

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Tinjauan pustaka menjelaskan beberapa teori-teori dan penjelasan yang berkaitan dengan aplikasi atau media yang akan dibangun. Tinjauan pustaka yang digunakan dalam penyusunan Aplikasi rekomendasi tracking pendaki gunung berdasarkan karakter pendaki berbasis android. Meliputi pengertian mendaki gunung, Android, GPS, API dll[9].

2.2 Pendaki Gunung

Pendaki adalah seseorang yang melakukan kegiatan mendaki gunung atau aktivitas pendakian gunung. Kegiatan pendakian gunung melibatkan berbagai faktor seperti kondisi cuaca, medan yang sulit, dan risiko kecelakaan yang dapat mempengaruhi keselamatan dan kesuksesan pendakian[10].

Banyak orang melakukan pendakian gunung sebagai bentuk kegiatan rekreasi dan petualangan. Di dalam pendakian ada klasifikasinya yaitu :

1. *Hill Climbing atau Hiking*

Hiking merupakan gerakan berjalan pada area atau bukit yang landai. Area landai dalam pendakian sebagai area yang nyaman untuk dilakukan kegiatan, karena tidak memerlukan teknik khusus dan tenaga yang besar dalam melaluinya.

2. *Scrambling*

Scrambling merupakan gerakan pendakian pada tebing yang tidak terlalu terjal. Teknik scrambling membutuhkan atau menjaga keseimbangan ketika berjalan agar tidak jatuh.

3. *Climbing*

Climbing merupakan sebuah aktivitas atau pergerakan pendakian pada tebing yang curam. Teknik climbing melakukan pergerakannya dengan cara merayap memanfaatkan celah , harus dilakukan untuk bisa melanjutkan perjalanan. Selain itu teknik climbing membutuhkan beberapa peralatan pendukung khusus seperti : *Harnest, Karnmantel, Helmet*, dll.

2.3 Gunung

Gunung adalah sebuah formasi geografis yang terdiri dari lereng dan puncak yang tinggi dan curam. Gunung terbentuk melalui proses geologis yang kompleks, yang melibatkan pergerakan lempeng tektonik, letusan vulkanik, dan erosi[11].

Gunung terbentuk melalui proses geologi yang kompleks, termasuk aktivasi tektonik, erosi, dan proses vulkanik. Ada beberapa jenis gunung, termasuk :

1. Gunung Berapi (Volcano): Gunung berapi terbentuk ketika material magma dari dalam bumi mencapai permukaan dan membentuk kubah atau kerucut di atas permukaan. Letusan gunung berapi dapat mengakibatkan pelepasan gas, abu, dan material vulkanik lainnya.
2. Gunung Plika (Fold Mountain): Gunung plika terbentuk ketika lempeng tektonik bertabrakan dan menghasilkan lipatan dan patahan dalam kerak bumi. Ini adalah jenis gunung yang sering ditemukan dalam rangkaian pegunungan.
3. Gunung Blok (Block Mountain): Gunung blok terbentuk ketika blok besar tanah diangkat atau turun akibat pergerakan tektonik, sering kali melalui patahan besar.
4. Gunung Patahan (Fault-block Mountain): Gunung patahan terbentuk ketika pergerakan vertikal dalam kerak bumi menghasilkan blok-blok terangkat dan daerah rendah antara patahan.

Gunung memiliki peran penting dalam ekologi dan lingkungan, serta budaya dan sejarah manusia. Mereka dapat menjadi habitat unik bagi berbagai jenis kehidupan dan memiliki dampak terhadap pola cuaca dan iklim di daerah sekitarnya. Gunung dapat ditemukan diberbagai jenis lingkungan, seperti di dataran tinggi, dataran rendah, pegunungan, atau di pulau-pulau yang terpisah. Beberapa gunung terkenal di indonesia antara lain Gunung Semeru di Jawa Timur, Gunung Gede di Jawa Barat, Gunung Rinjani di Nusa Tenggara Barat.

2.4 Rekomendasi

Rekomendasi adalah yang bertujuan untuk membantu pengguna dalam menemukan atau memilih produk, layanan, atau konten yang relevan atau sesuai dengan kebutuhan atau keinginan mereka.

Misalkan ketika seseorang sedang berada di sebuah toko buku yang memiliki

berbagai judul buku yang beragam. Orang tersebut memiliki minat dalam genre fiksi ilmiah dan ingin menemukan buku baru yang menarik untuk dibaca. Kemudian, seorang teman Anda datang dan merekomendasikan sebuah buku dengan sangat antusias, berdasarkan minat dan kesukaan Anda yang dia ketahui[12].

2.5 Tracking

Tracking adalah proses untuk mengumpulkan data posisi atau lokasi suatu objek pada suatu waktu tertentu, sehingga bisa dilacak atau ditelusuri pergerakan objek tersebut[13]. Dalam aplikasi yang berkaitan dengan tracking, objek tersebut biasanya adalah pengguna atau kendaraan yang dilengkapi dengan perangkat GPS atau teknologi serupa yang mampu memancarkan sinyal posisi[14].

