

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam era digitalisasi pendidikan, sekolah dituntut untuk mengelola proses administrasi secara cepat, akurat, dan efisien. Salah satu aktivitas administratif yang memiliki tingkat kompleksitas tinggi adalah pengelolaan data kehadiran siswa. Aktivitas ini dilakukan setiap hari, melibatkan ratusan siswa dan puluhan kelas, sehingga menghasilkan volume transaksi data yang besar dan berulang. Pengelolaan manual melalui panggilan nama dan tanda tangan tidak hanya menyita waktu guru di awal pembelajaran, tetapi juga menambah beban administrasi dalam proses rekapitulasi harian hingga bulanan. Kondisi tersebut menjadikan proses absensi sebagai tugas rutin yang rumit dan tidak efisien, terutama ketika data harus diintegrasikan dengan laporan kehadiran secara digital.

SMA Negeri 5 Medan, sebagai institusi pendidikan dengan jumlah siswa yang besar, menghadapi tantangan serupa. Pengolahan data kehadiran yang masif memerlukan sistem yang mampu mengotomatisasi pencatatan dan validasi data secara cepat, akurat, dan hemat sumber daya. Penerapan teknologi mobile menjadi solusi potensial, karena perangkat seluler kini telah menjadi sarana umum bagi siswa dan guru. Namun, sistem absensi daring konvensional masih menyisakan permasalahan validasi lokasi siswa dapat melakukan absensi dari luar area sekolah selama perangkat terhubung ke internet.

Untuk mengatasi hal tersebut, teknologi Geofencing dapat diterapkan sebagai mekanisme verifikasi lokasi berbasis koordinat GPS. Dengan membangun pagar virtual di sekitar area sekolah, sistem hanya akan menerima absensi dari

perangkat yang benar-benar berada di dalam zona yang telah ditentukan. Penerapan algoritma Axis-Aligned Bounding Box (AABB) digunakan untuk memastikan validasi posisi secara akurat dalam ruang tiga dimensi (3D), mencakup batas koordinat horizontal dan vertikal.

Oleh karena itu, penelitian ini dilakukan untuk mengimplementasikan algoritma geofencing pada sistem daftar hadir siswa berbasis mobile di SMA Negeri 5 Medan. Tujuannya adalah menciptakan sistem absensi yang efisien, cerdas, dan sesuai dengan kebutuhan operasional sekolah dengan meminimalkan beban kerja administratif guru, mengurangi kesalahan manusia, serta meningkatkan keakuratan dan efektivitas proses pencatatan kehadiran.

## **1.2 Perumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, maka perumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang sistem daftar hadir siswa berbasis mobile yang mampu mengotomatisasi proses absensi harian dengan jumlah transaksi data yang tinggi secara efisien?
2. Bagaimana penerapan algoritma geofencing dengan metode AABB (Axis-Aligned Bounding Box) dapat memastikan validasi lokasi siswa secara akurat di area SMA Negeri 5 Medan?
3. Bagaimana sistem yang dikembangkan dapat meningkatkan efisiensi waktu, akurasi data, dan efektivitas kerja guru dalam proses administrasi kehadiran siswa?

### 1.3 Batasan Masalah

Adapun Batasan masalah yang digunakan dalam penelitian ini diantaranya :

1. Aplikasi yang dibangun merupakan aplikasi *mobile*.
2. Spesifikasi minimum perangkat adalah Android API 28+ (Android 9 keatas)
3. Area pagar virtual (*Geofencing*) ditetapkan sebagai volume 3D yang dibatasi oleh enam nilai: empat batas koordinat horizontal (Utara, Selatan, Timur, Barat) dan dua batas vertikal (ketinggian minimum dan maksimum).
4. Logika validasi utama menggunakan Algoritma *AABB* (*Axis-Aligned Bounding Box*), yang memverifikasi lokasi siswa pada sumbu 2D (*latitude*, *longitude*) dan juga terhadap rentang vertikal (*altitude*) yang telah ditetapkan sebagai volume 3D area sekolah.
5. Penelitian ini tidak membahas aspek administrasi pengkajian atau penilaian akademik, melainkan fokus pada pencatatan dan rekapitulasi data kehadiran.
6. Pengguna sistem diklasifikasikan menjadi tiga peran utama dengan hak akses yang berbeda: Siswa (melakukan absensi *online*), Guru (mengelola sesi kelas, melakukan absensi manual, dan membuat laporan kehadiran), dan Admin (mengelola keseluruhan data *master* sistem, dan pengaturan *Geofencing*).
7. Sistem ini memerlukan konektivitas internet yang stabil untuk memastikan komunikasi antara aplikasi siswa, aplikasi guru, dan server hosting eksternal yang menjadi pusat data sistem.
8. Kinerja sistem secara keseluruhan bergantung pada kestabilan koneksi internet (Wifi SMAN 5 Medan) dan layanan server hosting eksternal yang

digunakan. Penelitian ini tidak mencakup skenario gangguan pada layanan hosting atau jaringan internet (*force majeure*), seperti gangguan teknis dari penyedia layanan. Apabila terjadi gangguan tersebut, sistem tidak dapat diakses, dan proses absensi akan kembali dilakukan secara manual oleh guru.

9. Sistem belum diuji terhadap penggunaan data lokasi palsu (fake GPS atau GPS spoofing). Penelitian ini berfokus pada implementasi dan validasi algoritma *geofencing* menggunakan data lokasi asli yang dihasilkan oleh perangkat pengguna. Oleh karena itu, aspek keamanan terkait manipulasi lokasi oleh pengguna tidak dibahas dan tidak menjadi fokus penelitian ini.

#### **1.4 Tujuan Penelitian**

Tujuan yang ingin dicapai dalam penelitian Tugas Skripsi ini adalah:

1. Membangun purwarupa (prototype) aplikasi daftar hadir siswa berbasis mobile yang mampu menangani transaksi data absensi dalam jumlah besar secara efisien.
2. Menerapkan algoritma Geofencing berbasis metode AABB (Axis-Aligned Bounding Box) sebagai mekanisme utama verifikasi lokasi kehadiran.
3. Melakukan pengujian fungsional (*black box testing*) dan efektivitas sistem dalam meningkatkan efisiensi dan akurasi proses absensi di SMA Negeri 5 Medan.

## **1.5 Manfaat Penelitian**

Adapun manfaat penelitian ini adalah sebagai berikut:

1. Menyediakan solusi sistem absensi siswa berbasis mobile yang efisien, modern, dan akurat, serta menjadi alat bantu bagi sekolah dalam menegakkan kedisiplinan dan meningkatkan ketertiban kehadiran siswa.
2. Menjadi sarana penerapan pengetahuan teoretis yang telah diperoleh selama perkuliahan ke dalam pengembangan sistem nyata, serta menambah pengalaman praktis dalam pengembangan perangkat lunak berbasis mobile.
3. Dapat menjadi referensi dan dasar pengembangan penelitian selanjutnya, khususnya yang berkaitan dengan penerapan teknologi berbasis lokasi (*location-based services*) pada sektor pendidikan.

## **1.6 Sistematika Penulisan**

Sistematika penulisan Tugas Skripsi ini dibagi atas beberapa bab, dimana masing-masing bab dibagi atas beberapa sub agar mempermudah penjelasan mengenai penelitian yang dilakukan dan mempermudah pembaca dalam memahami isi penelitian. Adapun sistematika penulisan Tugas Skripsi ini adalah sebagai berikut:

### **BAB 1: PENDAHULUAN**

Pada bab ini berisikan tentang pembahasan umum yang berkaitan dengan penelitian, meliputi Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, dan Sistematika Penulisan.

## **BAB 2: TINJAUAN PUSTAKA**

Bab ini berisi tinjauan pustaka dari penelitian terkait sebelumnya serta penjelasan teori-teori dasar yang mendukung perancangan dan implementasi sistem. Teori tersebut mencakup konsep pengembangan *mobile cross-platform* menggunakan *framework* Flutter, pengembangan *backend* menggunakan *framework* PHP Laravel, arsitektur API *RESTful*, teknologi basis data *real-time* Firebase, layanan berbasis lokasi (*Geolocation*), teknologi Mapbox API untuk visualisasi 3D, konsep pagar virtual 3D, serta algoritma inti *AABB* (*Axis-Aligned Bounding Box*) untuk validasi lokasi berbasis volume.

## **BAB 3: METODE PENELITIAN**

Bab ini menjelaskan langkah-langkah sistematis yang telah dilakukan dalam penelitian. Bagian ini mencakup metode pengembangan perangkat lunak yang digunakan, teknik pengumpulan data, serta perancangan arsitektur sistem secara rinci, mulai dari perancangan alur (UML), arsitektur *frontend* Flutter, arsitektur *backend* Laravel, hingga integrasi dengan Mapbox dan Firebase. Bab ini juga ditutup dengan penjelasan teknik pengujian sistem yang meliputi pengujian *black box* dan pengujian akurasi lokasi.

