

**HASIL**

**IMPLEMENTASI TEKNOLOGI *LOCATION BASED SERVICE (LBS)*  
DAN METODE *HAVERSINE FORMULA* PADA SISTEM MONITORING  
KEHADIRAN SISWA SECARA *REAL TIME*  
(STUDI KASUS : SMAN 1 WONGGEDUKU)**

**Diajukan Untuk Memenuhi  
Salah Satu Syarat Memperoleh Derajat Sarjana Teknik**



**CHINDY ELFIANA SUCI NATASIYA  
E1E118067**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS HALU OLEO  
KENDARI  
2022**

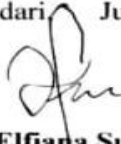
## LEMBAR PENGESAHAN

Hasil

**IMPLEMENTASI TEKNOLOGI *LOCATION BASED SERVICE (LBS)* DAN  
METODE *HAVERSINE FORMULA* PADA SISTEM MONITORING  
KEHADIRAN SISWA SECARA *REAL TIME*  
(STUDI KASUS : SMAN 1 WONGGEDUKU)**

Adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Universitas Halu Oleo ataupun institusi pendidikan lainnya.

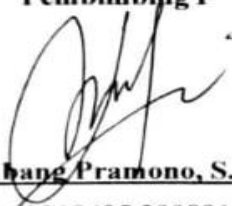
Kendari, Juli 2022



**Chindy Elfiana Suci Natasiya**

**E1E1 18 067**

**Pembimbing I**



**Bambang Pramono, S.Si., MT**

**NIP. 19710425 200801 1 010**

**Pembimbing II**



**Adha Mashur Saijah, ST., MT**

**NIP. 19910623 201803 1 001**

**Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Teknik Universitas Halu Oleo**

**Sutardi, S.Kom., MT**

**NIP. 19760222 201012 1 001**

## INTISARI

Chindy Elfiana Suci Natasiya, E1E118067

### **IMPLEMENTASI TEKNOLOGI *LOCATION BASED SERVICE (LBS)* DAN METODE *HAVERSINE FORMULA* PADA SISTEM MONITORING KEHADIRAN SISWA SECARA *REAL TIME***

Skripsi, Fakultas Teknik, 2022

Kata Kunci: Absensi, *Location Based Service*, *Haversine Formula*, Android

Absensi dapat dikatakan suatu pendataan kehadiran yang merupakan bagian dari aktivitas pelaporan yang ada dalam sebuah institusi. Sejalan dengan berkembangnya teknologi, sistem absensi dalam dunia pendidikan umumnya masih dilakukan secara manual. Oleh karena itu peneliti memanfaatkan teknologi location based service sebagai sistem absensi dan menerapkan metode haversine formula untuk menghitung jarak antara titik lokasi kelas dan perangkat seluler. Dengan penerapan konsep tersebut juga diharapkan sistem ini membantu guru bimbingan konseling (BK) untuk memonitoring kehadiran siswa/i disetiap mata pelajaran apabila siswa/i tidak melakukan absensi pada jam pelajaran dimulai hingga mata pelajaran selesai maka siswa tersebut akan dianggap tidak hadir. Hasil yang didapatkan pada penelitian ini bahwa metode *haversine formula* sebagai penghitung jarak antara siswa dan titik kelas untuk absensi berhasil diimplementasikan. Presentasi perhitungan jarak menggunakan metode *haversine formula* pada sistem maupun secara manual tidak jauh berbeda, presentasi menunjukkan 98,98% akurat.

## **ABSTRACT**

Chindy Elfiana Suci Natasiya, E1E118067

### ***IMPLEMENTATION OF LOCATION BASED SERVICE (LBS) TECHNOLOGY AND HAVERSINE FORMULA METHOD IN REAL TIME STUDENT ATTENDANCE MONITORING SYSTEM***

*Thesis, Faculty of Engineering, 2022*

*Keywords: Attendance, Location Based Service, Haversine Formula, Android*

*Attendance can be said to be an attendance data collection which is part of the reporting activities that exist within an institution. In line with the development of technology, the attendance system in education is generally still done manually. Therefore, researchers use location based service technology as an attendance system and apply the haversine formula method to calculate the distance between class location points and mobile devices. With the application of this concept, it is also hoped that this system will help guidance and counseling teachers to monitor student attendance in each subject. The results obtained in this study that the haversine formula method as a calculation of the distance between students and class points for absenteeism was successfully implemented. The presentation of distance calculation using the haversine formula method on the system or manually is not much different, the presentation shows that it is 98.98% accurate.*

## DAFTAR ISI

LEMBAR PENGESAHAN .....	<b>Error! Bookmark not defined.</b>
INTISARI .....	ii
ABSTRACT .....	iii
DAFTAR ISI .....	iv
DAFTAR GAMBAR.....	vi
DAFTAR TABEL.....	viii
BAB I PENDAHULUAN.....	1
BAB II LANDASAN TEORI.....	7
2.2 <i>Location Based Service</i> (LBS).....	10
2.3 <i>Haversine Formula</i> .....	11
2.4     PHP .....	13
2.5 <i>Hypertext Markup Language</i> (HTML) .....	14
2.6 <i>Cascading Style Sheet</i> (CSS) .....	14
2.7 <b>Android</b> .....	15
2.8     MySQL.....	16
2.9     Flutter .....	18
2.10    Dart.....	19
2.11 <i>Rational Unified Process</i> (RUP).....	21
2.12 <i>Unified Modeling Language</i> (UML).....	23
2.13    Diagram Dalam UML.....	23
BAB III METODOLOGI PENELITIAN .....	28
3.1.    Waktu dan Tempat Penelitian .....	28
3.1.1.    Waktu.....	28
3.1.2.    Tempat Penelitian .....	28
3.2.    Metode Pengumpulan Data.....	28
3.3.    Metode Pengembangan Sistem .....	29
3.3.1.    Permulaan ( <i>Inception</i> ).....	29
3.3.2.    Perluasaan / Perencanaan ( <i>Elaboration</i> ).....	29
3.3.3.    Konstruksi ( <i>Construction</i> ) .....	29
3.3.4.    Transisi ( <i>Transition</i> ) .....	29
BAB IV ANALISIS DAN PERANCANGAN SISTEM.....	30
4.1. <b>Analisis Kebutuhan Sistem</b> .....	30

4.1.1.	Kebutuhan Fungsional .....	30
4.1.2.	Kebutuhan <i>Nonfungsional</i> .....	31
4.2.	Analisis Perancangan Sistem .....	32
4.2.1.	Arsitektur Sistem .....	32
4.2.2.	Ilustrasi Algoritma .....	32
4.2.3.	<i>Unified Modeling Language</i> (UML).....	33
4.2.4.	Perancangan Antarmuka <i>Interface</i> .....	54
BAB V	IMPLEMENTASI DAN PEMBAHASAN .....	63
5.1	Implementasi Sistem.....	63
5.2	Data.....	63
5.3	Implementasi <i>Interface</i> .....	63
5.3.1	Implementasi <i>Interface System</i> Berbasis Web .....	63
5.3.2	Implementasi <i>Interface System</i> Berbasis <i>Mobile</i> .....	69
5.5	<b>Pengujian</b> Sistem .....	71
5.5.1	Pengujian <i>Black Box</i> .....	72
5.5.2	Pengujian Hitungan Manual .....	73
BAB VI	PENUTUP .....	76
6.1	Kesimpulan .....	76
6.2	Saran .....	76
DAFTAR PUSTAKA	.....	77

## DAFTAR GAMBAR

Gambar 2. 1 <i>Source Code Variable Dan Tipe</i> .....	19
Gambar 2. 2 <i>Source Code Input Output</i> .....	20
Gambar 2. 3 <i>Source Code Conditional</i> .....	20
Gambar 2. 4 <i>Source Code Loop</i> .....	21
Gambar 2. 5 <i>Arsitektur Rational Unified Process (Hartawan, 2017)</i> .....	22
Gambar 4. 1 <i>Arsitektur Sistem</i> .....	32
Gambar 4. 2 <i>Ilustrasi Algoritma</i> .....	33
Gambar 4. 3 <i>Use Case Diagram</i> .....	34
Gambar 4. 4 <i>Activity Diagram Login (User)</i> .....	44
Gambar 4. 5 <i>Activity Diagram Tampil Data Kelas</i> .....	45
Gambar 4. 6 <i>Activity Diagram Tampil Data Siswa</i> .....	45
Gambar 4. 7 <i>Activity Diagram Tambah Data Siswa/i</i> .....	46
Gambar 4. 8 <i>Activity Diagram Edit Data Siswa/i</i> .....	46
Gambar 4. 9 <i>Activity Diagram Hapus Data siswa/i</i> .....	47
Gambar 4. 10 <i>Activity Diagram Tampil Data Guru</i> .....	47
Gambar 4. 11 <i>Activity Diagram Tampil Mata Pelajaran</i> .....	48
Gambar 4. 12 <i>Activity Diagram Tampil Jadwal Mata Pelajaran</i> .....	48
Gambar 4. 13 <i>Activity Diagram Tampil Halaman User</i> .....	49
Gambar 4. 14 <i>Activity Diagram Tampil Jadwal Kelas</i> .....	49
Gambar 4. 15 <i>Activity Diagram Tampil Absensi Siswa</i> .....	50
Gambar 4. 16 <i>Activity Diagram Tampil Monitoring Siswa</i> .....	50
Gambar 4. 17 <i>Activity Diagram Cetak Surat</i> .....	51
Gambar 4. 18 <i>Activity Diagram Absensi</i> .....	52
Gambar 4. 19 <i>Activity Diagram Tampil Data Pribadi Siswa</i> .....	53
Gambar 4. 20 <i>Activity Diagram Tampil Riwayat Absensi Siswa</i> .....	53
Gambar 4. 21 <i>Class Diagram</i> .....	54
Gambar 4. 22 <i>Menu Dashboard</i> .....	55
Gambar 4. 23 <i>Menu Kelola Kelas</i> .....	55
Gambar 4. 24 <i>Menu Kelola Siswa/i</i> .....	56
Gambar 4. 25 <i>Menu Kelola Guru</i> .....	56
Gambar 4. 26 <i>Menu Kelola Jadwal</i> .....	57
Gambar 4. 27 <i>Menu Mata Pelajaran</i> .....	57

Gambar 4. 28 Menu Akun Pengguna.....	58
Gambar 4. 29 Menu Absensi Siswa.....	58
Gambar 4. 30 Form Absensi Manual Siswa .....	59
Gambar 4. 31 Form Edit Absensi Siswa.....	59
Gambar 4. 32 Menu Monitoring .....	60
Gambar 4. 33 Halaman Login Mobile .....	60
Gambar 4. 34 Halaman Home Mobile.....	61
Gambar 4. 35 Halaman Mata Pelajaran Mobile.....	61
Gambar 4. 36 Notifikasi Absensi Berhasil .....	62
Gambar 5. 1 Halaman Login.....	63
Gambar 5. 2 Halaman <i>Dashboard</i> .....	64
Gambar 5. 3 Halaman Kelola Kelas .....	64
Gambar 5. 4 Halaman Kelola Siswa.....	65
Gambar 5. 5 Halaman Kelola Guru .....	65
Gambar 5. 6 Halaman Kelola Mata Pelajaran .....	65
Gambar 5. 7 Halaman Kelola Jadwal .....	66
Gambar 5. 8 Halaman Kelola Akun Pengguna.....	66
Gambar 5. 9 Halaman Login.....	67
Gambar 5. 10 Halaman <i>Dashboard</i> .....	67
Gambar 5. 11 Halaman Jadwal .....	67
Gambar 5. 12 Halaman Kelola Absensi Siswa .....	68
Gambar 5. 13 Halaman <i>Login</i> .....	68
Gambar 5. 14 Dashboard .....	69
Gambar 5. 15 Halaman Monitoring .....	69
Gambar 5. 16 Halaman Login.....	70
Gambar 5. 17 Halaman <i>Dashboard</i> .....	70
Gambar 5. 18 Tampilan halaman Absensi.....	71
Gambar 5. 19 Tampilan Halaman Riwayat Absensi.....	71



## DAFTAR TABEL

Tabel 2. 1 Tipe Data <i>Database MySQL</i> .....	17
Tabel 2. 2 Simbol Use Case Diagram .....	24
Tabel 2. 3 Simbol <i>Class Diagram</i> .....	25
Tabel 2. 4 Simbol <i>Activity Diagram</i> .....	26
Tabel 3. 1 <i>Gannt Chart</i> Waktu Penelitian .....	28
Tabel 4. 1 Spesifikasi Perangkat .....	31
Tabel 4. 2 Spesifikasi Perangkat Lunak.....	31
Tabel 4. 3 Deskripsi Case Login (Admin, Siswa, Guru dan Guru BK).....	35
Tabel 4. 4 Deskripsi Case Kelola Data Kelas .....	35
Tabel 4. 5 Deskripsi Case Kelola Data Siswa.....	36
Tabel 4. 6 Deskripsi Case Kelola Data Guru .....	37
Tabel 4. 7 Deskripsi Case Kelola Mata Pelajaran.....	37
Tabel 4. 8 Deskripsi Case Kelola Jadwal Pelajaran.....	38
Tabel 4. 9 Deskripsi Case Kelola User .....	39
Tabel 4. 10 Deskripsi Case Melihat Jadwal Kelas.....	39
Tabel 4. 11 Deskripsi Case Kelola Absensi Siswa .....	40
Tabel 4. 12 Deskripsi Case Monitoring Kehadiran Siswa .....	41
Tabel 4. 13 Deskripsi Case Absensi Siswa.....	41
Tabel 4. 14 Deskripsi Case Melihat Data Pribadi.....	42
Tabel 4. 15 Deskripsi Case Melihat Jadwal Mata Pelajaran.....	43
Tabel 4. 16 Deskripsi Case Melihat Riwayat Absensi.....	43
Tabel 5. 1 Pengujian <i>Black Box</i> .....	72
Tabel 5. 2 Pengujian Jarak Haversine Dan Jarak Sesungguhnya .....	74

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Absensi dapat dikatakan suatu pendataan kehadiran yang merupakan bagian dari aktivitas pelaporan yang ada dalam sebuah institusi. Sistem absensi memiliki peranan penting dalam kehidupan sehari-hari terutama di lingkungan sekolah, universitas, pabrik, perkantoran, rumah sakit dan tempat lainnya sebagai tanda kehadiran. Sejalan dengan berkembangnya teknologi, sistem absensi dalam dunia pendidikan umumnya masih dilakukan secara manual. Sulitnya mengetahui posisi keberadaan bagi siswa yang ingin melakukan absensi merupakan suatu permasalahan yang membutuhkan dukungan teknologi (Irmayana dkk., 2021).

Salah satu teknologi umum yang dimiliki perangkat seluler sekarang adalah *Global Positioning System* (GPS), yaitu sebuah sistem yang tertanam di smartphone pengguna dan berbasis satelit. Teknologi GPS mampu memberikan informasi mengenai posisi, kecepatan, dan waktu secara cepat, akurat, dan murah. Sehingga teknologi ini sering digunakan dalam pengimplementasian teknologi *Location Based Services* (LBS). Dengan kata lain, teknologi GPS merupakan salah satu teknologi yang sering digunakan sebagai komponen posisi dalam pengimplementasian teknologi LBS. Sedangkan teknologi LBS merupakan salah satu bentuk pengimplementasian dari layanan *Geographics Information System* (GIS) yang dapat diakses menggunakan *mobile device* (perangkat seluler) dengan menggunakan jaringan internet (Yuwamahendra & Ratnasari, 2020).

Adapun proses absensi lokasi bekerja dengan menghitung jarak antara lokasi kelas dan lokasi perangkat seluler yang memiliki teknologi GPS. Proses perhitungan jarak antara dua titik lokasi dapat menggunakan beberapa metode seperti *Haversine Formula*, *Euclidean Distance* atau *Manhattan Distance*. *Euclidean distance* adalah perhitungan untuk mengukur jarak dua titik dalam *euclidean space* yang mempelajari hubungan antara sudut dan jarak (Derisma, Firdaus, & Yusya, 2016). *Haversine formula* adalah metode perhitungan jarak

antara dua titik di bumi berdasarkan panjang garis lurus antara dua titik tanpa mengabaikan kelengkungan bumi (Farid & Yunus, 2017). *Manhattan distance* adalah metode perhitungan jarak pada ruang jarak dengan menerapkan konsep selisih mutlak (Buaton, Sundari, & Maulita, 2016).

Metode *Manhattan* menghasilkan deviasi paling besar. Konsep perhitungan jarak *manhattan* yaitu menerapkan konsep pencarian selisih murni antar data. Hal tersebut kurang cocok terhadap perhitungan jarak yang menggunakan variabel koordinat latitude dan longitude. Perhitungan *euclidean* menerapkan konsep *pythagoras* sedangkan *haversine* menerapkan konsep perhitungan jarak pada permukaan bola dengan tidak menghiraukan kemiringan.

*Haversine* memiliki tingkat akurasi yang lebih tinggi dari pada *euclidean* dan *Manhattan*. Dari hal tersebut didapatkan hasil analisis bahwa perhitungan jarak pada permukaan bumi harus tetap mengikutsertakan kemiringan permukaan bumi (Miftahuddin dkk., 2020).

Dari pembahasan di atas pada penelitian ini digunakan teknologi *location based service* untuk mengetahui titik lokasi siswa dan menerapkan metode *haversine formula* untuk menghitung jarak antara titik lokasi kelas dan perangkat seluler. Dengan penerapan konsep tersebut juga diharapkan sistem ini membantu guru bimbingan konseling (BK) untuk memonitoring kehadiran siswa/i disetiap mata pelajaran apabila siswa/i tidak melakukan absensi pada jam pelajaran dimulai hingga mata pelajaran selesai maka siswa tersebut akan dianggap tidak hadir. Setelah itu sistem akan mengakumulasi kehadiran siswa apabila telah melebihi jumlah maksimum ketidakhadiran maka sistem akan otomatis membuat surat dalam bentuk pdf yang akan dikirimkan ke orang tua siswa/i tersebut.

Maka dari itu penelitian ini akan dituangkan ke dalam bentuk skripsi dan memberikan judul **“Implementasi Teknologi *Location Based Service* (LBS) dan Metode *Haversine Formula* Pada Sistem Monitoring Kehadiran Siswa Secara *Real Time*”**.

## 1.2. Rumusan Masalah

Rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan algoritma *Haversine Formula* sebagai metode absensi dan monitoring kehadiran siswa di SMA 1 Wonggeduku ?
2. Bagaimana mengembangkan sistem absensi dan monitoring agar dapat terdigitalisasi dan dapat membantu rekapitulasi kehadiran siswa ?
3. Bagaimana sistem absensi dan monitoring memberikan informasi kehadiran siswa di setiap mata pelajaran secara *real time* kepada guru bimbingan konseling (BK)?

## 1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem ini hanya menerapkan algoritma *Haversine Formula*.
2. Ponsel *Android* pengguna aplikasi harus terkoneksi jaringan internet dengan baik.
3. Sistem ini hanya digunakan dalam lingkup SMA 1 Wonggeduku.
4. Sistem ini menggunakan handphone yang mempunyai fitur GPS.
5. Algoritma *Haversine Formula* tidak memperhitungkan ketinggian lokasi.
6. Sistem ini membaca titik lokasi siswa hanya pada saat proses absensi.
7. Status kehadiran yang digunakan dalam sistem ini hanya hadir dan tidak hadir (alpa).