2.6 Navigasi

Dalam konteks navigasi di gunung adalah proses menentukan dan mengikuti jalur atau rute di dalam lingkungan hutan yang kompleks. Navigasi dalam gunung melibatkan penggunaan terampilan dan alat tradisional, seperti kompas, peta topografi, pengetahuan tentang tanda-tanda alam, dan pengalaman visual. Keterampilan navigasi gunung melibatkan beberapa aspek:

1. Pemahaman Topografi: Ini melibatkan memahami peta topografi, yang memberikan informasi tentang bentuk permukaan bumi, ketinggian, kontur, sungai, dan fitur-fitur penting lainnya. Pemahaman tentang bagaimana membaca dan menginterpretasi peta topografi sangat penting.
2. Menggunakan Kompas: Kompas digunakan untuk menentukan arah utara dan membantu mengorientasikan diri dalam lingkungan. Ini membantu mengarahkan perjalanan sesuai dengan rute yang diinginkan.
3. Penggunaan GPS: GPS (Global Positioning System) adalah alat yang sering digunakan dalam navigasi gunung modern. Ini memberikan informasi akurat tentang lokasi Anda dalam bentuk koordinat geografis (latitude dan longitude).
4. Panduan Arah: Navigasi gunung melibatkan menavigasi di sekitar fitur-fitur alam, seperti bukit, lembah, dan sungai, serta menghindari rute yang terlalu berbahaya.
5. Perencanaan Rute: Sebelum memulai perjalanan, penting untuk merencanakan

rute dengan cermat berdasarkan peta dan informasi tentang area tersebut. Ini termasuk memperhitungkan waktu, jarak, elevasi, dan kondisi cuaca.

6. Pengetahuan Cuaca dan Lingkungan: Mengetahui perkiraan cuaca dan sifat lingkungan adalah bagian penting dari navigasi gunung. Kondisi cuaca dan lingkungan dapat berdampak signifikan pada perjalanan dan keamanan.

Navigasi gunung yang baik memungkinkan pendaki atau petualang untuk menjelajahi area pegunungan dengan aman, menghindari risiko tersesat, dan menemukan rute yang diinginkan menuju tujuan[15].

2.7 Aplikasi

Aplikasi merupakan program komputer yang dirancang dan dikembangkan untuk dijalankan diperangkat elektronik seperti smartphone, tablet, komputer, dll. Aplikasi umumnya memiliki antarmuka pengguna (UI) yang memungkinkan pengguna berinteraksi dengan program, melakukan tugas, mengakses informasi, atau mendapatkan hiburan. Aplikasi dapat berisi berbagai fitur, seperti analisis data, komunikasi, pengolahan gambar, permainan, dan banyak lagi, tergantung pada tujuan dan fungsi spesifik dari aplikasi tersebut[16]. Aplikasi dapat memiliki berbagai tujuan dan cakupan, yaitu:

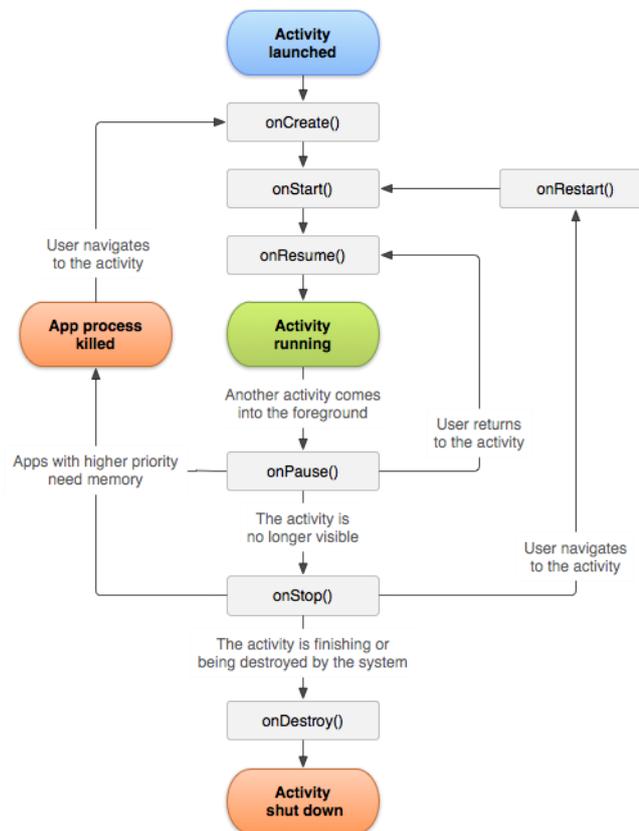
1. Aplikasi Seluler: Aplikasi yang dirancang untuk dijalankan di ponsel cerdas atau tablet. Ini termasuk aplikasi seperti permainan, jejaring sosial, aplikasi produktivitas, dan banyak lagi.
2. Aplikasi Desktop: Aplikasi yang diinstal dan dijalankan pada komputer desktop atau laptop. Ini bisa berupa perangkat lunak produktivitas seperti pengolah kata, spreadsheet, atau perangkat lunak kreatif seperti pengedit foto dan video.
3. Aplikasi Web: Aplikasi yang diakses melalui peramban web dan dijalankan di server. Pengguna dapat mengaksesnya dengan mengunjungi situs web tertentu. Contoh termasuk email web, aplikasi jejaring sosial, dan alat kerjasama online.
4. Aplikasi Server: Aplikasi yang berjalan di server dan menyediakan layanan atau data kepada aplikasi klien atau pengguna. Ini termasuk basis data, layanan cloud, dan banyak lagi.

5. Aplikasi Industri dan Khusus: Aplikasi yang dibangun untuk memenuhi kebutuhan industri atau bisnis tertentu, seperti aplikasi perbankan, sistem manajemen perusahaan, atau perangkat lunak medis.

