**Kata Pengantar**

DAFTAR ISI

[Kata Pengantar i](#_Toc499808846)

[DAFTAR ISI iii](#_Toc499808847)

[DAFTAR TABEL vii](#_Toc499808848)

[DAFTAR GAMBAR viii](#_Toc499808849)

[1 BAB I PENDAHULUAN 1](#_Toc499808850)

[1.1. Latar Belakang 1](#_Toc499808851)

[1.2. Rumusan Masalah 3](#_Toc499808852)

[1.3. Tujuan dan Manfaat 3](#_Toc499808853)

[1.4. Batasan dan Asumsi Masalah 3](#_Toc499808854)

[1.5. Sistematika Penulisan 4](#_Toc499808855)

[1.6. Jadwal Pelaksanaan Kerja Praktik 5](#_Toc499808856)

[1.6.1 Tahapan Penelitian 5](#_Toc499808857)

[1.6.2 Waktu Pelaksanaan 6](#_Toc499808858)

[2 BAB II LANDASAN TEORI 7](#_Toc499808859)

[2.1 Konsep Dasar Absensi 7](#_Toc499808862)

[2.2 Konsep Dasar Sistem 7](#_Toc499808863)

[2.2.1 Definisi Dasar Sistem 7](#_Toc499808864)

[2.2.2 Karakteristik Sistem 8](#_Toc499808865)

[2.2.3 Klasifikasi Sistem 9](#_Toc499808866)

[2.3 Konsep Dasar Basis Data 10](#_Toc499808867)

[2.4 Konsep Dasar Object Oriented 11](#_Toc499808868)

[2.4.1 Konsep Object Oriented 11](#_Toc499808869)

[2.4.2 Karakteristik Object Oriented 13](#_Toc499808870)

[2.5 Extreme Programing 14](#_Toc499808871)

[2.5.1 Sejarah Extreme Programming 14](#_Toc499808872)

[2.5.2 Pengertian Extreme Programming 15](#_Toc499808873)

[2.5.3 Kelebihan dan Kelemahan Extreme Programming (XP) 15](#_Toc499808874)

[*2.6* Konsep Dasar *Flowchart* 16](#_Toc499808875)

[*2.6.1* Pendahuluan *Flowchart* 16](#_Toc499808876)

[*2.6.2* Pendoman Penggunaan *Flowchart* 16](#_Toc499808877)

[*2.6.3* Simbol *Flowchart* 17](#_Toc499808878)

[*2.7* *Unifield Modeling Language (UML)* 20](#_Toc499808879)

[*2.7.1* Konsep Dasar *Unifield Modeling Language* 21](#_Toc499808880)

[*2.7.2* Komponen Komponen *UML* 22](#_Toc499808881)

[2.7.3 Keuntungan Dan Kelemahan *UML* 26](#_Toc499808882)

[*2.8* *GPS* 26](#_Toc499808883)

[2.8.1 Akurasi Global Positioning System 27](#_Toc499808884)

[2.8.2 Android dan Global Positioning System 28](#_Toc499808885)

[2.9 Location Base Service 28](#_Toc499808886)

[2.9.1 Komponen Location Based Service (LBS) 29](#_Toc499808887)

[2.9.2 Latitude (Garis Lintang) 29](#_Toc499808888)

[2.9.3 Longitude (Garis Bujur) 30](#_Toc499808889)

[2.10 Android 30](#_Toc499808890)

[2.10.1 Arsitektrur Android 31](#_Toc499808891)

[2.10.2 Komponen Android 33](#_Toc499808892)

[2.11 Peralatan Pendukung 34](#_Toc499808893)

[2.11.1 Java 35](#_Toc499808894)

[2.11.2 MySQL 36](#_Toc499808895)

[2.11.3 Android Studio 36](#_Toc499808896)

[2.11.4 Software Development Kit 37](#_Toc499808897)

[2.11.5 Java Development Kit 38](#_Toc499808898)

[2.11.6 Java Script Object Notation (JSON) 38](#_Toc499808899)

[2.11.7 PHP 39](#_Toc499808900)

[3 BAB III METODOLOGI PENELITIAN 40](#_Toc499808901)

[3.1. Analisis Sistem 40](#_Toc499808902)

[3.1.1. Analisis Kebutuhan Software 40](#_Toc499808903)

[3.1.2. Analisis Kebutuhan Hardware 41](#_Toc499808904)

[3.1.3. Analisis Kebutuhan Pengguna 41](#_Toc499808905)

[3.2. Kerangka Berfikir 42](#_Toc499808906)

[3.3. Objek Penelitian 43](#_Toc499808907)

[3.3.1. Sejarah Singkat SMP Negeri 5 Majalengka 43](#_Toc499808908)

[3.3.2. Struktur Organisasi 44](#_Toc499808909)

[3.3.3. Visi dan Misi SMP Negeri 5 Majalengka 44](#_Toc499808910)

[3.3.4. *Flowchart* Sistem Absensi Berjalan 45](#_Toc499808911)

[3.3.5. *Flowchart* Sistem Absensi Usulan 46](#_Toc499808912)

[*3.4.* *Use-case Diagram* **Error! Bookmark not defined.**](#_Toc499808913)

[*3.4.1.* *Use-case Diagram* **Error! Bookmark not defined.**](#_Toc499808914)

[3.4.2. Definisi Aktor **Error! Bookmark not defined.**](#_Toc499808915)

[3.4.3. Definisi Login **Error! Bookmark not defined.**](#_Toc499808916)

[3.4.4. Definisi Daftar **Error! Bookmark not defined.**](#_Toc499808917)

[3.4.5. Definisi Absen **Error! Bookmark not defined.**](#_Toc499808918)

[3.5. Activity Diagram **Error! Bookmark not defined.**](#_Toc499808919)

[3.6. Class Diagram **Error! Bookmark not defined.**](#_Toc499808920)

[BAB 4 PERANCANGAN 47](#_Toc499808921)

[BAB 5 IMPLEMENTAS 59](#_Toc499808922)

[BAB 6 PENUTUP 60](#_Toc499808923)

[Daftar Pustaka 61](#_Toc499808924)

# DAFTAR TABEL

[Tabel 1.1 Waktu Pelaksanaan 6](#_Toc499808824)

[Tabel 2.1 Simbol-simbol *Flowchart* (Subrata, 2017) 17](#_Toc499808825)

[Tabel 2.2 Simbol-simbol *Flowchart (*lanjutan 1) 18](#_Toc499808826)

[Tabel 2.3 Simbol-simbol *Flowchart* (lanjutan 2) 19](#_Toc499808827)

[Tabel 2.4 Simbol-simbol *Flowchart* (lanjutan 3) 20](#_Toc499808828)

[Tabel 2.5 *Sequence* Diagram **Error! Bookmark not defined.**](#_Toc499808829)

# DAFTAR GAMBAR

[Gambar 2.1 *Extreme Programming* (Proboyekti, 2011) 15](#_Toc499808808)

[Gambar 2.2 Diagram *UML* (sumber:http://uml.org) 21](#_Toc499808809)

[Gambar 2.3 *Use Case* Diagram (Haviluddin, 2011) 23](#_Toc499808810)

[Gambar 2.4 *Activity* Diagram (Haviluddin, 2011) 24](#_Toc499808811)

[Gambar 2.5 *Squence* Diagram (Haviluddin, 2011) 25](#_Toc499808812)

[Gambar 2.6 *Class* Diagram (Haviluddin, 2011) 25](#_Toc499808813)

[Gambar 2.7 Latitude Kota Majalengka 30](#_Toc499808814)

[Gambar 2.8 *Longitude* kota Majalengka 30](#_Toc499808815)

[Gambar 2.9 *Arsitektur Android* (Developer Android, 2017) 32](#_Toc499808816)

[Gambar 2.10 Logo Java 35](#_Toc499808817)

[Gambar 2.11 Logo MySQL (Fathansya, 2012) 36](#_Toc499808818)

[Gambar 2.12 Logo Android Studio 37](#_Toc499808819)

[Gambar 2.13 Software Development Kit 37](#_Toc499808820)

[Gambar 2.14 Java Development Kit 38](#_Toc499808821)

[Gambar 2.15 Logo JSON 38](#_Toc499808822)

[Gambar 3.1 Kerangka Berfikir 42](#_Toc499808823)

# BAB I PENDAHULUAN

**BAB I   
PENDAHULUAN**

## Latar Belakang

Dalam era globalisasi pada saat ini, perkembangan ilmu pengetahuan dan teknologi berkembang sangat pesat sehingga memudahkan kita dalam melakukan aktifitas. Saat ini hampir seluruh aspek kehidupan telah memanfaatkan teknologi sebagai faktor pendukung untuk menyelesaikan dan mempermudah pekerjaan disemua bidang seperti bidang pendidikan, rumah tangga, dan industri. Kehadiran teknologi tersebut dimaksudkan untuk mencapai hasil yang lebih efektif dan efisien pada sebuah proses di perusahaan. Salah satunya yaitu perkembangan teknologi dalam mempermudah proses absensi di suatu instansi.

Berdasarkan kamus Bahasa Indonesia, absen adalah tidak bekerjanya seorang pegawai pada saat hari kerja karena sakit, izin, alpa atau cuti. Absensi adalah daftar administrasi kehadiran atau ketidakhadiran pegawai. Dimana pegawai yang tidak hadir akan tercatat di daftar abensi kepegawaian dan kapan saja bisa dicek. Kadang kala absensi sering terabakaikan oleh kita tetapi sangat berpengaruh untuk melihat kinerja karyawan dilapangan.

SMPN 5 Majalengka merupakan salah satu sekolah menengah pertama negeri yang ada di kecamatan Majalengka walaupun tempatnya tidak berada dipusat kota Majalengka SMPN 5 Majalengka ini merupakan salah satu sekolah menengah yang berprestasi, tetapi di SMPN 5 Majalengka masih ada masalah kurang efektif dengan sistem absensi kepegawaiannya.

Absensi yang sekarang berjalan di SMPN 5 Majalengka masih menggunakan absensi manual dengan menggunakan tulis tangan, jadi total 36 orang pegawai dan tenaga pengajar yang ada di SMPN 5 Majalengka masih menggunakan absensi manual. Dengan absensi manual ini kurang begitu efisien dalam proses pengecekan apakah karyawan tersebut datang tepat waktu atau terlambat dalam melakukan absensi. Bisa saja dengan absensi di manifulasi oleh sebagian orang. Dan kepala sekolah atau bagian tata usaha juga kesulitan mendapatkan informasi absensi kepegawaian secara cepat.

Maka oleh karena itu dibutuhkan sebuah sistem yang dapat memonitor secara langsung aktifitas absensi kepegawaian secara realtime. agar memudahkan kepala sekolah dan bagian tata usaha dalam mengelola data kepegawaian. Yaitu dengan menggunakan fingerprint untuk sistem absensi tetapi untuk menggunakan sistem absensi menggunakan *fingerprint* atau sidik jari pihak sekolah harus mengadakan alat *fingerprint* yang cukup mahal oleh karena itu disarankan menggunakan opsi lain yaitu dengan sistem absensi menggunakan *smartphone* yang relatif terjangkau karena setiap pegawai yang ada di lingkungan sekolah sudah menggunakan *smartphone* dan *smartphone* pengawai tersebut digunakan untuk melakukan absensi jadi pihak sekolah tidak perlu mengeluarkan biaya untuk mengadak alat untuk sistem absensi.

Dengan menggunakan *smartphone* hampir semua pegawai sudah menggunakannya untuk kehidupan sehari-hari. Ketika penulis melakukan wawancara dengan pegawai yang ada dilingkungan SMPN 5 Majalengka. Hampir semua pegawai yang ada dilingkungan SMPN 5 Majalengka sudah menggunakna *smartphone*.

