

---

## **APLIKASI PENCARIAN BENGKEL TAMBAL BAN DAN SPBU TERDEKAT DI KOTA MEDAN MENGGUNAKAN METODE DIJKSTRA DAN HAVERSINE BERBASIS ANDROID**

Robet<sup>[1]</sup>, Yoshida Sary<sup>[2]</sup>, Dearfan<sup>[3]</sup>

Program Studi Teknik Informatika STMIK TIME Medan

Jl. Merbabu No.32 AA-BB Medan 20212, Telp:061-4561932

e-mail: robertdetime@gmail.com, yochie56@gmail.com, atingajang@gmail.com

---

### **Abstrak**

Sarana transportasi yang paling banyak dimiliki dan digunakan oleh masyarakat kota Medan pada saat ini adalah sepeda motor. Walaupun saat ini peningkatan jumlah sepeda motor yang pesat menimbulkan berbagai permasalahan dari sisi ekonomi, sosial dan lingkungan. Namun di sisi lain, sepeda motor memiliki harga yang mudah dijangkau oleh masyarakat dalam menunjang berbagai aktivitas bisnis, pekerjaan, pendidikan untuk berpergian dari satu tempat ke tempat yang lain. Dengan kemudahan mobilitas yang tinggi dan jangkauan akses ke suatu tempat, namun ada kekurangan dari sepeda motor yaitu mudahnya terjadi kebocoran ban serta mudahnya kehabisan bahan bakar minyak dikarenakan umumnya tangki minyak sepeda motor yang kapasitasnya kecil, sehingga pengendara sepeda motor terpaksa harus berhenti dan mendorong motornya. Fakta di lapangan para pengendara sepeda motor yang mengalami kebocoran ban dan kehabisan bahan bakar minyak, untuk mencari lokasi tambal ban ataupun SPBU seringkali dilakukan secara konvensional. Namun yang menjadi permasalahan, karena luasnya kota Medan membuat kesulitan bagi pengendara sepeda motor untuk menelusuri satu per satu lokasi bengkel tambal ban ataupun SPBU apalagi ketika malam hari tidak banyak kedua tempat tersebut buka. Memang saat ini pencarian tempat tambal ban dan SPBU sudah bisa dilakukan melalui aplikasi Google Map, namun kekurangannya adalah aplikasi Google Map belum dapat memberikan rekomendasi bengkel tambal ban dan SPBU terdekat dari posisi pengendara sepeda motor. Oleh sebab itu maka perlu dilakukan penelitian untuk membangun aplikasi pencarian bengkel tambal ban dan SPBU berbasis Android dengan penerapan metode Dijkstra dan Haversine dalam memberikan rekomendasi lokasi terdekat bengkel tambal ban dan SPBU di kota Medan.

**Kata Kunci:** *Android, Tambal Ban, SPBU, Metode Dijkstra, Haversine*

### **1. Pendahuluan**

Kemudahan dan pembiayaan yang ringan dalam memiliki sepeda motor saat ini membuat meningkatnya jumlah pengendara sepeda motor di kota Medan. Di sisi lain karena mudahnya dalam melakukan mobilitas tinggi dan menjangkau akses ke suatu lokasi merupakan salah satu tolak ukur masyarakat kota Medan menggunakan sepeda motor. Saat ini penggunaan sepeda motor kebanyakan dimanfaatkan oleh masyarakat kota Medan dalam menunjang berbagai aktivitas bisnis, pekerjaan, pendidikan untuk berpergian dari suatu tempat ke tempat yang lain. Dari kelebihan yang tampak jelas dari penggunaan sepeda motor, namun ada kekurangan dari penggunaan sepeda motor yaitu mudahnya terjadi kebocoran ban dan kehabisan bahan bakar minyak yang dikarenakan kapasitas tangki minyak yang pada umumnya relatif kecil, sehingga pengendara sepeda motor terpaksa harus berhenti dan mendorong sepeda motornya. Hal ini tidak begitu menjadi masalah jika terjadi pada waktu pagi atau siang hari, namun ketika malam hari, tentunya informasi ketersediaan dari bengkel tambal ban dan SPBU menjadi informasi yang sangat diperlukan untuk menghemat waktu dan juga menghindari resiko dari tindak kejahatan.

