Jurnal Informatika Terpadu

https://journal.nurulfikri.ac.id/index.php/JIT
ISSN ONLINE : 2460-8998

IMPLEMENTASI SISTEM PEMESANAN HOTEL MENGGUNAKAN ALGORITMA *HAVERSINE* UNTUK OPTIMALISASI REKOMENDASI LOKASI

Hamka Lukmanul Hakim Adhani[1], Mufti Ari Bianto[2], Alif Nanda Pratama[3], Septina Alfiani Hidayah[4]

[1,2,3,4] Teknik Komputer, Universitas Muhammadiyah Lamongan
Lamongan, Jawa Timur, Indonesia 62218
hamkalukmanul@gmail.com, muftiari10@gmail.com, akhiyananda@gmail.com, septinafifi03@gmail.com

Abstract

A location-based lodging recommendation system helps users find nearby hotels efficiently through a web-based platform. The system utilizes the Haversine algorithm to calculate the distance between the user's location and the hotel by automatically retrieving coordinates via the Geolocation API. Calculated distances are compared with hotel data stored in a MySQL database, and the results are displayed on a web interface integrated with the Google Maps API. Testing was conducted on six hotels with distances ranging from 6.73 km to 23.97 km, and results were compared with Google Maps estimates. The system achieved an average distance difference of 0.0183 km, with an accuracy rate of 99.83%. These findings indicate that the Haversine algorithm provides highly accurate distance estimations and is reliable for location-based hotel recommendation systems.

Keywords: *Google Maps API, Haversine Algorithm, Hotel Reservation, Recommender System, User Location*

Abstrak

Sistem rekomendasi penginapan berbasis lokasi pengguna merupakan solusi yang membantu menemukan hotel terdekat secara efisien melalui platform website. Sistem ini memanfaatkan algoritma Haversine untuk menghitung jarak antara lokasi pengguna dan hotel dengan data koordinat yang diperoleh secara otomatis melalui Geolocation API. Hasil perhitungan jarak dibandingkan dengan data hotel yang tersimpan dalam basis data MySQL, kemudian ditampilkan melalui antarmuka web yang terintegrasi dengan Google Maps API. Pengujian dilakukan terhadap enam data penginapan dengan jarak terpendek 6,73 km dan jarak terpanjang 23,97 km, serta dibandingkan dengan estimasi jarak dari Google Maps. Hasil pengujian menunjukkan rata-rata selisih jarak sebesar 0,0183 km dengan tingkat akurasi mencapai 99,83%. Temuan ini menunjukkan bahwa algoritma Haversine mampu memberikan estimasi jarak yang sangat mendekati layanan peta profesional dan dapat diandalkan dalam sistem rekomendasi hotel berbasis lokasi.

Kata kunci: Algoritma *Haversine*, Google Maps API, Lokasi Pengguna, Pemesanan Hotel, Sistem Rekomendasi

1. PENDAHULUAN

Perkembangan teknologi digital telah membawa perubahan signifikan dalam berbagai bidang kehidupan, termasuk industri pariwisata dan layanan akomodasi. Salah satu solusi modern yang banyak dikembangkan adalah sistem rekomendasi berbasis lokasi, yaitu sistem yang dirancang untuk memberikan informasi atau saran kepada pengguna berdasarkan posisi geografis mereka secara *real-time* [1].

Agar sistem semacam ini bekerja secara akurat, diperlukan metode perhitungan jarak yang mampu mempertimbangkan bentuk permukaan bumi. Salah satu algoritma yang umum digunakan dalam konteks ini adalah *Haversine*, karena dapat menghitung jarak antar titik koordinat dengan mempertimbangkan kelengkungan bumi, sehingga memberikan estimasi jarak yang mendekati kenyataan [2].

Untuk meningkatkan ketepatan hasil rekomendasi, algoritma *Haversine* kerap dikombinasikan dengan teknik pembelajaran mesin seperti *K-Nearest Neighbor (KNN)*, sehingga sistem dapat memberikan hasil berdasarkan kedekatan fisik sekaligus kemiripan karakteristik pengguna [3].

Penerapan algoritma ini tidak terbatas pada rekomendasi hotel, tetapi juga terbukti efektif dalam membantu pencarian lokasi akomodasi penginapan berbasis data spasial, khususnya di kawasan perkotaan [4].

Diterima 10 Juli 2025, Direvisi 28 Agustus 2025, Diterima untuk publikasi 10 Oktober 2025

Metode ini juga digunakan dalam sistem distribusi pelanggan berbasis peta digital, di mana penghitungan jarak menjadi komponen penting dalam optimasi rute distribusi [5].

Bahkan, ketika dikombinasikan dengan metode pemilihan rute seperti *Dijkstra* dan *SAW*, algoritma *Haversine* mampu menghasilkan sistem pencarian jalur terpendek yang efisien untuk keperluan transportasi dalam kota [6].

Di kawasan terpencil seperti Kecamatan Nanggung, algoritma *Haversine* telah diterapkan untuk membantu pencarian objek wisata, yang menunjukkan keandalannya dalam skenario geografis yang kompleks [7].

Lebih lanjut, metode ini juga dimanfaatkan dalam sistem pencarian lokasi donasi berbasis Android, di mana kecepatan dan akurasi dalam menghitung jarak sangat penting untuk menyesuaikan posisi pengguna dengan titik donasi terdekat [8].

Sistem zonasi sekolah di kota besar seperti Depok juga menggunakan *Haversine* dalam menentukan jarak antara rumah siswa dan sekolah terdekat, yang menunjukkan fleksibilitas penerapannya dalam konteks pendidikan [9].

Dalam pengembangan sistem kehadiran pegawai berbasis *geofencing* dan pengenalan wajah, algoritma *Haversine* berperan sebagai komponen utama dalam menentukan lokasi kehadiran pengguna secara tepat [10].

Selain itu, metode perhitungan jarak ini juga digunakan sebagai bagian dari pendekatan proaktif dalam sistem rekomendasi yang mengacu pada aturan pengguna, guna meningkatkan personalisasi hasil yang diberikan [11].

Dalam sektor pariwisata, *Haversine* terbukti mendukung sistem rekomendasi lokasi tujuan wisata, khususnya dengan menggabungkan teknik *content-based filtering* untuk mengarahkan pengguna ke destinasi yang sesuai dengan preferensi mereka [12].

Pada sistem monitoring kehadiran siswa secara *real-time*, penggabungan layanan *Location-Based Service (LBS)* dan algoritma *Haversine* memungkinkan pelacakan lokasi yang presisi tanpa keterlambatan signifikan [13].