2.8 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi[17]. Di dalam sistem operasi android terdapat siklus hidup dimana untuk menavigasi antara tahap activity life-cycle, android itu sendiri dengan menyediakan 6 method inti callback yaitu *onCreate()*, *onStart()*, *onResume()*, *onPause*, *onStop()*, dan *onDestroy()*[18].

Berikut adalah Gambar 2.4 Siklus hidup pada *platform* android :



Gambar 2.1 Siklus Hidup Android

Sumber Gambar :

<https://developer.android.com/guide/components/activities/activity->

lifecycle?hl=id

Method callback tersebut memiliki fungsinya masing-masing berikut penjelasannya :

1. onCreate merupakan *method callback* yang dipanggil ketika *activity* pertama kali dibuat.
2. onStart merupakan *method callback* yang dipanggil ketika *activity* mulai ditampilkan.
3. onResume merupakan *method callback* yang dipanggil ketika *activity* menjadi aktif dan siap menerima interaksi pengguna.
4. onPause merupakan *method callback* yang dipanggil ketika *activity* tidak lagi fokus dan mulai menjadi tidak aktif.
5. onStop merupakan *method callback* yang dipanggil ketika *activity* tidak lagi ditampilkan dan akan dikeluarkan dari layar.
6. onDestroy merupakan *method callback* yang dipanggil ketika *activity* dihapus dan tidak akan digunakan lagi

2.9 Web Service

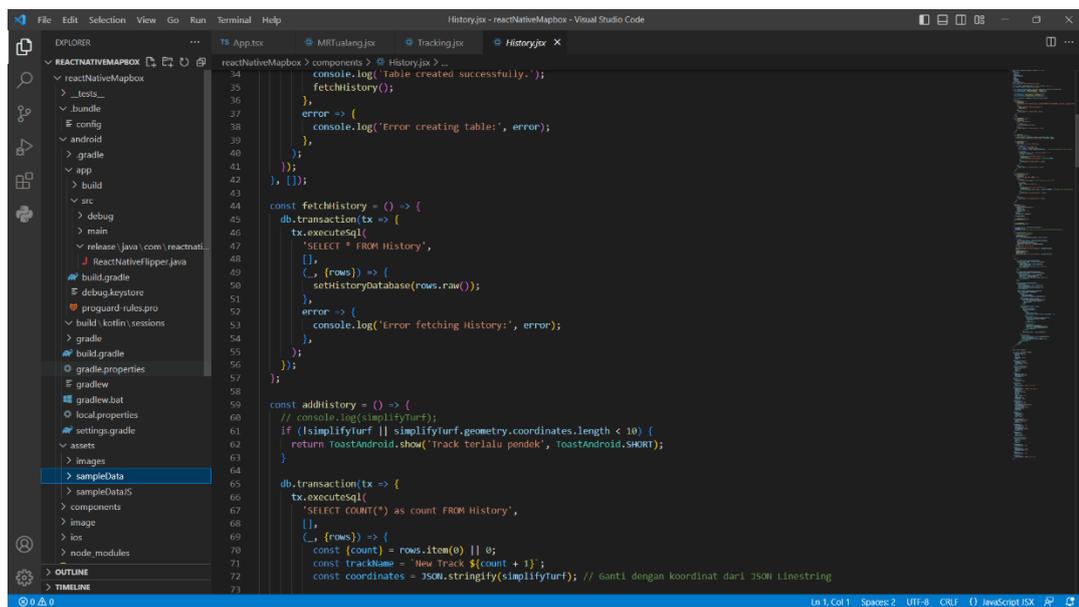
Web service merupakan aplikasi yang terdiri dari kumpulan basis data (*database*) dan perangkat lunak (*software*) atau bagian dari perangkat lunak yang dapat diakses dari jarak jauh yang diakses dari jarak jauh oleh perangkat dengan menggunakan perantara tertentu.

Web service memiliki kemampuan untuk melakukan pertukaran data tanpa memperhatikan sumber database, bahasa yang digunakan, atau *platform* dimana data tersebut diterima. Kemampuan ini memungkinkan web service berperan sebagai penghubung yang menghubungkan berbagai sistem.

Sebagai contoh, ketika pengguna ingin mengakses layanan yang tersedia pada sebuah website, pengguna akan memasukkan alamat URL dari website tersebut sebagai *request*. *Request* tersebut akan dikirim ke server yang dituju, dan server tersebut akan segera memberikan *response* yang kemudian ditampilkan sebagai halaman web[19]. Ini adalah pola hubungan yang umum antara klien dan server dalam sebuah website[20].

2.10 Visual Studio Code

Visual Studio Code adalah sebuah *text editor* yang berguna bagi pengembang perangkat lunak, termasuk penyorotan sintaks, penyelesaian otomatis kode, pemformatan kode, dll. Visual Studio Code merupakan sebuah *Cross Platform* yang dapat berguna di berbagai OS, seperti windows, mac OS, dan linux. Sehingga dengan multi device tersebut dapat memudahkan programmer untuk mengerjakan proyek. Tampilan halaman visual studio code dapat dilihat pada gambar 2.4.



Gambar 2.2 Tampilan Visual Studio Code

2.11 React Native

React Native adalah *framework* pengembang aplikasi mobile yang untuk membuat aplikasi mobile menggunakan Javascript. Dengan menggunakan React Native, pengembang dapat membuat aplikasi mobile yang dapat berjalan di platform Android dan iOS dengan menggunakan satu basis kode yang sama[21].