Fakta di lapangan, untuk mencari lokasi bengkel tambal ban dan SPBU seringkali dilakukan secara konvensional yaitu dengan mendorong sepeda motor dan mencari kedua tempat tersebut dengan menelusuri satu per satu lokasi ataupun bertanya kepada masyarakat di sekitar. Yang menjadi permasalahan adalah luasnya kota Medan menjadi satu kesulitan bagi pengendara sepeda motor, apalagi ketika di malam hari tidak banyak kedua tempat tersebut buka. Selain dengan cara konvensional tersebut memang ada aplikasi Google Map yang bisa digunakan untuk mencari kedua tempat tersebut, namun kekurangannya adalah aplikasi Google Map belum bisa memberikan rekomendasi bengkel tambal ban dan juga SPBU terdekat dari posisi pengendara sepeda motor.

Oleh karena itu, maka penelitian ini dilakukan untuk merancang sebuah aplikasi pencarian bengkel tambal ban dan SPBU terdekat di kota Medan berbasis Android sebagai salah satu solusi dan kemudahan bagi pengendara sepeda motor di kota Medan. Dengan bantuan Google Map dan Global Positioning System (GPS) dari *smartphone* pengendara sepeda motor untuk penentuan titik lokasi dan juga penggunaan metode Dijkstra dan Haversine untuk memecahkan permasalahan dalam menentukan jarak terpendek (*shortest path problem*)

### **2. Landasan Teori**

#### **a. Aplikasi**

Aplikasi adalah suatu rancangan program yang dibuat oleh *programmer* yang digunakan untuk melakukan suatu tugas khusus, sehingga dapat meminimalkan pekerjaan manusia. Program aplikasi adalah suatu program yang siap pakai atau dibuat untuk melaksanakan suatu fungsi bagi pengguna. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok permasalahan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu [1].

Adapun aplikasi *software* yang dirancang untuk penggunaan praktisi khusus diklasifikasi menjadi 2 yaitu

- 1) Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu
- 2) Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu yang lebih kompleks dan memerlukan proses yang banyak

Dari kedua pengertian di atas dapat disimpulkan bahwa aplikasi adalah sekumpulan perintah atau kode yang disusun secara terstruktur dan sistematis untuk menjalankan suatu perintah yang diberikan oleh manusia melalui komponen atau *hardware* computer yang digunakan oleh manusia dalam menjalankan program aplikasi, dengan demikian bisa membantu manusia untuk memberikan solusi dari apa yang diinginkan [1].

#### b. Bengkel

Bengkel adalah sebuah bangunan yang menyediakan ruang dan peralatan untuk melakukan konstruksi atau manufaktur, dan/atau memperbaiki benda. Pada kondisi tertentu, kendaraan bermotor memerlukan perawatan atau perbaikan. Untuk menjaga umur pakai kendaraan lebih panjang atau paling tidak sama dengan umur pakai yang telah diprediksikan dan dirancang oleh pabrik pembuat, maka perlu dilakukan perawatan dan perbaikan. Meskipun demikian, perawatan dan perbaikan kendaraan bukan merupakan pekerjaan mudah. Hal tersebut memerlukan pengetahuan khusus [2].

#### c. Tambal ban

Tambal adalah memperbaiki sesuatu (rumah, ban, dan lain-lain) yang tidak menyeluruh (hanya bagian yang rusak)[3]. Sedangkan ban berarti benda bulat dari karet yang dipasang melingkar pada roda (sepeda, mobil, dan sebagainya) [4]. Ban terdiri dari 3 macam, diantaranya : [5]

- 1) Ban dalam adalah ban yang berada di dalam lebih tipis, tempat udara dipompakan.
- 2) Ban luar adalah ban yang berada di luar, tebal dan keras sebagai lapis penutup ban dalam.
- 3) Ban tanpa tube adalah ban yang dirancang tanpa menggunakan ban dalam

#### d. Stasiun Pengisian Bahan Bakar Umum (SPBU)

SPBU adalah tempat di mana kendaraan bermotor bisa memperoleh bahan bakar. Di Indonesia SPBU menyediakan beberapa jenis bahan bakar minyak dan gas, misalnya : [6]

- 1) Bensin dan beragam varian produk bensin
- 2) Bahan bakar diesel
- 3) E85 (Etanol dan bensin)
- 4) LPG dalam berbagai ukuran tabung
- 5) Minyak tanah
- 6) Compressed Natural Gas (CNG)

#### e. Google Maps API

Google Maps API merupakan pengembangan teknologi dari google yang digunakan untuk menanamkan Google Map di suatu aplikasi yang tidak dibuat oleh Google. Google Maps API adalah suatu *library* yang berbentuk *Javascript* yang berguna untuk memodifikasi peta yang ada di Google Maps sesuai kebutuhan. Dalam perkembangannya Google Maps API diberikan kemampuan untuk mengambil gambar peta statis. Melakukan *geocoding*, dan memberikan penuntun arah. Google Maps API bersifat gratis untuk publik. [7].