## 1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan algoritma *Haversine Formula* sebagai metode absensi dan monitoring kehadiran siswa di SMA 1 Wonggeduku.
2. Mengembangkan sistem absensi dan monitoring agar dapat terdigitalisasi dan dapat membantu rekapitulasi kehadiran siswa.
3. Membangun sistem aplikasi yang dapat menginformasikan kehadiran siswa di setiap mata pelajaran secara *real time* kepada guru bimbingan konseling (BK).

### **1.5. Manfaat Penelitian**

Adapun manfaat dari penelitian ini adalah dapat membantu guru bimbingan konseling (BK) dalam memonitoring kehadiran siswa di SMA 1 Wonggeduku.

### **1.6. Sistematika Penulisan**

Sistematika penulisan proposal Tugas Akhir ini adalah sebagai berikut:

## **BAB I PENDAHULUAN**

Bab ini menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

## **BAB II LANDASAN TEORI**

Bab ini memuat pengertian-pengertian dan teori-teori yang menjadi acuan dalam pembuatan analisa dan pemecahan dari permasalahan yang dibahas meliputi *LBS*, *GPS*, *Android*, PHP, DBMS, MySQL, dan pendukung lain.

## **BAB III METODOLOGI PENELITIAN**

Bab ini berisi metode penelitian yang digunakan. Langkah – langkah pengumpulan data, prosedur pengembangan perangkat lunak dan perangkat keras yang dilakukan dalam penelitian.

## **BAB IV ANALISIS DAN PERANCANGAN SISTEM**

Bab ini berisi tentang gambaran umum sistem, desain perangkat keras dan perancangan sistem yang telah dibuat.

## **BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM**

Bab ini membahas mengenai implementasi dan pengujian sistem terhadap sistem yang telah dibuat.

## **BAB VI PENUTUP**

Bab ini berisi kesimpulan yang diambil dari hasil pembuatan sistem serta saran-saran untuk pengembangan dari penulis.

## 1.7 Tinjauan Pustaka

Pada tahun 2019, Wiwin Susanty, Ismail Nanda Astari dan Taqwan Thamrin melakukan penelitian dengan judul Aplikasi GIS Menggunakan *Metode Location Based Service (LBS)* Berbasis Android. Dalam penelitian ini penulis melakukan penelitian dengan menggunakan metode *Location Based Service (LBS)*. Konsep dari metode *Location Based Service* ini sendiri menggunakan database informasi geografis yang digabungkan dengan teknologi *Global Positioning System (GPS)* yang tertanam di *smartphone* pengguna untuk melacak suatu pergerakan *device* pengguna dan mengirimkan informasi yang dibutuhkan oleh *device* pengguna (Susanty dkk., 2019).

Pada tahun 2018, Erviana Devie dan Edy Winarno melakukan penelitian dengan judul Aplikasi *Location Based Service* Untuk Informasi Kuliner Di Yogyakarta. Dalam penelitian *Location Based Service (LBS)* dimanfaatkan sebagai metode layanan berbasis lokasi untuk mengetahui rute perjalanan, yang memberikan hasil dengan tingkat keakuratan yang cukup tinggi serta dapat memberikan informasi lokasi. Layanan ini menggunakan teknologi *global positioning service (GPS)* dan *cell-based location* dari *Google*. Pengujian akurasi jarak tempuh sebelumnya dilakukan dengan data dari beberapa jenis menu makanan yang kemudian akan ditampilkan pada tampilan Maps dan dihitung secara manual menggunakan rumus *haversine* (Devie & Winarno, 2019).

Pada tahun 2020, Gede Aditra Pradnyana dan A.A. Gede Raka Wahyu Brahma melakukan penelitian dengan judul Pengembangan Sistem Informasi Monitoring Kehadiran guru Secara *Realtime* Berbasis Web. Penelitian dilakukan dengan tujuan untuk untuk membuat sistem informasi absensi yang mampu memonitoring dan memberikan informasi yang cepat dan tepat mengenai kehadiran seorang guru di kampus (Pradnyana & Brahma, 2020).

Pada tahun 2021, Naviza Qois dan Yuwan Jumaryadi melakukan penelitian dengan judul Implementasi *Location Based Service* Pada Sistem Informasi Kehadiran Pegawai Berbasis Android. Dalam penelitian ini *Location Based Service (LBS)* dimanfaatkan untuk melakukan konversi koordinat menjadi sebuah

alamat. Pada Android Studio LBS menggunakan *geocoder* sebagai pendukung untuk mengetahui lokasi *user* secara *realtime*. Dengan menggunakan *geocoder* ini letak koordinat yang berupa latitude dan longitude dapat dikonversi menjadi sebuah alamat yang dapat dibaca dan diketahui letak atau posisinya. *Mobile apps* yang menggunakan metode ini dapat dilakukan tracking lokasi secara *realtime*. Penelitian ini juga menggunakan GPS untuk mendapatkan lokasi kehadiran dari absensi yang dilakukan karyawan berupa koordinat longitude dan latitude (Qois & Jumaryadi, 2021).

Pada tahun 2020, Ahmad Khairi dan Fathur Rizal melakukan penelitian dengan judul Aplikasi Monitoring Kehadiran Menggunakan *Global Positioning System* Berbasis Android Untuk Peningkatan Kinerja Karyawan Di Universitas Nurul Jadid (UNUJA). Dalam penelitian ini GPS dimanfaatkan untuk menentukan posisi pengguna. Pengumpulan data pada penelitian ini dilakukan dengan mengamati proses presensi karyawan UNUJA serta dilakukan pula wawancara kepada unsur pimpinan dan sivitas akademik UNUJA untuk mengetahui proses serta pengolahan data presensi yang telah berjalan dan mendata kendala serta permasalahan yang ditemukan. Dalam penelitian ini juga menggunakan studi literatur untuk memperoleh referensi yang berkaitan dengan presensi karyawan. Hasil pengumpulan data yang diperoleh akan dijadikan acuan dalam menganalisa serta merancang sistem presensi kehadiran karyawan di UNUJA (Fathur Rizal & Ahmad Khairi, 2020).

## **BAB II**

### **LANDASAN TEORI**

#### **2.1     *Global Navigation Satellite System (GNSS)***

*Global Navigation Satellite System (GNSS)* merupakan istilah singkatan dari suatu sistem satelit navigasi yang menyediakan posisi geospasial dalam lingkup global. GNSS beroperasi secara penuh sejak Desember 2009. Diawali dengan *Global Positioning System (GPS)* yang merupakan suatu konstelasi yang terdiri tidak kurang dari 24 satelit yang menyediakan informasi koordinat posisi yang akurat secara global. GPS mempergunakan satelit dan komputer untuk melakukan penghitungan posisi dimanapun di muka bumi ini. Sistem ini dimiliki, dioperasikan dan dikontrol oleh *United States Departement of Defenses (DoD)*. GNSS dapat dipergunakan secara global dimanapun dan oleh siapapun dimuka bumi ini secara gratis (Wahyono, Eko Budi; Suhattanto, 2019). Seiring dengan perkembangan Satelit GPS, GLONASS yang merupakan sistem GNSS yang dimiliki oleh Russia mempunyai cakupan seluruh dunia dengan 18 satelit yang tersedia sejak Desember 2009, dan satelit GALILEO milik Eropa juga COMPASS milik China sedang dikembangkan. GLONASS (*GLO*bal'*n*aya *NA*vigatsionnaya *Sputnikovaya Systema*, atau *Global Navigation Satellite System*) merupakan sistem navigasi ruang angkasa milik Russia yang bisa disamakan dengan sistem GNSS milik Rusia. Satelit berjumlah 21 pada 3 bidang orbit datar (Wahyono, Eko Budi; Suhattanto, 2019).

GNSS sekarang ini terdiri dari 6 Satelit, yaitu:

1.     NAVSTAR GPS (*NAVigation Satellite Timing and Ranging Global Positioning System (USA)*).

NAVSTAR GPS merupakan satelit yang dibuat dan dioperasikan Amerika Serikat diluncurkan sejak 22 Februari 1978 untuk kepentingan penentuan posisi dan navigasi. Sampai saat ini terdapat 32 satelit GPS yang sehat untuk kepentingan penentuan posisi dan navigasi.



2. GLONASS (*Global'naya Navigatsionnaya Sputnikovaya Systema*) (Rusia).

GLONASS adalah sistem satelit navigasi global kedua setelah GPS. Pembangunan satelit GLONASS dimulai pada tahun 2001 dengan peluncuran satelit ke orbitnya oleh pemerintah Rusia. Hingga saat ini terdapat 29 satelit yang aktif, namun hanya terdapat 24 satelit dengan kondisi yang sehat.

3. Galileo (Eropa)

Galileo merupakan satelit yang dibuat oleh negara-negara yang tergabung dengan Uni Eropa. Pembangunan satelit GLONASS dilaksanakan oleh Komisi Eropa (*European Commission*) dan ESA (*European Space Agency*). ESA bekerja sama dengan the *Galileo Industries Company GmbH* (sekarang menjadi the *European Satellite Navigation Industries* = ESNI). Jika GPS dan GLONASS tujuan awalnya untuk kepentingan militer, maka Galileo tujuan awalnya memang untuk kepentingan pembangunan ekonomi dunia dan kepentingan sipil. Pada tahun 2013, menurut tahapan pembangunannya sudah bisa beroperasi secara penuh dengan 27 (+3) satelit Galileo beroperasi penuh.

4. Compass (China) / Beidou

China pada tahun 2000 memulai pengembangan satelit untuk keperluan penentuan posisi dan navigasi. Satelit untuk penentuan posisi dan navigasi ini diberi nama COMPASS tetapi jika dalam bahasa China dinamakan Beidou. Terdapat tiga tahap pengembangan satelit Beidou ini: Tahap I (periode 2000 – 2003) pembangunan awal satelit sistem penentuan posisi di dalam negeri China, Tahap II pada tahun 2012 sudah dapat menjangkau seluruh kawasan Asia Pasifik, Tahap II pada tahun 2020 sudah bisa menjangkau seluruh dunia. Segmen angkasa sistem COMPASS direncanakan akan terdiri dari lima satelit GEO dan 30 satelit non-GEO.

5. QZSS (*Quasi-Zenith System Satellite*)

*Quasi-Zenith System Satellite* (QZSS) atau dalam bahasa Jepang Jun-Ten-Cho mulai dibangun oleh Pemerintah Jepang pada tahun 2003. QZS akan meningkatkan kinerja GPS dalam dua cara, yaitu peningkatan ketersediaan sinyal GPS, dan peningkatan performa GPS (mencakup akurasi dan keaslian sinyal GPS) (Service of QZSS). QZSS terdiri dari 3 (tiga) satelit dan akan memberikan layanan

posisi satelit secara regional serta komunikasi dan broadcasting. Setiap satelit akan berada dalam 3 bidang orbit yang berbeda, di mana mempunyai kemiringan 450 terhadap Geostationary Orbit (GEO). Satelit pertama yang diberi nama Michibiki telah diluncurkan pada tanggal 11 September 2010. Diharapkan QZSS ini akan beroperasi secara penuh pada tahun 2013. Dalam orbitnya tersebut, satelit QZSS akan melengkapi system GNSS lainnya yang selama ini digunakan Jepang. Selain itu QZSS akan mencakup wilayah Australia dan daerah Asia. sistem satelit QZSS diaplikasikan untuk menyediakan layanan berbasis komunikasi (video, audio, dan data), dan informasi posisi (Quasi-Zenith Satellite System, 2008).

6. IRNSS (*India Regional Navigation Satellite System*)

IRNSS adalah sistem satelit navigasi yang dikembangkan oleh badan antariksa India, *India Space Research Organisation* (ISRO) yang berada di bawah kontrol pemerintah India.

Kelebihan dan Kekurangan Penentuan Posisi dengan Teknologi GNSS.

1. Kelebihan Penentuan Posisi dengan Teknologi GNSS

Adapun kelebihanannya, yaitu:

- a. GNSS / GPS Geodetic dapat digunakan setiap saat tanpa tergantung waktu dan cuaca.
- b. Satelit-satelit GNSS mempunyai ketinggian orbit yang cukup tinggi yaitu sekitar 20.000 km di atas permukaan bumi serta dengan jumlah yang relatif cukup banyak. Hal ini menjadikan GNSS dapat meliputi wilayah yang cukup luas sehingga dapat digunakan oleh banyak orang sekaligus.
- c. Penggunaan GPS Geodetic dalam penentuan posisi relatif tidak terlalu terpengaruh dengan kondisi topografis daerah survei dibandingkan dengan penggunaan metode terestris.
- d. Posisi yang ditentukan oleh GNSS / GPS Geodetic mengacu ke suatu datum global yang relatif teliti dan mudah direalisasikan, yaitu datum WGS 84.10
- e. GNSS dapat memberikan ketelitian posisi yang spektrumnya cukup luas. Dari yang sangat teliti (orde millimeter) sampai orde meter.
- f. Pemakaian sistem GNSS tidak dikenakan biaya.
- g. Lebih efisien dalam waktu, biaya operasional, dan tenaga.

- h. Celah untuk memanipulasi data pada pengukuran GNSS lebih sulit dibandingkan menggunakan metode terestris.
  - i. Relatif mudah dipelajari sekalipun oleh orang awam yang belum pernah menggunakan.
2. Kekurangan Penentuan Posisi dengan Teknologi GNSS.
- Adapun kekurangannya, yaitu :
- a. Tidak boleh ada penghalang antara receiver dan satelit.
  - b. Komponen tinggi yang dihasilkan adalah tinggi dengan acuan ellipsoid.
  - c. Perlu proses yang relatif tidak mudah untuk menganalisa data.

## **2.2 *Location Based Service (LBS)***

*Location Based Service (LBS)* atau layanan berbasis lokasi adalah layanan di android yang memungkinkan aplikasi dapat melacak lokasi pemakai yang sedang menggunakan ponsel (Qois & Jumaryadi, 2021). Melalui layanan ini, informasi mengenai bujur dan lintang bias diperoleh. Hal ini dapat dipakai untuk mendapatkan informasi lokasi pemakai, mencari rute jalan, atau lainnya di peta Google Map. Sebuah aplikasi LBS minimal harus mempunyai kemampuan untuk:

- 1. Mengetahui koordinat posisi *smartphone* pengguna,
- 2. Mengakses bank data yang menyimpan data koordinat lokasi dan informasi mengenai lokasi tersebut,
- 3. Menghitung jarak antara posisi pengguna dengan sebuah lokasi,
- 4. Menampilkannya menjadi informasi yang bisa dibaca pengguna contohnya kedalam Map (Google Map).

Terdapat lima komponen pendukung utama dalam teknologi layanan berbasis lokasi, antara lain:

- 1. Piranti *Mobile*

Merupakan salah satu komponen penting dalam LBS. Piranti ini berfungsi sebagai alat bantu bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar, dan lain sebagainya. Piranti mobile yang dapat digunakan bisa berupa PDA, *smartphone*, dan laptop. Selain itu piranti *mobile* dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya alat navigasi berbasis GPS.

## 2. Jaringan Komunikasi

Komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari piranti *mobile*-nya untuk kemudian dikirimkan ke penyedia layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.

## 3. Komponen *Positioning* (Penunjuk Posisi atau Lokasi)

Setiap layanan yang diberikan oleh penyedia layanan biasanya berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah atau pemroses yang menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut dapat didapatkan melalui jaringan komunikasi mobile ataupun menggunakan *Global Positioning System* (GPS).

## 4. Penyedia layanan dan aplikasi

Penyedia layanan merupakan komponen LBS yang memberikan berbagai macam layanan yang dapat digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar dapat mengetahui posisi pengguna saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, menemukan rute jalan, mencari data di *Yellow Pages* sesuai dengan permintaan, dan lain sebagainya.

## 5. Penyedia data dan konten

Penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolah, karena terdapat kemungkinan bahwa data dan informasi yang diolah tersebut berasal dari pengembang atau pihak ketiga yang memiliki otoritas untuk menyimpannya. Sebagai contoh basis data geografis dan lokasi bisa saja berasal dari badan milik pemerintah atau juga data-data perusahaan, bisnis atau industri bisa saja berasal dari *Yellow Pages*, maupun perusahaan penyedia data lainnya.

### 2.3 *Haversine Formula*

*Haversine Formula* merupakan metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat kelengkungan. Metode *haversine*

*formula* menghitung jarak antara 2 titik dengan berdasarkan panjang garis lurus antara 2 titik pada garis bujur dan lintang (Putra dkk., 2015). Sebelum menghitung jarak tertentu di perlukan beberapa nilai untuk memenuhi kriteria dari rumus *haversine* sendiri yaitu menentukan nilai latitude dan longitude serta  $\Delta lat$  dan  $\Delta long$ . Adapun nilai dari  $\Delta lat$  didapatkan dengan cara mengurangi nilai latitude titik 1 dan latitude titik 2 sedangkan nilai  $\Delta long$  didapatkan dengan cara mengurangi nilai dari longitude titik 1 dan longitude titik 2.

Berikut bentuk Rumus *Haversine Formula*:

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2\left(\frac{\Delta long}{2}\right) \quad (2.1)$$

$$d = 2r \cdot \arcsin(\sqrt{a}) \quad (2.2)$$

Keterangan :

r = jari-jari bumi 6371 (km)

d = jarak (km)

lat<sub>1</sub> = latitude titik 1

lat<sub>2</sub> = latitude titik 2

$\Delta lat$  = besaran perubahan latitude

$\Delta long$  = besaran perubahan longitude

**Contoh perhitungan *Haversine Formula*:**

Diketahui : r = 6371 (km)

$$lat_1 = -6,8971795$$

$$lat_2 = -6,897536468$$

$$long_1 = 107,63647$$

$$long_2 = 107,6363645$$

$$\Delta lat = 0,000356968$$

$$\Delta_{\text{long}} = 0,000105549$$

$$a = \sin^2\left(\frac{0,000356968}{2}\right) + \cos(-6,8971795) \cdot \cos(-6,897536468) \cdot \sin^2\left(\frac{0,000105549}{2}\right)$$

$$d = 2 \cdot 6371 \cdot \arcsin(\sqrt{a})$$

$$d = 44,3 \text{ m}$$

## 2.4 PHP

PHP Kepanjangan dari PHP adalah "*Hypertext Preprocessor*" (ini merupakan singkatan rekursif). PHP adalah bahasa *scriptingweb HTML-embedded*. Ini berarti kode PHP dapat disisipkan ke dalam HTML halaman Web. Ketika sebuah halaman PHP diakses, kode PHP dibaca atau "diurai" oleh *server*. *Output* dari fungsi PHP pada halaman biasanya dikembalikan sebagai kode HTML, yang dapat dibaca oleh *browser*. Karena kode PHP diubah menjadi HTML sebelum halaman dibuka, pengguna tidak dapat melihat kode PHP pada halaman. Ini membuat halaman PHP cukup aman untuk mengakses database dan informasi aman lainnya. (Ferdianto, 2013).

Banyak sintaks PHP yang hasil adaptasi dari bahasa lain seperti bahasa C, Java dan Perl. Namun, PHP memiliki sejumlah fitur unik dan fungsi tertentu juga. Tujuan dari bahasa pemrograman PHP adalah untuk memungkinkan pengembangan web untuk menulis halaman yang dihasilkan secara dinamis dengan cepat dan mudah. PHP juga bagus untuk menciptakan situs Web *database-driven*. Jika Anda ingin mempelajari lebih lanjut tentang PHP, situs resminya yaitu PHP.net. (Ferdianto, 2013).

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya *milis - milis* dan *developer* yang siap membantu dalam pengembangan.

4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem (Ferdianto, 2013).

