

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk dalam dunia pendidikan. Transformasi digital mendorong institusi pendidikan untuk mengadopsi sistem berbasis teknologi guna meningkatkan efektivitas dan efisiensi dalam pengelolaan akademik. Salah satu aspek yang mengalami perubahan adalah sistem absensi mahasiswa, yang sebelumnya dilakukan secara manual dan kini mulai beralih ke sistem digital. Implementasi sistem absensi berbasis teknologi tidak hanya bertujuan untuk mempermudah pencatatan kehadiran, tetapi juga untuk meningkatkan akurasi serta mengurangi potensi kecurangan dalam proses presensi mahasiswa (Malah et al., 2022).

Di lingkungan Fakultas Teknik Universitas Islam Sumatera Utara (UISU), sistem absensi konvensional masih memiliki beberapa kendala, seperti kemungkinan adanya titip absen, ketidaktepatan pencatatan serta sulitnya verifikasi lokasi mahasiswa dalam perkuliahan *online*. Dengan semakin berkembangnya metode pembelajaran daring, dibutuhkan sistem absensi yang mampu memastikan bahwa mahasiswa yang melakukan presensi benar-benar berada pada lokasi yang ditentukan. Oleh karena itu, penerapan teknologi berbasis lokasi menjadi salah satu solusi yang dapat diimplementasikan dalam sistem absensi *online*.

Salah satu metode yang dapat digunakan untuk meningkatkan akurasi sistem absensi berbasis lokasi adalah algoritma *Haversine*. Algoritma ini digunakan

untuk menghitung jarak antara dua titik koordinat berdasarkan garis lintang dan bujur pada permukaan bumi. Dengan menerapkan algoritma *Haversine* dalam sistem absensi, validasi lokasi mahasiswa dapat dilakukan secara lebih akurat, sehingga mengurangi kemungkinan kecurangan, seperti mahasiswa melakukan presensi dari lokasi yang tidak sesuai dengan ketentuan (Antono, 2020).

Pengembangan sistem absensi berbasis *online* menjadi solusi dalam penggunaan teknologi yang lebih efektif dan efisien. Teknologi dalam aplikasi mobile salah satunya *flutter* adalah sebuah *framework open-source* atau SDK yang dikembangkan oleh *Google* untuk membangun antarmuka (*Customer interface/UI*) aplikasi yang memiliki kinerja tinggi serta dapat dipublikasi ke platform Android dan iOS dari *codebase* tunggal (Muslim et al, 2022).

Implementasi algoritma *Haversine* dalam sistem absensi perkuliahan *online* di Fakultas Teknik UISU diharapkan dapat memberikan solusi terhadap permasalahan yang ada. Dengan adanya sistem yang mampu mendeteksi lokasi pengguna secara akurat, proses presensi dapat dilakukan secara lebih efisien dan transparan. Selain itu, penerapan teknologi ini juga dapat mendukung kebijakan akademik dalam meningkatkan disiplin dan keterlibatan mahasiswa dalam perkuliahan daring.

Berdasarkan permasalahan tersebut, judul dari penelitian ini adalah **“Implementasi Algoritma *Haversine* Pada Absensi Perkuliahan Berbasis *Online* Di Fakultas Teknik Universitas Islam Sumatera Utara”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang, berikut ini merupakan uraian dari rumusan masalah yang diambil, diantaranya adalah sebagai berikut:

1. Bagaimana menerapkan algoritma *Haversine* dalam aplikasi dan sistem absensi perkuliahan di Universitas Islam Sumatera Utara secara optimal?
2. Bagaimana merancang aplikasi dan sistem absensi pada perkuliahan di kampus Universitas Islam Sumatera Utara secara efisien dalam pencatatan kehadiran?

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini agar menciptakan penelitian yang lebih terarah dan tidak terlalu meluas dari pembahasan. Berikut ini merupakan batasan masalah pada penelitian, diantaranya:

1. Proses absensi dilakukan melalui aplikasi android dan hanya dapat diakses oleh mahasiswa dan dosen Fakultas Teknik UISU.
2. Batasan jarak pengambilan absensi *online* ditentukan sejauh 300 (tiga ratus) meter dari titik koordinat Fakultas Teknik Universitas Islam Sumatera Utara.
3. Sistem dibangun berbasis web dengan bahasa pemrograman *PHP framework laravel* dan aplikasi android dibangun dengan bahasa pemrograman *dart framework flutter*.
4. Aplikasi hanya dapat di jalankan pada android versi 12 (dua belas) keatas.
5. Manajemen *database* menggunakan *MySQL* dengan *integrasi API*.

