

BAB 2

LANDASAN TEORI

2.1 Landasan Teori

Landasan teori merupakan bagian dari penelitian yang memuat teori-teori dan hasil-hasil penelitian yang diperoleh dari mempelajari literatur, yang bertujuan sebagai kerangka teori untuk melakukan penelitian serta memberikan gambaran dari teori yang terkait dalam penelitian yang dilakukan, dalam hal ini yaitu untuk pembangunan aplikasi *android*. Landasan teori sebagai acuan dalam penyusunan pembangunan aplikasi *Tracking* kemacetan untuk membantu pengendara dalam mencari jalan yang efektif, serta menghindari tingginya volume kendaraan pada suatu jalan, hal ini meliputi ATCS, kemacetan, Aplikasi, Android, Android Studio, Google Maps API, Deteksi Objek, TensorFlow, *Fused Location Provider*, GPS, Latitude dan longitude, CCTV M3U8, API, JSON, Kotlin, XML, NodeJS, Database MySQL dan Unified Modeling Language (UML).

2.2 Area Traffic Control System

Area Traffic Control System atau lebih dikenal dengan sebutan ATCS merupakan pengendalian suatu sistem pada lalu lintas berbasis teknologi informasi di suatu kawasan tertentu yang memiliki tujuan untuk mengoptimalkan kinerja jaringan pada jalan dengan melalui optimasi dan koordinasi lampu lalu lintas dan CCTV saat ini sebanyak 74 titik penyebaran pada setiap persimpangan. Adapun beberapa sistem utama yang dimiliki ATCS diantaranya:

1. **Server & Workstation** yang bertindak sebagai pusat operasi yang memantau dan mengendalikan kondisi lalu lintas semua persimpangan di area yang sama.
2. **Wall Map** yang memberikan informasi tentang status dan status pengontrol lokal.
3. **Local Controller** yang berfungsi sebagai pengontrol persimpangan.
4. **Video Surveillance** yang berfungsi untuk memantau melalui jaringan kamera CCTV.

5. **Vehicle Detector** yang berfungsi untuk mengidentifikasi kendaraan yang dipantau oleh kamera CCTV.

2.3 Kemacetan

Kemacetan lalu lintas adalah suatu situasi atau kondisi dimana lalu lintas tidak berjalan dengan lancar atau bahkan terhenti. Penyebab kemacetan lalu lintas dikarenakan bertambahnya jumlah kendaraan yang melebihi kapasitas panjang dan lebar jalan, manajemen lalu lintas yang buruk, kondisi jalan yang buruk, disiplin pengguna jalan yang buruk, dan lain-lain. Kemacetan sering terjadi di kota-kota besar, terutama kota yang tidak memiliki angkutan umum yang baik. selain itu kebutuhan jalan dan kepadatan penduduk pada suatu kota yang juga tidak seimbang [11].

Dampak yang dihasilkan menimbulkan efek negatif yang tidak sedikit. Dari sisi ekonomi, kemacetan dapat menghambat produksi dan distribusi barang sehingga memperlambat laju perekonomian nasional. Kemacetan lalu lintas yang dihadapi pekerja kantoran setiap hari dapat mempengaruhi kondisi fisik dan psikologis mereka di tempat kerja. Kinerja karyawan tidak dapat mencapai hasil yang maksimal karena masalah kemacetan yang menguras tenaga dan pikiran. Sehingga kemacetan lalu lintas menimbulkan berbagai dampak negatif baik bagi pengemudi itu sendiri maupun dari aspek ekonomi dan lingkungan. Bagi pengemudi, kemacetan lalu lintas menimbulkan ketegangan (stres). Dampak negatif lainnya adalah hilangnya waktu karena perjalanan jauh dan meningkatnya biaya operasional kendaraan yang diparkir. Selain itu dampak negatif terhadap lingkungan diwujudkan dalam bentuk pencemaran udara dan kebisingan kendaraan atau gangguan kebisingan [12].

2.3.1 Tracking Kemacetan

Tracking kemacetan merupakan suatu proses pemantauan dan pengumpulan data mengenai aktivitas lalu lintas yang bertujuan untuk mengidentifikasi dalam memahami tingkat kemacetan yang terjadi pada jalur tertentu. Dalam hal ini melibatkan peranan teknologi dan metode tertentu dalam pengamatannya

sehingga dapat menganalisis serta melacak aliran lalu lintas serta faktor-faktor lain yang berkontribusi terhadap keadaan lalu lintas.

Penggunaan teknologi android dengan menggunakan fitur *Global Positioning System* (GPS) yang digunakan untuk memprediksi keadaan lalu lintas pada suatu jalan yang menerapkan suatu model arus lalu lintas dengan rekam jejak kendaraan dari data GPS. Dalam hal ini maka dapat terdeteksi dalam radius yang berdekatan dengan informasi data pergerakan lambat launnya suatu kendaraan, jika hal tersebut terjadi maka dapat dikategorikan kemacetan pada jalan. Penggunaan GPS ini dapat dimanfaatkan untuk mengetahui titik posisi koordinat pengguna jalan [13].

2.4 Aplikasi

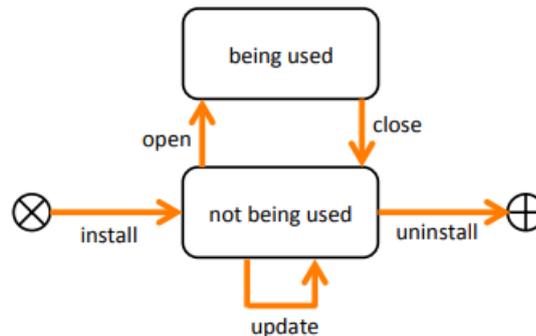
Aplikasi merupakan suatu program yang dibuat dan dirancang oleh manusia untuk membantu melakukan sesuatu yang dapat membantu manusia dalam menjalankan aktivitas. Istilah aplikasi itu sendiri pada dasarnya merupakan dari bahasa inggris yaitu *application* yang artinya penerapan atau penggunaan. Aplikasi tersebut digunakan sebagai program yang telah tersedia untuk melakukan suatu fungsi bagi sasaran yang akan dituju.

Aplikasi juga merupakan suatu program yang dikembangkan untuk memenuhi kebutuhan dari pengguna sebagai perangkat lunak yang menjalankan suatu pekerjaan tertentu, sehingga aplikasi dapat dikatakan suatu program yang dibuat dalam sebuah perangkat lunak dengan komputer untuk memudahkan pekerjaan pengguna seperti penerapan, penggunaan dan penambahan suatu data [14].

2.4.1 Aplikasi Mobile

Aplikasi *Mobile* merupakan suatu aplikasi dari perangkat lunak yang berjalan pada perangkat mobile seperti Smartphone, Tablet, Ipad, serta memiliki sistem yang dapat mendukung perangkat lunak sehingga dapat digunakan untuk *single user* maupun *multi user*. Pada platform pendistribusianya aplikasi mobile yang tersedia dikelola oleh pemilik bisnis dari mobile operating system seperti App Store (iOS) atau Google Play Store. Adapun Aplikasi yang sudah terinstall

sebelumnya dalam perangkat mobile, adapun yang diunduh melalui tempat pendistribusian [15].



Sumber Gambar:

https://brenthecht.com/publications/bhecht_mobilehci2011_sleepbirds.pdf

Gambar 2.1 Lifecycle Aplikasi Mobile

Dalam hal ini mengakses suatu informasi secara cepat melalui *smartphone* yang terkoneksi dengan internet melalui aplikasi mobile memberikan kemudahan pengguna dalam menggunakan aplikasi mobile untuk mendapatkan suatu informasi yang penting secara portable tanpa menggunakan alat lain seperti PC ataupun netbook, sehingga dalam pemanfaatannya dapat memperoleh informasi yang terbaru dapat terpenuhi tanpa adanya halangan waktu dan tempat keberadaan perangkat mobile serta area yang dapat menjangkau koneksi internet [15]. Selain itu untuk mengakses sebuah website dapat dilakukan menggunakan aplikasi mobile pengguna, yang dimana ukuran suatu layar akan secara otomatis menyesuaikan dengan ukuran halaman website dengan versi mobile sehingga, mengurangi jumlah kapasitas maksimum dari suatu kegiatan atau bahkan tidak memerlukan jumlah bandwidth yang besar [16].

