

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Menurut (Pitaloka & Koesdijarto, 2022) *Traveling Sales Problem* merupakan salah satu tantangan dalam bidang optimasi kombinatorial. Dalam masalah ini, kita diberikan sejumlah lokasi atau titik tujuan, masing-masing dengan biaya perjalanan tertentu untuk berpindah dari satu lokasi ke lokasi lainnya. Tujuan utamanya adalah menemukan rute yang paling efisien, baik dalam hal jarak terpendek maupun biaya terendah. Seorang *sales* ditugaskan untuk memulai perjalanannya dari satu kota asal, mengunjungi setiap kota yang telah ditentukan tepat satu kali, dan kemudian kembali ke kota asal tersebut. Tantangan ini sering digunakan untuk menguji kemampuan algoritma dalam mencari solusi optimal dalam situasi yang melibatkan banyak variabel dan batasan.

Penelitian terdahulu membahas beberapa studi serupa yang telah dilakukan sebelumnya, dengan tujuan untuk membedakan atau menyempurnakan penelitian tersebut sehingga dapat ditemukan aspek kebaruan. Oleh karena itu, dalam tinjauan pustaka ini, peneliti menampilkan temuan penelitian sebelumnya dalam tabel berikut.

Tabel 2.1 Penelitian Terdahulu

No	Judul	Permasalahan	Solusi	Kelebihan dan Kekurangan
1.	Implementation of Haversine Algorithm on Mapbox API for Searching Nearest Workshop Based on <i>Mobile Devices</i> (Prasetyo & Zaky, 2024)	Sulitnya mendapatkan informasi lokasi bengkel terdekat secara efisien	Pengembangan aplikasi Android untuk menampilkan lokasi bengkel terdekat	Kelebihannya adalah Algoritma Haversine mencapai tingkat akurasi 99%. Kekurangannya adalah Kurang cocok untuk lokasi dengan jarak Panjang.
2.	Metode Hybrid-DPSO dan Clustering untuk Rekomendasi Rute Perjalanan Wisata Berbasis <i>Mobile</i> Android (Nugroho et al., 2024)	Sulit menentukan rute perjalanan yang optimal untuk tempat pariwisata	Mengembangkan sistem dengan metode hybrid-dPSO yang memberikan rekomendasi rute yang lebih optimal untuk perencanaan perjalanan	Kelebihannya adalah Metode hybrid-DPSO menghasilkan rute yang lebih pendek sebesar 16,75% dibandingkan dengan metode DPSO. Kekurangannya adalah bergantung dari cluster.

No	Judul	Permasalahan	Solusi	Kelebihan dan Kekurangan
3.	Mobile Application Development to Solve Vehicle Routing Problems in Marketing or Tour Trip Planning (Nugroho et al., 2023)	Masalah pengoptimalan kombinatorial pada Vehicle Routing Problem (VRP)	Pengembangan aplikasi <i>mobile</i> untuk solusi Vehicle Routing Problem (VRP)	Kelebihannya ada Hasil yang valid dari 63 responden menunjukkan efektivitas solusi. Kekurangannya Masih bisa dikembangkan lebih baik lagi.
4.	Implementasi Spasial Algoritma Harvesine pada Mapbox API untuk Pemetaan Pariwisata (Adil et al., 2023)	Kesulitan menemukan lokasi wisata dan rute bagi pengunjung	Pengembangan aplikasi Android untuk menunjukkan lokasi wisata dan rute bagi pengunjung	Kelebihannya adalah dapat menunjukkan data spasial lokasi pariwisata di Lombok. Kekurangannya adalah Algoritma Harvesine masih belum meengabaikan bentuk asli permukaan rute
5.	Realtime Tracking Dan	Sulit menemukan	Mengembangkan	Kelebihannya adalah

No	Judul	Permasalahan	Solusi	Kelebihan dan Kekurangan
1	Pencarian Rute Optimal Menggunakan Mapbox Optimized API Untuk Mengatasi Travelling Sales Problem (Prasetya et al., 2021)	rute terbaik untuk mengunjungi beberapa lokasi konsumen	aplikasi <i>mobile</i> untuk mencari rute optimal ke lokasi penjual COD	Algoritma yang digunakan efektif dalam menentukan rute terpendek untuk pengiriman. Kekurangannya adalah waktu pemrosesan yang relatif lama.