## **BAB 4: HASIL DAN PEMBAHASAN**

Bab ini menyajikan hasil dari implementasi dan pengujian sistem. Bagian ini juga berisi pembahasan mendalam mengenai temuan

penelitian, analisis hasil pengujian fungsionalitas (*black box*) dan pengujian akurasi lokasi, serta antarmuka (*interface*) sistem yang telah dibangun.

## **BAB 5: KESIMPULAN DAN SARAN**

Bab terakhir ini berisi kesimpulan dari keseluruhan hasil penelitian yang menjawab rumusan masalah, serta saran-saran yang dapat diberikan untuk pengembangan sistem atau penelitian sejenis di masa mendatang.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

Penelitian yang terkait dengan studi ini digunakan sebagai acuan, perbandingan, dan untuk memposisikan penelitian yang dilakukan penulis. Berikut adalah tinjauan terhadap beberapa penelitian terdahulu yang berhubungan dengan topik sistem absensi berbasis lokasi.

Pertama, Penelitian oleh Fauzan dkk. (2025) Penelitian berjudul *"Implementasi Algoritma Haversine pada Absensi Perkuliahan Berbasis Online di Fakultas Teknik Universitas Islam Sumatera Utara"* ini bertujuan mengatasi kendala absensi konvensional seperti "titip absen" dan sulitnya verifikasi lokasi mahasiswa. Sistem ini dibangun menggunakan *framework* Flutter dan Laravel, yang relevan dengan tumpukan teknologi (*tech stack*) penelitian ini.

- **Persamaan:** Menggunakan tumpukan teknologi Flutter dan Laravel untuk sistem absensi di lingkungan pendidikan.
- **Perbedaan (Celah Penelitian):** Metode validasi lokasi yang digunakan adalah Algoritma Haversine. Algoritma ini menciptakan pagar virtual 2D berbentuk lingkaran (radius), dengan batas 300 meter dari satu titik pusat. Metode ini kurang presisi untuk area sekolah yang batasnya berbentuk kotak, dan sama sekali tidak memvalidasi ketinggian (*altitude*) siswa.

Kedua, Penelitian oleh Bagus Sudirman dkk. (2025) Penelitian berjudul *"Implementasi Geofencing pada Sistem Presensi Siswa dengan Verifikasi Lokasi Secara Real-Time: Studi Kasus SMK Islamic Center Baiturrahman"*. Penelitian ini

mengatasi masalah sistem presensi konvensional di lingkungan SMK, Solusi yang ditawarkan adalah sistem presensi siswa berbasis teknologi Geofencing yang terintegrasi dengan wali murid untuk meningkatkan transparansi dan akurasi. Metode pengembangan yang digunakan adalah *Research and Development (R&D)*. Sistem ini memastikan siswa hanya bisa melakukan absensi saat berada di dalam area yang telah ditentukan oleh sekolah. Hasilnya menunjukkan bahwa penerapan Geofencing membawa manfaat signifikan dalam efektivitas mekanisme presensi dan efisiensi administratif sekolah.

- Persamaan: Berfokus pada implementasi *Geofencing* untuk absensi siswa di tingkat sekolah (SMK/SMA).
- Perbedaan (Celah Penelitian): Penelitian ini berfokus pada penerapan *Geofencing* secara umum (pagar datar 2D) dan integrasinya dengan wali murid. Penelitian ini tidak membahas implementasi pagar virtual 3D (volume) yang dapat memvalidasi ketinggian, sehingga masih memiliki celah manipulasi secara vertikal (dari gedung tinggi di sebelah sekolah).

Ketiga, Penelitian oleh Semmy Wellem Taju dkk. (2024) dalam penelitian "*Implementing QR code and Geolocation Technologies for the Student Attendance System*". Penelitian ini mengatasi inefisiensi berbagai metode absensi manual di Universitas Klabat, seperti formulir kertas dan panggilan nama. Sistem yang diusulkan terdiri dari aplikasi web untuk dosen dan aplikasi *mobile* untuk mahasiswa, yang dikembangkan menggunakan *Prototyping Model*. Teknologi yang digunakan adalah QR Code untuk verifikasi dan Geolocation untuk memastikan mahasiswa berada di lokasi perkuliahan.

- Persamaan: Menggunakan aplikasi *mobile* dan validasi lokasi GPS untuk absensi.
- Perbedaan (Celah Penelitian): Penelitian ini hanya menggunakan *Geolocation* (mengambil titik GPS), bukan *Geofencing* (memvalidasi titik GPS di dalam sebuah batas). Sistem ini hanya mencatat "dimana" siswa berada, tetapi tidak secara otomatis menolak jika siswa berada "di luar" pagar yang ditentukan. Ini adalah tingkat validasi yang lebih lemah dibandingkan *Geofencing*.

Keempat, Penelitian oleh Miss. Harshada Sudam Gite dkk. (2023) dalam jurnal "GPS BASED ATTENDANCE SYSTEM USING GEOFENCING". Penelitian ini mengusulkan sistem absensi berbasis GPS menggunakan teknik geofencing. Sistem ini menciptakan batas-batas virtual di sekitar area yang telah ditentukan. Ketika siswa masuk atau keluar dari area tersebut, lokasi mereka dicatat secara *real-time* menggunakan GPS. Sistem ini terdiri dari perangkat GPS yang dipegang siswa yang mengirimkan data ke *server* pusat untuk dibandingkan dengan batas geofence.

- Persamaan: Menggunakan teknologi Geofencing dan GPS pada platform aplikasi (Android) untuk melacak kehadiran siswa.
- Perbedaan (Celah Penelitian): Fokusnya adalah pada penerapan batas virtual 2D dan tidak membahas validasi ketinggian (altitude). Penelitian ini juga menyertakan penggunaan Bluetooth Low Energy (BLE), yang menunjukkan fokus tambahan pada validasi jarak dekat atau dalam ruangan, bukan volume 3D.

Kelima, Penelitian oleh Muhammad Fadhlurrohman (2019) berjudul "Aplikasi Presensi Siswa Menggunakan Metode Geofencing". Penelitian ini

mengusulkan sebuah sistem absensi otomatis untuk siswa menggunakan teknologi geofencing pada perangkat *mobile* Android. Tujuannya adalah membangun sistem yang dapat memberikan informasi keberadaan siswa kepada orang tua dan guru. Metode pengembangan yang digunakan adalah Waterfall dan memanfaatkan Google Geofencing API , Google Maps API , serta Firebase Cloud Messaging.

- Persamaan: Mengimplementasikan Geofencing untuk absensi siswa di lingkungan sekolah pada platform Android.
- Perbedaan (Celah Penelitian): Penelitian ini menggunakan Google Geofencing API standar yang merupakan implementasi 2D (pagar datar). Tidak ada pembahasan mengenai validasi sumbu vertikal (altitude) atau penerapan volume 3D.

Keenam, Penelitian oleh Dedus Lihoko (2024) berjudul "Pembangunan Aplikasi Monitoring Kehadiran Siswa Berbasis Android Di SMA PLUS AL-WAHID". Penelitian ini bertujuan membangun aplikasi monitoring kehadiran untuk menggantikan sistem manual. Solusi ini menggunakan teknologi *Location-Based Service* (LBS), Geofencing, dan WhatsApp API untuk notifikasi *real-time* kepada wali murid. Metode pengembangan yang digunakan adalah SDLC Waterfall.

- Persamaan: Menggunakan Geofencing pada platform Android untuk validasi absensi di tingkat SMA.
- Perbedaan (Celah Penelitian): Implementasi Geofence dalam penelitian ini secara eksplisit dijelaskan sebagai batas 2D yang digambar sebagai poligon (menggunakan `latLngList`). Validasi dilakukan dengan memeriksa apakah siswa

berada "di dalam poligon" (*isInsidePolygon*), sehingga tidak ada validasi ketinggian (*altitude*).

Ketujuh, Penelitian oleh Babatunde, A.N. dkk. (2022) dalam jurnal "Mobile Based Student Attendance System Using Geo-Fencing With Timing and Face Recognition". Penelitian ini menggabungkan Geofencing dengan *Face Recognition* (Pengenalan Wajah) untuk validasi. Sebuah keunikan dari sistem ini adalah validasi *timing*, di mana absensi hanya dianggap sah jika siswa menghabiskan lebih dari 90% waktu di dalam area *geo-fence* yang telah diatur.

- Persamaan: Menggunakan Geofencing pada aplikasi *mobile* untuk absensi siswa.
- Perbedaan (Celah Penelitian): Penelitian ini menambahkan Pengenalan Wajah dan durasi waktu sebagai lapisan keamanan utamanya. Namun, *geo-fence* itu sendiri (yang menggunakan Algoritma Winding Number) masih merupakan validasi batas virtual 2D dan tidak mengisi celah validasi 3D (ketinggian).