### 2.5 *Hypertext Markup Language (HTML)*

HTML adalah sebuah bahasa pemrograman yang berbentuk skrip-skrip yang berguna untuk membuat sebuah halaman *web*. HTML dapat dibaca oleh berbagai *platform* seperti: Windows, Linux, Macintosh. Kata *Markup Language* pada HTML menunjukkan fasilitas yang berupa tanda tertentu dalam skrip HTML dimana pengguna bisa mengatur judul, garis, tabel, gambar, dan lain-lain dengan perintah yang telah ditentukan pada elemen HTML. HTML sendiri dikeluarkan oleh W3C (*World Wide Web Consortium*), setiap terjadi perkembangan level HTML harus dievaluasi ketat dan disetujui oleh W3C. Contoh tag HTML antara lain: `<head>`, `<body>` dan `<table>` (C, 2020).

### 2.6 *Cascading Style Sheet (CSS)*

*Cascading Style Sheets (CSS)* adalah salah satu bahasa pemrograman desain *web (style sheet language)* yang mengontrol format tampilan sebuah halaman *web* yang ditulis dengan menggunakan bahasa penanda (*markup language*). Biasanya CSS digunakan untuk mendesain sebuah halaman HTML dan XHTML, tetapi sekarang bahasa pemrograman CSS bisa diaplikasikan untuk segala dokumen XML, termasuk SVG dan XUL. CSS dibuat untuk memisahkan konteks utama (biasanya dibuat dengan menggunakan bahasa HTML dan sejenisnya) dengan tampilan dokumen yang meliputi *layout*, warna dan *font*. Pemisahan ini dapat meningkatkan daya akses konten pada *web*, menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi dari sebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi banyak halaman untuk sebuah formatting dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik *tables* pada layout desain *web (layout tanpa tabel)*

*Style Sheet* adalah sebuah *text file* yang sederhana (dimana berekstensi \*.css), ditulis menurut aturan bahasa yang dipaparkan pada rekomendasi CSS1 atau CSS2.

Dalam *Cascading Style Sheets* ada dua cara menghubungkan sebuah dokumen HTML dengan CSS

1. *Selector class*
2. *Selector ID*
3. *Selector Descendant* (turunan)
4. *Selector Link Pseudo Class*
5. *Selector Pseudo elemen*
6. *Selector dynamic pseudo class*
7. *Selector languange*
8. *Selector child*
9. *Selector first-child*
10. *Selector adjacent* (berdekatan).

## **2.7 Android**

Android adalah sistem operasi yang dikeluarkan oleh Google. Android dibuat khusus untuk *smartphone* dan tablet. Berbagai macam produsen telah menggunakan Android sebagai sistem operasi untuk peranti (*device*) yang mereka produksi. Android juga mempunyai *store* dengan lebih dari 2 miliar pengguna aktif, per Januari 2018 ketika tulisan ini dibuat.

Sejak dirilis tahun 2008, Google telah mengeluarkan beberapa versi, dengan "Pie" sebagai versi yang terbaru.

Pada tahun 2013, Android menjadi *operation system* (OS) terlaris pada tablet dan *smartphone*. Tercatat pada tahun 2016, *store* Android memiliki lebih dari 2.8 juta aplikasi.

Android menarik bagi perusahaan teknologi yang membutuhkan barang siap jadi, biaya rendah dan kustomisasi OS untuk perangkat teknologi tinggi mereka. Hal ini menjadi daya tarik bagi banyak perusahaan, sehingga mereka memilih Android.



*Source code* dari Android bersifat *open source*. Ini adalah hal menarik bagi komunitas *developer*, karena lisensi *open source* sangat mendukung untuk mengembangkan produknya dengan aman.

Versi-versi android antara lain :

1. Android versi 1.1
2. Android versi 1.5 (*Cupcake*)
3. Android versi 1.6 (*Donut*)
4. Android versi 2.0 / 2.1 (*Eclair*)
5. Android versi 2.2 Froyo (*Frozen Yoghurt*)
6. Android versi 2.3 (*Gingerbread*)
7. Android versi 3.0/3.1 (*Honeycomb*)
8. Android versi 4.0 ICS (*Ice Cream Sandwich*)
9. Android versi 4.1 – 4.3 (*Jelly Bean*)
10. Android versi 4.4 (*Kitkat*)
11. Android versi 5.0 – 5.1 (*Lollipop*)
12. Android versi 6.0 (*Marshmallow*)
13. Android versi 7.0 (*Nougat*)
14. Android versi 8.0 / 8.1 (*Oreo*)
15. Android versi 9.0 (*Pie*)

## 2.8 **MySQL**

*MySQL* adalah aplikasi *SQL database server* yang multi *user*. Oleh karena itu, *MySQL* digunakan dalam pembuatan aplikasi ini yang digunakan sebagai *database server* untuk menyimpan data lokasi yang dikirim oleh masing-masing telepon seluler. Dalam *MySQL* ada beberapa operasi dasar yang lebih dikenal dengan operasi CRUD yaitu :

### 1. *Create*

*Create* adalah operasi penambahan data baru ke dalam table. Terdapat 2 *Query* untuk menambah data, yang pertama adalah penambahan yang tidak menspesifikasikan nama kolom yang akan ditambahkan dan hanya memberikan isi dari tabelnya. *Query*nya adalah sebagai berikut:

```
INSERT INTO table_name
VALUES (value1, value2, value3,...);
```

Kedua adalah menginputkan data dengan mencantumkan nama kolom yang akan diisi dengan isi di dalamnya. *Query*nya adalah sebagai berikut:

```
INSERT INTO table_name (column1,column2,column3,...)
VALUES (value1, value2, value3,...);
```

## 2. *Read*

*Read* adalah operasi untuk menampilkan semua atau sebagian data yang berada di dalam *database*. *Query*nya adalah sebagai berikut:

```
SELECT column_name, column_name
FROM table_name;
```

## 3. *Update*

*Update* adalah operasi untuk mengubah data yang ada di dalam *database*. *Query*nya adalah sebagai berikut:

```
UPDATE table_name
SET column1=value1, column2=value2,...
WHERE some_column=some_value;
```

## 4. *Delete*

*Delete* adalah operasi untuk menghapus data yang ada di dalam table. *Query*nya adalah sebagai berikut:

```
DELETE FROM table_name
WHERE some_column = some_value;
```

Tabel 2. 1 Tipe Data *Database MySQL*

No.	Type	Keterangan
1.	<i>Varchar</i>	Tipe data karakter yang panjangnya tidak tetap
2.	TINYINT	Adalah tipe data bilangan bulat yang rentangnya -128 sampai 127 atau menggunakan atribut unsigned dari 0 hingga 255
3.	<i>Text</i>	Tipe data yang dapat menampung semua tipe data.
4.	<i>Date</i>	Tipe data yang digunakan untuk mendiskripsikan tanggal.

Tabel 2.1 Tipe Data *Database MySQL* (Lanjutan)

No.	Type	Keterangan
5.	<i>Smallint</i>	Adalah tipe data bilangan bulat yang rentangnya –32768 sampai 32767. The unsigned range is 0 to 65535
6.	<i>Mediumint</i>	Adalah tipe data bilangan bulat yang rentangnya –8388608 to 8388607. unsigned range-nya 0 sampai 16777215
7.	<i>Int</i>	Tipe data yang bernilai integer/bilangan bulat.
8.	<i>Time</i>	Tipe data waktu. Jangkauannya adalah '-838:59:59' hingga '838:59:59'. MySQL menampilkan TIME dalam format 'HH:MM:SS'.
9.	<i>Char</i>	Tipe data untuk menampung data yang bertipe karakter
10.	<i>Primary Key</i>	Kunci primer adalah suatu atribut atau satu set minimal atribut yang tidak hanya mendefinisikan secara unik suatu kejadian spesifik tetapi juga dapat mewakili setiap kejadian dari suatu kejadian

## 2.9 Flutter

*Flutter* adalah *software development kit* (SDK) buatan *Google* yang berfungsi untuk membuat aplikasi mobile menggunakan bahasa pemrograman *Dart*, baik untuk *android* maupun *iOS*. Dengan *Flutter*, Aplikasi *android* maupun *iOS* dapat dibuat menggunakan basis kode dan bahasa pemrograman yang sama, yaitu *dart*, bahasa pemrograman yang juga diproduksi oleh *google* pada tahun 2011. Sebelumnya, aplikasi murni (native) untuk *android* perlu dibuat menggunakan bahasa pemrograman *java*, *kotlin*, sedangkan aplikasi *ios* perlu dibuat menggunakan bahasa pemrograman *objective-C* atau *Swift*. *Flutter* ditunjukkan untuk mempermudah dan mempercepat proses pengembangan aplikasi *mobile* yang dapat berjalan di atas *android* dan *ios* tanpa harus mempelajari dua bahasa pemrograman secara terpisah. *Flutter* dapat dikatakan sebagai produk *google* yang masih relatif baru. Rilis perdana *Flutter*, versi Alpha (v.0.0.6), dipublikasikan pada bulan mei 2017 (Raharjo, 2019).

## 2.10 Dart

*Dart* adalah bahasa pemrograman yang diproduksi oleh *Google*, dirancang oleh Lars Bak dan Kasper Lund. *Dart* pertama kali dikenalkan pada 10 oktober 2011. Versi 1.0 dari bahasa pemrograman ini baru dirilis pada bulan November 2013. *Dart* dapat digunakan untuk membuat aplikasi *server* (berbentuk *command-line interface*), web maupun *mobile* (android dan ios). Aplikasi *Dart* akan di eksekusi secara langsung melalui *Dart VM (Virtual Machine)* tanpa melalui proses penerjemahan ke kode objek (*bytecode*) terlebih dahulu (Sukindar, 2016).

Bahasa pemrograman *Dart* merupakan bahasa pemrograman *general-purpose* yang dirancang oleh Lars Bak dan Kasper Lund. Bahasa pemrograman ini dikembangkan sebagai bahasa pemrograman aplikasi yang dapat dengan mudah untuk dipelajari dan disebarkan (Sukindar, 2016).

Bahasa pemrograman *Dart* dapat digunakan untuk mengembangkan aplikasi untuk dijalankan pada berbagai macam peramban modern. *Dart* juga dapat digunakan untuk mengembangkan aplikasi dari *codebase* tunggal menjadi aplikasi *Android* maupun *iOS*. Adapun contoh sintaks bahasa pemrograman *Dart* adalah sebagai berikut:

### 1. Variabel dan tipe data

Variabel adalah sebuah simbol yang digunakan untuk menyimpan nilai. Sedangkan tipe data adalah jenis nilai yang akan kita simpan. Tipe data dasar pada *Dart* dibagi menjadi tiga macam yaitu tipe data angka (*int*, *double*, dll), tipe data teks (*String*) dan tipe data *boolean* (*bool*). Untuk membuat variabel pada *dart*, dapat menggunakan kata kunci *var* dan menuliskan langsung tipe datanya. Gambar 2.1 merupakan contoh dari penggunaan *variable* dan tipe data dalam *dart*.

```
Var judul = "Belajar Pemrograman Dart";
Int harga = 123000;
Double berat = 2.23;
```

Gambar 2. 1 Source Code Variable Dan Tipe

## 2. *Input output*

Dalam *dart* cara untuk menampilkan data dapat dilakukan dengan menggunakan fungsi `print()` dan untuk membaca data yang dimasukkan dari *console* dapat menggunakan metode `stdin.readLineSync()`. Gambar 2.2 merupakan contoh dari penggunaan *input/output* dalam *dart*.

```

import 'dart:io';

Main() {
  Stdoutwrite("Siapa kamu: ");
  Var nama = stdin.readLineSync ();
  Print("Hello $nama! ");
}

```

Gambar 2. 2 *Source Code Input Output*

## 3. *Conditional*

*Conditional* dalam *dart* sama seperti bahasa lain. Menggunakan blok *if-else*. Gambar 2.3 merupakan contoh dari penggunaan *conditional* dalam *dart*.

```

void main()
{
  int myAge = 29;
  if(myAge <= 10)
  {
    print("Anda masih anak-anak");
  }else if (myAge > 10 && myAge <= 17) {
    print("Anda sudah remaja");
  }
  }else{
    print("Anda sudah dewasa");
  }
}

```

Gambar 2. 3 *Source Code Conditional*

#### 4. *Loop*

*Loop* atau pengulangan adalah *suatu* konstruksi program yang berfungsi untuk melakukan eksekusi perintah secara berulang sampai kondisi tertentu. Gambar 2.4 merupakan contoh dari penggunaan *while* dalam *dart*.

```
void main()
{
    var i = 1; while
    (i < 5) {
        print(i);
        i++;
    }
    // kode di atas akan memberi output 1 - 4
}
```

Gambar 2. 4 *Source Code Loop*

#### 2.11 *Rational Unified Process (RUP)*

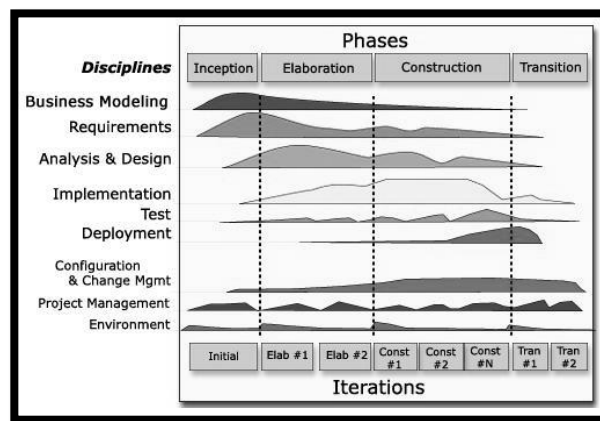
*Rational Unified Process* (RUP) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak. Gambar dibawah menunjukkan secara keseluruhan arsitektur yang dimiliki RUP (Rahardjo, 2018).

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui Gambar 2.5 dapat dilihat bahwa RUP memiliki 2 dimensi yaitu:

1. Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari fase selanjutnya. Setiap fase dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas *Inception*,

*Elaboration, Construction, dan Transition.*

2. Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas *Business Modeling*, *Requirement*, *Analysis and Design*, *Implementation*, *Test*, *Deployment*, *Configuration* dan *Change Management*, *Project Management*, *Environment*.



Gambar 2. 5 Arsitektur *Rational Unified Process* (Hartawan, 2017)

Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak yaitu:

### 1. *Inception*

Pada tahap ini pengembang mendefinisikan batasan kegiatan, melakukan analisis kebutuhan user, dan melakukan perancangan awal perangkat lunak (perancangan arsitektural dan *use case*). Pada akhir fase ini, prototipe perangkat lunak versi *Alpha* harus sudah dirilis

### 2. *Elaboration*

Pada tahap ini dilakukan perancangan perangkat lunak mulai dari menspesifikasikan fitur perangkat lunak hingga perilsan prototipe versi *Betha* dari perangkat lunak.

### 3. *Construction*

Pengimplementasian rancangan perangkat lunak yang telah dibuat dilakukan pada tahap ini. Pada akhir tahap ini, perangkat lunak versi akhir yang sudah disetujui administrator dirilis beserta dokumentasi perangkat lunak.

### 4. *Transition*

Instalasi *deployment* dan sosialisasi perangkat lunak dilakukan pada tahap ini. Pada tahapan ini dilakukan tahap evaluasi dari sistem yang telah direncanakan, dianalisa dan dibangun. Dari tahap ini bisa dilihat kekurangan dari sistem yang telah dibangun.

## 2.12 *Unified Modeling Language (UML)*

*Unified Modeling Language (UML)* merupakan pengganti dari metode analisis berorientasi objek dan desain berorientasi objek (OOA&D) yang dimunculkan sepenggunan akhir tahun 80-an dan awal tahun 90-an. UML merupakan gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobson. Tetapi UML ini akan mencakup lebih luas daripada OOA&D. Pada pertengahan pengembangan UML dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi. UML merupakan bahasa standar untuk penulisan *blue print software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak (Shalahudin, 2011).

## 2.13 *Diagram Dalam UML*

Dalam UML terdapat diagram-diagram yang bisa menggambarkan bagian atau aspek tertentu dari sebuah sistem. Sehingga dapat terlihat jelas alur dan









gambaran umum dari perangkat lunak yang dibangun. Ada beberapa jenis diagram dalam UML yaitu:





#### 1. *Use case Diagram*

Menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams*. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem (Shalahudin, 2011).

Tabel 2. 2 Simbol Use Case Diagram

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (dependent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent)
3.		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> )
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.


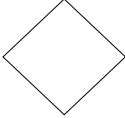
Tabel 2.2 Simbol *Use Case Diagram* (Lanjutan)

NO	SIMBOL	NAMA	KETERANGAN
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
8.		<i>Use case</i>	Deskripsi dari uraian aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu actor
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi)
10.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

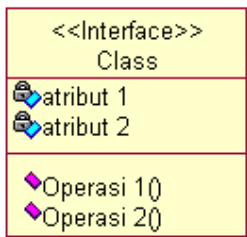



## 2. Class Diagram

Menggambarakan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialized* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram* (Rosa dan Shalahudin, 2011).

Tabel 2. 3 Simbol *Class Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
2.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek



Tabel 2.3 Simbol *Class Diagram* (Lanjutan)

NO	SIMBOL	NAMA	KETERANGAN
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu actor
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek
6.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri




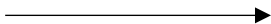
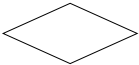
### 3. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi (Shalahudin, 2011).

Tabel 2. 4 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing jurusan antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi

Tabel 2.4 Simbol *Activity Diagram* (Lanjutan)

NO	GAMBAR	NAMA	KETERANGAN
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
6		<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol yang lain
7		<i>Decision</i>	Menunjukkan suatu keputusan yang mempunyai satu atau lebih transisi sesuai dengan kondisi

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Waktu dan Tempat Penelitian

##### 3.1.1. Waktu

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Maret 2022 sampai dengan Juli 2022. Rincian kegiatan dapat dilihat pada Tabel 3.1 berikut:

Tabel 3. 1 *Gannt Chart* Waktu Penelitian

No	Uraian	Waktu																			
		Maret				April				Mei				Juni				Juli			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	<i>Inception</i>																				
2.	<i>Elaboration</i>																				
3.	<i>Construction</i>																				
4.	<i>Transition</i>																				

##### 3.1.2. Tempat Penelitian

Adapun tempat penelitian tugas akhir dilakukan di SMA Negeri 1 Wonggeduku.

#### 3.2. Metode Pengumpulan Data

Dalam penelitian ini metode pengumpulan data yang digunakan pada perancangan aplikasi tersebut adalah studi literatur, mulai dari buku-buku, jurnal, data, maupun artikel, dan sumber-sumber lain. Metode ini dilaksanakan dengan melakukan studi kepustakaan yang relevan. Studi kepustakaan ini dilakukan untuk mencari sumber pelengkap yang berhubungan dengan sistem yang dibangun, yaitu dengan mencari referensi yang membahas tentang penggunaan metode *Location Based Service*, sehingga metode *Location Based Service* dapat diimplementasikan dalam aplikasi Absensi siswa/i di SMAN 1 Wonggeduku.

### **3.3. Metode Pengembangan Sistem**

Metode pengembangan sistem yang digunakan dalam sistem ini adalah metode *Rational Unified Process* (RUP). Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak, yaitu:

#### **3.3.1. Permulaan (*Inception*)**

Pada fase ini dilakukan proses pengidentifikasian sistem, dilakukan dengan analisis kebutuhan akan aplikasi, melakukan kajian terhadap penelitian yang terkait dengan metode *Location Based Service*.