2.5 Android

Android merupakan suatu sistem operasi mobile yang dikembangkan oleh google. Sistem operasi android dibangun dan dirancang untuk digunakan pada perangkat mobile seperti *smartphone* dan tablet. Antar muka yang ditawarkan android memungkinkan pengguna yang intuitif serta mudah digunakan dengan dukungan aplikasi yang luas.

Pada sejarahnya android pertama kali dikembangkan oleh perusahaan yang bernama Android Inc. pada tahun 2003. Perusahaan tersebut pada tahun 2005 dibeli oleh perusahaan google. Sehingga pada tahun 2007. Google resmi mengumumkan bahwa akan mengeluarkan sistem operasi Android sebagai proyek *open source*. Versi pertama yang dirilis yaitu Android 1.0. Hingga saat ini sistem operasi Android merupakan yang menjadi pilihan banyak vendor, karena memiliki biaya lisensi yang terjangkau dan sifatnya *semi-open source*. Sifat Android yang terbuka juga telah mendorong komunitas besar pengembang perangkat lunak untuk menggunakan open source sebagai dasar proyek pengembangan aplikasi, Menambahkan fitur baru untuk pengguna tingkat lanjut atau untuk digunakan pada perangkat *Android* yang dirilis secara resmi dengan sistem operasi yang lain. Selain itu ekosistem yang ada pada android terdiri dari berbagai perangkat, aplikasi, dan layanan yang didukung oleh google serta pengembang dari pihak ketiga [17].

Evaluasi dan perkembangan menjadi salah satu kunci dari kesuksesan android dalam menarik hati para pengguna smartphone. Selain itu penamaan versi android menggunakan nama-nama dari makanan menjadi android semakin unik dan pengguna dapat dengan mudah mengingat tipe android. beberapa versi android di antaranya sebagai berikut:

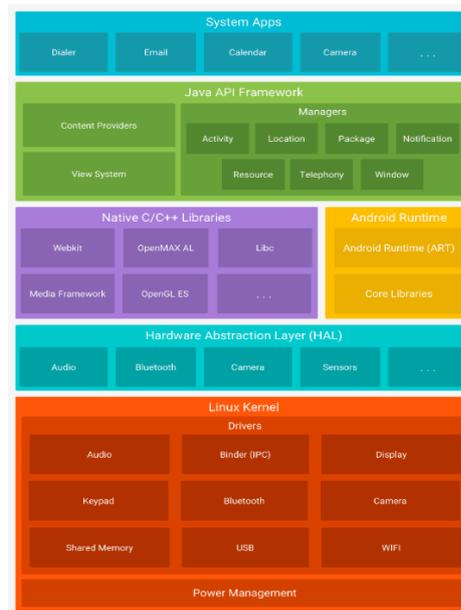
Tabel 2. 1 Versi OS Android

Nama OS	Versi	API level
Oreo	8.0	Api level 26
Nougat	7.1	API level 25
Nougat	7.0	API level 24
Marshmallow	6.0	API level 23
Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 - 4.4.4	API level 19
Jellybean	4.3.x	API level 18
Jellybean	4.2.x	API level 17

Jellybean	4.1.x	API level 16
Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Honeycomb	3.2.x	API level 13
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.0	API level 11
Gingerbread	2.3.3 - 2.3.7	API level 10
Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Froyo	2.2.x	API level 8, NDK 4
Eclair	2.1	API level 7, NDK 3
Eclair	2.0.1	API level 6
Eclair	2.0	API level 5
Donut	1.6	API level 4, NDK 2
Cupcake	1.5	API level 3, NDK 1

2.5.1 Arsitektur Android

Pada dasarnya Android dibangun menggunakan *object oriented*, dimana elemen-elemen dari penyusun sistem operasi yang berupa objek, sehingga dapat digunakan kembali atau sering disebut reusable. Untuk membuat aplikasi dengan baik tentunya harus mengetahui bagaimana arsitektur yang ada pada Android. Pada Gambar 2.2 merupakan gambaran arsitektur Android. Secara garis besar arsitektur android terdiri dari beberapa layer komponen, yaitu:



Sumber Gambar:

Sumber Gambar: <https://developer.android.com/guide/platform/index.html?hl=id>

Gambar 2. 2 Arsitektur Android

a. Linux Kernel

Linux Kernel merupakan lapisan yang paling bawah dari susunan Android sebagai penyedia suatu perantara bagi hardware dengan lapisan tepat di atasnya. Dengan berdasarkan pada Linux versi 2.6, Linux kernel menyediakan jenis multitasking yang memungkinkan program berbagi operasi (OS) dengan sumber daya perangkat keras yang mendasarinya atau disebut dengan *preemptive multitasking* seperti *memory*, *process*, dan *power management*. Kernel tersebut selain multitasking juga menyediakan tingkatan jaringan untuk perangkat keras (*hardware*) [18].

b. *Hardware Abstraction Layer* (HAL)

Hardware Abstraction Layer merupakan elemen yang menyediakan antarmuka standar yang memaparkan kemampuan perangkat keras internal perangkat ke API Java tingkat yang lebih tinggi. Diantaranya terdiri dari beberapa modul pustaka, yang masing-masing mengimplementasikan antarmuka dengan komponen perangkat keras tertentu seperti kamera atau modul Bluetooth. Saat

API *framework* memanggil untuk mengakses hardware, sistem Android memuat modul *library* untuk *hardware* tersebut [18].

c. Android Runtime

Android Runtime Merupakan lapisan yang membuat aplikasi Android berjalan dalam menggunakan implementasi Linux dalam prosesnya. Dalvik Virtual Machine (DVM) adalah mesin yang mendasari framework aplikasi Android. Android Runtime terdiri dari dua bagian, yaitu: salah satu lapisan yang membuat aplikasi Android berjalan, dimana prosesnya menggunakan implementasi Linux [18]. Dalvik Virtual Machine (DVM) adalah mesin yang mendasari framework aplikasi Android. Android Runtime terdiri dari dua bagian yaitu:

1. Core Libraries

Aplikasi *core library Android* dibuat dengan bahasa pemrograman Java. Sedangkan Dalvik sebagai mesin virtual bukan mesin virtual Java, sehingga diperlukan *library* untuk menerjemahkan bahasa Java/C yang ditangani oleh Library inti.

2. Dalvik Virtual Machine (DVM)

Mesin virtual berbasis register yang dioptimalkan untuk eksekusi yang efisien, peningkatan yang membuat kernel Linux mampu melakukan chaining dan manajemen tingkat rendah.

d. Native C/C++ Library

Beberapa komponen juga layanan sistem Android seperti *Android Runtime* dan *Hardware Abstraction Layer* yang memerlukan *library* asli yang tertulis dalam C atau C++. Sehingga memungkinkan kerja java API untuk mengekspos fungsionalitas dari beberapa library asli pada suatu aplikasi [18]. Beberapa library yang tersedia pada Android untuk Developer diantaranya sebagai berikut:

1. **android.app** untuk memberikan akses ke model aplikasi dari Android.
2. **android.content** untuk memfasilitasi akses konten, mempublikasikan, dan mengirim pesan antar aplikasi pada komponen aplikasi.
3. **android.database** untuk mengakses data yang dipublikasikan oleh content providers

4. **android.opengl** merupakan antarmuka Java ke Api penerjemah OpenGL ES 3D Graphic.
5. **android.os** sebagai penyedia aplikasi ke OS sistem service standar.
6. **android.text** untuk menyajikan dan memanipulasi teks yang ditampilkan pada perangkat
7. **android.view** merupakan dasar dari pembangunan user interface aplikasi.
8. **android.widget** merupakan koleksi dari beberapa user interface yang telah dibuat.
9. **android.webkit** merupakan sekumpulan kelas untuk memungkinkan kemampuan web browsing yang akan dibangun ke dalam aplikasi.

e. Java API Framework

Kerangka kerja aplikasi merupakan sekumpulan layanan yang bersama-sama membentuk lingkungan tempat aplikasi *Android* dijalankan dan dikelola. *Framework* ini menerapkan prinsip saling berbagi dan saling melengkapi, dimana sebuah aplikasi dapat saling berbagi fungsionalitas beserta informasi terkait untuk digunakan oleh aplikasi lainnya [16]. Beberapa program penting pada Application Framework ini diantaranya:

1. *Activity* Manager digunakan untuk mengontrol semua aspek dari siklus hidup aplikasi dan *Activity* Stack.
2. Content Providers digunakan untuk memberi izin aplikasi untuk mempublikasikan dan berbagi data dengan aplikasi lainnya.
3. Resource Manager digunakan untuk memberikan akses kepada resources yang bukan kode seperti strings, setting warna, dan layout User Interface.
4. Notifications Manager digunakan untuk membuat aplikasi dapat menampilkan pengingat dan notifikasi kepada pengguna.
5. View System digunakan untuk membuat User Interface aplikasi.