Penelitian ini memiliki beberapa perbedaan dari penelitian sebelumnya, yang disebut sebagai "*research gap*", sebagai berikut:

1. Objek penelitian ini adalah beberapa titik toko yang harus dikunjungi oleh *sales*.
2. Penelitian ini menggunakan posisi *sales* saat melakukan login pada aplikasi sebagai titik awal, memberikan fleksibilitas dalam penentuan lokasi.
3. Penelitian ini akan menggunakan Mapbox Optimization API v1 sebagai tools pemetaan rekomendasi rute toko yang dikunjungi. Sehingga dapat menghasilkan urutan rute optimal bagi *sales*.

2.2 Teori Terkait

Berikut beberapa teori terkait pada penelitian ini:

2.2.1 Sales

Sales adalah representatif dari perusahaan yang melaksanakan satu atau lebih kegiatan seperti mencari konsumen baru, berkomunikasi, melakukan penjualan, memberikan jasa pelayanan, mengumpulkan informasi, serta mengalokasikan produk. *Sales* juga berfungsi sebagai penghubung yang unik antara perusahaan dan konsumen (Hurrijal & Gupitha, 2021).

2.2.2 Tracking

Menurut (Zufria et al., 2020) *tracking* adalah kegiatan yang melibatkan pemantauan, pelacakan, dan pemetaan jalur selama perjalanan menuju lokasi yang dituju. Proses ini digunakan dalam berbagai bidang, termasuk pemetaan rute jalan dan sungai, serta pemantauan posisi kendaraan yang digunakan untuk pelayanan publik, seperti bus, ambulans, dan kendaraan lainnya. Dengan menggunakan teknologi *tracking*, pengelolaan perjalanan menjadi lebih efisien dan akurat, sehingga dapat mendukung berbagai kebutuhan operasional dan pelayanan masyarakat.

2.2.3 Android

Menurut (Hartiyani et al., 2023) Android adalah sistem operasi untuk ponsel yang berbasis Linux yang awalnya dikembangkan oleh

Android, Inc. sebelum Google membelinya sebagai bagian dari strateginya pada tahun 2005. Google berupaya agar Android tetap terbuka dan dapat diakses secara bebas. Oleh karena itu, sebagian besar kode sumber Android tersedia secara gratis melalui lisensi open-source Apache. Akibatnya, siapa saja dapat menggunakan, mengunduh, dan mengubah kode sumber Android secara gratis.

2.2.4 Kotlin

Kotlin adalah bahasa pemrograman yang berjalan di atas *Java Virtual Machine* (JVM). Bahasa ini dirancang secara pragmatis untuk pengembangan aplikasi Android, menggabungkan konsep *object-oriented* (OO) dan pemrograman fungsional. Kotlin memiliki interoperabilitas tinggi, yang memungkinkannya digunakan bersama dengan bahasa pemrograman Java dalam satu proyek. Kotlin juga dapat digunakan untuk pengembangan desktop, web, dan bagian backend. Dikembangkan oleh JetBrains, perusahaan dibalik IntelliJ IDEA, Kotlin telah dirilis sebagai proyek *open-source* dan terus berkembang pesat. Google juga secara resmi mendukung Kotlin sebagai bahasa pemrograman utama untuk pengembangan aplikasi Android (Feibriandirza, 2020).

2.2.5 Rest API

Salah satu istilah untuk antarmuka pemrograman aplikasi adalah API (*Application Programming Interface*), yang memungkinkan pengembang untuk mengakses aplikasi atau layanan yang sudah ada dalam program. Dengan API, pengembang dapat memanfaatkan fungsi yang sudah ada dalam aplikasi lain, sehingga mereka tidak perlu membangun aplikasi baru. API web memanggil fungsi dengan *Hyper Text Transfer Protocol* (HTTP) dan menerima tanggapan dalam format *Extensible Markup Language* (XML) atau *JavaScript Object Notation* (JSON).