*Smartphone* adalah perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengan telepon genggam konvensional tetapi sudah memiliki berbagai teknologi yang sudah canggih. Oleh Karena itu *smartphone* dapat digunakan untuk sistem absensi kepegawain yang ada di SMPN 5 Majalengka sebagai pengganti sistem absensi konvensional. Hampir semua *smartphone* yang beredar saat ini sudah memiliki fitur canggih *Global Positiong System* atau bisa disebut juga dengan *GPS*. *GPS* merupakan sistem navigasi dengan bantuan satelit yang berfungsi untuk menentukan posisi, arah kecepatan.

Dengan tekologi *GPS* ini maka sangat membantu dalam memastikan pegawai itu melakukan absensi disekitar sekolah maka digunakanlah teknlogi yang ada pada *smartphone* yaitu teknologi *Global Positioning System (GPS).* Dengan menggunakan *GPS* di dapat sebuah longitude dan lantitude yang menandakan lokasi dari *smartphone* yang digunakan absensi oleh pegawai.

Dilihat dari permasahalah yang ada di SMPN 5 Majalengka maka peneliti melihat ada peluang untuk merancang suatu sistem yang dapat memonitoring proses absensi yang ada di sekolah tersebut. Oleh Karena itu dalam penelitian ini penulis mengambil judul **“PERANCANGAN ABSENSI KEPEGAWAIAN DIGITAL MENGGUNAKAN TEKNOLOGI GLOBAL POSITIONING SYSTEM PADA SMARTPHONE”.** Diharapkan dengan adanya sistem ini dapat membantu pihak sekolah dalam mengelola absensi yang lebih baik lagi.

## Rumusan Masalah

Berdasarkan latar belakang diatas dapat disimpulkan suatu rumusan masalah sebagai berikut:

1. Bagaimana membuat aplikasi absensi pegawai yang bisa digunakan di smartphone berbasis GPS?
2. Bagaimana membuat laporan absensi berbasis web?

## Tujuan dan Manfaat

Tujuan dari penelitian ini yaitu sebagai berikut:

1. Membuat aplikasi absensi pegawai yang bisa digunakan di smartphone berbasis GPS;
2. Membuat laporan absensi berbasis web.

Manfaat dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Membantu bagian Tata Usaha dalam mengelola absensi pegawai;
2. Mempermudah proses monitoring *realtime* oleh kepala sekolah.

## Batasan dan Asumsi Masalah

Dalam pembuatan laporan penelitian ini diperlukan batasan masalah, agar permasalahan yang ditinjau tidak terlalu luas dan sesuia dengan tujuan yang dicapai. Adapun batasan-batasannya adalah sebagai berikut:

1. Dalam hal ini hanya akan dibahas mengenai konsep perancangan sistem absensi menggunakan *smartphone;*
2. Sistem yang akan dibangun hanya melakukan sistem absensi sedangkan proses lain seperti *input output* nilai, sistem penggajian pegawai dan lainnya tidak akan diatur;
3. Semua pegawai diasumsikan sudah menggunakan *smartphone android;*
4. Semua pegawai diasumsikan melakukan absensi dengan *smartphone* masing-masing berdasarkan imei dan no *handphone* sebagai id;
5. Diasumsikan *handphone* pegawai selalu menyala ketika berada di lingkungan sekolah ketika jam kerja;
6. Koordinat sekolah diasumsikan sebagai area untuk tempat melakukan absensi selain diarea tersebut tidak bisa dilakukan;
7. Untuk akurasi dari GPS tidak diperhitungakan atau diabaikan.

## Sistematika Penulisan

Untuk sistematika penulisan adalah sebagai berikut:

**BAB I : PENDAHULUAN**

Bab ini menguraikan latar belakang permasalahan, identifikasi masalah, rumusan masalah, batasan masalah, manfaat penelitian, metode penelitian, *time schedule*  dan sistematika penulisannya.

**BAB II : LANDASAN TEORI**

Bab ini menjelaskan tentang pengertian sistem, informasi dan definisi sistem informasi beserta komponen-komponennya, perangkat lunak yang digunakan dalam pembuatan aplikasi, serta teori-teori yang berhubungan dengan masalah yang akan dibahas.

**BAB III : METODE PENELITIAN**

Pembahasan menguraikan tentang kerangka kerja penelitian dan analisis permasalahan-permasalahan yang ada dan bagaimana pemecahan dari permasalahan tersebut.

**BAB IV : PERANCANGAN SISTEM**

Pembahasan menguraikan tentang perencanaan pembuatan keseluruhan sistem dalam aplikasi yang akan dibuat.

**BAB V : IMPLEMENTASI SISTEM**

Menguraikan tentang kebutuhan implementasi dari aplikasi, serta output yang dihasilkan dari aplikasi yang dibuat.

**BAB VI : PENUTUP**

Bab ini berisi kesimpulan tentang seluruh pembahasan dan pemecahan masalah yang telah dilakukan serta mengenai hal-hal yang terdapat dalam bab sebelumnya dan saran-saran untuk mengembangkan aplikasi berikutnya.

## Jadwal Pelaksanaan Kerja Praktik

Pada jadwal pelaksanaan kerja praktik ini dibagi menjadi dua bagian:

### Tahapan Penelitian

Pada tahapan penelitian ini dibagi menjadi dua bagian yaitu tahapan pengumpulan data dan tahapan pengembangan sistem.

1. Metodologi Pengumpulan Data

Wawancara (*Interview*)

Wawancara(*Interview*) yaitu pengumpulan data dengan mengajukan pertanyaan-pertanyaan dan tanya jawab kepada seluruh pegawai.

Observasi

Observasi adalah metode pengumpulan data melalui pengamatan langsung atau peninjauan secara cermat dan langsung dilapangan atau lokasi kerja praktik.

1. Metodologi Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *Extreme Programing* (XP). *Extreme Programing* (XP) merupakan suatu pendekatan yang paling banyak digunakan untuk pengembangan perangkat lunak cepat. Alasan menggunakan metode *Extreme Programing* (XP) karena sifat dari aplikasi yang dikembangkan dengan cepat melalui tahapan-tahapan yang ada meliputi: *Planning*/Perencanaan, *Design*/Perancangan, *Coding*/Pengkodean dan *Testing*/Pengujian. (Proboyekti, 2011)

Adapaun tahapan *Extreme Programing* dapat dijelaskan sebagai berikut:

Metode penelitian yang digunakan dalam pengembangan sistem ini adalah metode *Extreme Programming (Xp)* yang memiliki beberapa mekanisme sebagai berikut :

1. *Requirtment*

Pada tahap *Requirtment* ditentukan latar belakang masalah, tujuan penelitian, serta rumusan dan batasan masalah;

1. Analisis Sistem

Menganalisis kebutuhan sistem berdasarkan batasan masalah dan ketersediaan data. Dimana analisis sistem ini akan menjadi acuan dalam tahap *design;*

1. *Design*

Pada tahap ini, penulis melakukan pemodelan sistem yang akan dibuat. Mulai dari dari perancangan *user interface* yang akan digunakan;

1. Coding

Pada tahap ini, penulis melakuan pemograman aplikasi yang akan digunakan;

1. Testing

pada tahap ini, penulis melakukan uji coba pada aplikasi yang sudah dibuat oleh penulis apakah terjadi *error* atau tidak pada aplikasi yang telah dibuat oleh penulis. (Proboyekti, 2011)

### Waktu Pelaksanaan

Waktu pelaksanaan kerja praktik yang penulis lakukan di UPTD SMPN 5 Majalengka adalah sebagai berikut:

Tabel 1.1 Waktu Pelaksanaan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Kegiatan | September | | | | Oktober | | | | November | | | | Desember | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | *Requirtment* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Analisis |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Coding |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Seminar |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# BAB II LANDASAN TEORI

**BAB II  
LANDASAN TEORI**

Untuk mendukung penyusunan laporan ini, maka dibutuhkan sebuah teori-teori yang mendukung dengan permasalahan dan ruang lingkup pembahasan akan dibahas pada bab ini sebagai landasan teori.



## Konsep Dasar Absensi

Berdasarkan kamus Bahasa Indonesia, absen adalah tidak bekerjanya seorang pegawai pada saat hari kerja, karena sakit, izin, alpa atau cuti. Absensi adalah daftar administrasi kehadiran atau ketidakhadiran pegawai. Dimana pegawai yang tidak hadir akan tercatat di daftar abensi kepegawaian dan kapan saja bisa dicek. Kadang kala absensi sering terabakaikan oleh kita tetapi sangat berpengaruh untuk melihat kinerja karyawan dilapangan.

## Konsep Dasar Sistem

Dibahwah ini adalah penjelasan dari sistem adalah sebagai berikut:

### Definisi Dasar Sistem

Pada dasarnya Sistem adalah suatu kerangka dari prosedur – prosedur yang saling berhubungan, yang disusun sesuai dengan skema yang menyeluruh untuk melaksanakan suatu kegiatan. Sedangkan hal paling sederhana dari sistem itu sendiri mempunyai masukan dan keluaran, selain itu sebuah sistem mempunyai karakteristik atau sifat – sifat tertentu yang mencirikan bahwa hal tersebut bias dikatakan sebagai suatu sistem .

Sedangkan ada banyak pendapat dari sistem itu menurut para ahli. Berikut pengertian sistem menurut para ahli:

1. Sistem Informasi Akutansi : Sistem adalah kumpulan / group dari sub sistem/bagian/komponen apapun baik pisik maupun non pisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu; (Susanto, 2013)
2. Sistem adalah kmuplan elemen yang saling berinteraksi dalam suatu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama; (Sutarman, 2009)
3. Sistem dapat di definisikan dengan pendekatan prosedurdan dengan pendekatan komponen; (Jogiyanto, 2010)
4. Sistem adalah jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan-kegiatan pokok perusahaan. (Mulyadi, 2010)

### Karakteristik Sistem

Secara umum sebuah sistem adalah ada input, proses dan output. Sedangkan karakteristik sistem itu sendiri terbagi menjadi beberapa bagian diantaranya:

1. Komponen Sistem

Komponen – Komponen sistem tersebut dapat berupa bentuk subsistem, setip subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan;

1. Batasan Sistem (Boundary)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem denga sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukan ruang lingkup (scope) dari sistem tersebut;

1. Lingkungan Luar Sistem (Environment)

Bentuk apapun yang ada diluar runga lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut;

1. Penghubung (Interface)

Media yang menghubungkan sistem dengan subsitem yang lain disebut penghubung sistem atau Interface. Penghubung ini memungkinkan sumber – sumber dadya mengalir dari suatu subsistem ke subsistem yang lain. Bentuk keluaran dari subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrase sistem yang membentuk suatu kesatuan;

1. Masukan Sistem (Input)

Merupakan energi yang dimaksukan kedalam sistem. Masukna dapat berupa masukan perawatan (maintenance input) dan masukan sinya (signal input). Maintenance input adalah energi yang dimasukan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah maintenance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah mejadi informasi;

1. Keluaran Sistem (Output)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini dapat menjadi masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputuasan atau hal – hal lain yang mejadi input bagi subsistem lain;

1. Pengelola Sistem (Process)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran yang diingikan ;

1. Sasaran (Objective) dan Tujuan (Goal)

Kalau sistem tidak mempunya sasaran, maka operasi sistem tersebut tidak akan ada gunanya.

### Klasifikasi Sistem

Suatu sistem dapat diklasifikasikan:

1. Sistem Abstrak

Sistem yang berupa pemikiran atau ide – ide yang tidak tampak secara fisik ;

1. Sistem fisik

Merupakan suatu sistem yang ada secara fisik (sistem komuter, sistem akutansi, sistem produksi dll);

1. Sistem alamiah

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadi siang dan malam, dan pergantian musim;

1. Sistem Buatan Manusia

Sistem buatan manusia adalah merupakan sistem yang melibatakan hubungan manusia dengan mesin, yang disebut dengan human machine system. Sistem informasi berbasis komputer merupakan contohnya, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia;

1. Sistem Tertentu

Sistem tertentu adalah sistem yang tingkah lakunya dapat ditentukan/ diperkirakan sebelumnya.

