

HASIL
IMPLEMENTASI *GLOBAL NAVIGATION SATELLITE SYSTEM* (GNSS)
PADA SISTEM PRESENSI BERBASIS *ANDROID* MENGGUNAKAN
METODE *SPATIAL MAP MATCHING*

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Sarjana Teknik



MUHAMAD DANIL

E1E117040

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HALU OLEO
KENDARI
2021

LEMBAR PENGESAHAN

Seminar Hasil

IMPLEMENTASI *GLOBAL NAVIGATION SATELLITE SYSTEM* (GNSS) PADA *SYSTEM* PRESENSI BERBASIS *ANDROID* MENGGUNAKAN METODE *SPATIAL MAP MATCHING* (Studi Kasus : Teknik Informatika)

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, Maret 2021

Muhamad Danil

NIM. E1E1 17 040

Pembimbing I

Pembimbing II

Dr. La Ode Golok Jaya, ST., M.T.

NIP. 19761020 200501 1 002

L.M. Tajidun, S.T., M.Eng

NIDN. 0030048107

Mengetahui,

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

Sutardi, S.Kom.,MT

NIP. 19760222 201012 1 001

INTISARI

Muhamad Danil, E1E117040

IMPLEMENTASI *GLOBAL NAVIGATION SATELLITE SYSTEM* (GNSS) PADA SISTEM PRESENSI BERBASIS *ANDROID* MENGGUNAKAN METODE *SPATIAL MAP MATCHING*

Skripsi, Fakultas Teknik, 2021

Kata Kunci: Presensi, *GNSS*, *Spatial Map Matching*, *Android*

Absensi adalah dokumen yang mencatat kehadiran mahasiswa pada setiap proses perkuliahan. Universitas Halu Oleo merupakan salah satu universitas yang masih menggunakan sistem manual dalam penanganan daftar hadir di masing-masing jurusan. Untuk setiap jurusan sekitar 300 mahasiswa aktif dengan lebih dari 20 mata kuliah disetiap semester, tentu menjadi tugas yang tidak mudah bagi pegawai staff di jurusan untuk mengontrol data tersebut. Oleh karena itu peneliti mendapat gagasan untuk memanfaatkan *Global Navigation Satellite System* (GNSS) dan *smartphone Android* untuk mengembangkan sistem daftar hadir. Dimana dalam sistem absensi ini nantinya akan menyimpan data setiap mahasiswa dan dapat dicetak. Peneliti menerapkan suatu algoritma yaitu Algoritma *Spatial Map Matching* yang fungsinya sebagai penentuan titik koordinat pada saat proses absensi berlangsung. Hasil yang didapatkan dari penelitian ini bahwa metode *Spatial map matching* berhasil diimplementasikan untuk mencocokkan lokasi mahasiswa dan lokasi kelas saat proses absensi berlangsung. Kecepatan absensi hanya berkisar 5 sampai 10 detik saja untuk setiap mahasiswa yang mengabsen.

ABSTRACT

Muhamad Danil, E1E117040

IMPLEMENTATION GLOBAL NAVIGATION SATELLITE SYSTEM (GNSS) OF SYSTEM PRESENCE BASED ON ANDROID USING THE METHOD SPATIAL MAP MATCHING

Thesis, Faculty Of Engineering, 2021

Keywords : Presence, *GNSS*, *Spatial Map Matching*, *Android*

Attendance is a document that records student attendance at each lecture process. Halu Oleo University is one of the universities that still uses a manual system in handling attendance lists in each department. For each department, there are about 300 active students with more than 20 courses in each semester. Of course, it is not an easy task for staff in the department to control the data. Therefore, researchers got the idea to take advantage of the Global Navigation Satellite System (GNSS) and Android smartphones in developing an attendance list system. Where in the Global Navigation Satellite System (GNSS), each student's data will be stored and can be printed. The researcher applied an algorithm, namely the Spatial Map Matching Algorithm, which functions as the determination of the coordinate points during the attendance process. The results obtained from this study show that the Spatial map matching method was successfully implemented to match the location of students and class locations during the attendance process. The attendance speed only ranges from 5 to 10 seconds for each student to attend.

DAFTAR ISI

HALAMAN JUDUL	ii
LEMBAR PENGESAHAN.....	ii
INTISARI.....	iii
ABSTRACT.....	ii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	vii
DAFTAR TABEL.....	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	4
1.7 Tinjauan Pustaka.....	5
BAB II LANDASAN TEORI.....	8
2.1 <i>Global Navigation Satellite System (GNSS)</i>	8
2.2 PHP.....	10
2.3 <i>Hypertext Markup Language (HTML)</i>	11
2.4 <i>Cascading Style Sheet (CSS)</i>	12
2.5 <i>Android</i>	13
2.6 <i>MySQL</i>	14
2.7 Flutter.....	16
2.8 Dart.....	16
2.9 <i>Spatial Matching (S-Matching)</i>	18
2.10 <i>Rational Unified Process (RUP)</i>	22
2.11 <i>Unified Modeling Language (UML)</i>	24
BAB III METODOLOGI PENELITIAN.....	29
3.1 Waktu dan Tempat Penelitian.....	29
3.1.1 Waktu.....	29
3.1.2 Tempat Penelitian.....	29

3.2	Metode Pengumpulan Data.....	29
3.3	Metode Pengembangan Sistem.....	30
3.3.1	Permulaan (<i>Inception</i>).....	30
3.3.2	Perluasan / Perencanaan (<i>Elaboration</i>).....	30
3.3.3	Konstruksi (<i>Construction</i>).....	30
3.3.4	Transisi (<i>Transition</i>).....	30
BAB IV ANALISIS DAN PERANCANGAN SISTEM.....		31
4.1	Analisis Kebutuhan sistem.....	31
4.1.1	Kebutuhan Fungsional.....	31
4.1.2	Kebutuhan Non fungsional.....	32
4.2	Analisis Perancangan Sistem.....	33
4.2.1	<i>Unified Modeling Language</i> (UML).....	33
4.4	Perancangan Antarmuka (<i>Interface</i>).....	49
4.4.1	<i>Form Login</i>	49
4.4.2	Menu <i>Dashboard</i>	49
4.4.3	Menu Absen.....	50
4.4.4	Menu Mahasiswa.....	51
4.4.5	Menu Dosen.....	51
4.4.6	Halaman <i>Login Mobile</i>	52
4.4.7	Halaman <i>Home Mobile</i>	52
4.4.8	Halaman Mata kuliah <i>Mobile</i>	53
4.4.9	Halaman Absensi.....	54
BAB V IMPLEMENTASI DAN PEMBAHASAN.....		55
5.1	Implementasi Sistem.....	55
5.2	Data.....	55
5.2.1	Implementasi <i>Interface</i>	55
5.3	Implementasi <i>Interface System</i> Berbasis Web.....	55
5.3.1	Halaman <i>Login</i>	55
5.4	Implementasi <i>Interface System</i> Berbasis <i>Mobile</i>	62
5.4.1	Halaman <i>Login</i>	62
5.4.2	Halaman <i>Dashboard</i>	63
DAFTAR PUSTAKA.....		74

DAFTAR GAMBAR

Gambar 2.1 <i>Source Code Variable</i> dan Tipe.....	17
Gambar 2.2 <i>Source Code Input Output</i>	17
Gambar 2.3 <i>Source Code Conditional</i>	18
Gambar 2.4 <i>Source Code Loop</i>	18
Gambar 2.5 Area Akurasi GPS.....	19
Gambar 2.6 Probabilitas Observasi.....	20
Gambar 2.7 Probabilitas Transmisi.....	20
Gambar 2.8 <i>Arsitektur Rational Unified Process</i>	23
Gambar 4.1 <i>Use Case Diagram</i>	35
Gambar 4.2 <i>Activity Diagram Login (Admin dan User)</i>	40
Gambar 4.3 <i>Activity Diagram Beranda/Dashboard</i>	41
Gambar 4.4 <i>Activity Diagram Tampil Data Absensi</i>	41
Gambar 4.5 <i>Activity Diagram Cetak Absen</i>	42
Gambar 4.6 <i>Activity Diagram Tambah Data Mahasiswa</i>	42
Gambar 4.7 <i>Activity Diagram Edit Data Mahasiswa</i>	43
Gambar 4.8 <i>Activity Diagram Hapus Data mahasiswa</i>	44
Gambar 4.9 <i>Activity Diagram Mata Kuliah</i>	44
Gambar 4.10 <i>Activity Diagram Absensi Kode</i>	45
Gambar 4.11 <i>Activity Diagram Absensi Lokasi</i>	46
Gambar 4.12 <i>Class Diagram</i>	47
Gambar 4. 13 <i>Sequence Diagram Login</i>	48
Gambar 4. 14 <i>Sequence Diagram Absensi</i>	48

Gambar 4. 15 <i>Sequence Diagram</i> Tambah Data.....	49
Gambar 4. 16 <i>Sequence Diagram</i> Tambah Data.....	49
Gambar 4. 17 Menu <i>Dashboard</i>	50
Gambar 4.18 Menu <i>Dashboard</i>	51
Gambar 4. 19 Menu Absen.....	51
Gambar 4.20 Menu Mahasiswa.....	52
Gambar 4.21 Menu Dosen.....	52
Gambar 4.22 Halaman <i>Login Mobile</i>	53
Gambar 4.23 Halaman <i>Home Mobile</i>	54
Gambar 4.24 Halaman Mata Kuliah <i>Mobile</i>	54
Gambar 4. 25 Halaman Absensi	55
Gambar 5.1 Tampilan <i>Login (Admin dan Dosen)</i>	57
Gambar 5.2 Tampilan <i>Dashboard (Admin)</i>	57
Gambar 5.3 Tampilan Mahasiswa (<i>Admin</i>)	58
Gambar 5.4 Tampilan Dosen (<i>Admin</i>)	58
Gambar 5.5 Tampilan Mata kuliah (<i>Admin</i>)	59
Gambar 5.6 Tampilan Absensi (<i>Admin</i>)	60
Gambar 5.7 Tampilan Absensi (<i>Admin</i>)	60
Gambar 5.8 Tampilan <i>User (Admin)</i>	61
Gambar 5.9 Tampilan Ruangan (<i>Admin</i>)	61
Gambar 5.10 Tampilan <i>Dashboard (Dosen)</i>	62
Gambar 5.11 Tampilan Jadwal (<i>Dosen</i>)	62
Gambar 5.12 Tampilan Absensi (<i>Dosen</i>)	63
Gambar 5.13 Tampilan Login (<i>Mahasiswa</i>)	64
Gambar 5.14 Tampilan Halaman Presensi (<i>Mahasiswa</i>)	65

Gambar 5.15 Tampilan Fitur Kalender (Mahasiswa)	66
Gambar 5.16 Tampilan Halaman Semua Jadwal (Mahasiswa)	67
Gambar 5.17 Tampilan Halaman Absensi (Mahasiswa)	68
Gambar 5.18 Tampilan Halaman Absensi <i>Barcode</i> (Mahasiswa)	69
Gambar 5.19 Tampilan Halaman Riwayat Presensi (Mahasiswa)	70
Gambar 5.20 Tampilan Halaman Lihat Lebih (Mahasiswa)	71

DAFTAR TABEL

Tabel 2.1 Tipe Data <i>Database MySQL</i>	15
Tabel 2.2 Simbol <i>Use Case Diagram</i>	25
Tabel 2.3 Simbol <i>Class Diagram</i>	26
Tabel 2.4 Simbol <i>Sequence Diagram</i>	27
Tabel 2.5 Simbol <i>Activity Diagram</i>	28
Tabel 3.1 <i>Gannt Chart</i> Waktu Penelitian.....	29
Tabel 4.1 Spesifikasi Perangkat Keras.....	33
Tabel 4.2 Spesifikasi Perangkat Lunak.....	34
Tabel 4.3 Deskripsi <i>Case Login (Admin, Mahasiswa dan Dosen)</i>	36
Tabel 4.4 Deskripsi <i>Case Mengolah Mata Kuliah</i>	36
Tabel 4.5 Deskripsi <i>Case Mengolah Data Dosen</i>	37
Tabel 4.6 Deskripsi <i>Case Mengolah Data Mahasiswa</i>	37
Tabel 4.7 Deskripsi <i>Case Jadwal Mata Kuliah</i>	38
Tabel 4.8 Deskripsi <i>Case Home</i>	38
Tabel 4.9 Deskripsi <i>Case Absensi Mahasiswa</i>	39
Tabel 4.10 Deskripsi <i>Case Logout</i>	39
Tabel 5.1 Pengujian <i>Black Box</i>	72
Tabel 5.2 Kordinat awal mahasiswa mengabsen	73
Tabel 5.3 Kordinat yang telah diperbaiki.....	73

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era modern seperti sekarang ini, kemajuan teknologi berkembang semakin pesat. Berbagai sistem mulai diciptakan menggunakan teknologi dengan tujuan untuk mempermudah pekerjaan manusia. Mulai dari bidang pemerintahan, perusahaan, politik dan pendidikan seperti sekolah dan universitas. Adapun di universitas sendiri ada salah satu sistem pengelolaan yang masih di kerjakan secara manual yaitu daftar hadir. Sistem daftar hadir merupakan salah satu faktor yang sangat penting dalam dunia universitas. Selain digunakan sebagai alat kontrol mahasiswa, daftar hadir juga digunakan sebagai salah satu penunjang penilaian dosen dan syarat dokumentasi barang setiap 5 tahun sekali.

Universitas Halu Oleo merupakan salah satu universitas yang masih menggunakan sistem manual dalam penanganan daftar hadir di masing-masing jurusan. Tanpa terkecuali jurusan Teknik Informatika yang setiap semester harus mencetak lembaran daftar hadir dan merekap data tersebut di akhir semester. Dari 300 mahasiswa aktif dengan lebih dari 20 mata kuliah di setiap semester, tentu menjadi tugas yang tidak mudah bagi pegawai staff di jurusan untuk mengontrol data tersebut. Maka dengan kemajuan teknologi, tidak menutup kemungkinan untuk mengembangkan sebuah aplikasi agar dapat membantu dan memudahkan sistem daftar hadir tersebut.

Telpon seluler (*Smartphone*) merupakan tools yang tepat dalam membangun sistem tersebut. Dengan mendeteksi lokasi menggunakan salah satu fungsi dari *Smartphone* yaitu GNSS. Memunculkan ide untuk memanfaatkan *Global Navigation Satellite System* (GNSS) dan *smartphone Android* dalam mengembangkan sistem daftar hadir tersebut. Dimana dalam *Global Navigation Satellite System* (GNSS) ini nantinya akan menyimpan data setiap mahasiswa dan dapat dicetak.

Di masa pandemik seperti sekarang ini, pemerintah menerapkan berbagai kebijakan untuk memutus mata rantai penyebaran virus covid-19 di Indonesia. Upaya yang dilakukan pemerintah salah satunya dengan menerapkan himbauan kepada masyarakat untuk melakukan *physical distancing* yaitu himbauan untuk menjaga jarak diantara masyarakat, menjauhi aktivitas dalam segala bentuk kerumunan, perkumpulan, dan menghindari adanya pertemuan yang melibatkan banyak orang.