Adapun konsep yang dapat digunakan react native sebagai arsitektur dalam memisahkan logika bisnis aplikasi, tampilan pengguna, dan pengelolaan data. Berikut ini konsep arsitektur yang dapat digunakan di react native :

1. Model

Model merupakan tempat dimana data diambil, dimanipulasi, dan disimpan.

Berupa data yang diambil dari server, data yang disimpan di penyimpanan lokal, atau data yang dihasilkan dari interaksi pengguna.

2. View

View merupakan tempat dimana untuk menampilkan antarmuka pengguna (UI) dari aplikasi kepada pengguna. Berupa komponen yang digunakan untuk menampilkan elemen-elemen UI seperti tombol, teks, gambar, dan sebagainya.

3. Controller

Controller merupakan tempat dimana untuk menghubungkan model dan view sebagai perubahan state, menyampaikan data antara komponen. Controller mengendalikan aliran data antara Model dan View, serta bertanggung jawab untuk menangani perubahan state dan menyampaikan data antara komponen.

Dengan menggunakan konsep Model-View-Controller (MVC) memudahkan programmer dalam mengorganisir pengkodean.

2.12 JavaScript

Javascript adalah bahasa pemrograman yang digunakan untuk mengontrol perilaku elemen-elemen HTML, melakukan manipulasi DOM(Document Object Model), menangani interaksi pengguna, serta mengambil dan mengirim data antara halaman aplikasi dengan server. Selain itu, javascript juga merupakan bahasa pemrograman yang berbasis skrip, artinya kode JavaScript dapat dieksekusi pada saat halaman aplikasi dimuat atau berinteraksi dengan pengguna[22]. Berikut contoh program sederhana javasript yang dapat dilihat pada gambar 2.5.

```

<!DOCTYPE html>
<html>
<head>
  <title>Hello World Javascript</title>
</head>
<body>
  <script>
    console.log("Saya belajar Javascript");
    document.write("Hello World!");
  </script>
</body>
</html>

```

Gambar 2.3 Contoh Pengkodean Program Sederhana Javascript

2.13 JSON

JSON (Javascript Object Notation) adalah format data ringan yang digunakan untuk pertukaran data antar aplikasi. JSON berbasis teks dan terdiri dari urutan pasangan nama/kunci (key-value pairs). JSON menggunakan sintaks yang mirip dengan objek Javascript, sehingga mudah dibaca dan ditulis oleh manusia, serta mudah dipahami oleh mesin[23]. Struktur JSON yang dapat dilihat pada gambar 2.6.



Gambar 2.4 Struktur JSON

2.14 Mapbox API

Mapbox adalah sebuah platform pengembangan pemetaan(mapping) yang

menyediakan berbagai alat dan layanan untuk membuat dan menampilkan peta interaktif. Dibangun di atas teknologi pemetaan geospasial, Mapbox memungkinkan pengembang untuk mengintegrasikan peta interaktif ke dalam aplikasi.

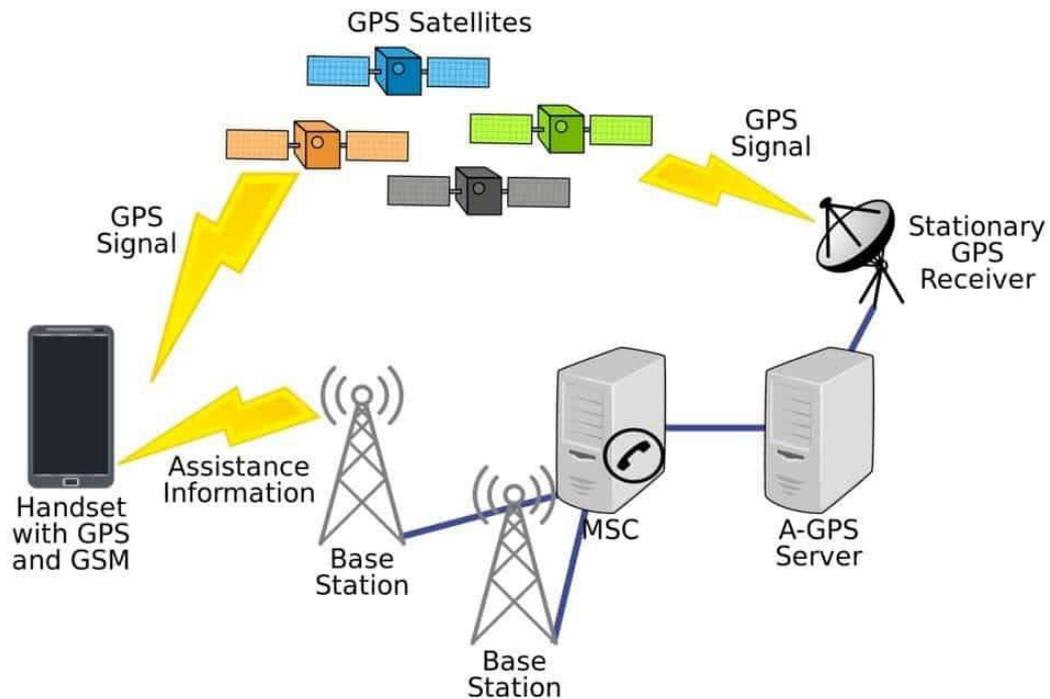
Mapbox API memungkinkan pengembang untuk mengintegrasikan peta interaktif, data geospasial, dan visualisasi lokasi ke dalam aplikasi mereka. Dengan menggunakan Mapbox API, pengembang dapat memanfaatkan infrastruktur pemetaan yang kuat untuk menciptakan pengalaman berbasis lokasi yang kaya dan responsif dalam aplikasi mereka[24][25]. Beberapa komponen utama dari Mapbox API meliputi:

1. Mapbox Maps API: Ini adalah API inti yang memungkinkan pengembang untuk membuat dan menyesuaikan peta interaktif yang dapat ditanamkan di dalam aplikasi atau situs web mereka. Anda dapat mengatur tampilan peta, lapisan data, simbol, dan berbagai elemen visual lainnya.
2. Mapbox GL JS: Ini adalah pustaka JavaScript yang memungkinkan pengembang untuk membuat peta interaktif dengan menggunakan WebGL (Web Graphics Library) untuk performa grafis yang tinggi di peramban web.
3. Mapbox Studio API: Ini memungkinkan pengembang untuk mengatur dan mengelola gaya peta kustom di Mapbox Studio (alat desain peta Mapbox) melalui kode.
4. Mapbox Geocoding API: API ini memungkinkan geocode (mengubah alamat menjadi koordinat geografis) dan dekode (mengubah koordinat menjadi alamat) dalam aplikasi.
5. Mapbox Directions API: API ini menyediakan arahan rute dan perhitungan waktu tempuh berdasarkan data lalu lintas dan pilihan lainnya.
6. Mapbox Navigation SDK: Ini adalah perangkat pengembangan perangkat lunak (SDK) yang memungkinkan pembangunan navigasi berbasis lokasi yang kompleks, seperti dalam aplikasi navigasi mobil.

2.15 GPS

GPS adalah singkatan dari *Global Positioning System*, yang merupakan

sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit[26]. GPS dapat menerima informasi, seperti : Waktu, Lokasi, Kecepatan, Arah Perjalanan, Simpan Lokasi, dan Komulasi Data[27]. Prinsip kerja dalam gps dapat dilihat pada gambar 2.7.



Gambar 2.5 Prinsip Kerja GPS

Sumber : <https://carisinyal.com/wp-content/uploads/2019/05/ee.jpg>

2.16 OpenWeather API

OpenWeather merupakan sebuah layanan atau API yang menyediakan informasi terkait cuaca dan prakiraan cuaca. Informasi cuaca yang disediakan oleh *OpenWeather* meliputi suhu, kelembaban, kecepatan angin, tekanan udara, dan prakiraan cuaca harian untuk beberapa hari ke depan[28]. Melalui OpenWeather API, pengembang dapat mengakses berbagai jenis data cuaca, termasuk:

1. Data Cuaca Aktual: Termasuk informasi seperti suhu saat ini, tekanan atmosfer, kelembaban, kecepatan angin, arah angin, kondisi cuaca (cerah, berawan, hujan, dll.), dan banyak lagi.
2. Prakiraan Cuaca: Data prakiraan cuaca yang mencakup perkiraan suhu,

kondisi cuaca, kelembaban, dan informasi lainnya untuk beberapa periode waktu ke depan, seperti per jam atau per hari.

3. Data Matahari dan Bulan: Informasi tentang waktu matahari terbit, matahari terbenam, fajar, senja, fase bulan, dan lainnya.
4. Data Cuaca Historis: Beberapa versi API OpenWeather juga menyediakan akses ke data cuaca historis, yang memungkinkan pengguna untuk mengambil data cuaca dari masa lalu.
5. Informasi Lokasi: API juga dapat memberikan informasi tentang koordinat geografis, zona waktu, dan nama kota atau lokasi yang berkaitan dengan data cuaca.

Ilustrasi kerja OpenWeather API dapat dilihat pada gambar 2.8.



Gambar 2.6 Ilustrasi Kerja OpenWeather API

Sumber :<https://www.niagahoster.co.id/blog/wpcontent/uploads/2020/11/Ilustrasi-Mengenal-API-Pengertian-Fungsi-dan-Cara-Kerjanya-1024x345.jpg>

2.17 Socket.io

Socket.io adalah sebuah pustaka (library) JavaScript yang memungkinkan komunikasi waktu nyata (real-time) antara klien (client) dan server melalui protokol WebSocket. Socket.io dirancang untuk mengatasi batasan komunikasi HTTP tradisional yang bersifat satu arah, di mana klien harus secara teratur meminta informasi dari server. Dengan Socket.io, server dapat mengirimkan data kepada klien secara aktif begitu data tersebut tersedia, tanpa perlu adanya permintaan khusus dari klien[29].

Selain itu, Socket.io dapat digunakan dalam berbagai jenis aplikasi web, termasuk aplikasi jejaring sosial, permainan multi-pemain, alat kerjasama dalam waktu nyata, dan banyak lagi. Untuk menggunakannya, Anda dapat menginstal pustaka Socket.io di sisi klien (biasanya dengan menggunakan npm atau Yarn) dan

mengintegrasikannya dengan server Anda untuk menciptakan komunikasi waktu nyata yang efisien dan andal antara klien dan server.

2.18 NodeJs

Node.js adalah lingkungan runtime yang dibangun di atas mesin JavaScript V8 yang dikembangkan oleh Google. Node.js memungkinkan Anda menjalankan kode JavaScript di sisi server, bukan hanya di sisi klien (browser). Ini berarti Anda dapat membuat aplikasi server-side yang dapat menangani permintaan dari klien, melakukan operasi jaringan, mengakses basis data, dan melakukan tugas lainnya[31].