1.4 Tujuan Penelitian

Dalam rumusan yang telah ditentukan, maka didapatlah tujuan dari penelitian ini, diantaranya adalah:

1. Menerapkan algoritma *Haversine* pada aplikasi absensi perkuliahan *online* di Universitas Islam Sumatera Utara.
2. Merancang aplikasi serta sistem absensi perkuliahan berbasis *online* di Fakultas Teknik Universitas Islam Sumatera Utara.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi Penulis
 - a. Dengan adanya penelitian ini, dapat menambah ilmu serta wawasan yang lebih luas kepada penulis.
 - b. Sebagai salah satu syarat untuk memperoleh gelar S1 pada Program Studi Teknik Informatika Fakultas Teknik Universitas Islam Sumatera Utara.
2. Bagi Program Studi
 - a. Mendapatkan inovasi baru dari penelitian yang dilakukan oleh mahasiswa.
 - b. Dapat dijadikan referensi untuk penelitian selanjutnya.
3. Bagi Universitas Islam Sumatera Utara

Penelitian ini dapat menjadi referensi dan kontribusi bagi program studi dalam memperkaya bahan ajar terkait pengembangan sistem berbasis web dan teknologi informasi dalam dunia pendidikan. Selain itu, penelitian ini juga dapat menjadi bukti nyata kontribusi universitas dalam memecahkan

permasalahan di dunia pendidikan melalui penerapan teknologi yang tepat guna.

1.6 Sistematika Penulisan

Pada sistematika penulisan skripsi terbagi menjadi beberapa bab dan beberapa sub untuk mempermudah penjelasan mengenai penelitian yang dilakukan kepada pembaca. Berikut ini merupakan sistematika penulisan diantaranya adalah:

BAB I : PENDAHULUAN

Pada bab ini, berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian dan sistematika penulisan pada penelitian ini.

BAB II : TINJAUAN PUSTAKA

Pada bab ini berisi tentang teori yang dan referensi dari para ahli yang menjadi informasi pendukung pada penelitian ini.

BAB III : METODE PENELITIAN

Pada bab ini menjelaskan tentang objek dan metode yang digunakan dalam penelitian, sehingga penelitian lebih terstruktur saat pengerjaan dilakukan.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini merupakan pemaparan hasil dari penelitian serta pembahasan terkait penelitian yang telah dilakukan serta implementasinya.

BAB V : KESIMPULAN DAN SARAN

Bab ini merupakan akhir dari penelitian dimana keseluruhan isi dan capaian yang didapat diuraikan pada kesimpulan dan saran yang dapat diajukan untuk penelitian lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Algoritma Secara Umum

Algoritma merupakan suatu urutan langkah-langkah logis yang disusun secara sistematis untuk menyelesaikan suatu permasalahan. Dalam dunia komputasi, algoritma menjadi fondasi penting karena setiap program komputer pada dasarnya merupakan implementasi dari satu atau lebih algoritma. Definisi umum dari algoritma adalah instruksi atau prosedur yang terbatas dan pasti untuk memecahkan masalah tertentu. Menurut Knuth (1973), algoritma adalah suatu metode untuk menyelesaikan masalah melalui urutan langkah yang terbatas dan spesifik (Suhartini et al., 2023).

Selain itu, algoritma memiliki karakteristik penting, yaitu memiliki *input*, *output*, instruksi yang jelas, bersifat deterministik dan harus berakhir dalam waktu tertentu (*finiteness*). Dalam penerapannya, algoritma harus disusun seefisien mungkin, baik dari sisi waktu eksekusi maupun penggunaan sumber daya komputer. Oleh karena itu, pengembangan algoritma menjadi bidang penting dalam ilmu komputer, matematika, bahkan di banyak bidang teknik lainnya.

2.1.1 Jenis-Jenis Algoritma

Banyak jenis algoritma telah dikembangkan sesuai dengan kebutuhan permasalahan di berbagai bidang. Misalnya, dalam bidang pencarian data dikenal algoritma pencarian seperti *Linear Search* dan *Binary Search*, dalam pengurutan data terdapat algoritma *Bubble Sort*, *Quick Sort* dan *Merge Sort*. Dalam pengolahan

data geografis terdapat algoritma untuk menghitung jarak seperti algoritma *Vincenty* dan *Haversine*. Masing-masing algoritma dirancang untuk menyelesaikan masalah secara efektif dan efisien sesuai karakteristik datanya.