#### f. Global Positioning System (GPS)

*Global Positioning System* (GPS) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyalarsan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Sistem yang serupa dengan GPS antara lain GLONASS Rusia, Galileo Uni Eropa, IRNSS India [8]. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS *reciever* yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi diubah menjadi titik yang dikenal dengan nama *Way-point* nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian di layar pada peta elektronik. Sejak tahun 1980, layanan GPS yang dulunya hanya untuk keperluan militer mulai terbuka untuk publik. Uniknya, walau satelit-satelit tersebut berharga ratusan juta dolar, namun setiap orang dapat menggunakannya dengan gratis [8].

#### g. Sistem Operasi Android

Android adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android

menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak [9].

#### h. Metode Dijkstra

Algoritma Dijkstra ditemukan oleh Edsger Wybe Dijkstra pada tahun 1959. Algoritma Dijkstra merupakan algoritma yang sering digunakan dalam pencarian rute terpendek. Dalam mencari solusi, Algoritma Dijkstra menggunakan prinsip greedy, yaitu mencari solusi optimum pada setiap langkah yang dilewati, dengan tujuan untuk memperoleh solusi terbaik pada langkah selanjutnya yang akan mengarah pada solusi optimum. Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya [10]. Langkah pertama tentukan mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ketitik selanjutnya tahap demi tahap. Adapun langkah-langkah kerja metode Dijkstra adalah sebagai berikut: [10]

- 1) Beri nilai bobot (jarak) untuk setiap titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain.
- 2) Set semua node "belum terjamah" dan set node awal sebagai node keberangkatan.
- 3) Dari node keberangkatan pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan sebagai contoh jika titik keberangkatan A ke B memiliki bobot 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi  $6+2 = 8$ . Jika jarak ini lebih kecil dari jarak sebelumnya hapus data lama, simpan ulang data jarak dengan jarak yang baru.
- 4) Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang terjamah sebagai node terjamah. Node terjamah tidak akan pernah dicek kembali, jarak yang disimpan adalah jarak terakhir dan yang minimal.
- 5) Set node belum terjamah dengan jarak terkecil sebagai node keberangkatan selanjutnya dengan kembali ke langkah 3.

#### i. Formula Haversine

Formula Haversine merupakan suatu cara penentuan jarak dari titik koordinat berdasarkan posisi garis lintang dan garis bujur atau dalam aplikasinya kini menggunakan *Latitude* dan *Longitude* pada Google Map, hasil dari perhitungan dengan Formula Haversine adalah jarak dari kedua titik yang dapat digambarkan dalam peta menggunakan fasilitas API atau *Application Programming Interface* pada Google Map [11].

Berikut ini adalah rumus dari Formula Haversine antara lain: [12]

$$\begin{aligned} \Delta \text{lat} &= \text{lat2} - \text{lat1} \\ \Delta \text{long} &= \text{long2} - \text{long1} \\ A &= \sin^2(\Delta \text{lat}/2) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2(\Delta \text{long}/2) \\ c &= 2 \arctan2(\sqrt{A}, \sqrt{1-A}) \\ d &= R \cdot c \end{aligned}$$

Dimana :

$$\begin{aligned} R &= \text{jari-jari bumi sebesar } 6371(\text{km}) \\ \Delta \text{lat} &= \text{besaran perubahan latitude} \\ \Delta \text{long} &= \text{besaran perubahan longitude} \\ C &= \text{kalkulasi perpotongan sumbu} \\ d &= \text{jarak (km)} \\ 1 \text{ derajat} &= 0.0174532925 \text{ radian} \end{aligned}$$

#### j. Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) untuk mengembangkan aplikasi Android. Android Studio berbasis pada "IntelliJ IDEA" Java-IDE dari JetBrains dan diperkenalkan oleh Google. Android Studio ini diumumkan pada Mei 2013. Android Studio direncanakan akan menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi pada platform Android [13].