Implementasi algoritma ini dalam aplikasi angkutan umum seperti sistem informasi angkutan kota juga menunjukkan bahwa metode ini dapat digunakan untuk mengarahkan penumpang ke jalur terdekat secara efisien [14].

Dalam sistem keamanan publik berbasis Android, *Haversine* digunakan untuk mendeteksi dan menampilkan estimasi posisi pengguna saat laporan darurat dikirimkan ke pusat komando [15].

Untuk meningkatkan kualitas rekomendasi hotel, pendekatan berbasis pengelompokan data seperti *K-Means clustering* sering digunakan bersama *Haversine*, karena mampu mengelompokkan akomodasi berdasarkan jarak, harga, dan fasilitas yang tersedia [16].

Jika digabungkan dengan data histori pengguna dan metode pembelajaran mesin, sistem rekomendasi menjadi lebih adaptif terhadap pola pencarian yang berbeda-beda, memberikan pengalaman pengguna yang lebih pribadi [17].

Dalam pengembangan sistem, posisi pengguna secara otomatis diperoleh melalui *HTML5 Geolocation API*, kemudian dihitung jaraknya terhadap data penginapan yang tersimpan dalam basis data *MySQL* menggunakan rumus *Haversine* [18].

Selain itu, sistem presensi menggunakan *Global Positioning Systems (GPS)* dan *Location-Based Service (LBS)* juga menggunakan rumus *Haversine*. Tujuannya adalah untuk meningkatkan keakuratan deteksi posisi. Menurut [19], tingkat akurasi lebih dari 90% dapat dicapai.

Penelitian terdahulu banyak berfokus pada penerapan algoritma *Haversine* dalam sistem pencarian rute atau distribusi pelanggan, sedangkan penelitian ini berfokus pada pengoptimalan rekomendasi lokasi penginapan dengan integrasi Google Maps API secara *real-time*.

Hasil rekomendasi akhirnya ditampilkan dalam bentuk antarmuka web yang interaktif yang terintegrasi dengan *Google Maps API*, sehingga pengguna dapat melihat letak hotel dalam bentuk peta digital serta membandingkan opsi akomodasi berdasarkan kedekatan jarak.

2. METODE PENELITIAN

2.1 Desain Sistem

Tahapan awal dalam proses pengembangan sistem dimulai dengan melakukan analisis kebutuhan pengguna untuk menetapkan fitur utama yang harus tersedia. Beberapa fitur yang dirancang meliputi kemampuan pencarian penginapan berdasarkan lokasi pengguna, penghitungan jarak antar lokasi, serta penyajian hasil rekomendasi yang diurutkan berdasarkan jarak terdekat [1].

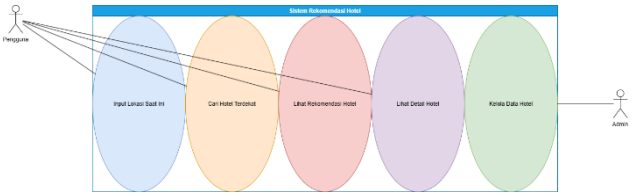
Sistem ini dirancang secara modular dengan mengedepankan antarmuka pengguna yang responsif dan mudah digunakan. Antarmuka dikembangkan menggunakan teknologi web seperti *HTML, CSS, dan Bootstrap*, agar tampil optimal pada berbagai perangkat, baik desktop maupun *mobile*. Sementara itu, pengelolaan data dilakukan menggunakan basis data relasional *MySQL*, yang telah terbukti efektif untuk sistem skala menengah hingga besar [2].

Untuk proses perhitungan jarak antara pengguna dan lokasi penginapan, digunakan algoritma *Haversine*. Algoritma ini menghitung jarak berdasarkan koordinat lintang dan bujur dengan mempertimbangkan kelengkungan bumi, sehingga hasil estimasinya lebih akurat dibandingkan metode linier biasa [3].

Berdasarkan penelitian terdahulu, *Haversine* telah banyak diterapkan dalam berbagai sistem informasi geografis dan aplikasi berbasis lokasi karena kesederhanaannya serta akurasi hasil perhitungannya. Hal ini menjadikannya metode yang ideal untuk diterapkan dalam sistem rekomendasi penginapan berbasis lokasi seperti yang dikembangkan dalam penelitian ini [4].

2.1.1 *Use Case Diagram*

Pada gambaran *Use Case Diagram*, ditunjukkan interaksi antara aktor dan fungsionalitas dari sistem secara lebih rinci. Gambar 1 memperlihatkan dua aktor utama, yaitu Pengguna dan Admin. Pengguna memiliki kemampuan untuk memasukkan lokasi, melihat rekomendasi hotel terdekat, serta mengakses informasi detail tentang hotel. Sedangkan, Admin memperoleh hak akses yang bersifat administratif untuk mengelola data penginapan yang tersimpan dalam sistem.



Gambar 1. *Use Case Diagram*

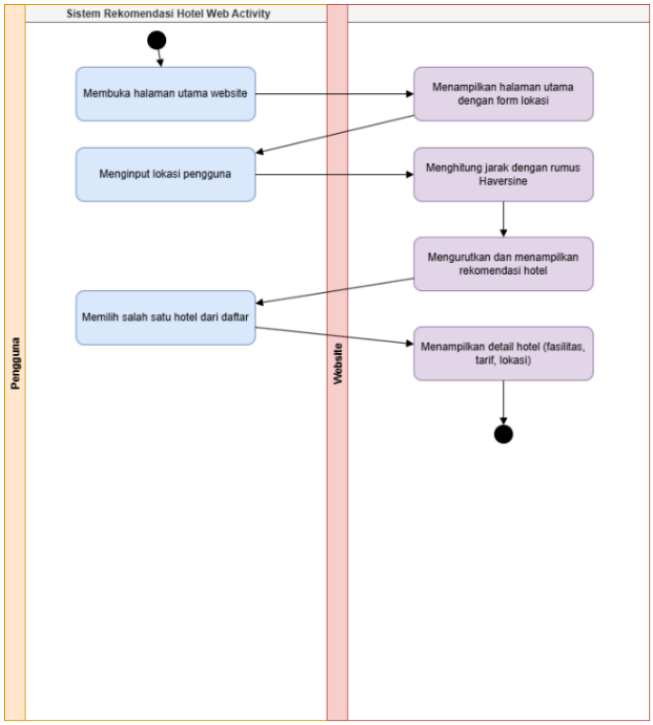
2.1.2 *Flowchart* Sistem

Gambar 2 menampilkan sebuah *flowchart* yang menggambarkan proses utama dalam sistem rekomendasi hotel berbasis lokasi. Dalam alur ini, proses dimulai dari penerimaan *input* lokasi pengguna, diikuti dengan pengambilan data hotel dari basis data, kemudian dilakukan perhitungan jarak menggunakan rumus *Haversine*. Setelah itu, hasil rekomendasi diurutkan dan disajikan berdasarkan jarak terdekat. Diagram ini memudahkan pemahaman mengenai logika dan alur kerja sistem secara keseluruhan.