#### **3.3.2. Perluasaan / Perencanaan (*Elaboration*)**

Setelah menentukan ruang lingkup penelitian, tahap ini dilakukan perancangan dan analisis sistem menggunakan flowchart meliputi flowchart Aplikasi Absensi, dan flowchart metode *Location Based Service*. Pada perancangan ini, digunakan juga *UML (Unified Modelling Language)* yang meliputi *use case diagram*, *activity diagram*, *class diagram* dan *sequence diagram*.

#### **3.3.3. Konstruksi (*Construction*)**

Proses yang dilakukan pada tahap ini yaitu membangun aplikasi dengan perancangan yang telah dilakukan sebelumnya, mulai dari tampilan interface sampai implementasi rancangan UML. Proses yang juga dilakukan pada tahap ini yaitu penerapan coding metode *Location Based Service* pada sistem.

#### **3.3.4. Transisi (*Transition*)**

Pada tahap Transition difokuskan untuk melakukan proses pengujian terhadap aplikasi. Dalam penelitian ini, dilakukan pengujian menggunakan black box terhadap aplikasi yang meliputi pengujian keakuratan lokasi siswa/i yang diambil dari setiap *handphone* siswa/i dan dikirimkan ke *web server*.

## **BAB IV**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **4.1. Analisis Kebutuhan Sistem**

Analisis kebutuhan sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem dimana aplikasi dibangun, meliputi perangkat lunak, dan hasil analisis terhadap sistem serta elemen-elemen sistem. Pembahasan yang ada pada analisis sistem ini yaitu analisis kebutuhan fungsional dan analisis kebutuhan *nonfungsional*.

##### **4.1.1. Kebutuhan Fungsional**

Analisis kebutuhan fungsional adalah segala bentuk data yang dibutuhkan oleh sistem agar sistem dapat berjalan sesuai dengan prosedur yang dibangun. Setelah melalui tahapan analisis, maka telah ditetapkan kebutuhan-kebutuhan untuk membangun sistem meliputi *input*, proses, dan *output*.

##### **1. Analisis Kebutuhan Input**

*Input* dari aplikasi yang dibangun terdiri dari :

- a. Informasi data siswa/i SMAN 1 Wonggeduku.
- b. Informasi data guru SMAN 1 Wonggeduku.
- c. Informasi data mata pelajaran SMAN 1 Wonggeduku.
- d. Informasi jadwal mata pelajaran SMAN 1 Wonggeduku
- e. Informasi data Ruang/kelas SMAN 1 Wonggeduku

##### **2. Analisis Kebutuhan Proses**

Kebutuhan proses dalam aplikasi ini adalah proses *pembuatan aplikasi menggunakan Bahasa pemrograman PHP di sisi webservice* dan menggunakan Bahasa pemrograman *Dart di sisi aplikasi user*.

##### **3. Analisis Kebutuhan Output**

*Output* yang diperoleh dalam sistem ini adalah guru dapat mengetahui lokasi siswa/i saat sedang mengikuti proses pembelajaran dan aplikasi berbasis *android* untuk mengirimkan lokasi saat sedang mengikuti pembelajaran hingga akhir pembelajaran selesai dan mendata absensi siswa/i SMAN 1 Wonggeduku.

#### 4.1.2. Kebutuhan *Nonfungsional*

Analisis kebutuhan *nonfungsional* adalah sebuah langkah dimana seorang pembangun aplikasi menganalisis sumber daya yang dibutuhkan untuk membangun aplikasi yang dibangun. Analisis kebutuhan *nonfungsional* yang dilakukan dibagi dalam dua tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak.

##### 1. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan pada pembangunan sistem ini, sebagai berikut:

Tabel 4. 1 Spesifikasi Perangkat

No	Nama Perangkat	Spesifikasi
1.	<i>PC</i>	<i>Toshiba l-40</i>
2.	<i>Processor</i>	<i>Intel dual core</i>
3.	<i>RAM</i>	<i>4 GB</i>
4.	<i>Harddisk</i>	<i>128 GB SSD</i>
5.	<i>Monitor</i>	<i>Monitor 14 Inch</i>

##### 2. Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan pada pembangunan sistem ini, sebagai berikut:

Tabel 4. 2 Spesifikasi Perangkat Lunak

No.	Nama Perangkat	Fungsi	Spesifikasi
1.	<i>Mysql</i>	<i>Database Management</i>	<i>Mysql 5.7</i>
2.	<i>PHP</i>	<i>Web development</i>	<i>PHP 7.3</i>
3.	<i>Dart</i>	<i>Android Development</i>	<i>Dart 3.4.1</i>
4.	<i>Chrome</i>	<i>Web browser</i>	<i>Chrome version 76.0</i>
5.	<i>Android Studio</i>	<i>SDK Development Tool</i>	<i>Android Studio 3.3</i>
6.	<i>Visual Studio Code</i>	<i>Code Editor</i>	<i>VS Code v1.38</i>
7.	<i>Flutter</i>	<i>Framework Android</i>	<i>Flutter 3.4.1</i>

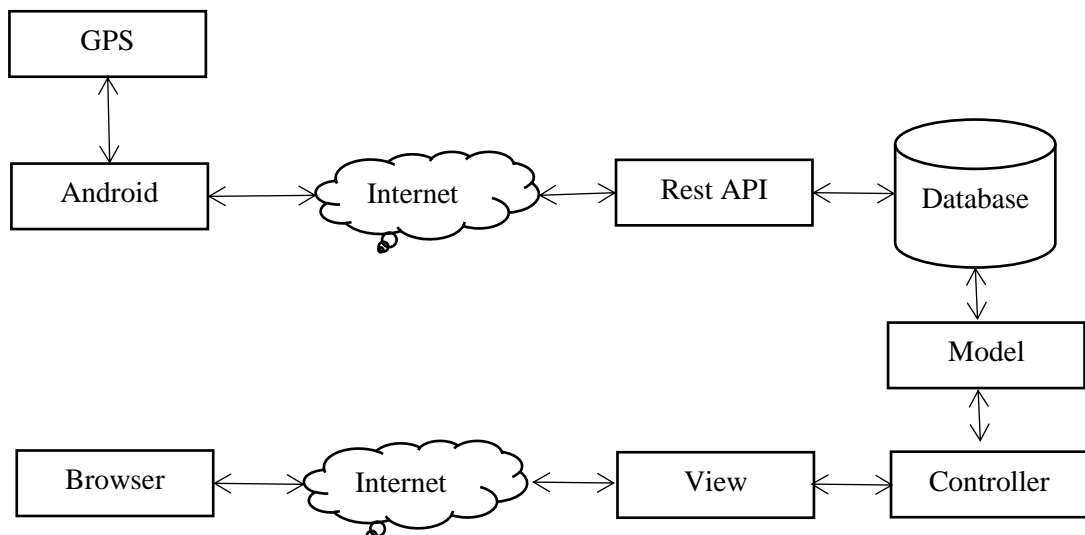


## 4.2. Analisis Perancangan Sistem

Perancangan sistem yang dibangun yaitu perancangan UML dan perancangan *user interface* sistem.

### 4.2.1. Arsitektur Sistem

Secara umum aplikasi ini dirancang untuk melakukan absensi yang dilakukan oleh siswa. Siswa melakukan absensi menggunakan aplikasi mobile pada lokasi yang telah ditentukan. Kemudian data tersebut dikirim ke server dan dengan menggunakan API yang telah dirancang untuk menerima data yang dikirim kemudian data disimpan di *database* yang ada pada sistem. Data hasil absensi yang telah tersimpan di database akan dapat dilihat oleh guru menggunakan aplikasi Web yang telah dirancang. Secara umum diagram sistem absensi dapat dilihat pada gambar 4.1 berikut

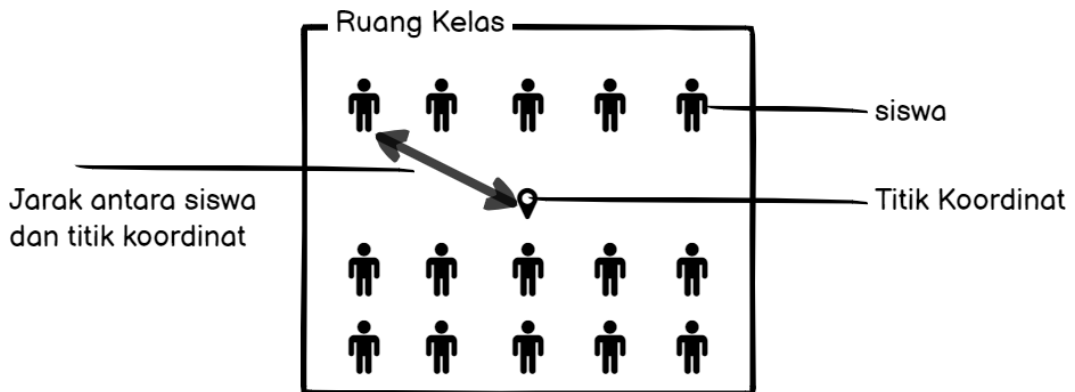


Gambar 4. 1 Arsitektur Sistem

### 4.2.2. Ilustrasi Algoritma

Pada gambar 4.2 merupakan ilustrasi dari algoritma haversine pada sistem monitoring kehadiran siswa di SMA 1 Wonggeduku. Algoritma ini berjalan ketika siswa melakukan absensi, sistem dengan otomatis menarik titik koordinat dari siswa dan membandingkan dengan titik koordinat kelas. Selanjutnya *haversine formula* akan mengukur jarak kedua titik tersebut jika jarak antara kedua titik melebihi 15 meter sistem akan menampilkan informasi “Absen gagal, Anda berada

diluar kelas”, dan ketika jarak berada kurang dari 15 meter sistem akan menampilkan informasi “Absen Berhasil.”



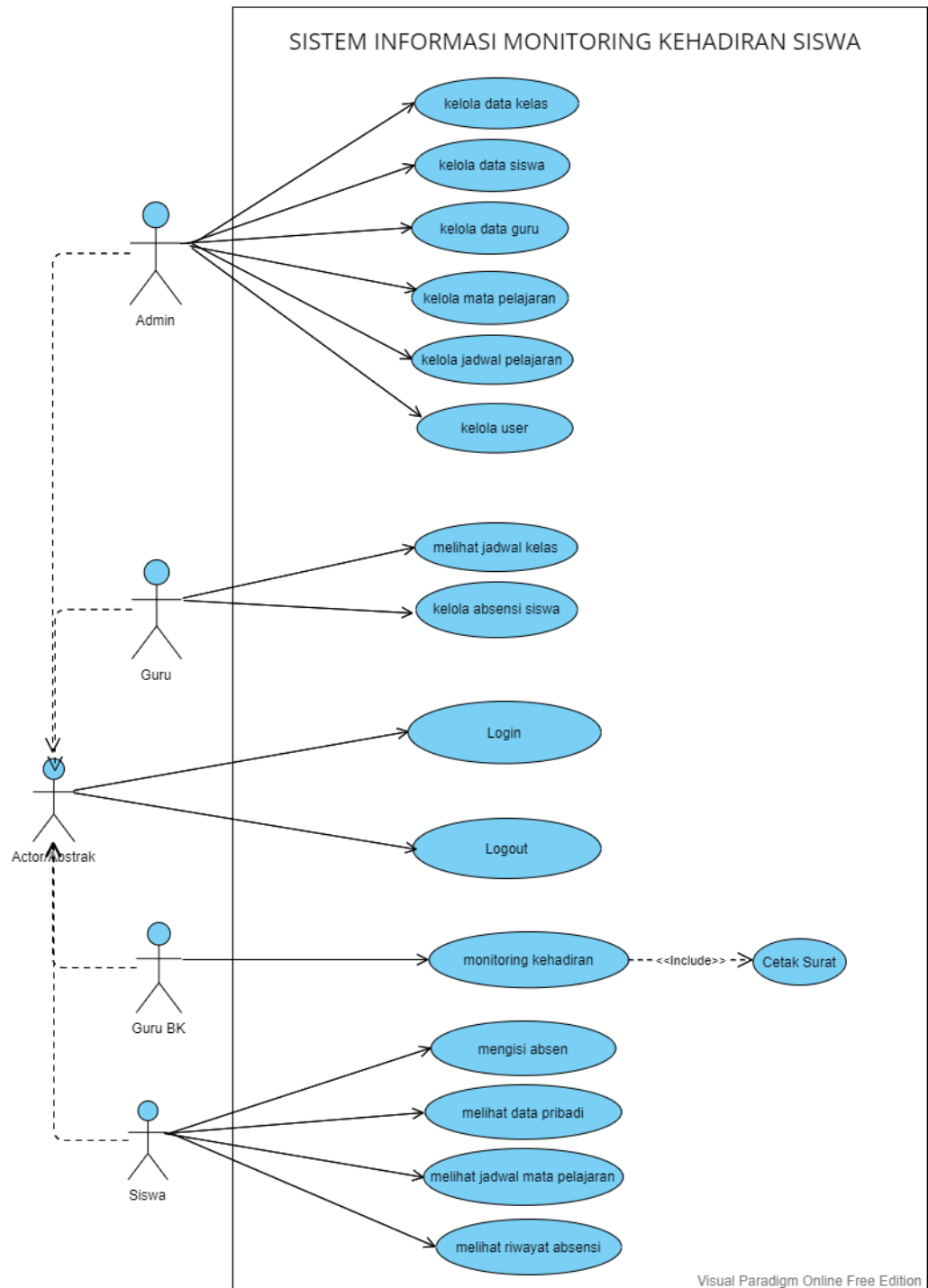
Gambar 4. 2 Ilustrasi Algoritma

#### 4.2.3. *Unified Modeling Language (UML)*

Aplikasi dibangun dengan menggunakan *Unified Modeling Language (UML)*. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*.

##### 1. *Use Case Diagram*

*Use Case Diagram* adalah sebuah diagram yang dapat merepresentasikan interaksi yang terjadi antara user dengan sistem. *Use Case Diagram* ini mendeskripsikan siapa saja yang menggunakan sistem dan bagaimana cara mereka berinteraksi dengan sistem. *Use Case Diagram* dari sistem yang akan dibangun adalah sebagai berikut.



Gambar 4. 3 Use Case Diagram

Tabel 4. 3 Deskripsi Case Login (Admin, Siswa, Guru dan Guru BK)

<b>Nama</b>	<b>Login</b>
Deskripsi Singkat:	Proses Keamanan login untuk <i>Admin</i> , guru, guru bk dan siswa.
Persyaratan	<i>Admin</i> , guru, guru bk dan siswa harus memasukkan <i>username</i> dan <i>password</i>
Kondisi Akhir	Masuk untuk manage sistem
Situasi Kesalahan	<ol style="list-style-type: none"> <li>i. Yang masuk bukan <i>Admin</i>, guru, guru bk dan siswa</li> <li>ii. Salah memasukkan <i>username</i> dan <i>password</i></li> </ol>
Status sistem dalam saat terjadi kesalahan	<ol style="list-style-type: none"> <li>1. Menampilkan pesan <i>error</i></li> <li>2. Tidak dapat masuk kedalam sistem</li> </ol>
Aktor	<i>Admin</i> , guru, guru bk dan siswa
Trigger	Keamanan <i>login</i> untuk <i>Admin</i> , guru, guru bk dan siswa
Proses Standar	<ol style="list-style-type: none"> <li>1. <i>Admin</i>, guru, guru bk dan siswa masuk ke sistem</li> <li>2. <i>Admin</i>, guru, guru bk dan siswa memasukkan <i>username</i></li> <li>3. <i>Admin</i>, guru, guru bk dan siswa memasukkan <i>password</i></li> </ol>
Proses Alternatif	Mengecek <i>username</i> dan <i>password</i>

Tabel 4. 4 Deskripsi Case Kelola Data Kelas

<b>Nama</b>	<b>Kelola Data Kelas</b>
Deskripsi Singkat:	<i>Admin</i> ingin melihat atau mengelolah data kelas
Persyaratan	<i>Admin</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Data kelas di tampilkan
Situasi Kesalahan	-

Tabel 4.4 Deskripsi *Case* Kelola Data Kelas (Lanjutan)

<b>Nama</b>	<b>Kelola Data Kelas</b>
Status sistem dalam saat terjadi kesalahan	-
Aktor	<i>Admin</i>
Trigger	<i>Admin</i> ingin menambahkan, menghapus atau mengedit data kelas
Proses Standar	<ol style="list-style-type: none"> <li>1. <i>Admin</i> masuk ke sistem dan telah melakukan proses <i>login</i></li> <li>2. <i>Admin</i> melihat data kelas</li> <li>3. <i>Admin</i> menambahkan data kelas</li> <li>4. <i>Admin</i> menghapus data kelas</li> <li>5. <i>Admin</i> mengedit data kelas</li> </ol>

Tabel 4. 5 Deskripsi *Case* Kelola Data Siswa

<b>Nama</b>	<b>Kelola Data Siswa</b>
Deskripsi Singkat:	<i>Admin</i> ingin melihat atau mengelolah data siswa
Persyaratan	<i>Admin</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Data siswa di tampilkan
Aktor	<i>Admin</i>
Trigger	<i>Admin</i> ingin menambahkan, menghapus atau mengedit data siswa
Proses Standar	<ol style="list-style-type: none"> <li>1. <i>Admin</i> masuk ke sistem dan telah melakukan proses <i>Login</i></li> <li>2. <i>Admin</i> melihat data siswa</li> <li>3. <i>Admin</i> menambahkan data siswa</li> <li>4. <i>Admin</i> menghapus data siswa</li> <li>5. <i>Admin</i> mengedit data siswa</li> </ol>

Tabel 4. 6 Deskripsi Case Kelola Data Guru

<b>Nama</b>	<b>Kelola Data Guru</b>
Deskripsi Singkat:	<i>Admin</i> ingin melihat atau mengelolah data guru
Persyaratan	<i>Admin</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Data guru di tampilkan
Situasi Kesalahan	-
Status sistem dalam saat terjadi kesalahan	-
Aktor	<i>Admin</i>
Trigger	<i>Admin</i> ingin menambahkan, menghapus atau mengedit data guru
Proses Standar	<ol style="list-style-type: none"> <li>1. <i>Admin</i> masuk ke sistem dan telah melakukan proses <i>Login</i></li> <li>2. <i>Admin</i> melihat data guru</li> <li>3. <i>Admin</i> menambahkan data guru</li> <li>4. <i>Admin</i> menghapus data guru</li> <li>5. <i>Admin</i> mengedit data guru</li> </ol>
Proses Alternatif	

Tabel 4. 7 Deskripsi Case Kelola Mata Pelajaran

<b>Nama</b>	<b>Kelola Mata Pelajaran</b>
Deskripsi Singkat:	<i>Admin</i> ingin melihat atau mengelolah mata pelajaran
Persyaratan	<i>Admin</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Mata pelajaran di tampilkan
Status sistem dalam saat terjadi kesalahan	-
Aktor	<i>Admin</i>

Tabel 4.7 Deskripsi Case Kelola Mata Pelajaran (Lanjutan)

<b>Nama</b>	<b>Kelola Mata Pelajaran</b>
Trigger	<i>Admin</i> ingin menambahkan, menghapus atau mengedit Mata Pelajaran
Proses Standar	<ol style="list-style-type: none"> <li>1. <i>Admin</i> masuk ke sistem dan telah melakukan proses <i>Login</i></li> <li>2. <i>Admin</i> melihat mata pelajaran</li> <li>3. <i>Admin</i> menambahkan mata pelajaran</li> <li>4. <i>Admin</i> menghapus mata pelajaran</li> <li>5. <i>Admin</i> mengedit mata pelajaran</li> </ol>
Proses Alternatif	