Kedelapan, Penelitian oleh Ridwan (2018) berjudul "Pengembangan Aplikasi Absensi Siswa Menggunakan Teknologi Geofencing dan NFC Berbasis Mobile Application". Penelitian ini mengembangkan aplikasi absensi yang mengkombinasikan dua teknologi: Geofencing dan NFC. Geofencing digunakan untuk memantau apakah siswa berada di dalam area tempat kursus, sedangkan proses absensi utamanya dilakukan dengan *tapping* kartu NFC siswa ke *smartphone* pengajar.

Persamaan: Menggunakan Geofencing sebagai salah satu komponen untuk validasi lokasi siswa.

Perbedaan (Celah Penelitian): Validasi utama sistem ini bergantung pada NFC, bukan Geofencing. Geofencing hanya bertindak sebagai pemantau area sekunder. Implementasi *geo-fence* yang dideskripsikan adalah sebagai "area" 2D, bukan volume 3D.

<b>Judul Penelitian</b>	<b>Peneliti</b>	<b>Objek Penelitian</b>	<b>Metode &amp; Teknologi</b>
Implementasi Algoritma Haversine pada Absensi Perkuliahan Berbasis Online di Fakultas Teknik Universitas Islam Sumatera Utara	Rahmat Fauzan, Khairuddin Nasution, & Tasliyah Haramaini	Mahasiswa	Flutter, Laravel, Algoritma Haversine (Pagar Lingkaran 2D / Radius)
Implementasi Geofencing pada Sistem Presensi Siswa dengan Verifikasi Lokasi Secara Real-Time: Studi Kasus SMK Islamic Center Baiturrahman	Bagus Sudirman, Jarot Dian Susatyono, & Muhammad Nazir Azhari	Mahasiswa	<i>Prototyping</i> , Web & Mobile, QR Code, Geolocation (GPS)

<b>Judul Penelitian</b>	<b>Peneliti</b>	<b>Objek Penelitian</b>	<b>Metode &amp; Teknologi</b>
Implementing QR code and Geolocation Technologies for the Student Attendance System	Semmy Wellem Taju, Yonatan Putra Mamahit, & Jeremy Andrew Pongantung	Mahasiswa	R&D, Geofencing 2D (Pagar Datar)
GPS Based Attendance System Using Geofencing	Miss. Harshada Sudam Gite, Miss. Pooja Balasaheb Kawade, Miss. Harshada Sanjay Aushikar, Miss. Ashwini Mothabhau Gangurde, Prof. N. S. Khairnar	Siswa	GPS, Android App, MySQL, Bluetooth Low Energy
Aplikasi Presensi Siswa Menggunakan Metode Geofencing	Muhammad Fadhlorrohman	Siswa	Geofencing, Waterfall, OOAD, Android, Google Geofencing API,

<b>Judul Penelitian</b>	<b>Peneliti</b>	<b>Objek Penelitian</b>	<b>Metode &amp; Teknologi</b>
			Google Maps API, Firebase Cloud Messaging.
Pembangunan Aplikasi Monitoring Kehadiran Siswa Berbasis Android Di SMA PLUS AL-WAHID	Dedus Lihoko	Siswa	SDLC Waterfall, Android, Location-Based Service (LBS), Geofencing, WhatsApp API, Firebase.
Mobile Based Student Attendance System Using Geo-Fencing With Timing and Face Recognition	Babatunde, A.N., Oke, A.A., Babatunde, R.S, Ibitoye, O. & Jimoh, E.R.	Siswa	Hybrid approach , Winding Number algorithm (untuk Geofence), Geofencing (Google API, Geofire) , Face Recognition (OpenCV, Google ML Kit) , Android (Android Studio,

<b>Judul Penelitian</b>	<b>Peneliti</b>	<b>Objek Penelitian</b>	<b>Metode &amp; Teknologi</b>
			JAVA, XML) , Firebase , SQLite
Pengembangan Aplikasi Absensi Siswa Menggunakan Teknologi Geofencing dan NFC Berbasis Mobile Application	Ridwan	Siswa	Extreme Programming (XP), Geofencing, NFC, Mobile Application (Android), MySQL.

Tabel 2.1 *Rekapitulasi* Penelitian Terdahulu

Berdasarkan tinjauan pustaka pada tabel 2.1, dapat disimpulkan bahwa penelitian tentang absensi berbasis lokasi telah banyak dilakukan, namun ditemukan celah penelitian (*research gap*) yang signifikan. Seluruh delapan penelitian yang ditinjau, meskipun menggunakan berbagai pendekatan teknis, secara konsisten masih terbatas pada implementasi pagar virtual 2D (datar).

Pembatasan ini terlihat jelas dalam berbagai metodologi yang digunakan:

1. Fauzan dkk. (2025) menggunakan Algoritma Haversine (lingkaran 2D).
2. Lihoko (2024) secara eksplisit menggunakan validasi poligon 2D (`isInsidePolygon`).

3. Babatunde dkk. (2022) menggunakan Algoritma Winding Number untuk batas 2D.
4. Fadhlurrohman (2019) dan Gite dkk. (2023) mengandalkan API Geofencing standar Google yang pada dasarnya beroperasi secara 2D.
5. Penelitian lain seperti Sudirman dkk. (2025) berfokus pada penerapan umum Geofencing 2D.
6. Sementara Taju dkk. (2024) dan Ridwan (2018) menggunakan validasi yang lebih lemah seperti *Geolocation* (titik) atau NFC, dengan *Geofencing 2D* hanya bertindak sebagai pemantau area sekunder.

Metode yang umum digunakan, seperti Algoritma Haversine oleh Fauzan dkk. (2025), juga cenderung menciptakan pagar berbentuk lingkaran (*radius*) yang kurang presisi untuk memetakan area sekolah yang lazimnya berbentuk persegi panjang atau kotak.

Lebih jauh lagi, ditemukan celah manipulasi vertikal yang signifikan. Tidak ada satupun dari delapan penelitian yang ditinjau yang membahas atau mengimplementasikan validasi ketinggian (*altitude*). Hal ini menyebabkan semua sistem 2D tersebut masih dapat dimanipulasi oleh siswa dari gedung tinggi di sekitar sekolah yang masih masuk dalam radius atau poligon pagar 2D.

Oleh karena itu, penelitian ini mengusulkan sebuah kebaruan (*novelty*) untuk mengisi celah tersebut dengan mengimplementasikan teknologi geofencing volume 3D menggunakan Algoritma AABB (Axis-Aligned Bounding Box). Algoritma ini tidak hanya memvalidasi lokasi siswa dalam perimeter 2D (*latitude*, *longitude*) yang berbentuk kotak, tetapi juga memvalidasi ketinggian (*altitude*) siswa dalam volume 3D yang telah ditentukan. Pendekatan *rectangular* (kotak)

dengan AABB juga lebih sesuai untuk merepresentasikan area sekolah yang umumnya berbentuk persegi panjang, dibandingkan dengan pendekatan *circular* (lingkaran) menggunakan Haversine. Dengan demikian, penelitian ini diharapkan dapat meningkatkan akurasi dan keamanan sistem absensi di SMA Negeri 5 Medan.

## **2.2 Landasan Teori**

### **2.2.1 Pengertian Aplikasi**

Aplikasi adalah suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan berbagai aktifitas. Untuk menulis program aplikasi dibutuhkan bahasa pemrograman (*language software*) yang bisa berbentuk *assembler*, *compiler*, ataupun *interpreter*. *Language software* ini berfungsi agar dapat menulis program dengan bahasa yang lebih mudah dan akan diterjemahkan ke dalam bahasa mesin agar bisa dimengerti oleh komputer. (Buyens: 2001).

### **2.2.2 Pengertian Mobile Application**

Aplikasi mobile berasal dari kata *application* dan *mobile*. *Application* memiliki arti penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. *Mobile* memiliki arti perpindahan dari suatu tempat ke tempat yang lain. Aplikasi mobile dapat diartikan sebagai sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah-pindah dari suatu tempat ke tempat yang lain dan juga mempunyai ukuran yang ringkas. Aplikasi mobile dapat digunakan melalui bermacam-macam perangkat nirkabel, seperti *pager*, *PDA*, *smart phone*, dan lainnya. Buyens (2001).

### **2.2.3 Pengertian Absensi**

Absensi adalah suatu pendataan kehadiran, bagian dari pelaporan aktivitas suatu institusi, atau komponen institusi itu sendiri yang berisi data-data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan (Tresnani dan Munir, 2012).

#### **2.2.3.1 Jenis-Jenis Absensi**

Kita mengenal beberapa jenis absensi. Yang membedakan jenis-jenis absensi tersebut adalah cara penggunaannya dan tingkat dayagunanya. Secara umum jenis-jenis absensi dapat dikelompokkan menjadi dua, yaitu:

##### **1. Absensi Manual**

Siswa manual dapat diartikan sebagai proses pendataan kehadiran dengan cara menggunakan alat tulis seperti pena berbentuk paraf atau tanda tangan.