1. Sistem Tak Tentu

Sistem tak tentu adalah sistem tingkah lakunya tidak dapat ditentukan sebelumnya. Contohnya adalah sistem aplikasi komputer merupakan contoh sistem yang tingakh lakunya dapat ditentukan sebelumnya.

1. Sistem Tertutup

Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar benar tertutup, yang ada hanyalah realively closed system ( secara relative tertutup, tidak benar-benar tertutup);

1. Sistem Terbuka

Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. (Sutabri, 2012)

## Konsep Dasar Basis Data

Basis data merupakan media dimana suatu informasi disimpan, seperti sebuah filling kabinet yang menyimpan arsip – arsip surat pada suatu perusahaan. Konsep basis data dalam organisasi dianggap sebagai sumber daya dasar yang penting bagi suatu organisasi.

Menurut Stephens, (2003) basis data adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basisdata, pengguna dapat menyimpaan data secara terorganisir.

Menurut “Siberschatz, dkk.;(2002) mendefinisikan basisdata sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan. Sistem managemen basisdata (DBMS) adalah kumpulan data yang saling berhubungan dan kumpulan program untuk mengakses data. Tujuan untam sistem managemen basisdata adalah menyediakan cara menyimpan dan mengambil informasi basisdata secara mudah dan efisien”. (Informatika, 2017)

Menurut Ramakrishnan & Gehrke, (2003), basis data sebagai kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan. (Informatika, 2017)

## Konsep Dasar Object Oriented

Pemograman berorientasi objek (*object oriented programming*) disingkat menjadi OOP merupakan paradigm pemrograman yang berorientasi kepada objek. Semua data dan fungsi didalam paradigm ini dibungkus dalam kelas-kelas atau objek-objek. Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dan abstraksi objek, kelas, hubungan antarkelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dibungkus) menjadi satu kesatuan.

### Konsep Object Oriented

Konsep Object Oriented adalah sebagai berikut:

1. Kelas (*Class*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dan kelas yang lainnya, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru;

1. Objek (*Object*)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan;

1. Metode (*Method*)

Operasi atau metode atau *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek;

1. Atribut (*Attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya. Atribut sebaiknya bersifat privat untuk menjaga konsep enkapsulasi;

1. Abstraksi (*Abstraction*)

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan;

1. Enkapsulasi (*Encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya;

1. Pewarisan (*Inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya;

1. Antarmuka (*Interface*)

Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain. Antarmuka atau *interface* biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas, mengakses antarmukanya;

1. *Reusabilily*

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut;

1. Generalisasi dan Spesialisasi

Menunjukan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus;

1. Komunikasi Antar Objek

Komunikasi antar-objek dilakukan lewat pesan (*message*) yang dikirim dan satu objek ke objek lainnya;

1. Polimorfisme (*Polimorphism*)

Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program;

1. *Package*

*Package* adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

### Karakteristik Object Oriented

1. Semua adalah objek;
2. Sebuah pesan merupakan permintaan atas sekumpulan aksi dengan semua argument yang diperlukan untuk menyelesaikan suatu tugas tertentu;
3. Komputasi dilakukan dengan komunikasi antar objek. Setiap objek berkomunikasi dengan objek yang lain melalui pengiriman dan penerimaan pesan;
4. Setiap objek memiliki memori sendiri, yang dapat terdiri dari objek-objek lainnya;
5. Setiap objek adalah wakil atau representasi dari suatu kelas sebuah kelas dapat mewakili sekelompok objek yang sama;
6. Kelas merupakan kumpulan tingkah laku yang berkaitan dengan suatu objek. Jadi, semua objek yang merupakan wakil dari kelas yang sama dapat melakukan aksi yang sama pula;
7. Kelas-kelas diorganisir ke dalam struktur pohon yang berakar tunggal, yang dinamakan dengan jenjang pewarisan (inheritance hierarchy);
8. Setiap objek pada umumnya memiliki tiga sifat, yaitu keadaan, operasi dan identitas objek;
9. Operasi merupakan tindakan yang dapat dilakukan oleh sebuah objek;
10. Keadaan objek merupakan koleksi dari seluruh informasi yang dimiliki oleh objek pada suatu saat;
11. Informasi yang terkandung pada objek tersebut pada akhirnya memberikan identitas khusus yang membedakan suatu objek dengan objek lainya.

## Extreme Programing

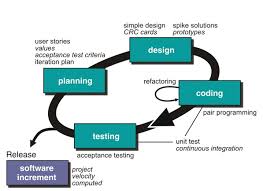
*Extreme programming* merupakan bagian dari proses meodel *Agile Method. Agile* berarti bersifat cepat, ringan, bebas bergerak, waspada. Kata lain ini digunakan sebagai kata yang menggambar konsep model proses yang berbeda dari konsep model-model proses yang sudah ada. Konsep *agile Software Development* dicetuskan oleh Kent Beck dan 16 rekannya dengan menyatakan bahwa *Agile Software Development* adalah cara membangun *software* dengan melakukannya dan membantu orang lain membangunnya sekaligus.

Dalam *Agile Software Development* interaksi dan personil lebih penting dari pada proses dan alat, *software* yang berfungsi lebih penting dari pada dokumentasi yang lengkap, kolaborasi dnegan klien lebih penting dari pada negosiasi kontrak, dan sikap tanggap terhadap perubahan perubahan lebih penting dari pada mengikuti rencana. Namun demikian, sama seperti model proses yang lain, *Agile Software Development* memiliki kelebihan dan tidak cocok untuk semua jenis proyek, produk, orang dan situasi. *Agile Software Development* memungkinkan model proses yang toleransi terhadap perubahan kebutuhan sehingga perubahan dapat dengan cepat ditanggapi. Namun disisi lain menyebakan produktifitas menurun. (Proboyekti, 2011)

### Sejarah Extreme Programming

Proyek *pemrograman Extreme* pertama dimulai pada tanggal 6 maret 1996. *Extreme Programming* adalah salah satu dari beberapa proses *Agile popular.* Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *Extreme Programming (XP)* merupakan suatu pendekatan yang paling banyak digunakan untuk pengembangan perangkat lunak cepat. Alasan menggunakan metode *Exreme Programming (XP)* karena sifat dari aplikasi yang dikembangkan dengan cepat melalui tahapan-tahapan yang ada meliputi: Planning/perencanaan, Design/Perancangan, Coding/Pengkodean, dan Testing/Pengujian. (Proboyekti, 2011)

Adapun tahapan dari *Exteme programming (XP)* adalah sebagai berikut:



Gambar 2.1 *Extreme Programming* (Proboyekti, 2011)

### Pengertian Extreme Programming

1. Planning/Perencanaan

Pada tahap perencanaa ini dimulai dari pengumpulan kebutuhan yang membantu tim teknikal untuk memahami kontek bisnis dari sebuah aplikasi. Selain itu pada tahap ini juga mendefinisikan *output* yang akan dihasilkan, fitur yang dimiliki oleh aplikasi dan fungsi dari aplikasi yang akan dikembangkan;

1. Design/Perancangan

Metode ini menekankan desain aplikasi yang sederhana, untuk mendesain aplikasi dapat menggunakan *Class-Responbility-Collaborator(CRC) card* yang mengidentifikasikan dan mengatur kelas pada *object* *oriented*;

1. Coding/Pengkodean

Konsep utama dari tahap pengkodean pada *extreme programming* adalah *pair programming,* melibatkan lebih dari satu orang untuk menyusun kode;

1. Testing/Pengujian

Pada tahapan ini lebih fokus pada pengujian fitur dan fungsionalitas dari aplikasi. (Proboyekti, 2011)

### Kelebihan dan Kelemahan Extreme Programming (XP)

1. Kelebihan *Extreme Programming (XP)*
2. Menjalin komunikasi yang baik dengan klien. (*Planing phase);*
3. Menurunkan biaya pengembangan (*Implementationi phase);*
4. Meningkatkan komunikasi dan sifat saling menghargai antar developer (*Implementation phase);*
5. *XP* merupakan metode yang semi formal (*Planning phase)*
6. Kelemahan *Extreme Programming (XP)*

Tidak dapat membuat kode yang detail diawal (prinsip *simplicity* dan juga anjuran untuk melakukan yang perlu dilakukan dihari itu juga). Selain dari keunggulan dan kelemahan XP yang telah disebutkan diatas, XP juga memiliki keunggulan yang sekaligus menjadi kelemahannya, yaitu XP tidak memiliki dokumentasi formal yang dibuat selama pengembangan. Satu-satunya dokumentasi adalah dokumentasi awal yang dilakukan oleh user.

## Konsep Dasar *Flowchart*

*Flowchart* adalah gambaran dalam bentuk diagram alir dari algoritmat-algoritma dalam suatu program, yang menyatakan arah alur program tersebut. (Subrata, 2017)

### Pendahuluan *Flowchart*

Diagram arus sistem (Sistem *Flowchart)* adalah peralatan yang digunakan untuk menggambarkan proses sistem secara rinci untuk menggambarkan aliran sistem informasi dan diagram arus sistem untuk menggambarkan aliran program. (Subrata, 2017)

Berdasarkan pemaparan diatas, maka diagram arus sistem (Sistem *Flowchart)* adalah bentuk (model) yang berupa simbol simbol untuk menggambarkan alur sistem agar sistem tersebut mudah dimengerti.

### Pendoman Penggunaan *Flowchart*

Bila seorang analis dan *programmer* akan membuat *Flowchart,* ada beberapa petunjuk yang harus diperhatikan, seperti:

1. *Flowchart* digambarakan dari halaman atas ke bawah dan dari kiri ke kanan;
2. Aktifitas yang digambarkan harus didefinisikan secara hati-hati dan definisi itu harus dapat dimengetti oleh pembacanya;
3. Kapan aktifitas di mulai dan berakhir harus ditentukan secara jelas;
4. Setip langkah dari aktifitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan menghitung pajak penjualan;
5. Setiap langkah dari aktifitas harus berada pada urutan yang benar;
6. Lingkup dan *range* dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktifitas yang sedang digambarkan tidak perlu digambarkan pada *Flowchart* yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berhubungan dengan sistem;
7. Gunakan simbol-simbol *Flowchart* yang standar. (Subrata, 2017)

### Simbol *Flowchart*

Simbol-simbol *Flowchart* yang digunakan biasanya adalah simbol-simbol *Flowchart* standar yang dikeluarkan oleh ANSI dan ISO.