Gambar 2. *Flowchart* Sistem

2.1.3 *Activity Diagram*

Activity diagram ini memberikan gambaran yang lebih mendalam mengenai alur aktivitas antara pengguna dan sistem. Pada Gambar 3, terlihat bahwa pengguna pertama-tama membuka halaman utama, kemudian memasukkan lokasi yang diinginkan. Selanjutnya, sistem akan menghitung jarak berdasarkan data tersebut, mengurutkan daftar penginapan sesuai jarak, dan menampilkan rekomendasi yang relevan. Setelah itu, pengguna dapat memilih salah satu penginapan untuk melihat detail lebih lengkap. Diagram ini membantu memperjelas interaksi secara langsung dan *real-time* antara pengguna dan sistem.



Gambar 3. *Activity Diagram*

2.1.4 Deskripsi Sistem

Sistem pemesanan hotel yang dikembangkan dalam penelitian ini merupakan sebuah aplikasi berbasis web yang dirancang untuk memberikan rekomendasi penginapan secara otomatis, berdasarkan lokasi geografis pengguna. Sistem ini dibangun menggunakan bahasa pemrograman PHP untuk bagian *backend*, dan didukung oleh basis data *MySQL* sebagai media penyimpanan utama data tersebut.

Dalam rangka mendukung fitur rekomendasi lokasi, sistem secara terintegrasi menggunakan *Google Maps API*. Integrasi ini berfungsi untuk menampilkan lokasi penginapan secara visual pada peta digital dan juga memfasilitasi proses pengambilan koordinat secara otomatis saat admin melakukan *input* data.

Pengguna dapat menggunakan sistem ini lewat *browser* di berbagai perangkat, desktop maupun *mobile*. Dari awal memang dirancang supaya tampak bagus dan nyaman dipakai di semua layar, pakai *HTML, CSS, dan Bootstrap*.

Dengan demikian, pengguna tidak perlu mengunduh aplikasi tambahan, cukup membuka situs web melalui browser dan izinkan lokasi kamu supaya bisa langsung menggunakan fitur rekomendasi.

Dengan pendekatan ini, sistem dirancang untuk memberikan kemudahan penggunaan sekaligus menawarkan tingkat fleksibilitas yang tinggi dalam membantu pengguna menemukan penginapan terdekat sesuai lokasi saat ini.

2.2 Hasil Pengolahan Data

2.2.1 Pengumpulan dan Penyimpanan Data

Hotel booking system ini menggunakan *MySQL* sebagai tempat utama untuk menyimpan seluruh data yang dibutuhkan untuk rekomendasi, reservasi, dan pengelolaan penginapan. Berikut penjelasan sederhana mengenai struktur tabel-tabel utama yang digunakan:

1. tb_penginapan
   Menyimpan data terkait setiap penginapan yang terdaftar dalam sistem. Atribut yang disimpan meliputi:
   - id_penginapan (*Primary Key*)
   - nama_penginapan
   - lokasi (alamat lengkap)
   - deskripsi
   - harga_penginapan
   - *rating*
   - fasilitas
   - gambar
   - *latitude* dan *longitude* (koordinat geografis)

2. tb_user
   Pengelolaan informasi akun pengguna, untuk pengguna umum maupun admin, meliputi atribut-atribut yang dikelola seperti:
   - id_*user* (*Primary Key*)
   - nama_*user*
   - *email*
   - *password*
   - hak akses (*user*/admin)

3. tb_kamar
   Data ini digunakan untuk menyimpan informasi akun penginapan yang berkaitan dengan tipe kamar dari setiap properti, tercatat dengan untuk keperluan pengguna umum maupun administrator. Atribut-atribut yang dikelola meliputi beragam informasi terkait kategori kamar, kapasitas, serta fasilitas yang tersedia.
   - id_kamar (*Primary Key*)
   - id_penginapan (*Foreign Key*)
   - tipe_kamar
   - kapasitas
   - stok
   - harga_kamar

4. tb_pemesanan
   Tabel ini menyediakan catatan mengenai aktivitas pemesanan yang dilakukan oleh pengguna, seperti yang tercantum di bawah ini:
   - id_pemesanan (*Primary Key*)
   - id_*user*
   - id_kamar
   - tanggal_*checkin*
   - tanggal_*checkout*
   - status_pemesanan

5. tb_pembayaran
   Digunakan untuk menyimpan informasi transaksi pembayaran, antara lain:
   - id_pembayaran (*Primary Key*)
   - id_pemesanan (*Foreign Key*)
   - metode_pembayaran
   - jumlah
   - status_pembayaran
   - bukti_transaksi

2.2.2 Pengambilan Lokasi Pengguna

Sistem secara otomatis mengenali lokasi geografis pengguna saat halaman *website* pertama kali diakses. Proses ini dilakukan dengan memanfaatkan teknologi *HTML5 Geolocation API*, yang telah tersedia di hampir semua browser modern. Melalui *API* ini, sistem mendapatkan data lintang (*latitude*) dan bujur (*longitude*) pengguna, yang kemudian digunakan sebagai dasar utama dalam menghitung jarak dengan algoritma *Haversine*.

Namun, karena tingkat akurasi lokasi pengguna dapat beragam tergantung pada perangkat serta izin lokasi yang telah diberikan, sistem menawarkan fitur “Lokasi Saya”. Fitur ini berupa tombol yang dapat ditekan oleh pengguna untuk meminta pembaruan lokasi secara manual, apabila informasi lokasi otomatis sebelumnya dirasa kurang akurat. Dengan menekan tombol ini, sistem akan kembali memanggil fungsi *Geolocation API* untuk memperoleh koordinat terbaru yang lebih tepat dan terpercaya.