Salah satu tujuan utama API adalah untuk memungkinkan pertukaran data antara aplikasi yang berbeda. API juga mempercepat proses pengembangan aplikasi dengan menyediakan beberapa fungsi yang sudah ada, sehingga pengembang tidak perlu membuat fitur yang sama dari awal. Selain itu, API yang beroperasi pada tingkat sistem operasi

memungkinkan aplikasi berkomunikasi dengan lapisan dasar dan satu sama lain menggunakan berbagai protokol dan spesifikasi yang telah disesuaikan. (Hari Asgar et al., 2022)

REST API (Representational State Transfer) adalah jenis API yang dirancang untuk membuat layanan web yang dapat diakses melalui protokol HTTP. REST API berfokus pada sumber daya (resources) yang ditunjukkan oleh URL dan berinteraksi dengan sumber daya tersebut melalui metode HTTP seperti GET, POST, PUT, dan DELETE.

2.2.6 Retrofit 2

Retrofit 2 adalah pustaka klien HTTP yang aman untuk digunakan dalam aplikasi Android dan Java. Pustaka ini berfungsi sebagai klien REST, yang memungkinkan pengembang untuk terhubung dengan layanan web REST tanpa harus membuat metode koneksi sendiri. Sementara itu, Gson adalah pustaka yang digunakan untuk mengkonversi objek Java ke format JSON dan sebaliknya, dari JSON ke objek Java. Retrofit 2 menyederhanakan proses pertukaran data antara format JSON atau XML, yang kemudian diubah menjadi Plain Old Java Objects (POJOs) (Lumba, 2021).

Retrofit 2 dibangun dengan menggunakan berbagai pustaka dan alat bantu lainnya. Pustaka ini memanfaatkan protokol OkHttp yang dikembangkan oleh pengembang dari Square, Inc. untuk mengelola proses transmisi data (request) melalui jaringan. OkHttp berfungsi sebagai dasar bagi Retrofit 2, memastikan efisiensi dan keandalan dalam pengiriman permintaan dan pengambilan data dari server.

2.2.7 JSON (JavaScript Object Notation)

Menurut (Hari Asgar et al., 2022) JSON (JavaScript Object Notation) adalah format yang digunakan untuk berbagi data. Namanya berasal dari bahasa pemrograman JavaScript, tetapi API dapat digunakan dalam berbagai bahasa pemrograman lainnya, seperti Python, Ruby, PHP, dan Java. Format ini sering dilafalkan seperti "Jason" dan menggunakan ekstensi .json ketika disimpan sebagai file terpisah. Dalam konteks file lain, seperti .html, JSON dapat dituliskan dalam tanda petik sebagai JSON string atau disimpan dalam variabel. Salah satu keunggulan JSON adalah kemudahan dalam

mentransfer data antara server web dan klien atau browser, sehingga menjadikannya pilihan yang populer untuk pertukaran data dalam aplikasi web modern.

2.2.8 JWT (JSON Web Token)

JWT, juga dikenal sebagai web JSON token, adalah token dalam bentuk string panjang yang acak yang digunakan untuk sistem autentikasi dan pertukaran informasi. API menggunakan konsep JWT untuk autentikasi, berbeda dengan aplikasi web biasa yang menggunakan sesi untuk mengingat siapa yang login.

JWT terdiri dari tiga bagian yang dipisahkan oleh tanda titik (.):

- 1. Header**

Menyimpan informasi tentang jenis encoding yang digunakan dan algoritma yang diterapkan untuk menghasilkan signature.

- 2. Payload**

Berisi nilai-nilai informasi yang ditransaksikan, seperti data pengguna dan klaim lainnya.

- 3. Signature**

Memuat nilai hash yang digunakan untuk memverifikasi keaslian payload, memastikan bahwa data tidak diubah setelah token dibuat.