Tabel 2.1 dibawah ini merupakan simbol-simbol dari *Flowchart:*

Tabel 2.1 Simbol-simbol *Flowchart* (Subrata, 2017)

|  |  |
| --- | --- |
| Simbol | Arti |
| Input / Output | Merpresentasikan Input data tau Output data yang diproses atau Informasi |
| Proses  ***Pengertian Flowchart dan Contoh Simbolnya*** | Merepresentasikan operasi |
| Penghubung | Keluar ke atau masuk dati bagian lain flowchart |
| Anak Panah | Merpresentasikan alur kerja |
| Penjelasan | Digunakan untuk tambahan |
| *Symbol Manual Operation*  ***Pengertian Flowchart dan Contoh Simbolnya*** | Simbol yang menunjukkan pengolahan yang tidak dilakukanoleh komputer |

Tabel 2.2 Simbol-simbol *Flowchart (*lanjutan 1)

|  |  |
| --- | --- |
| Simbol | Arti |
| Input / Output | Merpresentasikan Input data tau Output data yang diproses atau Informasi |
| Proses  ***Pengertian Flowchart dan Contoh Simbolnya*** | Merepresentasikan operasi |
| Penghubung | Keluar ke atau masuk dati bagian lain flowchart |
| Anak Panah | Merpresentasikan alur kerja |
| Penjelasan | Digunakan untuk tambahan |
| *Symbol Manual Operation*  ***Pengertian Flowchart dan Contoh Simbolnya*** | Simbol yang menunjukkan pengolahan yang tidak dilakukanoleh komputer |
| *Symbol Decision*  ***Pengertian Flowchart dan Contoh Simbolnya*** | *Symbol Decision* (Simbol untuk kondisi yang akan menghasilkan beberapa   kemungkinan jawaban/aksi) |

Tabel 2.3 Simbol-simbol *Flowchart* (lanjutan 2)

|  |  |
| --- | --- |
| Simbol | Arti |
| *Symbol Predefined Process*  *Pengertian Flowchart dan Contoh Simbolnya* | *Symbol Predefined Process* (Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage) |
| *Symbol Terminal*  *Pengertian Flowchart dan Contoh Simbolnya* | Symbol Terminal (Simbol untuk permulaan atau akhir dari suatu program) |
| Symbol Off-line Storage  *Pengertian Flowchart dan Contoh Simbolnya* | Symbol Off-line Storage (Simbol yang menunjukkan bahwa data di dalam symbol ini akan disimpan) |
| *Symbol Manual Input*  *Pengertian Flowchart dan Contoh Simbolnya* | Symbol Manual Input (Simbol untuk pemasukan data secara manual on-line keyboard) |
| *Symbol Keying Operation*  *Pengertian Flowchart dan Contoh Simbolnya* | Symbol Keying Operation (Simbol operasi dengan menggunakan mesin yang mempunyai keyboard) |
| *Symbol magnetic-tape unit*  Pengertian Flowchart dan Contoh Simbolnya | Symbol magnetic-tape unit (Symbol yang menyatakan input berasal pita magnetic atau output disimpan ke pita magnetic) |

Tabel 2.4 Simbol-simbol *Flowchart* (lanjutan 3)

|  |  |
| --- | --- |
| Simbol | Arti |
| *Symbol punched card*  Pengertian Flowchart dan Contoh Simbolnya | *Symbol punched card* (Symbol yang menyatakan input berasal dari kartu atau output ditulis ke kartu) |
| *Symbol disk*  Pengertian Flowchart dan Contoh Simbolnya | Symbol disk and on-line storage (Symbol untuk menyatakan input berasal dari disk atau output disimpan ke disk) |
| *Symbol display*  Pengertian Flowchart dan Contoh Simbolnya | Symbol display (Symbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer, dan sebagainya) |
| *Symbol dokumen*  Pengertian Flowchart dan Contoh Simbolnya | *Symbol dokumen* (symbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas) |

## *Unifield Modeling Language (UML)*

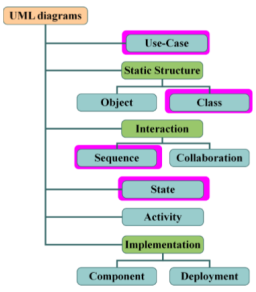
UML (Unified Modeling Language) adalah sebuah bahasa yang  
berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan,  
membangun, dan pendokumentasian dari sebuah sistem pengembangan  
software berbasis OO (Object-Oriented). UML sendiri juga memberikan  
standar penulisan sebuah sistem blueprint, yang meliputi konsep bisnis  
proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema  
database, dan komponenkomponen yang diperlukan dalam sistem  
software. (Haviluddin, 2011)

### Konsep Dasar *Unifield Modeling Language*

UML adalah salah satu tool/ model untuk merancang pengembangan  
software yang berbasis Object Oriented. UML sebagai sebuah bahasa yang  
memberikan vocabulary dan tatanan penulisan kata-kata dalam ‘MS  
Word’ untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah  
bahasa yang mempunyai vocabulary dan konsep tatanan / aturan penulisan  
serta secara fisik mempresentasikan dari sebuah sistem. UML adalah  
sebuah bahasa standard untuk pengembangan sebuah software yang dapat  
menyampaikan bagaimana membuat dan membentuk model-model, tetapi  
tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang  
merupakan salah satu proses implementasi pengembangan software.

Secara filosofi *UML*  diilhami oleh konsep yang telah ada yaitu konsep pemodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh objek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. (Haviluddin, 2011)

Berikut gambar 2.2 tentang diagram *UML*



Gambar 2.2 Diagram *UML* (sumber:http://uml.org)

Berikut tujuan utama dalam desain *UML* adalah:

1. Menyediakan bagi penguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekpresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna;
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep ini;
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunan maka *UML* bersifat independen terhadap bahasa pemograman tertentu;
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan;
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorietasi objek;
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem;
7. Memiliki interaksi praktik terbaik. (Haviluddin, 2011)

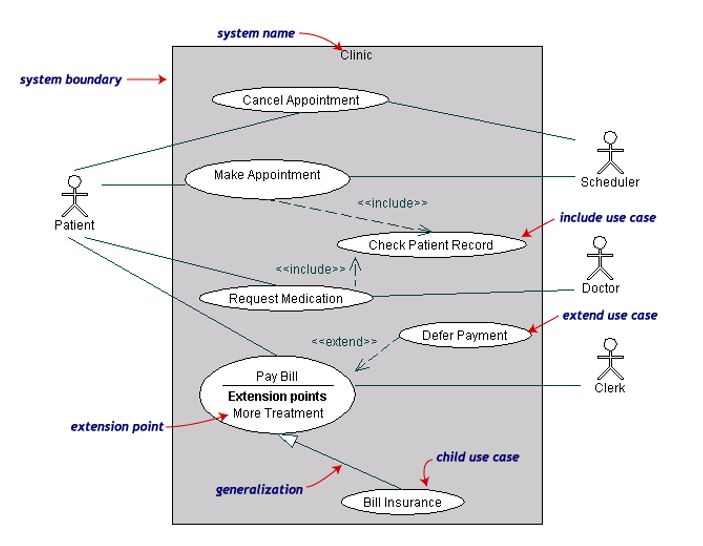
### Komponen Komponen *UML*

Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena *UML* memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik.

*UML* tidak hanya merupakan sebuah bahasa pemograman visual saja,  
namun juga dapat secara langsung dihubungkan ke berbagai bahasa  
pemograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan  
secara langsung ke dalam sebuah *object-oriented database*. Begitu juga  
mengenai pendokumentasian dapat dilakukan seperti; *requirements,  
arsitektur, design, source code, project plan, tests, dan prototype*. Untuk  
dapat memahami UML membutuhkan bentuk konsep dari sebuah Bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML seperti *building  
block*, aturan-aturan yang menyatakan bagaimana *building block*diletakkan secara bersamaan, dan beberapa mekanisme umum (*common*).

UML terdiri atas beberapa diagram, yaitu :

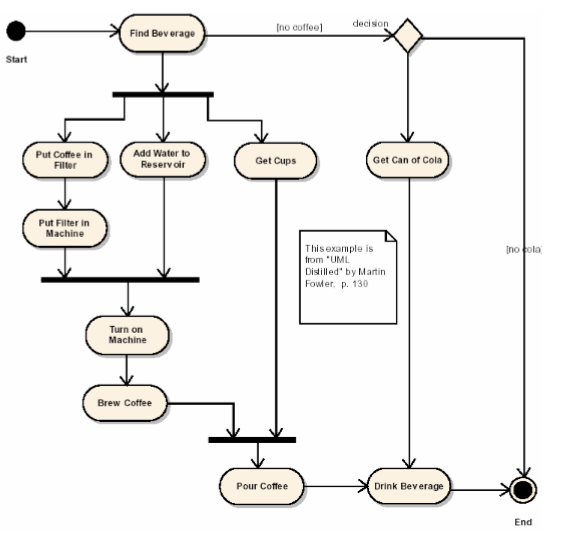
1. *Use Case* Diagram



Gambar 2.3 *Use Case* Diagram (Haviluddin, 2011)

Yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case* diagram juga dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.

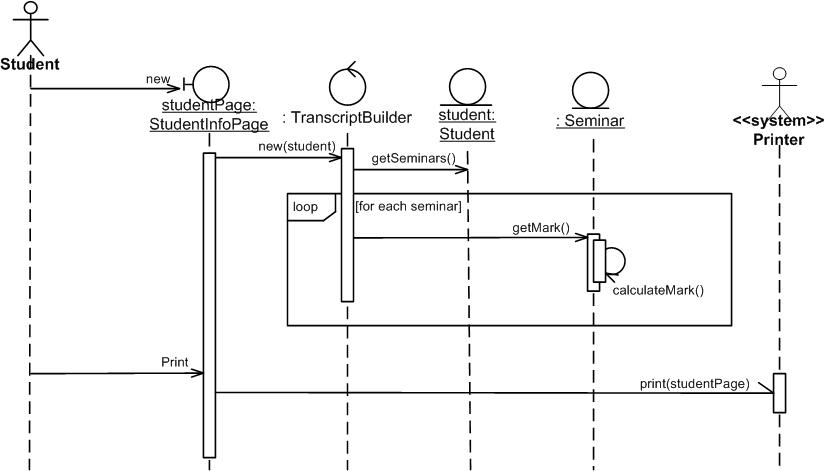
1. *Activity* Diagram / Diagram aktifitas



Gambar 2.4 *Activity* Diagram (Haviluddin, 2011)

Class Diagram, merupakan sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

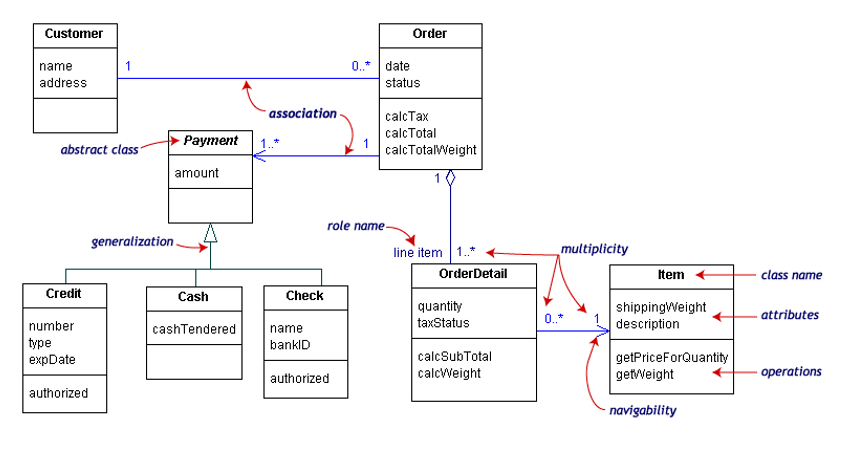
1. *Sequence* Diagram



Gambar 2.5 *Squence* Diagram (Haviluddin, 2011)

Yaitu salah satu jenis diagram pada UML yang menjelaskan  
interaksi objek yang berdasarkan urutan waktu, *sequence* diagram juga  
dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk  
dapat menghasilkan sesuatu seperti pada *use case* diagram.

1. *Class* Diagram



Gambar 2.6 *Class* Diagram (Haviluddin, 2011)

Yaitu salah satu jenis diagram pada UML yang digunakan  
untuk menampilkan kelas-kelas maupun paket-paket yang ada pada  
suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat  
memberikan sebuah gambaran mengenai sistem maupun relasi-relasi  
yang terdapat pada *system* tersebut. (Haviluddin, 2011)

### Keuntungan Dan Kelemahan *UML*

1. Keuntungan

Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu. Taylor (1992) menyatakan bahwa membangun software menggunakan pendekatan teknologi objek memberikan beberapa keuntungan, antara lain: memungkinkan penggunaan kembali objek yang ada (reusable), memungkinkan software yang baru dengan konstruksi yang lebih besar, software berorientasi objek secara umum lebih mudah dimodifikasi dan dirawat karena sebuah objek dapat dimodifikasi tanpa banyak berpengaruh pada objek yang lain. (Haviluddin, 2011)

1. Kelemahan

UML dipandang masih mempunyai kekurangan terutama dalam meng-generate kode program secara komplit. Hal ini karena kurangnya cara memodelkan aspek kelakuan internal perangkat lunak untuk dipetakan ke dalam kode program. Seperti yang kita ketahui, diagram UML yang dapat menghasilkan kode hanyalah diagram class, namun itupun hanya sebatas kerangka kodenya saja dan tidak bisa meng-generate badan program-nya.