Proses ini berjalan dengan cepat dan tanpa memerlukan muat ulang halaman (*refresh*), karena pemanggilan lokasi dilakukan secara asinkron menggunakan *JavaScript*. Hasil koordinat yang diperoleh langsung dikirim ke server(*backend)* untuk diolah dan digunakan dalam perhitungan jarak relatif terhadap data penginapan yang tersimpan di basis data, melalui penerapan sistem otomatis yang didukung oleh fitur *fallback* manual, tingkat akurasi lokasi pengguna dapat terjaga secara optimal, sehingga hasil rekomendasi yang disajikan oleh sistem menjadi lebih sesuai dan relevan.

2.2.3 Pencarian dan Perhitungan Jarak

Setelah lokasi pengguna berhasil diidentifikasi, sistem akan menyediakan dua pilihan untuk menampilkan data

penginapan. Pengguna dapat memasukkan kata kunci pencarian, seperti nama hotel atau lokasi tertentu, atau memilih untuk membiarkan sistem menampilkan seluruh daftar penginapan yang tersedia dalam basis data secara otomatis.

Ketika pengguna memasukkan kata kunci, sistem akan melakukan proses penyaringan terhadap data penginapan yang tersimpan dalam tabel tb_penginapan. Penyaringan ini dilakukan dengan mencocokkan kata kunci tersebut terhadap kolom nama_penginapan dan lokasi. Jika tidak ada *input* yang dimasukkan, maka seluruh data penginapan yang aktif akan ditampilkan berdasarkan perhitungan jarak sebagai acuan utama.

Setiap data akomodasi yang berhasil melewati proses pencarian tersebut akan dihitung jaraknya terhadap lokasi pengguna menggunakan algoritma *Haversine*. Metode ini menghitung jarak antara dua titik koordinat lintang dan bujur di permukaan bumi dengan memperhitungkan kelengkungan planet, sehingga memberikan estimasi jarak yang lebih nyata dan akurat dibandingkan dengan metode *linier* atau *Euklides.*

Rumus *Haversine* yang dipakai dalam sistem ini adalah sebagai berikut:

```
//  query SQL dengan rumus Haversine
if (!is_null($lat_user) && !is_null($long_user)) {
    $query = "SELECT *, (
            6371 * 2 * ASIN(SQRT(
                POWER(SIN(RADIANS(lat_penginapan - $lat_user) / 2), 2) +
                COS(RADIANS($lat_user)) * COS(RADIANS(lat_penginapan)) *
                POWER(SIN(RADIANS(long_penginapan - $long_user) / 2), 2)
            ))
        ) AS jarak
        FROM tb_penginapan
        WHERE lokasi LIKE '%$kota%'
        AND id_kategori = $id_kategori
        ORDER BY jarak ASC
    ";
} else {
    $query = "
        SELECT * FROM tb_penginapan
        WHERE lokasi LIKE '%$kota%'
        AND id_kategori = $id_kategori
    ";
}
```

Gambar 4. Rumus *Haversine* yang dipakai dalam sistem

Keterangan:

- Koordinat lat dan long diberikan dalam satuan radian.
- Variabel R menunjukkan jari-jari bumi, yaitu sekitar 6371 km.
- Nilai d adalah jarak antara dua titik, yang dihitung dalam kilometer.

Pada gambar 4 menunjukkan bagaimana rumus *Haversine* diterapkan dalam *query SQL* yang digunakan oleh sistem. Rumus ini digunakan untuk menghitung jarak antara lokasi pengguna dan penginapan dengan mengacu pada koordinat lintang (*latitude*) dan bujur (*longitude*). Nilai jari-jari Bumi sebesar 6371 km menjadi dasar perhitungan, sehingga hasilnya mendekati jarak nyata di permukaan Bumi.

Penerapan rumus *Haversine* secara langsung dalam *query* basis data membawa keuntungan besar, karena proses perhitungan jarak dapat dilakukan saat data diambil. Hal ini mengurangi beban proses di sisi aplikasi dan meningkatkan kecepatan sistem. Berdasarkan hasil perhitungan ini, penginapan dapat diurutkan menurut jarak terdekat, yang menjadi dasar utama dalam penentuan rekomendasi lokasi. Setiap jarak yang diperoleh kemudian disimpan dalam sebuah *array* sementara yang berisi pasangan antara id_penginapan dan jarak_km. Data tersebut digunakan sebagai basis untuk menyusun urutan rekomendasi yang relevan dan akurat.

2.2.4 Pengurutan dan Pemilihan Rekomendasi

Setelah sistem menghitung jarak antara lokasi pengguna dan setiap penginapan menggunakan algoritma *Haversine*, hasilnya disimpan dalam sebuah array atau struktur data sementara. Setiap elemen dalam array ini berisi informasi penting seperti id_penginapan, nama_penginapan, dan nilai jarak_km.

Langkah berikutnya adalah melakukan proses pengurutan terhadap array tersebut berdasarkan nilai jarak terkecil hingga yang terbesar. Sistem menggunakan metode pengurutan standar dari bahasa pemrograman *PHP*, yaitu fungsi *usort*(), dengan kriteria pengurutan berdasarkan elemen jarak_km. Tujuan dari proses ini adalah untuk memastikan bahwa penginapan yang paling dekat akan muncul terlebih dahulu dalam daftar rekomendasi.

Setelah proses pengurutan selesai, sistem secara otomatis akan memilih sejumlah penginapan terbaik, umumnya sekitar lima hingga sepuluh data terdekat dari lokasi pengguna. Jumlah ini dapat disesuaikan sesuai dengan preferensi pengguna atau kebijakan sistem. Pengambilan penginapan teratas dilakukan menggunakan fungsi *array_slice*() dari *array* yang telah diurutkan, sehingga menghasilkan daftar pilihan yang relevan dan efisien.

Penginapan-penginapan yang telah dipilih akan menjadi data utama dalam rekomendasi. Selain itu, data tersebut akan dilengkapi dengan informasi tambahan dari basis data, seperti harga, rating, fasilitas, dan gambar, sehingga dapat ditampilkan secara lengkap dan informatif di antarmuka pengguna.

Dengan pendekatan ini, sistem tidak hanya menampilkan daftar penginapan secara acak, melainkan berdasarkan **urutan berdasarkan jarak terdekat**, yang sangat relevan dan memberikan manfaat besar dalam konteks pencarian lokasi terdekat.