Pemerintah juga menerapkan kebijakan *Work From Home* (WFH), kebijakan ini merupakan upaya yang diterapkan kepada masyarakat agar dapat menyelesaikan segala pekerjaan di rumah. Pendidikan di Indonesia pun menjadi salah satu bidang yang terdampak akibat covid-19. Dengan adanya pembatasan interaksi, Kementrian Pendidikan di Indonesia juga mengeluarkan kebijakan yaitu dengan meliburkan universitas dan mengganti proses Kegiatan Belajar Mengajar (KBM) dengan menggunakan sistem dalam jaringan (daring). Proses pembelajaran menggunakan sistem jaringan bukan tatap muka lagi, sehingga absennya pun sudah menggunakan sistem *online*. Oleh karena itu aplikasi absen *online* ini penerapan-nya sangat dibutuhkan.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan algoritma *Spatial Map Matching* sebagai metode peningkatan akurasi lokasi *Global Navigation Satellite System* (GNSS) pada aplikasi daftar hadir Universitas Halu Oleo?
2. Bagaimana mengembangkan sistem daftar hadir agar dapat terdigitalisasi dan dapat memudahkan staff Universitas Halu Oleo?
3. Bagaimana mengembangkan aplikasi daftar hadir yang praktis, efektif dan efisien?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi ini hanya menerapkan algoritma *Spatial Map Matching* pada data *Global Navigation Satellite System* (GNSS).
2. Tidak ada *system* dalam aplikasi ini yang dibuat untuk mahasiswa.
3. Ponsel *Android* pengguna aplikasi harus terkoneksi jaringan internet dengan baik.
4. Aplikasi ini hanya digunakan dalam lingkup universitas Universitas Halu Oleo.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan algoritma *Spatial Map Matching* untuk meningkatkan akurasi lokasi *Global Navigation Satellite System* (GNSS) pada aplikasi daftar hadir Universitas Halu Oleo.
2. Mengembangkan sistem daftar hadir agar dapat terdigitalisasi dan dapat memudahkan staff Universitas Halu Oleo.
3. Membangun aplikasi daftar hadir yang praktis dan cepat serta dokumentasi yang mudah di Universitas Halu Oleo.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan algoritma *Spatial Map Matching* sebagai metode peningkatan akurasi lokasi *Global Navigation Satellite System* (GNSS) aplikasi daftar mahasiswa Universitas Halu Oleo.
2. Mengubah sistem daftar hadir mahasiswa yang semula manual menjadi sistem yang terdigitalisasi dan memberikan kemudahan kepada staf Universitas Halu Oleo dalam dokumentasi daftar hadir.
3. Mengembangkan aplikasi daftar hadir yang praktis dan cepat.

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 GNSS, *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 *SYSTEM*

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

BAB V IMPLEMENTASI DAN PENGUJIAN *SYSTEM*

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 2020, Farhan Zahid dan Egi Ferdiana melakukan penelitian dengan judul Penerapan Algoritma *Spatial Map Matching* Dengan *API* Menggunakan *GPS* Untuk Posisi Tumpangan Kendaraan. Berdasarkan pengalaman penulis menemukan masalah kemacetan lalu lintas Kota Bogor. Sehingga Diperlukan solusi untuk mengatasi permasalahan kemacetan lalu lintas yang disebabkan oleh jumlah kendaraan yang melebihi kapasitas jalan. Hal tersebut yang membuat peneliti mulai mengimplementasikan *Spatial Map Matching* untuk ditanamkan kedalam *GPS* sehingga tumpangan kendaraan berjalan sesuai dengan yang diharapkan (Zayid & Ferdiana, 2020).

Pada tahun 2018, Mochamad Rizal Maftukhin melakukan penelitian dengan judul Implementasi Algoritma *Spatial Map Matching* Untuk Mengetahui Lokasi Kendaraan Melalui Aplikasi *GPS Tracker*. Dalam penelitian ini, *GPS* dimanfaatkan untuk melacak suatu kendaraan. *GPS Tracker* mengirimkan titik koordinat dari kendaraan kepada aplikasi pelacak untuk memantau pergerakan dari kendaraan. Selain itu juga, peneliti menggunakan *Spatial Map Matching* untuk meminimalisir kesalahan dalam menampilkan posisi kendaraan (Maftukhin, 2018).

Pada tahun 2020, Syaiful Amrial Khoir, Anton Yudhana, dan Sunardi, melakukan penelitian dengan judul Implementasi *GPS (Global Positioning System)* Pada Presensi Berbasis *Android* Di BMT Insan Mandiri. Dalam penelitian ini, *GPS* dimanfaatkan sebagai aplikasi absensi online berbasis mobile *Android* dengan security pengenalan wajah. Selain itu juga, peneliti menggunakan *Spatial Map Matching* untuk meminimalisir kesalahan dalam menampilkan posisi jarak (Khoir et al., 2020).

Pada tahun 2016, Adriana Fanggidae, Yulianto Triwahyuadi Polly, melakukan penelitian dengan judul *System* Presensi Dosen Menggunakan *IMEI* dan *GPS Smartphone* dengan Data Terenkripsi. Dalam penelitian ini, *IMEI (International Mobile Equipment Identity)*, dan *GPS (Global Positioning System)* dimanfaatkan sebagai aplikasi presensi *online* berbasis *mobile Android* untuk dosen. Selain itu juga, peneliti menggunakan *algoritme cipher aliran* dengan

metode pengacakan kunci sebagai keamanan aliran data dalam sistem (Fanggidae & Polly, 2016).

Khoirul Muhtajib, Mochamad Sidqon, S.Si.,M.Si, melakukan penelitian dengan judul Aplikasi Presensi Online Menggunakan IMEI Dan GPS Pada Smartphone *Android*. Dalam penelitian ini, *IMEI (International Mobile Equipment Identity)*, dan *GPS (Global Positioning System)* dimanfaatkan sebagai Aplikasi Presensi dengan menggunakan *smartphone Android* yang nantinya terintegrasi dengan *imei smartphone* masing-masing karyawan dan juga dengan menggunakan *GPS* agar lokasi presensi karyawan lebih akurat sesuai dengan lokasi dengan kantor (Muhtajib et al., 2017).

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 *system 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 (*GLObal'naya NAVigatsionnaya 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 Satelite 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).

GLONAS 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 COMPAS 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 Pasific, Tahap II pada tahun 2020 sudah bisa menjangkau seluruh dunia. Segmen angkasa *system* 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 45° 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.
- 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 PHP

PHP Kepanjangan dari PHP adalah "*Hypertext Preprocessor*" (ini merupakan singkatan rekursif). PHP adalah bahasa scripting web *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 *system*. (Ferdianto, 2013).

2.3 ***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 dievakuasi ketat dan disetujui oleh W3C. Contoh tag HTML antara lain: <head>, <body> dan <table> (C, 2020).

2.4 *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 *control* 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. Cara kerja CSS dengan menggunakan dua buah elemen penting untuk pemformatan tampilannya, diantaranya selektor dan deklarator. Dua buah elemen ini berfungsi sebagai penentu format tampilan dan lainnya menempatkan format tampilan tersebut. Deklarator berisi beberapa perintah-perintah CSS untuk menentukan format dari sebuah elemen pada halaman *web*. Sedangkan selektor adalah sebuah perintah lanjut dari deklarator dan berfungsi menempatkan format tampilan dari deklarator.

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*

7. *Selector language*
8. *Selector child*
9. *Selector first-child*
10. *Selector adjacent* (berdekatan).

2.5 *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)

2.6 *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 (column, 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_nam 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; s

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 rentang-nya -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.
5.	<i>Smallint</i>	Adalah tipe data bilangan bulat yang rentang-nya – 32768 sampai 32767. The unsigned range is 0 to 65535
6.	<i>Mediumint</i>	Adalah tipe data bilangan bulat yang rentang-nya – 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

2.7 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).

2.8 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() {
  stdout.write("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

2.9 *Spatial Matching (S-Matching)*

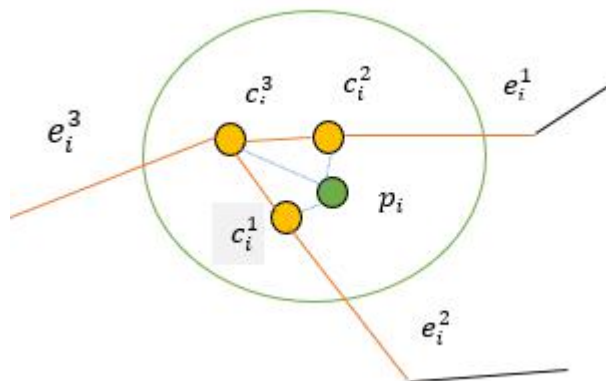
Spatial matching merupakan salah satu dari beberapa algoritma map *matching* yang menggunakan data GPS pada peta digital dan menggunakan analisa spasial geografis untuk membuat kandidat grafik berdasar data GPS yang ada, lalu setelahnya akan dilakukan perhitungan jarak terdekat untuk menentukan atau mencocokkan-nya grafik tersebut ke dalam peta yang asli. Sebelum membahas *spatial matching* lebih jauh, lebih baiknya membahas terlebih dahulu konsep analisa *spasial*.

Analisa *spasial* merupakan sekumpulan metoda untuk menemukan dan menggambarkan tingkatan/pola dari sebuah fenomena spasial, sehingga dapat dimengerti dengan lebih baik. Dengan melakukan analisis spasial, diharapkan muncul informasi baru yang dapat digunakan sebagai dasar pengambilan

keputusan di bidang yang dikaji. Metode yang digunakan sangat bervariasi, mulai observasi visual sampai ke pemanfaatan matematika/statistik terapan (Sadahiro, 2006).

Spatial matching termasuk algoritma yang membutuhkan sampel data yang sedikit atau disebut *low sampling rate*. Dalam analisis *spatial* ini menggunakan informasi geometrik dan topologi jaringan jalan untuk mengevaluasi kandidat poin. Informasi geometrik digabungkan dengan menggunakan probabilitas observasi, dan informasi topologi diekspresikan dengan menggunakan probabilitas transmisi

Seperti yang ditampilkan pada gambar 2.5 disitu digambarkan bahwa garis berwarna merah mewakili gambar jalan pada peta sedangkan lingkaran biru menggambarkan area akurasi dari GPS. Disini C merupakan titik kandidat yang telah disiapkan sebelumnya.

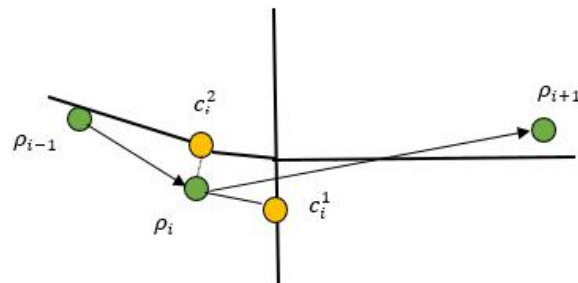


Gambar 2.5 Area Akurasi GPS

Kemudian P merupakan titik koordinat yang didapatkan dari GPS. Dari gambar tersebut diketahui juga bahwa tidak selamanya atau terkadang titik GPS bisa tidak akurat apabila ditampilkan pada peta dari aplikasi GPS tracker. Seharusnya titik GPS P berada pada salah satu titik kandidat C yang menunjukkan jalan sebenarnya, sebab titik P dalam gambar menunjukan posisi yang melenceng dari posisi seharusnya ia berada. Sehingga perlu dilakukan langkah-langkah agar titik tersebut menunjukkan posisi yang tepat pada peta GPS tracker.

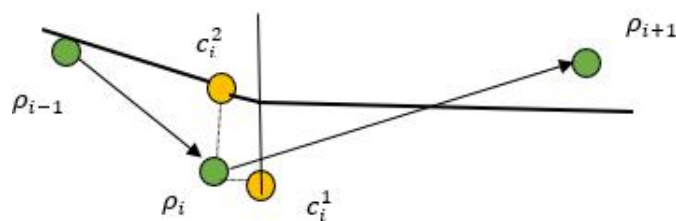
Pada *spatial map matching* menggunakan baik segi geografis dan topologi dari suatu wilayah. Untuk mendapatkan informasi geografis dilakukan dengan menggunakan probabilitas observasi dan informasi topologi-nya dilakukan dengan probabilitas transmisi.

Untuk menjelaskan probabilitas observasi dapat dilihat pada Gambar 2.6. P menunjukkan jalur yang dilalui. Di gambar tersebut titik P_i merupakan titik yang didapat dari GPS, titik c_2^i dan c_1^i merupakan titik kandidat yang tepat seharusnya titik P_i berada.



Gambar 2.6 Probabilitas Observasi

Dengan menggunakan probabilitas observasi dicari titik kandidat yang paling dekat dan berada pada jalur yang dilalui, dalam hal ini titik yang paling dekat adalah titik c_2^i , maka titik yang tepat dimana titik P_i seharusnya berada adalah pada titik c_2^i .



Gambar 2.7 Probabilitas Transmisi

Hal tersebut akan berbeda lagi bila menemui masalah seperti pada Gambar 2.7. Pada gambar ditunjukkan bahwa titik P_i lebih dekat dengan titik kandidat c_1^i

i padahal jalur yang dilalui berada pada titik kandidat c_{i-1} . Apabila menemui hal yang seperti ini maka dilakukan cara probabilitas transmisi agar titik P_i tetap berada pada posisi sebagaimana dalam jalur yang dilalui. Probabilitas transmisi dilakukan dengan menghitung jarak terpendek dari posisi sebelumnya ke titik kandidat yaitu titik p_{i-1} dan c_{i-1} . Dengan begitu posisi sebenarnya dari titik P_i akan tetap berada pada jalur yang telah dilalui. Probabilitas observasi dapat dinyatakan sebagai berikut:

$$N(c_i^j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i^j - \mu)^2}{2\sigma^2}} \quad (2.1)$$

Dimana $x_i^j = \text{dist}(c_{ij}, p_i)$ adalah jarak antara p_i dan c_{ij} . Namun, probabilitas observasi tidak memperhitungkan konteks posisi dari titik GPS, sehingga terkadang menyebabkan hasil pencocokan titik yang salah. Maka untuk menghindari hal tersebut digunakan perhitungan probabilitas transmisi agar titik GPS yang melenceng dapat diminimalisir dan dapat dicocokkan ke titik yang seharusnya berada. Probabilitas transmisi dinyatakan sebagai berikut:

$$V(c_{i-1}^t \rightarrow c_i^s) = \frac{d_{i-1 \rightarrow i}}{w_{(i-1,t) \rightarrow (i,s)}} \quad (2.2)$$

Dimana $d_{i-1 \rightarrow i} = \text{dist}(p_i, p_{i-1})$ adalah jarak euclidean antara p_i dan p_{i-1} , dan $w_{(i-1,t) \rightarrow (i,s)}$ adalah panjang dari jalur terpendek dari c_{i-1}^t ke c_i^s .

Dari rumus-rumus yang telah didapat diatas, maka persamaan dari spatial analysis dapat dinyatakan seperti berikut ini sebagai gabungan dari probabilitas observasi dan probabilitas transmisi:

$$F_s(c_{i-1}^t \rightarrow c_i^s) = N(c_i^j) \times V(c_{i-1}^t \rightarrow c_i^s), 2 \leq i \leq n \quad (2.3)$$

Dimana c_{i-1}^t dan c_i^s adalah dua titik kandidat untuk dua titik GPS bersebelahan p_{i-1} dan p_i .

Persamaan di atas menghitung kemungkinan bahwa suatu benda bergerak dari $c_{i-1} t$ ke c_{is} menggunakan produk dari dua fungsi probabilitas, sehingga informasi geometrik dan topologi dipertimbangkan. Perhatikan bahwa dalam praktiknya, tidak mungkin benda bergerak untuk selalu mengikuti jalur terpendek. Oleh karena itu probabilitas observasi $N(c_{ij})$ tidak dapat dihilangkan dari persamaan. Dengan analisa spasial untuk dua titik GPS bersebelahan p_{i-1} dan p_i , satu set jalur kandidat $c_{i-1} t \rightarrow c_{is}$ akan dihasilkan. Setiap jalur diberi nilai pengukuran spasial yang dihitung dari persamaan di atas.

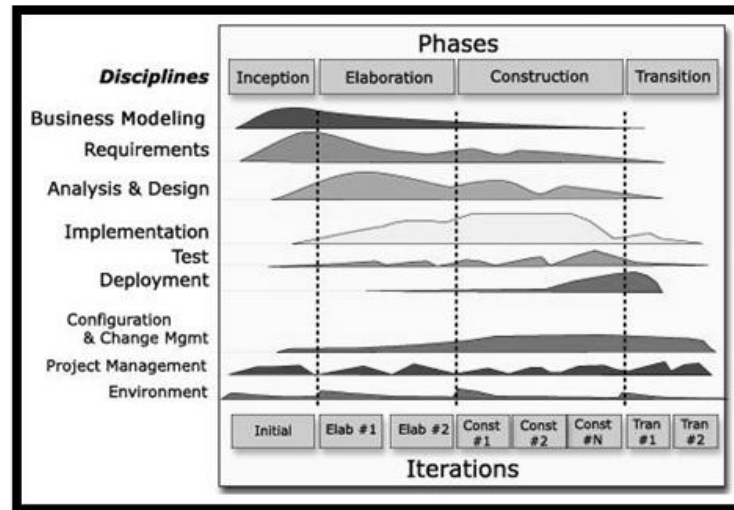
2.10 Rational Unified Process (RUP)

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 2.8 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 Manegement, Project Management, Environtment.*



Gambar 2.8 Arsitektur Rational Unified Process

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 *system* yang telah direncanakan, dianalisa dan dibangun. Dari tahap ini bisa dilihat kekurangan dari *system* yang telah dibangun.

