SPBU Medan Merak Jingga	Sesuai
6	Bengkel AC Mobil & Bus Resmi Denso	Bengkel AC Mobil & Bus Resmi Denso	Sesuai
7	Mitra Jaya <i>Service</i>	Mitra Jaya <i>Service</i>	Sesuai
8	Darianto Bengkel Mesin	Darianto Bengkel Mesin	Sesuai
9	Karya Motor	Karya Motor	Sesuai
10	Samudera Jaya Motor	Samudera Jaya Motor	Sesuai

$$\begin{aligned}
 \text{Akurasi} &= (\text{Jumlah Rekomendasi Sesuai} / \text{Jumlah Pengujian}) * 100 \\
 &= (9/10) * 100 = 0,9 * 100 \\
 &= 90\%
 \end{aligned}$$

4.4. Pembahasan

Selanjutnya, pada bagian ini akan dilakukan pembahasan mengenai hasil temuan penelitian yang didapatkan. Berdasarkan hasil pengujian yang dilakukan kombinasi *Haversine Formula* dan algoritme *Dijkstra* memiliki akurasi yang cukup baik yaitu sebesar 90%. Hasil penelitian yang didapatkan sedikit bertolak belakang dengan penelitian yang dilakukan oleh Sumaryo, dkk. pada tahun 2020 dimana kombinasi *Haversine Formula* dan algoritme *Dijkstra* pada penentuan jalur terpendek pendakian gunung Merapi jalur Seo yang didapatkan pada penelitian mereka hanya mendapatkan akurasi 77% [15]. Kondisi ini terjadi dikarenakan akurasi dari GPS yang kurang begitu akurat di daerah pegunungan sehingga akurasi yang didapatkan juga menjadi rendah. Namun, hasil penelitian yang didapatkan peneliti sejalan dengan penelitian yang dilakukan oleh Barus dan William pada tahun 2022 dimana kombinasi *Haversine Formula* dan algoritme *Dijkstra* dalam rekomendasi rumah makan vegetarian terdekat yang didapatkan pada penelitian mereka mendapatkan akurasi yang sama dengan penelitian ini yaitu sebesar 90% [14]. Berdasarkan hasil temuan hasil akurasi yang didapatkan 90% dikarenakan kekurangan dari metode *Haversine Formula* dalam melakukan penghitungan jarak ketika jarak antar objek yang akan diukur terlalu berdekatan sehingga seringkali terjadi kesalahan dalam proses rekomendasi.

5. Simpulan

Setelah penelitian selesai dilakukan, maka perlu dipaparkan kesimpulan dari hasil penelitian yang didapatkan yaitu aplikasi "*Vehicle Help*" berbasis *mobile* yang dibangun dapat memberikan informasi tempat pengisian bensin dan bengkel *service* kendaraan secara mudah dan lengkap melalui pemetaan digital dan fitur rekomendasi yang tersedia dan aplikasi yang dibangun mengimplementasikan *Haversine Formula* dan algoritme *Dijkstra* pada fitur pencarian informasi lokasi tempat pengisian bensin dan bengkel *service* dimana berdasarkan hasil pengujian didapatkan persentase keakuratan rekomendasi sebesar 90%. Akurasi yang didapatkan belum mendapatkan persentase maksima dikarenakan kekurangan dari metode *Haversine Formula* dalam melakukan penghitungan jarak ketika jarak antar objek yang akan diukur terlalu berdekatan sehingga seringkali terjadi kesalahan dalam proses rekomendasi. Oleh sebab itu, penelitian mendatang dapat mengimplementasi metode lainnya seperti

Manhattan Distance ataupun *Euclidean Distance* untuk meningkatkan akurasi rekomendasi rute terdekat dari algoritme *Dijkstra*.

