

BAB 2

TINJAUAN PUSTAKA

2.1 Profil Dinas Pariwisata Provinsi Nusa Tenggara Timur

Dinas Pariwisata Provinsi Nusa Tenggara Timur adalah salah satu dinas yang mempunyai tugas merencanakan operasional bidang pemasaran pariwisata meliputi pengembangan pasar dan informasi pariwisata, promosi pariwisata serta promosi wisata konvensi, insentif, event dan minat khusus berdasarkan ketentuan dan prosedur yang berlaku agar tercipta Nusa Tenggara Timur sebagai daerah tujuan wisata yang kompetitif.

2.1.1 Visi dan Misi

Visi

Terwujudnya Nusa Tenggara Timur Sebagai Salah Satu Destinasi Utama Pariwisata Indonesia Di Tahun 2018.

Misi

Dalam rangka mewujudkan Visi tersebut, Dinas Pariwisata Provinsi Nusa Tenggara Timur menetapkan Misi sebagai berikut :

1. Mengembangkan Destinasi yang berkualitas, berdaya saing, menarik dan berkelanjutan yang mampu mendorong pembangunan ekonomi daerah.
2. Mengembangkan Promosi Pariwisata yang berkualitas dengan memanfaatkan pengakuan dunia terhadap Komodo dan Kelimutu sebagai salah satu keajaiban dunia serta berbagai objek wisata lainnya dalam rangka meningkatkan arus kunjungan dan investasi di Nusa Tenggara Timur.
3. Meningkatkan Ekonomi Kreatif berbasis Seni Budaya dengan memaksimalkan potensi lokal, guna mendorong potensi ekonomi masyarakat.

4. Mengembangkan Ekonomi Kreatif berbasis Media, Desain dan Ilmu Pengetahuan dan Teknologi (IPTEK) dalam rangka membuka usaha baru bagi masyarakat.

2.1.2 Logo

Berikut ini adalah logo instansi dinas di Provinsi Nusa Tenggara Timur dapat dilihat pada Gambar 2.1 :



Gambar 2. 1 Logo Dinas Pariwisata Provinsi Nusa Tenggara Timur

2.1.3 Pulau Timor

Pulau Timor merupakan sebuah pulau yang berada di Provinsi Nusa Tenggara Timur (NTT), yang luas wilayah mencapai 4.937.62 km². Pulau Timor merupakan Pulau terbesar di jajaran kepulauan Sunda Kecil. Terdiri dari Timor Barat yang masuk wilayah Indonesia dan Timor Timur yang sekarang menjadi negara Republik Demokrasi Timor Leste. Pulau dengan julukan “Nusa Cendana” ini mempunyai ragam wisata seperti pantai, padang sabana, gunung, air terjun dan lain sebagainya. Berikut ini daftar tempat wisata yang ada di Pulau Timor.

Tabel 2. 1 Daftar Tempat Wisata

Nama Wisata	Lokasi			Koordinat	
	Desa/ Kelurahan	Kecamatan	Kota/ Kabupaten	Latitude	Longitude
Pantai Tedis	LLBK	Kota Lama	Kota Kupang	10.161301	123.576726