2.2.5 Penyajian dan Visualisasi Hasil Rekomendasi

Setelah sistem mengurutkan daftar penginapan berdasarkan jarak terdekat, hasil rekomendasi tersebut ditampilkan kepada pengguna melalui sebuah tampilan antarmuka web yang bersifat responsif dan interaktif. Setiap entri penginapan yang muncul mencantumkan berbagai

informasi penting yang diambil dari basis data, termasuk di antaranya:

- Nama Penginapan
- Jarak dari lokasi pengguna (dalam kilometer)
- Harga per malam (dalam Rupiah)
- *Rating* atau ulasan pengguna (skala 1–5)
- Fasilitas utama dan gambar penginapan

Tampilan data disusun dalam bentuk kartu (*card*) atau daftar, dengan tata letak yang responsif dan nyaman digunakan di perangkat *mobile* maupun *desktop*. Pengguna dapat dengan mudah membandingkan berbagai pilihan penginapan berdasarkan informasi yang disajikan secara lengkap dan intuitif.

Selain menampilkan daftar secara konvensional, sistem juga menawarkan visualisasi geografis yang interaktif dalam bentuk peta. Dengan mengintegrasikan *Google Maps API*, sistem secara *real-time* memuat lokasi pengguna serta penanda tempat penginapan di atas peta digital. Penanda tersebut dilengkapi dengan berbagai label dan ikon yang membedakan, sehingga memudahkan pengguna dalam mengidentifikasi setiap lokasi dengan cepat dan akurat.

2.2.6 Analisis Akurasi Sistem

Dalam rangka memastikan ketepatan sistem dalam memberikan rekomendasi lokasi penginapan berdasarkan jarak, dilakukan pengujian terhadap hasil perhitungan algoritma *Haversine* yang diimplementasikan dalam sistem ini. Pengujian tersebut dilakukan dengan membandingkannya terhadap estimasi jarak yang dihasilkan oleh *Google Maps* sebagai acuan pembanding resmi.

Pengujian dilakukan dengan memilih secara acak sepuluh sampel penginapan dari basis data. Setiap sampel kemudian dianalisis untuk menghitung jaraknya dari lokasi pengguna saat ini. Hasil pengukuran dari sistem dibandingkan dengan jarak yang ditunjukkan oleh *Google Maps*, kemudian dihitung selisihnya dalam satuan kilometer. Berikut ini adalah contoh hasil pengujiannya:

Tabel 1. Hasil perbandingan perhitungan jarak antara sistem dengan Google Maps

No	Nama Penginapan	Jarak Sistem (km)	Jarak *Google Maps* (km)	Selisih (km)
1	Hotel Aqilla Syariah	6.73 km	6.75 km	0.02 km
2	Hotel Grand Mahkota Lamongan	8.65 km	8.67 km	0.02 km
3	Hotel Elresas	8.66 km	8.68 km	0.02 km
4	Grand Wisma Hotel Syariah	9.13 km	9.15 km	0.02 km
5	Hotel Bougenville Syari'ah	8.41 km	8.42 km	0.1 km
6	Tanjung Kodok Beach	23.97 km	23.99 km	0.02 km

Berdasarkan tabel 1 hasil perbandingan, dapat disimpulkan bahwa selisih jarak antara sistem yang dikembangkan dan *Google Maps* berkisar antara **0,01 km hingga 0,02 km**. Rata-rata perbedaan yang tercatat adalah sekitar **±0,0183 km**. Nilai ini mengindikasikan bahwa sistem tersebut memiliki kemampuan yang sangat dalam memperkirakan jarak antar titik lokasi, dengan tingkat akurasi yang tinggi dan mendekati hasil aktual yang disediakan oleh layanan pemetaan profesional seperti *Google Maps*.

Untuk menilai tingkat ketepatan dari pengukuran sistem, digunakan rumus berikut:

$$\text{Akurasi} = \left(1 - \frac{\text{Rata-rata Selisih}}{\text{Rata-rata Jarak } \textit{Google Maps}}\right) \times 100\%$$

Rata-rata jarak dari keenam penginapan yang diuji, menurut hasil pengukuran melalui *Google Maps*, adalah sebesar **10,94 km**. Berdasarkan data ini, sistem dapat diukur akurasinya sebagai berikut:

$$\text{Akurasi} = \left(1 - \frac{0.0183}{10.94}\right) \times 100\% \approx 99.83\%$$

Dengan tingkat akurasi yang melebihi 99%, dapat disimpulkan bahwa algoritma *Haversine* yang diadopsi dalam sistem ini mampu memberikan estimasi jarak yang sangat akurat, dengan tingkat kesalahan yang minim. Hasil ini menegaskan bahwa algoritma tersebut sangat relevan dan layak diterapkan dalam sistem rekomendasi berbasis lokasi, yang menuntut respons cepat, efisiensi tinggi, serta ketelitian yang dapat diandalkan.

2.3 Hasil Pengelolaan Data

Tahap akhir dari proses dalam sistem rekomendasi ini adalah menyusun daftar penginapan berdasarkan jarak terdekat dari lokasi pengguna. Sebelum jarak dapat dihitung, sistem terlebih dahulu memperoleh posisi pengguna dalam bentuk koordinat geografis berupa lintang dan bujur menggunakan *HTML5 Geolocation API*. Lintang menunjukkan posisi utara–selatan relatif terhadap garis *khatulistiwa*, sementara bujur menunjukkan posisi timur–barat relatif terhadap garis meridian utama *Greenwich*.

Berikut adalah contoh hasil lokasi pengguna yang berhasil didapatkan:

- *Latitude* pengguna: -7,10599
- *Longitude* pengguna: 112,3874

Semua data penginapan yang tersimpan di dalam basis data telah dilengkapi dengan atribut *latitude* dan *longitude*. Atribut ini sebelumnya dimasukkan secara manual oleh

administrator melalui antarmuka peta yang disediakan oleh *Google Maps API*. Koordinat tersebut memungkinkan sistem untuk menghitung jarak antara dua lokasi dengan tingkat akurasi tinggi, menggun akan rumus *Haversine* yang memperhitungkan kelengkungan Bumi secara keseluruhan.

Setelah seluruh jarak dihitung, sistem akan menyusun hasil berdasarkan urutan dari penginapan yang paling dekat hingga yang paling jauh. Berikut adalah contoh hasil pengolahan data yang ditampilkan oleh sistem saat pengguna berada di wilayah Lamongan, Jawa Timur.