Dengan struktur ini, JWT memungkinkan pengembang untuk mengimplementasikan autentikasi yang aman dan efisien dalam aplikasi berbasis web dan API. Jika Anda tertarik untuk mempelajari lebih lanjut tentang JWT, banyak sumber daya online yang menawarkan tutorial dan dokumentasi mendalam (Hari Asgar et al., 2022).

2.2.9 Mapbox

Mapbox adalah platform yang menawarkan alat yang dapat diintegrasikan dengan mudah ke dalam aplikasi *mobile*. Banyak pengembang aplikasi menambahkan peta yang terhubung ke internet untuk memastikan lokasi pengguna lebih mudah diakses (Hidayatulloh & Airlangga, 2022).

Salah satu keunggulan Mapbox adalah kemampuan untuk membuat peta menggunakan perangkat lunak open-source yang kemudian

diberikan lisensi kepada pengembang untuk digunakan dalam aplikasi mereka. Mapbox mengintegrasikan berbagai sumber data dari berbagai lembaga pemerintah dan open-source, termasuk OpenStreetMap, NASA, dan U.S. Geological Survey (Prasetya et al., 2021).

Kemudahan integrasi dengan Native Kotlin juga menjadi faktor penting, karena Mapbox menyediakan SDK yang dirancang khusus untuk pengembangan aplikasi Android. Selain itu, Mapbox menawarkan kuota gratis yang cukup besar, termasuk hingga 100.000 permintaan per bulan untuk API Optimasi, yang sangat mendukung proyek pengembangan dengan anggaran terbatas.

Tabel 2.2 Keuntungan Menggunakan Mapbox SDK

Keuntungan Utama	Detail	Kuota Bulanan Gratis
Integrasi Mudah & Dokumentasi Lengkap	API dan SDK Mapbox mudah diintegrasikan, didukung dokumentasi dan contoh kode yang komprehensif.	-
Pengalaman Pengguna Lebih Baik	Fitur seperti Address Autocomplete, Search Box, dan Geocoding mempercepat input, meningkatkan akurasi.	Address Autocomplete: Kuota gratis sesuai Search API session
Efisiensi Operasional & Penghematan Biaya	Infrastruktur Mapbox mengurangi biaya pemeliharaan server dan menawarkan harga yang kompetitif.	-
Kustomisasi Tinggi	Mapbox memungkinkan kustomisasi tampilan peta, ikon, dan fitur sesuai kebutuhan brand atau	-

Keuntungan Utama	Detail	Kuota Bulanan Gratis
	aplikasi.	
Data & Cakupan Global	Mendukung pencarian dan geocoding di ratusan juta alamat dan POI di seluruh dunia.	Geocoding: 100.000 permintaan API/bulan
Fitur Lanjutan untuk Bisnis	Mendukung analisis lokasi, optimasi rute, matrix API, isochrone, dan movement data untuk insight.	-
Skalabilitas & Performa Tinggi	Infrastruktur Mapbox mendukung permintaan volume tinggi dengan latensi rendah dan keandalan tinggi.	-
Dukungan Online & Offline	SDK Mapbox Search mendukung pencarian lokasi baik saat online maupun offline.	-
Pengayaan Data Lokasi	API seperti Details API dan Boundaries API memperkaya data dengan detail lokasi, batas wilayah, dll.	-
Maps SDKs for Mobile (iOS/Android/Flutter)	Penggunaan SDK Maps di aplikasi mobile dihitung berdasarkan Monthly Active Users (MAU).	Kuota gratis MAU, detail di halaman pricing

2.2.10 Android Studio

Menurut (Sondang Sibuea et al., 2022) Pada acara Google I/O 2013, Android Studio, sebuah Integrated Development Environment (IDE) untuk platform Android, diluncurkan oleh Google. IDE ini dikembangkan sebagai penerus Eclipse IDE dan didasarkan pada IntelliJ IDEA, salah satu IDE populer untuk Java. Android Studio sekarang menjadi IDE resmi untuk pengembangan aplikasi Android. Sebagai penerus Eclipse, Android Studio menawarkan berbagai fitur baru yang tidak tersedia di Eclipse. Salah satu perbedaannya adalah penggunaan Gradle sebagai build environment, menggantikan Ant yang digunakan di Eclipse. Beberapa fitur lainnya antara lain:

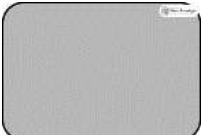
1. Menggunakan sistem build berbasis Gradle.
2. Mendukung build untuk beberapa APK.
3. Menyediakan *sample* untuk Google Services dan beragam jenis perangkat.
4. Tingkat kemampuan *editor layout*.
5. Dukungan bawaan untuk Google Cloud Platform, yang membuat integrasi dengan Google Cloud Messaging dan App Engine lebih mudah.
6. Memungkinkan import library langsung dari repositori Maven.

2.2.11 Activity Diagram

Salah satu jenis diagram dalam Unified Modeling Language (UML) adalah *activity diagram*, yang digunakan untuk memodelkan aktivitas atau alur kerja dalam suatu sistem atau proses. Diagram ini menampilkan rangkaian kegiatan, tindakan, serta keputusan yang berlangsung seiring waktu (Rasiban et al., 2024).

Dengan kata lain, Activity Diagram adalah representasi visual dari alur kerja dalam suatu sistem atau proses yang menunjukkan urutan aktivitas, tindakan, dan keputusan yang terjadi, serta bagaimana elemen-elemen tersebut berinteraksi satu sama lain seiring berjalaninya waktu. Diagram ini membantu memetakan proses bisnis atau aliran logis dalam suatu sistem.

Tabel 2.3 Keterangan Activity Diagram

Gambar	Nama	Keterangan
	Status Mulai	Menandakan posisi awal
	Aktivitas	Biasanya akan diawali dengan kata kerja
	Percabangan/ <i>Decision</i>	Memiliki pilihan lebih dari satu aktivitas
	Penggabungan / <i>Join</i>	Penggabungan beberapa aktivitas
	Status Akhir	Penanda sebagai posisi selesai

2.2.12 Postman

Postman merupakan sebuah platform kolaboratif yang dirancang khusus untuk pengembangan API. Diciptakan oleh Abhinav Asthana, seorang programmer dan desainer berbakat dari Bangalore, India, Postman menawarkan solusi yang efisien untuk pengujian, pengembangan, dan dokumentasi API (Sari & Hidayat, 2022). Dengan antarmuka yang intuitif, Postman memungkinkan pengembang untuk melakukan pengujian

API secara cepat dan efektif. Prinsip kerja Postman berfokus pada pengategorian permintaan (request) berdasarkan metode, URL, dan berbagai parameter yang terkait. Kesederhanaan fitur-fitur Postman menjadikannya alat yang sangat berguna dalam memfasilitasi proses pengembangan API yang lebih lancar dan terstruktur.

2.2.13 MVVM

Model View ViewModel (MVVM) menurut (Fajri & Rani, 2022) adalah pola desain arsitektur yang terdiri dari tiga komponen utama, yaitu Model, View, dan ViewModel. Masing-masing komponen memiliki peran yang berbeda:

1. Model berfungsi sebagai representasi data yang digunakan dalam logika bisnis. Biasanya Model dibuat dalam bentuk objek sederhana seperti kelas POJO atau data class di Kotlin.
2. View merupakan bagian antarmuka pengguna, terdiri dari layout dan komponen seperti Activity atau Fragment. Tampilan pada layout akan dikontrol secara dinamis melalui Activity atau Fragment.
3. ViewModel bertindak sebagai penghubung antara View dan Model. Komponen ini menyediakan beberapa variabel yang dapat diamati oleh View, dan memiliki kesadaran terhadap siklus hidup ViewModel tetap aktif selama View belum dihancurkan, sehingga data tetap tersimpan di dalam ViewModel Scope. Ketika View dihancurkan, data di ViewModel juga akan dihapus secara otomatis.