2.11 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan pengganti dari metode analisis berorientasi objek dan desain berorientasi objek (OOA&D) yang dimunculkan 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 *system* perangkat lunak (Shalahudin, 2011)

2.12 *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* simbol 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 diatasnya 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.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
8.		<i>Use case</i>	Deskripsi dari uraian aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor


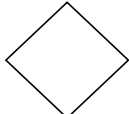

Tabel Lanjutan 2.2 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
9.		<i>Use case</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

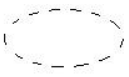
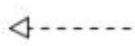

2. *Class Diagram*

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
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

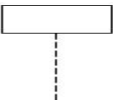


Tabel Lanjutan 2.3 Simbol Class Diagram

No	Simbol	Nama	Keterangan
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem 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

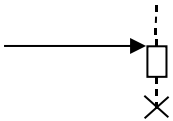
3. Sequence Diagram

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksiantara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi *system* (Shalahudin, 2011).

Tabel 2.4 Simbol Sequence Diagram

No	Simbol	Nama	Keterangan
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi






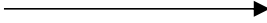

Lanjutan Tabel 2.4 Simbol Sequence Diagram

No	Simbol	Nama	Keterangan
4		<i>Message</i>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

4. 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 2.5 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 <i>system</i> yang mencerminkan eksekusi dari suatu aksi
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 Conector</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 Oktober 2020 sampai dengan Februari 2021. Rincian kegiatan dapat dilihat pada Tabel 3.1 berikut:

Tabel 3.1 Gantt Chart Waktu Penelitian

No	Uraian	Waktu (2020)																			
		November				Desember				Januari				Februari				Maret			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Inception																				
2	Elaboration																				
3	Constructio n																				
4	Transition																				

3.1.2 Tempat Penelitian

Adapun tempat penelitian tugas akhir yang akan dilakukan di Jurusan Teknik Informatika Fakultas Teknik Universitas Halu Oleo.

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 dari Universitas Halu Oleo 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 akan dibangun, yaitu dengan mencari referensi yang membahas tentang penggunaan metode *Spatial Map Matching*, sehingga metode *Spatial Map Matching* dapat diimplementasikan dalam aplikasi Absensi mahasiswa di Universitas Halu Oleo.

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 *Spatial Map Matching*.

3.3.2 Perluasan / Perencanaan (*Elaboration*)

Setelah menentukan ruang lingkup penelitian, tahap ini akan dilakukan perancangan dan analisis sistem menggunakan *flowchart* meliputi *flowchart* Aplikasi Absensi, dan *flowchart* metode *Spatial Map Matching*. 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 *Spatial Map Matching* pada *system*.

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 mahasiswa yang diambil dari setiap *handphone* mahasiswa 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*.

a. Analisis Kebutuhan *Input*

Input dari aplikasi yang dibangun terdiri dari:

1. Informasi data mahasiswa Universitas Halu Oleo.
2. Informasi data dosen Universitas Halu Oleo.
3. Informasi data mata kuliah Universitas Halu Oleo.

b. Analisis Kebutuhan Proses

Kebutuhan proses dalam aplikasi ini antara lain:

1. Proses pembuatan aplikasi menggunakan Bahasa pemrograman *PHP* di sisi *webservice* dan menggunakan Bahasa pemrograman *Dart* di sisi aplikasi *user*.

c. Analisis Kebutuhan *Output*

Output yang diperoleh dalam sistem ini adalah dosen dapat mengetahui lokasi mahasiswa saat sedang mengikuti proses pembelajaran dan aplikasi berbasis *Android* untuk mengirimkan lokasi saat sedang mengikuti pembelajaran hingga akhir pembelajaran selesai dan mendata absensi mahasiswa Universitas Halu Oleo.

4.1.2 Kebutuhan Non fungsional

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

a. Kebutuhan Perangkat Keras

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

Tabel 4.1 Spesifikasi Perangkat Keras

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>

b. 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>Windows</i>	<i>Operating System</i>	<i>Windows 10 Home</i>
2.	<i>Xampp</i>	<i>Universal development environment</i>	<i>Xampp v3.2.2</i>
3.	<i>Apache</i>	<i>Web Server</i>	<i>Apache 2.4 win64-VC14</i>
4.	<i>Mysql</i>	<i>Database Management</i>	<i>Mysql 5.7</i>
5.	<i>PHP</i>	<i>Web development</i>	<i>PHP 7.3</i>
6.	<i>Dart</i>	<i>Android Development</i>	<i>Dart 3.4.1</i>
7.	<i>Chrome</i>	<i>Web browser</i>	<i>Chrome version 76.0</i>
8.	<i>Android Studio</i>	<i>SDK Development Tool</i>	<i>Android Studio 3.3</i>
9.	<i>Visual Studio Code</i>	<i>Code Editor</i>	<i>VS Code v1.38</i>
10.	<i>Flutter</i>	<i>Framework Android</i>	<i>Flutter 3.4.1</i>

4.2 Analisis Perancangan Sistem

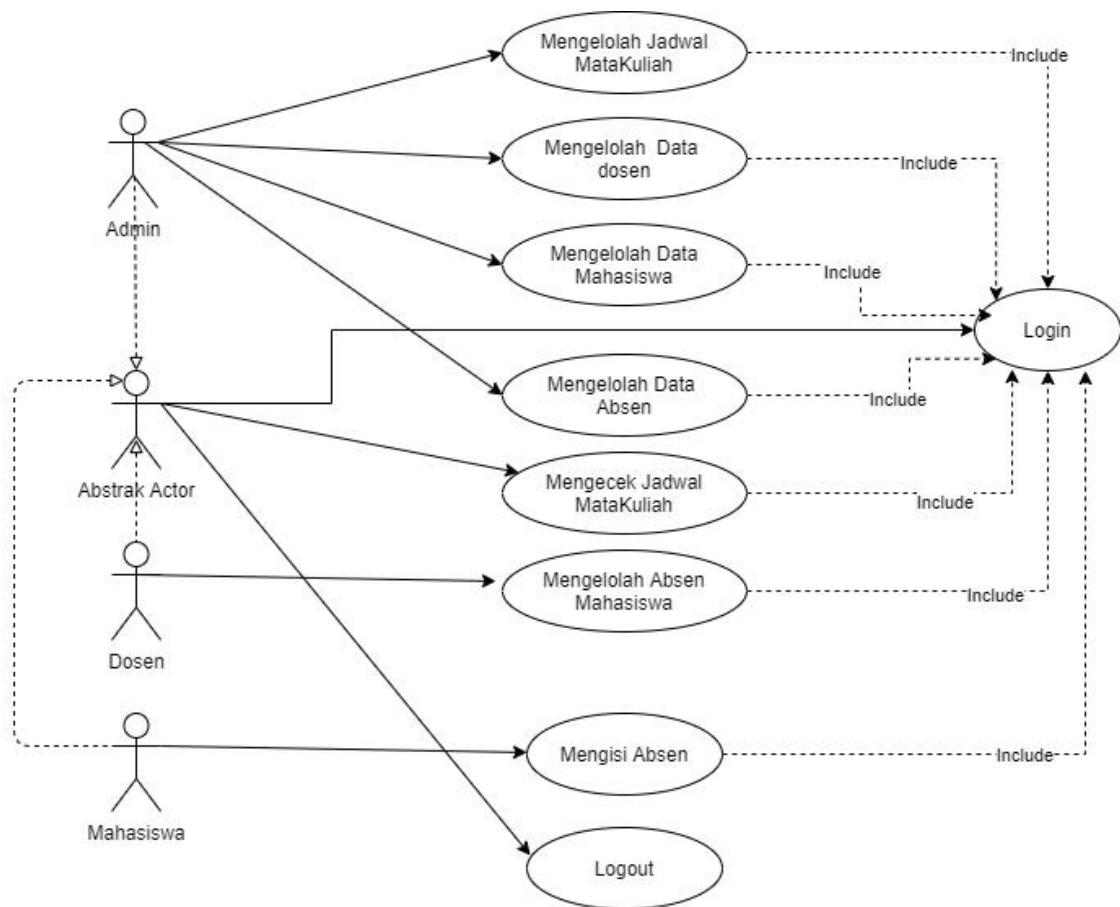
Perancangan sistem yang akan dibangun yaitu perancangan UML dan perancangan *user interface* sistem.

4.2.1 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 *system* dan bagaimana cara mereka berinteraksi dengan sistem. *Use Case Diagram* dari sistem yang akan dibangun adalah sebagai berikut.



Gambar 4.1 Use Case Diagram

Tabel 4.3 Deskripsi Case Login (Admin, Mahasiswa dan Dosen)

Nama	Login
Deskripsi Singkat:	Proses Keamanan login untuk <i>Admin</i> , Dosen dan Mahasiswa
Persyaratan	<i>Admin</i> , Dosen dan Mahasiswa harus memasukkan <i>username</i> dan <i>password</i>
Kondisi Akhir	Masuk untuk manage sistem
Situasi Kesalahan	i. Yang masuk bukan <i>Admin</i> , Dosen, Mahasiswa ii. Salah memasukkan <i>username</i> dan <i>password</i>
Status sistem dalam saat terjadi kesalahan	1. Menampilkan pesan <i>error</i> 2. Tidak dapat masuk kedalam sistem
Aktor	<i>Admin</i> , Dosen dan Mahasiswa
Trigger	Keamanan <i>login</i> untuk <i>Admin</i> , Mahasiswa dan Dosen
Proses Standar	1. <i>Admin</i> , Dosen dan Mahasiswa masuk ke sistem 2. <i>Admin</i> , Dosen dan Mahasiswa memasukkan <i>username</i> 3. <i>Admin</i> , Dosen dan Mahasiswa memasukkan <i>password</i>
Proses Alternatif	Mengecek <i>username</i> dan <i>password</i>

Tabel 4.4 Deskripsi Case Mengolah Mata Kuliah

Nama	Mengelola Mata Kuliah
Deskripsi Singkat:	<i>Admin</i> ingin melihat atau mengelola mata kuliah
Persyaratan	<i>Admin</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Data mata kuliah di tampilkan
Situasi Kesalahan	-
Status sistem dalam saat terjadi kesalahan	-
Aktor	<i>Admin</i>
Trigger	<i>Admin</i> ingin menambahkan, menghapus atau mempernaruhi mata kuliah
Proses Standar	1. <i>Admin</i> masuk ke sistem dan telah melakukan proses <i>login</i> 2. <i>Admin</i> melihat data mata kuliah 3. <i>Admin</i> menambahkan mata kuliah 4. <i>Admin</i> menghapus mata kuliah 5. <i>Admin</i> mengedit mata kuliah
Proses Alternatif	

Tabel 4.5 Deskripsi Case Mengolah Data Dosen

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

Tabel 4.6 Deskripsi Case Mengolah Data Mahasiswa

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

Tabel 4.7 Deskripsi Case Jadwal Mata Kuliah

Nama	Data Absen
Deskripsi Singkat:	<ol style="list-style-type: none"> 1. Dosen ingin merekapitulasi melihat, mengelolah atau memperbaharui data absen. 2. <i>Admin</i> ingin melihat, merekapitulasi data absen mahasiswa.
Persyaratan	<i>Admin</i> dan Dosen melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Data absen di tampilkan
Situasi Kesalahan	-
Status sistem dalam saat terjadi kesalahan	-
Aktor	<i>Admin</i> dan Dosen
Trigger	<ol style="list-style-type: none"> 1. Dosen ingin menambahkan, menghapus atau memperbaharui data absen 2. <i>Admin</i> ingin merekapitulasi data absen
Proses Standar	<ol style="list-style-type: none"> 1. <i>Admin</i> atau Dosen masuk ke sistem dan melakukan <i>login</i> 2. Dosen melihat data absen 3. Dosen menambahkan data absen 4. Dosen menghapus data absen 5. Dosen mengedit data absen 6. Admin merekapitulasi data absen
Proses Alternatif	-

Tabel 4.8 Deskripsi Case Home

Nama	Home
Deskripsi Singkat:	Berisi data pribadi setiap Mahasiswa
Persyaratan	Mahasiswa masuk ke sistem
Kondisi Akhir	Menampilkan <i>profil</i> , Jadwal hari ini, dan Kalender
Situasi Kesalahan	-
Status sistem dalam saat terjadi kesalahan	-
Aktor	Mahasiswa
Trigger	-
Proses Standar	<ol style="list-style-type: none"> 1. Mahasiswa masuk kedalam sistem 2. Mahasiswa dapat melihat <i>profil</i> 3. Mahasiswa dapat melihat jadwal hari ini 4. Mahasiswa dapat melakukan absensi 5. Mahasiswa dapat melihat kalender
Proses Alternatif	

Tabel 4.9 Deskripsi Case Absensi Mahasiswa

Nama	Absensi
Deskripsi Singkat:	Mahasiswa melakukan absensi ketika perkuliahan dimulai
Persyaratan	Mahasiswa masuk ke sistem absensi
Kondisi Akhir	Menampilkan halaman absensi data kehadiran mahasiswa
Situasi Kesalahan	Lokasi mahasiswa di luar jangkauan lokasi ruangan kelas
Status sistem dalam saat terjadi kesalahan	1. Menampilkan pesan <i>error</i> 2. Tidak dapat melakukan absensi
Aktor	Mahasiswa
Trigger	Mahasiswa mengisi data absensi sesuai lokasi agar dianggap hadir dalam matakuliah.
Proses Standar	1. Mahasiswa memilih menu absen 2. Mahasiswa mengisi kode absensi 3. Mahasiswa mengirim lokasi saat ini
Proses Alternatif	-

Tabel 4.10 Deskripsi Case Logout

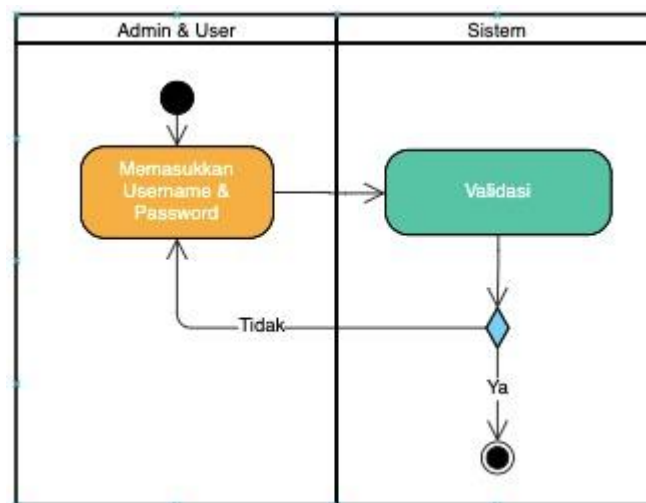
Nama	Absensi
Deskripsi Singkat:	Mahasiswa melakukan absensi ketika perkuliahan dimulai
Persyaratan	Mahasiswa masuk ke sistem absensi
Kondisi Akhir	Menampilkan halaman absensi data kehadiran mahasiswa
Situasi Kesalahan	Lokasi mahasiswa di luar jangkauan lokasi ruangan kelas
Status sistem dalam saat terjadi kesalahan	1. Menampilkan pesan <i>error</i> 2. Tidak dapat melakukan absensi
Aktor	Mahasiswa
Trigger	Mahasiswa mengisi data absensi sesuai lokasi agar dianggap hadir dalam matakuliah.
Proses Standar	1. Mahasiswa memilih menu absen 2. Mahasiswa mengisi kode absensi 3. Mahasiswa mengirim lokasi saat ini
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*

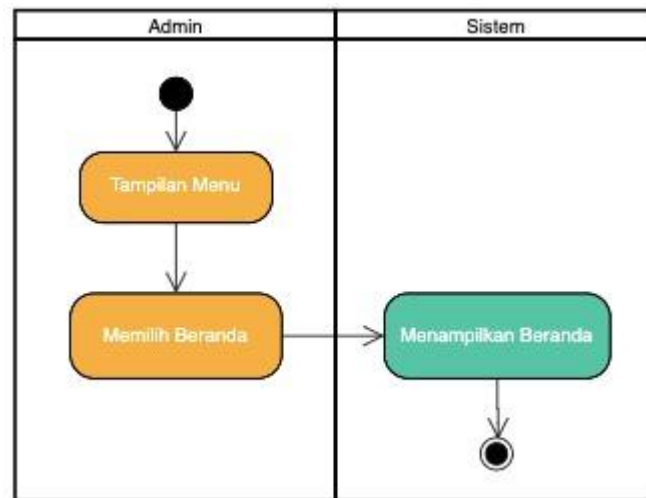
Pada Gambar 4.2 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* dan *user* ketika memasukkan *username* dan *password*, lalu sistem akan menampilkan melakukan validasi.