Dalam konteks penghitungan jarak antar titik di permukaan bumi, algoritma *Haversine* menjadi salah satu algoritma yang banyak digunakan. Hal ini karena algoritma *Haversine* mempertimbangkan bentuk bumi yang bulat, sehingga perhitungan jaraknya menjadi lebih akurat dibandingkan hanya menggunakan jarak lurus atau rumus Euclidean biasa (Wirastuti et al., 2023). Oleh karena itu, algoritma ini banyak diaplikasikan dalam bidang navigasi, sistem informasi geografis (GIS) dan pengembangan aplikasi berbasis lokasi.

2.1.2 Algoritma *Haversine*

Algoritma *Haversine*, yang pertama kali dipopulerkan oleh ahli geodesi James Andrew pada abad ke-19, merupakan metode perhitungan jarak antara dua titik berdasarkan garis lintang (*latitude*) dan garis bujur (*longitude*) sebagai variabel dalam perhitungannya di permukaan bumi. Algoritma ini mempertimbangkan kelengkungan bumi, sehingga menghasilkan perhitungan jarak yang lebih akurat. Algoritma *Haversine* telah banyak digunakan dalam berbagai aplikasi berbasis lokasi, seperti sistem pencarian fasilitas publik dan navigasi berbasis peta. Studi terbaru menunjukkan bahwa penggunaan algoritma *Haversine* dalam sistem geolokasi memiliki tingkat akurasi yang tinggi, dengan deviasi perhitungan jarak yang sangat kecil, sekitar 0,003 km hingga 0,005 km (Antono, 2020).

Dalam konteks sistem absensi berbasis lokasi, algoritma *Haversine* dapat membantu memastikan mahasiswa yang melakukan presensi berada dalam radius

tertentu dari lokasi yang telah ditentukan. Berikut ini merupakan rumus dari algoritma *Haversine*:

$$\begin{aligned}
 x &= (long2 - long1) \times \cos\left(\frac{lat1 + lat2}{2}\right) \\
 y &= (lat2 - lat1) \\
 d &= \sqrt{(x \times x) + (y \times y)} \times R \quad \dots\dots\dots 2.1)
 \end{aligned}$$

Keterangan simbol:

lat1 = *Latitude* awal

lat2 = *Latitude* tujuan

long1 = *Longitude* awal

long2 = *Longitude* tujuan

x = *Longitude* x (bujur)

y = *Latitude* y (lintang)

d = Jarak

R = Radius bumi (6371 km)

Pada praktek dalam pemrograman, khususnya bahasa pemrograman yang akan digunakan yaitu *PHP* (*Perl: Hypertext Preprocessor*), algoritma *Haversine* dapat diterapkan dengan *source code* berikut ini:

```

public function calculateDistance(Request $request){
    // Validasi input
    $validated = $request->validate([
        'lat1' => 'required|numeric',

```

```
'lon1' => 'required|numeric',
'lat2' => 'required|numeric',
'lon2' => 'required|numeric',
]);

// Ambil input
$lat1 = $validated['lat1'];
$lon1 = $validated['lon1'];
$lat2 = $validated['lat2'];
$lon2 = $validated['lon2'];

// Hitung jarak
$distance = $this->haversineDistance($lat1, $lon1, $lat2, $lon2);

// Return response
return response()->json([
    'distance_km' => round($distance, 2) . ' km',
]);
}

private function haversineDistance($lat1, $lon1, $lat2, $lon2){
    $earthRadius = 6371; // Radius bumi dalam kilometer

    // Konversi derajat ke radian
```

```

$lat1 = deg2rad($lat1);
$lon1 = deg2rad($lon1);
$lat2 = deg2rad($lat2);
$lon2 = deg2rad($lon2);

// Menghitung perbedaan koordinat
$deltaLat = $lat2 - $lat1;
$deltaLon = $lon2 - $lon1;

// Rumus haversine
$a = sin($deltaLat / 2) ** 2 +
    cos($lat1) * cos($lat2) *
    sin($deltaLon / 2) ** 2;
$c = 2 * atan2(sqrt($a), sqrt(1 - $a));
$distance = $earthRadius * $c;

return $distance;
}

```

2.2 *Global Positioning System (GPS)*

Global Positioning System (GPS) adalah sistem navigasi berbasis satelit yang digunakan untuk menentukan lokasi suatu objek di permukaan bumi dengan tingkat akurasi yang tinggi. Sistem ini dikembangkan oleh Departemen Pertahanan Amerika Serikat dan mulai digunakan secara global sejak tahun 1995. *GPS* bekerja dengan memanfaatkan sinyal dari minimal empat satelit yang mengorbit bumi

untuk menghitung posisi pengguna dalam koordinat *lintang*, bujur dan ketinggian (El-Rabbany, 2020).