### 3. Metode Penelitian

Berikut adalah tahapan-tahapan pada penelitian yang dilakukan

#### a. Metode Pengumpulan Data

Adapun pengumpulan data yang dilakukan pada penelitian ini adalah sebagai berikut :

- 1) Wawancara yaitu mewawancarai beberapa masyarakat pengendara sepeda motor untuk menggali informasi yang mendukung permasalahan penelitian.
- 2) Studi pustaka yaitu mencari referensi dari penelitian sebelumnya dan mencari teori-teori pendukung dari metode yang digunakan dalam penelitian.

## b. Analisis Data

Pada tahap ini dilakukan analisis terhadap semua data yang telah dikumpulkan, dan kemudian dilakukan proses pemilahan data yang sesuai untuk digunakan dalam aplikasi yang akan dirancang. Proses analisis terbagi menjadi 2 yaitu:

- 1) Analisis proses yang menggambarkan cara kerja dari algoritma yang dipakai. Metode yang digunakan pada penelitian ini adalah metode Dijkstra dan Formula Haversine. Formula Haversine digunakan untuk mengukur jarak antara posisi pengguna terhadap lokasi bengkel tambal ban dan SPBU, sedangkan metode Dijkstra digunakan untuk memberikan rekomendasi bengkel tambal ban dan SPBU terdekat dari posisi pengguna berdasarkan perhitungan jalur paling dekat. Penerapan metode Dijkstra dan Haversine akan ditunjukkan melalui contoh kasus sederhana.

Diasumsikan terdapat 3 lokasi SPBU yaitu lokasi B, lokasi D, dan lokasi E. Sebagai contoh dihitung jarak antara posisi pengguna C terhadap lokasi D.

Posisi pengguna C dengan koordinat:

- a) Latitude ( $\varphi_1$ ) = 3,5913294 derajat dikonvert ke dalam radian (0,062680525535 radian)
- b) Longitude ( $\lambda_1$ ) = 98,6991185 derajat dikonvert ke dalam radian (1,722624586637 radian)
- c) Posisi lokasi D dengan koordinat :
- d) Latitude ( $\varphi_2$ ) = 3,616161 derajat dikonvert ke dalam radian (0,063113915732 radian)
- e) Longitude ( $\lambda_2$ ) = 98,671368 derajat dikonvert ke dalam radian (1,72214024904 radian)

Rumus untuk merubah derajat ke radian : **variable X**  $\frac{\pi}{180}$

$$\varphi_1 = 0,062680525535$$

$$\varphi_2 = 0,063113915732$$

$$\lambda_1 = 1,722624586637$$

$$\lambda_2 = 1,72214024904$$

$$\varphi_2 - \varphi_1 = 0,063113915732 - 0,062680525535 = 0,000433393$$

$$\lambda_2 - \lambda_1 = 1,72214024904 - 1,722624586637 = -0,000484338$$

$$d = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

$$2r \arcsin \left( \sqrt{\sin^2 \left( \frac{0,000433393}{2} \right) + \cos(0,062680525535) \cos(0,062680525535_2) \sin^2 \left( \frac{-0,000484338}{2} \right)} \right)$$

$$2r \arcsin$$

(

$$\sqrt{\sin^2(0,000216697) + \cos(0,062680525535) \cos(0,062680525535_2 + \sin^2(-0,000242169))}$$

$$2r \arcsin (\sqrt{4,6957589073995 * 10^{-8} + 5,8414120036643 * 10^{-8}})$$

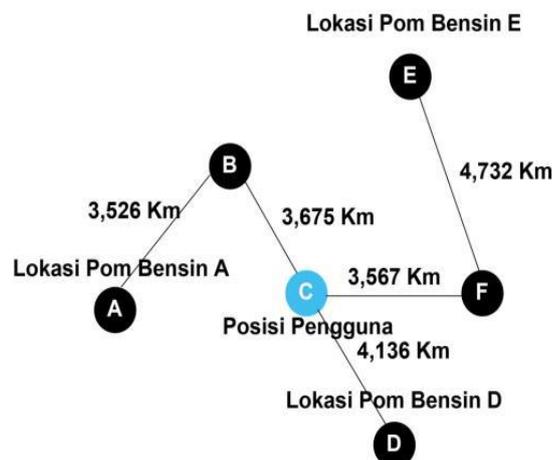
$$2r \arcsin (0,00032461008781402$$

$$2 * 6371000 * 0,00032461009351481$$

$$\mathbf{4.136,1818115657 \text{ meter}}$$

Jadi jarak antara pengguna C terhadap lokasi D adalah 4.136,1818115657 meter. Kemudian dikonversikan menjadi kilometer menjadi 4,136 km agar lebih mudah untuk dibandingkan. Proses perhitungan jarak akan dilanjutkan pada lokasi berikutnya.