Gambar 4.2 *Activity Diagram* Login (Admin dan User)

b. *Activity Diagram Beranda/Dashboard*

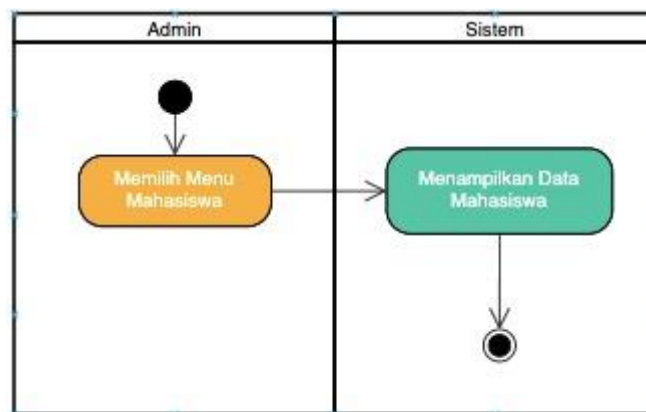
Pada Gambar 4.14 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* ketika memilih menu beranda, lalu sistem akan menampilkan menu beranda yang berisikan tampilan menu beranda.



Gambar 4.3 Activity Diagram Beranda/Dashboard

c. *Activity Diagram Tampil Data Mahasiswa*

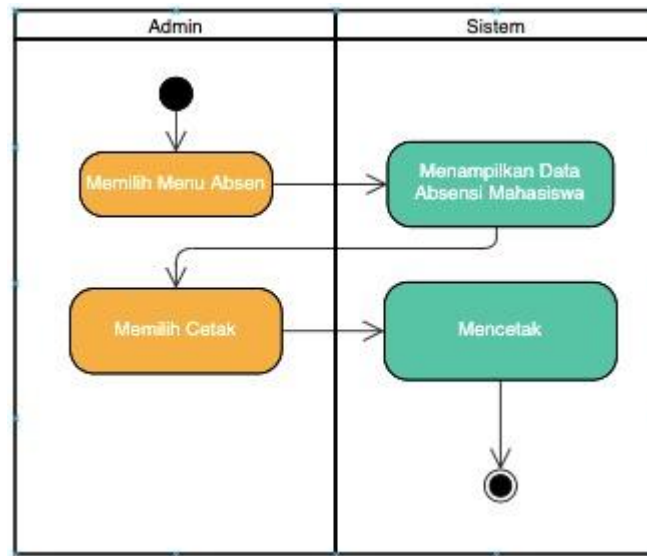
Pada Gambar 4.4 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan halaman data mahasiswa.



Gambar 4.4 Activity Diagram Tampil Data Absensi

d. *Activity Diagram Cetak Absen*

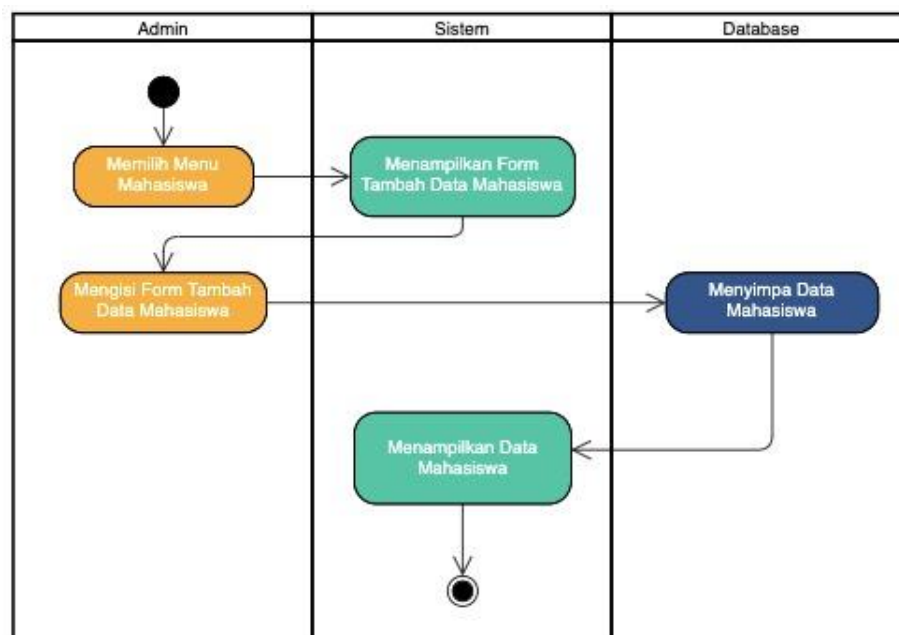
Pada Gambar 4.5 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu absen lalu sistem menampilkan halaman cetak absen.



Gambar 4.5 Activity Diagram Cetak Absen

e. *Activity Diagram Tambah Data Mahasiswa*

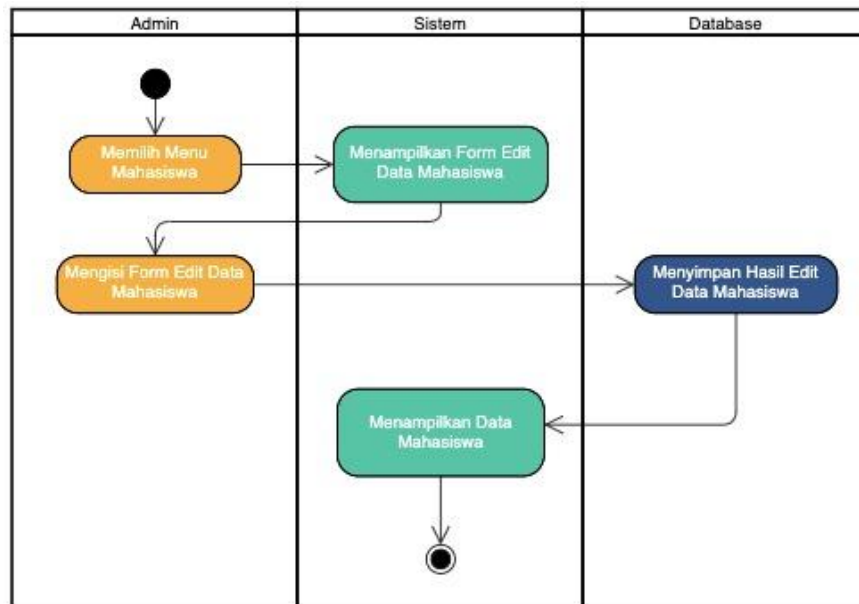
Pada Gambar 4.6 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan *form* tambah data mahasiswa, kemudian *admin* mengisi *form* tambah mahasiswa lalu *database* akan menyimpan data setelah itu sistem akan menampilkan data mahasiswa.



Gambar 4.6 Activity Diagram Tambah Data Mahasiswa

f. *Activity Diagram Edit Data mahasiswa*

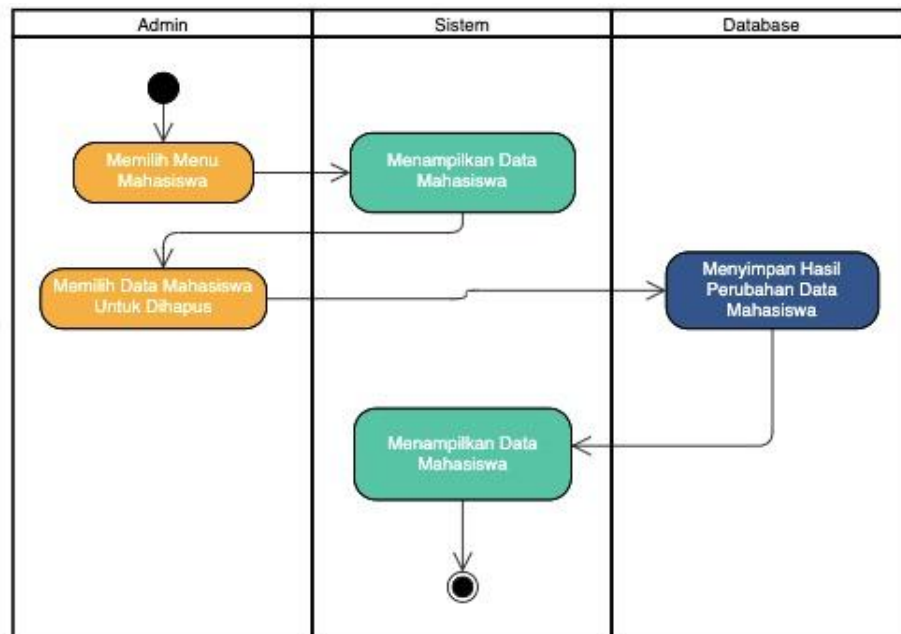
Pada Gambar 4.7 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan *form edit* data mahasiswa. Kemudian *admin* mengedit data mahasiswa, kemudian *database*



Gambar 4.7 Activity Diagram Edit Data Mahasiswa

g. *Activity Diagram Hapus Data Mahasiswa*

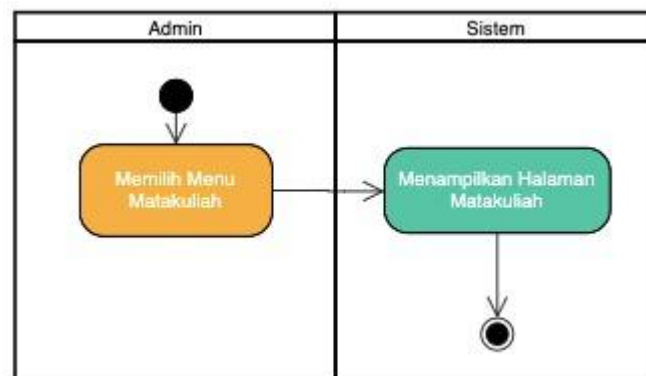
Pada Gambar 4.8 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan data mahasiswa lalu *admin* memilih hapus kemudian sistem menghapus data mahasiswa tersebut.



Gambar 4.8 Activity Diagram Hapus Data Mahasiswa

h. *Activity Diagram Mata Kuliah*

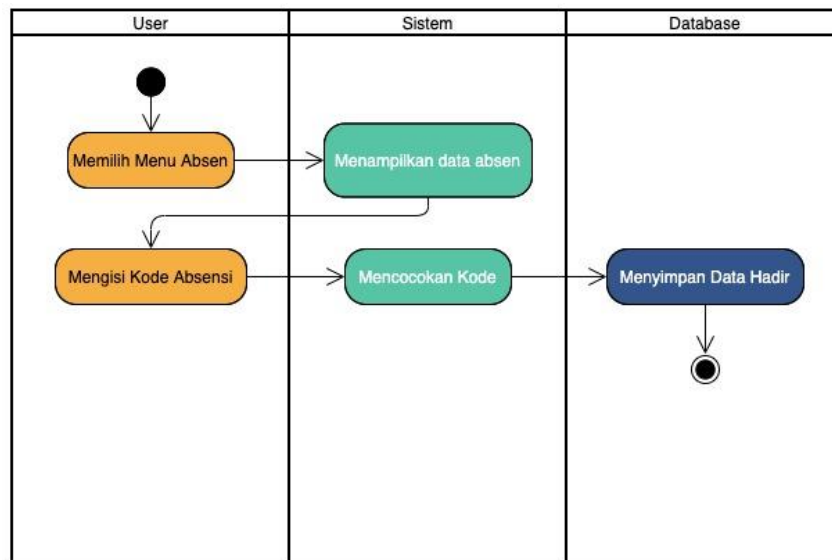
Pada Gambar 4.9 merupakan diagram aktivitas yang menunjukkan aktivitas *user* memilih menu mata kuliah lalu sistem menampilkan halaman mata kuliah.



Gambar 4.9 Activity Diagram Mata Kuliah

i. *Activity Diagram Absensi Kode*

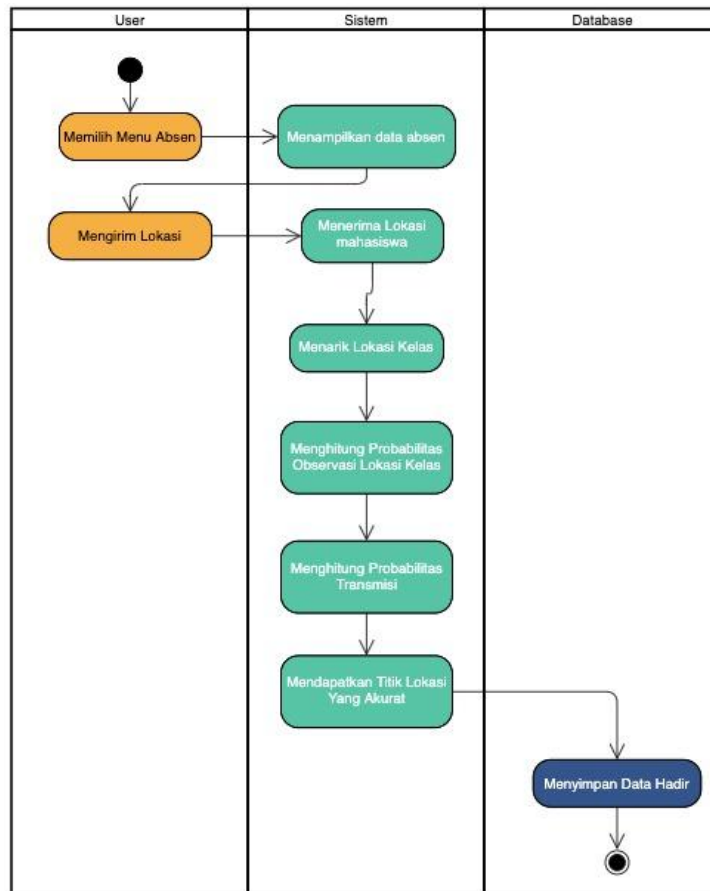
Pada Gambar 4.10 merupakan diagram aktivitas yang menunjukkan aktivitas *user* memilih menu absen lalu sistem akan meminta kode mata kuliah dan mencocokkan kode yang digenerate oleh dosen mata kuliah tersebut.