## *GPS*

*Global Positioning System(GPS)* merupakan sistem navigasi atau penentu posisi berbasis satelit. Sistem ini mengirim sinyal gelombang mikro ke bumi dengan menggunakan 24 satelit. Sinyal yang dikirimkan oleh satelit akan diterima oleh GPS *receiver.* Contohnya sebagai berikut:

1. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
2. Lokasi. GPS memberikan informasi lokasi dalam tiga dimensi.
3. Latitude
4. Longitude
5. Elevasi
6. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukan informasi kecepatan berpindah tersebut.
7. Arah perjalanan. GPS dapat menunjukan arah tujuan.
8. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingi dikunjungi bisa disimpan oleh GPS *receiver.*
9. Komulasi data. GPS *receiver* dapat menyimpan informasi *track,* seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan dan sebagainya.

GPS terdiri dari tiga bagian yaitu sistem kontrol, satelit dan pengguna. Sistem kontrol merupakan bagian yang mengontrol pergerakan satelit-satelit yang ada dan saling beriteraksi satu sama lain, kemudian pengguna adalah alat navigasi yang digunakan seperti perangkan *mobile* yang kini sudah memiliki fitu GPS didalamnya. GPS biaanya digunakan untuk menunjukan suatu lokasi yang berbeda dipermukaan bumi dengan tingkat akurasi yang cukup baik yaitu kurang dari 10 meter selama tidak ada benda padat yang dapat menghambat sinyal untuk mendapatkan lokasi pengguna

### Akurasi Global Positioning System

Posisi yang ditunjukan oleh suatu *GPS* mempunyai factor kesalahan atau juga disebut tingkatan akurasi. Sebagai contoh suatu alat *GPS* menunjukan titik koordinat dengan tingkat akurasi 5 meter, itu berarti posisi pengguna bisa berada dalam *range* radius 5 meter dari titik yang ditunjukan tersebut.

Ada beberapa hal yang mempengaruhi tingkat akurasi tersebut, antara lain.

1. Kesalahan *Ephemeris.* Terjadi jika satelit tidak dapat mentrasmisikan posisi deorbit dengan tepat;
2. Keadaan *Ionosphere. Ionosphere* berada pada jarak sekitar 43-50 mil diatas permukaan bumi. Satelit yang melewati *ionosphere* akan menjadi lambat dikarenakan adalanya *plasma (*gas dengan tingkat kepadatan rendah). Walaupun *GPS receiver* berusaha untuk mengkoreksi/memperbaiki factor keterlambatan yang terjadi tetap saja aktivitas tertentu dari *plasma* bisa menyembabkan kesalahan perhitungan;
3. Keadaan *Troposphere*. *Troposphere* adalah bagian terendah dari *atmosfer* sampai dengan ketinggian sekitar 11 mil dari permukaan tanah. Variasi pada temperature, tekanan dan kelebaban bisa menyebakan perbedaan kecepatan penerima gelombang radio;
4. Kesalahan waktu. Kesalahan waktu dari *GPS receiver* yang tidak presisi dapat menimbulkan ketidakakurasian;
5. Kesalahan *Multipath.* Terjadi karena sinyal satelit membentur permukaan keras (seperti bangunan atau tebing) sebelum mencapai *GPS receiver.* Hal tersebut bisa menyebabkan terjadinya *delay* sehingga perhitungan jarak menjadi tidak akurat;
6. Buruknya sinyal satelit. Keadaan sekitar atau keadan lingkungan dapat juga menyebabkan *GPS* sulit menerima data satelit. (Sallatan, 2013)

### Android dan Global Positioning System

Saat ini teknologi berkembang dengan pesat terutama pada teknologi komunikasi. Dulu ponsel hanya sekedar digunakan untuk menelepon dan mengirim *Short Message Service(SMS)* atau pesan singkat saja. Salah satu ponsel saat ini adalah ponsel dengan sistem operasi android. (Oklilas, Anti, & Rachman, 2015)

Dengan standarisasi fitur dan *hardware* yang dimiliki, saat ini banyak orang yang menyukai ponsel dengan sistem operasi Android. Bukan karena canggih karena adanya fitur *Multimedia Messaging service(MMS),* radio atau internet berkecepatan tinggi tapi juga karena dilengkapinya fitur teknologi satelit didalamnya. Ya, perangkat *GPS receiver* yang saat ini sudah bisa dimiliki dengan hanya membeli sebuah ponsel.

Sebagai contoh pengguna dapat mengetahui jarak yang akan ditempuh dari titik asal pengguna hingga ke tempat tujuan pengguna. Tidak hanya itu saat ini *GPS* dapat digunakan untuk membantu memberikan peringatan awal terhadap terjadinya bencana alam.

## Location Base Service

*Location Based Service* (LBS) atau layanan berbasis lokasi merupakan  
layanan informasi yang dapat diakses melalui *mobile device* dengan menggunakan  
*mobile network*, yang dilengkapi dengan kemampuan untuk memafaatkan lokasi  
dari *mobile device* tersebut. *Location Based Service* dapat berfungsi sebagai  
layanan untuk mengidemtifikasi lokasi dari seseorang atau suatu objek tertentu, seperti menemukan tempat wisata atau lokasi lainnya. (Almuzakk, 2013)

* + 1. **Komponen Location Based Service (LBS)**

Dalam menggunakan *Location Based Service* atau layanan berbasis lokasi  
terdapat beberapa elemen yang diperlukan. Elemen yang diperlukan adalah sebagai  
berikut.

1. *Mobile Devices*

Sebuah alat yang digunakan untuk meminta informasi yang dibutuhkan seperti perangkat *mobile smartphone* yang memiliki fasilitas navigasi;

1. *Comunication Network*

Jaringan seluler yang mengirim data pengguna dan permintaan layanan;

1. *Positioning Component*

Untuk dapat memperoleh layanan harus mementukan lokasi pengguna. Posisi pengguna ini dapat diperoleh menggunakan jaringan komunikasi atau dengan menggunakn *Global Positioning System (GPS);*

1. *Service and Aplication Provider*

Penyedia layanan penguna seluler yang bertanggung jawab untuk memperoses suatu layanan;

1. *Data and Content Provider*

Penyedia layanan informasi data yang diminta oleh pengguna. (Kuswaha & Vineet, 2011)

* + 1. **Latitude (Garis Lintang)**

*Latitude* adalah garis lintang yang menghubungkan sisi bagian bumi dari  
timur dan barat. *Latitude* dibedakan menjadi dua wilayah yaitu lintang utara dan  
selatan dimana nilai koordinat dibagian utara selalu positif dan koordinat selatan  
selalu negatif. *Latitude* dapat digunakan untuk menentukan suatu lokasi di  
permukaan bumi. Titik yang dipakai dari 0 ke 90 derajat ke arah kutub utara dan 0  
ke -90 derajat ke kutub selatan. Contoh membaca posisi bumi dari koordinat GPS  
untuk kota Majalengka. *Latitude* untuk kota Majalengka adalah -6.837735. *Latitude* -6, maksudnya berada dibawah garis khatulistiwa.



Gambar 2.7 Latitude Kota Majalengka

* + 1. **Longitude (Garis Bujur)**

*Longitude* adalah garis bujur yang menghubungkan antara sisi utara dan sisi  
selatan permukaan bumi *Longitude* dibedakan menjadi dua wilayah yaitu bujur  
timur dan bujur barat dimana nilai koordinat yang berada di timur selalu negatif  
dan nilai koordinat dibagian barat selalu positif. Titik diawali dari 0 ke 180 derajat  
dan dari 0 ke -180 ke arah sebaliknya. Contoh membaca posisi bumi dari koordinat GPS untuk kota Majalengka. *Longitude* kota Majalengka adalah 108.229495. Karena logitude 0 berada di Ghana, sedangkan arah +180 berada di Hawai maka posisi *logitude* untuk kota Majalengka adalah +107 derajat.



Gambar 2.8 *Longitude* kota Majalengka

## Android

*Android* adalah sistem operasi yang berbasis *linux* untuk telepon seluler seperti telepon pintar (*Smartphone*) dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya *Google Inc.* membeli *Android Inc*. pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android,* dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile* dan *Nvidia.*

Beberapa pengertian lain dari Android, yaitu:

1. Merupakan *platform* terbuka (Open Source) bagi para programmer untuk membuat atau mengembangkan aplikasi;

2. Merupakan sistem operasi yang dibeli oleh *Google Inc* dari *Android Inc*.

### Arsitektrur Android

Tentunya *Android* mempunyai arsitektur tersendiri, arsitektur *android* dapat digambarkan seperti pada gambar 2.1. arsitektur *android*  dapat di jelaskan sebagai berikut:



Gambar 2.9 *Arsitektur Android* (Developer Android, 2017)

1. *Aplication* dan *Widgets*

Aplication dan Widgets merupakan layer yang berhubungan dengan aplikasi-aplikasi inti yang berjalan pada Adroid OS. Seperti klien email, kalender, *browser,* peta, kontak, dan lain-lain. Semua aplikasi ini dibuat dengan menggunakan Bahasa java.

1. *Aplication Framework*

*Application Framework* merupakan *layer* dimana para pembuat aplikasi menggunakan komponen-komponen yang ada untuk membuat aplikasi mereka. Beberapa contoh komponen yang termasuk didalam Apliction Framework adalah sebagai berikut:

1. *View*
2. *Content Provider*
3. *Resource Manager*
4. *Notivication Manager*
5. *Activity Manager*
6. *Libraries*

*Libraries* merupakan *layer* tempat fitur-fitur a*ndroid* berada. Pada umumnya *libraries* diakses untuk mejalankan aplikasi. Beberapa *library* yang terdapat pada *android* diantaranya adalah *libraries* media untuk memutar media *video,* atau *audio, libraries* untuk mejalankan tampilan, *libraries graphic, libraries SQLite* untuk dukungan *database.*

1. *Android Runtime*

*Android Runtime* merupakan *layer* yang membuat aplikasi *android* dapat dijalankan. *Android Runtime* dibagi menjadi dua bagian.

1. Android Runtime: sebuah mesin virtual yang dioptimalkan untuk mejalanan fungsi-fungsi pada *android* secara efisien.
2. *Core Libraries*: berfungsi untuk menterjemahkan Bahasa Java/C.
3. *Linux Kernel*

*Linux Kernel* adalah layer dimana inti dari sistem operasi *android* itu berada. Berisi file-file sistem yang mengatur sistem *processing, memory, resource, drivers,* dan sistem-sistem operasi *android* lainya. *Linux Kernel* ini berfungsi sebagai jembatan penghubung antara perintah-perintah yang dibuat terhadap *hardware* yang digunakan pada android. (Developer Android, 2017)

### Komponen Android

Aplikasi android dibangun menggunakan Bahasa pemograman Java. Untuk membuat aplikasi Android terdapat beberapa komponen utama didalamnya. Ada 4 jenis komponen pada aplikasi android, setiap jenis komponen memiliki tujuan yang berbeda dan memiliki siklus yang berbeda pula. Dibawah ini adalah komponen android yang dimaksud.

1. *Activity*

Merupakan suatu tampilan (*User Interface)* untuk pengguna melakukan interaksi dengan aplikasi Android. Sebuah aplikasi Android biasanya memiliki lebih dari satu *activity* tergantung tujuan dan desain aplikasi tersebut. Untuk berpindah dari satu *activity* ke *activity* lain dapat dilakukan dengan suatu *event* seperti klik tombol.

1. *Service*

*Service* ini tidak memiliki *Grafic User Interface (GUI)* seperti komponen *activity* tetapi *service* berjalan secara *background.* Maksudnya jika kita sedang menjalankan satu aplikasi, maka kita bisa membukan aplikasi yang lain selama aplikasi yang pertama masih berjalan.