Aplikasi berbasis *GPS* memungkinkan sistem untuk menentukan apakah suatu perangkat berada dalam radius tertentu dari lokasi yang ditentukan. Dalam sistem absensi berbasis lokasi, *GPS* digunakan untuk memastikan kehadiran pengguna secara valid berdasarkan posisi geografis mereka.

2.3 Website

World Wide Web (WWW) atau yang biasa dikenal dengan istilah *website*, merupakan layanan yang terdapat informasi ataupun dokumen berupa media seperti, teks, gambar, suara, animasi, video dan lain-lainnya. Halaman pada *website* memiliki keterkaitan satu sama lain yang biasa ditempatkan pada sebuah server, yang dapat diakses melalui *internet* ataupun *Local Area Network* (LAN) (Parinsi et al., 2021).

Website juga merupakan kumpulan halaman yang dapat diakses melalui *internet* menggunakan *browser* yang dibangun dengan teknologi seperti HTML, CSS dan JavaScript serta dapat bersifat statis atau dinamis tergantung pada fungsinya. Dalam pengembangannya, *website* sering menggunakan *framework* dan *Content Management System* (CMS) untuk meningkatkan efisiensi dan kemudahan pengelolaan konten.

2.4 Web Service

Web service atau layanan web adalah sebuah sistem komunikasi pertukaran data antar aplikasi berbeda melalui jaringan. *Web service* menggunakan standar

layanan yang diberikan untuk tindakan yang melibatkan HTTP, XML atau JSON untuk kegunaan suatu teknologi berbasis *website*. Teknologi ini banyak digunakan diberbagai aplikasi seperti layanan *e-commerce*, perbankan dan sistem berbasis lokasi lainnya yang dapat meng*integrasikan* data dari berbagai sumber tanpa harus memahami detail implementasi *internalnya* (Halim et al., 2021).

Penerapan *web service* diantaranya memiliki beberapa layanan salah satunya adalah *API (Application Programming Interface)*. Dalam pengembangan suatu sistem *API* dapat digunakan untuk mempermudah dan mempercepat proses pengembangan.

2.4.1 *Application Programming Interface (API)*

Application Programming Interface (API) adalah sekumpulan aturan dan protokol yang memungkinkan aplikasi untuk berkomunikasi dengan sistem lain. *API* bertindak sebagai perantara yang memungkinkan pengembang untuk mengakses fungsi tertentu tanpa harus mengetahui bagaimana sistem *internalnya* bekerja. REST *API* menjadi salah satu yang paling populer karena kemudahan penggunaannya dan kompatibilitasnya dengan protokol HTTP (Fielding & Taylor, 2022).

Dalam konteks pengembangan aplikasi berbasis *web* dan *mobile*, *API* sering digunakan untuk menghubungkan aplikasi dengan layanan eksternal, seperti *database*, autentikasi pengguna atau sistem pemetaan seperti *Google Maps API*.

2.4.2 *Google Maps API*

Google Maps API merupakan layanan yang diberikan oleh *Google* secara gratis yang sangat populer untuk mengintegrasikan peta, pencarian lokasi dan navigasi kedalam aplikasi. Peta yang ada di dunia dapat dilihat dengan mudah hanya dengan mengaksesnya saja melalui *browser*. *API* ini menyediakan berbagai fitur, termasuk pemetaan *interaktif*, perhitungan rute, *geocoding* (mengubah alamat menjadi koordinat geografis) dan penentuan lokasi pengguna.

Google Maps API berbasis REST dan JavaScript, sehingga dapat dengan mudah digunakan dalam berbagai platform, baik *web* maupun *mobile*. Dalam aplikasi berbasis lokasi, seperti sistem absensi berbasis *GPS*, *Google Maps API* digunakan untuk memvalidasi posisi pengguna berdasarkan koordinat yang dikirimkan oleh perangkat mereka (Malah et al., 2022).

2.5 **Android**

Android adalah sistem operasi berbasis Linux yang dikembangkan oleh *Google* untuk perangkat *mobile* seperti *smartphone* dan *tablet*. Versi android yang digunakan dalam penelitian ini adalah Android 11, yang membawa berbagai peningkatan dalam keamanan, privasi dan pengelolaan aplikasi. Android 11 memperkenalkan fitur seperti izin akses satu kali, manajemen notifikasi yang lebih baik, serta peningkatan dalam performa aplikasi latar belakang (Darmansah et al., 2020).

