



Implementation of Haversine Algorithm on Mapbox API for Searching Nearest Workshop Based on Mobile Devices

Implementasi Algoritma Haversine pada Mapbox API Guna Pencarian Bengkel Terdekat Berbasis Perangkat Mobile

Sushilo Prasetyo^{1*}, Umar Zaky²

¹Program Studi Informatika, Universitas Teknologi Yogyakarta, Indonesia

²Program Studi Sistem Informasi, Universitas Teknologi Yogyakarta, Indonesia

E-Mail: ¹sushiloprasetyo@gmail.com, ²umarzaky@gmail.com

Received Aug 19th 2024; Revised Sept 26th 2024; Accepted Oct 06th 2024
Corresponding Author: Sushilo Prasetyo

Abstract

The automotive industry in Indonesia has experienced rapid growth in line with the increase in per capita income and the rising demand for transportation. One critical sector within this industry is automotive workshop services, which play a vital role in vehicle maintenance and repair. While several workshop search applications have been developed, many remain web-based and lack direct communication features between customers and workshops. This study aims to develop a mobile Android-based workshop service application that implements the Haversine algorithm. The application is designed to facilitate users in finding the nearest workshop, calculating the distance, and directly communicating with the workshop through a chat feature. The results show that the application accurately calculated distances, with the nearest workshop being 3.229 kilometers away from the user's location. The use of the Haversine algorithm provides precise distance calculations with an accuracy rate of up to 99%. Thus, this study contributes to improving the quality of workshop services in Indonesia, particularly by utilizing location-based technology for enhanced user convenience and experience.

Keyword: Haversine Algorithm, Mobile Application, Workshop Services, Workshop Search

Abstrak

Industri otomotif di Indonesia mengalami pertumbuhan pesat seiring dengan meningkatnya pendapatan per kapita dan kebutuhan masyarakat akan transportasi. Salah satu sektor penting dalam industri ini adalah layanan bengkel kendaraan bermotor, yang memiliki peran krusial dalam perawatan dan perbaikan kendaraan. Meskipun sudah ada beberapa aplikasi pencarian bengkel, sebagian besar masih berbasis web dan tidak menyediakan fitur komunikasi langsung antara pelanggan dan bengkel. Penelitian ini bertujuan untuk mengembangkan aplikasi layanan bengkel berbasis *mobile* Android yang mengimplementasikan algoritma Haversine. Aplikasi ini dirancang untuk memudahkan pengguna dalam menemukan bengkel terdekat, menghitung jarak, dan berkomunikasi langsung dengan bengkel melalui fitur *chat*. Hasil penelitian menunjukkan bahwa aplikasi ini mampu menghitung jarak dengan akurasi tinggi, di mana bengkel terdekat ditemukan berjarak 3,229 kilometer dari lokasi pengguna. Penggunaan algoritma Haversine dalam aplikasi ini memberikan hasil perhitungan jarak yang akurat dengan tingkat akurasi mencapai 99%. Dengan demikian, penelitian ini berkontribusi dalam meningkatkan kualitas layanan bengkel di Indonesia, terutama dalam memanfaatkan teknologi berbasis lokasi untuk kemudahan dan kenyamanan pengguna.

Kata Kunci: Algoritma Haversine, Aplikasi *Mobile*, Layanan Bengkel, Pencarian Bengkel

1. PENDAHULUAN

Industri otomotif di Indonesia telah mengalami perkembangan pesat selama dekade terakhir, didorong oleh kelas menengah yang terus berkembang serta kebijakan pemerintah yang mendukung sektor ini. Berdasarkan data dari Gabungan Industri Kendaraan Bermotor Indonesia (GAIKINDO), Indonesia merupakan pasar otomotif terbesar di *Association of Southeast Asian Nations* (ASEAN) dengan kapasitas produksi yang terus meningkat. Stabilitas pertumbuhan ekonomi dan kebijakan pemerintah, terutama pada periode booming komoditas 2006-2015, juga memainkan peran penting dalam mendukung sektor ini [1]. Pada tahun 2017, Produk Domestik Bruto (PDB) per kapita Indonesia mencapai Rp. 51,89 juta, meningkat 8,1 persen

dibandingkan tahun sebelumnya yang sebesar Rp. 47,97 juta [2]. Peningkatan pendapatan ini telah menyebabkan perubahan gaya hidup dan meningkatnya kebutuhan akan transportasi.

Seiring dengan perkembangan industri otomotif, meningkat pula kebutuhan masyarakat terhadap layanan perawatan kendaraan bermotor, terutama bengkel [3]. Bisnis bengkel, yang menyediakan jasa perawatan dan perbaikan kendaraan, menjadi sangat penting untuk menjaga performa dan keamanan kendaraan. Beberapa penelitian terdahulu telah mengembangkan aplikasi pencarian bengkel. Misalnya, Mutmainna dan rekan-rekan membangun aplikasi berbasis Google Maps API untuk mencari bengkel di Makassar, sementara Ferawaty dan rekan-rekan mengimplementasikan Haversine Formula dan algoritme Dijkstra untuk menghitung jarak terpendek ke bengkel [4], [5]. Penelitian lain oleh Ipo dan rekan-rekan menggunakan sistem informasi geografis untuk memetakan persebaran bengkel di Pontianak [6].

Penelitian oleh Dauni dan rekan-rekan (2019) menggunakan Haversine Formula untuk melacak lokasi sekolah terdekat melalui aplikasi Android berbasis *Global Positioning System* (GPS), tetapi tanpa fitur interaksi pengguna [7]. Andreou dan rekan-rekan (2023) memodifikasi Haversine dan Vincenty Formulas untuk optimasi lintasan *Unmanned Aerial Vehicle* (UAV) di kota pintar [8]. Ali dan rekan-rekan (2020) menerapkan Haversine Formula pada sistem pelacakan kendaraan anti-maling berbasis IoT [9]. Meskipun penelitian-penelitian tersebut telah memberikan kontribusi dalam mengatasi masalah pencarian bengkel, ada beberapa kelemahan yang belum teratasi, seperti tidak adanya fitur komunikasi langsung antara pelanggan dan bengkel, serta keterbatasan dalam pemanfaatan platform Android. Oleh karena itu, pengembangan aplikasi *mobile* dengan fitur *chat*, integrasi Mapbox, dan Haversine Formula menjadi penting dan relevan.