1. *Broadcat Receiver*

*Broadcat Receiver* adalah komponen aplikasi yang menanggapi *system-wide broadcat announcements.* Contohnya adalah memunculkan notifikasi ketika adanya aplikasi baru yang ter-*install. Broadcast Receiver* juga tidak memiliki tampilan, tetapi dapat menjalankan suatu *activity* atau menampikan notifikasi.

1. *Intent*

*Intent* merupakan pesan *Asynchronous* yang memungkinkan aplikasi untuk memberi *request* secara fungsional dari komponen yang berbeda-beda didalam sistem android.

## Peralatan Pendukung

*Tools* pengukung yang digunakan dalam penelitian ini merupakan alat yang digunakan untuk menggambarkan bentuk dari logika model dari suatu sistem dengan menggunakan simbol-simbol, lambang – lambang, diagram – diagram yang menunjukan secara tepat arti dan fungsinya. Adapun alat pendukung yang digunakan adalah sebagai berikut:

### Java



Gambar 2.10 Logo Java

Sebuah program yang dapat diletakkan di berbagai macam perangkat elektronik, sehingga Java harus bersifat tidak bergantung pada platform (*platform  
independent*). Itulah yang menyebabkan dalam dunia pemrograman  
Java dikenal adanya istilah „*write once, run everywhere*‟, yang berarti  
kode program hanya ditulis sekali, namun dapat dijalankan di bawah  
kumpulan pustaka (*platform*) manapun, tanpa harus melakukan  
perubahan kode program.

Menurut definisi *Sun Microsystem*, di dalam buku Salahuddin dan Rosa,( 2011) *Java* adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (*stand alone*) ataupun pada lingkungan jaringan.

Java berdiri di atas sebuah mesin penterjemah (*interpreter*) yang diberi nama *Java Virtual Machine (JVM).* *JVM* inilah yang akan membaca kode bit (*bytecode*) dalam file *.class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada system operasi tersebut terdapat *JVM*. Alasan utama pembentukan bahasa Java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat elektronik, sehingga Java harus bersifat tidak bergantung pada platform (*platform independent*). Itulah yang menyebabkan dalam dunia pemrograman Java dikenal adanya istilah „*write once, run everywhere*‟, yang berarti kode program hanya ditulis sekali, namun dapat dijalankan di bawah kumpulan pustaka (*platform*) manapun, tanpa harus melakukan perubahan kode program.

### MySQL



Gambar 2.11 Logo MySQL (Fathansya, 2012)

MySQL merupakan DBMS(*Database Management System)* yang pertama kali dimulai di kembangkan pada tahun 1994 oleh sebuah perusahaan *software* bernama *Text Data Konsult AB* yang kemudian berganti hari label menjadi MySQL-AB. “My” pada kata MySQL sebenarnya bukan berarti MY dalam bahasa Inggris, tetapi konon merupakan nama putri dari Michael Widenius, pemrogram DBMS tersebut. Versi lain menyebutkan “My” adalah kependekan dari “Monty” yang merupakan julukan untuk Michael Widenius. (Fathansya, 2012)

Dewasa ini MySQL digunakan oleh sebagian besar *Web Server* yang ada di jagat internet. Disamping karena dianggap simple, juga dapat di-*porting* pada berbagai Sistem Operasi sekelas *server*, seperti Windows, Linux, Solaris, Max OS, BSD, Unix, IBM-AIX. (Fathansya, 2012)

MySQL (My Structure Query Language) adalah adalah aplikasi atau sistem untuk mengelola database atau manajemen data.Untuk menyimpan segala informasi kekomputer mengunakan data. MYSQL bertugas mengatur dan mengelola data-data pada database, selain itu MYSQL dikenal sebagai sistem yang efisien dan reliable, proses query cepat dan mudah, sehingga cocok digunakan untuk aplikasi berbasis web. (Fathansya, 2012)

### Android Studio

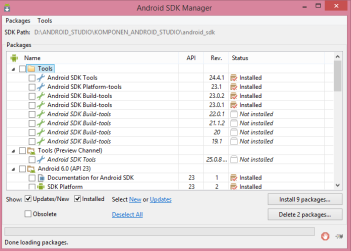
Android Studio merupakan sebuah IDE (*Integreted Development Environment*) untuk platform Android. Android Studio termasuk kedalam *tools* yang bersifat *open source*. Android Studio ini mendukung pengembangan berbasis *Gradle*, dapat melakukan perbaikan dengan cepat, memiliki sebuah layout editor yang kaya dan memungkinkan pengembang untuk melakukan *drag* and *drop User Interface (UI*). Untuk dapat melihat tampilan layout yang dibuat, pengguna tidak perlu memindahkan tab *layout*, cukup memilih tampilan priview, pengguna dapat mendesain sekaligus melihat tampilannya tanpa berpindah tab layout.



Gambar 2.12 Logo Android Studio

### Software Development Kit

Software Development Kit (SDK) adalah suatu kit atau library dari bahasa pemograman untuk pengembangan atau pembangunan suatu perangkat lunak dan biasanya SDK erdiri dari kumpulan tools yang dibutuhkan. Misalnya Bahasa pemograman Java, mempunyai SDK yang berisi suatu library yang dapat digunakan untuk membuat suatu aplikasi berbasis Java. Setiap kali Google merilis Android versi terbaru, sebuah SDK yang sesuai dengan versi Android juga dirilis. Sehingga pengembang dapat membuat aplikasi Android dengan fitur terbaru.



Gambar 2.13 Software Development Kit

### Java Development Kit

Java Development Kit (JDK) merupakan sekumpulan perangkat lunak yang dapat digunakan untuk mengembangkan perangkat lunak yang berbasis Java, sedangkan JRE adalah sebuah implementasi dari Java Virtual Machine yang benarbenar digunakan untuk menjalankan program Java. Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber compiler java, building, debuggers, development libraries dan lain sebagainya. Perbedaan JDK dengan SDK yaitu JDK adalah sebuah SDK tetapi sebuah SDK tidak harus menjadi sebuah JDK.



Gambar 2.14 Java Development Kit

### Java Script Object Notation (JSON)

*Java Script Object Notation (JSON)* adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. *JSON* merupakan format teks yang tidak bergantung pada Bahasa pemograman apapun karena menggunakan gaya Bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, Java Script, Python dll. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran data. *JSON* terbuat dari dua struktur. (JSON, 2017)



Gambar 2.15 Logo JSON

### PHP

*Hypertext Preprocessor(PHP)* adalah Bahasa pemograman *server side* yang dirancang khusus untuk pengembangan aplikasi berbasis *web. Script PHP* disisipkan kedalam *script Hyper Text Markup Language (HTML)* yang kemudian akan di proses pada suatu *web server* dan akan menghasilkan suatu halaman *web. PHP* termasuk kedalam *open source product* sehingga mengubah *source code* dan mendistribusikan secara bebas. (Nugroho, 2012)

# BAB III METODOLOGI PELAKSANAAN

**BAB III  
METODOLOGI PELAKSANAAN**

Pada bab ini akan dibahas mengenai Metodologi Pelaksanaan yang berisi kerangka berfikir yang didalamnya terdapat metode pengumpulan data, metode pengembangan sistem, objek penelitian analisis sistem yang sedang berjalan , dan sistem yang akan dibangun.

## Analisis Sistem

analisis sitem dari software yang akan digunakan serta spesifikasi dari minimum yang diperlukan.

### Analisis Kebutuhan Software

Perangkat lunak yang digunakan dalam pembuatan Sistem Absensi Pegawai adalah sebagai berikut:

1. Android Studio;
2. Database MySQL;
3. Editor Sublime;
4. CorelDraw X7;
5. Sistem Operasi Windows 10;
6. NetFramwok 4.5;
7. XAMPP.

### Analisis Kebutuhan Hardware

1. Monitor;
2. Perangkat Pheriperal;
3. Printer;
4. CPU dengan spesifikasi:

Processor Inter Core I3 2.3Ghz;

RAM DD3 4GB;

VGA NVIDIA 940;

DVD-RW.

### Analisis Kebutuhan Pengguna

Pengguna sistem absensi ini adalah semua pegawai yang ada dilingkungan SMPN 5 Majalengka dengan kebutuhan minimal *smartphone* sebagai berikut:

1. Menggunakan Android minimal versi 4.4 atau diatasnya;
2. RAM 1GB;
3. GPU Andreno 206;
4. Prosesor Quad Core 1.2 GHz;
5. Sensor GPS Pada *Smartphone.*

## Kerangka Berfikir



Gambar 3.1 Kerangka Berfikir

Untuk lebih jelas mengenai gambar kerangka berfikir diatas adalah sebagai berikut:

1. Pada tahapan studi pustaka, studi lapangan, dan studi literature membahas mengenai cara pengumpulan bahan atau data suatu penelitian yang akan dijadikan sebagai acuan dalam pembuatan kerja praktik;
2. Pada tahapan Scope Definition membahas mengenai cakupan umum dan batasan yang akan digunakan dalam pembuatan suatu aplikasi seperti gambaran umum dan lingkungan penelitian dari objek yang diteliti;
3. Pada tahapan Analisis Sistem membahas mengenai seluruh aspek dari suatu sistem yang akan kita buat mulai dari sistem yang sedang berjalan, kelemahan sistem yang sedang berjalan, analisis sistem usulan, analisis sistem pemecahan masalah dan perbandingan literatur sejenis;
4. Pada tahapan Desain Sistem membahas mengenai kerangka *interface sistem*,  
   kerangka *database sistem* dan kerangka *interface* sistem yang didalamnya  
   terdapat beberapa digram – diagram tergantung dari metodologi  
   pengembangan sistem yang digunakan;
5. Pada tahapan Implementasi Sistem membahas mengenai proses instalasi dan  
   pengujian sistem yang telah dibuat.

## Objek Penelitian

Objek penelitian ini adalah dimana penulis melakukan kerja praktik. Penulis melakukan kerja praktik di SMP Negeri 5 Majalengka.

### Sejarah Singkat SMP Negeri 5 Majalengka

Sekolah Menengah Pertama Negeri 5 Majalengka atau disingkat SMPN 5 Majalengka berdiri sejak 1999, beralamat di Jalan Raya Cibodas No. 08 Desa Cibodas. Berdasakan Surat Keputusan SK Mendikbud Nomor : 442/Kep. 613.Dis PK/02 tanggal 16 September 2002. SMPN 5 Majalengka memiliki luas tanah 5.765 M² yang didalamnya terdiri dari 11 ruang belajar kelas, laboratorium, 2 ruang guru, 1 ruang kepala sekolah, 1 ruang Tata Usaha, 9 WC, 1 ruang perpustakaan, 1 mushola, 1 ruang dapur, 1 ruang UKS.

Jumlah siswa yang belajar di sekolah ini yaitu 275 siswa untuk tahun ajaran 2014/ 2015, 280 siswa untuk tahun ajaran 2015/ 2016, 291 untuk tahun ajaran 2016/2017 dan 295 untuk tahun ajaran 2017/2018. SMP Negeri 5 Majalengka memiliki 26 Guru Pengajar, dan 10 Pegawai Tata Usaha.

### Struktur Organisasi

Struktur Organisasi

SMP Negeri 5 Majalengka

Tahun Pelajaran 2017-2018



Gambar 3.2 Struktur Organisasi

### Visi dan Misi SMP Negeri 5 Majalengka

1. Visi SMPN 5 Majalengka

MANDIRI

(Mutu Andal dalam Nuansa yang Demokratis dan Islami Rakyat Indonesia)

1. Misi SMPN 5 Majalengka
2. Terdidik dalam tatakrama budaya Indonesia yang Islami;
3. Pembaharu budaya mutu Pendidikan dan ilmu pengetahuan dalam kerangka Manajemen Peningkatan Mutu Pendidikan berbasis sekolah;
4. Terciptanya suasana pembinaan Kesiswaan yang demokratis, kreatif, inovatif dan berkeunggulan serta membekali siswa kecakapan hidup (life skill) yang mandiri serta berdaya guna;
5. Terwujudnya wawasan Wiyata Mandala dengan berbasis masyarakat yang dapat dibanggakan.