Tabel 2. Data koordinat penginapan beserta jarak dari lokasi pengguna

No	Nama Penginapan	*Latitude*	*Longitude*	Jarak (km)
1	Hotel Bougenville Syari'ah	-7.109181	112.402806	1.5 km
2	Grand Wisma Hotel Syariah	-7.118165	112.402568	1.93 km
3	Hotel Aqilla Syariah	-7.117586	112.450910	3.27 km
4	Hotel Elresas	-7.122014	112.414622	3.34 km
5	Hotel Grand Mahkota	-7.122302	112.415191	6.89 km
6	Tanjung Kodok Beach	-6.866575	112.357730	25.5 km

Dari tabel 2 tersebut, dapat dipahami bahwa sistem tidak hanya menampilkan informasi mengenai hotel berupa nama, harga, dan *rating*, tetapi juga menyertakan koordinat lokasi *real* dari masing-masing penginapan, yang menjadi dasar utama dalam proses penghitungan jarak.

Data tersebut diperoleh melalui serangkaian proses berikut: Pertama, sistem memperoleh lokasi pengguna berupa koordinat *latitude* dan *longitude*. Selanjutnya, sistem menarik data penginapan dari basis data yang tersedia. Kemudian, dilakukan perhitungan jarak antara posisi pengguna dan setiap penginapan menggunakan metode *Haversine*. Hasil perhitungan tersebut kemudian diurutkan berdasarkan jarak terdekat dan ditampilkan kepada pengguna.

Proses pengolahan ini dilakukan secara langsung dan terintegrasi dengan antarmuka visual yang berbasis *Google Maps API*, sehingga pengguna dapat melihat lokasi penginapan secara langsung pada peta interaktif.

Hasil ini menunjukkan bahwa sistem mampu mengelola data spasial secara efisien dan memberikan rekomendasi berdasarkan data geografis yang akurat, mendekati tingkat ketelitian hasil yang biasanya dihasilkan oleh platform profesional seperti *Google Maps*.

3. HASIL DAN PEMBAHASAN

Penelitian ini berhasil merancang serta mengimplementasikan sebuah sistem berbasis web yang mampu memberikan rekomendasi penginapan terdekat dengan memanfaatkan perhitungan jarak geografis melalui algoritma *Haversine*. Sistem ini dirancang untuk membantu pengguna dalam secara otomatis dan *real-time* menemukan penginapan yang paling dekat dari posisi mereka.

3.1 Hasil Implementasi Sistem

Sistem ini dirancang berbasis web dengan mengintegrasikan fitur *geolokasi HTML5* yang mampu mendeteksi posisi pengguna secara otomatis. Setelah posisi pengguna diperoleh, sistem akan menghitung jarak antara pengguna dan penginapan yang tersimpan dalam database menggunakan rumus *Haversine*. Perhitungan ini memperhitungkan kelengkungan bumi, sehingga dapat memberikan estimasi jarak yang akurat [1].

Hasil perhitungan tersebut divisualisasikan melalui sebuah peta *interaktif* yang dibangun menggunakan *Leaflet.js*, dengan sumber data berasal dari *OpenStreetMap*. Pada peta, lokasi pengguna ditandai secara langsung, sementara sistem menampilkan lokasi penginapan di sekitarnya dalam bentuk *marker*. Selanjutnya, daftar rekomendasi disusun berdasarkan jarak terdekat secara berurutan.

3.2 Hasil Pengujian Sistem

Uji coba ini dilakukan di wilayah Lamongan, Jawa Timur, dengan menempatkan titik lokasi pengguna secara strategis. Sistem secara otomatis mampu menampilkan rekomendasi penginapan yang paling relevan berdasarkan jarak terdekat dari lokasi pengguna. Daftar penginapan disajikan dalam bentuk visual yang menampilkan informasi lengkap, termasuk nama, gambar, harga, dan penilaian *rating*, sehingga memudahkan pengguna dalam memilih akomodasi yang sesuai. Berikut adalah data hasil rekomendasi yang diberikan oleh sistem:

Tabel 3. Hasil rekomendasi penginapan berdasarkan jarak terdekat

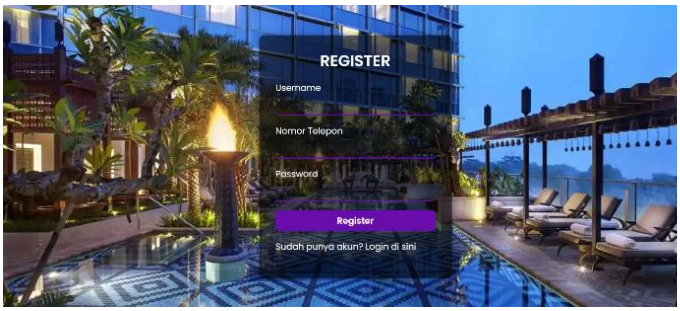
No	Nama Penginapan	Jarak (km)	Harga (Rp)	Rating
1	Hotel Bougenville Syariah	1.53 km	300.000	2,0
2	Grand Wisma Hotel Syariah	1.96 km	400.000	4,0
3	Hotel Elresas	3.3 km	900.000	4,0
4	Hotel Grand Mahkota Lamongan	3.37 km	400.000	3,0
5	Hotel Aqilla Syariah	6.92 km	200.000	3,0
6	Tanjung Kodok Beach	26.91 km	900.000	4,0

Tabel 3 yang disajikan memperlihatkan kemampuan sistem dalam menyusun daftar penginapan berdasarkan jarak secara akurat. *Validitas* penghitungan jarak diuji dengan membandingkan hasil sistem ini dengan estimasi yang diperoleh dari *Google Maps*. Hasil pengujian menunjukkan bahwa selisih rata-rata antara keduanya adalah kurang dari 0,1 km. Hal ini mengindikasikan bahwa penerapan rumus

Haversine berjalan dengan dan dapat dipercaya dalam konteks pengolahan data spasial.

3.3 Implementasi Antarmuka

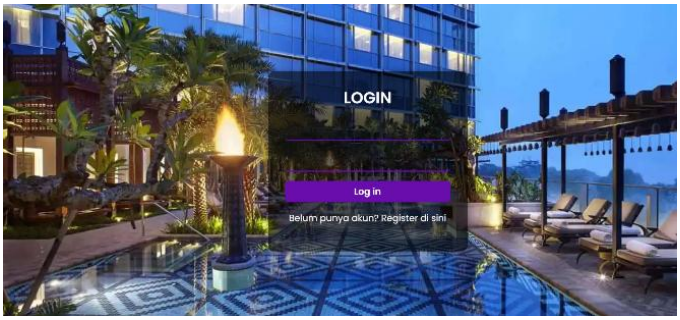
3.3.1 Registrasi



Gambar 5. Antarmuka Registrasi Akun Pengguna

Formulir pendaftaran pengguna yang terlihat pada Gambar 5 merupakan langkah awal agar pengguna dapat dengan mudah membuat akun dalam sistem ini. Pada halaman tersebut, pengguna diharuskan mengisi ***username*, nomor telepon, dan kata sandi**. Data yang dimasukkan lalu disimpan dalam basis data dan akan digunakan saat proses login berikutnya.