- a. `onCreate()`: Merupakan fungsi yang pertama kali dipanggil ketika membuka sebuah *Activity*.
- b. `onStart()`: Merupakan fungsi yang dipanggil setelah *Activity* dihentikan, sebelum dimulai lagi. Selalu diikuti oleh `onStart()`.
- c. `onResume()`: Merupakan fungsi yang dipanggil saat aktivitas akan mulai berinteraksi dengan pengguna. Pada titik ini *Activity* berada di puncak tumpukan aktivitasnya, dengan masukan pengguna. Selalu diikuti oleh `onPause()`.
- d. `onPause()`: Merupakan aktivitas untuk menghentikan aplikasi secara sementara hal ini disebabkan oleh peralihan ke aplikasi lain, tetapi pengguna masih dapat melihatnya.
- e. `onStop()`: Merupakan situasi yang terjadi saat aplikasi tidak terlihat oleh pengguna ataupun situasi dimana sudah dikeluarkan akan tetapi meninggalkan service di background [19].

2.5.3 Android Studio

Android studio merupakan lingkungan pengembangan terintegrasi (IDE) resmi untuk sistem operasi Google Android. Android Studio dibangun di atas perangkat lunak dan dirancang khusus untuk pengembangan Android. Sistem operasi berbasis Windows, macOS dan Linux dapat mengunduh Android studio. Ini menggantikan Eclipse Android Development Tools (ADT) sebagai IDE utama untuk pengembangan aplikasi Android [18].

Peneliti menggunakan Android Studio sebagai kode editor dan IDE (*Integrated Development Environment*) untuk pengembangan aplikasi android. Kotlin dan Java merupakan bahasa pemrograman yang bisa digunakan untuk mengembangkan aplikasi Android. Namun merekomendasikan Kotlin sebagai bahasa pemograman yang digunakan. Android Studio memiliki fitur yang lengkap untuk mengembangkan aplikasi seperti berikut:

1. *Template*: Merupakan *template* memulai *project* maupun *Activity* tanpa harus membuatnya dari nol.

2. Intelligent code editor: Merupakan code completion yang memudahkan penulisan kode dengan cepat tanpa harus menuliskannya secara lengkap. Selain itu, ada juga peringatan apabila terdapat kesalahan penulisan kode.
3. Design tool: Digunakan untuk mendesain aplikasi serta melihat preview secara langsung sebelum dijalankan.
4. Flexible build system: Android Studio menggunakan Gradle yang fleksibel untuk menciptakan build varian yang berbeda untuk berbagai device. sehingga juga dapat menganalisa prosesnya secara mendetail.
5. Emulator: Digunakan untuk menjalankan aplikasi tanpa harus menggunakan device Android. Dilengkapi dengan Instant Run untuk melihat perubahan tanpa harus build project dari awal.
6. Debugging: memudahkan untuk mencari tahu masalah.
7. Testing: Digunakan untuk menjalankan pengujian untuk memastikan semua kode aman sebelum rilis.
8. Publish: Digunakan untuk membuat berkas AAB/APK dan menganalisanya guna dibagikan dan di-publish ke PlayStore.
9. Integrasi: Terhubung dengan berbagai layanan yang memudahkan untuk mengembangkan aplikasi, seperti Github, Firebase, dan Google Cloud.

2.6 Deteksi Objek

Pendeteksian objek (Object Detection) merupakan sebuah metode dalam bidang visi komputer yang bertujuan untuk mengidentifikasi objek-objek dalam gambar atau video. Pada umumnya dalam pendeteksian objek menggunakan algoritma teknik pembelajaran mesin atau deep learning untuk memberikan hasil yang signifikan. Pendeteksi objek dapat dengan mudah mengenali dan menemukan objek dalam waktu singkat ketika manusia melihat sebuah gambar atau video. Metode ini berbeda dengan komputer yang memerlukan komputasi yang kompleks. Pendeteksian objek memiliki tujuan untuk meniru kecerdasan manusia dalam memproses informasi visual menggunakan komputer. Pendeteksian objek melibatkan penempatan kotak pembatas (bounding box) di sekitar objek yang terdeteksi dalam gambar selama proses kerjanya. Hal ini umumnya melibatkan dua tahap, tahap pertama adalah pengklasifikasian jenis

objek yang terdapat dalam gambar, kemudian tahap selanjutnya menarik kotak pembatas di sekitar objek tersebut. Klasifikasi gambar dan pendeteksian objek secara umum sebenarnya memiliki kemiripan. Klasifikasi gambar memiliki fungsi untuk mengelompokkan gambar ke dalam kategori tertentu, sementara pendeteksian objek bertujuan untuk mengidentifikasi lokasi objek dalam gambar, misalnya menghitung jumlah objek yang sama dalam gambar tersebut. [20].

2.6.1 TensorFlow

TensorFlow adalah suatu rangkaian deep learning yang berperan sebagai perpustakaan untuk ilmu data yang gratis dan sumber terbuka yang dikembangkan oleh peneliti dari tim Google. TensorFlow dapat digunakan dalam berbagai bidang seperti dalam bidang deteksi objek, yakni terdapat API deteksi objek TensorFlow yang merupakan alat yang membantu dalam proses pembuatan, pelatihan, dan implementasi model deteksi objek. API deteksi objek TensorFlow menyediakan model deteksi objek yang telah dilatih kepada pengguna, tetapi juga tidak menutup kemungkinan bila pengguna dapat menggunakan model deteksi objek lain yang telah dilatih seperti Faster R-CNN, SSD, Retinanet, Resnet50, dan sebagainya. [21]. Jenis-jenis TensorFlow diantaranya yaitu sebagai berikut:

1. TensorFlow Lite

Merupakan versi TensorFlow yang optimal untuk mengeksekusi model pembelajaran mesin pada suatu perangkat dengan sumber daya terbatas, seperti pada *smartphone*, *Internet Of Things*, dan *microcontroller*.

2. TensorFlow Extended (TFX)

Merupakan platform TensorFlow yang dirancang sebagai pengelola model pembelajaran mesin pada lingkungan produksi. TFX juga menyediakan komponen untuk dikelola.

3. TensorFlow.js

Merupakan versi TensorFlow yang berjalan pada browser ataupun di lingkungan JavaScript. TensorFlow.js dapat membangun serta melatih model pembelajaran mesin di browser.

4. TensorFlow Probability

Merupakan perpustakaan dari TensorFlow yang dapat menyediakan suatu alat model probabilistik dan statistik, sehingga memungkinkan pengguna membangun serta melatih model probabilistik dan menganalisis statistik.

5. TensorFlow Hub

Merupakan platform yang menyediakan model pembelajaran mesin yang telah dilatih. Sehingga pengguna dapat mencari, mengunduh, hingga dapat menggunakan model.

2.7 Application Programming Interface (API)

Application Programming Interface (API) adalah data yang terdiri dari antarmuka, fungsi, kelas, struktur, dan lainnya untuk membangun perangkat lunak. API memudahkan developer untuk melakukan “*unpack*” *software*, yang kemudian dapat dikembangkan atau diintegrasikan dengan *software* lain. Menjadi penghubung antara suatu aplikasi dengan aplikasi lain yang memungkinkan untuk developer mengakses fungsi-fungsi sistem juga merupakan salah satu fungsi API. Proses ini dikendalikan oleh sistem operasi. Kelebihan dari API ini adalah memungkinkan sebuah aplikasi untuk berkomunikasi dengan satu sama lain dengan aplikasi lain [22].

Selain itu, API juga dapat disesuaikan sepenuhnya, berbasis komponen, atau dikembangkan sesuai dengan standar industri untuk memastikan interoperabilitas. Beberapa API harus didokumentasikan, dimaksudkan untuk "dijelajahi" atau untuk menentukan fungsi yang didukung. Karena komponen atau sistem lain hanya bergantung pada API, sistem yang menyediakan API secara ideal dapat mengubah data di dalam API tanpa memengaruhi penggunanya [22].