Gambar 4.10 Activity Diagram Absensi Kode

j. *Activity Diagram Absensi Lokasi*

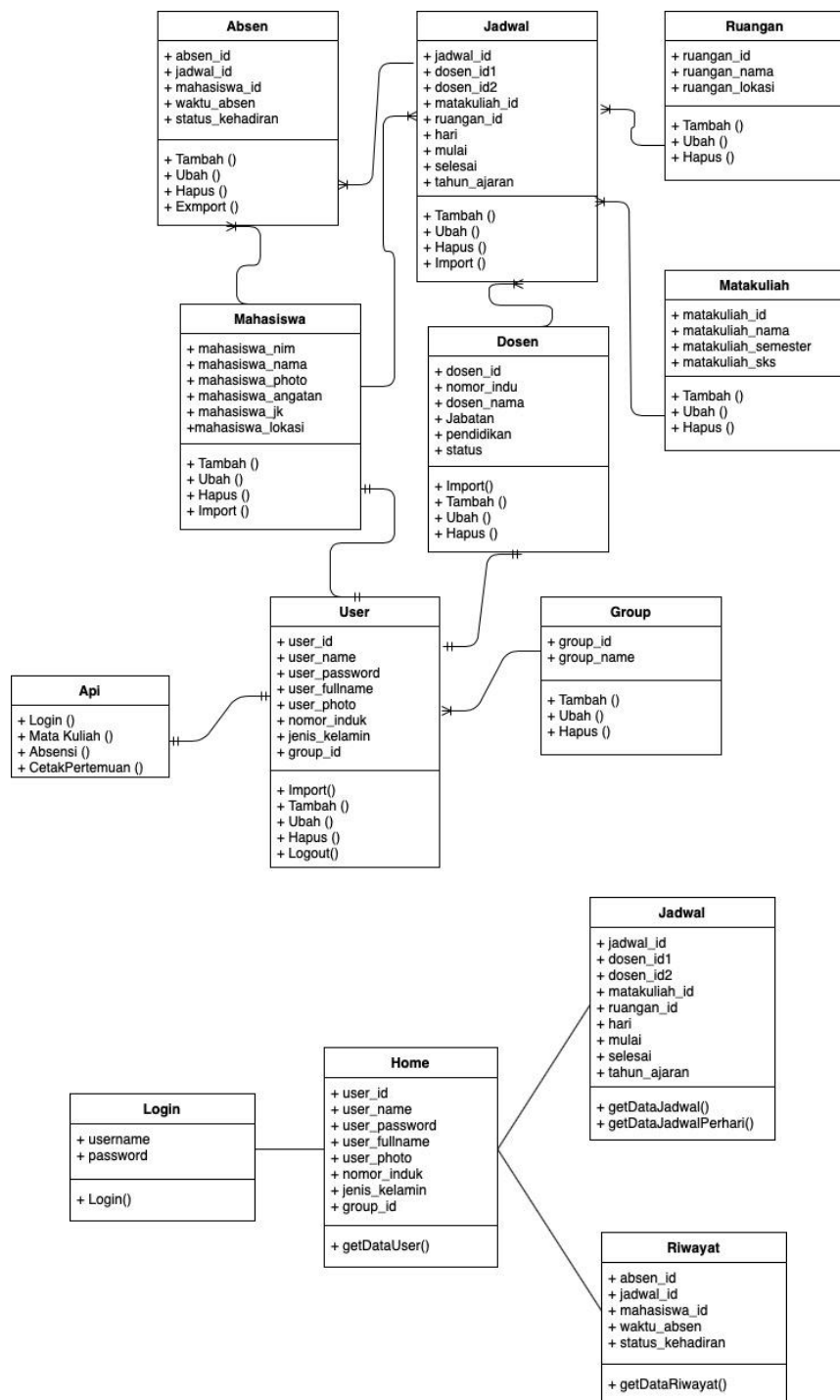
Pada Gambar 4.11 merupakan diagram aktivitas yang menunjukkan aktivitas *user* memilih menu absen lalu sistem akan menarik data lokasi mahasiswa, menarik data lokasi kelas setelah itu menghitung probabilitas observasi dari lokasi kelas. Setelah mendapatkan probabilitas observasi lokasi kelas selanjutnya melakukan observasi transmisi dengan cara mencocokkan lokasi hasil observasi kelas dengan lokasi mahasiswa setelah itu akan mendapatkan lokasi yang akurat sesuai lokasi kelas.



Gambar 4.11 Activity Diagram Absensi Lokasi

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* sistem.

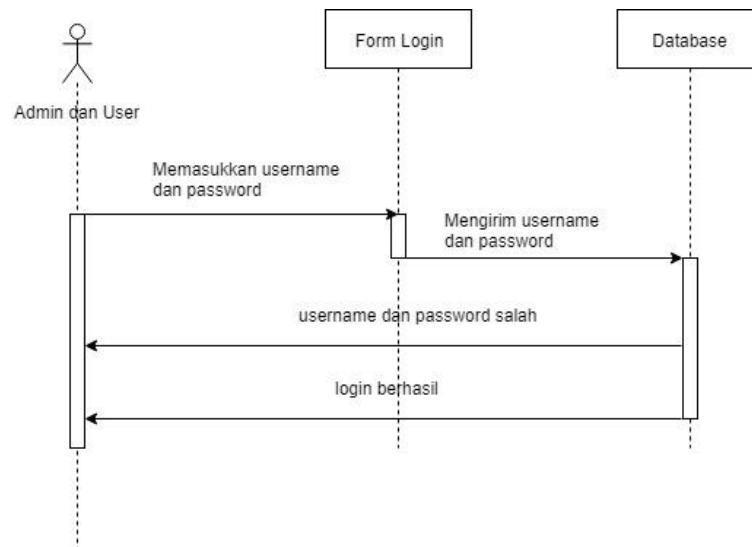


Gambar 4.12 Class Diagram

4. Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang digambarkan terhadap waktu. Berikut ini adalah *Sequence Diagram* yang akan menggambarkan interaksi antar objek dan sistem.

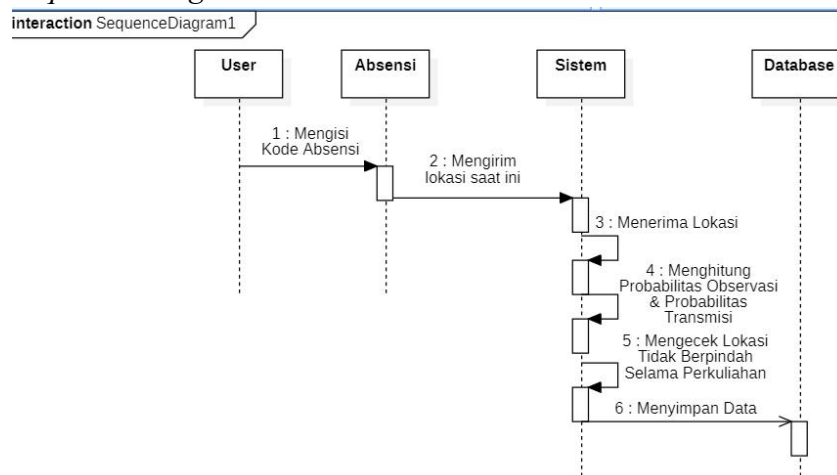
a. Sequence Diagram Login



Gambar 4. 13 Sequence Diagram Login

Gambar 4. 13 adalah *sequence* yang menunjukkan proses login bagi admin dan user dimana *actor* memasukkan *username* dan *password* lalu divalidasi oleh *database*.

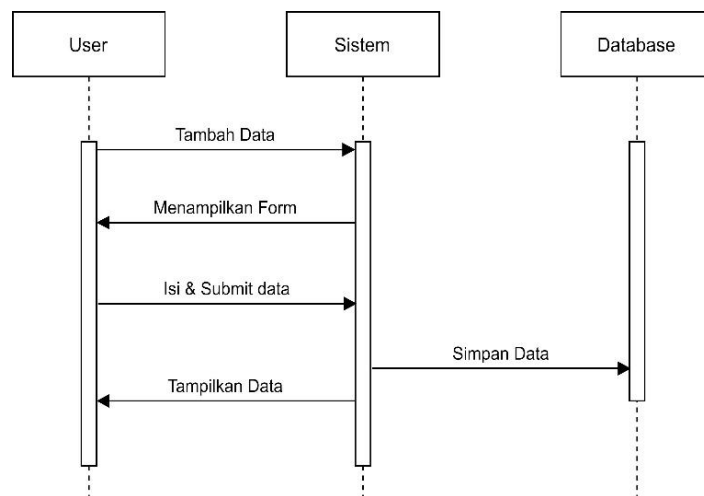
b. Sequence Diagram Absensi



Gambar 4. 14 Sequence Diagram Absensi

Gambar 4.14 adalah *sequence* yang menunjukkan proses absensi di mana mahasiswa mengisi kode absensi dan aplikasi akan mengirimkan lokasi keberadaan mahasiswa saat ini sampai proses pembelajaran selesai lalu sistem akan mengecek kesamaan lokasi dari menit pertama pembelajaran sampai proses pembelajaran selesai lalu menyimpan ke dalam database.

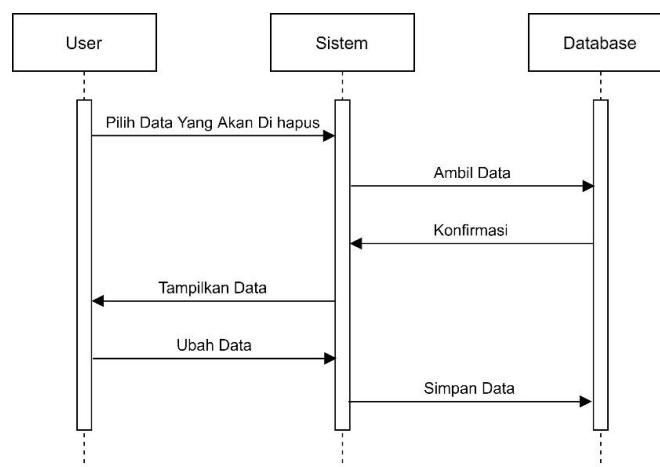
c. *Sequence Tambah Data*



Gambar 4. 15 Sequence Diagram Tambah Data

Gambar 4.15 adalah *sequence* yang menunjukkan proses tambah data di mana mahasiswa mengisi data memilih tombol submit, kemudian *database* akan lalu sistem akan menampilkan data.

d. *Sequence Hapus Data*



Gambar 4. 16 Sequence Diagram Hapus Data

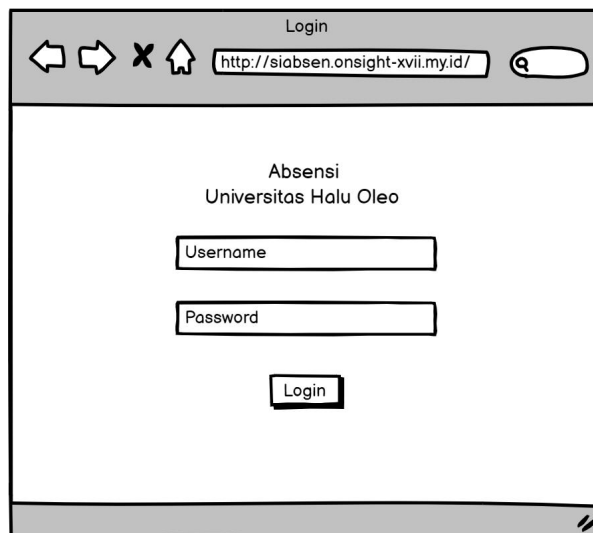
Gambar 4.16 adalah *sequence* yang menunjukkan proses menghapus data di mana mahasiswa memilih data yang ingin di hapus, kemudian *database* akan menghapus data tersebut dan mengkonfirmasi pada sistem kemudian sistem akan menampilkan data.

4.4 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*.

4.4.1 *Form Login*

Pada *form login* menampilkan *username* dan *password* yang akan di isi oleh *admin* untuk masuk ke dalam web. Bentuk dari tampilan *form login* dapat dilihat pada gambar berikut.

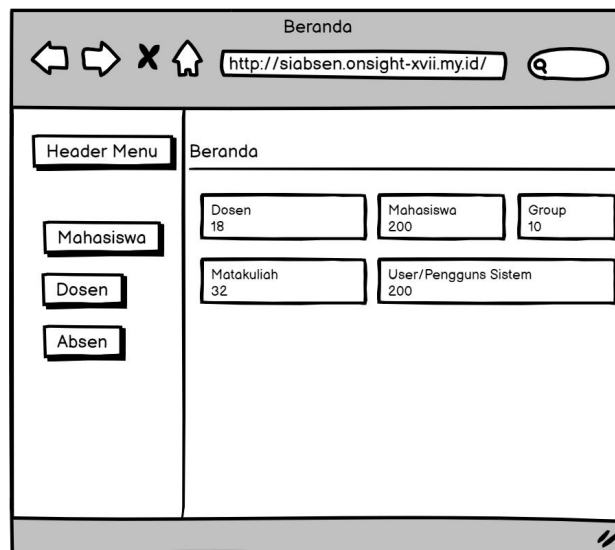


The image shows a web browser window with the title 'Login'. The address bar contains the URL 'http://siabsen.onsight-xvii.my.id/'. The main content area displays the text 'Absensi Universitas Halu Oleo' centered. Below this text are two input fields: 'Username' and 'Password'. A 'Login' button is positioned below the password field. The browser window has a standard toolbar with back, forward, and search icons.

Gambar 4. 17 Menu *Dashboard*

4.4.2 *Menu Dashboard*

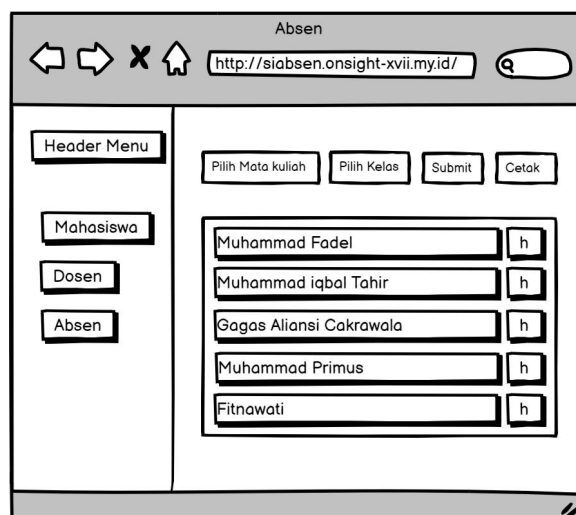
Menu Dashboard menampilkan data aplikasi secara umum. Mulai dari jumlah dosen, mahasiswa, mata kuliah dan *user*. Bentuk dari tampilan menu *dashboard* dapat dilihat pada gambar berikut.



Gambar 4.18 Menu Dashboard

4.4.3 Menu Absen

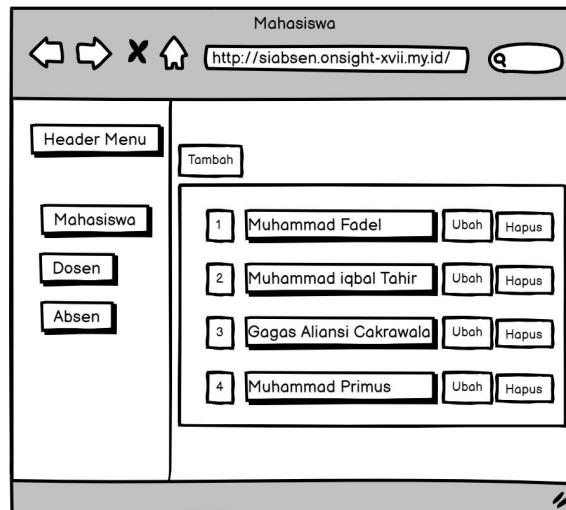
Menu absen menampilkan data absensi berdasarkan nama mata kuliah dan jurusan. Terdapat tabel yang berisikan nama dan nim masing-masing mahasiswa serta daftar kehadirannya dari pertemuan 1 sampai pertemuan ke 16. Serta informasi presentase kehadiran mahasiswa tersebut. Selain itu, juga terdapat tombol cetak untuk melakukan *print out* data absensi tersebut. Bentuk dari tampilan menu absen dapat dilihat pada gambar berikut.