Node.js awalnya dibangun untuk mengatasi masalah skalabilitas dalam pengembangan aplikasi berbasis real-time. Namun, dengan cepat populer dalam pengembangan berbagai jenis aplikasi, termasuk aplikasi web, aplikasi jaringan, alat baris perintah, aplikasi desktop, dan banyak lagi.

2.19 SQLite

SQLite adalah sebuah sistem pengelola basis data relasional (RDBMS) yang sederhana dan ringan, dirancang untuk menyimpan, mengelola, dan mengakses data dalam sebuah basis data. Yang membedakan SQLite dari sistem pengelola basis data lainnya adalah bahwa itu adalah basis data serverless, yang berarti tidak ada server terpisah yang perlu dijalankan atau dikonfigurasi. Basis data SQLite disimpan dalam sebuah berkas tunggal yang dapat dengan mudah diintegrasikan ke dalam aplikasi. SQLite sering digunakan dalam aplikasi seluler, aplikasi desktop, aplikasi berbasis web, dan proyek-proyek yang memerlukan penyimpanan data lokal[32]. Namun, karena sifatnya yang ringan dan sederhana, SQLite mungkin kurang cocok untuk proyek-proyek besar dengan permintaan basis data yang tinggi atau kebutuhan skalabilitas yang rumit.

2.20 Rumus Euclidean

Rumus Euclidean adalah rumus matematika yang digunakan untuk menghitung jarak antara dua titik dalam ruang Euclidean dengan dimensi tertentu. Ruang Euclidean adalah ruang geometri yang umum digunakan dalam matematika, di mana jarak antara dua titik diukur dalam garis lurus, dan memiliki sifat-sifat yang konsisten dengan intuisi kita tentang ruang fisik[33].

Rumus Jarak Euclidean antara dua titik A dan B dalam ruang Euclidean 2D (dua dimensi) adalah:

$$\mathbf{Jarak} = \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)}$$

Untuk ruang Euclidean 3D (tiga dimensi), rumus jarak Euclidean menjadi :

$$\mathbf{Jarak} = \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2)}$$

Rumus Euclidean memiliki banyak aplikasi dalam berbagai bidang, seperti geometri, analisis data, pembelajaran mesin, komputer grafis, pemrosesan gambar, dan banyak lagi. Ini adalah cara umum untuk mengukur jarak antara titik-titik dalam ruang numerik, dan sering digunakan dalam solusi matematis dan algoritmik.

2.21 UML

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun[14].

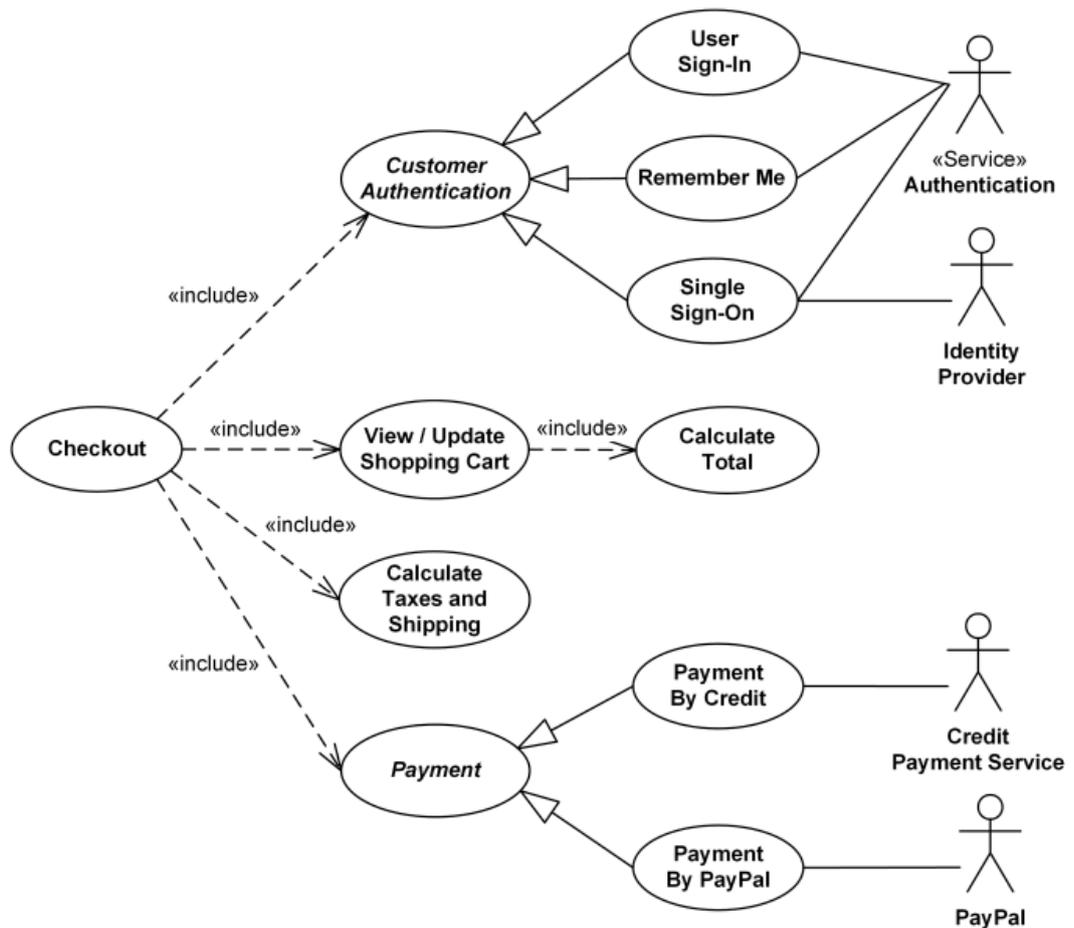
UML membantu dalam merencanakan, merancang, dan mengelola proyek pengembangan perangkat lunak dengan lebih efisien dan efektif. Ini telah menjadi alat yang sangat penting dalam rekayasa perangkat lunak modern dan digunakan di seluruh industri untuk menggambarkan dan berkomunikasi tentang desain sistem kompleks[34]. Beberapa jenis diagram UML yang umum digunakan termasuk:

1. Diagram Kelas (Class Diagram): Digunakan untuk menggambarkan struktur kelas, hubungan antara kelas, atribut, dan metode dalam perangkat lunak.
2. Diagram Use Case: Menunjukkan interaksi antara aktor (pengguna atau sistem lain) dan sistem, menggambarkan fungsionalitas yang diinginkan dari sistem.
3. Diagram Aktivitas (Activity Diagram): Menggambarkan aliran kerja atau urutan aktivitas dalam sistem, sering digunakan untuk menggambarkan proses bisnis.
4. Diagram Sekuens (Sequence Diagram): Menunjukkan interaksi antara objek-objek dalam sistem dalam urutan waktu, membantu memvisualisasikan komunikasi antar objek.

5. Diagram Komponen (Component Diagram): Menggambarkan struktur fisik dan komponen perangkat lunak serta ketergantungannya.
6. Diagram Penyekrupan (Deployment Diagram): Menggambarkan bagaimana komponen perangkat lunak diimplementasikan dan didistribusikan di lingkungan perangkat keras.
7. Diagram Kenyataan (State Diagram): Digunakan untuk menggambarkan aliran keadaan yang berbeda yang dimiliki oleh objek dalam sistem sepanjang waktu.

2.21.1 Use Case

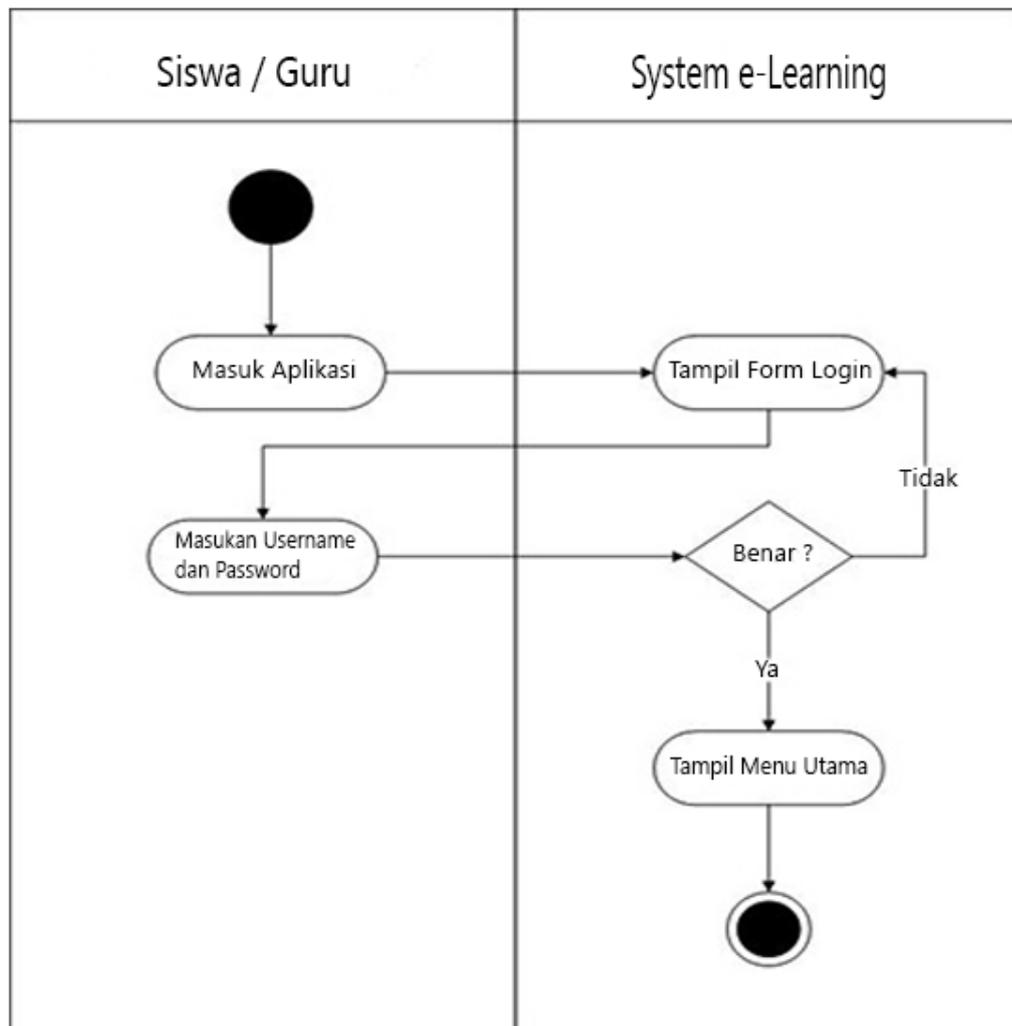
Use case adalah teknik dalam analisis kebutuhan perangkat lunak dan rekayasa perangkat lunak yang digunakan untuk memahami dan menggambarkan interaksi antara pengguna atau aktor dengan sistem yang akan dibangun[35]. Contoh penggunaan use case dapat dilihat pada gambar 2.10.