### *Flowchart* Sistem Absensi yang Sedang Berjalan



Gambar 3.3 Sistem Yang dedang

Penjelasan dari alur diatas adalah:

1. Pegawai datang kesekolah;
2. Pegawai mengambil kertas absensi;
3. Pegawai memeriksa namanya pada kertas absensi;
4. Bila ada nama pegawai maka melanjutkan absensi bila tidak ada kembali mengamambil kertas absensi;
5. Pegawai melakukan absensi dikertas yang sudah disediakan.

### *Flowchart* Sistem Absensi Usulan



Gambar 3.4 Sistem usulan

Penjelasan dari alur diatas adalah:

1. Pegawai datang kesekolah;
2. Pegawai membuka aplikasi absensi digital;
3. Pegawai melakukan abasensi;
4. Data absensi tersimpan kedatabase;
5. Proses absensi selesai.

# PERANCANGAN

Pada bab ini penulis menggambarkan seperti apa aplikasi absensi digital yang dirancan untuk SMP Negeri 5 Majalengka, mulai dari perancangna database, perancangan alur sistem sampai ke perancangan tampilan aplikasi abasensi digital.

## Perancangan Basis Data(*Database)*

Perancangan basisdata ini berisi tentang struktur table yang diperlukan dalam basisdata untuk pengolahan data didalam sistem absensi kepegawai digital di SMPN 5 Majalengka. Adapun tabel database yang digunakan sebagai berikut:

### Tabel Pegawai

Tabel pegawai digunakan untuk menyimpan data pegawai yang ada di SMPN 5 Majalengka. Tabel transaksi ini terdiri dari 8 field dan sebagai *primary key* NIP. Atribut-atribut yang ada pada tabel pegawai yaitu terlihat pada tabel dibawah ini:

Nama Tabel : pegawai

Primaty key : NIP

Tabel 4.1 Tabel *Database* Pegawai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama Field | Tipe | Panjang | Keterangan |
| 1 | Nip | Int | 12 | NIP |
| 2 | Nama | Varchar | 20 | Nama Pegawai |
| 3 | Alamat | Varchar | 50 | Alamat pegawai |
| 4 | Jk | Varchar | 10 | Jenis Kelamin Pegawai |
| 5 | Jabatan | Varchar | 20 | Jabatan pegawai |
| 6 | Email | Varchar | 25 | Email pegawai |
| 7 | Nohp | Varchar | 20 | No Handphone pegawai |
| 8 | imei | Varchar | 25 | Imei pegawai |

### Tabel Absensi

Tabel absensi digunakan untuk menyimpan data absensi pegawai yang ada di SMPN 5 Majalengka. Tabel transaksi ini terdiri dari 6 field dan sebagai *primary key* NIP. Atribut-atribut yang ada pada tabel pegawai yaitu terlihat pada tabel dibawah ini:

Nama Tabel : absensi

Primary key : absensiId

Tabel 4.2 Tabel *Database* Absensi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama Field | Tipe | Panjang | Keterangan |
| 1 | absensiId | Int | 50 | Absensi id |
| 2 | Nip | Int | 15 | NIP Pegawai |
| 3 | Tanggal | Date | - | Tanggal absensi |
| 4 | Jam\_masuk | Varchar | 50 | Jam masuk pegawai |
| 5 | Jam\_keluar | Varchar | 50 | Jam keluar pegawai |
| 6 | Selisih | Varchar | 50 | Waktu kerja pegawai |

## Diagram UML (*Unified Modeling Language)*

Seperti definisinya UML yaitu Bahasa permodelan untuk sistem atau perangkat lunak yang berorientasi objek dan digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami, untuk itu dibuatkan UML dalam pembuatan aplikasi absensi kepegawaian digital ini meliputi Use Case Diagram, class diagram, sequence diagram, dan activity diagram.

### Use Case Diagram

*Usecase* diagram adalah diagram *usecase* yang digunakan untuk menggambarkan secara ringkas siapa saja yang menggunakan sistem dan apa saja yang bias dilakukannya. Diagram *usecase* tidak menjelaskan secara detai tentang pengunaan *usecase*, namun hanya memberi gambaran singkat hubungan antara *usecase*, aktor, dan sistem. Melalui diagram *usecase* dapat diketahui fungsi-fungsi apa saja yang ada pada sistem. Nama suatu *usecase* harus didefinisikan semudah mungkin dan dapat dipahami.



Gambar 4.1 Usecase Diagram Alur sistem

Berikut spesipikasi Usecase Diagram aplikasi absensi digital kepegawaian log in, absensi masuk, absensi keluar, log out, dan laporan.

Tabel 4.3 *Usecase Log In*

|  |  |
| --- | --- |
| Nama UseCase | Log In |
| Aktor | Pegawai dan Admin |
| Deskripsi | Untuk mengakses tombol absensi |
| Precondition | Tombol absensi tidak bisa ditekan |
| Passcondition | Tombol absensi bisa ditekan |
| Asumsi | Jika belum login tombol absensi tidak bisa ditekan |
| Kriteria UseCase | High |
| Prioritas | High |
| Resiko | High |

Tabel 4.4 *Usecase* Absensi Masuk

|  |  |
| --- | --- |
| Nama UseCase | Absensi Masuk |
| Aktor | Pegawai dan Admin |
| Deskripsi | Untuk melakukan absensi |
| Precondition | Belum bisa melakukan absensi |
| Passcondition | Sudah bisa melakukan absensi |
| Asumsi | Jika log in tidak berhasil maka tombol abensi masuk tidak akan bisa ditekan |
| Kriteria UseCase | High |
| Prioritas | High |
| Resiko | High |

Tabel 4.5 *Usecase* Absensi Keluar

|  |  |
| --- | --- |
| Nama UseCase | Absensi Keluar |
| Aktor | Pegawai dan Admin |
| Deskripsi | Untuk melakukan absensi keluar |
| Precondition | Belum bisa melakukan absensi |
| Passcondition | Sudah bisa melakukan absensi |
| Asumsi | Jika absensi masuk belum sukses belum berhasil maka tombol absensi keluar tidak bisa di tekan tetapi kalau absensi masuk berhasil tombol absensi keluar tersedia |
| Kriteria UseCase | High |
| Prioritas | High |
| Resiko | High |

Tabel 4.6 *Usecase Log out*

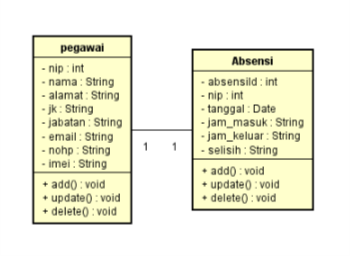
|  |  |
| --- | --- |
| Nama UseCase | Log Out |
| Aktor | Pegawai dan admin |
| Deskripsi | Untuk keluar dari aplikasi |
| Precondition | Belum dapat keluar dari aplikasi |
| Passcondition | Sudah dapat keluar dari aplikasi |
| Asumsi | Jika tidak logout maka belum bisa keluar dari aplikasi |
| Kriteria UseCase | High |
| Prioritas | High |
| Resiko | High |

Tabel 4.7 *Usecase* laporan

|  |  |
| --- | --- |
| Nama UseCase | Laporan |
| Aktor | Admin |
| Deskripsi | Untuk menampikan Laporan absensi |
| Precondition | Belum dapat menampilkan laporan |
| Passcondition | Sudah dapat menampilkan laporan |
| Asumsi | Jika membuka halaman web maka laporan bisa dibuka |
| Kriteria UseCase | High |
| Prioritas | High |
| Resiko | High |

### Class Diagram

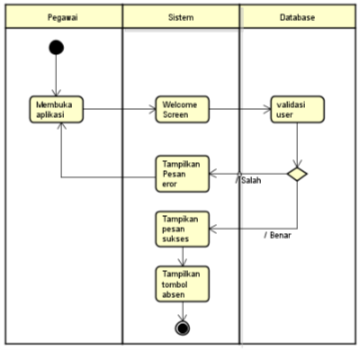
Class diagram yang digunakan untuk menampikan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang kita gunakan, berikut adalah diagram aktifitas dari sistem absensi kepegawaian digital sebagai berikut:



Gambar 4.2 Class Diagram Aplikasi Absensi Kepegawaian Digital

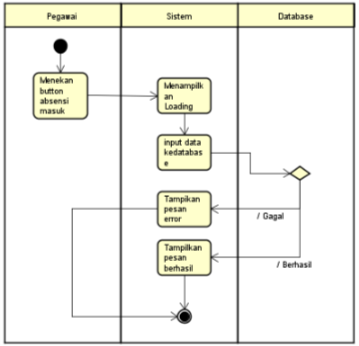
### Activity Diagram

*Activity Diagram* menggambarkan penggunaan aplikasi absensi kepegawaian digital yang dirancang, dibuat berdasarkan form-form yang terdapat di aplikasi, berikut gambar *activity diagram* yang dirancang:

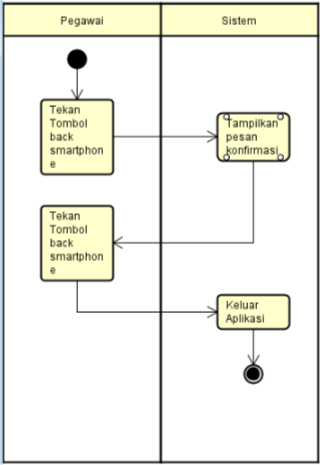


Gambar 4.3 *Activity Diagram Log In*

Selanjutnya adalah *activity diagram* absensi masuk, berikut adalah activitynya:



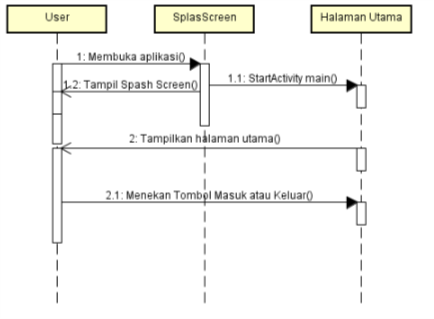
Gambar 4.4 *Activity Diagram* Absensi Masuk



Gambar 4.5 Activity Log Out

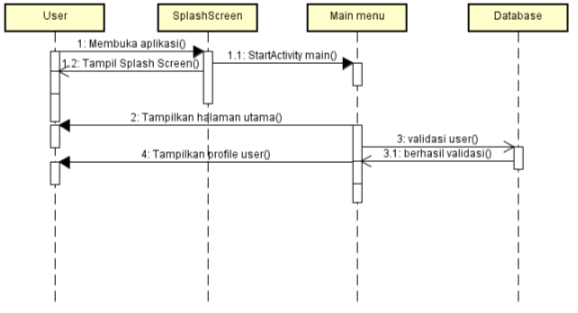
### Squence Diagram

Perancangan *Squence diagram* Dibuat untuk menggambarkan kolaborasi dinamis antar sejumlah objek, kegunaanya untuk menujukkan rangkaian pesan yang dikirim antar objek, berikut *Squence diagram:*



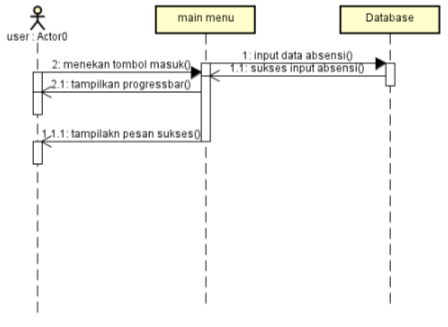
Gambar 4.6 *Sequence diagram* menggunakan aplikasi