Gambar 4. 19 Menu Absen

4.4.4 Menu Mahasiswa

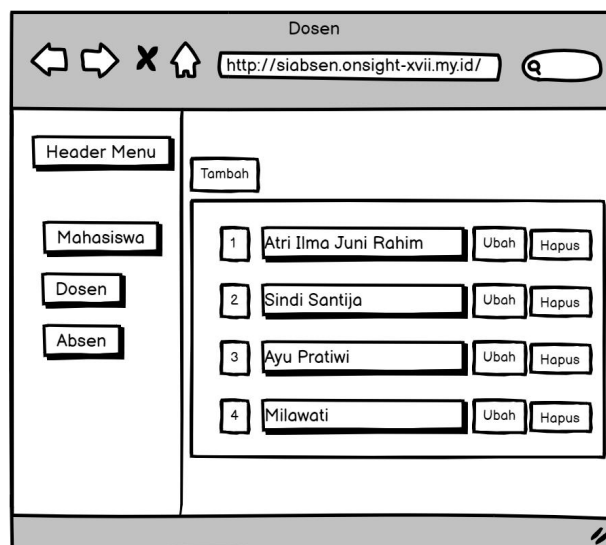
Menu mahasiswa berisikan data pribadi setiap mahasiswa. Bentuk dari tampilan menu mahasiswa dapat dilihat pada gambar berikut.



Gambar 4.20 Menu Mahasiswa

4.4.5 Menu Dosen

Menu dosen berisi data setiap dosen yang ada pada Universitas Halu Oleo. Terdapat pula foto dari masing data dosen agar lebih memudahkan dalam mencari data atau informasi tentang dosen. Bentuk dari tampilan menu dosen dapat dilihat pada gambar berikut.



Gambar 4.21 Menu Dosen

4.4.6 Halaman *Login Mobile*

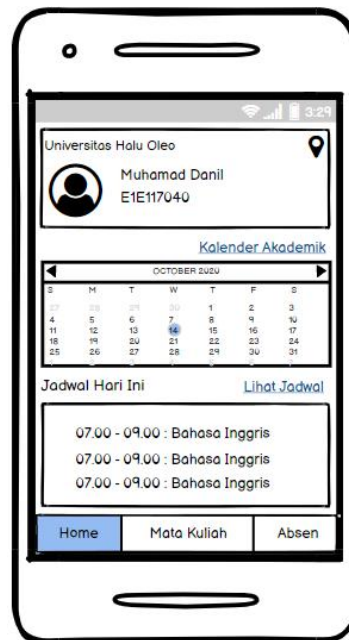
Halaman login pada aplikasi *mobile* menampilkan *form login* yang terdiri dari *username* dan *password*. Setiap *field* harus diisi agar dapat melakukan *login*. Selain itu, *form* ini juga telah ditambahkan fungsi validasi agar dapat menyaring data *user* yang benar. Bentuk dari tampilan *login mobile*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



Gambar 4.22 Halaman *Login Mobile*

4.4.7 Halaman *Home Mobile*

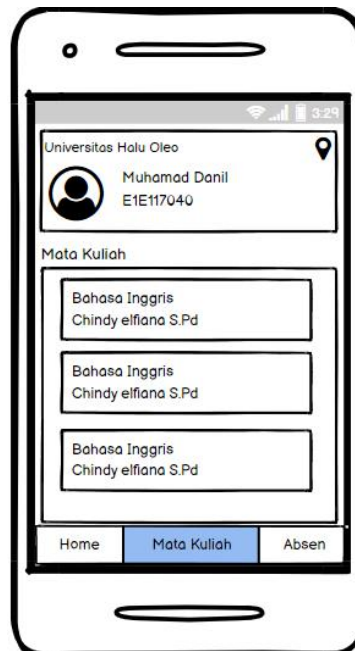
Halaman *home mobile* menampilkan identitas dari user, kalender akademik dan jadwal hari ini. Bentuk dari tampilan *home mobile* dapat dilihat pada gambar berikut.



Gambar 4.23 Halaman *Home Mobile*

4.4.8 Halaman Mata kuliah *Mobile*

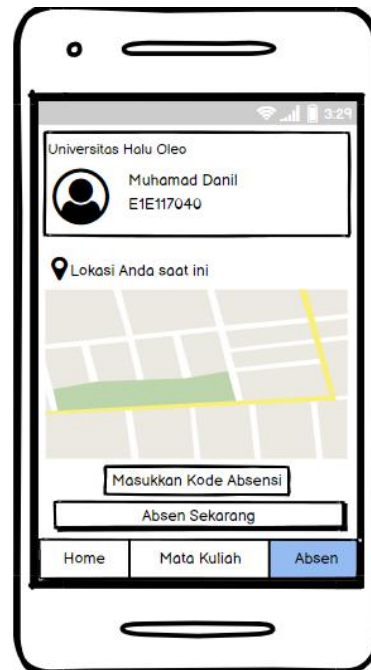
Halaman mata kuliah mobile menampilkan daftar mata kuliah dan nama dosen yang mengajar. Bentuk dari tampilan mata kuliah mobile dapat dilihat pada gambar berikut.



Gambar 4.24 Halaman Mata Kuliah *Mobile*

4.4.9 Halaman Absensi

Halaman absensi menampilkan *maps* lokasi saat ini dan inputan kode absensi sesuai mata pelajaran berlangsung. Adapun bentuk dari tampilan halaman absensi dapat dilihat pada gambar berikut.



Gambar 4. 25 Halaman Absensi

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 *Spatial Map Matching* dan pengujian sistem.

5.2 Data

Data yang digunakan pada pengembangan sistem Absensi berbasis *Android* dan *Web* adalah data diri dari mahasiswa Universitas Halu Oleo, Fakultas Teknik, jurusan Teknik Informatika, angkatan 2019 ini didapatkan dari staf fakultas.

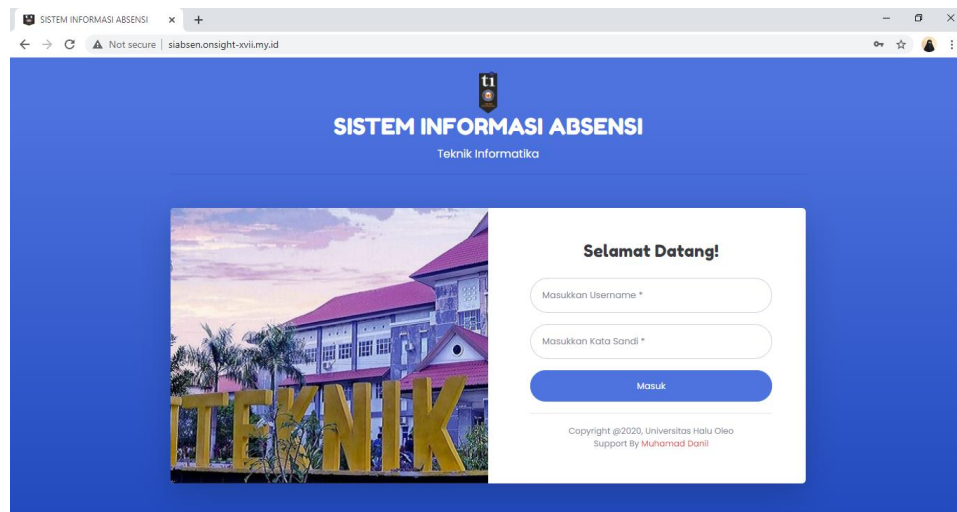
5.2.1 Implementasi *Interface*

Implementasi *interface* dibuat berdasarkan perancangan Admin Dosen dan Mahasiswa. Implementasi disesuaikan dengan komponen yang tersedia pada *framework Android* dan *Web*. Implementasi *interface admin*, dosen dan mahasiswa akan ditampilkan dalam bentuk *screen shoot Android* dan *web*.

5.3 Implementasi *Interface System* Berbasis Web

5.3.1 Halaman *Login*

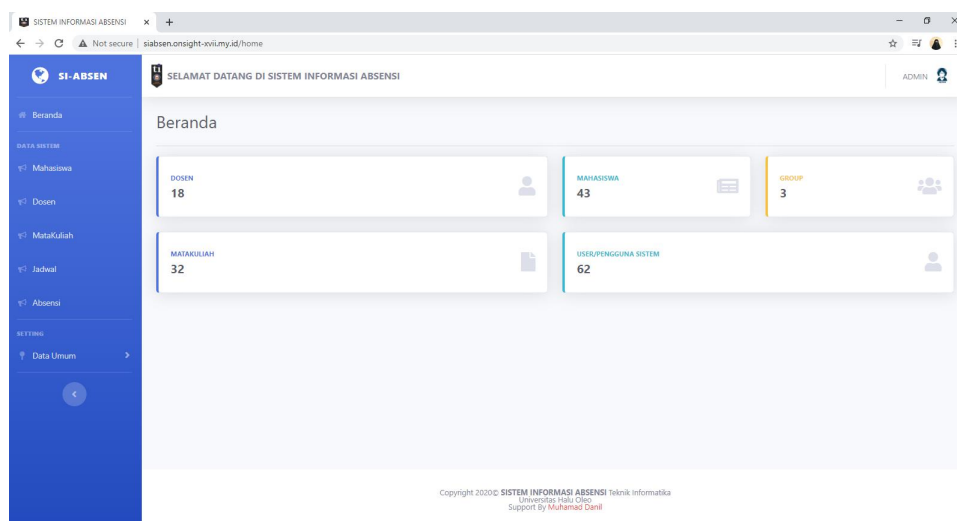
Pada Gambar 5.1 merupakan halaman *login* dimana *admin* dan dosen memasukkan *username* dan *password* terlebih dahulu untuk bisa masuk kedalam sistem.



Gambar 5.1 Tampilan *Login* (Admin dan Dosen)

5.3.2 Halaman *Dashboard* (Admin)

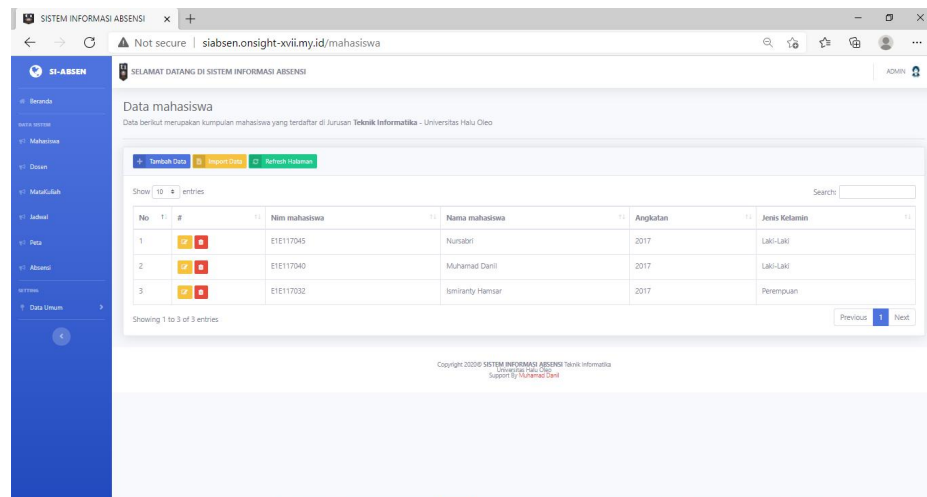
Pada Gambar 5.2 merupakan halaman pertama setelah admin berhasil melakukan *login* kedalam sistem. Halaman ini berisi tentang jumlah dosen, mahasiswa, mata kuliah dan *user*.



Gambar 5.2 Tampilan *Dashboard* (Admin)

5.3.3 Halaman Mahasiswa

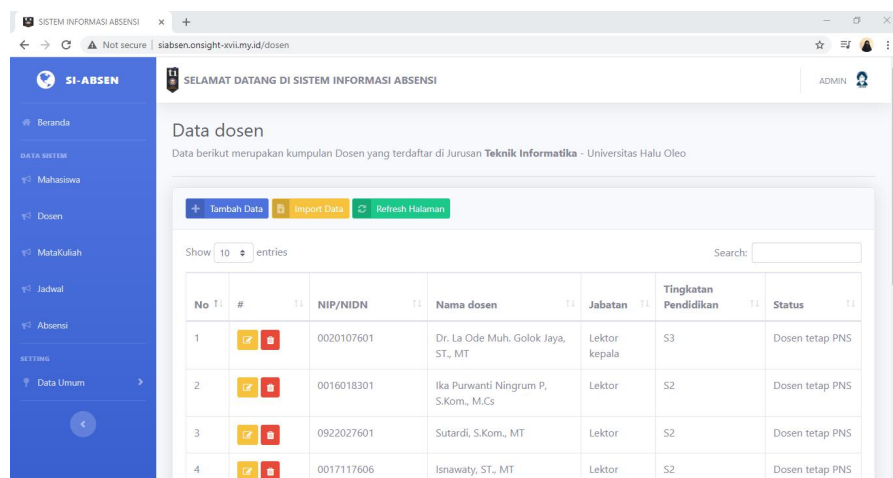
Pada Gambar 5.3 merupakan halaman mahasiswa setelah admin mengklik menu mahasiswa. Halaman ini berisi tentang data mahasiswa dan juga beberapa tombol yaitu tombol tambah data, *import* data, *refresh* halaman, hapus dan *edit* data mahasiswa.



Gambar 5.3 Tampilan Mahasiswa (Admin)

5.3.4 Halaman Dosen

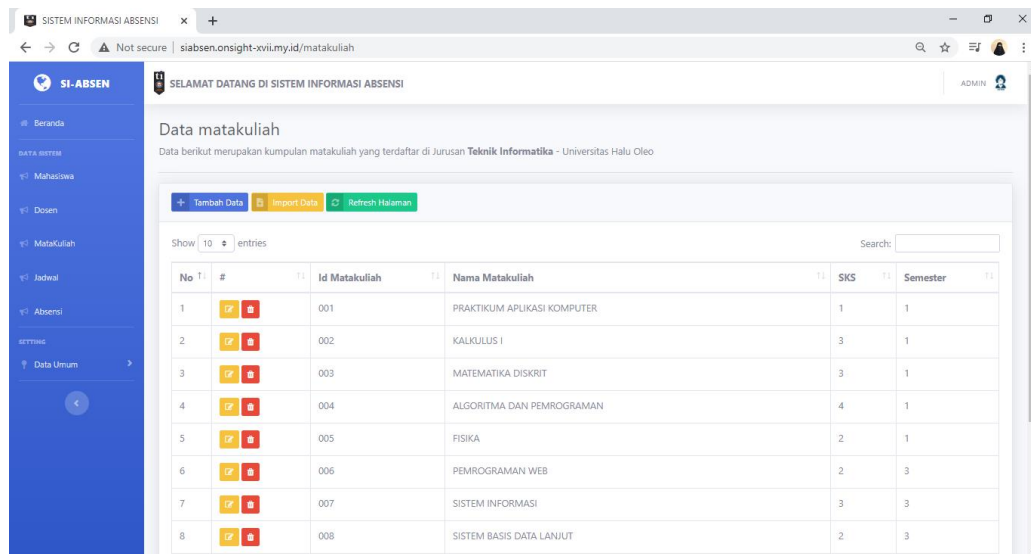
Pada Gambar 5.4 merupakan halaman dosen setelah admin mengklik menu dosen. Halaman ini berisi tentang data dosen dan juga beberapa tombol yaitu tombol tambah data, *import* data, *refresh* halaman, hapus dan edit data dosen.



Gambar 5.4 Tampilan Dosen (Admin)

5.3.5 Halaman Mata kuliah

Pada Gambar 5.5 merupakan halaman matakuliah setelah admin mengklik menu mata kuliah. Halaman ini berisi tentang data matakuliah dan juga beberapa tombol yaitu tombol tambah data, *import* data, *refresh* halaman, hapus dan *edit* data mata kuliah.



SI-ABSEN

SELAMAT DATANG DI SISTEM INFORMASI ABSENSI

Data matakuliah

Data berikut merupakan kumpulan matakuliah yang terdaftar di Jurusan Teknik Informatika - Universitas Halu Oleo

Tambah Data Import Data Refresh Halaman

Show 10 entries

No	#	Id Matakuliah	Nama Matakuliah	SKS	Semester
1		001	PRAKTIKUM APLIKASI KOMPUTER	1	1
2		002	KALKULUS I	3	1
3		003	MATEMATIKA DISKRIT	3	1
4		004	ALGORITMA DAN PEMROGRAMAN	4	1
5		005	FISIKA	2	1
6		006	PEMROGRAMAN WEB	2	3
7		007	SISTEM INFORMASI	3	3
8		008	SISTEM BASIS DATA LANJUT	2	3

Gambar 5.5 Tampilan Mata kuliah (Admin)

5.3.6 Halaman Jadwal

Pada Gambar 5.6 merupakan halaman jadwal kuliah setelah *admin* mengklik menu jadwal. Halaman ini berisi tentang data jadwal mata kuliah dan juga beberapa tombol yaitu tombol tambah data, *import* data, *refresh* halaman, *export* jadwal, hapus, *edit* data mata kuliah dan kolom pilih mata kuliah juga tombol sort untuk menampilkan jadwal berdasarkan mata kuliah.