2.6 Absensi

Absensi merupakan proses pencatatan kehadiran seseorang dalam suatu kegiatan, baik dalam lingkungan pendidikan maupun dunia kerja. Absensi banyak digunakan untuk mengukur kedisiplinan, partisipasi, dan keterlibatan individu, seperti dalam absensi karyawan, mahasiswa, maupun siswa sekolah. Seiring perkembangan teknologi, sistem absensi juga mengalami inovasi, dari yang semula berbasis manual menggunakan kertas, kemudian berkembang menggunakan perangkat *finger print*, program komputer, hingga teknologi biometrik seperti pemindaian retina atau wajah (Panggabean et al., 2023). Menurut penelitian oleh Setiawan et al. (2022), penerapan sistem absensi berbasis teknologi informasi mampu meningkatkan keakuratan data kehadiran hingga 95% dibandingkan sistem manual.

Dalam konteks pendidikan tinggi, khususnya pada Fakultas Teknik Universitas Islam Sumatera Utara (UISU), sistem absensi masih dilakukan secara manual, yaitu mahasiswa menandatangani daftar hadir pada setiap sesi perkuliahan. Sistem ini cukup efektif dalam kondisi normal, namun memiliki beberapa kelemahan, seperti risiko manipulasi data kehadiran, kehilangan daftar hadir dan ketidakakuratan *rekapitulasi*. Oleh karena itu, terdapat kebutuhan mendesak untuk mengembangkan sistem absensi berbasis teknologi, seperti absensi berbasis lokasi geografis atau QR code, yang dapat mengurangi kecurangan serta meningkatkan efisiensi administrasi. Upaya pengembangan ini sejalan dengan penelitian yang dilakukan oleh Rachman dan Dewi (2023), yang menyatakan bahwa sistem absensi berbasis lokasi mampu mengurangi potensi kecurangan hingga 40% (Rachman & Dewi, 2023).

Absensi dalam kegiatan perkuliahan sendiri memiliki peranan penting sebagai salah satu indikator untuk memberikan motivasi dan tanggung jawab terhadap kehadiran mahasiswa dalam proses belajar mengajar. Kehadiran yang tercatat dengan baik juga menjadi salah satu syarat administratif untuk mengikuti ujian, sehingga sistem absensi yang efektif berkontribusi langsung terhadap kelancaran proses akademik mahasiswa.

2.7 Unified Modeling Language (UML)


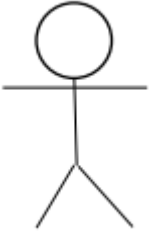




Unified Modeling Language (UML) adalah salah satu standar dalam penentuan kebutuhan (*requirement*), analisis (*analysis*) dan desain (*design*). *UML* merupakan metodologi untuk pengembangan sistem yang dapat mendukung proses pengerjaannya (Amin et al., 2024).

Dalam pengembangan menggunakan *UML* berikut ini terdapat juga beberapa metode pemodelan diantaranya adalah sebagai berikut.

2.7.1 Usecase Diagram

Usecase diagram berfungsi untuk melakukan pekerjaan tertentu yang menggambarkan bisnis proses sistem itu sendiri, *usecase diagram* yang akan digunakan untuk menjelaskan fitur yang dapat digunakan oleh admin (Prasetya et al., 2022). Berikut adalah simbol-simbol yang terdapat pada *usecase diagram* dapat dilihat pada tabel 2.1.





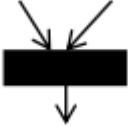

Tabel 2.1 Simbol *Usecase Diagram*
(Sumber: Prasetya et al., 2022)

Gambar	Nama	Keterangan
	<i>Usecase</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dengan aktor yang dinyatakan.
	<i>Actor</i>	Merupakan abstraksi dari <i>user</i> sistem yang mengoperasikan fungsi dari beberapa modul sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian peran dan tugas-tugas yang sesuai konteks target sistem, tetapi tidak memiliki kontrol penuh terhadap <i>usecase</i> .
	<i>Association</i>	Asosiasi antara aktor dan <i>usecase</i> , digambarkan dengan garis. Tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	<i>Association</i>	Asosiasi antara aktor dan <i>usecase</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i>	<i>Usecase</i> oleh <i>usecase</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i>	Merupakan perluasan dari <i>usecase</i> lain jika kondisi atau syarat terpenuhi.


2.7.2 Activity Diagram

Activity diagram adalah aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas (Prasetya et al., 2022). Berikut adalah simbol-simbol yang digunakan dalam *activity diagram*, dapat dilihat pada tabel 2.2.