##### **2. Absensi Non-Manual (menggunakan alat)**

Siswa non-manual adalah suatu cara untuk melakukan pendataan kehadiran dengan menggunakan sistem terkomputerisasi, bisa menggunakan kartu dengan *barcode*, *fingerprint* ataupun dengan memasukkan NIP dan sebagainya.

### **2.2.3.2 Pengelolaan Absensi**

Beberapa manfaat melakukan pengelolaan absensi dengan menggunakan teknologi informasi, seperti:

1. Absensi bisa dilakukan dengan lebih mudah dan cepat
2. Informasi atau data hasil pencarian yang disajikan lebih lengkap
3. Antar divisi saling terhubung (pencarian data kehadiran dari satu divisi ke divisi lain)
4. Mempermudah dalam melakukan Entry dan Update data
5. Memudahkan pembuatan laporan dan rekapitulasi
6. Terdapat fasilitas informasi

Dengan demikian sistem yang telah terkomputerisasi dapat mempermudah pekerjaan bagian admin, serta meningkatkan pelayanan dan kinerja dari suatu instansi atau perusahaan.

### **2.2.4 System**

Sistem adalah sekelompok elemen yang saling berhubungan yang berinteraksi dan saling mempengaruhi untuk mencapai tujuan bersama, dengan mempertimbangkan lingkungan dan kendala yang mempengaruhi kinerja sistem. (Hanna, D. P., & Schreiber, R. L, 2021).

### **2.2.5 Location access control**

*Kontrol akses lokasi* adalah mekanisme keamanan yang memungkinkan pengguna untuk mengakses sebuah sistem atau aplikasi berdasarkan lokasi tertentu. Dalam konteks perancangan sistem absensi siswa dengan akses lokasi menggunakan *geofencing*, kontrol akses lokasi digunakan untuk memastikan

bahwa siswa hanya dapat mencatat kehadiran mereka ketika mereka berada di dalam area sekolah yang telah ditentukan. Definisi lain dari kontrol akses lokasi adalah mekanisme keamanan yang memungkinkan aplikasi atau sistem membatasi akses pengguna berdasarkan lokasi *geografis*. Kontrol akses lokasi dapat diterapkan pada berbagai aplikasi, seperti sistem perpustakaan *online*, aplikasi transportasi, dan lainnya. Pada praktiknya, kontrol akses lokasi sering dikombinasikan dengan teknologi *geofencing* untuk memastikan bahwa pengguna hanya dapat mengakses aplikasi atau sistem didalam area tertentu yang telah ditetapkan (Munir, R., Saputra, A. H., & Febriani, R, 2021)

### **2.2.6 Geofencing**

Geofencing adalah teknik yang digunakan untuk memantau dan mendeteksi pergerakan perangkat mobile dalam suatu zona geografis tertentu. Dengan menetapkan perimeter virtual di sekitar area yang diinginkan, sistem ini dapat memberikan peringatan ketika perangkat mobile melintasi batas tersebut. Geofencing sering digunakan dalam konteks bisnis untuk memastikan bahwa karyawan atau kendaraan perusahaan tetap berada dalam area yang relevan untuk aktivitas mereka, seperti pengemudi pengiriman atau teknisi lapangan (Reclus, 2013).

Dibawah ini adalah definisi geofencing dari beberapa ahli:

#### **1. Hossein Soltani**

Geofencing adalah teknologi berbasis lokasi yang memungkinkan pengguna untuk membuat area geografis virtual dan mengirimkan notifikasi atau memicu tindakan otomatis ketika sebuah perangkat

seluler memasuki atau keluar dari area tersebut (Hossein Soltani dkk., 2020)

## 2. Jeonghoon Mo dan Jun Seok Lee

Geofencing merupakan teknologi penentuan posisi lokasi yang memungkinkan pengguna untuk membuat zona virtual di peta digital dan mengaktifkan notifikasi saat sebuah perangkat masuk atau keluar dari zona tersebut (Jeonghoon Mo dan Junseok Lee, 2021).

### 2.2.6.1 Sejarah Perkembangan Geofencing

Konsep geofencing pertama kali muncul pada awal tahun 2000-an seiring dengan berkembangnya teknologi GPS dan mobile computing. Berikut adalah perkembangan historis geofencing:

#### 1. Fase Awal (2000-2005):

Geofencing pertama kali diimplementasikan dalam sistem manajemen armada kendaraan komersial untuk melacak pergerakan truk dan kendaraan logistik (Schiller & Voisard, 2004). Teknologi masih terbatas pada perangkat GPS khusus dengan akurasi rendah.

#### 2. Fase Pertengahan (2006-2012):

Munculnya *smartphone* dengan GPS built-in (iPhone 2007, Android 2008) membuka peluang aplikasi geofencing consumer-level, Google memperkenalkan Geofencing API pada Android platform (2011), Aplikasi pertama yang populer: location-based reminders dan retail marketing.

### 3. Fase Modern (2013-Sekarang):

*Integrasi dengan IoT (Internet of Things) dan beacon technology, Peningkatan akurasi dengan hybrid positioning (GPS + Wi-Fi + cellular), Penggunaan meluas ke berbagai sektor: pendidikan, kesehatan, keamanan, dan smart city (Küpper, 2005; Zeimpekis et al., 2018)*

#### 2.2.6.2 Metode Dasar Geofencing

Terdapat beberapa pendekatan geometris dalam implementasi geofencing:

##### 1. Circular Geofencing (Radial)

Metode paling sederhana yang menggunakan radius dari titik pusat. Sebuah titik dianggap berada dalam geofence jika jarak dari pusat  $\leq$  radius yang ditentukan.

Kelebihan:

- Komputasi sangat cepat ( $O(1)$ )
- Mudah diimplementasikan

Kekurangan:

- Tidak cocok untuk area dengan bentuk tidak beraturan
- Tidak dapat mengikuti bentuk bangunan sebenarnya

##### 2. Polygonal Geofencing

Menggunakan poligon (segi banyak) untuk mendefinisikan area dengan bentuk tidak beraturan. Menggunakan algoritma *Point-in-Polygon* seperti *Ray Casting* atau *Winding Number*.

Kelebihan:

- Fleksibel untuk area kompleks
- Akurat mengikuti bentuk geografis sebenarnya

Kekurangan:

- Kompleksitas komputasi lebih tinggi ( $O(n)$ )
- Membutuhkan lebih banyak koordinat untuk definisi area

### 3. Rectangular Geofencing (AABB)

Menggunakan kotak dengan sisi sejajar sumbu koordinat (bounding box). Validasi dilakukan dengan membandingkan koordinat dengan batas min/max.

Point(x,y)  $\in$  Rectangle jika:  $x_{\min} \leq x \leq x_{\max}$  DAN  $y_{\min} \leq y \leq y_{\max}$ .....(2.1)

Kelebihan:

- Komputasi sangat efisien (6 perbandingan untuk 3D)
- Mudah dikonfigurasi dan dipahami

Kekurangan:

- Terbatas untuk area berbentuk kotak
- Tidak fleksibel untuk kontur kompleks

### 4. 3D Geofencing (Volumetric)

Menambahkan dimensi ketinggian (altitude) pada metode geofencing tradisional untuk mencegah spoofing lokasi dengan ketinggian palsu.

$$(p_{\text{Min}.x} \leq p.x \leq p_{\text{Max}.x}) \text{ DAN } (p_{\text{Min}.y} \leq p.y \leq p_{\text{Max}.y}) \text{ DAN } (p_{\text{Min}.z} \leq p.z \leq p_{\text{Max}.z}) \dots\dots\dots(2.2)$$

**2.2.6.3 Kelebihan dan Kekurangan Geofencing**

Kelebihan Geofencing Secara Umum:

1. Sistem dapat memberikan respons instan tanpa intervensi manual  
(Gupta & Kumar, 2021)
2. Mengurangi kebutuhan pemantauan manual dalam manajemen lokasi karyawan atau asset
3. Dapat menangani banyak perangkat dan zona secara bersamaan
4. Area dapat disesuaikan dengan kebutuhan spesifik organisasi
5. Dapat diintegrasikan dengan sistem absensi, ERP, atau manajemen keamanan.