Setelah ketiga jarak lokasi tersebut didapatkan, maka selanjutnya adalah menerapkan metode Dijkstra untuk merekomendasi SPBU dengan jalur dan jarak terdekat dari posisi pengguna.



Gambar 1. Contoh kasus rekomendasi SPBU terdekat Metode Dijkstra

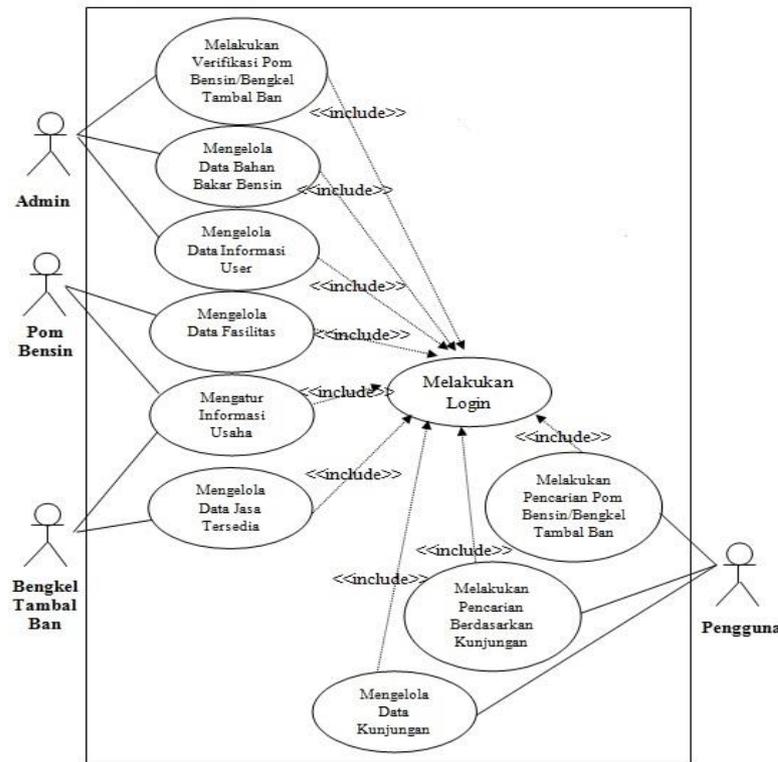
Pada gambar 1, ditunjukkan bahwa terdapat beberapa *vertex* yakni *vertex* A yang merupakan lokasi A, *vertex* C atau *vertex* sumber yang merupakan lokasi pengguna, *vertex* D yang merupakan lokasi D, dan *vertex* E yang merupakan lokasi E. Masing-masing *vertex* terhubung dengan *vertex* lainnya dan memiliki nilai bobot *edge* yang diasumsikan sebagai jarak yang telah dihitung dengan Formula Haversine. Sesuai dengan metode Dijkstra, maka akan dilakukan proses iterasi hingga mendapatkan rekomendasi lokasi SPBU yang paling terdekat dari posisinya.

Tabel 1. Perhitungan dan Rekomendasi Jarak Terdekat Metode Dijkstra

Iterasi	Vertex Sumber	Vertex Yang Dikunjungi	Perhitungan Jarak (km)
1	C	C → B → A Lokasi A	3,675 + 3,526 = 7,201
2	C	C → F → E Lokasi E	3,567 + 4,732 = 8,299
3	C	C → D Lokasi D	4,136

Berdasarkan tabel 1, dapat dilihat bahwa ketika melakukan iterasi pencarian jalur terdekat, maka didapatkan bahwa lokasi D yang paling dekat dengan jarak 4,136 Km. Maka secara otomatis aplikasi akan memberikan rekomendasi jalur dari lokasi C ke lokasi D sehingga lokasi D menjadi lokasi yang paling dekat dari lokasi pengguna dan menjadi rekomendasi lokasi tujuan.