Tabel 2.2 Simbol *Activity Diagram*
(Sumber: Prasetya et al., 2022)

Gambar	Nama	Keterangan
	<i>Start Point</i>	merupakan simbol yang diletakkan pada pojok kiri atas dan merupakan awal suatu aktivitas.
	<i>End Point</i>	merupakan simbol yang diletakkan pada pojok bawah dan merupakan berakhirnya suatu aktivitas.
	<i>Activities</i>	menggambarkan suatu proses / kegiatan.
	<i>Fork</i>	percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i>	(penggabungan) atau <i>join</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Point</i>	menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .



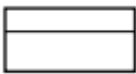


Tabel 2.2 Simbol *Activity Diagram* (lanjutan)

Gambar	Nama	Keterangan
	<i>Swimlane</i>	pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

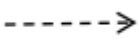

2.7.3 Class Diagram

Adapun *class diagram* yang terdapat pada aplikasi, dimana merupakan gambaran dari struktur relasi pada *database* aplikasi (Prasetya et al., 2022). Berikut ini merupakan simbol-simbol yang terdapat pada *class diagram*, dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol *Class Diagram*
(Sumber: Prasetya et al., 2022)

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek (ancestor)
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari urutan objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

Tabel 2.3 Simbol *Class Diagram* (lanjutan)

Gambar	Nama	Keterangan
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>).
	<i>Association</i>	Menghubungkan antara objek satu dan objek lainnya.

2.8 Aplikasi Pengembangan Sistem

Pada penelitian ini akan dibangun beberapa alat yang digunakan untuk membuat sistem, oleh karena itu beberapa alat tersebut telah disitasi dalam beberapa penelitian. Diantaranya adalah sebagai berikut.

2.8.1 WAMP Server

WAMP Server merupakan paket server bahasa independen yang berjalan secara *local* dan diinstal di perangkat sistem operasi windows. *WAMP* sendiri merupakan suatu singkatan dari *windows and the principal components of the package: Apache, MySQL and PHP (or Perl or Python)*.

Secara umum kita mengetahui bahwa Apache merupakan suatu *web server*, *MySQL* merupakan *database* dan *PHP* merupakan bahasa yang digunakan dalam *scripting* programnya. Dalam *package server WAMP* ini sudah komplit dimana dapat langsung menciptakan *web* yang dinamis dari *database* serta konten yang di *request* dari *browser* (Jony et al., 2022).

2.8.2 *PHP (Perl Hypertext Preprocessor)*

Perl *Hypertext Preprocessor* atau *PHP* adalah bahasa pemrograman *side server script* yang dirancang lebih cenderung kepada proses pembuatan dan pengembangan *website*. *PHP* memiliki kelebihan yang mudah dipelajari, *open-source*, ringkas dan memiliki komunitasnya yang besar. Namun, disamping kelebihannya yang membuat bahasa pemrograman *PHP* banyak diminati terdapat juga beberapa kekurangannya yakni, mudah dibajak, tidak dilengkapi tipe data dan persaingan ketat, sehingga membuat para pengembang wajib menambahkan bahasa pemrograman lain yang bisa menutupi kekurangan dari bahasa *PHP* itu sendiri kedalam sistem yang akan dibangun (Parinsi et al., 2021).

2.8.3 *Framework Laravel*

Laravel adalah *framework PHP* yang bersifat *open-source* dan dirancang untuk mempermudah pengembangan aplikasi web dengan *sintaks* yang ekspresif dan elegan. *Laravel* menggunakan arsitektur *Model-View-Controller* (MVC) yang membantu dalam pemisahan logika bisnis dan tampilan, sehingga membuat kode lebih terstruktur dan mudah dipelihara. Beberapa fitur utama *Laravel* meliputi *Eloquent ORM* untuk manajemen *database*, sistem *routing* yang fleksibel, serta dukungan bawaan untuk autentikasi dan keamanan aplikasi (Pernando, 2021).

2.8.4 *Dart*

Dart adalah bahasa pemrograman yang dikembangkan oleh *Google* dan dirancang untuk pengembangan aplikasi *multiplatform*, termasuk *web*, *mobile* dan *desktop*. *Dart* memiliki *sintaks* yang mirip dengan bahasa pemrograman seperti

Java dan JavaScript, serta menggunakan paradigma berbasis objek dengan *garbage collection* otomatis. Bahasa ini digunakan secara luas dalam pengembangan aplikasi menggunakan *framework Flutter*, karena memiliki performa tinggi dengan kompilasi langsung ke kode mesin (Afergan et al., 2022).