Kelebihan penelitian ini dibandingkan penelitian sebelumnya adalah penyediaan fitur komunikasi langsung dan integrasi penuh dengan platform Android. Fitur ini memudahkan pengguna dalam mencari bengkel terdekat, menghitung jarak, serta berkomunikasi langsung dengan bengkel, yang tidak ditemukan dalam penelitian-penelitian sebelumnya. Penelitian ini berfokus pada pengembangan aplikasi yang tidak hanya menyediakan pencarian bengkel terdekat tetapi juga memfasilitasi komunikasi langsung antara pelanggan dan bengkel, yang diharapkan dapat meminimalisir potensi miskomunikasi. Aplikasi ini diharapkan menjadi solusi yang lebih efektif dalam memenuhi kebutuhan pelanggan bengkel, memberikan layanan yang lebih cepat dan transparan terkait lokasi, estimasi biaya, dan layanan tambahan seperti konsultasi suku cadang.

2. PENELITIAN YANG TERKAIT

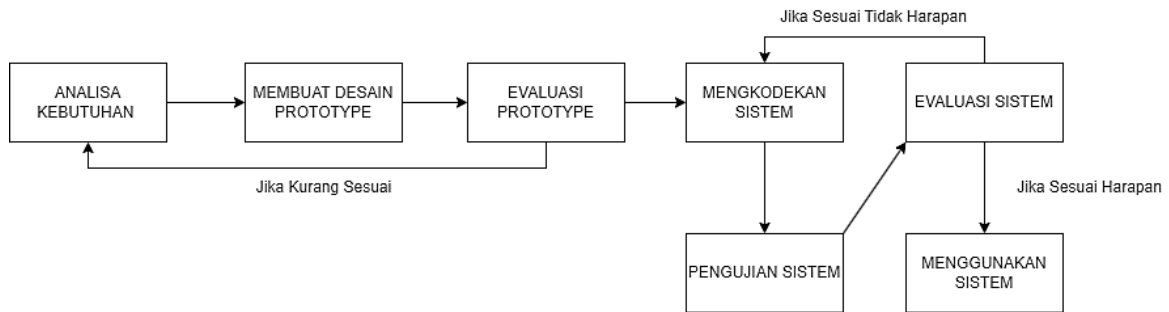
Berikut merupakan beberapa referensi penelitian yang telah dilakukan sebelumnya yang menjadi dasar dalam penelitian ini:

1. Penelitian yang berjudul "Rancang Bangun Aplikasi Pencarian Bengkel Aktif Berbasis Google Maps API di Wilayah Kota Makassar" oleh Mutmainna dan Andi Mashdarul Khair tahun 2022 menggunakan MySQL, Google Maps API, dan UML untuk mengembangkan aplikasi berbasis web yang memungkinkan pengguna menemukan bengkel aktif dengan cepat dan akurat di Kota Makassar. Hasil penelitian menunjukkan efektivitas aplikasi dalam memberikan informasi lokasi bengkel, dengan saran pengembangan berupa pembuatan versi Android untuk meningkatkan aksesibilitas [4].
2. Penelitian "Implementasi Haversine Formula dan Algoritme Dijkstra pada Aplikasi Vehicle Help Sebagai Layanan Konsultasi Pemilik Kendaraan" oleh Ferawaty dan Albert Cenderawan pada tahun 2023, memanfaatkan Android, Google Maps API, MySQL, dan GPS dengan metode Waterfall. Penelitian ini mengembangkan aplikasi "Vehicle Help" yang merekomendasikan lokasi SPBU dan bengkel servis terdekat menggunakan Haversine Formula dan algoritme Dijkstra. Hasil penelitian menunjukkan bahwa aplikasi ini memberikan rekomendasi lokasi yang akurat dan efektif untuk pengguna [5].
3. Penelitian "Rancang Bangun Sistem Informasi Geografis Persebaran Bengkel, Tambal Ban, dan Minyak Eceran di Kota Pontianak Berbasis Web" oleh Ego Suparius Ipo, Yus Sholva, dan M. Azhar Irwansyah pada tahun 2023 menggunakan Google Maps API dan Haversine Formula untuk mengembangkan sistem informasi geografis berbasis web. Sistem ini menampilkan lokasi bengkel, tambal ban, dan minyak eceran di Kota Pontianak, membantu pengendara menemukan layanan terdekat dengan informasi yang rinci [6].
4. Penelitian "Aplikasi Layanan Perbaikan Motor *Online* Berbasis Android dengan Memanfaatkan Google Maps" oleh Faiz Ataka Ubaidillah dan Anita Fira Waluyo pada tahun 2023 mengembangkan aplikasi berbasis Android dengan Google Maps API, MySQL, dan PHP menggunakan framework CodeIgniter. Aplikasi ini memungkinkan pengguna melakukan transaksi perbaikan motor secara *online* dan menemukan bengkel terdekat, dengan pengembangan sistem yang mencakup pemahaman kebutuhan pengguna dan pengujian aplikasi [10].

Perbedaan utama penelitian ini terletak pada pengembangan fitur *chat* yang memungkinkan komunikasi langsung antara pelanggan dan pihak bengkel, serta penerapan *Mapbox* untuk tampilan navigasi dan *Haversine Algorithm* untuk menghitung jarak terdekat. Fitur ini belum diterapkan dalam penelitian sebelumnya, yang juga sebagian besar belum menyediakan platform berbasis Android.

3. METODE PENELITIAN

Dalam pengembangan sistem ini akan menggunakan metode Software Development Life Cycle Prototype atau SDLC Prototype. SDLC ini cocok digunakan dalam pengembangan suatu sistem jangka pendek yang membutuhkan adaptasi perubahan yang cepat. Tahapan dari SDLC Prototype dapat dilihat pada Gambar 1.



Gambar 1. Metode Penelitian

Dalam tahap awal pada pengembangan ini adalah analisa kebutuhan, yaitu kebutuhan dari pengguna dikumpulkan secara mendetail. Tahap kedua dilanjutkan membuat desain prototype, yaitu menghasilkan model atau tampilan visual yang dapat diuji oleh pengguna. Tahap ketiga adalah evaluasi prototype, yaitu memungkinkan adanya perubahan dan perbaikan sebelum pengembangan lebih lanjut. Tahap keempat adalah mengkodekan sistem, yaitu jika prototipe telah disetujui oleh pengguna, pengembang mulai mengkodekan atau mengimplementasikan sistem sesuai dengan spesifikasi yang ada pada prototipe. Tahap kelima adalah pengujian sistem yaitu, pengujian terhadap perangkat lunak yang dikembangkan, pengujian dilakukan untuk memastikan sistem berfungsi sesuai spesifikasi dan bebas dari kesalahan. Tahap keenam adalah evaluasi sistem yaitu, jika perangkat lunak sudah sesuai, pengembangan lanjut ke tahap menggunakan sistemn namun jika belum, pengembang memperbaiki bagian yang perlu, lalu mengulangi pengujian hingga memenuhi kriteria yang diinginkan. Tahap ketujuh adalah menggunakan sistem yaitu, perangkat lunak yang telah diuji dan disetujui siap digunakan oleh pengguna. Setelah diintegrasikan ke lingkungan kerja, sistem mulai digunakan dalam operasional, dan perbaikan serta peningkatan dilakukan sesuai kebutuhan.