Tabel 4. 8 Deskripsi Case Kelola Jadwal Pelajaran

<b>Nama</b>	<b>Kelola Jadwal Pelajaran</b>
Deskripsi Singkat:	<i>Admin</i> ingin melihat atau mengelolah jadwal pelajaran
Persyaratan	<i>Admin</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Jadwal pelajaran di tampilkan
Aktor	<i>Admin</i>
Trigger	<i>Admin</i> ingin menambahkan, menghapus atau mengedit jadwal pelajaran
Proses Standar	<ol style="list-style-type: none"> <li>1. <i>Admin</i> masuk ke sistem dan telah melakukan proses <i>Login</i></li> <li>2. <i>Admin</i> melihat jadwal pelajaran</li> <li>3. <i>Admin</i> menambahkan jadwal pelajaran</li> <li>4. <i>Admin</i> menghapus jadwal pelajaran</li> <li>5. <i>Admin</i> mengedit jadwal pelajaran</li> </ol>
Proses Alternatif	

Tabel 4. 9 Deskripsi Case Kelola User

<b>Nama</b>	<b>Kelola User</b>
Deskripsi Singkat:	<i>Admin</i> ingin melihat atau mengelolah User
Persyaratan	<i>Admin</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	User di tampilkan
Situasi Kesalahan	-
Status sistem dalam saat terjadi kesalahan	-
Aktor	<i>Admin</i>
Trigger	<i>Admin</i> ingin menambahkan, menghapus atau mengedit user
Proses Standar	<ol style="list-style-type: none"> <li>1. <i>Admin</i> masuk ke sistem dan telah melakukan proses <i>Login</i></li> <li>2. <i>Admin</i> melihat user</li> <li>3. <i>Admin</i> menambahkan user</li> <li>4. <i>Admin</i> menghapus user</li> <li>5. <i>Admin</i> mengedit user</li> </ol>
Proses Alternatif	

Tabel 4. 10 Deskripsi Case Melihat Jadwal Kelas

<b>Nama</b>	<b>Melihat Jadwal Kelas</b>
Deskripsi Singkat:	<i>Guru</i> ingin melihat jadwal kelas
Persyaratan	<i>Guru</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Jadwal kelas di tampilkan
Aktor	Guru



Tabel 4.10 Deskripsi Case Melihat Jadwal Kelas (Lanjutan)

<b>Nama</b>	<b>Melihat Jadwal Kelas</b>
Trigger	Guru ingin melihat jadwal kelas
Proses Standar	<ol style="list-style-type: none"> <li>1. guru masuk ke sistem dan telah melakukan proses <i>Login</i></li> <li>2. Guru melihat jadwal kelas</li> </ol>
Proses Alternatif	

Tabel 4. 11 Deskripsi Case Kelola Absensi Siswa

<b>Nama</b>	<b>Mengisi Absensi Siswa Secara Manual</b>
Deskripsi Singkat:	<i>Guru</i> ingin mengelolah absensi siswa
Persyaratan	<i>Guru</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Absensi di tampilkan
Situasi Kesalahan	-
Status sistem dalam saat terjadi kesalahan	-
Aktor	<i>Guru</i>
Trigger	<i>Guru</i> ingin mengisi atau mengedit absensi siswa
Proses Standar	<ol style="list-style-type: none"> <li>1. <i>Guru</i> masuk ke sistem dan telah melakukan proses <i>Login</i></li> <li>2. <i>Guru</i> mengisi absensi siswa</li> <li>3. <i>Guru</i> mengedit absensi siswa</li> </ol>
Proses Alternatif	

Tabel 4. 12 Deskripsi Case Monitoring Kehadiran Siswa

<b>Nama</b>	<b>Monitoring Kehadiran</b>
Deskripsi Singkat:	<i>Guru BK memonitoring kehadiran siswa</i>
Persyaratan	<i>Guru BK harus melakukan login terlebih dahulu</i>
Kondisi Akhir	Menampilkan rekapitulasi siswa yang tidak masuk pada saat proses pembelajaran berlangsung
Aktor	<i>Guru BK</i>
Trigger	<i>Guru BK ingin melihat rekapitulasi siswa yang tidak masuk pada saat proses pembelajaran berlangsung dan mencetak surat</i>
Proses Standar	<ol style="list-style-type: none"> <li>1. <i>Guru BK masuk ke sistem dan telah melakukan proses Login</i></li> <li>2. <i>Guru BK melihat rekapitulasi</i></li> <li>3. <i>Guru BK mencetak surat</i></li> </ol>
Proses Alternatif	

Tabel 4. 13 Deskripsi Case Absensi Siswa

<b>Nama</b>	<b>Presensi</b>
Deskripsi Singkat:	Siswa melakukan absensi ketika pelajaran dimulai
Persyaratan	Mahasiswa masuk ke sistem absensi
Kondisi Akhir	Menampilkan halaman absensi data kehadiran siswa
Situasi Kesalahan	Lokasi siswa di luar jangkauan lokasi ruangan kelas
Status sistem dalam saat terjadi kesalahan	<ol style="list-style-type: none"> <li>1. Menampilkan pesan <i>error</i></li> <li>2. Tidak dapat melakukan absensi</li> </ol>

Tabel 4.13 Deskripsi *Case* Absensi Siswa (Lanjutan)

<b>Nama</b>	<b>Presensi</b>
Aktor	Siswa
Trigger	Siswa mengisi data absensi sesuai lokasi agar dianggap hadir dalam mata pelajaran.
Proses Standar	<ol style="list-style-type: none"> <li>1. Siswa memilih menu absensi</li> <li>2. Siswa memilih mata pelajaran untuk melakukan absensi</li> </ol>
Proses Alternatif	-

Tabel 4. 14 Deskripsi Case Melihat Data Pribadi

<b>Nama</b>	<b>Melihat Data Pribadi</b>
Deskripsi Singkat:	<i>Siswa</i> ingin melihat data pribadi
Persyaratan	<i>Siswa</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Data pribadi di tampilkan
Situasi Kesalahan	-
Status sistem dalam saat terjadi kesalahan	-
Aktor	siswa
Trigger	Siswa ingin melihat data pribadi
Proses Standar	<ol style="list-style-type: none"> <li>1. Siswa masuk ke sistem dan telah melakukan proses <i>Login</i></li> <li>2. Siswa melihat data pribadi</li> </ol>
Proses Alternatif	

Tabel 4. 15 Deskripsi Case Melihat Jadwal Mata Pelajaran

<b>Nama</b>	<b>Melihat Jadwal Mata Pelajaran</b>
Deskripsi Singkat:	<i>Siswa</i> ingin melihat jadwal mata pelajaran
Persyaratan	<i>Siswa</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Jadwal mata pelajaran di tampilkan
Situasi Kesalahan	-
Status sistem dalam saat terjadi kesalahan	-
Aktor	Siswa
Trigger	Siswa ingin melihat jadwal mata pelajaran
Proses Standar	<ol style="list-style-type: none"> <li>1. siswa masuk ke sistem dan telah melakukan proses <i>Login</i></li> <li>2. siswa melihat jadwal mata pelajaran</li> </ol>
Proses Alternatif	

Tabel 4. 16 Deskripsi Case Melihat Riwayat Absensi

<b>Nama</b>	<b>Melihat Riwayat Absensi</b>
Deskripsi Singkat:	<i>Siswa</i> ingin melihat riwayat absensi
Persyaratan	<i>Siswa</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Riwayat absensi di tampilkan
Aktor	siswa
Trigger	Siswa ingin melihat riwayat absensi

Tabel 4.16 Deskripsi *Case* Melihat Riwayat Absensi (Lanjutan)

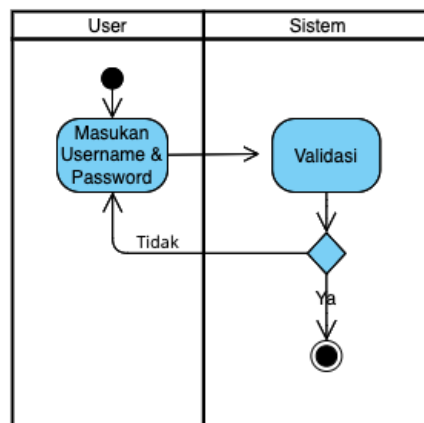
Nama	Melihat Riwayat Absensi
Proses Standar	1. siswa masuk ke sistem dan telah melakukan proses <i>Login</i>  2. siswa melihat riwayat absensi
Proses Alternatif	

## 2. Activity Diagram

*Activity* diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut ini adalah *activity* diagram yang akan menggambarkan alir aktivitas sistem.

### a. Activity Diagram Login

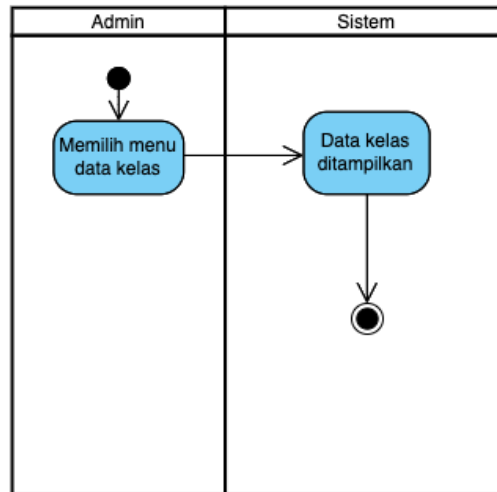
Pada Gambar 4.4 merupakan diagram aktivitas yang menunjukkan aktivitas *user* ketika memasukkan *username* dan *password*, lalu sistem akan menampilkan melakukan validasi.



Gambar 4. 4 Activity Diagram Login (User)

**b. Activity Diagram *Tampil Data Kelas***

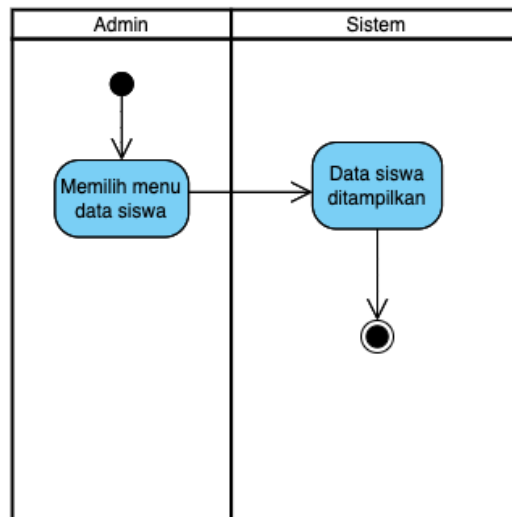
Pada Gambar 4.5 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu data kelas lalu sistem menampilkan halaman data kelas.



Gambar 4. 5 Activity Diagram Tampil Data Kelas

**c. Activity Diagram *Tampil Data Siswa***

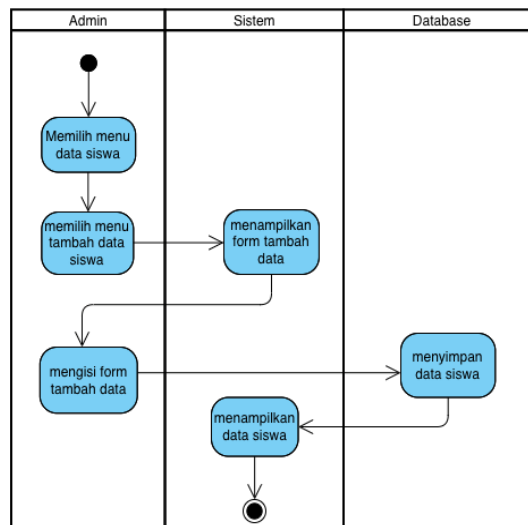
Pada Gambar 4.6 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu data siswa lalu sistem menampilkan halaman data siswa/i.



Gambar 4. 6 Activity Diagram Tampil Data Siswa

**d. Activity Diagram *Tambah Data Siswa/i***

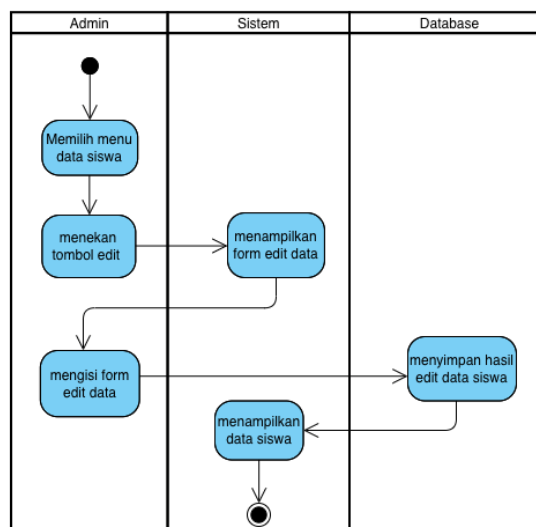
Pada Gambar 4.7 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu siswa/i lalu sistem menampilkan form tambah data siswa/i.



Gambar 4. 7 Activity Diagram Tambah Data Siswa/i

**e. Activity Diagram Edit Data siswa/i**

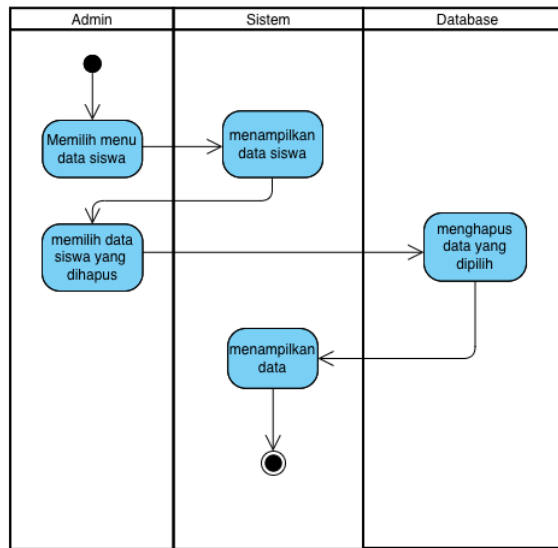
Pada Gambar 4.8 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu siswa/i lalu sistem menampilkan form edit data siswa/i.



Gambar 4. 8 Activity Diagram Edit Data Siswa/i

**f. Activity Diagram Hapus Data siswa/i**

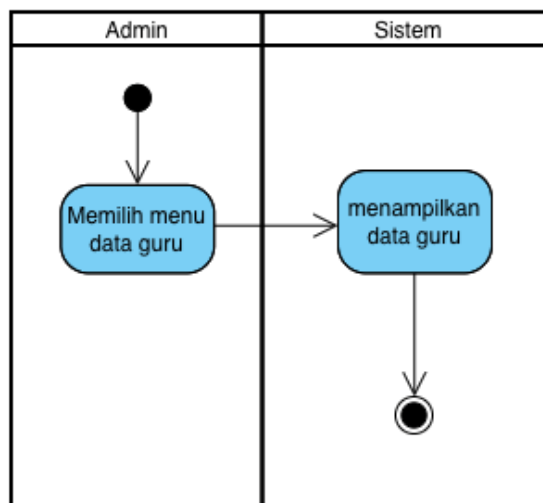
Pada Gambar 4.9 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu siswa/i lalu sistem menampilkan data siswa/i lalu *admin* memilih hapus kemudian sistem menghapus data siswa/i tersebut.



Gambar 4. 9 Activity Diagram Hapus Data siswa/i

**g. Activity Diagram Tampil Data Guru**

Pada Gambar 4.10 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu data guru lalu sistem menampilkan halaman data guru.

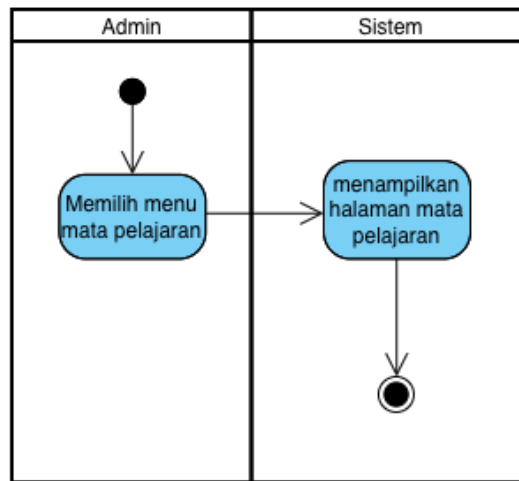


Gambar 4. 10 Activity Diagram Tampil Data Guru

**h. Activity Diagram Tampil Mata Pelajaran**

Pada Gambar 4.11 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mata pelajaran lalu sistem menampilkan halaman mata pelajaran .

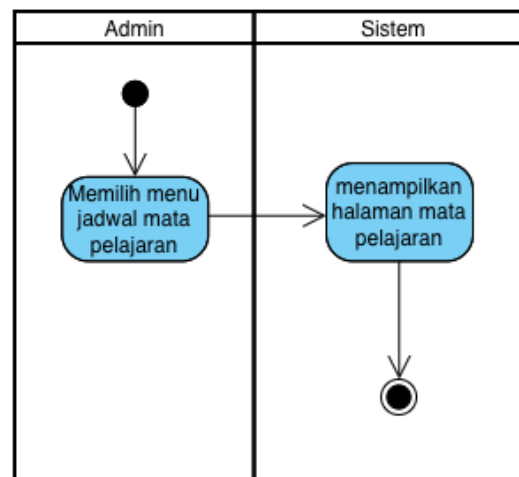




Gambar 4. 11 Activity Diagram Tampil Mata Pelajaran

**i. Activity Diagram Tampil Jadwal Mata Pelajaran**

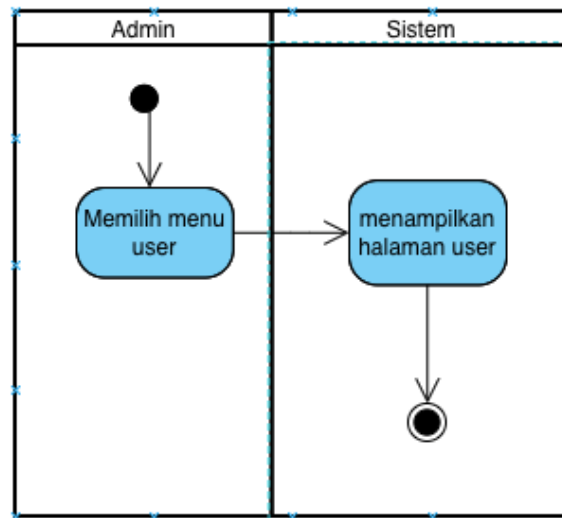
Pada Gambar 4.12 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu jadwal mata pelajaran lalu sistem menampilkan halaman jadwal mata pelajaran.



Gambar 4. 12 Activity Diagram Tampil Jadwal Mata Pelajaran

**j. Activity Diagram Tampil Halaman User**

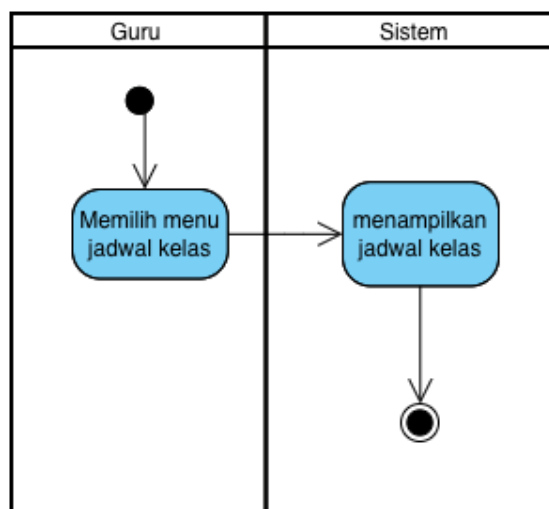
Pada Gambar 4.13 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu *user* lalu sistem menampilkan halaman *user*.