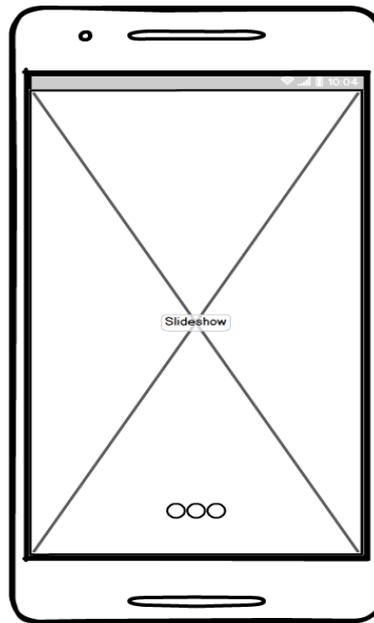
- 2) Analisis kebutuhan fungsional menggunakan *Use Case Diagram* yang akan menggambarkan hubungan pengguna dengan sistem.



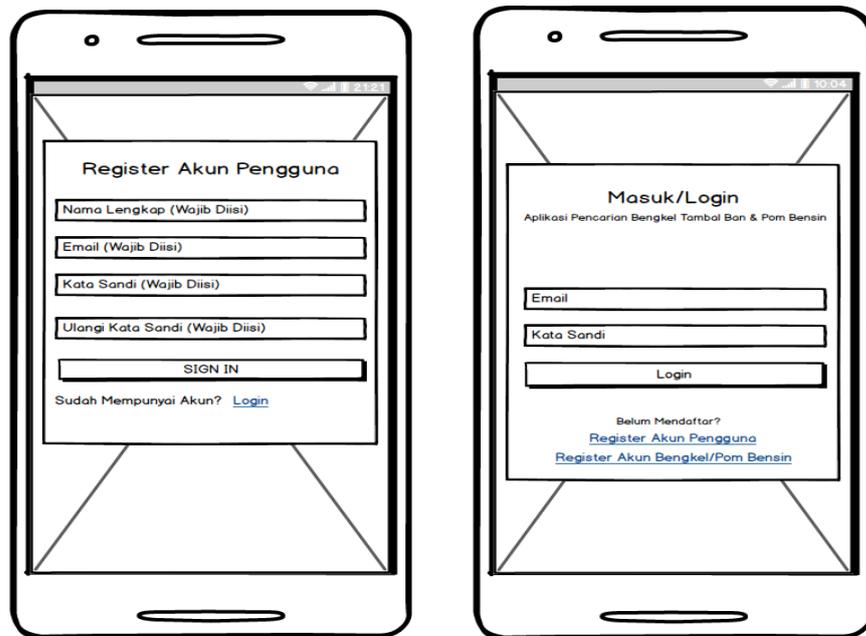
Gambar 2. Use Case Diagram Sistem

c. Perancangan dan Pembangunan Aplikasi

Perancangan antar muka menggunakan *balsamiq Mockups* yang akan diimplementasikan ke dalam program.



Gambar 3 Tampilan Antar Muka Awal Untuk Pengguna



Gambar 4. Tampilan Antar Muka Untuk Daftar dan Login ke Aplikasi

#### 4. Hasil Penelitian

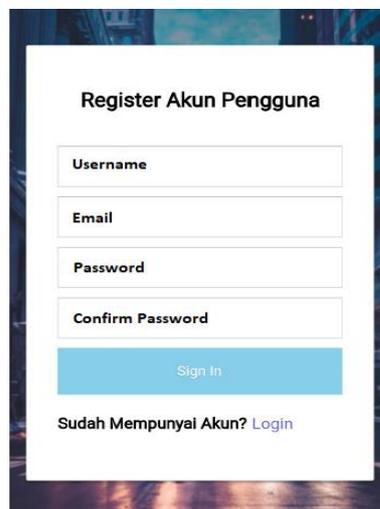
Pada tahapan ini, seluruh hasil penelitian yang dilakukan berupa dibangunnya sebuah aplikasi pencarian bengkel tambal ban dan SPBU berbasis Android ditampilkan dalam bentuk *screenshot* hasil :

- a. Hasil Tampilan Awal Untuk *User* Pengunjung



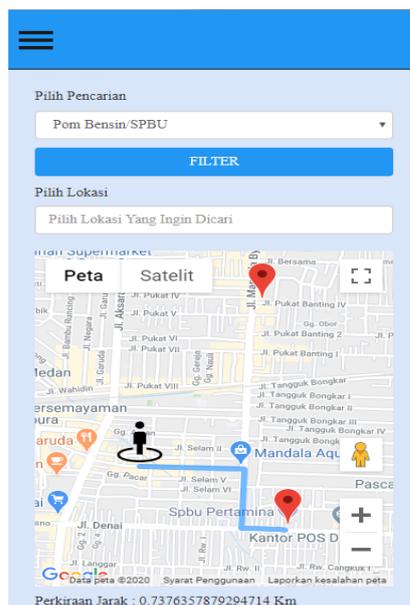
Gambar 5. Tampilan awal berisikan *slideshow* pengenalan aplikasi.