Kekurangan Geofencing:

1. Memerlukan koneksi internet dan *GPS* aktif untuk fungsi optimal  
(Zeimpekis et al., 2018)
2. Akurasi *GPS komersial* ±5-10 meter, dapat terpengaruh oleh: Efek *multipath* di area urban, Gangguan atmosfer, Kualitas perangkat keras  
(Zandbergen, 2009)
3. Pemantauan lokasi kontinu menguras baterai perangkat mobile
4. Pelacakan lokasi terus-menerus menimbulkan isu *privasi* pengguna

5. Lokasi *GPS* dapat dipalsukan menggunakan aplikasi fake *GPS* atau perangkat keras khusus (Jansen et al., 2017)

#### **2.2.6.4 Cara Kerja Geofencing**

Berikut ini adalah cara kerja geofencing menurut Gupta dan Kumar (Gupta, A., & Kumar, V, 2021)

1. Mendefinisikan Area Geografis, Pertama, pengguna harus mendefinisikan area geografis yang akan dipantau dan dikontrol. Area ini dapat ditentukan dengan menandai titik-titik atau batas-batas pada peta.
2. Mengatur Parameter, Setelah area geografis ditentukan, pengguna dapat mengatur parameter yang akan dimonitor. Parameter ini dapat mencakup pergerakan, waktu, atau aktivitas pengguna.
3. Memantau Perangkat Seluler, Sistem geofencing secara terus menerus memantau perangkat seluler pengguna untuk melacak posisi dan aktivitasnya. Jika perangkat seluler berada dalam area geografis yang ditentukan dan parameter yang ditentukan terpenuhi, sistem akan memicu respons atau tindakan yang telah ditentukan sebelumnya.
4. Memberikan Respons, Respons atau tindakan yang dipicu oleh sistem geofencing dapat mencakup pemberitahuan, pesan teks, atau tindakan otomatis lainnya. Tanggapan ini dapat diterima oleh pengguna atau pihak terkait yang berwenang.

### **2.2.7 Algoritma AABB (Axis-Aligned Bounding Box)**

*Axis-Aligned Bounding Box (AABB)* merupakan metode deteksi containment yang efisien dalam ruang tiga dimensi berbasis geometri komputasional. AABB didefinisikan sebagai sebuah kotak pembatas (bounding box) yang sisi-sisinya sejajar (aligned) dengan sumbu koordinat kartesian X, Y, dan Z, sehingga memudahkan proses perhitungan komputasi tanpa memerlukan transformasi rotasi (Ericson, 2004).

Metode AABB banyak digunakan dalam berbagai aplikasi, termasuk:

- Computer Graphics: Collision detection dan culling optimization (Akenine-Möller et al., 2018)
- Game Development: Spatial partitioning dan physics simulation (Gregory, 2018)
- Location-Based Services (LBS): Geofencing dan proximity detection (Küpper, 2005)

#### **2.2.7.1 Prinsip Kerja Algoritma AABB**

Pengujian containment untuk menentukan apakah sebuah titik P dengan koordinat tiga dimensi (p.x, p.y, p.z) berada didalam sebuah bounding box dilakukan dengan pendekatan interval testing (Ericson, 2004). Sebuah titik dinyatakan berada didalam AABB jika dan hanya jika koordinat titik tersebut memenuhi batas minimum dan maksimum pada setiap sumbu koordinat secara bersamaan.

Kondisi *Containment*, Sebuah titik  $P \in \text{AABB}$  jika:

$$(pMin.x \leq p.x \leq pMax.x) \wedge (pMin.y \leq p.y \leq pMax.y) \wedge (pMin.z \leq p.z \leq pMax.z) \dots\dots\dots(2.3)$$

Dimana:

- Sumbu X (Longitude): Representasi batas bujur timur-barat
- Sumbu Y (Latitude): Representasi batas lintang utara-selatan
- Sumbu Z (Altitude): Representasi batas ketinggian vertikal

### 2.2.7.2 Kompleksitas Algoritma

Algoritma AABB memiliki kompleksitas waktu  $O(1)$  untuk setiap operasi point-in-box test karena hanya melibatkan enam perbandingan sederhana (Ericson, 2004). Hal ini menjadikan AABB sangat efisien untuk sistem real-time seperti aplikasi mobile berbasis lokasi.

### 2.2.7.3 Penerapan dalam Geofencing

Dalam konteks geofencing tiga dimensi, AABB menyediakan mekanisme validasi lokasi yang efisien dengan mempertimbangkan tidak hanya koordinat horizontal (latitude dan longitude), tetapi juga komponen vertikal (altitude) untuk mencegah spoofing lokasi menggunakan ketinggian palsu (Zandbergen, 2009).

## 2. 2.8 Integrasi Aplikasi Mobile

integrasi aplikasi *mobile* merupakan proses menggabungkan dua atau lebih aplikasi *mobile* yang berbeda menjadi satu aplikasi yang terintegrasi secara unik (Nazeer, I., & Awan, M. A, 2021).

Secara umum integrasi aplikasi *mobile* didefinisikan sebagai proses yang mengintegrasikan dua atau lebih aplikasi *mobile* yang berbeda untuk memperluas fungsionalitas dan memungkinkan aplikasi bekerja bersama secara harmonis, hal ini memungkinkan pengguna untuk menggunakan berbagai layanan dan fitur dari aplikasi yang berbeda, yang sebelumnya tidak dapat dilakukan dalam aplikasi tunggal.

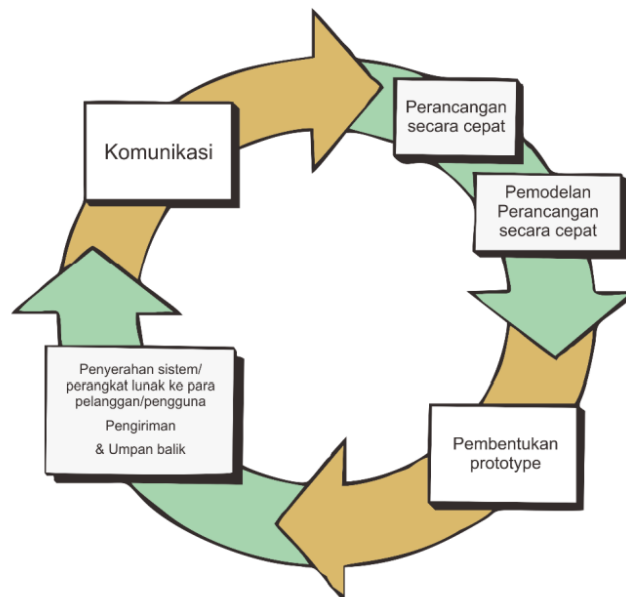
Integrasi aplikasi *mobile* juga dapat membantu meningkatkan efisiensi penggunaan aplikasi, menghemat waktu dan usaha pengguna, serta meningkatkan produktivitas.

### **2.2.9 Metodologi Pengembangan Sistem**

Model pengembangan yang digunakan dalam perancangan aplikasi ini adalah model *Prototype*. Model *Prototype* digunakan pada penelitian ini dimaksudkan untuk mendapatkan representasi dari pemodelan aplikasi yang akan dibuat. Rancangan aplikasi awalnya berbentuk *mockup* selanjutnya akan dievaluasi pengguna (Yoko et al., 2019). Manfaat utama *prototype* adalah untuk mengurangi resiko tidak diterimanya hasil pengembangan suatu perangkat lunak serta pengulangan kerja di kemudian hari". Model *Prototype* dipilih karena memungkinkan pengembangan sistem secara iteratif dengan melibatkan pengguna sejak tahap awal. Model ini terdiri dari lima tahapan utama: komunikasi, perencanaan cepat, pemodelan, konstruksi, dan penyerahan kepada pengguna (Pressman & Maxim, 2020).

Pada komunikasi tahap adalah tahap awal dalam *prototype* guna mengetahui permasalahan. Pada perencanaan, tahapan ini dikerjakan dengan

kegiatan penentuan sumberdaya, spesifikasi untuk pengembangan berdasarkan kebutuhan sistem, dan tujuan berdasarkan pada hasil komunikasi yang dilakukan agar pengembangan dapat sesuai dengan yang diharapkan. Pemodelan, tahapan selanjutnya ialah representasi atau menggambarkan model sistem yang akan dikembangkan seperti proses dengan perancangan program. Konstruksi, tahapan ini digunakan untuk membangun, menguji-coba sistem yang dikembangkan. Proses instalasi dan penyediaan user-support juga dilakukan agar sistem dapat berjalan dengan sesuai. Berikut gambar 2.1 alur Prototype serta penjelasan :



Gambar 2.1 alur prototype

## 1. Komunikasi

Tahap komunikasi yang pertama ada menyiapkan dan menganalisis kebutuhan dari program yang akan dikerjakan. Kebutuhan dilakukan secara

intensif untuk mempesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user.

## 2. Perancangan

Tahap perancangan membuat gambaran yang jelas mengenai tampilan dan antarmuka yang kemudian akan dibuat program. Tahap ini mentranslasi kebutuhan perangkat dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

## 3. Pemodelan

Tahap pemodelan aplikasi menggunakan kode program yang telah ditentukan dan merubah rancangan menjadi program sehingga memudahkan dalam pembuatan dan penggunaan oleh user.

## 4. Konstruksi

Tahap ini membangun kerangka atau rancangan prototipe, dilakukan pengujian pada program yang sudah dibuat untuk mengetahui kekurangan dari program.