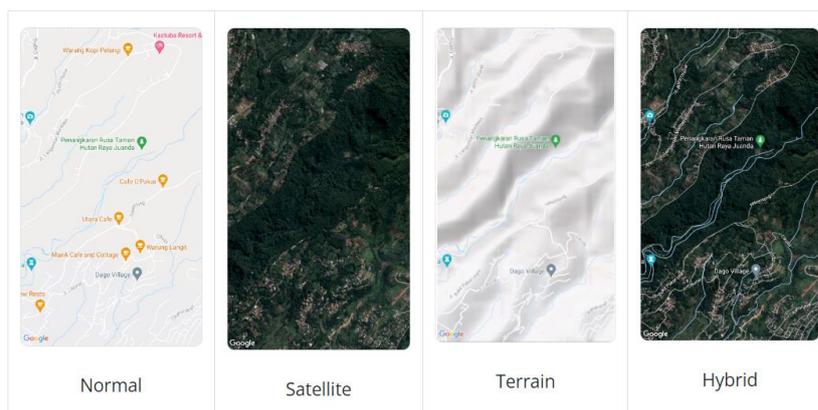
2.7.1 Google Maps API

Google maps memberikan layanan dalam bentuk pemetaan yang dikembangkan oleh Google. Layanan tersebut mencakup berupa citra satelit, peta pada jalan, panorama, kondisi lalu lintas pada suatu jalan, dan perencanaan rute untuk menuju ke suatu tujuan dengan cara berjalan kaki, mobil, sepeda, maupun angkutan umum. Pada umumnya, Google maps digunakan pada aplikasi web yang

bertujuan untuk menampilkan lokasi, memberikan petunjuk arah, atau rute menuju tujuan perjalanan [8].

Untuk menampilkan Google Maps di website atau blog sangat mudah. Dalam pengimplementasiannya hanya membutuhkan keterampilan HTML dan JavaScript serta koneksi internet yang sangat stabil. Kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal dengan menggunakan Google Maps API, sehingga kita dapat fokus hanya pada data yang ditampilkan. Dengan kata lain, kita hanya menghasilkan data ketika peta yang ditampilkan merupakan milik Google, sehingga kita tidak perlu untuk membuat peta suatu tempat, bahkan peta dunia [11]. Namun Google Maps mulai dikembangkan sebagai aplikasi khusus yang dapat diakses dengan mudah dan lebih praktis setelah adanya smartphone berbasis Android yang merupakan bagian dari Google Enterprise.

Selain dari sisi UI, kita juga terlebih dahulu perlu membuat object Google Maps pada sisi kodenya untuk mengunduh dan menampilkan data pada peta. Tidak hanya itu saja, object ini juga memungkinkan pengguna gunakan untuk berinteraksi dengan peta yang sedang ditampilkan seperti mengatur jenis dan style maps, menyeleksi data yang ditampilkan, menambahkan penanda (marker), menyesuaikan zoom dan animasi, merespon tindakan dari pengguna, baik berupa *pinch*, *drag*, maupun klik. Adapun tipe-tipe peta yang dapat dipilih, antara lain:



Gambar 2. 4 Tipe-tipe Peta

- a. **Normal** merupakan tampilan peta yang fokus menunjukkan jalan dan sungai.
- b. **Satellite** merupakan tampilan peta hasil fotografi dari satelit. Jalan dan data lokasi tidak tampil di sini.

- c. **Terrain** merupakan tampilan peta topografi yang memperlihatkan kontur dan warna ketinggian suatu tempat.
- d. **Hybrid** merupakan tampilan peta hasil fotografi dari satelit disertai dengan jalan dan data lokasi.

2.7.2 UI Control pada Peta

Selain menampilkan peta, Google Maps juga menawarkan beberapa kontrol yang memudahkan untuk melihat peta. pengguna dapat mengaktifkan atau menonaktifkan kontrol ini menggunakan kelas *UI Settings* dengan objek Google Maps. Berikut adalah beberapa kontrol yang dapat dikonfigurasi:

- a. Zoom Control yang digunakan untuk mengatur zoom in (memperbesar) atau zoom out (memperkecil).
- b. Indoor Level Picker merupakan fitur yang jarang diketahui. Jika suatu bangunan memiliki lebih dari satu lantai dan informasinya disimpan di Google Maps, peta akan memberikan opsi lantai yang ingin Anda lihat.
- c. Compass digunakan apabila peta di rotasi menggunakan dua jari, biasanya akan tampil ikon kompas yang menunjukkan arah mata angin.
- d. Map Toolbar digunakan ketika memilih suatu marker (penanda), Google Map akan menampilkan mini toolbar berisi dua buah aksi, yakni membuka navigasi ke lokasi tersebut atau membuka lokasi pada aplikasi Google Maps. Lokasinya ada pada pojok kanan bawah sebelah zoom.
- e. Tombol My Location digunakan untuk menandai lokasi dan mengontrolnya. Maka dari itu memerlukan hak khusus untuk menggunakan fitur ini.

2.7.3 BMKG API

Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) adalah lembaga negara yang berstatus non kementerian (LPND) yang dipimpin oleh kepala lembaga tersebut. BMKG antara lain mempunyai tugas seperti menyelenggarakan tugas penyelenggaraan negara di bidang meteorologi, iklim, kualitas udara, dan geofisika sesuai dengan peraturan perundang-undangan yang berlaku. Adapun tugas dari BMKG yaitu mengamati dan memahami fenomena meteorologi, iklim, kualitas udara, dan geofisika. Selain itu, BMKG juga menyediakan data,

informasi, dan layanan meteorologi, iklim, kualitas udara, dan geofisika yang andal [23]. Adapun cara kerja BMKG dalam memprediksi cuaca yaitu:

1. Pengamatan Unsur-Unsur Cuaca

BMKG biasanya melakukan pengamatan unsur cuaca dengan bantuan dinas khusus BMKG, bandara, pelabuhan, kapal laut dan perkebunan.

2. Pengamatan Kualitas Udara dan Unsur Medan Bumi

BMKG juga melakukan pengamatan kualitas udara di stasiun GAW (Global Atmosphere Watch) dan pengamatan elemen Bumi seperti percepatan tanah, gempa bumi dan magnet bumi di Stasiun Geofisika.

3. Pengamatan Menggunakan Alat-Alat Canggih

Alat canggih untuk mendukung analisis dan peramalan data dikerahkan di semua area di luar jangkauan stasiun pengamatan. Misalnya, pendeteksi petir, melacak pergerakan awan, pengukur hujan otomatis, pengukuran otomatis elemen cuaca.

4. Menganalisis Hasil Pengamatan

Data pengamatan yang dihasilkan oleh stasiun dan alat canggih kemudian ditransfer ke server komputer dan diatur secara sistematis.

5. Membagikan Informasi Prediksi Cuaca

Hasil analisis pengamatan yang dilakukan BMKG kemudian dibagikan ke masyarakat sebagai informasi prediksi cuaca.

2.8 Global Positioning System (GPS)

Global Positioning System atau yang sering disebut GPS adalah sistem navigasi elektronik yang menerima data dari 12 satelit untuk memungkinkan GPS dapat menghitung posisi kita di Bumi. Satelit GPS melakukan pemancaran lokasi satelit dan jarak penerima GPS kita dari satelit bukan mengirimkan informasi lokasi pengguna [7].

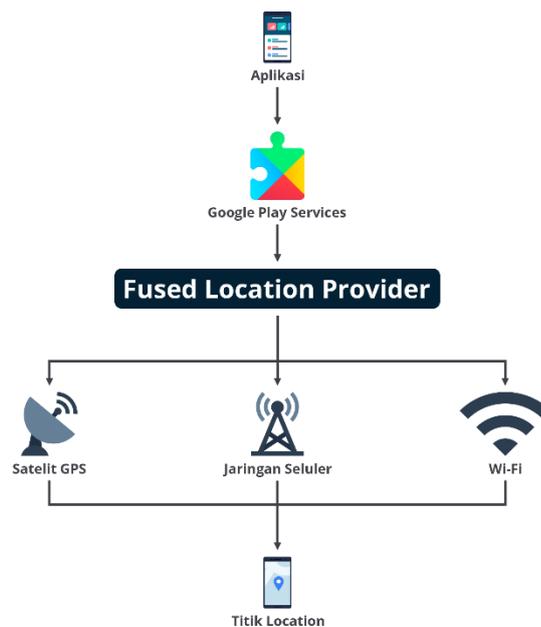
Beberapa hal yang dapat dilakukan oleh penerima GPS yakni merekam informasi rute seperti jarak tempuh total, kecepatan rata-rata, kecepatan maksimum, kecepatan minimum, waktu/jam untuk mencapai tujuan, dan lainnya.