Gambar 2.7 penggunaan use case

2.21.2 Activity Diagram

Activity diagram ialah yang digunakan untuk memodelkan alur kerja atau aktivitas dalam suatu proses. Activity diagram menggambarkan aktivitas atau tindakan yang dilakukan dalam proses, serta hubungan antara aktivitas tersebut[36]. Contoh penggunaan activity diagram dapat dilihat pada gambar 2.11.

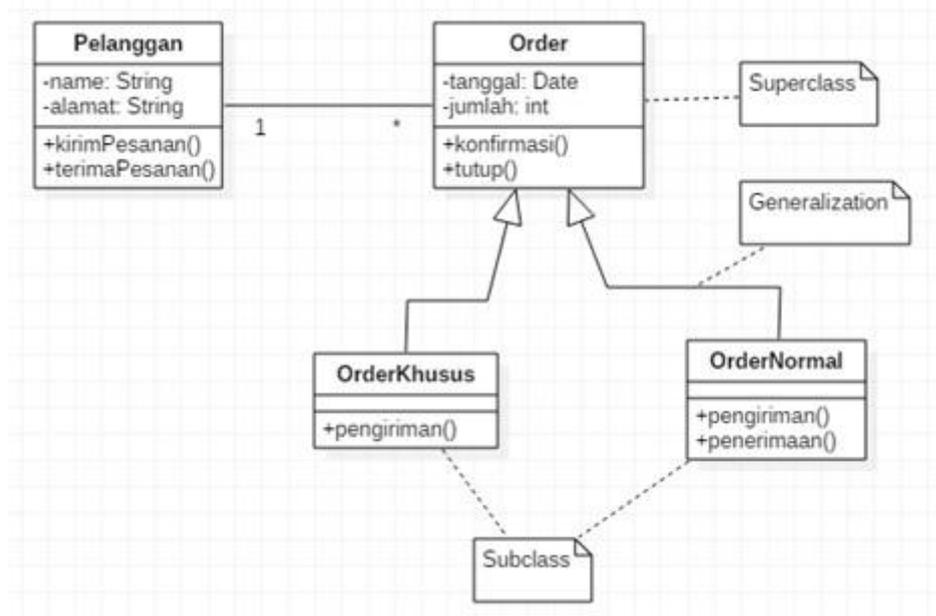


Gambar 2.8 Penggunaan Activity Diagram

2.21.3 Class Diagram

Class diagram ialah yang digunakan untuk memodelkan struktur objek atau kelas dalam suatu sistem. Diagram ini menggambarkan hubungan antara kelas-

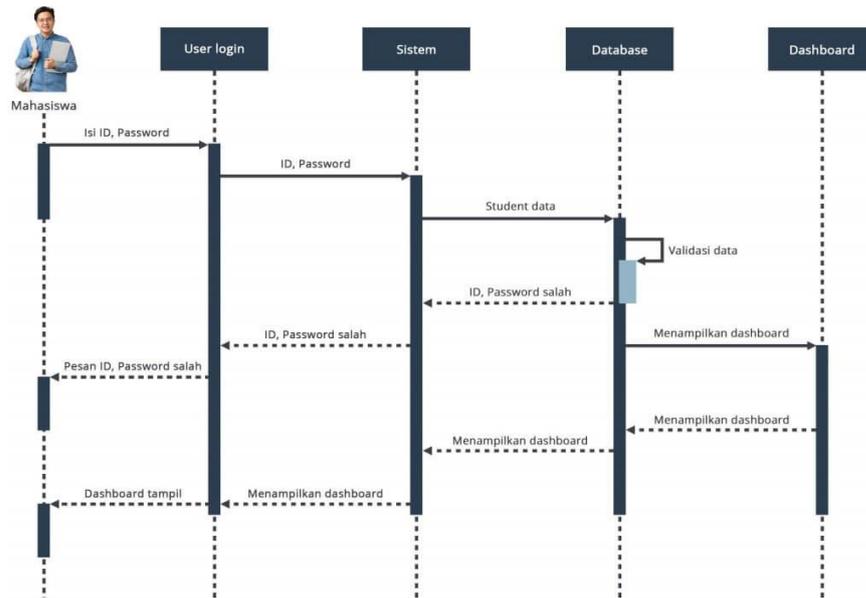
kelas, atribut, dan metode dalam sistem. Contoh penggunaan class diagram dapat dilihat pada gambar 2.12.



Gambar 2.9 Penggunaan Class Diagram

2.21.4 Sequence Diagram

Sequence diagram ialah yang digunakan untuk memodelkan interaksi antara objek dalam suatu sistem atau proses. Diagram ini menggambarkan urutan tindakan yang dilakukan oleh objek dalam proses, serta hubungan antara objek tersebut. Contoh penggunaan sequence diagram dapat dilihat pada gambar 2.13.



Gambar 2.10 Penggunaan Sequence Diagram