2.8.5 Framework Flutter

Flutter adalah *framework* UI yang dikembangkan oleh *Google* untuk membangun aplikasi *multiplatform* dari satu basis kode. Menggunakan bahasa *Dart*, *Flutter* memungkinkan pengembang membuat aplikasi dengan antarmuka yang menarik dan responsif untuk *Android*, *iOS*, *web*, serta *desktop*. Salah satu keunggulan utama *Flutter* adalah fitur *hot reload*, yang memungkinkan pengembang melihat perubahan kode secara instan tanpa harus menunggu kompilasi ulang. Selain itu, *Flutter* menggunakan *widget* yang dapat dikustomisasi, sehingga memberikan fleksibilitas tinggi dalam pembuatan tampilan aplikasi (Krisnantoro, 2023).

2.8.6 MySQL

MySQL adalah sistem yang digunakan dalam pengaturan struktur dalam *database*, baik dalam proses pengelolaan maupun proses pembuatan. *MySQL* juga merupakan singkatan dari *My Structure Query Language*, bahasa ini sangat digemari oleh kalangan *programmer web* dikarenakan bahwa program ini merupakan media *database* yang cukup stabil (Agustin, 2020).

2.8.7 *Visual Studio Code*

Visual Studio Code atau lebih sering disebut VSCode merupakan suatu software pengelola teks yang biasa digunakan untuk *editor* kode. *Software* ini merupakan rancangan dari perusahaan Microsoft yang dapat digunakan *multiplatform*, artinya tersedia juga untuk sistem operasi Linux, Mac dan Windows (Syarif et al., 2023).

Text editor VSCode ini bersifat *open source*, dimana kode sumbernya dapat dilihat dan berkontribusi untuk pengembangannya. Hal ini juga yang membuat VSCode menjadi favorit para pengembang *software*, karena dapat ikut serta dalam perkembangan VSCode kedepannya.

2.9 Universitas Islam Sumatera Utara

Universitas Islam Sumatera Utara (UISU) adalah salah satu perguruan tinggi swasta tertua di Sumatera Utara yang berdiri pada tahun 1951. UISU memiliki misi untuk menyelenggarakan pendidikan tinggi yang berlandaskan nilai-nilai keislaman, berwawasan global, serta mampu bersaing di tingkat nasional dan *internasional*. Saat ini UISU menaungi berbagai fakultas mulai dari Fakultas Hukum, Ekonomi, Pertanian, hingga Teknik.

Sebagai salah satu universitas yang memiliki sejarah panjang dalam dunia pendidikan di Indonesia, UISU berkomitmen untuk terus mengembangkan sumber daya manusia yang beriman, berilmu, dan beramal. Dengan moto "Berilmu Amaliah, Beramal Ilmiah, dan Berakhlaqul Karimah," UISU telah menghasilkan ribuan alumni yang tersebar di berbagai bidang profesional di seluruh Indonesia.

2.9.1 Fakultas Teknik UISU

Fakultas Teknik Universitas Islam Sumatera Utara (FT-UISU) merupakan salah satu fakultas yang berada di lingkungan UISU dengan berbagai program studi unggulan di bidang keteknikan. Fakultas ini berdiri dengan tujuan menghasilkan lulusan yang kompeten, profesional, dan berakhlak mulia dalam bidang teknik. Program studi yang tersedia antara lain Teknik Sipil, Teknik Mesin, dan Teknik Elektro, yang telah terakreditasi oleh Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT).

Fakultas Teknik UISU juga aktif dalam kegiatan penelitian, pengabdian kepada masyarakat, dan pengembangan inovasi teknologi di Sumatera Utara. Dengan visi menjadi fakultas teknik yang unggul berbasis nilai-nilai Islam, FT-UISU terus berupaya meningkatkan mutu pendidikan, fasilitas laboratorium, serta kerja sama dengan dunia industri untuk mendukung pengembangan keterampilan mahasiswa.

2.9.2 Absensi Fakultas Teknik UISU

Di Fakultas Teknik Universitas Islam Sumatera Utara (UISU), sistem absensi masih dilakukan secara konvensional, yaitu dengan mengisi daftar hadir secara manual pada setiap pertemuan kelas. Cara ini memiliki beberapa kekurangan, seperti risiko kehilangan data absensi, keterlambatan *rekapitulasi*, serta potensi manipulasi data. Oleh karena itu, terdapat kebutuhan untuk mengembangkan sistem absensi yang lebih modern dan akurat, terutama dengan memanfaatkan teknologi berbasis lokasi.