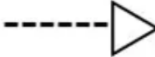




## 5. Penyerahan

Tahap ini dilakukan pemeliharaan pada program serta perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

### **2.2.9.1 Use Case Diagram**

Use case diagram banyak digunakan untuk menggambarkan kebutuhan dan fungsionalitas yang diinginkan dari produk perangkat lunak,

pengguna 12 aplikasi dapat menggunakan aplikasi berdasarkan kebutuhan yang telah ditentukan. Berikut ini alur-alur pada use case diagram ditampilkan pada tabel 2.2

No	Nama	Gambar	Keterangan
1	Actor		Mendeskripsikan seorang pengguna aplikasi atau alat berkomunikasi dengan use case
2	Generalization		Mendeskripsikan sebuah objek yang mewarisi objek di atasnya.
3	Include		Menyatakan bahwa penggunaan use case selalu fungsionalitas pada use case lainnya.
4	Extend		Menyatakan satu use case secara opsional menggunakan fungsionalitas dari use case lainnya jika kondisi penuh.
5	Association		Menghubungkan actor dengan use case.
6	Use case		Interaksi antara sistem dan aktor agar menghasilkan suatu hasil aplikasi

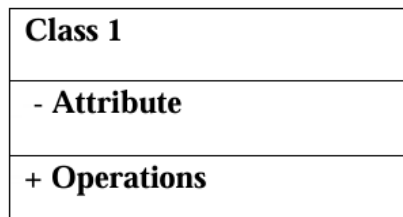
Tabel 2.2 Notasi-notasi *use case*

### 2.2.9.2 Class Diagram

Class diagram digunakan sebagai salah satu kumpulan umum transformasi class yang bertujuan untuk meningkatkan kualitas diagram class tingkat spesifikasi atau desain, class diagram dapat membantu pengembangan aplikasi untuk merancang struktur aplikasinya sebelum penulisan source code. Berikut ini adalah notasi dari class diagram:

#### 1. Class

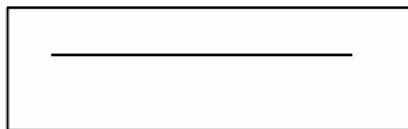
Menjelaskan kumpulan objek-objek seperti class dibagian atas, atribut dibagian tengah, serta operasi dibagian bawah. Objek- objek tersebut akan menghubungkan class yang satu dengan yang lain atau sebaliknya.



Gambar 2.2 contoh *class*

#### 2. Association

Association menggambarkan kedua Class kedua kelas memiliki koneksi yang sama dan bukan sebagai kelas abstrak atau sub kelas.



Gambar 2.3 contoh *association*

### 3. Composition

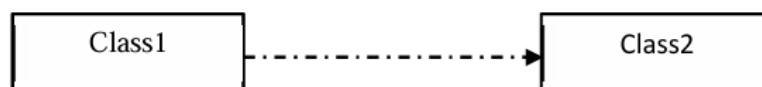
Hubungan ditampilkan sebagai garis padat dengan berlian yang diisi diujung asosiasi, yang terhubung ke kelas yang mewakili keseluruhan. Pada gambar dibawah ini, kelas 2 tidak dapat berdiri sendiri.



Gambar 2.4 contoh *Composition*

### 4. Dependency

Suatu Hubungan antara dua class memberikan penjelasan bahwa perubahan yang terjadi disalah satu class memberikan pengaruh pada class lainnya. dapat dilihat contoh gambar dibawah ini :



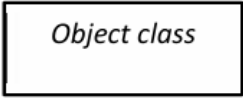




Gambar 2.5 contoh *dependency*

#### 2.2.9.3 Sequence Diagram

Sequence diagram digunakan terutama untuk menunjukkan interaksi antara objek secara berurutan agar interaksi tersebut terjadi. Sama seperti diagram class, pengembang biasanya berpikir urutan diagram dimaksudkan khusus. Diagram ini dimulai dari atas ke bawah dan menggambarkan aliran





use case dengan setiap objek memiliki garis hidup masing-masing. Berikut ini notasi sequence diagram dapat dilihat pada tabel dibawah ini :



No	Simbol	Keterangan
1		Simbol yang melambangkan seorang user dan biasanya pengguna sistem
2		Life line atau garis hidup, yang menandakan aktivitas objek dalam basis
3		aktu Objek-objek kelas pada diagram.
4		Activation Waktu hidup atau life
5	Form data barang 	Simbol yang mengindikasikan sebuah objek yang telah dibuat, komunikasi antara objek

Tabel 2.3 Notasi *sequence diagram*

#### 2.2.9.4 Activity diagram

Diagram Activity memperlihatkan alur kerja aktivitas satu dengan aktivitas yang lainnya. Activity Diagram mempunyai peran seperti halnya Flowchart, akan tetapi perbedaannya dengan Flowchart adalah Activity Diagram bisa mendukung perilaku paralel sedangkan Flowchart tidak bisa, Diagram Activity akan menunjukkan tahapan pengambilan suatu keputusan dan percabangan dari awal hingga akhir. Berikut ini adalah tabel notasi *Diagram Activity 2.3*.

No	Simbol	Nama dan Kegunaan
1		Start point atau initial activity merupakan awal aktivitas modul sistem aplikasi
2		End poin atau final activity menunjukkan akhir dari aktivitas modul sistem aplikasi
3		Activity atau Action state menunjukkan aktivitas yang dilakukan dilakukan dalam modul sistem aplikasi
4		Transition untuk menghubungkan aktivitas selanjutnya setelah aktivitas sebelumnya

No	Simbol	Nama dan Kegunaan
5		Decisions menunjukkan aktivitas yang harus dipilih apakah pilihan pertama atau kedua
6		Fork digunakan untuk menunjukkan kegiatan secara paralel

Tabel 2.4 Notasi *Diagram Activity*

### 2.2.10 Unified Modeling Language (UML)

Menurut (Salahudin & Rosa, 2013) “UML (*Unified Modelling Language*) merupakan bahasa visual untuk pemodala dan komunikasi mengenai sebuah sistem diagram dan tek pendukung, requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasikan objek. Tujuan pembuatan UML yaitu merepresentasikan suatu sistem dengan menggunakan elemen grafis yang membentuk suatu diagram. Tahapan– tahapan pembuatan sistem UML terdapat 4 segmen, yakni *use case diagram*, *class diagram*, *sequence diagram*, dan *activity diagram*

### 2.2.11 Android

Android merupakan sistem operasi berbasis *Linux* yang digunakan untuk *mobile* (Lee, 2011:2). *Android* awalnya dikembangkan oleh sebuah *startup* bernama *Android.Inc*. Pada tahun 2005 sebagai bagian dari strategi memasuki ruang *mobile*, Google membeli *Android* dan mengambil alih tim

pengembangnya. Setelah *Android.Inc* menjadi grup Google, para pendiri *Android* diantaranya Andy Rubin, Rich Milner, Nick Sears, dan Chris White berpindah ke Google.

Didalam buku yang berjudul “Mudah Membuat dan Berbisnis Aplikasi Android dengan Android Studio”, definisi dari Android adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet (Yudha Yudhanto dan Ardhiy Wijayanto, 2017:1).

Terdapat beberapa faktor pertimbangan untuk realisasi pembuatan aplikasi berbasis yakni sebagai berikut:

1. Faktor kecepatan, tingkat efisiensi aplikasi dalam menyajikan data, proses, dan memberikan *output* data secara cepat dan sesuai keinginan konsumen.
2. Aspek produktivitas, kebermanfaatan aplikasi dalam peningkatan produktivitas pengguna.
3. Kreativitas desain, nilai tambah yang dapat menarik minat pengguna menggunakan aplikasi tersebut.
4. Fleksibilitas, aplikasi pada *android* lebih memungkinkan untuk berfungsi dengan baik disegala kondisi.

Sistem operasi *android* sebagai sistem operasi *mobile* yang terbuka (*open source*) memiliki banyak versi dalam perkembangannya, diantaranya:

1. Android (1.6) *Donuts*
2. Android (2.0) *Eclair*

3. Android (2.2) *Froyo*
4. Android (2.3) *Gingerbread*
5. Android (3.0) *Honeycomb*
6. Android (4.0) *Ice Cream Sandwich*
7. Android (4.1) *Jelly Bean*
8. Android (4.4) *KitKat*
9. Android (5.0) *Lollipop*
10. Android (6.0) *Marshmallow*
11. Android (7.0) *Nougat*
12. Android (8.0) *Oreo*
13. Android (9.0) *Pie*
14. Android (10) *Queen Cake*
15. Android (11) *Red Velvet Cake*
16. Android (12) *Snow Cone*
17. Android (13) *Tiramisu*
18. Android (14) *Upside Down Cake*
19. Android (15) *Vanilla Ice Cream*(terbaru)

#### **2.2.12 Android SDK (Software Development Kit)**

*Android SDK* adalah *tool API* (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman Java. *Android* merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware, dan aplikasi kunci yang rilis oleh Google. Saat ini disediakan *Android SDK* (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada

*platform Android* menggunakan bahasa pemrograman Java (Nazrudin Safaat 2011).