3.1 Pengumpulan Data

Pada penelitian ini, data yang digunakan meliputi data sekunder yang diperoleh dari beberapa bengkel, berupa informasi nama, jenis bengkel, serta koordinat lokasi (*latitude* dan *longitude*) yang ditampilkan dalam Tabel 1 dan Tabel 2.

Tabel 1. Data Bengkel

No.	Nama Bengkel	Latitude	Longitude
1	Bintang Jaya Motor	-7.7428494	110.3628454
2	Indo Mobil Motor	-7.7542034	110.3619505
3	Mutiara Motor (Pak Totok)	-7.7538586	110.3549934
4	Bengkel Motor DELTA Interlokal	-7.7421233	110.4005395
5	Bengkel Wijayanto Motor	-7.7270269	110.3655030

Tabel 2. Data Sampel

No.	Data Sampel	Latitude	Longitude
1	A	-7.782952	110.366264

Data ini diperoleh melalui observasi, studi literatur, dan pencarian internet. Observasi dilakukan untuk memahami kondisi di lapangan dan mengamati masalah yang dihadapi pelanggan serta pihak bengkel. Studi literatur bertujuan untuk mendapatkan dasar teori yang relevan terkait konsep implementasi algoritma Haversine. Selain itu, internet searching digunakan untuk mencari data tambahan yang mendukung penelitian. Pengumpulan data berlangsung selama satu hari, dari 29 Maret hingga 10 April 2024, meliputi informasi mengenai nama, jenis, dan lokasi bengkel.

3.2 Algoritma Haversine

Algoritma Haversine adalah salah satu metode yang digunakan untuk menghitung jarak antara dua titik pada permukaan bola, seperti bumi, berdasarkan koordinat lintang dan bujur. Asumsi utama dalam penggunaan algoritma ini adalah bahwa bumi memiliki bentuk bulat sempurna dengan radius rata-rata sekitar 6371 km [11]. Algoritma ini secara umum digunakan dalam navigasi karena kemampuannya memberikan perhitungan jarak yang lebih akurat pada permukaan bola dibandingkan dengan metode Euclidean yang menggunakan ruang

datar [12]. Dalam penerapannya, Haversine menghindari kesalahan yang muncul ketika menghitung jarak pada sudut kecil, yang merupakan kelemahan dari rumus kosinus bola biasa [13]. Hal ini membuat Haversine menjadi pilihan utama dalam konteks aplikasi yang memerlukan presisi tinggi pada jarak yang pendek, seperti yang ditemukan pada pengembangan sistem informasi geografis berbasis lokasi, termasuk pencarian bengkel terdekat [14]. Secara matematis, formula Haversine dituliskan pada persamaan 1.

$$d = 2R \cdot \sin^{-1}\left(\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)\right) \quad (1)$$

Dimana:

- d : Jarak antara dua titik (dalam kilometer),
- R : Jari-jari bumi (6371 km),
- ϕ_2 & ϕ_1 : Lintang dari dua titik, dan
- $\Delta\phi$ dan $\Delta\lambda$: Perbedaan lintang dan bujur antara dua titik tersebut (Maria et al., 2020).

Algoritma Haversine diaplikasikan secara luas dalam berbagai studi yang melibatkan perhitungan jarak, mulai dari penentuan lokasi optimal fasilitas publik [15] hingga integrasi dengan algoritma lain seperti DBSCAN untuk menentukan pusat lokasi fisik internet [11]. Algoritma Haversine telah digunakan dalam berbagai konteks studi yang melibatkan perhitungan jarak geografis antar dua titik di permukaan bumi. Dengan memanfaatkan perbedaan lintang dan bujur serta mengasumsikan bumi sebagai bola sempurna dengan jari-jari 6371 km, algoritma ini dapat menghitung jarak secara akurat, baik untuk jarak pendek maupun menengah. Galindo-Muro et al. (2023) menerapkan algoritma Haversine untuk mengoptimalkan rute pengiriman kendaraan listrik di lingkungan perkotaan, di mana jarak dihitung untuk meminimalkan konsumsi energi dan waktu tempuh. Hal ini penting dalam pengiriman instan, di mana ketepatan waktu sangat krusial, dan algoritma ini memastikan rute yang dipilih efisien berdasarkan jarak dan kebutuhan energi kendaraan [16].

Selain itu, Jabid et al. (2023) menggunakan Haversine dalam sistem komunikasi antar kendaraan (*device-to-device communication*) yang dirancang untuk mengurangi risiko kecelakaan. Dengan menghitung jarak antar kendaraan secara *real-time*, sistem ini dapat mengirim peringatan jika jarak antar kendaraan terlalu dekat, yang dapat mengurangi kecelakaan di jalan raya [17]. Penggunaan Haversine juga terlihat dalam studi Maria et al. (2020) yang mengukur jarak terdekat untuk menentukan lokasi optimal fasilitas umum, serta Helmi et al. (2022) yang mengintegrasikan algoritma ini dengan DBSCAN untuk menentukan pusat lokasi fisik internet. Algoritma ini terbukti efisien dalam berbagai konteks berbasis lokasi, baik untuk transportasi, komunikasi, maupun optimasi lokasi fasilitas [11], [15].

4. HASIL DAN PEMBAHASAN

4.1. Tahap Perhitungan Jarak dengan Formula Haversine

Dimana diasumsikan terdapat 5 lokasi bengkel yaitu, lokasi A sebagai Bintang Jaya Motor, lokasi B sebagai Indo Mobil Motor, lokasi C sebagai Mutiara Motor (Pak Totok), lokasi D sebagai Bengkel Motor DELTA Interlokal, lokasi E sebagai Bengkel Wijayanto Motor dan lokasi P sebagai lokasi titik awal pengguna. Sebagai contoh dihitung jarak antar posisi pengguna P dengan bengkel B.