Ketersediaan infrastruktur seperti jaringan *internet* yang stabil, perangkat *smartphone*, serta *integrasi* dengan sistem akademik berbasis teknologi informasi, membuat Fakultas Teknik UISU berpotensi untuk menerapkan sistem absensi berbasis aplikasi. Sistem ini diharapkan tidak hanya meningkatkan akurasi pencatatan kehadiran, *tetapi* juga memudahkan dosen dan staf administrasi dalam monitoring kehadiran mahasiswa secara *real-time*.

2.10 Penelitian Terkait

Dalam penelitian ini, membutuhkan beberapa penelitian terkait sebagai sumber informasi lanjutan dan acuan dalam Implementasi Algoritma *Haversine* Pada Absensi Perkuliahan *Online* di Fakultas Teknik UISU. Berikut merupakan beberapa penelitian terkait dengan penelitian saat ini, diantaranya:

1. Penelitian ini dilakukan oleh Fajar Antono dan Saruni Dewi pada tahun 2022 dengan judul Implementasi Absensi Karyawan Menggunakan Algoritma *Haversine* dengan Global Positioning System Berbasis Android. Kesimpulan yang diperoleh dari penelitian ini yaitu, penerapan *Haversine Formula* mampu memberikan informasi jarak secara akurat antara lokasi pengguna dengan lokasi objek yang dimaksud, seperti kantor. Perhitungan jarak yang dihasilkan oleh sistem menggunakan rumus ini menunjukkan perbedaan yang sangat kecil dibandingkan dengan perhitungan manual, yakni sekitar $\pm 0,0018$ meter. Hal ini menunjukkan bahwa *Haversine Formula* memiliki tingkat akurasi yang tinggi dan dapat diandalkan. Selain itu, rumus ini juga dapat dimanfaatkan dalam sistem pencarian jarak terdekat, yaitu

dengan cara menentukan nilai jarak terkecil sebagai lokasi terdekat dari pengguna.

2. Penelitian ini dilakukan oleh Hidayatul Sidiq, Deni Satria dan Humaira pada tahun 2022 dengan judul Implementasi Algoritma *Haversine* Pada Absensi Kepegawaian Berbasis Android. Kesimpulan yang diperoleh dari penelitian ini yaitu, implementasi algoritma *Haversine* pada aplikasi absensi kepegawaian berbasis Android, diperoleh beberapa kesimpulan penting. Aplikasi yang dikembangkan terbukti mampu mempermudah proses absensi pegawai, pengajuan cuti, peninjauan riwayat absensi, akses informasi sekolah, serta pengelolaan data pengguna yang ada di lingkungan Sekolah Dasar Al-Azhar. Dari sisi admin, sistem memungkinkan pengelolaan data kepegawaian secara menyeluruh, termasuk verifikasi cuti, pengelolaan data pengguna, serta pembuatan laporan absensi setiap pegawai. Penerapan algoritma *Haversine* dalam aplikasi ini digunakan untuk memastikan keakuratan lokasi saat absensi dilakukan, sehingga dapat mencegah terjadinya kecurangan. Absensi hanya dapat diambil jika posisi pengguna berada dalam radius maksimal 20 (dua puluh) meter dari lokasi sekolah, yang ditentukan sebagai batas validasi lokasi absensi.
3. Penelitian ini dilakukan oleh Muhammad Fauzan, Efrans Christian dan Putu Bagus A. A. P. pada tahun 2024 dengan judul Rancang Bangun Aplikasi Presensi Menggunakan Algoritma *Haversine* (Studi Kasus Badan Kepegawaian Daerah Provinsi Kalimantan Tengah) Berbasis Android. Kesimpulan yang diperoleh dari penelitian ini yaitu, proses pengumpulan data dilakukan melalui observasi, wawancara, dan konsultasi, sedangkan

perancangan sistem dilakukan menggunakan *flowchart* dan dimodelkan dengan *Unified Modeling Language (UML)*, seperti *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Aplikasi dikembangkan menggunakan bahasa pemrograman *PHP* dengan *framework Laravel*, *CSS*, dan *MySQL* untuk bagian web (*backend* dan *frontend*), serta bahasa *Dart* dengan *framework Flutter* untuk antarmuka aplikasi Android. Aplikasi ini menyediakan alternatif dalam pencatatan dan pengelolaan data kehadiran pegawai dengan menambahkan batas geografis virtual yang ditentukan berdasarkan koordinat *latitude* dan *longitude*, sehingga presensi hanya dapat dilakukan jika posisi pegawai berada dalam radius yang ditentukan dari kantor. Berdasarkan hasil *blackbox testing*, aplikasi ini dinyatakan berjalan dengan baik dan sesuai dengan fungsionalitas yang dirancang.