### **2.2.13 Android Studio**

Android Studio merupakan sebuah *software tools Integrated Development Environment (IDE)* untuk *platform Android*. Definisi tersebut dikutip dari buku yang berjudul Mudah Membuat dan Berbisnis Aplikasi Android dengan Android Studio, (Yudha Yudhanto dan Ardhiy Wijayanto, 2017:1).

Android Studio menyediakan alat pengembang android terintegrasi untuk pengembangan dan debugging:

<http://developer.android.com/tools/studio/index.html>

Diatas kemampuan yang Anda harapkan dari IntelliJ, Android Studio menawarkan:

1. Berbasis Gradle membangun dukungan.
2. Refactoring android, spesifik dan perbaikan yang cepat.
3. Alat Lint untuk menangkap kinerja, kegunaan, kompatibilitas versi dan masalah lainnya.
4. ProGuard dan aplikasi-penanda tanda tangan kemampuan.
5. Penyihir berbasis template untuk membuat desain Android umum dan komponen.
6. Sebuah layout editor yang memungkinkan untuk drag-and-drop UI komponen, layout pratinja pada beberapa konfigurasi layar, dan banyak lagi.

#### **2.2.14 PHP (PHP Hypertext Processor)**

*PHP* adalah bahasa *server-side scripting* yaitu bahasa yang berbetuk skrip yang ditempatkan didalam server dan diproses diserver, dapat menyatu dengan *HTML* untuk membuat halaman web yang dinamis. *PHP* merupakan software yang *open source* dan mampu lintas platform. *PHP* dapat dibangun sebagai modul pada *web server Apache* dan sebagai binary yang dapat berjalan sebagai *CGI*.

#### **2.2.15 Laravel**

*Laravel* adalah *framework* berbasis pemrograman *PHP* untuk membantu proses pengembangan sebuah *website* agar lebih maksimal (Dewaweb, 2022). Dengan menggunakan *Laravel*, *website* yang dihasilkan akan lebih dinamis dan kehadiran *framework Laravel* menjadikan bahasa pemrograman *PHP* menjadi lebih powerful (Dewaweb, 2022).

Menurut Dicoding (2024), *Laravel* adalah *framework PHP open-source* yang dirancang untuk membuat pengembangan web menjadi lebih mudah dan lebih cepat. Telkom University (2024) menjelaskan bahwa *Laravel* merupakan *framework PHP open-source* yang powerful dan mudah dipahami dengan mengikuti pola desain model-view-controller.

#### **2.2.16 JSON (JavaScript Object Notation)**

*JSON (JavaScript Object Notation)* adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar *ECMA-262* Edisi ke-3 – Desember

1999. *JSON* merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python* dll. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran-data (Putra, 2012).

### 2.2.17 MySQL

*MySQL* merupakan sebuah software yang berfungsi sebagai suatu *database server* yang cukup terkenal. *Database server* itu sendiri merupakan suatu *software* yang bertugas untuk melayani permintaan (*request*) *query* dari *client* (Putra, 2012).

*MySQL* sebagai suatu *database server* mempunyai beberapa kemampuan, salah satunya harus menyediakan suatu sistem manajemen *database* yang dapat mengatur bagaimana menyimpan, memanipulasi, mengakses dan mentransaksikan data lainnya. *MySQL* cepat sekali berkembang, karena *MySQL* merupakan *software* yang *Open Source*.

*MySQL* adalah *Relational Database Management System (RDBMS)* yang didistribusikan secara gratis dibawah *lisensi GPL (General Public License)*. Dimana setiap orang bebas untuk mengembangkan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan suatu konsep utama dalam *database* yaitu, *SQL (Structured Query Language)*. *SQL* adalah sebuah konsep pengolahan *database*, terutama untuk pemilihan atau seleksi data dari *database* dan penggunaannya dapat dikerjakan dengan mudah secara otomatis (Eko 2006).

Keandalan suatu sistem *database (DBMS)* dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah *SQL*, yang dibantu oleh user maupun program-program aplikasinya. Sebagai *database server*, *MySQL* dapat dilakukan pengolahan *database server* dengan menggunakan *query* data. Hal ini terbukti untuk *query* yang diakses oleh *single user*, kecepatan *query MySQL* bisa sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase*.

### **2.2.18 Application Programming Interface (API)**

*Application Programming Interface (API)* adalah Bahasa dan format pesan yang digunakan aplikasi atau program untuk berkomunikasi dengan *Operational System (OS)* atau program kontrol lainnya seperti manajemen database. Implementasi API dengan cara menulis fungsi yang ada dalam program yang menyediakan tautan pada instruksi program yang akan dieksekusi. API sering disebut dengan *web service* karena API memberikan layanan *service* pada sisi *client*. Adapun fungsi pada API yaitu: JSON, XML, JavaScript, SOAP, REST (Pahlevy, 2017).

*Application Programming Interface (API)* merupakan antarmuka yang digunakan untuk mengakses layanan atau aplikasi dari suatu program. API memungkinkan developer tidak perlu mengulang dari awal pengembangan sistem dengan memanfaatkan fungsi yang sudah ada dari aplikasi lain. Keuntungan menggunakan API yaitu: penggunaan API fleksibel yang artinya dapat digunakan disistem operasi mana saja, dan bahasa yang digunakan mudah dimengerti seperti JSON (Hasanuddin, 2022).

### 2.2.19 REST API

Teknologi Representational state transfer (REST) diciptakan oleh Roy Fielding dari University of California. Aplikasi REST sering digunakan untuk pengembangan layanan berbasis *web* atau *mobile* karena ringan dan sederhana apabila dibandingkan dengan SOAP. Prinsip utama desain dari REST API adalah kinerja, portabilitas dan kesederhanaan (Soni, 2019).

REST API mendukung beragam sistem untuk berinteraksi serta menerima/mengirim data dengan mudah. Setiap penggunaan REST API, didukung oleh URL dan HTTP. Data dalam *database* pada suatu aplikasi dipetakan dengan *endpoint* API pada REST API (Rizki, 2018).

### 2.2.20 Mapbox

*Mapbox* merupakan penyedia peta daring kustom untuk situs *web* dan aplikasi yang telah berkembang pesat sejak tahun 2010. *Mapbox* adalah salah satu penyedia peta daring *kustom* terbesar di situs-situs ternama seperti *Foursquare*, *Pinterest*, *Evernote*, *Financial Times*, dan *Uber Technologies*.

*Mapbox* memperbanyak pilihan peta kustomnya untuk mengisi keterbatasan yang dimiliki penyedia peta seperti *Google Maps* (Permana, 2019). *Mapbox* adalah platform yang dirancang untuk mengintegrasikan elemen geospasial ke dalam aplikasi *web* dan *mobile*, menawarkan berbagai *tools* dan layanan yang memungkinkan *developer* membuat peta yang dapat disesuaikan dan interaktif dengan data *real-time* (Novriansyah dkk., 2023).

Menurut Hidayatulloh & Airlangga (2022), mengintegrasikan aplikasi pihak ketiga dapat membuat sistem informasi *geografis* terpadu dengan menggunakan *Mapbox* tetapi juga bisa mempunyai fitur yang lebih banyak.

*Mapbox* menyediakan layanan lokasi yang komprehensif termasuk data peta global, pembaruan lalu lintas *real-time*, *geocoding*, pencarian alamat, *routing*, dan panduan *navigasi* (Novriansyah dkk., 2023).

### **2.2.21 Firebase**

*Firebase* adalah suatu layanan dari Google untuk memberikan kemudahan bahkan mempermudah para developer aplikasi dalam mengembangkan aplikasinya (Dicoding, 2020). *Firebase* alias *BaaS (Backend as a Service)* merupakan solusi yang ditawarkan oleh Google untuk mempercepat pekerjaan developer (Dicoding, 2020). Dengan menggunakan *Firebase*, apps developer bisa fokus dalam mengembangkan aplikasi tanpa memberikan effort yang besar untuk urusan *backend* (Dicoding, 2020).

Penelitian ini menggunakan fitur yang disediakan *Firebase* yaitu *Firebase Realtime Database* yang merupakan database NoSQL di-hosting di cloud yang memungkinkan data disinkronkan secara realtime ke semua klien yang terhubung (Hasibuan & Triase, 2022). Menurut Payara & Tanone (2018), *Firebase Realtime Database* memungkinkan developer untuk menyimpan dan mensinkronkan data antar pengguna dan perangkat secara realtime.

Maulana (2020) menjelaskan bahwa *Firebase Realtime Database* menyediakan penyimpanan data yang cepat dan aman dengan kemampuan sinkronisasi real-time, yang sangat berguna untuk aplikasi yang memerlukan update data secara langsung.