Langkah awal mengubah atau mengkonversi nilai derajat ke bentuk radian dengan rumus $radian = derajat \times \frac{\pi}{180}$

$$Latitude1 = 7.7829523 \times \frac{\pi}{180} = -0.1358381432$$

$$Latitude2 = 7.7829523 \times \frac{\pi}{180} = -0.1353363802$$

$$Longitude1 = 7.7829523 \times \frac{\pi}{180} = 1.926254697$$

$$Longitude2 = 7.7829523 \times \frac{\pi}{180} = 1.926179405$$

Setelah melakukan konversi nilai derajat ke bentuk radian, dilanjutkan dengan menghitung perbandingan nilai *latitude* dan *longitude*.

$$\Delta lat2 - lat1 = -0.1353363802 - (-0.1358381432) = 0.000501763$$

$$\Delta lon2 - lon1 = 1.926179405 - 1.926254697 = -0.000075292$$

Lalu dilanjutkan dengan menambahkan nilai yang telah dihitung sebelumnya kedalam formula haversine.

$$\alpha = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat1) \times \cos(lat2) \times \sin^2\left(\frac{\Delta lon}{2}\right)$$

$$\alpha = \sin^2\left(\frac{-0.000506889}{2}\right) + \cos(-0.135264095) \times \cos(-0.135770984) \times \sin^2\left(\frac{0.000074916}{2}\right)$$

$$\alpha = 6.41936 \times 10^{-8}$$

Kemudian dilanjutkan dengan menghitung nilai c dan d

$$c = 2x \operatorname{atan}^2(\sqrt{a}, \sqrt{1-a}) = 2x \operatorname{atan}^2(0.000253381, 0.999999968)$$

$$c = 0.00050692 \text{ radian}$$

$$d = R \times c = 6371 \times 0.00050692 = 3,22958732 \text{ meter}$$

Setelah melakukan pengukuran, jarak yang terukur antara pengguna P dan lokasi B adalah sebesar 3,22958732 meter. Untuk memudahkan pemahaman dalam konteks yang lebih luas, jarak ini kemudian dikonversikan ke dalam satuan kilometer. Dengan konversi tersebut, jarak antara pengguna P dan lokasi B menjadi 3,229 kilometer.

4.2. Hasil Perhitungan Jarak

Berdasarkan data yang diperoleh mengenai jarak dari titik awal P ke beberapa lokasi tujuan, hasil analisis menunjukkan bahwa, jarak dari P ke A adalah 4,475 km, jarak dari P ke B adalah 3,229 km, jarak dari P ke C adalah 3,465 km, jarak dari P ke D adalah 5,905 km, jarak dari P ke E adalah 6,219 km.

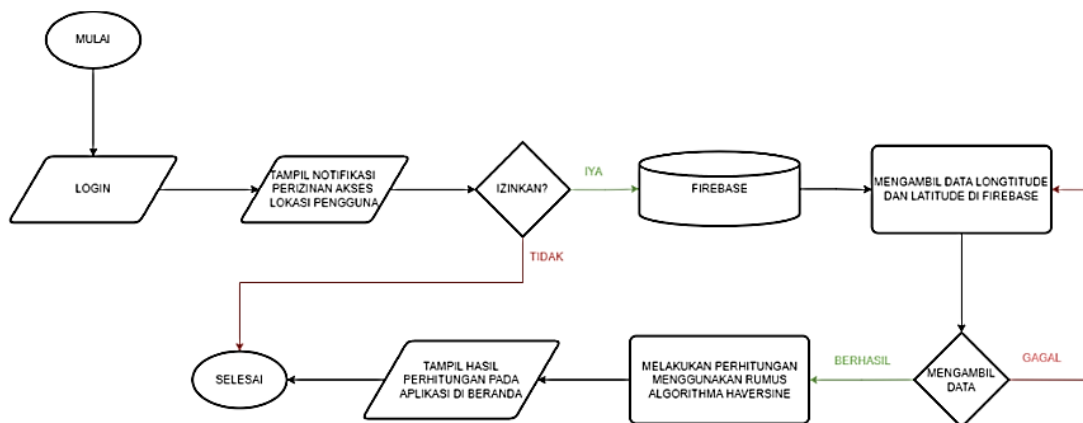
Tabel 3. Hasil Perhitungan dari 5 Lokasi Bengkel

Titik Awal	Tujuan	Jarak (Km)
P	A	4,475 km
P	B	3,229 km.
P	C	3,465 km
P	D	5,905 km
P	E	6,219 km

Dengan mempertimbangkan data jarak yang ada, dapat disimpulkan bahwa lokasi B merupakan tujuan terdekat dari titik awal P dengan jarak terukur sebesar 3,229 km. Ini menunjukkan bahwa dari semua lokasi tujuan yang terdaftar, B adalah yang paling dekat dari P, dan informasi ini penting untuk perencanaan rute atau penentuan prioritas kunjungan berdasarkan jarak tempuh. Dimana jarak yang telah dihasilkan melalui perhitungan secara manual sebelumnya akan menjadi keakurasian dari perhitungan jarak melalui aplikasi.

4.3. Analisa dan Perancangan Sistem

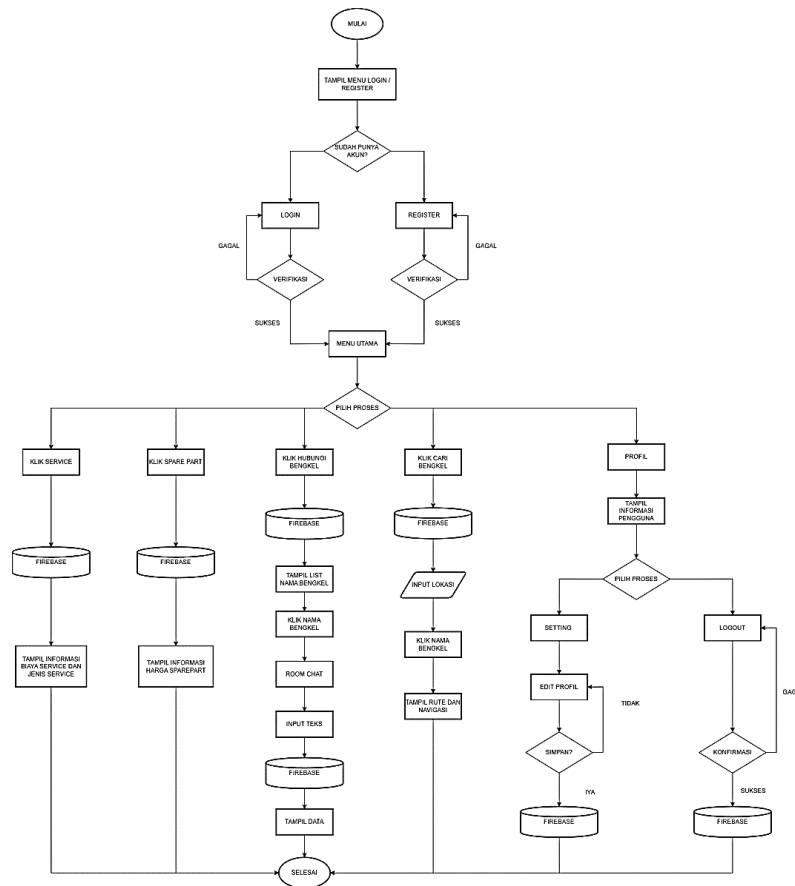
Proses perhitungan rumus algoritma Haversine divisualisasikan ke dalam *Flowchart* pada Gambar 2, yang menggambarkan alur sistem aplikasi yang dimulai dengan meminta izin lokasi pengguna. Jika izin diberikan, sistem akan mengakses data koordinat *Longitude* dan *Latitude* dari *Firebase*. Setelah data berhasil diambil, sistem menghitung jarak antara lokasi pengguna dan lokasi bengkel yang tersimpan di *Firebase* menggunakan rumus Haversine. Hasil perhitungan tersebut kemudian ditampilkan di aplikasi pada bagian bengkel terdekat di halaman beranda.