Daftar Referensi

- [1] N. K. Nur *et al.*, "Sistem Transportasi," Medan: Yayasan Kita Menulis, 2021.
- [2] R. M. Usup and S. Susafaati, "Perancangan Aplikasi Informasi Bengkel Mobil Daerah Jakarta Barat Berbasis Android," *JIKA (Jurnal Inform.*, vol. 5, no. 2, p. 174, 2021, doi: 10.31000/jika.v5i2.3618.
- [3] Andi, R. Purba, and R. Yunis, "Application of Blockchain Technology to Prevent The Potential Of Plagiarism in Scientific Publication," 2019, doi: 10.1109/ICIC47613.2019.8985920.
- [4] Andi, C. Juliandy, Robet, and O. Pribadi, "Securing Medical Records of COVID-19 Patients Using Elliptic Curve Digital Signature Algorithm (ECDSA) in Blockchain," *CommIT J.*, vol. 16, no. 1, pp. 87–96, 2022, doi: 10.21512/COMMIT.V16I1.7958.
- [5] E. Devia, Junaidi, and H. A. Sumantri, "Rancang Bangun Aplikasi Pencarian Bengkel Motor Berbasis Android Di Wilayah Kota Bekasi," *OKTAL J. Ilmu Komput. ...*, vol. 1, no. 02, pp. 129–138, 2022.
- [6] Sutriyono, "Implementasi Algoritme Dijkstra Pada Aplikasi Pencarian Bengkel Tambal Ban Terdekat di Kota Medan Berbasis Android," *J. Maj. Ilm. Inf. dan Teknol. Ilm.*, vol. 7, no. 1, pp. 41–45, 2019.
- [7] M. Y. Rukmana and F. Ramdani, "Implementasi Algoritme Dijkstra pada Webgis untuk Pencarian Lokasi SPBU di Kota Malang," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 6, pp. 2141–2149, 2018, [Online]. Available: <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/1520>.
- [8] I. K. J. Arta and I. P. E. Indrawan, "Rancang Bangun Sistem Informasi Geografis Pencarian Fasilitas Umum Terdekat Berbasis Android di Denpasar," *J. Manaj. dan Teknol. Inf.*, vol. 11, no. 2, pp. 19–28, 2021.
- [9] S. Aisa, "Aplikasi Pencarian Bengkel Aktif dengan Google Maps API Berbasis Web," *DoubleClick J. Comput. Inf. Technol.*, vol. 4, no. 2, p. 61, 2021, doi: 10.25273/doubleclick.v4i2.8001.
- [10] R. Umar, A. Yudhana, and A. Prayudi, "Analisis Perbandingan Algoritme Dijkstra, A-Star, Dan Floyd Warshall Dalam Pencarian Rute Terdekat Pada Objek Wisata Kabupaten Dompus," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 2, pp. 227–234, 2021, doi: 10.25126/jtiik.202182866.
- [11] R. Purnomo, T. D. Putra, H. Kusmara, W. Priatna, and F. Mukharom, "Haversine Formula to Find The Nearest PetShop," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 3, pp. 2205–2221, 2022, doi: 10.35957/jatisi.v9i3.2434.
- [12] A. Maulana, A. Solichin, and M. Syafrullah, "Penerapan Metode Haversine Pada Sistem Informasi Geografis Untuk Penentuan Lokasi Pembangunan Menara Telekomunikasi Pada Kota Tangerang," *Indones. J. Softw. Eng.*, vol. 4, no. 1, 2018, doi: 10.31294/ijse.v4i1.6294.
- [13] R. Ridlo, A. Hakim, Y. Z. Arief, A. Pangestu, and A. Jaenul, "Penggunaan Algoritme Dijkstra untuk Berbagai Masalah: Mini Review Insulation via vege based oil View project," *Artif. Intell. (JGU Thesis)*, pp. 1–10, 2021, doi: 10.13140/RG.2.2.21389.05602/1.
- [14] O. P. Barus and J. V. William, "PERANCANGAN DAN IMPLEMENTASI SISTEM INFORMASI GEOGRAFIS PEMETAAN RUMAH MAKAN VEGETARIAN MENGGUNAKAN METODE DIJKSTRA," *Inf. Syst. Dev.*, vol. 7, no. 2, pp. 62–69, 2022.
- [15] R. Y. Sumaryo, P. Harsadi, and D. Nugroho, "Implementasi Algoritme Dijkstra Dan Metode Haversine Pada Penentuan Jalur Terpendek Pendakian Gunung Merapi Jalur Selo Berbasis Android," *J. Teknol. Inf. dan Komun.*, vol. 8, no. 1, 2020, doi: 10.30646/tikomsin.v8i1.483.