3.3.2 *Login*



Gambar 6. Halaman *login* yang digunakan untuk Autentikasi Pengguna

Seperti yang terlihat pada Gambar 6, sistem menyajikan halaman *login* sebagai mekanisme untuk memastikan identifikasi pengguna. Pengguna diharapkan memasukkan *username* dan kata sandi yang sesuai dengan data yang tercatat saat pendaftaran. Apabila data yang dimasukkan benar, sistem akan mengarahkan pengguna ke halaman utama; sebaliknya, jika terjadi kesalahan, maka muncul pesan *error* sebagai notifikasi.

3.3.3 Halaman Utama

Gambar 7. Tampilan Halaman Utama yang menjadi Beranda Sistem

Setelah pengguna berhasil masuk ke dalam sistem, mereka akan diarahkan ke halaman utama yang ditampilkan pada Gambar 7. Halaman ini berfungsi sebagai pusat *navigasi* yang menghubungkan pengguna dengan berbagai fitur penting seperti pencarian penginapan, pengelolaan pemesanan, dan akses ke halaman profil pengguna. Tata letak menu dirancang secara sederhana dan terstruktur rapi agar pengguna dapat memahami alur penggunaan sistem dengan mudah, bahkan bagi mereka yang baru pertama kali mengakses aplikasi ini.

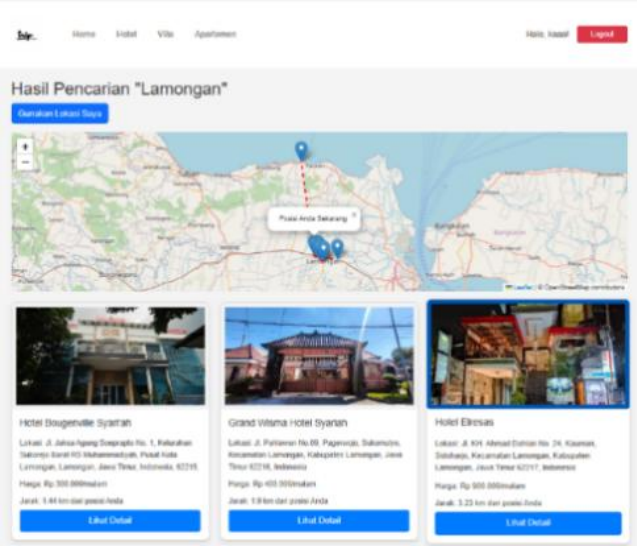
3.3.4 Pencarian

Gambar 8. Formulir Pencarian Penginapan berdasarkan Kota dan Tanggal

Gambar 8 menampilkan formulir pencarian penginapan, yang dirancang untuk memudahkan pengguna dalam menentukan kriteria pencarian mereka. Formulir ini memungkinkan pengguna memilih dari berbagai kategori seperti hotel, vila, atau apartemen, serta memasukkan lokasi yang diinginkan dan tanggal menginap. Kehadiran fitur ini memastikan proses pencarian menjadi lebih terfokus dan sesuai dengan kebutuhan pengguna.

Selain itu, formulir ini dilengkapi dengan mekanisme validasi data, misalnya memastikan bahwa tanggal *check-in* tidak lebih akhir dari tanggal *check-out*. Validasi semacam ini membantu memastikan bahwa data yang diterima oleh sistem adalah sah dan benar, sehingga mengurangi kemungkinan kesalahan dalam proses pencarian maupun pemesanan.

3.3.5 Hasil Pencarian



Gambar 9. Hasil Pencarian yang menampilkan Daftar Hotel beserta Lokasinya pada Peta

Gambar 9 ini menampilkan daftar hotel yang sesuai dengan kota yang telah dipilih oleh pengguna. Pengguna dapat melihat lokasi hotel pada peta, mengaktifkan fitur 'Gunakan Lokasi Pengguna' untuk secara otomatis mendeteksi posisi mereka, serta mengakses informasi lengkap tentang masing-masing hotel, seperti nama, alamat, tarif per malam, dan jarak dari lokasi pengguna saat ini. Terdapat pula tombol khusus yang memungkinkan pengguna mendapatkan detail lengkap dari setiap hotel yang ditampilkan. Fitur 'Gunakan Lokasi Pengguna' memanfaatkan teknologi lokasi perangkat untuk mengetahui posisi pengguna secara otomatis, sehingga sistem dapat menghitung jarak hotel dari lokasi pengguna dengan akurat dan mudah digunakan.

4. KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sebuah sistem rekomendasi pemesanan hotel berbasis web yang mengintegrasikan algoritma *Haversine* untuk menghitung jarak geografis antara lokasi pengguna dan penginapan. Sistem ini telah diuji secara langsung melalui akses daring, dan mampu memberikan rekomendasi dengan tingkat akurasi yang tinggi. Hasil perhitungan jarak dalam sistem menunjukkan selisih yang sangat kecil jika dibandingkan dengan estimasi yang dihasilkan oleh peta digital seperti *Google Maps*.

Sistem ini tampilannya lengkap dengan peta interaktif dan fitur pencarian berdasarkan lokasi, yang membuat pengguna jadi lebih mudah dan nyaman saat menggunakannya. Hasil uji coba juga menunjukkan bahwa sistem ini bisa cepat dan efisien dalam mengidentifikasi serta menampilkan daftar penginapan terdekat. Jadi, sistem ini sangat sesuai dijadikan solusi praktis buat pengguna yang ingin mencari penginapan secara cepat dan tepat sesuai lokasi mereka.

Ke depannya, sistem ini bisa dikembangkan lagi dengan menambahkan fitur ulasan dari pengguna, sistem pemesanan langsung, dan juga personalisasi rekomendasi berdasarkan riwayat pencarian kamu. Dengan demikian, layanan dapat lebih optimal dan sesuai dengan kebutuhan pengguna.