Gambar 4. 13 Activity Diagram Tampil Halaman User

**k. Activity Diagram Tampil Jadwal Kelas**

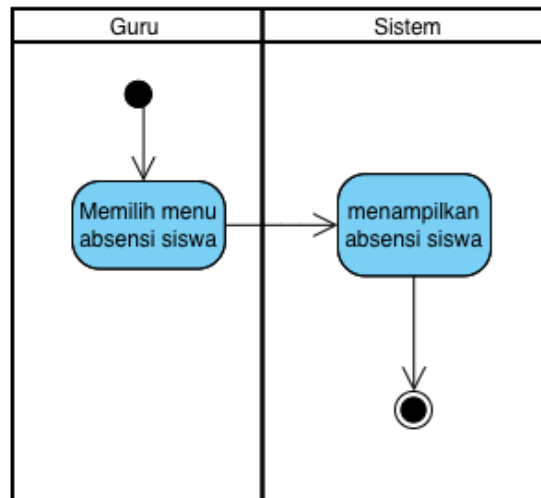
Pada Gambar 4.14 merupakan diagram aktivitas yang menunjukkan aktivitas guru memilih menu jadwal kelas lalu sistem menampilkan jadwal kelas.



Gambar 4. 14 Activity Diagram Tampil Jadwal Kelas

**l. Activity Diagram Tampil Absensi Siswa**

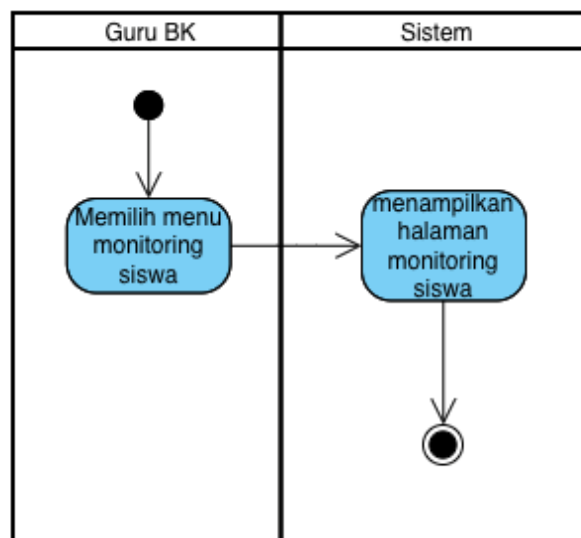
Pada Gambar 4.15 merupakan diagram aktivitas yang menunjukkan aktivitas guru memilih menu absensi lalu sistem menampilkan halaman data absensi siswa/i.



Gambar 4. 15 Activity Diagram Tampil Absensi Siswa

**m. Activity Diagram Tampil *Monitoring* Siswa**

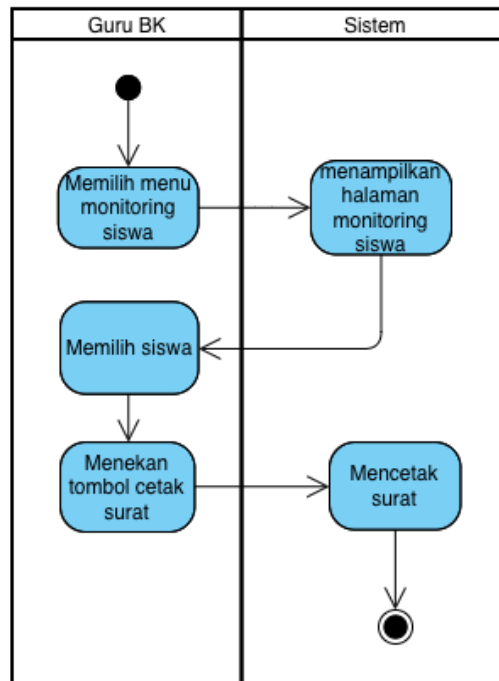
Pada Gambar 4.16 merupakan diagram aktivitas yang menunjukkan aktivitas guru BK memilih menu monitoring siswa/i lalu sistem menampilkan halaman *monitoring* absensi siswa/i.



Gambar 4. 16 Activity Diagram Tampil Monitoring Siswa

**n. Activity Diagram Cetak Surat**

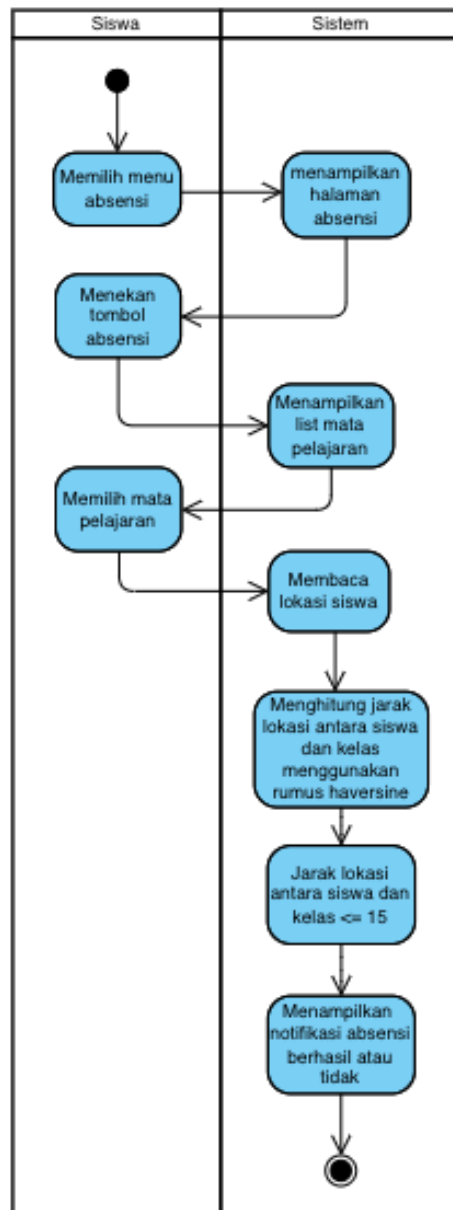
Pada Gambar 4.17 merupakan diagram aktivitas yang menunjukkan aktivitas guru BK pada saat melakukan cetak surat.



Gambar 4. 17 Activity Diagram Cetak Surat

#### o. *Activity Diagram Absensi*

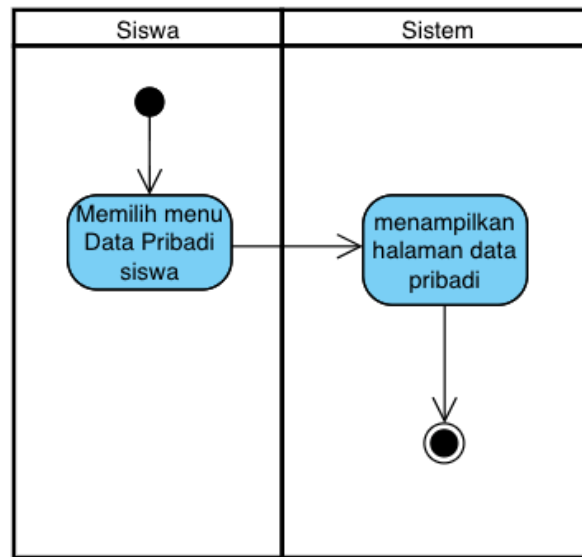
Pada Gambar 4.18 merupakan diagram aktivitas yang menunjukkan aktivitas siswa memilih menu absen lalu sistem akan menampilkan halaman absensi dan siswa akan melakukan absensi dgn menekan tombol absensi dan memilih mata pelajaran setelah itu sistem akan membaca lokasi siswa dan menghitung jarak lokasi antara siswa dan kelas kemudian sistem akan menampilkan notifikasi absensi berhasil atau tidak.



Gambar 4. 18 Activity Diagram Absensi

**p. Activity Diagram Tampil Data Pribadi Siswa**

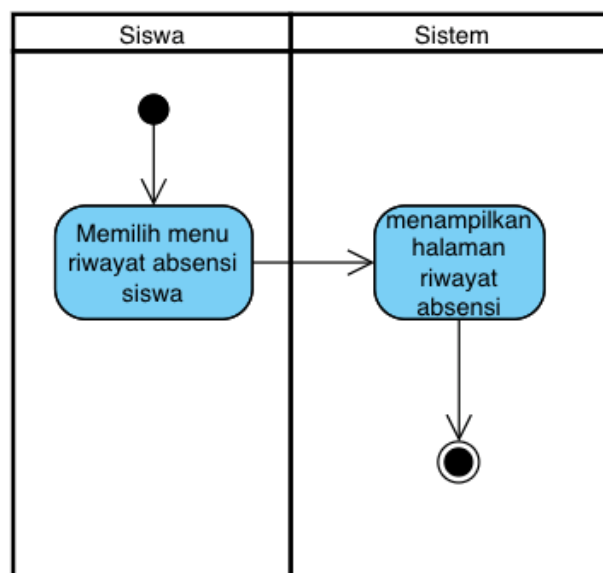
Pada Gambar 4.19 merupakan diagram aktivitas yang menunjukkan aktivitas siswa memilih menu data pribadi lalu sistem menampilkan halaman data pribadi.



Gambar 4. 19 Activity Diagram Tampil Data Pribadi Siswa

**q. Activity Diagram Tampil Riwayat Absensi Siswa**

Pada Gambar 4.20 merupakan diagram aktivitas yang menunjukkan aktivitas siswa memilih menu riwayat absensi lalu sistem menampilkan halaman riwayat absensi.

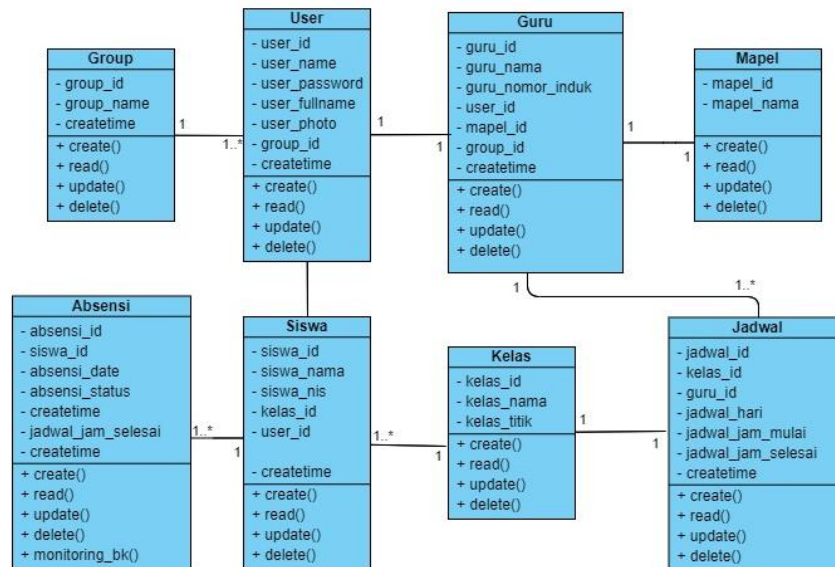


Gambar 4. 20 Activity Diagram Tampil Riwayat Absensi Siswa

### 3. Class Diagram

*Class diagram* merupakan diagram untuk menjelaskan pemodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem

yang sedang dibangun dan bagaimana mereka saling berinteraksi untuk mencapai suatu tujuan. Berikut ini adalah *class diagram system*.



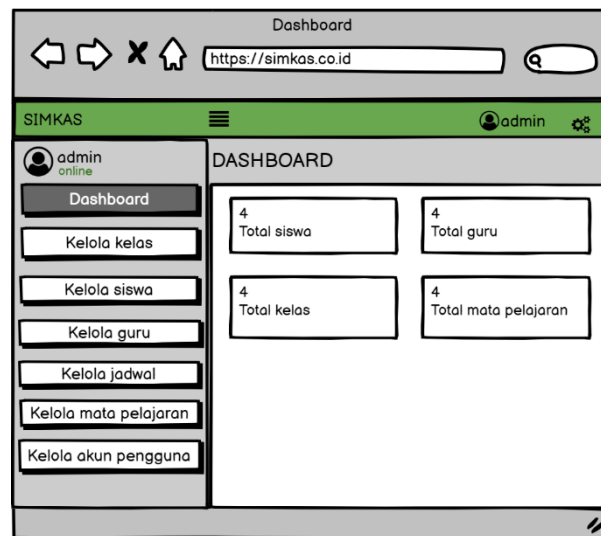
Gambar 4. 21 Class Diagram

#### 4.2.4. Perancangan Antarmuka *Interface*

Rancangan antarmuka pengguna atau *design user interface* merupakan penggambaran tampilan yang digunakan secara langsung oleh pengguna terhadap sistem. Rancangan *user interface* ini dibuat sederhana agar mudah dimengerti pengguna dan tidak ada kerumitan dalam menjalankannya sehingga mencapai tujuan perangkat lunak yang *user friendly*.

##### 1. Menu *Dashboard*

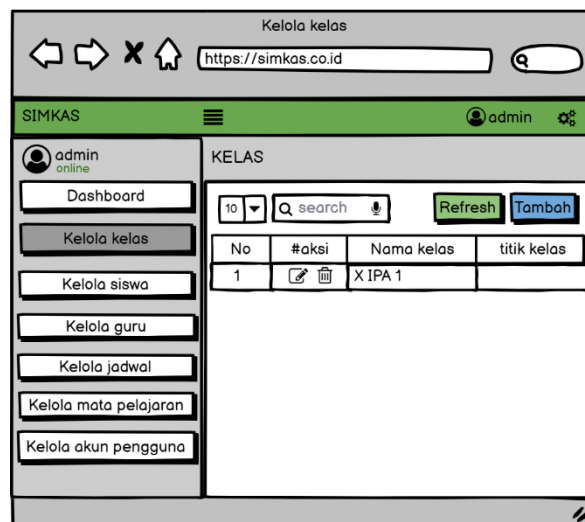
Menu *dashboard* menampilkan data aplikasi secara umum. Mulai dari jumlah siswa, guru, kelas dan mata pelajaran. Bentuk dari tampilan menu *dashboard* dapat dilihat pada gambar berikut.



Gambar 4. 22 Menu Dashboard

## 2. Menu Kelola Kelas

Menu kelola kelas berisikan data kelas dan ruangan di sekolah. Bentuk dari tampilan menu kelola kelas dapat dilihat pada gambar berikut.

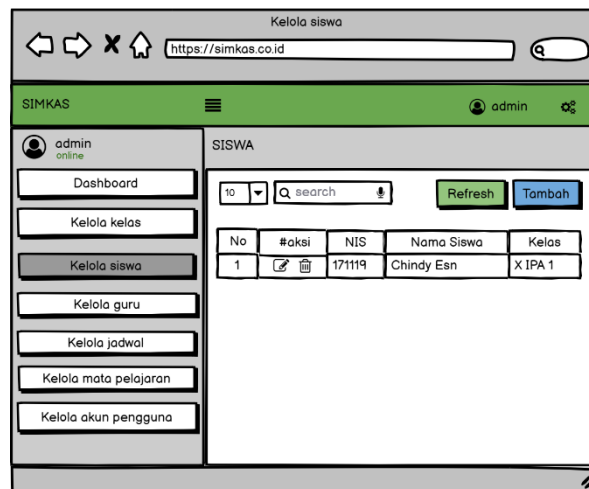


Gambar 4. 23 Menu Kelola Kelas

## 3. Menu Kelola Siswa/i

Menu kelola siswa/i berisikan data pribadi siswa/i. Bentuk dari tampilan menu kelola siswa dapat dilihat pada gambar berikut.

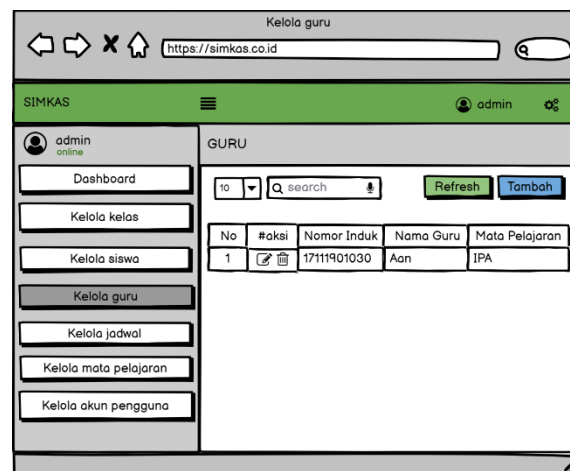




Gambar 4. 24 Menu Kelola Siswa/i

#### 4. Menu Kelola Guru

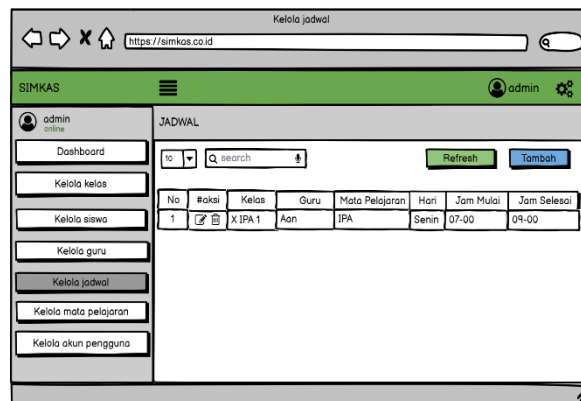
Menu kelola guru berisikan data guru serta mata pelajaran yang diajarkan. Bentuk dari tampilan menu kelola guru dapat dilihat pada gambar berikut.



Gambar 4. 25 Menu Kelola Guru

#### 5. Menu Kelola Jadwal

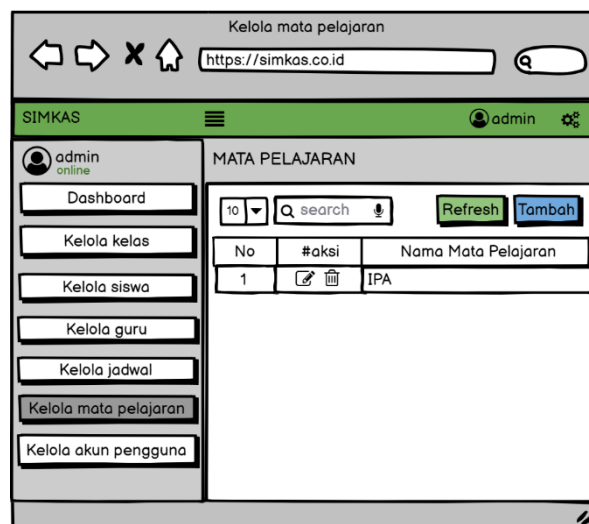
Menu kelola jadwal berisikan jadwal mata pelajaran guru pada kelas-kelas yang diajarkan. Bentuk dari tampilan menu kelola jadwal dapat dilihat pada gambar berikut.



Gambar 4. 26 Menu Kelola Jadwal

## 6. Menu Kelola Mata Pelajaran

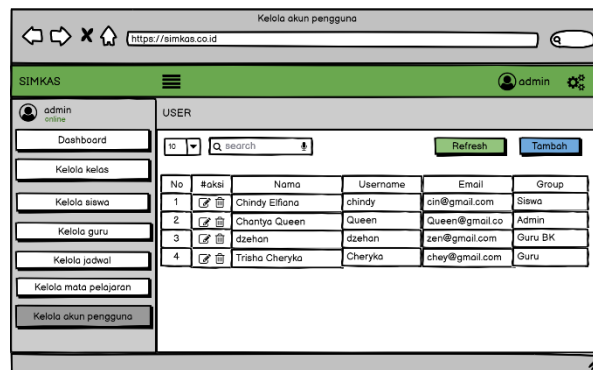
Menu kelola mata pelajaran berisikan data mata pelajaran yang diajarkan. Bentuk dari tampilan menu kelola mata pelajaran dapat dilihat pada gambar berikut.