No	#	KodeMk	Mata Kuliah	SKS	Dosen 1	Dosen 2	Hari	Mulai	Selesai	Ruangan	Kelas
1		002	KALKULUS I	3	Natalis Ransi, S.Kom, M.Kom	Rahmat Ramadhan, S.Si, M.Cs	Jumat	07:30:00	10:00:00	IT-1	Garjil
2		002	KALKULUS I	3	Natalis Ransi, S.Kom, M.Kom	Rahmat Ramadhan, S.Si, M.Cs	Jumat	07:30:00	10:00:00	IT-1	Garjil
3		002	KALKULUS I	3	Natalis Ransi, S.Kom, M.Kom	Rahmat Ramadhan, S.Si, M.Cs	Jumat	07:30:00	10:00:00	IT-1	Garjil
4		002	KALKULUS I	3	Natalis Ransi, S.Kom, M.Kom	Rahmat Ramadhan, S.Si, M.Cs	Jumat	07:30:00	10:00:00	IT-1	Garjil
5		002	KALKULUS I	3	Natalis Ransi, S.Kom, M.Kom	Rahmat Ramadhan, S.Si, M.Cs	Jumat	07:30:00	10:00:00	IT-1	Garjil
6		002	KALKULUS I	3	Natalis Ransi, S.Kom, M.Kom	Rahmat Ramadhan, S.Si, M.Cs	Jumat	07:30:00	10:00:00	IT-1	Garjil
7		002	KALKULUS I	3	Natalis Ransi, S.Kom, M.Kom	Rahmat Ramadhan, S.Si, M.Cs	Jumat	07:30:00	10:00:00	IT-1	Garjil

Gambar 5.6 Tampilan Absensi (*Admin*)

5.3.7 Halaman Absensi

Pada Gambar 5.7 merupakan halaman absensi setelah admin mengklik menu absensi. Halaman ini berisi tentang data absensi mahasiswa saat pelaksanaan kuliah dan juga beberapa tombol yaitu tombol *refresh* halaman, dan *export* jadwal.

No	Nim Mahasiswa	Nama Mahasiswa	MataKuliah	Waktu Absen	Status Kehadiran	Kelas	Tahun Ajaran
1	E1E119062	Lily Karmila	ORGANISASI DAN ARSITEKTUR KOMPUTER	2021-02-25	Alpha	Garjil	20201
2	E1E119033	Muhammad Rafli	ORGANISASI DAN ARSITEKTUR KOMPUTER	2021-02-25	Alpha	Garjil	20201
3	E1E119017	A.Gamma Brilliant Arizha	ORGANISASI DAN ARSITEKTUR KOMPUTER	2021-02-25	Alpha	Garjil	20201
4	E1E119053	Fadli Brilian Daksa	ORGANISASI DAN ARSITEKTUR KOMPUTER	2021-02-25	Alpha	Garjil	20201
5	E1E119049	Annisaa Dwi Amalia Santoso	ORGANISASI DAN ARSITEKTUR KOMPUTER	2021-02-25	Alpha	Garjil	20201
6	E1E119034	Nadisa Qur'atul 'Ain	ORGANISASI DAN ARSITEKTUR KOMPUTER	2021-02-25	Alpha	Garjil	20201
7	E1E119022	Asfan Meliawan Kahanuddin	ORGANISASI DAN ARSITEKTUR KOMPUTER	2021-02-25	Alpha	Garjil	20201
8	E1E119076	Ridha entim aprilya	ORGANISASI DAN ARSITEKTUR KOMPUTER	2021-02-25	Alpha	Garjil	20201
9	E1E119016	Wade St Fadila	ORGANISASI DAN ARSITEKTUR KOMPUTER	2021-02-25	Alpha	Garjil	20201

Gambar 5.7 Tampilan Absensi (*Admin*)

5.3.8 Halaman Data Umum

Pada halaman data umum terdapat dua tombol yaitu *user* dan ruangan. Pada Gambar 3.8 merupakan tampilan dari halaman *user*, halaman ini berisi

tentang *user* yang terdaftar di dalam aplikasi dan juga beberapa tombol yaitu tombol tambah data, *import* data, *refresh* halaman, hapus dan *edit* data *user*.

No	ID	Username	Fullname	Group
1		admin	Administrator Sistem	Administrator
2	E1E118067		Chindy Effana Suci Natasya	Mahasiswa
3	E1E117040		Muhamad Daniil	Mahasiswa
4	0020107601		Dr. La Ode Muh. Golok Jaya, ST., MT	Dosen
5	0016018301		Ika Purwanti Ningrum P. S.Kom., M.Cs	Dosen
6	0922027601		Sutardi, S.Kom., MT	Dosen
7	0017117606		Isnawaty, ST., MT	Dosen
8	0025047107		Bambang Pramono, ST., MT	Dosen

Gambar 5.8 Tampilan User (Admin)

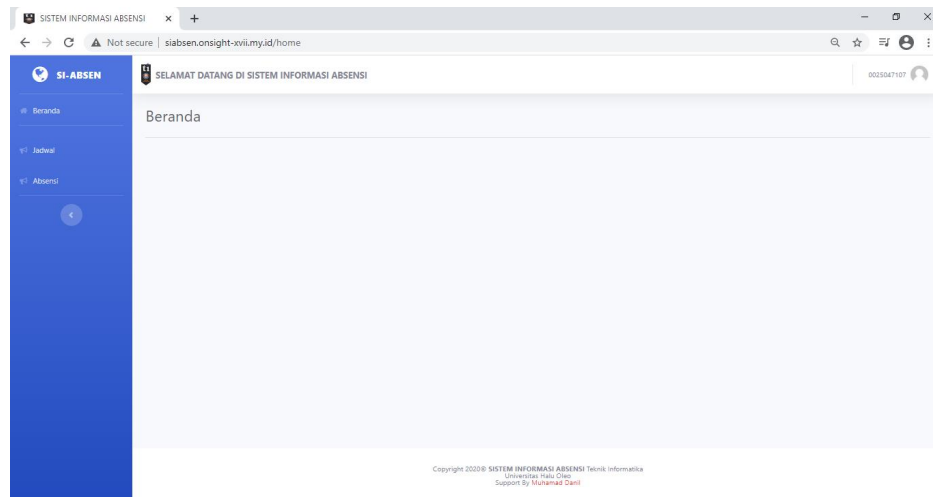
Pada Gambar 3.8 merupakan tampilan dari halaman ruangan, halaman ini berisi tentang ruangan yang ada di jurusan teknik informatika dan juga beberapa tombol yaitu tombol tambah data dan *refresh* halaman data ruangan.

No	ID	Ruangan	Lokasi
1		IT-1	-3.9741058839990986, 122.52200135954097
2		IT-2	-3.9741058839990986, 122.52200135954097
3		IT-3	-3.9741058839990986, 122.52200135954097
4		Software Engineering	-3.9741058839990986, 122.52200135954097
5		Computer systems and network laboratory	-3.9741058839990986, 122.52200135954097
6		Computer science & artificial intelligence	-3.9741058839990986, 122.52200135954097

Gambar 5.9 Tampilan Ruangan (Admin)

5.3.9 Halaman *Dashboard* (Dosen)

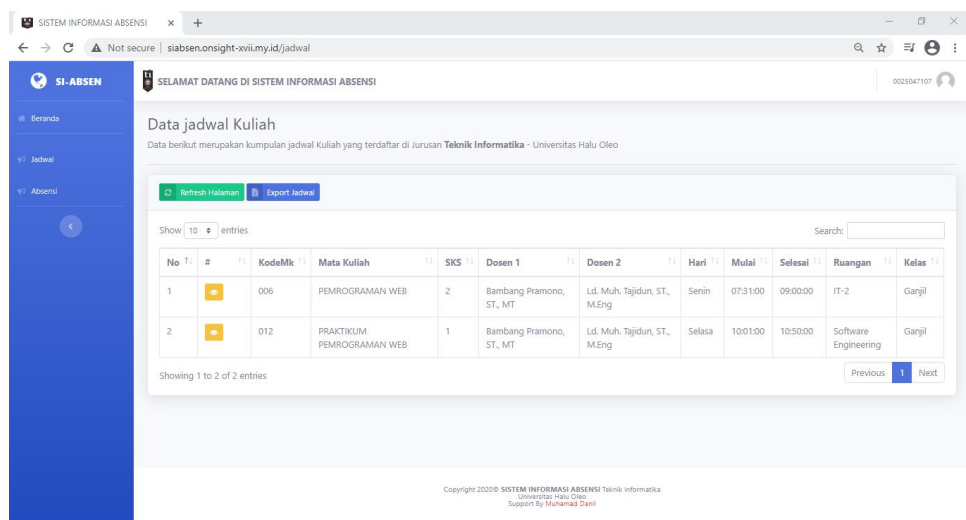
Pada Gambar 5.10 merupakan halaman pertama setelah dosen berhasil melakukan *login* kedalam sistem.



Gambar 5.10 Tampilan *Dashboard* (Dosen)

5.3.10 Halaman Jadwal (Dosen)

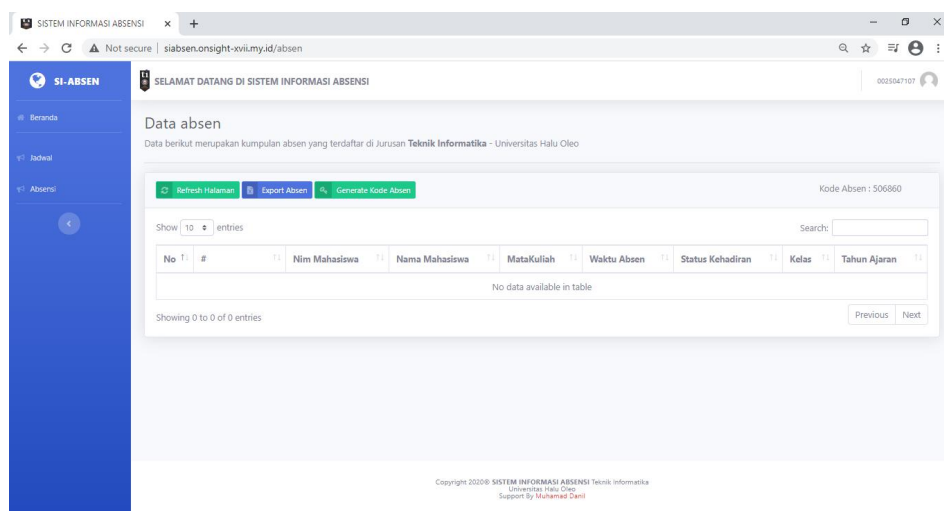
Pada Gambar 5.11 merupakan halaman jadwal setelah admin mengklik menu jadwal. Halaman ini berisi tentang data jadwal mata kuliah dan juga beberapa tombol yaitu tombol *refresh* halaman, *export* jadwal dan tombol melihat lokasi mahasiswa.



Gambar 5.11 Tampilan Jadwal (Dosen)

5.3.11 Halaman Absensi (Dosen)

Pada Gambar 5.12 merupakan halaman absensi setelah dosen mengklik menu absensi. Halaman ini berisi tentang data absensi mahasiswa saat pelaksanaan kuliah dan juga beberapa tombol yaitu tombol *refresh* halaman, *export* jadwal dan *generate* kode absen yang berbeda – beda dari setiap dosen yang dapat dikirimkan kepada mahasiswa dan dapat digunakan oleh mahasiswa yang berada diluar kota untuk melakukan absensi.

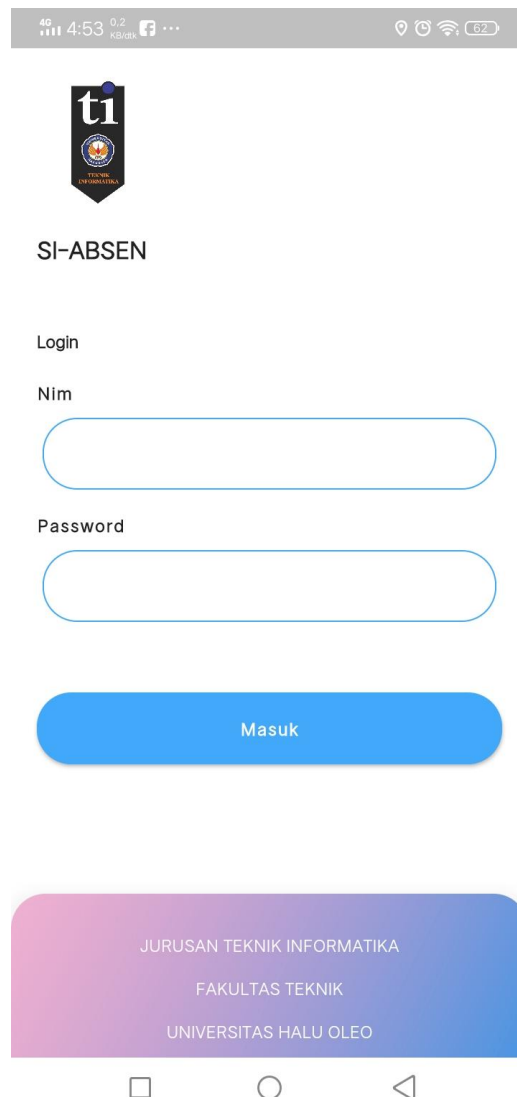


Gambar 5.12 Tampilan Absensi (Dosen)

5.4 Implementasi *Interface System* Berbasis *Mobile*

5.4.1 Halaman *Login*

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



4G 4:53 0.2 KB/s

ti

SI-ABSEN

Login

Nim

Password

Masuk

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HALU OLEO

Gambar 5.13 Tampilan Login (Mahasiswa)

5.4.2 Halaman *Dashboard*

Pada halaman *Dashboard* ini merupakan halaman pertama setelah mahasiswa berhasil melakukan *login* kedalam sistem. Halaman ini berisi tombol presensi dan riwayat.

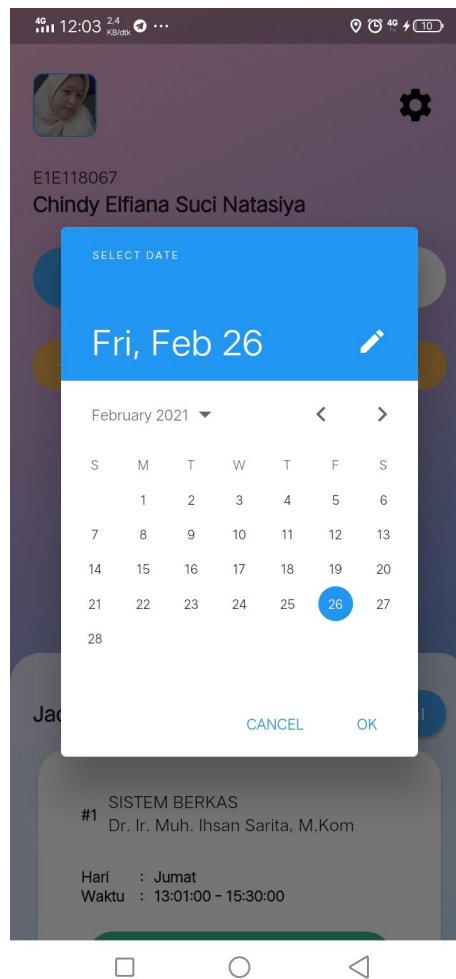
1. Tampilan Halaman Presensi

Pada Gambar 5.14 merupakan tampilan halaman presensi yang berisi tentang jadwal matakuliah sesuai dengan hari.