Gambar 2. Flowchart Diagram Proses Perhitungan Rumus Algoritma Haversine

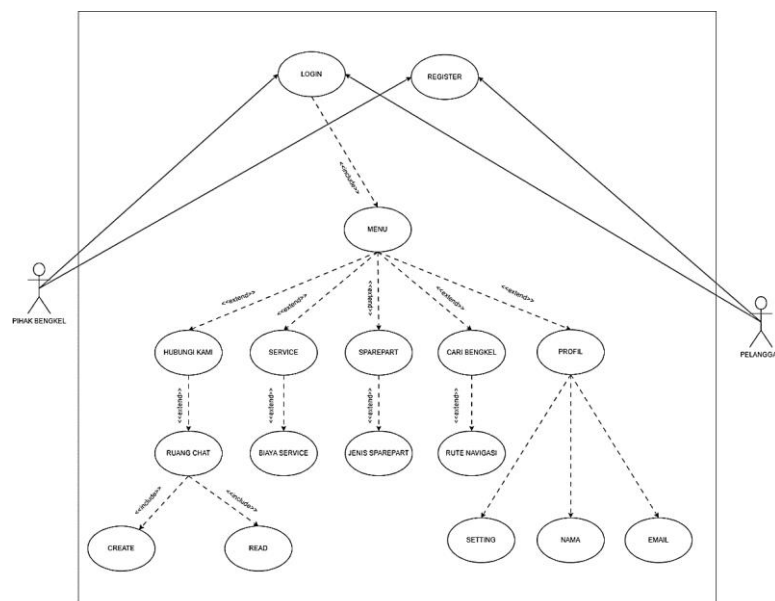
Dalam penelitian ini, analisis dan perancangan sistem aplikasi layanan bengkel meliputi beberapa aspek penting. Kebutuhan fungsional mencakup proses utama yang harus ada, seperti *login* dan registrasi untuk pelanggan dan pihak bengkel, pengiriman pesan, serta pendaftaran dan pengelolaan data oleh admin. Sistem juga harus menyediakan rute ke bengkel dan komunikasi antara pengguna dan admin. Kebutuhan non-fungsional mencakup perangkat keras dan perangkat lunak yang dibutuhkan, termasuk spesifikasi laptop, sistem operasi, dan perangkat lunak pengembangan [18].

Perancangan konseptual melibatkan pembuatan berbagai diagram untuk menggambarkan alur sistem dan interaksi antar aktor [19].



Gambar 3. Flowchart Diagram Aplikasi

Flowchart diagram menunjukkan alur proses di mana pengguna harus login atau registrasi terlebih dahulu, kemudian dapat memilih fitur seperti melihat informasi layanan servis, spare part atau suku cadang, kontak bengkel atau menghubungi pihak bengkel, mencari bengkel terdekat lalu mendapatkan rute ke arah bengkel, dan mengelola profil pengguna. Setiap fitur dirancang untuk mempermudah akses informasi dan meningkatkan interaksi antara pengguna dengan bengkel.



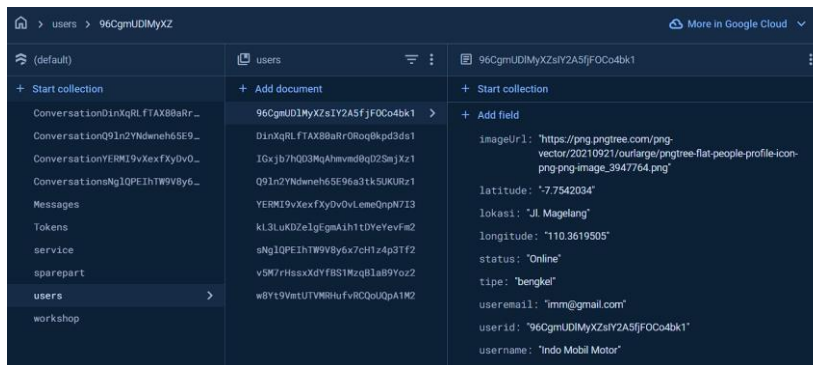
Gambar 4. Use Case Diagram Aplikasi

Use Case Diagram memetakan aktivitas yang dapat dilakukan oleh pelanggan dan bengkel yang memudahkan interaksi antara bengkel dan pelanggan melalui fitur registrasi dan login. Setelah login, pengguna dapat mengakses berbagai layanan seperti komunikasi dengan bengkel, mendapatkan informasi servis, mendapatkan informasi ketersediaan spare part, pencarian bengkel terdekat, serta pengelolaan profil. Aplikasi ini dirancang untuk meningkatkan kemudahan komunikasi, akses informasi, dan efisiensi layanan antara pelanggan dan bengkel.

Activity Diagram menggambarkan alur kerja untuk fitur-fitur seperti pencarian bengkel terdekat. Dimana pelanggan dapat melihat bengkel terdekat, lalu sistem akan menampilkan list nama bengkel, jarak dan icon navigasi ke arah bengkel, lalu pelanggan dapat memilih icon navigasi, lalu sistem akan mencari rute terdekat ke arah bengkel, kemudian sistem akan menampilkan rute dan pelanggan dapat memilih memulai navigasi dan sistem akan menampilkan rute navigasi ke arah bengkel.