Gambar 4. 27 Menu Mata Pelajaran

## 7. Menu Kelola Akun Pengguna

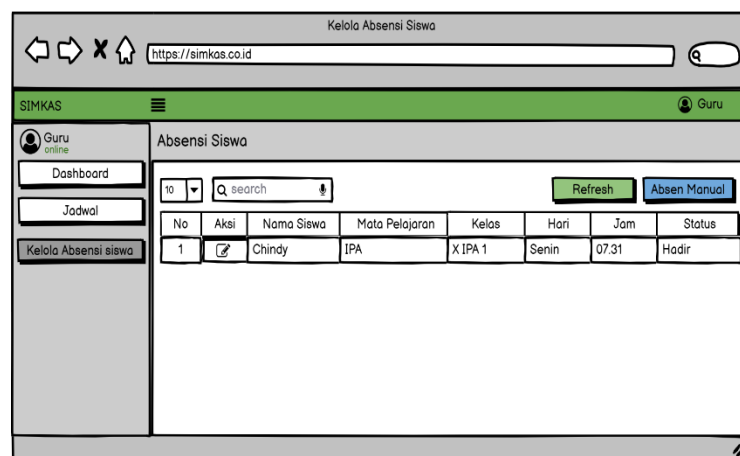
Menu kelola akun pengguna berisikan data akun pengguna aplikasi. Bentuk dari tampilan menu kelola akun pengguna dapat dilihat pada gambar berikut.



Gambar 4. 28 Menu Akun Pengguna

## 8. Menu Kelola Absensi Siswa

Menu kelola absensi siswa berisikan data kehadiran siswa yang telah melakukan absensi. Bentuk dari tampilan menu kelola absensi siswa dapat dilihat pada gambar berikut.



Gambar 4. 29 Menu Absensi Siswa

## 9. Form Absensi Manual Siswa

Bentuk dari tampilan form absensi manual siswa dapat dilihat pada gambar berikut.

Kelola Absensi Siswa

https://simkas.co.id

SIMKAS

Guru online

Absensi Siswa

to search

Refresh Absen Manual

No	Aksi	Nama Siswa	Hari	Jam	Status
1		Chindy	Senin	07.31	Hadir

Absen Manual

Nama Siswa: Chindy

Hari: Selasa

Status Kehadiran: Hadir

Izin: Izin Sakit Alpa

Batal Simpan

Gambar 4. 30 Form Absensi Manual Siswa

## 10. Form Edit Absensi Siswa

Bentuk dari *form* edit absensi siswa dapat dilihat pada gambar berikut.

Kelola Absensi Siswa

https://simkas.co.id

SIMKAS

Guru online

Absensi Siswa

to search

Refresh Absen Manual

No	Aksi	Nama Siswa	Hari	Jam	Status
1		Chindy	Senin	07.31	Hadir

Edit Kehadiran Siswa

Nama Siswa: Chindy

Hari: Senin

Status Kehadiran: Hadir

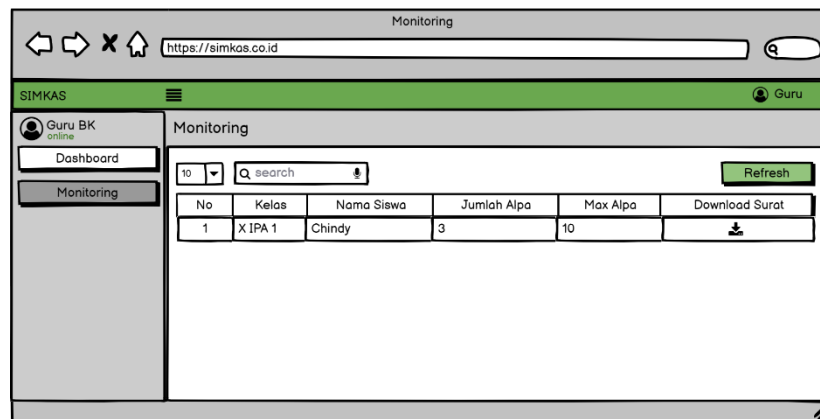
Izin: Izin Sakit Alpa

Batal Simpan

Gambar 4. 31 Form Edit Absensi Siswa

## 11. Menu Monitoring

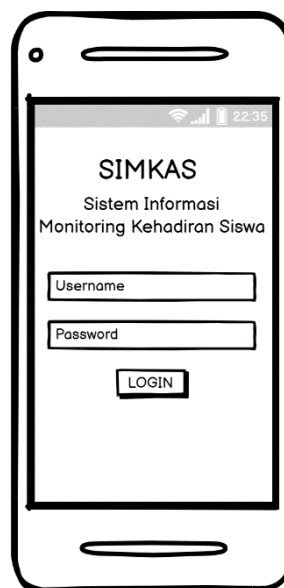
Menu monitoring berisikan data kehadiran siswa yang tidak melakukan absensi. Bentuk dari tampilan menu monitoring dapat dilihat pada gambar berikut.



Gambar 4. 32 Menu Monitoring

## 12. Halaman *Login Mobile*

Halaman login pada aplikasi *mobile* menampilkan *form login* yang terdiri dari *username* dan *password*. Setiap *field* harus diisi agar dapat melakukan *login*. Selain itu, *form* ini juga telah ditambahkan fungsi validasi agar dapat menyaring data *user* yang benar. Bentuk dari tampilan *login mobile* dapat dilihat pada gambar berikut.



Gambar 4. 33 Halaman Login Mobile

## 13. Halaman *Home Mobile*

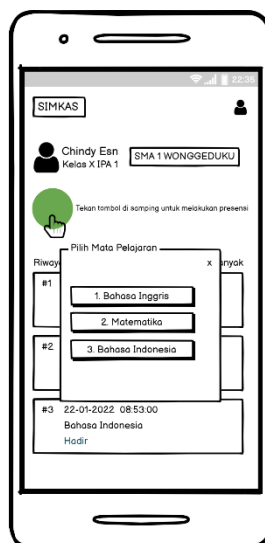
Halaman *home mobile* menampilkan identitas dari siswa, tombol absensi dan riwayat absensi. Bentuk dari tampilan *home mobile* dapat dilihat pada gambar berikut.



Gambar 4. 34 Halaman Home Mobile

#### 14. Halaman Mata Pelajaran *Mobile*

Halaman mata pelajaran *mobile* menampilkan daftar mata pelajaran. Bentuk dari tampilan mata pelajaran *mobile* dapat dilihat pada gambar berikut.



Gambar 4. 35 Halaman Mata Pelajaran Mobile

## 15. Notifikasi Absensi Berhasil

Adapun bentuk dari tampilan notifikasi absensi berhasil dapat dilihat pada gambar berikut.



Gambar 4. 36 Notifikasi Absensi Berhasil

## BAB V

### IMPLEMENTASI DAN PEMBAHASAN

#### 5.1 Implementasi Sistem

Pada tahap Implementasi merupakan tahap penerapan kode program yang dilakukan untuk membuat aplikasi berdasarkan rancangan dan desain sistem. Pada tahap ini akan dijelaskan mengenai data yang digunakan pada sistem, implementasi *interface*, implementasi algoritma *Haversine* dan pengujian sistem.

#### 5.2 Data

Data yang digunakan pada pengembangan sistem absensi berbasis *Android* dan *Web* adalah data diri dari siswa dan guru SMA 1 Wonggeduku, yang didapatkan dari staf tata usaha.

#### 5.3 Implementasi Interface

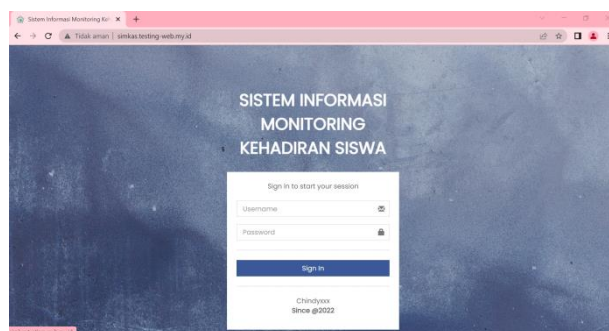
Implementasi *interface* dibuat berdasarkan perancangan admin, guru dan siswa. Implementasi disesuaikan dengan komponen yang tersedia pada *framework Android* dan *Web*. Implementasi *interface admin*, guru dan siswa akan ditampilkan dalam bentuk *screen shoot Android* dan *web*.

##### 5.3.1 Implementasi Interface System Berbasis Web

#### 1. Tampilan Interface Admin

##### a. Halaman Login

Pada Gambar 5.1 merupakan halaman *login admin* memasukkan *username* dan *password* terlebih dahulu untuk bisa masuk kedalam sistem.

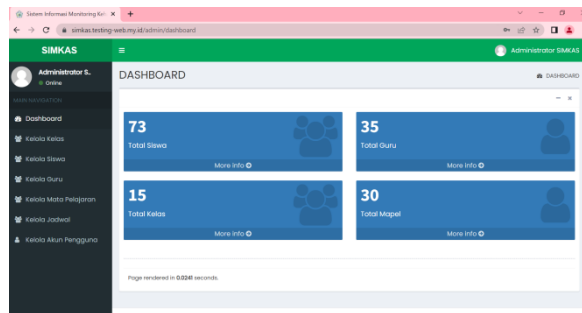


Gambar 5. 1 Halaman Login



### b. Halaman *Dashboard*

Pada Gambar 5.2 merupakan halaman pertama setelah *user* berhasil melakukan *login* kedalam sistem. Halaman ini berisi tentang jumlah guru, mahasiswa, mata pelajaran dan jumlah *user*



Gambar 5. 2 Halaman *Dashboard*

### c. Halaman *Kelola Kelas*

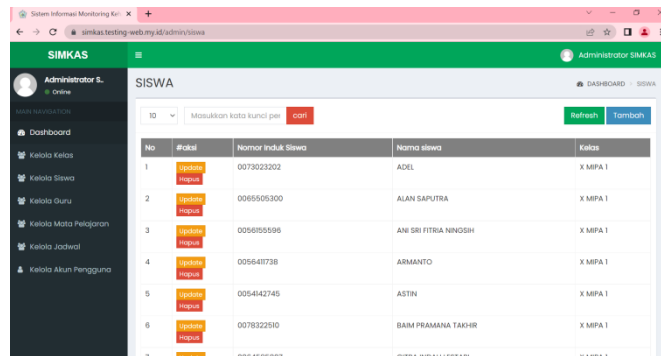
Pada Gambar 5.3 merupakan halaman kelola kelas setelah *admin* mengklik menu kelola kelas. Halaman ini berisi tentang data kelas dan juga beberapa tombol yaitu tombol tambah data,, *refresh* halaman, hapus dan *update* data kelas.

No	#Detail	Nama kelas	TSK kelas
1		X MIPA 1	~4.004645, 122.583343
2		X MIPA 2	-
3		X IPS 1	-
4		X IPS 2	-
5		X IPS 3	-
6		X IPS 1	-

Gambar 5. 3 Halaman *Kelola Kelas*

### d. Halaman *Kelola Siswa*

Pada Gambar 5.4 merupakan halaman kelola siswa setelah *admin* mengklik menu kelola siswa. Halaman ini berisi tentang data siswa dan juga beberapa tombol yaitu tombol tambah data, *refresh* halaman, hapus dan *update* data siswa.

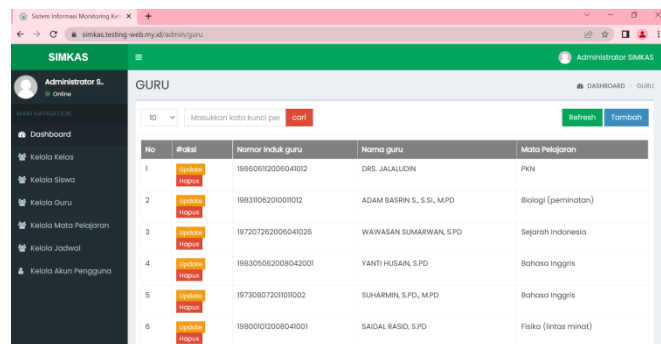


No	#Kelas	Nomor Induk Siswa	Nama siswa	Kelas
1	Update	0073023202	ADEL	X MIPA 1
2	Update	0065505300	ALAN SAPUTRA	X MIPA 1
3	Update	0056055596	ANI SRI FITRIA NINGSIH	X MIPA 1
4	Update	005640738	ARMANTO	X MIPA 1
5	Update	005442745	ASTIN	X MIPA 1
6	Update	0078322610	BAIM PRAMANA TAKHIR	X MIPA 1
7	Update	0064006807	CITRA INFANIL FETARI	X MIPA 1

Gambar 5. 4 Halaman Kelola Siswa

#### e. Halaman Kelola Guru

Pada Gambar 5.5 merupakan halaman kelola guru setelah admin mengklik menu kelola guru. Halaman ini berisi tentang data guru dan juga beberapa tombol yaitu tombol tambah data, *refresh* halaman, hapus dan update data guru.

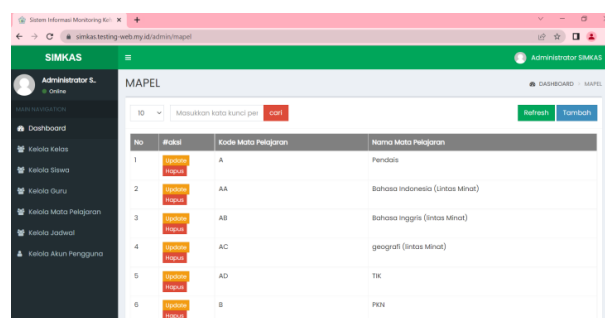


No	#Kelas	Nomor Induk guru	Nama guru	Mata Pelajaran
1	Update	1966081200604012	Drs. JALALUDIN	PKN
2	Update	1983106201001012	ADAM BASRI S., S.Si., M.Pd	Biologi (peminatan)
3	Update	19720726200604026	WAWASAN SUMARWAN, S.Pd	Sejarah Indonesia
4	Update	198305062008042001	YANTI HUSAIN, S.Pd	Bahasa Inggris
5	Update	197308072010101002	SUHARMIN, S.Pd., M.Pd	Bahasa Inggris
6	Update	19800102008040101	SAIDAL RASID, S.Pd	Fisika (Intas minat)

Gambar 5. 5 Halaman Kelola Guru

#### f. Halaman Kelola Mata Pelajaran

Pada Gambar 5.6 merupakan halaman kelola mata pelajaran setelah admin mengklik menu kelola mata pelajaran. Halaman ini berisi tentang data mata pelajaran dan juga beberapa tombol yaitu tombol tambah data, *refresh* halaman, hapus dan *update* data mata mata pelajaran.



No	#Kelas	Kode Mata Pelajaran	Nama Mata Pelajaran
1	Update	A	Pendidikan
2	Update	AA	Bahasa Indonesia (Intas Minat)
3	Update	AB	Bahasa Inggris (Intas Minat)
4	Update	AC	geografi (Intas Minat)
5	Update	AD	TIK
6	Update	B	PKN

Gambar 5. 6 Halaman Kelola Mata Pelajaran

### g. Halaman Kelola Jadwal

Pada Gambar 5.7 merupakan halaman kelola jadwal setelah *admin* mengklik menu kelola jadwal. Halaman ini berisi tentang data jadwal mata pelajaran dan juga beberapa tombol yaitu tombol tambah data, *refresh* halaman, hapus, *update* data jadwal.

No	#Kelas	Kelas	Guru	Mata Pelajaran	Hari	Jam Mulai	Jam Selesai
1		X MIPA 1	ADAM BASRIN S., S.SI, MPD	Biologi (peminatan)	Senin	07:45:00	09:30:00
2		X MIPA 1	SANDRAWATI, S.PD	Matematika (peminatan)	Senin	09:31:00	11:45:00
3		X MIPA 1	TINO SAPUTRA, S.PD	Matematika (Wajib)	Senin	11:46:00	12:55:00
4		X MIPA 1	TINO SAPUTRA, S.PD	Matematika (Wajib)	Selasa	07:30:00	08:40:00
5		X MIPA 1	YANTI HUSAIN, S.PD	Bahasa Inggris	Selasa	08:41:00	09:50:00
6		X MIPA 1	WAWASAN SUMARWAN, S.PD	Sejarah Indonesia	Selasa	10:25:00	11:35:00

Gambar 5. 7 Halaman Kelola Jadwal

### h. Halaman Kelola Akun Pengguna

Pada Gambar 5.8 merupakan halaman kelola akun pengguna setelah *admin* mengklik menu kelola akun pengguna. Halaman ini berisi tentang data akun pengguna dan juga terdapat beberapa tombol yaitu tombol tambah data, *refresh* halaman, hapus, *update* data akun pengguna.

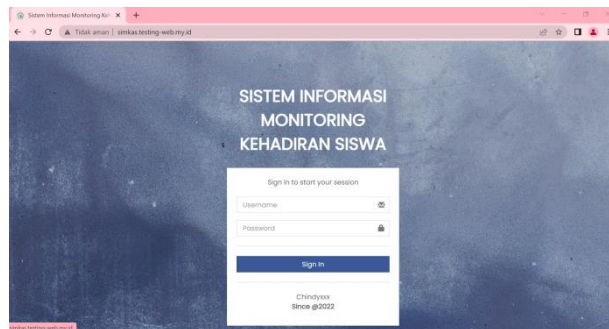
No	#Kelas	Nama	Username	Email	Group
1		Administrator SIMKAS	administrator	admin_simkas@gmail.com	Administrator
2		ADEL	0073023202		Siswa
3		ALAN SAPUTRA	0065505300		Siswa
4		ANI SRI FITRIA NINGSIH	0056155596		Siswa
5		ARMANTO	005641738		Siswa
6		ASTIN	005442745		Siswa
7		BAHM PRAMANA TAKHIR	0078322510		Siswa
8		CITRA INDAH LESTARI	0064595807		Siswa
9		DEBY EFRIDA	0051949378		Siswa
10		DELA SAPITRI	0063199632		Siswa

Gambar 5. 8 Halaman Kelola Akun Pengguna

## 2. Tampilan *Interface* Guru

### a. Halaman *Login*

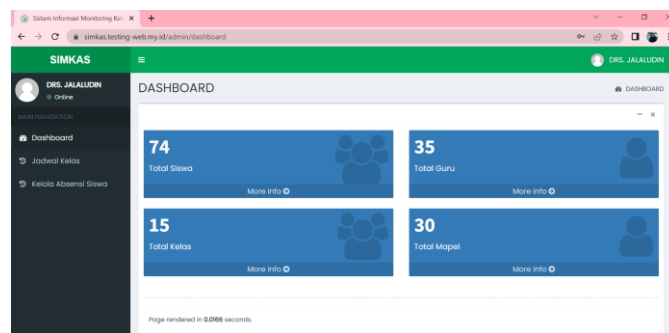
Pada Gambar 5.9 merupakan halaman *login*. Guru memasukkan *username* dan *password* terlebih dahulu untuk bisa masuk kedalam sistem.