### **2.2.22 Flutter**

*Flutter* adalah *framework* yang dibuat *Google* menggunakan bahasa *Dart* yang bersifat *cross platform*, dimana pengembangan menggunakan *framework* ini dapat membuat satu aplikasi yang dapat berjalan di dua sistem operasi baik *iOS* dan *Android* (Santoso dkk., 2020). *Flutter* merupakan sebuah *framework* yang dikembangkan serta dipopulerkan oleh perusahaan *teknologi* ternama *Google Inc.* dengan slogan '*build one run anywhere*' yang dapat dikembangkan serta dijalankan pada dua platform besar *Android* dan *iOS* (Saputra & Irawan, 2020).

Setiawan dkk. (2022) menjelaskan bahwa *Flutter* adalah *framework open source* yang dikembangkan oleh *Google* untuk membangun aplikasi *mobile*, *web*, dan *desktop* dari satu basis kode yang sama. Herdiansah dkk. (2022) menambahkan bahwa *Flutter* merupakan *framework* pengembangan aplikasi *mobile* yang memungkinkan developer untuk membuat aplikasi *native* di *iOS* dan *Android* dengan menggunakan satu basis kode yang sama.

### **2.2.23 Dart**

*Flutter* menggunakan bahasa pemrograman *Dart* yang dikembangkan oleh *Google* (Wijaya dkk., 2023). *Dart* adalah bahasa pemrograman berorientasi objek untuk terminal data *mobile* yang diperkenalkan *Google* pada tahun 2011 (Etteplan, 2024). *Dart* menawarkan *Flutter strong typing* dan *AOT (Ahead of Time) compilation*, yang memudahkan untuk membuat aplikasi yang cepat dan dapat diprediksi (Etteplan, 2024).

### 2.2.24 Local Development Environment

*Local development environment* adalah lingkungan pengembangan perangkat lunak yang diinstal pada komputer lokal developer, menyediakan semua komponen yang diperlukan untuk mengembangkan, menguji, dan *debug* aplikasi sebelum *deployment* ke *server* produksi (Humble & Farley, 2010). Dalam pengembangan aplikasi web berbasis *PHP* dan *framework Laravel*, *local development environment* umumnya menyediakan stack yang terdiri dari web server (*Apache/Nginx*), database management system (*MySQL/PostgreSQL*), dan *PHP* interpreter (Otwell, 2021).

Beberapa tools populer yang menyediakan *local development environment* antara lain *XAMPP*, *WAMP*, *MAMP*, dan *Laragon*. Masing-masing memiliki karakteristik berbeda dalam hal performa, kemudahan konfigurasi, dan fitur yang ditawarkan. Pemilihan tools yang tepat dapat signifikan meningkatkan produktivitas development dan mengurangi waktu setup environment (Spinellis, 2017).

Dalam penelitian ini, *Laragon* dipilih sebagai *local development environment* karena beberapa pertimbangan teknis yang relevan dengan kebutuhan pengembangan sistem absensi berbasis *Laravel*. Pertama, *Laragon* menyediakan fitur auto virtual host yang secara otomatis membuat *domain* lokal (*absensi.test*) ketika project *Laravel* dibuat, sehingga mempercepat proses setup dan testing integrasi dengan aplikasi *mobile Flutter*. Kedua, *Laragon* memiliki performa startup yang sangat cepat ( $\pm 1$  detik) dan konsumsi resource minimal dibandingkan *XAMPP*, yang penting untuk produktivitas development dalam waktu penelitian yang terbatas. Ketiga, dukungan multiple versi *PHP*

memungkinkan kompatibilitas testing dengan versi *PHP* yang sama dengan server produksi tanpa konflik. Keempat, integrasi built-in dengan *Composer*, *Git*, dan *terminal Cmder* memudahkan manajemen dependencies *Laravel* dan version control *source code*.

#### **2.2.25 Black– Box testing**

*Black-Box* Testing, yang juga dikenal sebagai Behavioral Testing atau *Specification-Based* Testing, adalah metode pengujian perangkat lunak yang berfokus pada fungsionalitas sistem tanpa memperhatikan struktur internal, desain, atau implementasi kode program (Pressman & Maxim, 2020). Pendekatan ini dinamakan "*black box*" karena penguji memperlakukan sistem sebagai kotak hitam yang hanya dapat diamati dari sisi input dan output, tanpa pengetahuan tentang mekanisme internal yang menghasilkan output tersebut (Sommerville, 2016).

Menurut Myers et al. (2011), *Black-Box* Testing adalah teknik pengujian yang memeriksa fungsionalitas aplikasi tanpa mengetahui struktur internal atau cara kerjanya. Penguji hanya perlu mengetahui apa yang seharusnya dilakukan oleh sistem, bukan bagaimana sistem melakukannya. Metode ini sangat efektif untuk memvalidasi apakah sistem memenuhi *requirement* yang telah ditentukan dalam tahap analisis kebutuhan.

Joshi (2022) menambahkan bahwa *Black-Box* Testing merupakan pendekatan komplementer terhadap *White-Box* Testing, dimana keduanya saling melengkapi untuk mengidentifikasi berbagai jenis kesalahan dalam perangkat lunak. *White-Box* Testing fokus pada struktur internal kode, sementara *Black-*

Box Testing fokus pada validasi *requirement* dan perilaku sistem dari perspektif pengguna akhir.

### **2.2.25.1 Teknik-Teknik Black-Box Testing**

Terdapat beberapa teknik yang umum digunakan dalam Black-Box Testing:

#### 1. Equivalence Partitioning (Pembagian Kelas Ekuivalen)

Teknik ini membagi input domain menjadi kelas-kelas data dimana setiap anggota kelas diharapkan menghasilkan hasil yang sama. Dengan memilih satu nilai representatif dari setiap kelas, penguji dapat mengurangi jumlah test case tanpa mengorbankan coverage (Khan & Khan, 2012).

#### 2. Boundary Value Analysis (Analisis Nilai Batas)

Teknik ini fokus pada nilai-nilai dibatas atau boundary dari domain input, karena kesalahan sering terjadi diarea batas (Pressman & Maxim, 2020).

#### 3. Decision Table Testing

Menggunakan tabel keputusan untuk menguji kombinasi input dan aturan bisnis yang menghasilkan output tertentu (Joshi, 2022).

#### 4. State Transition Testing

Menguji transisi antar state dalam sistem, berguna untuk aplikasi yang memiliki state berbeda berdasarkan kondisi tertentu (Sommerville, 2016).

## 5. Use Case Testing

Merancang test case berdasarkan use case atau skenario penggunaan sistem oleh user (Myers et al., 2011).

### 2.3 Gambaran Umum Objek Penelitian

#### 2.3.1 Profil dan Lokasi Sekolah

SMA Negeri 5 Medan adalah institusi pendidikan menengah atas berstatus Negeri yang berlokasi dipusat Kota Medan. Sekolah ini didirikan pada tahun 1985 dan saat ini memiliki Akreditasi A.

Identitas dan lokasi sekolah yang relevan dengan penelitian ini dirangkum dalam tabel berikut:

<b>Atribut</b>	<b>Keterangan</b>
Nama Sekolah	SMAN 5 MEDAN
NPSN	10210858
Alamat	Jl. Pelajar No. 17, Kel. Teladan Timur, Kec. Medan Kota
Kota/Provinsi	Kota Medan, Sumatera Utara
Akreditasi	A
Website	<a href="http://www.smanlimedan.sch.id">http://www.smanlimedan.sch.id</a>
Titik Koordinat	Latitude: 3.565200 Longitude: 98.698400

Tabel 2.5 Identitas Objek Penelitian

### 2.3.2 Visi, Misi, dan Sarana Pendukung

Visi dan misi sekolah memberikan konteks terhadap urgensi penelitian ini, terutama dalam hal penegakan kedisiplinan.

Visi:

"Menjadi sekolah menengah atas unggulan yang menghasilkan lulusan berkarakter, berakhlak mulia, dan siap menghadapi tantangan global dengan tetap menjunjung tinggi nilai-nilai luhur budaya bangsa."

Misi:

1. Menyelenggarakan pendidikan yang berkualitas dengan Kurikulum Merdeka
2. Mengembangkan potensi akademik dan non-akademik siswa
3. Membentuk karakter siswa yang jujur, disiplin, dan bertanggung jawab
4. Membangun lingkungan belajar yang kondusif dan inspiratif
5. Menjalin kerjasama dengan berbagai pihak untuk kemajuan Pendidikan

Dari sisi infrastruktur teknologi, SMA Negeri 5 Medan memiliki sarana pendukung yang memadai untuk implementasi sistem digital. Fasilitas yang relevan dengan penelitian ini mencakup beberapa laboratorium komputer, sumber listrik yang stabil dari PLN, dan jaringan WiFi dengan akses internet *fiber optic* yang tersedia diseluruh area sekolah. Ketersediaan infrastruktur ini mendukung kelayakan teknis penerapan sistem absensi berbasis *mobile* yang bergantung pada konektivitas jaringan.