- b. Tampilan untuk pengguna melakukan pendaftaran ke aplikasi



Gambar 6. Tampilan Untuk Pengguna Melakukan Pendaftaran

- c. Tampilan untuk melakukan pencarian kedua tempat tersebut dengan penerapan metode Dijkstra dan Haversine.



Gambar 7. Tampilan Untuk Pencarian Bengkel Tambah Ban atau SPBU

## 5. Kesimpulan

Berdasarkan analisis dan pengujian yang telah dilakukan dan penelitian ini, maka kesimpulan yang dapat diambil adalah :

- a. Aplikasi yang dibangun memudahkan pengguna dalam melakukan pencarian bengkel tambal ban dan SPBU melalui fitur pencarian dengan bantuan pemetaan dari Google Map.
- b. Penerapan metode Dijkstra dan Haversine mampu merekomendasikan bengkel tambal ban ataupun SPBU terdekat dari posisi pengguna.
- c. Sistem yang dirancang mampu memberikan hasil pemetaan lokasi terdekat beserta informasi mengenai lokasi yang dituju dengan lengkap

## 6. Daftar Pustaka

- [1]. G. M. Marakas dan J. A. O'Brien, *Pengantar Sistem Informasi*, Edisi 16, Jakarta: Salemba Empat, 2017.
- [2]. Daryanto, *Peralatan Bengkel Otomotif*, Tangerang: Tsmart, 2018.
- [3]. KBBI, "Arti kata tambal", *KBBI* [Online]. Tersedia: <https://kbbi.web.id/tambal> [Diakses: 14 Januari 2020].
- [4]. KBBI, "Arti kata ban", *KBBI* [Online]. Tersedia: <https://kbbi.web.id/ban> [Diakses: 14 Januari 2020].
- [5]. Buntarto, *Sistem Ban & Roda*, Yogyakarta: Pustaka Baru Press, 2015.
- [6]. [https://id.wikipedia.org/wiki/Stasiun\\_pengisian\\_bahan\\_bakar](https://id.wikipedia.org/wiki/Stasiun_pengisian_bahan_bakar)
- [7]. M. A. Suryawan dan M. Mukimin, "Penerapan Google Maps API pada Aplikasi Android untuk Mengetahui Lokasi Situs Sejarah Benteng Keraton Buton", *Seminar Nasional APTIKOM (SEMNASITIK)*, 2019.
- [8]. S. Muis, *Global Positioning System Sebuah Pengantar Untuk Metode, Sistem dan Perancangan Sistem*, Edisi 2, Surabaya: Teknosain, 2018.
- [9]. Y. Supardi dan I. Setiawan, *Mudah dan Cepat Membuat Program Skripsi dan Tugas Akhir dengan Android*, Jakarta: Elex Media Komputindo, 2018.
- [10]. N. Azizah dan D. Mahendra, "Geolocation dengan Metode Dijkstra untuk Menentukan Jalur Terpendek Lokasi Peribadatan", *Jurnal Sistem Informasi Bisnis*, vol. 2, 2017.
- [11]. A. Maulana, A. Solichin, M. Syafrullah, "Penerapan Metode Haversine Pada Sistem Informasi Geografis Untuk Penentuan Lokasi Pembangunan Menara Telekomunikasi Pada Kota Tangerang", *Indonesian Journal on Software Engineering*, vol. 4, no. 1, 2018.
- [12]. A. M. Abdillah, Rianto, N. I. Kurniati, "Penerapan Metode Haversine pada Aplikasi Layanan Perbaikan Kendaraan Berbasis Location Based", *Jurnal Informatika (JUITA)*, vol. 7, no. 2, November 2019.
- [13]. Herlinah dan Musliadi, *Pemrograman Aplikasi Android dengan Android Studio, Photoshop dan Audition*, Jakarta: PT. Elex Media Komputindo, 2019.