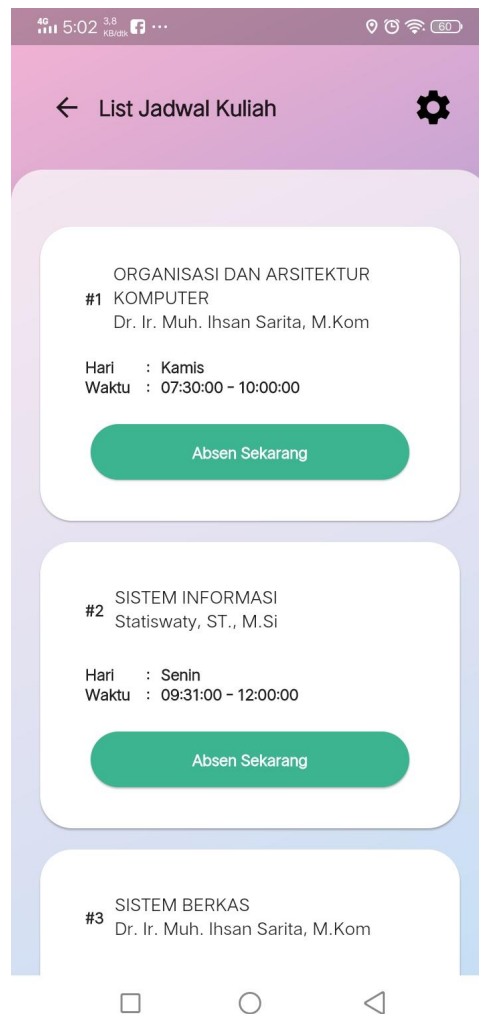
Gambar 5.14 Tampilan Halaman Presensi (Mahasiswa)

Pada halaman ini juga terdapat tambahan fitur kalender untuk melihat tanggal yang bisa dilihat pada Gambar 5.15.



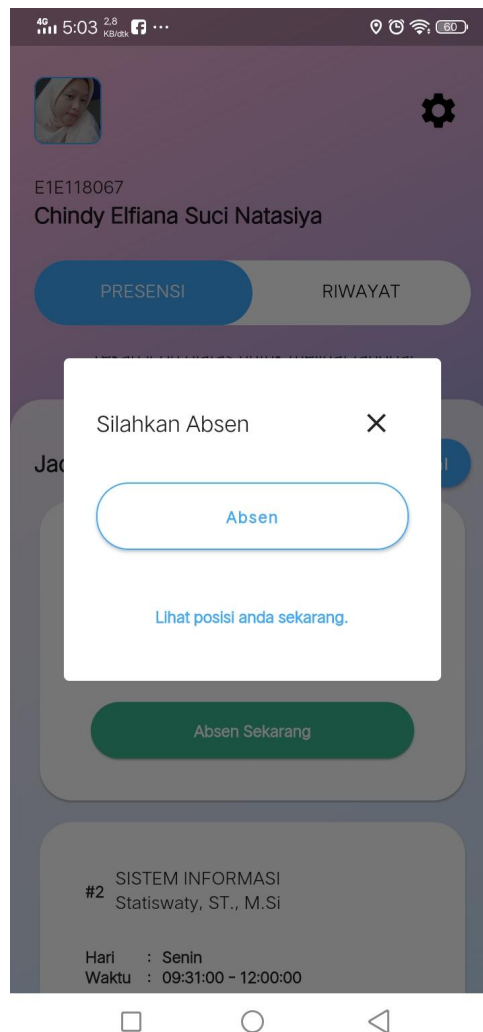
Gambar 5.15 Tampilan Fitur Kalender (Mahasiswa)

Pada halaman ini juga terdapat tombol semua jadwal untuk melihat semua mata kuliah yang di program semester ini yang bisa dilihat pada Gambar 5.16.



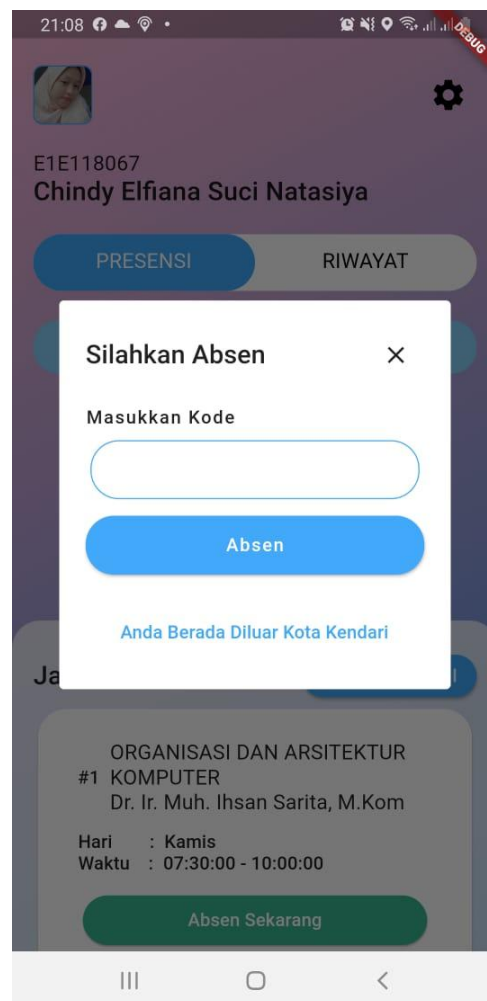
Gambar 5.16 Tampilan Halaman Semua Jadwal (Mahasiswa)

Juga terdapat tombol absen sekarang agar mahasiswa yang berada di dalam kota mengisi absen yang ditunjukkan pada Gambar 5.17.



Gambar 5.17 Tampilan Halaman Absensi (Mahasiswa)

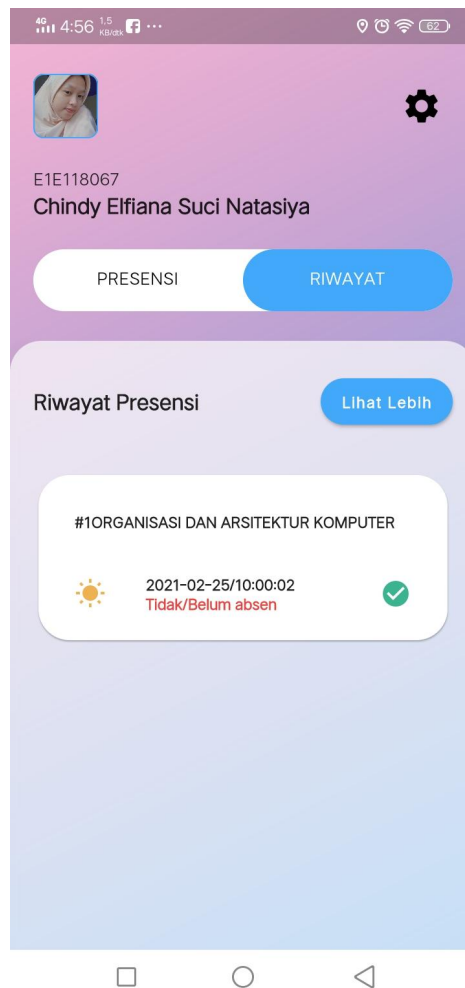
Kemudian ada juga absensi yang bisa digunakan oleh mahasiswa yang berada diluar kota. Dimana sistem mengecek lokasi mahasiswa secara otomatis dengan cara mengabsen menggunakan kode yang diberikan oleh dosen mata kuliah yang ditunjukan pada Gambar 5.17.



Gambar 5.18 Tampilan Halaman Absensi *Barcode* (Mahasiswa)

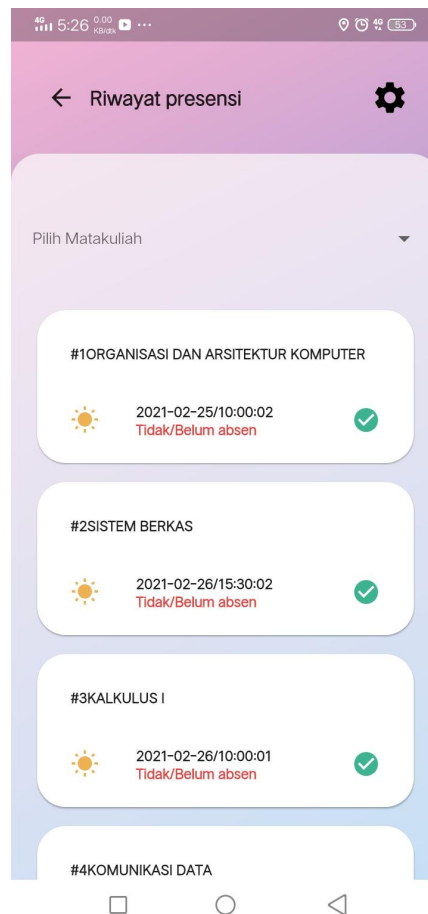
2. Tampilan Halaman Riwayat

Pada Gambar 5.19 merupakan tampilan halaman riwayat yang berisi tentang riwayat absensi yang dilakukan oleh mahasiswa dan juga terdapat pemberitahuan apakah mahasiswa telah melakukan absensi tepat waktu atau tidak.



Gambar 5.19 Tampilan Halaman Riwayat Presensi (Mahasiswa)

Selain itu pada halaman ini terdapat tombol lihat lebih yang akan menampilkan semua riwayat absensi mahasiswa yang bisa dilihat pada gambar Gambar 5.20.



Gambar 5.20 Tampilan Halaman Lihat Lebih (Mahasiswa)

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*

<i>Input/Event</i>	<i>Output</i>	Hasil Uji
Memilih menu <i>dashboard</i>	Menampilkan menu <i>dashboard</i>	Berhasil
Memilih menu absen	Menampilkan menu absen	Berhasil
Memilih menu mata kuliah	Menampilkan menu mata kuliah	Berhasil
Memilih menu dosen	Menampilkan menu dosen	Berhasil
Masuk ke halaman home <i>mobile</i>	Menampilkan halaman home <i>mobile</i>	Berhasil
Masuk ke halaman mata kuliah <i>mobile</i>	Menampilkan halaman mata kuliah <i>mobile</i>	Berhasil
Masuk ke halaman mahasiswa <i>mobile</i>	Menampilkan halaman mahasiswa <i>mobile</i>	Berhasil
Mengaktifkan <i>scanner</i>	Mengaktifkan kamera dan menampilkan alat <i>scanner</i>	Berhasil

5.5.2 Pengujian Metode *Spatial Map Matching*

Pengujian ini dilakukan paling akhir setelah beberapa pengujian aplikasi selesai. Pengujian aplikasi ini perlu dilakukan agar dapat mengetahui tingkat akurasi posisi lokasi mahasiswa yang diperbaiki menggunakan metode *spatial map matching*.

Tabel 5.2 Kordinat awal mahasiswa mengabsen

No.	Nim	Latitude	Longitude
1.	E1E118067	-4.037596	122.555989
2.	E1E119062	-4.038190	122.523980
3.	E1E119033	-4.037230	122.555190
4.	E1E119017	-4.037596	122.535800
5.	E1E119053	-4.037516	122.572321
6.	E1E119049	-4.037886	122.555278
7.	E1E119034	-4.037589	122.555467
8.	E1E119022	-4.037386	122.555890
9.	E1E119076	-4.037986	122.555370
10.	E1E119016	-4.037522	122.555180

Jadi, dari data di atas sistem akan melakukan serangkaian pemrosesan yang bertujuan untuk memperbaiki beberapa posisi yang tidak tepat sesuai lokasi kelas. Setelah melalui proses dari metode *spatial map matching* maka akan memperoleh hasil sebagai berikut.

Tabel 5.3 Kordinat yang telah diperbaiki

No.	Nim	Latitude	Longitude
1.	E1E118067	-4.037586	122.555900
2.	E1E119062	-4.037586	122.555900
3.	E1E119033	-4.037586	122.555900
4.	E1E119017	-4.037586	122.555900
5.	E1E119053	-4.037586	122.555900
6.	E1E119049	-4.037586	122.555900
7.	E1E119034	-4.037589	122.555467
8.	E1E119022	-4.037586	122.555900
9.	E1E119076	-4.037586	122.555900
10.	E1E119016	-4.037522	122.555180

Tingkat akurasi perbaikan kordinat mahasiswa saat mengabsen pada pengujian data di atas yaitu sebesar 80 % dari 10 kali percobaan mahasiswa absen di kelas.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai implementasi *Global Navigation Satellite System* (GNSS) pada sistem presensi berbasis *android* menggunakan metode *spatial map matching*, maka diperoleh kesimpulan yaitu:

1. Sistem ini menggunakan 2 absensi yaitu absensi kode dan lokasi yang menyesuaikan dengan kondisi perkuliahan saat ini.
2. Kecepatan proses absen setiap mahasiswa berjalan cukup cepat yaitu dengan kisaran waktu 5 sampai 10 detik saja.
3. Kelebihan algoritma *spatial map matching* yaitu dapat memperbaiki posisi lokasi mahasiswa yang melenceng dari lokasi kelas agar sesuai dengan lokasi kelas yang sebenarnya.

6.2 Saran

Beberapa saran yang perlu diperhatikan untuk pengembangan penelitian selanjutnya yaitu sebagai berikut.

1. Peningkatan akurasi lokasi dapat dilakukan dengan menggunakan algoritma lain yang dianggap lebih akurat dan sesuai dengan perkembangan teknologi.
2. Pada penelitian selanjutnya sistem ini dapat dibuat lebih kompleks lagi seperti adanya sistem tersedia untuk mahasiswa sehingga masing-masing dari mereka dapat melihat atau mengolah data.

DAFTAR PUSTAKA

- C, A. (2020). *Pengertian HTML*. <https://www.hostinger.co.id/tutorial/apa-itu-html/>
- Fanggidae, A., & Polly, Y. T. (2016). *System Presensi Dosen Menggunakan IMEI dan GPS Smartphone dengan Data Terenkripsi*. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 5(3), 1–8.
<https://doi.org/10.22146/jnteti.v5i3.249>
- Ferdianto, Y. (2013). *Pengertian PHP*. <https://www.carawebs.info/2013/04/apa-itu-php.html>
- Khoir, S. A., Yudhana, A., & S, S. (2020). Implementasi GPS (Global Positioning System) Pada Presensi Berbasis *Android* DI BMT Insan Mandiri. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 4(1), 9.
<https://doi.org/10.30645/j-sakti.v4i1.182>
- Maftukhin, M. R. (2018). *IMPLEMENTASI ALGORITMA SPATIAL MAP MATCHING UNTUK MENGETAHUI LOKASI KENDARAAN MELALUI APLIKASI GPS TRACKER SKRIPSI Oleh : MOCHAMMAD RIZAL MAFTUKHIN*.
- Muhtajib, K., Sidqon, M., Si, S., & Si, M. (2017). *Aplikasi Presensi Online Menggunakan IMEI Dan GPS Pada Smartphone Android*. 45.
- Rahardjo, A. (2018). *Pengertian RUP (Rational Unified Process)*.
<https://medium.com/@andrerahardjo/pengertian-rup-rational-unified-process-1bec9c664458>
- Raharjo, B. (2019). *Pemrograman Android dengan Flutter (Pertama)*. Informatika Bandung.
- Sadahiro, Y. (2006). *Course #716-26 Advanced Urban Analysis E. Lecture Title: -*

Spatial Analysis using GIS – Associate professor of the Department of Urban (Engineerin). University of Tokyo.

Sukindar. (2016). *Pengertian Dart*. <https://teknojurnal.com/kelebihan-bahasa-pemrograman-dart/>

Wahyono, Eko Budi; Suhattanto, M. A. (2019). Survey Satelit Pertanahan. *Jurnal Sosial Politik*, 5(2). <https://doi.org/10.22219/sospol.v5i2.10695>

Zayid, F., & Ferdiana, E. (2020). Penerapan Algoritma Spatial Map Matching Dengan API Menggunakan GPS Untuk Posisi Tumpangan Kendaraan. *Teknois : Jurnal Ilmiah Teknologi Informasi Dan Sains*, 10(1), 45–56. <https://doi.org/10.36350/jbs.v10i1.80>