Keakuratan posisi GPS yang ditunjukkan memiliki faktor kesalahan, atau disebut juga sebagai tingkat akurasi. Misalnya, perangkat GPS menampilkan koordinat dengan akurasi 5 meter, maka lokasi pengguna mungkin dapat berada dalam jarak 5 meter dari titik yang ditampilkan[13].

2.8.1 Fused Location Provider

Fused Location Provider adalah solusi terbaru untuk mendapatkan informasi lokasi tanpa perlu khawatir dengan jenis sumber lokasi yang dipilih. Informasi lokasi dapat diperoleh dari GPS satelit, Internet melalui BTS, atau jaringan Wi-Fi. Library ini secara otomatis memilih sumber data berdasarkan persyaratan akurasi dan efisiensi baterai.

Fused Location Provider tentunya berbeda dengan pendahulunya yaitu *Location Manager*. Pada library tersebut tentunya harus menggunakan sumber data tersendiri, baik itu melalui GPS, network, atau bahkan sensor. Hal tersebut tentunya tidak efisien dan lambat jika salah dalam mengaturnya. Untuk meminta pembaruan lokasi Method `requestLocationUpdates` dari Fused Location Provider dibutuhkan tiga parameter `GoogleApiClient`, `LocationRequest` yang digunakan untuk mengatur aspek API FusedLocation yaitu `GoogleApiClient`, `LocationRequest`, `LocationListener` [24].



Gambar 2.5 Fused Location Provider

Dalam hal ini *Fused Location Provider* memiliki 3 jenis scenario yang dapat dimanfaatkan yaitu:

1. *Last Known Location*

Last Known Location digunakan untuk mengetahui lokasi terakhir dari perangkat yang digunakan untuk mengetahui lokasi yang diperlukan saja, seperti membagikan lokasi saat ini. Library ini otomatis akan mengambil data lokasi terakhir yang pernah di cari oleh perangkat tanpa perlu mengakses ulang, sehingga dalam pemrosesannya akan sangat cepat dan dapat menghemat baterai.

2. *Current Location*

Digunakan untuk mengetahui lokasi saat ini dengan data terbaru yang lebih akurat. Penggunaan ini nantinya akan mengakses sensor secara langsung yang menyebabkan proses menjadi lebih lama.

3. *Location Update*

Digunakan untuk mengetahui lokasi terbaru dalam jangka waktu tertentu. Biasa digunakan untuk berbagi lokasi terkini dan melacak posisi driver yang sedang berjalan [24].

2.8.2 Location Tracker

Salah satu kelebihan dalam membangun aplikasi Android dibandingkan platform lain seperti web dan desktop adalah sensor lokasi. Dengan mengetahui lokasi pengguna, kita dapat membuat aplikasi yang lebih akurat dalam menyelesaikan masalah pengguna. Misalnya mengetahui lokasi pengemudi pada aplikasi ojek berbasis web atau menyimpan rute lari dan bersepeda. Untuk menggunakan sensor lokasi, ada dua jenis akurasi lokasi yang dapat digunakan, yaitu:

1. *Precise*

Digunakan untuk mendapatkan lokasi yang sangat presisi dengan akurasi 4-50 meter. Cocok digunakan untuk melacak objek dan detail alamat. Permission yang dibutuhkan adalah `ACCESS_FINE_LOCATION`.

2. *Approximate*

Digunakan untuk menentukan perkiraan lokasi dengan akurasi hanya 1 mil (1,6 km). Sangat tepat digunakan untuk keperluan di media social atau cuaca di sekitar. Permission yang dibutuhkan dalam akses tersebut adalah ACCESS_COARSE_LOCATION [24].

2.8.3 Location Settings

Location settings digunakan ketika kita ingin mendapatkan data lokasi, beberapa kriteria dapat diatur sesuai kebutuhan menggunakan *object LocationRequest* [24]. Beberapa konfigurasi yang bisa pengguna atur antara lain:

- a. *Update interval* digunakan untuk menentukan jumlah slot waktu dan untuk mengambil data lokasi dalam mili detik. Perhatikan bahwa interval ini tidak tetap (bisa lebih lambat atau lebih cepat) untuk memastikan penggunaan baterai yang efisien.
- b. *Fastest Update interval* dipergunakan untuk mengambil data interval lokasi
- c. Priority untuk mengatur seberapa penting request yang dijalankan.

2.9 CCTV M3U8

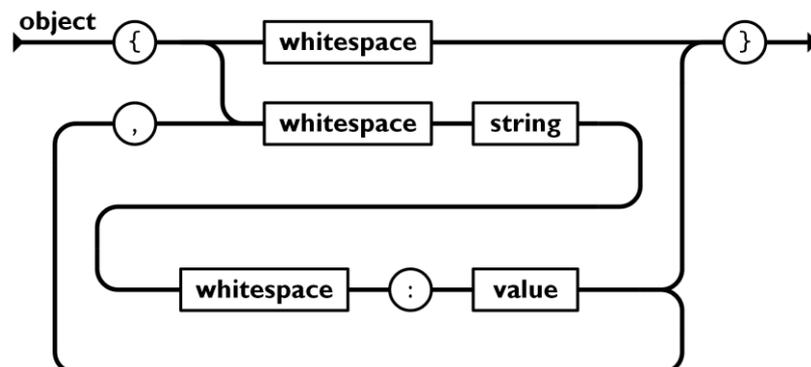
File **m3u* adalah metadata daftar putar yang mereferensikan file *.mp3* dan menyediakan metadata tambahan untuk item daftar putar. Setiap trek dan lokasi memiliki jalurnya sendiri. M3U (Moving Picture Experts Group Audio Layer 3 Uniform Resource Locator, URL MP3) adalah format file komputer yang menyimpan daftar putar multimedia. File ini pada mulanya bekerja di Winamp, meskipun sekarang telah didukung oleh banyak aplikasi seperti VLC media player, XMMS, foobar2000, juk, RealPlayer, Windows Media Player, iTunes, QuickTime Player, Yahoo! Mesin Musik, JetAudio, RokuLabs SOUNDBRIDGE, Spider Player, dan PlayStation Portable. Pada ekstensi M3U, tanda "#" juga mengarah pada ekstensi file M3U. Penggunaan umum format file M3U adalah untuk membuat file daftar putar trek tunggal untuk streaming online. File-file ini dirancang untuk memudahkan streaming dan dapat digunakan untuk hal-hal seperti mengunduh dari situs web, mengirim email, atau mendengarkan radio internet. Karakter Unicode UTF-8 merupakan salah satu muatan dari format M3U Versi Unicode Format M3U8 [25].

2.10 JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) adalah format file standar terbuka dan format pertukaran data yang menggunakan teks yang dapat dibaca manusia untuk menyimpan dan mengirimkan objek data yang terdiri dari pasangan atribut-nilai dan tipe data array. Ini adalah format data yang sangat umum di banyak aplikasi, misalnya sebagai pengganti XML dalam sistem AJAX. JSON adalah format transfer data sederhana. Selain itu, JSON adalah format data yang bebas bahasa. JSON merupakan turunan dari JavaScript, tetapi banyak bahasa pemrograman modern menyertakan kode untuk membuat dan mengurai format data JSON [7]. Application atau json merupakan media web resmi untuk JSON. JSON adalah gabungan pasangan nama atau nilai. Dalam beberapa bahasa, ini direpresentasikan sebagai objek, rekaman, struktur, kamus, tabel hash, daftar kunci, atau tabel asosiatif. Daftar nilai terurut direpresentasikan sebagai larik, vektor, daftar, atau urutan dalam sebagian besar bahasa. Dalam hal ini JSON menggunakan bentuk sebagai berikut:

a. Objek

Objek adalah pasangan nama/nilai yang tidak diurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti oleh : (titik dua) dan setiap pasangan nama/nilai dipisahkan dengan koma (koma). Berikut merupakan gambaran objek pada JSON pada Gambar 2.6 JSON Objek:

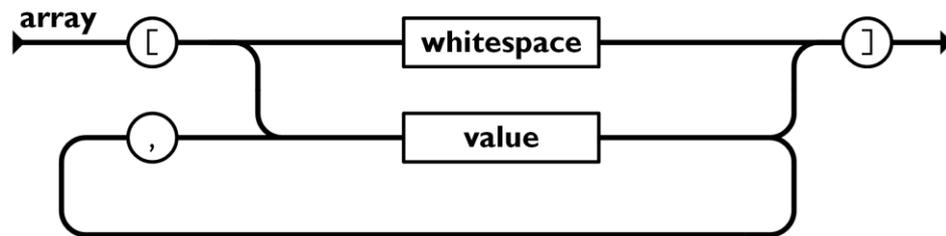


Sumber Gambar: <https://www.json.org/json-en.html>

Gambar 2. 6 JSON Objek

b. Larik

Larik adalah susunan nilai yang teratur. Baris dimulai dengan [(membuka tanda kurung siku) dan diakhiri dengan] (menutup tanda kurung siku). Setiap nilai dipisahkan dengan karakter (koma). Berikut merupakan gambaran Larik pada JSON pada Gambar 2.7 JSON Larik:

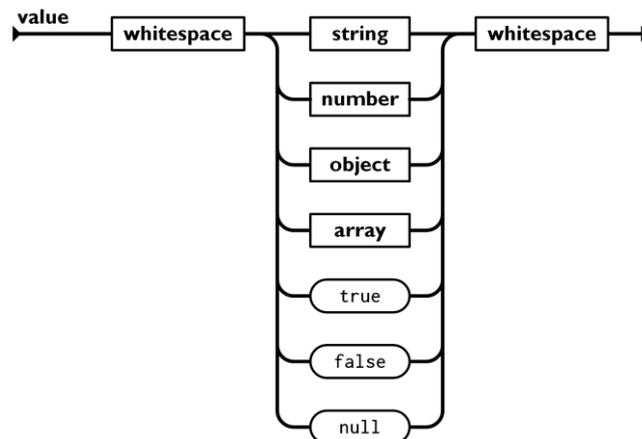


Sumber Gambar: <https://www.json.org/json-en.html>

Gambar 2. 7 JSON Larik

c. Nilai

Nilainya bisa berupa string atau angka yang dikutip, benar atau salah atau nol, atau objek atau larik. Struktur ini dapat diatur secara bertahap. Berikut merupakan gambaran dari nilai JSON pada Gambar 2.8 JSON Nilai:



Sumber Gambar: <https://www.json.org/json-en.html>

Gambar 2. 8 JSON Nilai

d. String

String adalah kumpulan dari nol atau lebih karakter Unicode yang dihimpit dengan tanda kutip. Di dalam string, dapat menggunakan garis miring "\" untuk membentuk karakter khusus. Sebuah karakter mewakili satu karakter dalam

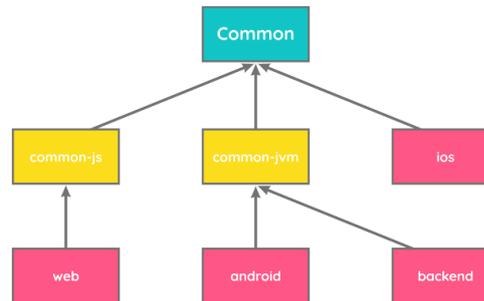
2.11 Kotlin

Kotlin merupakan suatu bahasa pemrograman berbasis Java Virtual Machine (JVM) yang dikembangkan oleh JetBrains. Kotlin adalah bahasa pemrograman pragmatis untuk Android yang menggabungkan pemrograman berorientasi objek (OO) dan pemrograman fungsional. Kotlin juga merupakan bahasa pemrograman yang dapat dioperasikan, memungkinkan penggunaanya untuk menggabungkan bahasa ini dengan bahasa pemrograman Java ke dalam satu proyek.

Bahasa pemrograman ini juga dapat digunakan untuk mengembangkan desktop, aplikasi berbasis web, dan bahkan aplikasi back-end. Beberapa keuntungan yang bisa didapat ketika pengembangan aplikasi berpindah ke Kotlin untuk pengembangan aplikasi di platform JVM adalah:

1. Dapat mengatasi *NullPointerException* yang umumnya terdapat pada Java.
2. Pada penulisannya umumnya menggunakan kotlin akan lebih ringkas di bandingkan dengan bahasa java.
3. Pada pembelajarannya kotlin tentunya mudah dipahami.
4. Dukungan IDE untuk mempermudah dalam pemrograman.

Kotlin menawarkan konsep lintas platform yang sedikit berbeda. Model distribusi kode saat ini memungkinkan banyak kode dengan fungsi serupa untuk dikelompokkan bersama. Misalnya, buat modul A yang berisi sekumpulan kode yang dapat digunakan semua platform, modul B yang berisi kode khusus platform, dan modul C yang menargetkan platform tertentu tetapi juga memiliki ketergantungan pada modul lain. Berikut gambaran modul pada kotlin yang akan di jelas akan pada Gambar pada Gambar 2.11 Modul Pada Project Kotlin:



Sumber Gambar: <https://www.dicoding.com/academies/80/tutorials/4036#>

Gambar 2. 11 Modul Pada Project Kotlin

Modul tersebut dapat kita bagi menjadi 3 (tiga) bagian sebagai berikut:

1. Common Module

Modul ini berisi kode yang tidak spesifik untuk platform apa pun. Di modul ini kita bisa meletakkan komponen-komponen yang bisa digunakan semua modul.

2. Platform Module

Pada modul ini kita bisa menambahkan komponen tertentu yang spesifik untuk satu platform. Biasanya modul ini merupakan implementasi dari common module.

3. Regular Module

Merupakan modul umum yang menargetkan platform tertentu. Modul ini bisa dependen atau pun menjadi dependensi dari platform module.

2.12 Node JS

Node.js adalah platform sumber terbuka yang memungkinkan pengembang menjalankan kode JavaScript di sisi server. Node.js mengizinkan pengembang menulis kode JavaScript untuk memproses permintaan klien dan menghasilkan respons sisi server. Ini sangat berguna karena JavaScript biasanya digunakan untuk membangun aplikasi web di sisi klien (front-end), sedangkan Node.js memungkinkan kita menggunakan JavaScript di sisi server (back-end) juga [26].

Node.js menggunakan model I/O non-blocking berbasis kasus yang memungkinkan aplikasi memproses beberapa permintaan secara bersamaan tanpa harus menunggu satu permintaan selesai sebelum memproses permintaan berikutnya. Dalam hal ini, keuntungan dari Node.js adalah menangani permintaan

terkait I/O, seperti permintaan untuk membaca atau menulis ke database, mengambil file dari server atau melakukan panggilan API di luar sistem.

Dalam hal ini, Node.js memiliki sistem modul yang memberikan kemungkinan pada pengembang menggunakan kode JavaScript yang ditulis oleh orang lain dan NPM (Node Package Manager) untuk mengelola dan memasang modul Node.js. NPM memungkinkan pengembang untuk mengelola dependensi yang digunakan dalam aplikasi dan memperbarui modul dengan mudah.

Node.js juga biasa digunakan untuk membangun aplikasi web menggunakan kerangka kerja seperti Express.js dan Hapi.js, yang menyediakan fitur seperti perutean, middleware, dan lainnya. Framework ini memfasilitasi pengembangan aplikasi web dengan menyediakan fungsi yang umum digunakan dan telah teruji di lapangan [26].

2.13 Database MySQL

MySQL adalah server RDBMS (Relational Database Management System). RDBMS adalah program yang memungkinkan pengguna basis data untuk membuat, mengelola, dan mengakses data dalam model relasional. Dengan demikian, tabel database memiliki hubungan antara satu tabel dengan tabel lainnya. MySQL AB menawarkan MySQL sebagai perangkat lunak gratis di bawah GNU *General Public License* (GPL). MySQL digunakan oleh banyak aplikasi web berbasis database seperti Drupal, Joomla, phpBB, dan WordPress. MySQL juga digunakan oleh berbagai situs web populer seperti Facebook, Flickr, MediaWiki, Twitter, dan YouTube [27].

MySQL telah lama menjadi turunan dari konsep database pusat yaitu SQL (Structured Query Language). SQL adalah konsep penggunaan basis data yang dirancang terutama untuk memilih dan memasukkan data, sehingga memungkinkan pemrosesan data otomatis yang mudah dan cepat. Keandalan sistem basis data (DBMS) tercermin dalam cara pengoptimal menangani perintah SQL yang diberikan oleh pengguna dan program aplikasi. Sebagai server basis data, dapat dikatakan bahwa MySQL lebih baik daripada server basis data lainnya dalam hal permintaan data. Hal ini terbukti untuk single user query, kecepatan

query MySQL bisa 10 kali lebih cepat dari PostgreSQL dan 5 kali lebih cepat dari Interbase [27].