Sequence Diagram menyajikan urutan interaksi sistem dari chat. Dimana pelanggan dapat melihat bengkel terdekat, lalu sistem akan menampilkan list nama bengkel, jarak dan icon navigasi ke arah bengkel, lalu pelanggan dapat memilih icon navigasi, lalu sistem akan mencari rute terdekat ke arah bengkel, kemudian sistem akan menampilkan rute dan pelanggan dapat memilih memulai navigasi dan sistem akan menampilkan rute navigasi ke arah bengkel.

Perancangan fisik mencakup struktur backend menggunakan *Firebase* untuk menyimpan data penting dan antarmuka pengguna aplikasi serta *website* admin. *Backend* dirancang untuk menyimpan data seperti informasi pengguna, layanan, *spare part*, dan pesan. Antarmuka pengguna meliputi halaman *login*, register, beranda, dan fitur tambahan seperti peta dan ruang obrolan. Desain ini bertujuan untuk menciptakan pengalaman pengguna yang intuitif dan efisien.



Gambar 5. *Firebase*

4.4. Implementasi Algoritma

Implementasi Algoritma Haversine pada aplikasi ini digunakan untuk menghitung jarak terdekat antara pelanggan dan bengkel menggunakan koordinat geografis, memastikan informasi lokasi yang akurat.

Algoritma Haversine

```

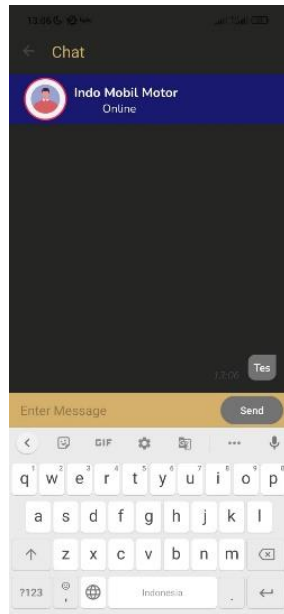
fun getDistance(currentLatitude: Double?,
                currentLongitude: Double?,
                destinationLatitude: Double?,
                destinationLongitude: Double?): Double {
    val earthRadius = 3958.75
    val dLatitude = Math.toRadians(destinationLatitude!! - currentLatitude!!)
    val dLongitude = Math.toRadians(destinationLongitude!! - currentLongitude!!)
    val a = Math.sin(dLatitude / 2) * Math.sin(dLatitude / 2) +
            Math.cos(Math.toRadians(currentLatitude)) *
            Math.cos(Math.toRadians(destinationLatitude)) *
            Math.sin(dLongitude / 2) * Math.sin(dLongitude / 2)
    val c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a))
    val dDistance = earthRadius * c
    val meterConversion = 1609
    val myDistance = dDistance * meterConversion
    return Math.floor(myDistance / 1000)
}

```

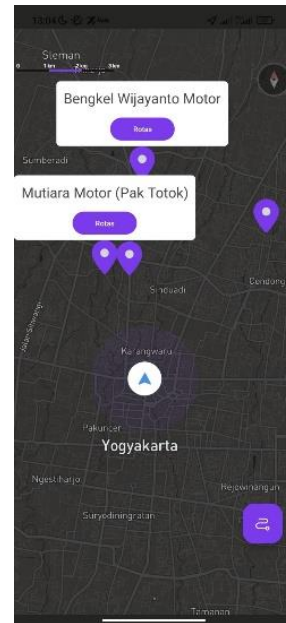
4.5. Aplikasi Mobile untuk Pelanggan dan Pihak Bengkel

Aplikasi *mobile*, dikembangkan dengan Android Studio, dirancang untuk platform Android dan memfasilitasi komunikasi antara pelanggan dan pihak bengkel melalui fitur *chat*. Tampilan *Login* memungkinkan pengguna untuk masuk dengan *username* dan *password*, serta mendaftar dengan informasi yang sesuai untuk pelanggan dan bengkel. Tampilan *Register* mengharuskan pelanggan untuk mengisi

username, *email*, dan *password*, sementara bengkel juga harus mengisi informasi lokasi (*longitude* dan *latitude*) [20]. Tampilan Beranda menyajikan menu layanan, spare part, dan pencarian bengkel, serta fitur navigasi dan informasi bengkel terdekat. Tampilan Informasi *Service* dan *Spare Part* memberikan detail tentang layanan dan spare part dengan opsi navigasi untuk memilih dan membeli. Tampilan Cari Bengkel dan Navigasi menampilkan peta dengan pin untuk lokasi bengkel dan rute navigasi. Tampilan Hubungi Bengkel dan *Chatting* memungkinkan komunikasi langsung antara pelanggan dan bengkel melalui ruang obrolan. Tampilan Profil dan *Setting* Profil memungkinkan pengguna untuk memperbarui informasi akun mereka, dengan opsi tambahan bagi bengkel untuk mengubah lokasi mereka. Adapun tampilan aplikasi *mobile* dapat dilihat pada gambar 6 sampai dengan gambar 7.



Gambar 6. Tampilan Menu *Chat*



Gambar 7. Tampilan Menu Cari Bengkel

Tampilan menu *chat* dilakukan oleh pelanggan dan pihak bengkel. Pelanggan dapat memilih bengkel dari daftar yang tersedia, sementara pihak bengkel dapat menghubungi pelanggan dari daftar pelanggan yang telah menghubungi mereka. Setelah pemilihan, pengguna diarahkan ke ruang obrolan (*chat room*) untuk berkomunikasi langsung. Fitur ini memudahkan pertanyaan, jawaban, penyampaian informasi, serta penjadwalan layanan, sehingga komunikasi antara pelanggan dan bengkel menjadi lebih cepat dan efektif. Tampilan halaman pencarian bengkel pada aplikasi menampilkan peta dengan pin yang menunjukkan lokasi setiap bengkel, lengkap dengan nama dan ikon navigasi. Halaman navigasi kemudian akan menampilkan rute menuju bengkel yang dipilih oleh pelanggan.