Berikut dibawah ini adalah *sequence diagram* untuk login:



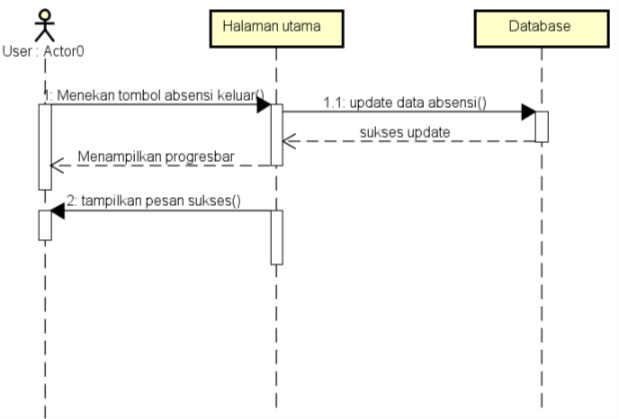
Gambar 4.7 *Sequence diagram Log In*

Rangakaian pesan yang dikirim antar objek dan juga interaksi antar objek berikutnya adalah *Sequence Diagram* absensi masuk, seperti yang terlihat pada gambar dibawah ini:



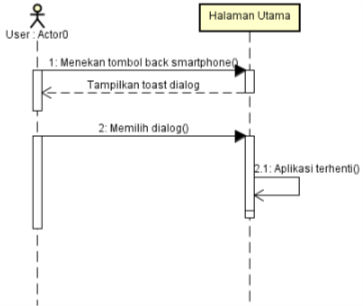
Gambar 4.8 *Sequence Diagram* Absensi Masuk

Rangakaian pesan yang dikirim antar objek dan juga interaksi antar objek berikutnya adalah *Sequence Diagram* absensi keluar, seperti yang terlihat pada gambar dibawah ini:



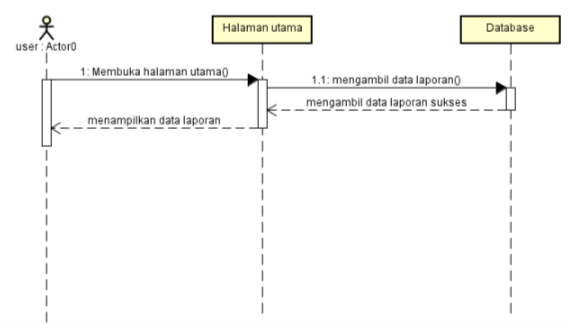
Gambar 4.9 *Sequence Diagram* Absensi Keluar

Rangakaian pesan yang dikirim antar objek dan juga interaksi antar objek berikutnya adalah *Sequence Diagram* keluar aplikasi, seperti yang terlihat pada gambar dibawah ini:



Gambar 4.10 *Sequence Diagram* Keluar Aplikasi

Rangakaian pesan yang dikirim antar objek dan juga interaksi antar objek berikutnya adalah *Sequence Diagram* Laporan, seperti yang terlihat pada gambar dibawah ini:

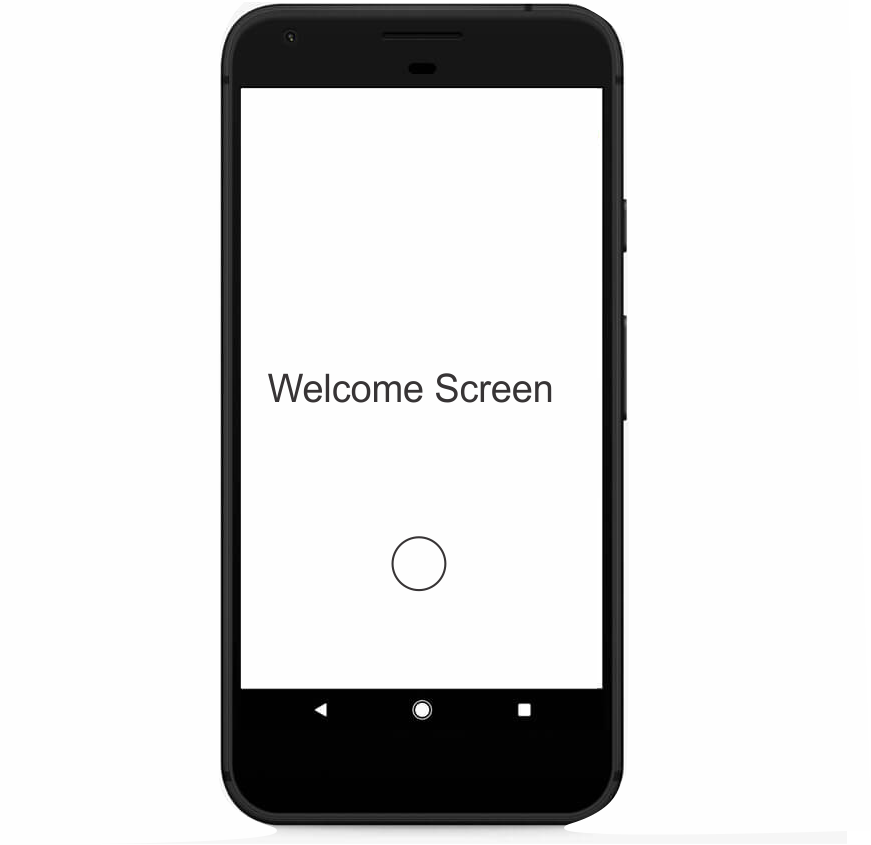


Gambar 4.11 *Sequence Diagram* Laporan

## Tampilan (*User Interface)*

1. Tampilan Splash Screen

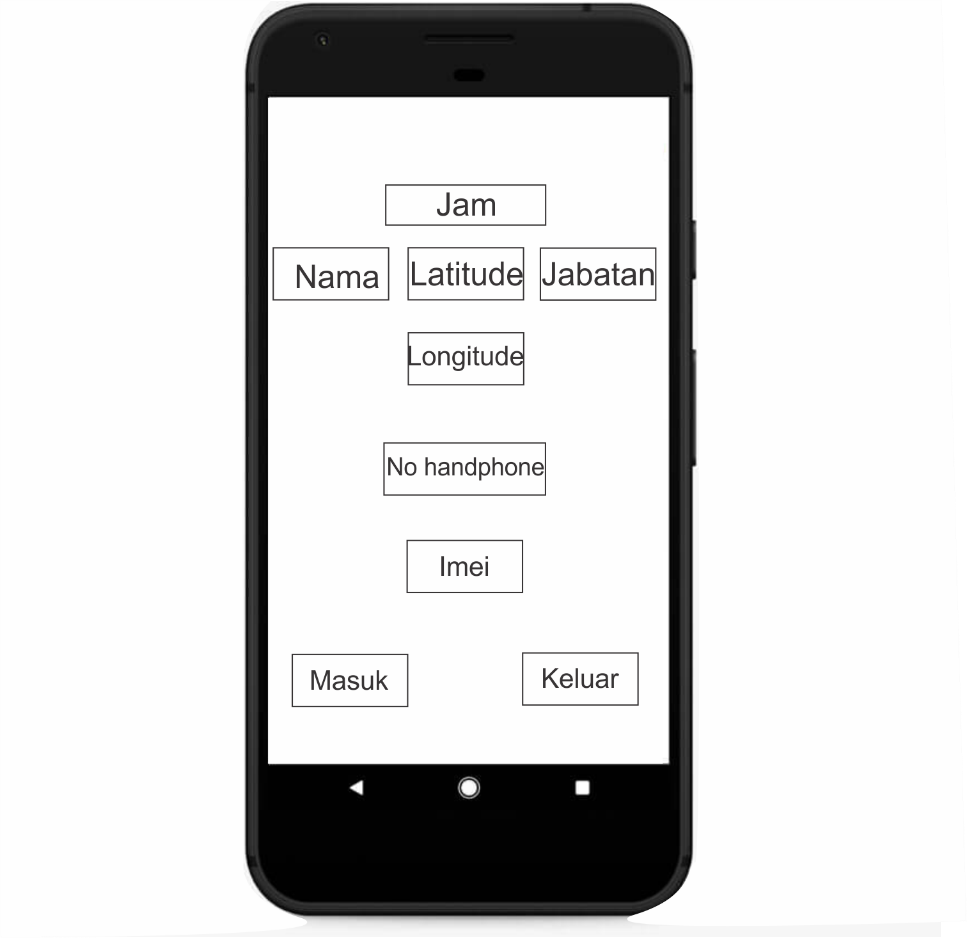
Tampilan ini berisi tentang tampilan awal aplikasi sebelum ke main menu tampilan ini berjalan ketika aplikasi pertama kali digunakan.



Gambar 4.12 Desain Splash Screen

1. Main Menu

Tampilan ini merupakan tampilan utama dalam aplikasi absensi digital kepegawaian yaitu tampilan yang menyimpan semua informasi atau tampilan lainnya baik itu tampilan help atau tampilan menu.



Gambar 4.13 Desain Main Menu

1. Help

Tampilan ini berisi tentang petunjuk penggunaan aplikasi absensi kepegawaian digital yang ditampikan petunjuk penggunaan aplikasi menunjukan bagaiman menggunakan aplikasi

# IMPLEMENTAS

# PENUTUP

# Daftar Pustaka

Almuzakk, M. A. (2013). *Rancang Bangun Aplikasi Location Based Service.* Semarang.

*Developer Android*. (2017, Oktober 24). Retrieved from developer.android.com: https://developer.android.com/

Fathansya. (2012). *Basis Data.* Bandung: Informatika.

Haviluddin. (2011). Memahami Penggunaan UML (Unifield Modelling Language). *Jurnal Informatika Mulawarman*, 15.

Informatika. (2017, Oktober 24). *Pengertian Basis Data*. Retrieved from Informatika.we.id: http://informatika.web.id/pengertian-basis-data.htm

Jogiyanto. (2010). *Analisis dan Desain.* Yogyakarta: Andi.

JSON. (2017, Oktober 24). *PENGENALAN*. Retrieved from JSON: http://www.json.org/json-id.html

Kristanto, A. (2008). *Perancangan Sistem Informasi dan Aplikasinya.* Yogyakarta: Gava Media.

Kuswaha, A., & Vineet, K. (2011). Location Based Service Using Android Mobile Operation System. *Internasional Jurnal of Advances in Engineering and Tecnology*, 14-20.

Mulyadi. (2010). *Sistem Akutansi.* Jakarta: Salemba Empat.

Nugroho, B. (2012). *Latihan Membuat Aplikasi Web PHP dan MySQL dengan Dreamweaver MX.* Yogyakarta.

Oklilas, A. F., Anti, S. D., & Rachman, M. D. (2015). Akurasi Pembacaan GPS pada Android untuk Location Based Service. 5.

Pressman. (2010). *Rekayasa Perangkat Lunak.* Yogyakarta: ANDI.

Proboyekti, U. (2011). *Extreme Programing.* Yogyakarta.

Ramakrishnan, & Gehrke. (2003). *Database Management System.* USA: Edition McGraw.

Salahuddin, M., & Rosa. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak.* Bandung: Modula.

Sallatan, Z. D. (2013). M-Library Berbasis Android Menggunakan Protokol Json.

SerbaTekno. (2017, Oktober 24). *Teknologi Location Based Service Pada Smartphone*. Retrieved from Serbatekno.com: http://www.serbatekno.com/2049/teknologi-location-based-service-pada-smartphone/

Stephens. (2003). *Database Design.* Sams Publising.

Subrata, K. (2017, 11 1). *Flowchat.* Retrieved from Academia.edu: https://www.academia.edu/6228702/Flowchart\_Jurnal

Susanto. (2013). *Sistem Informasi Akutansi.* Bandung: Linggar Jaya.

Sutabri, T. (2012). *Konsep Sistem Informasi.* Yogyakarta: Andi.

Sutarman. (2009). *Pengantar Teknologi Informasi.* Jakarta: Bumi Askara.