2.14 XAMPP

XAMPP adalah perangkat open source gratis yang dikembangkan oleh Apache Friends sebagai solusi server web serbaguna, yang terutama terdiri dari Apache HTTP Server, database MariaDB, dan juru bahasa skrip yang ditulis dalam bahasa pemrograman PHP dan Perl untuk sebagian besar instalasi server web menggunakan komponen yang sama dengan XAMPP, yang memungkinkan pengguna untuk beralih dari server pengujian lokal ke server langsung. Secara resmi, perancang XAMPP hanya ingin menggunakannya sebagai alat pengembangan, sehingga perancang web dan pemrogram dapat menguji pekerjaan mereka di komputer mereka tanpa Internet. XAMPP dapat melayani halaman web di World Wide Web. Bagian terpenting dari paket ini adalah alat perlindungan kata sandi khusus. XAMPP juga mendukung pembuatan dan pengeditan database MariaDB dan SQLite.

XAMPP merupakan web server lokal dengan paket lengkap atau web server yang berdiri sendiri yang dapat bekerja secara mandiri, karena aplikasi XAMPP berisi puluhan paket modul, bahasa pemrograman dan komponen lainnya. Ini dibahas lebih detail di bagian Fitur XAMPP. Berkat adanya XAMPP, kita tidak perlu memasang terlalu banyak modul atau komponen terpisah di komputer Anda. sehingga merancang, menulis, dan menguji situs web bisa lebih mudah [11]. Ada lima komponen utama dalam XAMPP yaitu:

1. X – Cross Platform

X adalah singkatan dari XAMPP, yang merupakan singkatan dari cross. Ini pertanda bahwa XAMPP merupakan aplikasi lintas platform yang kompatibel dengan berbagai jenis sistem operasi.

2. A – Apache

3. Apache adalah server web default yang digunakan oleh XAMPP. Server web adalah perangkat lunak yang mengelola proses transfer data antara situs web dan pengunjung. Selain itu, Apache juga berperan sebagai penyalur fungsi atau komponen lainnya.

4. M – MariaDB

5. MariaDB adalah perangkat lunak manajemen basis data bawaan XAMPP. DBMS adalah perangkat lunak untuk mengelola basis data dan informasi yang dikandungnya. MariaDB menggantikan DBMS MySQL sebelumnya dari XAMPP versi 5.5.30 dan 5.6.1

6. P – PHP

PHP adalah bahasa pemrograman untuk membuat website atau website dari backend.

7. P – Perl

Tambahan terbaru untuk XAMPP adalah Perl. Perl adalah contoh bahasa pemrograman yang digunakan untuk tujuan yang lebih kompleks. Misalnya, mengembangkan aplikasi GUI, pemrograman web, dan masih banyak lagi.

2.15 Formula Haversine

Formula Haversine merupakan persamaan yang penting dalam sistem navigasi peta, Formula Haversine digunakan pertama kali oleh Jsoef de Menzoda pada tahun 1835 pada penelitiannya yang berjudul “Major Problems of Nautical Astronomy” dari Formula Haversine ini akan menghasilkan suatu jarak pendek antara dua titik lokasi yang akan memberikan jarak lingkaran besar. Penggunaan Formula Haversine merupakan suatu metode yang memperhitungkan dengan asumsi bahwa bumi bulat dan bukan suatu bidang datar melainkan suatu bidang yang memiliki lengkung derajat [28]. Berikut merupakan rumus haversine.

$$\begin{aligned}x &= (\text{lon}2 - \text{lon}1) * \cos((\text{lat}1 + \text{lat}2)/2) \\y &= (\text{lat}2 - \text{lat}1) \\d &= \text{sqrt}(x * x + y * y) * R\end{aligned}$$

Keterangan:

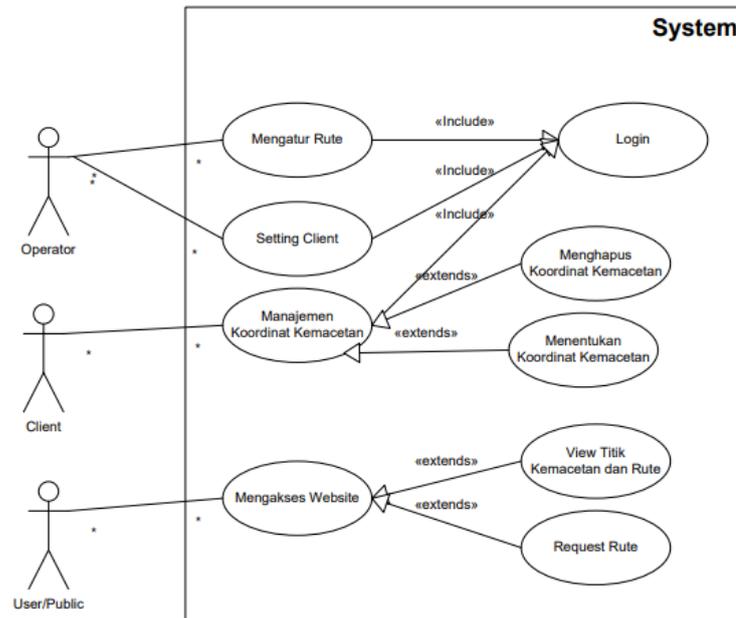
$$\begin{aligned}x &= \text{Latitude (Bujur)} \\y &= \text{Longitude (Lintang)} \\R &= \text{Radius Bumi} = 6371 \text{ km} \\1 \text{ derajat} &= 0.0174532925\end{aligned}$$

2.16 Unified Modeling Language (UML)

UML atau Unified Modeling Language adalah suatu pemodelan standar untuk mengembangkan sistem perangkat lunak berorientasi objek. UML sendiri juga menyediakan standar penulisan desain sistem yang mencakup konsep proses bisnis, penulisan kelas dalam bahasa pemrograman tertentu, skema basis data, dan komponen yang diperlukan dalam sistem perangkat lunak. Untuk merancang suatu aplikasi menggunakan Unified Modeling Language (UML), yang meliputi use case diagram untuk menggambarkan interaksi aktor dengan sistem, diagram aktivitas, dan sequence diagram untuk menggambarkan urutan proses sistem yang dimaksud. Dengan demikian, UML dapat membantu mengatasi kesulitan dalam merancang sistem multi-komponen dan memfasilitasi pembagian rancangan dengan tim [29]. Ada berbagai jenis diagram di UML, empat di antaranya adalah Use Case, *Activity*, Class, dan Sequence Diagram.

2.16.1 Use Case Diagram

Use Case Diagram merupakan jenis dari diagram UML digunakan untuk menggambarkan fungsionalitas sistem yang dibangun dari perspektif interaksi pengguna dengan sistem. Use case diagram dapat digunakan untuk menggambarkan tipe pengguna yang berbeda dan use case yang berbeda untuk sebuah sistem [29]. Dalam diagram use case terdapat dua jenis objek dalam diagram use case yaitu aktor dan naskah. Aktor adalah peran yang dapat dimainkan oleh pengguna saat berinteraksi dengan sistem. Sedangkan naskah adalah rangkaian peristiwa atau kejadian [30]. Berikut merupakan contoh penerapan yang akan menggambarkan tentang bagaimana seorang user sebagai yang digambarkan aktor terlibat dalam berbagai macam kejadian di dalam sebuah aplikasi mengenai kemacetan pada Gambar 2.12 Contoh Use Case Diagram:

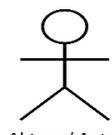


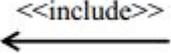
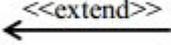
Sumber Gambar: <https://www.dicoding.com/blog/contoh-use-case-diagram/>

Gambar 2. 12 Contoh Use Case Diagram

Berikut merupakan penjelasan dari simbol Contoh Use Case Diagram yang ada pada contoh:

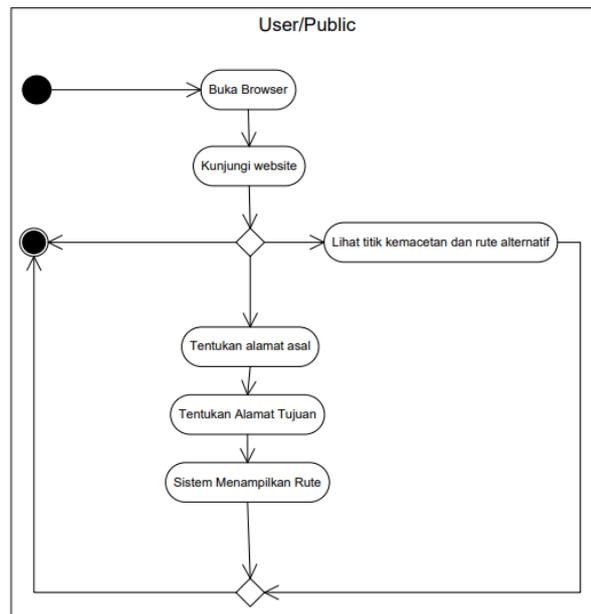
Tabel 2. 2 Arti Simbol Use Case

No.	Gambar Simbol	Nama Simbol	Keterangan
1.	 Akteur / Actor	<i>Actor</i>	Aktor adalah abstraksi dari orang dan sistem lain yang mengaktifkan fungsi sistem target. Orang atau sistem dapat memainkan banyak peran.
2.		<i>Use Case</i>	Use case adalah deskripsi tentang bagaimana sistem bekerja dari sudut pandang pengguna. Use case menggambarkan interaksi khas antara pengguna (user) sistem dan sistem itu sendiri dengan riwayat penggunaan sistem.
3.		<i>Association</i>	<i>Association</i> merupakan abstraksi dari penghubung antara <i>actor</i>

			dengan <i>use case</i> .
4.		<i>Generalisasi</i>	<i>Generalisasi</i> digunakan untuk menunjukkan spesialisasi aktor agar dapat berpartisipasi dengan <i>use case</i> .
5.		<i>Include</i>	Simbol tersebut menunjukkan bahwa <i>use case</i> dari keseluruhan merupakan fungsionalitas dari <i>use case</i> lainnya.
6.		<i>Extend</i>	Simbol tersebut menunjukkan bahwa <i>use case</i> merupakan tambahan dari <i>use case</i> yang lainnya dalam artian jika kondisi terpenuhi.
7.		<i>System Boundary</i>	Untuk menspesifikasikan paket yang menampilkan sistem secara terbatas.

2.16.2 Activity Diagram

Diagram aktivitas adalah representasi grafis dari alur kerja aktivitas dan aktivitas langkah demi langkah dengan dukungan pemilihan, pengulangan, dan konkurensi. Dalam bahasa pemodelan terpadu, diagram aktivitas dirancang untuk memodelkan proses komputasi dan organisasi (seperti alur kerja) dan arus informasi melalui aktivitas terkait. Meskipun tujuan utama dari diagram aktivitas adalah untuk menunjukkan aliran kontrol secara keseluruhan, hal itu juga dapat berisi elemen yang menunjukkan aliran data antar aktivitas melalui satu atau lebih penyimpanan suatu data [30]. Berikut merupakan contoh gambaran dari *Activity Diagram* pada Gambar 2.13 Contoh Activity Diagram:

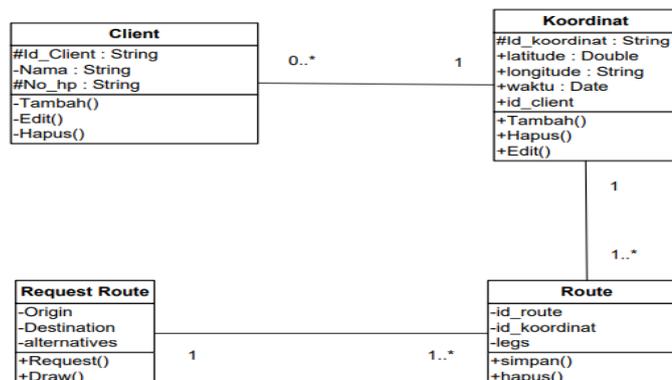


Sumber Gambar: <https://www.dicoding.com/blog/contoh-use-case-diagram/>

Gambar 2. 13 Contoh Activity Diagram

2.16.3 Class Diagram

Class Diagram merupakan diagram yang statis. Dalam hal ini mewakili tampilan aplikasi statis. Diagram kelas digunakan tidak hanya untuk memvisualisasikan, mendeskripsikan, dan mendokumentasikan berbagai aspek sistem, tetapi juga untuk membuat kode yang dapat dieksekusi untuk perangkat lunak [30]. Berikut merupakan contoh gambaran mengenai Class Diagram untuk pembangunan suatu aplikasi pada Gambar 2.14 Contoh Class Diagram:

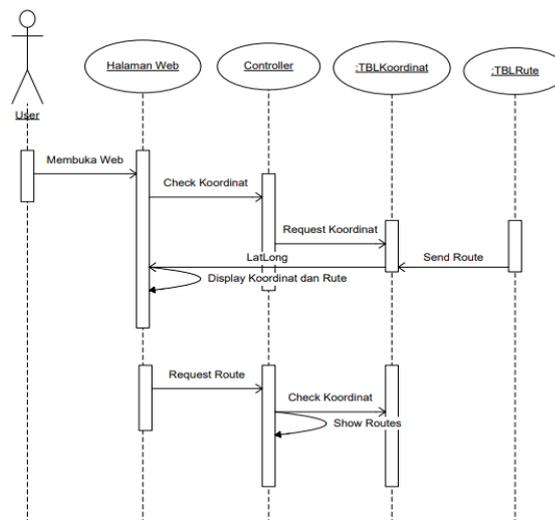


Sumber Gambar: <https://www.dicoding.com/blog/contoh-use-case-diagram/>

Gambar 2. 14 Contoh Activity Diagram

2.16.4 Sequence Diagram

Sequence Diagram digunakan untuk memodelkan interaksi skenario antara objek dalam sebuah sistem. *Sequence diagram* ini dapat membantu untuk memahami bagaimana suatu objek yang berhubungan digunakan dalam suatu sistem untuk mengidentifikasi suatu masalah dalam perancangan sistem sebelum melanjutkan ke tahap implementasi [29]. Diagram urutan dalam garis paralel mewakili beberapa proses atau objek yang aktif secara bersamaan, di antaranya pesan dipertukarkan sebagai panah horizontal sesuai urutan kemunculannya. Hal ini memungkinkan mendefinisikan skenario waktu proses yang sederhana secara grafis [30]. Berikut merupakan contoh penerapan *Sequence Diagram* dalam membangun sistem pada Gambar 2.15 Contoh Penerapan *Sequence Diagram*:



Sumber Gambar: <https://www.dicoding.com/blog/contoh-use-case-diagram/>

Gambar 2. 15 Contoh Penerapan *Sequence Diagram*

2.17 Perhitungan Kuesioner Skala Likert

Perhitungan kuesioner dilakukan menggunakan skala Likert. Skala Likert digunakan untuk menghitung dan menetapkan kategori dari kuesioner yang dibagikan kepada responden sehingga dapat diketahui persepsi pengguna terhadap aplikasi Tracking Kemacetan yang dibangun [31]. Untuk penentuan jumlah responden pada penelitian minimal berjumlah 30 responden [32]. Berikut

merupakan langkah-langkah atau cara dalam menghitung kuesioner menggunakan skala Likert.

- a. Dalam kuesioner yang diberikan kepada pengguna terdapat lima kategori nilai kategori yang dapat dipilih salah satunya dari setiap pertanyaan.

Tabel 2. 3 Penjelasan Kategori Penilaian

Nama Kategori	Skor
Sangat Setuju	5
Setuju	4
Ragu – ragu	3
Tidak Setuju	2
Sangat Tidak Setuju	1

- b. Menghitung jumlah maximum skor atau yang disebut dengan kriterium

$$= \text{Nilai Bobot Maksimal} \times \text{Jumlah Responden}$$

- c. Menghitung jumlah jawaban dari responden yang telah menjawab kuesioner

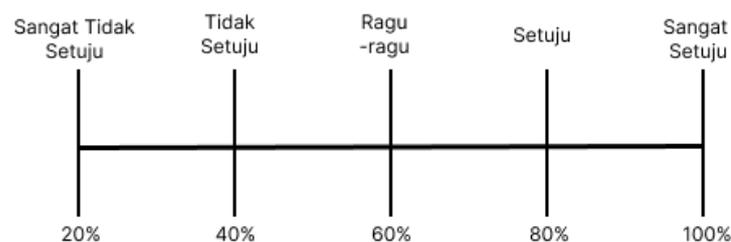
$$P = \frac{\text{Total nilai responden}}{\text{maximum skor}} \times 100\%$$

Keterangan:

P: nilai Persentase

Total nilai responden = bobot X frekuensi jawaban

- d. Setelah skor didapatkan selanjutnya akan disajikan kedalam grafik interval rating scale sebagai berikut:



Gambar 2. 16 Interval Rating Scale