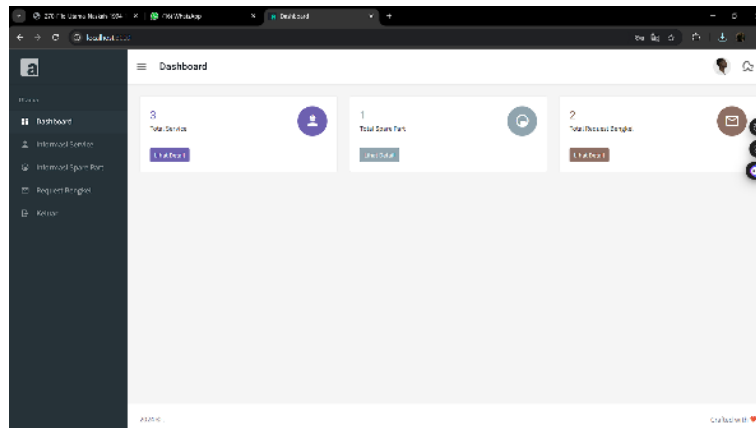
4.6. Website untuk Admin

Website dikembangkan dengan *node.js* untuk mengelola informasi *service*, *spare part*, dan pendaftaran bengkel. Tampilan *Login Website* menyajikan kolom untuk *email* dan *password* admin. Tampilan *Dashboard* menyediakan akses ke informasi layanan, *spare part*, dan permintaan bengkel dengan navigasi mudah. Tampilan Informasi *Service* dan *Spare Part* menampilkan tabel data layanan dan *spare part* yang dapat diedit atau dihapus oleh admin. Tampilan *Request* Bengkel menunjukkan informasi tentang bengkel yang meminta pendaftaran, dengan opsi untuk memberikan izin atau menolak permintaan. Tampilan Tambah Data *Service* dan *Spare Part* memudahkan admin untuk menambahkan informasi baru, sementara Tampilan *Update Data* memungkinkan pembaruan data layanan dan *spare part* yang sudah ada. Adapun tampilan website untuk admin dapat dilihat pada Gambar 8. Tampilan *dashboard website* memiliki navigasi bar di bagian *dashboard*, terdapat menu yang mengarahkan admin ke informasi penting terkait layanan, termasuk rincian layanan servis, daftar spare part yang tersedia, dan opsi untuk melakukan request bengkel untuk layanan lebih lanjut.

4.7. Pengujian

Pengujian aplikasi *mobile* dilakukan menggunakan metode black box untuk memastikan sistem berfungsi sesuai harapan. Hasil pengujian mencakup beberapa skenario seperti *login*, register, akses menu, *chatting*, navigasi, edit profil, dan perhitungan jarak menggunakan algoritma Haversine. Semua pengujian berhasil memenuhi kriteria yang diuji, memastikan aplikasi berfungsi secara optimal, responsif, dan sesuai spesifikasi. Pengujian website, seperti ditampilkan mencakup skenario *login*, tambah, *update*, dan hapus data

service atau *spare part*, serta perizinan bengkel. Pengujian ini memastikan *website* berfungsi dengan baik, memenuhi semua kriteria uji, serta memberikan pengalaman yang optimal bagi admin. Hasilnya menunjukkan aplikasi siap digunakan sesuai spesifikasi yang ditetapkan



Gambar 8. Tampilan Website Dashboard

4.8. Pembahasan

Penelitian ini melakukan pengujian sistem menggunakan metode *black box testing* untuk memastikan aplikasi berfungsi sesuai dengan harapan pengguna. Pengujian dilakukan pada dua *platform*, yaitu aplikasi *mobile* dan *website*. Pada aplikasi *mobile*, beberapa skenario pengujian mencakup login dan registrasi baik untuk pelanggan maupun pihak bengkel, akses ke menu seperti layanan servis, suku cadang, serta pencarian bengkel, fitur *chatting* antara pelanggan dan bengkel, navigasi ke arah bengkel menggunakan peta, pengeditan profil, dan perhitungan jarak menggunakan algoritma *Haversine*. Setiap skenario yang diuji menunjukkan hasil yang berhasil, sesuai dengan harapan, yang menandakan bahwa aplikasi berfungsi dengan baik dari segi fungsionalitas, keandalan, dan kinerja. Pengujian menunjukkan aplikasi dapat memberikan pengalaman pengguna yang optimal dan telah memenuhi semua spesifikasi yang ditetapkan, sehingga siap digunakan.

Pengujian pada *website* mencakup skenario seperti login untuk admin, penambahan dan penghapusan data layanan servis serta suku cadang, pemberian izin registrasi untuk bengkel, serta penghapusan izin registrasi bengkel. Semua pengujian berhasil memenuhi hasil yang diharapkan, yang menunjukkan bahwa *website* berfungsi sesuai spesifikasi dan siap digunakan oleh admin untuk mengelola data bengkel, layanan, dan suku cadang dengan baik.

Selain itu, penelitian ini juga melakukan analisis jarak antara posisi pengguna dan bengkel terdekat menggunakan formula *Haversine*. Perhitungan dilakukan dengan menghitung jarak dari posisi pengguna ke beberapa lokasi bengkel yang telah ditentukan. Contoh penghitungan menunjukkan bahwa jarak antara pengguna dan bengkel B adalah 3,229 kilometer. Dari lima lokasi bengkel yang dianalisis, bengkel B merupakan lokasi terdekat dari pengguna, dengan jarak 3,229 kilometer. Informasi ini penting untuk perencanaan rute atau penentuan prioritas kunjungan berdasarkan jarak tempuh, sehingga dapat meningkatkan efisiensi dalam pelayanan kepada pengguna. Hasil analisis ini menegaskan bahwa metode yang diterapkan telah berjalan dengan baik dan memberikan hasil yang akurat untuk kebutuhan pengguna aplikasi.

5. KESIMPULAN

Pengembangan model dan prototipe sistem pelayanan bengkel menunjukkan bahwa sistem ini memenuhi kebutuhan dasar dengan baik. Hasil pengujian fungsionalitas menunjukkan bahwa fitur utama seperti *login*, navigasi, *chatting*, dan pencarian bengkel terdekat berfungsi sesuai spesifikasi. *Prototype* mempermudah admin dalam mengelola informasi dan perizinan bengkel, serta memungkinkan pelanggan untuk dengan mudah mencari bengkel terdekat, mendapatkan rute, dan berkomunikasi dengan bengkel. Secara keseluruhan, sistem memiliki potensi besar untuk diimplementasikan penuh, meskipun beberapa area masih perlu dioptimalkan berdasarkan umpan balik dan hasil pengujian untuk meningkatkan kinerja dan memenuhi kebutuhan pengguna secara menyeluruh.