DAFTAR PUSTAKA

[1] G. H. Prakarsha, H. D. K. Lumbantobing, M. R. Ramadhan, and I. Prihandi, "*Haversine* algorithm design using the Google Maps API method for Android-based public security applications," Int. J. Comput. Trends Technol., vol. 69, no. 2, pp. 53–60, 2021. [Online]. Available: https://doi.org/10.14445/22312803/IJCTT-V69I2P108

[2] A. Muliawan, T. Badriyah, and I. Syarif, "Membangun sistem rekomendasi hotel dengan content-based filtering menggunakan K-nearest neighbor dan *Haversine* formula," Technomedia J., vol. 7, no. 2, pp. 231–247, 2022. [Online]. Available: https://doi.org/10.33050/tmj.v7i2.1893

[3] A. Hakim and M. Saefudin, "Aplikasi sistem informasi geografis menggunakan metode *Haversine* formula pencarian rumah kost daerah Jakarta Selatan," J. Inf. Syst. Informatics Comput., vol. 5, no. 2, pp. 397–408, Nov. 2021. [Online]. Available: https://doi.org/10.52362/jisicom.v5i2.640

[4] B. Setiawan and S. Samsudin, "Geographic information system for customer distribution using the *Haversine* algorithm," Sinkron: J. Penelit. Tek. Inform., vol. 8, no. 4, pp. 2737–2747, Oct. 2023. [Online]. Available: https://doi.org/10.33395/sinkron.v8i4.13091

[5] Z. A. Mulkan, I. R. Setiawan, and F. Frazna, "Penerapan algoritma Dijkstra dengan metode SAW dan *Haversine* pada pencarian rute terdekat di Sukabumi," J. Inf. Syst. Res. (JOSH), vol. 4, no. 4, pp. 1205–1218, Jul. 2023. [Online]. Available: https://doi.org/10.47065/josh.v4i4.3661

[6] F. Fatimah, S. H. Al Ikhsan, and B. Wulandari, "Implementation of the *Haversine* method for the application of finding tourist attractions in the Nanggung District," J. Pilar Nusa Mandiri, vol. 18, no. 1, pp. 59–64, 2022. [Online]. Available: https://doi.org/10.33480/pilar.v18i1.3000

[7] D. Daniel and D. Lasut, "Application of the *Haversine* method in the Android-based donation search application," Bit-Tech, vol. 6, no. 1, pp. 1–7, 2023. [Online]. Available: https://doi.org/10.32877/bt.v6i1.736

[8] D. Ikasari, W. Ikasari, and R. Andika, "Implementation of *Haversine* formula to determine the shortest path using web-based application for high school zoning in Depok," Am. J. Softw. Eng. Appl., vol. 10, no. 2, pp. 19–31, Sept. 2021. [Online]. Available: https://doi.org/10.11648/j.ajsea.20211002.11

[9] F. Lubis, I. Prihandi, W. Usino, and N. A. B. Ismail, "Development geofencing process and face recognition design using *Haversine* formula and the K-nearest neighbor algorithm in the employee attendance application," AIP Publishing, vol. 2987, no. 1, p. 020007, 2024. [Online]. Available: https://doi.org/10.1063/5.0200763

[10] Ilarri and R. Trillo-Lado, "An approach for proactive *mobile* recommendations based on user-defined rules," Expert Syst. Appl., vol. 242, p. 122714, 2024. [Online]. Available: https://doi.org/10.1016/j.eswa.2023.122714

[11] I. Yulfihani and M. Zakariyah, "Optimization of tourism destination recommendations in Batang Regency using content-based filtering," J. Appl. Inform. Comput., vol. 8, no. 2, 2022. [Online]. Available: https://doi.org/10.30871/jaic.v8i2.8618

[12] C. E. S. Natasiya, B. Pramono, A. M. Sajiah, S. Sutardi, and L. M. B. Aksara, "Implementasi teknologi location based service (LBS) dan metode *Haversine* formula pada sistem monitoring kehadiran siswa secara real time," SemanTIK: Tek. Inform., vol. 10, no. 1, 2024. [Online]. Available: http://dx.doi.org/10.55679/semantik.v10i1.27232

[13] R. N. Imamsyah, N. N. Kamala Sari, and A. Lestari, "Rancang bangun aplikasi AngkotKita menggunakan location based service dengan metode *Haversine* berbasis Android," J. Inf. Technol. Comput. Sci., vol. 3, no. 1, Mar. 2023. [Online]. Available: https://doi.org/10.47111/jointecoms.v3i1.10796

[14] R. A. F. Mustaqim, A. Nugroho, and F. A. Soni, "Aplikasi safety driving assistance dengan perhitungan *Haversine*," Smart Techno, vol. 6, no. 1, pp. 1–9, Feb. 2024. [Online]. Available: https://doi.org/10.59356/smart-techno.v6i1.109

[15] S. Prasetyo and U. Zaky, "Implementasi algoritma *Haversine* pada Mapbox API untuk pencarian bengkel terdekat berbasis *mobile*," MALCOM: Indones. J. Mach. Learn. Comput. Sci., vol. 4, no. 4, 2022. [Online]. Available: https://doi.org/10.57152/malcom.v4i4.1677

[16] W. A. Fitriani, A. Ikhwan, and M. Alda, "Implementasi algoritma *Haversine* formula untuk pencarian lokasi rumah makan halal terdekat di Kota Parapat berbasis *mobile*," J. Ilm. Sains dan Teknol., vol. 9, no. 1, pp. 65–77, 2025. [Online]. Available: https://doi.org/10.47080/saintek.v9i1.3645

[17] S. Nugroho, R. Mahendra, dan M. E. Santoso, "Measuring Distance Locating Nearest Public Facilities Using *Haversine* and Euclidean Methods," J. Phys. Conf. Ser., vol. 1450, no. 1, p. 012080, 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1450/1/012080

[18] N. M. A. E. D. Wirastuti, L. Verlin, I.-H. Mkwawa, dan K. G. Samarah,"Implementation of Geographic Information System Based on Google Maps API to Map Waste Collection Point Using the *Haversine* Formula Method,"J. Ilm. Tek. Elektro Komput. dan Inform., vol. 9, no. 3, pp. 731–745, Sept. 2023.[Online].Available:https://doi.org/10.26555/jiteki.v9i3.26588

[19] P. B. Utomo, D. Wahyudi, dan M. Mujiono, “Pengembangan Sistem Informasi Presensi Berbasis Global Positioning Systems dan Location-Based Service,” *Jurnal Informatika Terpadu*, vol. 11, no. 1, pp. 20–28, 2025. https://doi.org/10.54914/jit.v11i1.1563