Gambar 5. 9 Halaman Login

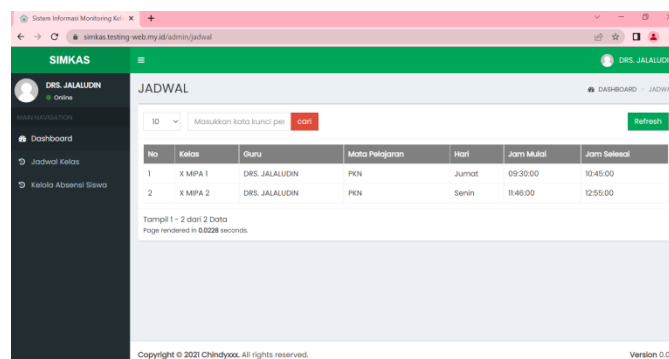
### b. Halaman *Dashboard*

Pada Gambar 5.10 merupakan halaman pertama setelah *user* berhasil melakukan *login* kedalam sistem. Halaman ini berisi tentang jumlah guru, jumlah siswa, mata pelajaran dan jumlah *user*

Gambar 5. 10 Halaman *Dashboard*

### c. Halaman *Jadwal*

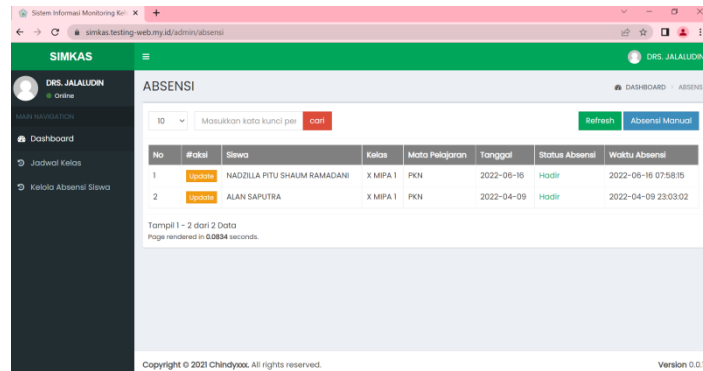
Pada Gambar 5.11 merupakan halaman jadwal setelah guru mengklik menu jadwal. Halaman ini berisi tentang data jadwal mata pelajaran dan juga tombol *refresh* halaman.



Gambar 5. 11 Halaman Jadwal

#### d. Halaman Kelola Absensi Siswa

Pada Gambar 5.12 merupakan halaman kelola absensi siswa setelah *guru* mengklik menu kelola absensi siswa. Halaman ini berisi mengenai data siswa yang telah melakukan absensi dan juga terdapat beberapa tombol yaitu tombol absensi manual, *refresh* halaman, dan *update* data absensi

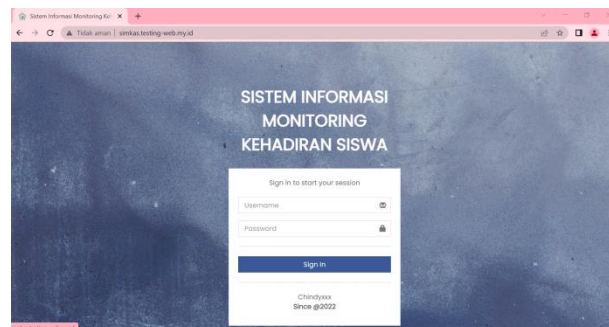


Gambar 5. 12 Halaman Kelola Absensi Siswa

### 3. Tampilan *Interface* Guru Bimbingan konseling (BK)

#### a. Halaman *Login*

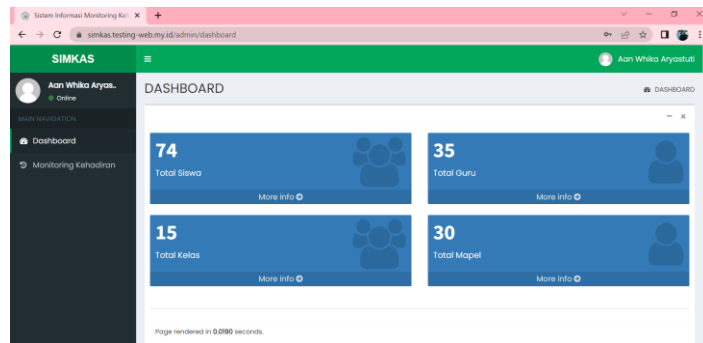
Pada Gambar 5.13 merupakan halaman *login*. Guru BK memasukkan *username* dan *password* terlebih dahulu untuk bisa masuk kedalam sistem.



Gambar 5. 13 Halaman *Login*

#### b. Halaman *Dashboard*

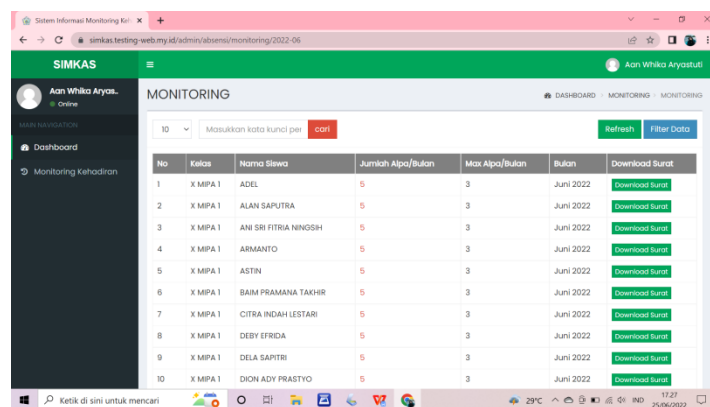
Pada Gambar 5.14 merupakan halaman pertama setelah *user* berhasil melakukan *login* kedalam sistem. Halaman ini berisi tentang jumlah guru, jumlah siswa, mata pelajaran dan jumlah *user*



Gambar 5. 14 Dashboard

### c. Halaman Monitoring

Pada Gambar 5.15 merupakan halaman monitoring setelah guru bk mengklik menu monitoring . Halaman ini berisi tentang data jumlah alpa siswa dan juga terdapat beberapa tombol yaitu tombol *refresh* halaman dan download surat.

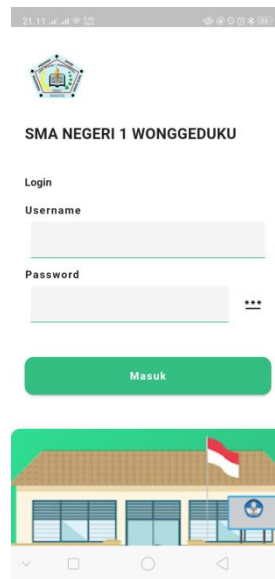


Gambar 5. 15 Halaman Monitoring

## 5.3.2 Implementasi Interface System Berbasis Mobile

### 1. Halaman Login

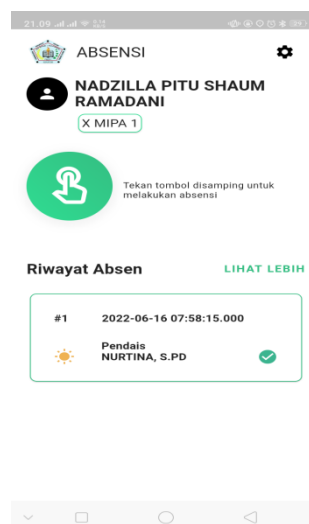
Pada Gambar 5.16 merupakan halaman *login* dimana siswa memasukkan *username* dan *password* terlebih dahulu untuk bisa masuk kedalam sistem.



Gambar 5. 16 Halaman Login

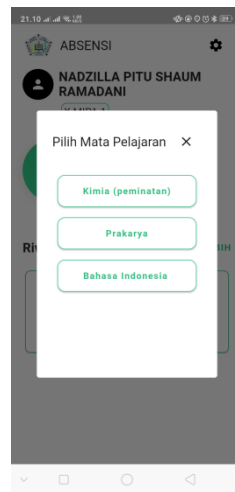
## 2. Halaman *Dashboard*

Pada gambar 5.17 merupakan tampilan halaman *Dashboard* . halaman *Dashboard* ini merupakan halaman setelah siswa berhasil melakukan *login* kedalam sistem. Halaman ini berisi tombol absensi dan riwayat absensi.

Gambar 5. 17 Halaman *Dashboard*

## 3. Tampilan halaman Absensi

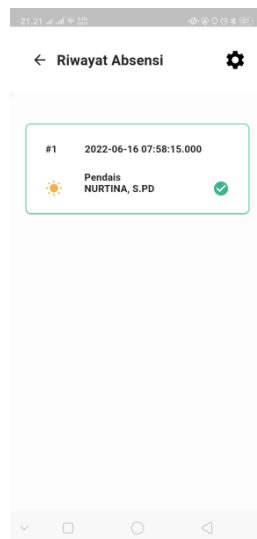
Pada Gambar 5.18 merupakan tampilan halaman absensi setelah siswa menekan tombol absensi. Halaman ini berisi jadwal mata pelajaran sesuai dengan hari tersebut.



Gambar 5. 18 Tampilan halaman Absensi

#### 4. Tampilan Halaman Riwayat Absensi

Pada Gambar 5.19 merupakan tampilan halaman riwayat yang berisi tentang riwayat absensi yang telah dilakukan oleh siswa.



Gambar 5. 19 Tampilan Halaman Riwayat Absensi

#### 5.5 Pengujian Sistem

Pengujian merupakan tahap yang utama dalam pembuatan suatu aplikasi. Hasil dari pengujian yang didapat, akan dijadikan sebagai tolak ukur dalam proses pengembangan sistem selanjutnya. Pada pengujian ini dilakukan untuk mengetahui hasil dari pembuatan sistem



### 5.5.1 Pengujian *Black Box*

Pengujian ini dilakukan dengan menguji perangkat lunak dari segi fungsionalitas-nya. Pada fungsionalitas perangkat lunak ini diuji sesuai dengan skenario pada tahap *desain* sistem.

Tabel 5. 1 Pengujian *Black Box*

<b>Input</b>	<b>Output</b>	<b>Hasil</b>
Memilih Menu Dashboard	Menampilkan Menu Dashboard	Berhasil
Memilih menu kelola kelas	Menampilkan menu kelola kelas	Berhasil
Memilih menu kelola siswa	Menampilkan menu kelola kelas	Berhasil
Memilih menu kelola guru	Menampilkan menu kelola guru	Berhasil
Memilih menu kelola mata pelajaran	Menampilkan menu kelola mata pelajaran	Berhasil
Memilih menu kelola jadwal	Menampilkan menu kelola jadwal	Berhasil
Memilih menu kelola akun pengguna	Menampilkan menu kelola akun pengguna	Berhasil
Memilih menu jadwal kelas	Menampilkan menu kelola jadwal	Berhasil
Memilih menu kelola absensi siswa	Menampilkan menu absensi manual	Berhasil
Memilih menu monitoring kehadiran	Menampilkan menu monitoring kehadiran	Berhasil
Menekan tombol absensi	Menampilkan jadwal pelajaran	Berhasil

Tabel 5.1 Pengujian *Black Box* (Lanjutan)

Input	Output	Hasil
Menekan jadwal pelajaran	Menampilkan notifikasi absensi berhasil	Berhasil
Menekan riwayat absen	Menampilkan riwayat absen	Berhasil

### 5.5.2 Pengujian Hitungan Manual

Diketahui :  $r = 6371$  (km)

$$\text{lat}_1 = -3.924648417827858$$

$$\text{lat}_2 = -3.924736054170059$$

$$\text{long}_1 = 122.15749695556916$$

$$\text{long}_2 = 122.1574681218231$$

$$\Delta\text{lat} = 0,00008763$$

$$\Delta\text{long} = 0,00002883$$

$$a = \sin^2\left(\frac{1,52954 \cdot 10^{-6}}{2}\right) + \cos(-0,068498037) \cdot \cos(-0,068499566) \cdot \sin^2\left(\frac{5,03244 \cdot 10^{-7}}{2}\right)$$

$$a = \sin^2(7,64771 \cdot 10^{-7}) + \cos(-0,068498037) \cdot \cos(-0,068499566) \cdot \sin^2(5,03244 \cdot 10^{-7})$$

$$a = 5,84875 \cdot 10^{-13} + 0,997654927 \cdot 0,997654822 \cdot 6,33136 \cdot 10^{-14}$$

$$a = 5,84875 \cdot 10^{-13} + 6,3017 \cdot 10^{-14}$$

$$a = 6,47892 \cdot 10^{-13}$$

$$d = 2.6371, 1. \arcsin(\sqrt{6,47892 \cdot 10^{-13}})$$

$$d = 0,010256 \text{ km}$$

$$d = 10.256 \text{ m}$$

Tabel 5. 2 Pengujian Jarak Haversine Dan Jarak Sesungguhnya

No	Nama Siswa	Titik Koordinat Siswa	Titik Koordinat Kelas	Jarak Haversine (M)	Jarak Sesungguhnya (M)	Galat (%)
1.	Nadzilla Pitu Shaum Ramadanani	-3.924648417827858, 122.15749695556916	-3.924736054170059, 122.1574681218231	10.26	10	2.60
2.	Adel	-3.9246704941592094, 122.1574915911513	-3.924736054170059, 122.1574681218231	7.74	7.79	0.64
3.	Alan Saputra	-3.9246263414959346, 122.15750768440488	-3.924736054170059, 122.1574681218231	12.96	12.99	0.23
4.	Armanto	-3.9246612492143527, 122.15749691832136	-3.924736054170059, 122.1574681218231	8.91	8.99	0.89
5.	Astin	-3.924690115211184, 122.15748547935762	-3.924736054170059, 122.1574681218231	5.46	5.60	2.50
6.	Deby Efrida	-3.9246149293569053, 122.15751912336863	-3.924736054170059, 122.1574681218231	14.61	14.58	0.21
7.	Dela Sapitri	-3.924619628472992, 122.15751374032685	-3.924736054170059, 122.1574681218231	13.9	13.95	0.36
8.	Jonatan	-3.9246404388439395, 122.1575056657642	-3.924736054170059, 122.1574681218231	11.42	11.35	0.62
9.	Eca Sukmawati	-3.9246552074939625, 122.15749759120156	-3.924736054170059, 122.1574681218231	9.57	9.41	1.70
10.	Muhamad Nabil	-3.9246337258211113, 122.1575056657642	-3.924736054170059, 122.1574681218231	12.12	12.17	0.41

Untuk mencari nilai *galat (%) error* berdasarkan pengujian pada tabel 5.2 digunakan persamaan rumus sebagai berikut :

$$galat (\%) = \frac{|\text{Jarak sesungguhnya} - \text{Jarak Metode}|}{\text{Jarak sesungguhnya}} * 100\%$$

$$galat (\%) = \frac{|10 - 10,26|}{10} * 100\% \\ = 2,60\%$$

Untuk mencari rata-rata *galat (%) error* menggunakan persamaan rumus sebagai berikut :

$$\text{Rata - rata galat (\%)} = \frac{\sum galat(\%)}{\text{Jumlah percobaan}} = \frac{10,15\%}{10} = 1,015\%$$

Dan untuk mencari nilai akurasinya menggunakan persamaan rumus sebagai berikut :

$$\text{Akurasi} = 100\% - galat(\%) = 100\% - 1,015\% = 98,98\%$$

Jadi nilai akurasi untuk pengujian pada tabel 5.2 adalah 98,98%.

## **BAB VI**

### **PENUTUP**

#### **6.1 Kesimpulan**

Berdasarkan hasil penelitian yang telah dilakukan mengenai Implementasi Teknologi *Location Based Service (Lbs)* Dan Metode *Haversine Formula* Pada Sistem Monitoring Kehadiran Siswa Secara *Real Time* , maka diperoleh kesimpulan yaitu:

1. Implementasi *haversine formula* sebagai penghitung jarak berhasil dilakukan serta otomatisasi dalam melakukan absensi berhasil didapatkan.
2. Hasil dari penggunaan rumus perhitungan jarak menggunakan *haversine formula* pada sistem maupun secara manual tidak jauh berbeda, presentasi menunjukkan 98,98% akurat.

#### **6.2 Saran**

Pada penelitian selanjutnya sistem ini dapat dibuat lebih kompleks lagi seperti penambahan status kehadiran serta adanya fitur upload surat izin/sakit untuk siswa.

## DAFTAR PUSTAKA

- C, A. (2020). *Pengertian HTML*. <https://www.hostinger.co.id/tutorial/apa-itu-html/>
- Devie, E., & Winarno, E. (2019). Aplikasi Location Based Service Untuk Informasi Kuliner Di Yogyakarta. *Dinamik*, 23(1), 15–21.  
<https://doi.org/10.35315/dinamik.v23i1.7174>
- Fathur Rizal, & Ahmad Khairi. (2020). Aplikasi Monitoring Kehadiran Menggunakan Global Positioning System Berbasis Android Untuk Peningkatan Kinerja Karyawan di Universitas Nurul Jadid. *Explore IT! : Jurnal Keilmuan Dan Aplikasi Teknik Informatika*, 12(2), 75–80.  
<https://doi.org/10.35891/explorit.v12i2.2282>
- Ferdianto, Y. (2013). *Pengertian PHP*. <https://www.carawebs.info/2013/04/apa-itu-php.html>
- Hartawan, G. P. (2017). Implementasi Rational Unified Process Dalam Sistem Informasi E-Sekolah(Studi Kasus SMA Negeri 1 Cibadak). *Jurnal SANTIKA : Jurnal Ilmiah Sains Dan Teknologi, Volume 7*  
N(<https://jurnal.ummi.ac.id/index.php/santika/issue/view/27>), 563–571.  
<https://jurnal.ummi.ac.id/index.php/santika/issue/view/27>
- Irmayana, A., Aryasa, K., & Herlinda, H. (2021). Sistem Absensi Dan Monitoring Kehadiran Siswa Menggunakan Metode Location Based Services (LBS). *SISITI: Seminar Ilmiah Sistem Informasi Dan Teknologi Informasi*, 10(2), 124–133.
- Miftahuddin, Y., Umaroh, S., & Karim, F. R. (2020). Perbandingan Metode Perhitungan Jarak Euclidean, Haversine, Dan Manhattan Dalam Penentuan Posisi Karyawan. *Jurnal Tekno Insentif*, 14(2), 69–77.  
<https://doi.org/10.36787/jti.v14i2.270>

- Pradnyana, G. A., & Brahma, A. A. G. R. W. (2020). Pengembangan Sistem Informasi Monitoring Kehadiran Dosen Secara Realtime Berbasis Web. *International Journal of Natural Science and Engineering*, 4(1), 11. <https://doi.org/10.23887/ijnse.v4i1.29036>
- Qois, N., & Jumaryadi, Y. (2021). Implementasi Location Based Service Pada Sistem Informasi Kehadiran Pegawai Berbasis Android. *Sistemasi*, 10(3), 550. <https://doi.org/10.32520/stmsi.v10i3.1369>
- Rahardjo, A. (2018). *Pengertian RUP (Rational Unified Process)*.
- Raharjo, B. (2019). *Pemrograman Android dengan Flutter (Pertama)*. Informatika Bandung.
- Sukindar. (2016). *Pengertian Dart*. <https://teknojurnal.com/kelebihan-bahasa-pemrograman-dart/>
- Susanty, W., Astari, I. N., & Thamrin, T. (2019). Aplikasi Gis Menggunakan Metode Location Based Service (Lbs) Berbasis Android. *Explore: Jurnal Sistem Informasi Dan Telematika*, 10(1). <https://doi.org/10.36448/jsit.v10i1.1218>
- Wahyono, Eko Budi; Suhattanto, M. A. (2019). Survey Satelit Pertanian. *Jurnal Sosial Politik*, 5(2). <https://doi.org/10.22219/sospol.v5i2.10695>
- Yuwamahendra, K. A., & Ratnasari, C. I. (2020). Penerapan Teknologi Location - Based Services dalam Mobile Application : Suatu Tinjauan Literatur. *Automata*.