Algoritma *Haversine* memiliki beberapa kekurangan, antara lain akurasi terbatas karena mengasumsikan bumi sebagai bola sempurna, yang dapat menyebabkan ketidakakuratan terutama pada jarak panjang. Kinerja algoritma ini juga kurang akurat untuk jarak sangat pendek. Selain itu, *Haversine* tidak memperhitungkan perbedaan ketinggian antara dua titik, melibatkan fungsi trigonometri yang dapat memperlambat perhitungan pada dataset besar, dan hasilnya sangat tergantung pada akurasi data koordinat yang digunakan. Terakhir, untuk area yang lebih kecil, metode jarak yang lebih sederhana mungkin lebih praktis. Mempertimbangkan kekurangan ini penting untuk memilih algoritma yang tepat sesuai kebutuhan.

REFERENSI

- [1] S. D. Negara and A. S. Hidayat, "Indonesia's Automotive Industry," *J. Southeast Asian Econ.*, vol. 38, no. 2, pp. 166–186, Sep. 2021, [Online]. Available: <https://www.jstor.org/stable/27041371>
- [2] S. Ayem, D. K. Wardani, and L. Mas'adah, "Pengaruh Fraud Pentagon terhadap Fraudulent Financial Statement dengan Komite Audit sebagai Variabel Moderasi," *Al-Kharaj J. Ekon. Keuang. & Bisnis Syariah*, vol. 5, no. 2, 2022, doi: 10.47467/alkharaj.v5i2.1355.
- [3] R. Yudistira, "Analisis Pengaruh Kualitas Pelayanan, kepuasan Pelanggan, Komitmen dan Kekuatan Hubungan Terhadap Word of Mouth (Studi Kasus pada Konsumen Bengkel SUZUKI MOBIL PT Persada Lampung Raya di Bandar Lampung)," *J. Kompetitif Bisnis*, vol. 1, no. 3, p. 8, 2020.
- [4] A. Mashdarul Khair, "RANCANG BANGUN APLIKASI Pencarian Bengkel Aktif Berbasis Google Maps API di Wilayah Kota Makassar," *J. Manaj. Inform. Sist. Inf. DAN Teknol. Komput.*, vol. 1, no. 1, 2022.
- [5] Ferawaty and A. Cenderawan, "Implementasi Haversine Formula dan Algoritme Dijkstra pada Aplikasi Vehicle Help Sebagai Layanan Konsultasi Pemilik Kendaraan," *Progresif J. Ilmu Komput.*, vol. 19, no. 1, 2023.
- [6] E. S. Ipo, Y. Shova, and M. A. Irwansyah, "Rancang Bangun Sistem Informasi Geografis Persebaran Bengkel, Tambal Ban dan Minyak Eceran di Kota Pontianak Berbasis Web," *J. Sist. dan Teknol. Inf.*, vol. 11, no. 1, 2023, doi: 10.26418/justin.v11i1.47917.
- [7] P. Dauni, M. D. Firdaus, R. Asfariani, M. I. N. Saputra, A. A. Hidayat, and W. B. Zulfikar, "Implementation of Haversine formula for school location tracking," *J. Phys. Conf. Ser.*, vol. 1402, no. 7, 2019, doi: 10.1088/1742-6596/1402/7/077028.
- [8] A. Andreou, C. X. Mavromoustakis, J. M. Batalla, E. K. Markakis, G. Mastorakis, and S. Mumtaz, "UAV Trajectory Optimisation in Smart Cities Using Modified A* Algorithm Combined With Haversine and Vincenty Formulas," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 9757–9769, 2023, doi: 10.1109/TVT.2023.3254604.
- [9] A. S. Ali, A. H. Hasan, and H. A. Lafta, "Antitheft vehicle tracking and control system based iot," *J. Crit. Rev.*, vol. 7, no. 9, pp. 88 – 92, 2020, doi: 10.31838/jcr.07.09.16.
- [10] F. A. Ubaidillah and A. F. Waluyo, "Aplikasi Layanan Perbaikan Motor Online Berbasis Android dengan Memanfaatkan Google Maps," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 3, pp. 1216–1226, 2023.
- [11] E. S. O. Helmi, O. Emam, and M. Abdel-Salam, "Deep Learning Framework for Locating Physical Internet Hubs using Latitude and Longitude Classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 7, pp. 235–240, 2022, doi: 10.14569/IJACSA.2022.0130731.
- [12] R. Purbaningtyas and A. Arizal, "Nearest Excellent Potential Location Using Distance Algorithm," *J. Phys. Conf. Ser.*, vol. 1413, no. 1, 2019, doi: 10.1088/1742-6596/1413/1/012032.
- [13] N. Chan Soe and T. Lai Lai Thein, "Haversine Formula and RPA Algorithm for Navigation System," *Int. J. Data Sci. Anal.*, vol. 6, no. 1, p. 32, 2020, doi: 10.11648/j.ijdsa.20200601.14.
- [14] M. Z. Rohman, Irwansyah, and W. E. Sari, "The Medical Facilities Selection Based on Location-Based Services Application Using SAW and TOPSIS Algorithm," *J. Phys. Conf. Ser.*, vol. 1577, no. 1, 2020, doi: 10.1088/1742-6596/1577/1/012012.
- [15] E. Maria, E. Budiman, Haviluddin, and M. Taruk, "Measure distance locating nearest public facilities using Haversine and Euclidean Methods," *J. Phys. Conf. Ser.*, vol. 1450, no. 1, 2020, doi: 10.1088/1742-6596/1450/1/012080.
- [16] A. B. Galindo-Muro, R. Cespi, and S. I. Vallarta-Serrano, "Applications of Electric Vehicles in Instant Deliveries," *Energies*, vol. 16, no. 4, 2023, doi: 10.3390/en16041967.
- [17] T. Jabid *et al.*, "An intelligent road accident reduction system using device-to-device communication," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 30, no. 3, pp. 1729–1739, 2023, doi: 10.11591/ijeecs.v30.i3.pp1729-1739.
- [18] W. Aliman, "Perancangan Perangkat Lunak untuk Menggambar Diagram Berbasis Android," *Syntax Lit. J. Ilm. Indones.*, vol. 6, no. 6, 2021, doi: 10.36418/syntax-literat.v6i6.1404.
- [19] P. B. A. A. Putra, V. H. Pranatawijaya, and N. N. K. Sari, "Implementasi Location Based Service Pada Aplikasi Mobile Penyajian Ruang Ujian," *J. Sains dan Inform.*, vol. 6, no. 1, 2020.
- [20] M. A. Saeed and N. M. Edan, "Design and Implementation of Peer-to-Peer Video and Chat Communication," *Tech. Soc. Sci. J.*, vol. 37, 2022, doi: 10.47577/tssj.v37i1.7570.