Nama Wisata	Lokasi			Koordinat	
	Desa/ Kelurahan	Kecamatan	Kota/ Kabupaten	Latitude	Longitude
Pantai Lasiana	Lasiana	Kelapa Lima	Kota Kupang	-10.1296736	123.6707237
Gua Kristal	Bolok	Kupang Barat	Kab. Kupang	-10.224566	123.5105867
Gunung Fatuleu	Oelbiteno	Fatuleu Tengah	Kab. Kupang	-9.952671	123.8624288
Air Terjun Oesusu	Oesusu	Takari	Kab. Kupang	-9.9851885	124.0098583
Gunung Timau		Amfoang Tengah	Kab. Kupang	-9.5802778	123.9397222
Pantai Panmuti	Noelbaki	Kupang Tengah	Kab. Kupang	-10.1089177	123.7061425
Pantai Manikin	Tarus	Kupang Tengah	Kab. Kupang	-10.1252066	123.6740469
Pantai Tablolong	Tablolong	Kupang Barat	Kab. Kupang	-10.318113	123.475198
Pantai Air Cina	Nefo	Kupang Barat	Kab. Kupang	-10.3260162	123.4636789
Air Terjun Oenesu	Oenesu	Kupang Barat	Kab. Kupang	-10.2673483	123.5646758
Air Terjun Tesbatan	Tesbatan	Amarasi	Kab. Kupang	-10.2152697	123.8774052
Pantai Teres	Buraen	Amarasi Selatan	Kab. Kupang	-10.2877585	123.9461385
Pantai Batu Tujuh	Sahraen	Amarasi Selatan	Kab. Kupang	-10.3408101	123.829215
Pantai Puru	Merbaun	Amarasi Barat	Kab. Kupang	-10.19588	123.7129159
Padang Fulan Fehan	Maudemu	Lamaknen	Kab. Belu	-9.12247981	125.0926093
Gunung Laknan	Maudemu	Lamaknen	Kab. Belu	-9.1067337	125.0614258
Gunung Mutis	Fatumnasi	Molo Utara	Kab. TTS	-9.566667	124.233333
Danau Fatukoto	Fatukoto	Molo Utara	Kab. TTS	-9.7103621	124.2032483
Bukit Fatuulan	Fatuulan	Ki'e	Kab. TTS	-9.9097428	124.6248039
Pantai Kolbano	Kolbano	Kolbano	Kab. TTS	-10.0204408	124.5385041
Pantai Oetune	Tuafanu	Kualin	Kab. TTS	-10.1690246	124.3971128
Taman Wisata Bawa Laut Pulau Kera	Uiasa	Sulamu	Kab. Kupang	-10.0896706	123.5560656
Pantai Liman	Oetefu Besar	Semau Selatan	Kab. Kupang	-10.2854544	123.3090133
Pantai Uitiuh Tuan	Oetefu Besar	Semau Selatan	Kab. Kupang	-10.2824604	123.3205544
Pantai Uih Make	Onanbalu	Semau Selatan	Kab. Kupang	-10.22867676	123.34196944
Pantai Naikean	Naikean	Semau Selatan	Kab. Kupang	-10.3057577	123.30540095

Nama Wisata	Lokasi			Koordinat	
	Desa/ Kelurahan	Kecamatan	Kota/ Kabupaten	Latitude	Longitude
Pantai Uinian	Uiboa	Semau Selatan	Kab. Kupang	-10.24121652	123.32394631
Pantai Uislaen	Uiboa	Semau Selatan	Kab. Kupang	-10.24533217	123.31973682

2.1.4 Jenis Liburan

Sebelum melakukan perjalanan wisata, sangat penting bagi wisatawan menentukan tujuan wisata yang akan dikunjungi saat berwisata nantinya. Menentukan tujuan wisata seharusnya sesuai dengan minat dan hobi wisatawan. Berikut adalah 3 jenis tujuan objek wisata yang terdapat di Pulau Timor – NTT :

1. Objek Wisata Alam.

Objek wisata Alam adalah tempat tujuan rekreasi, pengembangan pribadi, atau mempelajari daya tarik alam dengan memanfaatkan potensi sumberdaya alam, baik itu alami maupun budidaya. Beberapa contoh Objek Wisata Alam adalah wisata gunung, wisata bahari, wisata hutan, wisata sungai, wisata gua dan lain-lain.

2. Objek Wisata Budaya.

Objek Wisata Budaya adalah tempat tujuan rekreasi, pengembangan pribadi, atau mempelajari daya tarik budaya dengan memanfaatkan potensi budaya dari tempat yang dikunjungi tersebut. Beberapa contoh Objek Wisata Budaya adalah wisata religi, wisata edukasi, wisata sejarah, wisata kota dan lain-lain.

3. Objek Wisata Buatan.

Objek Wisata Buatan adalah tempat tujuan rekreasi, pengembangan pribadi, atau mempelajari daya tarik budaya dengan memanfaatkan potensi buatan seperti taman bermain dan kebun binatang.

2.2 Landasan Teori

Dalam penelitian ini digunakan landasan teori yang berisi teori – teori yang dapat dijadikan dasar dan acuan dalam Pembangunan Aplikasi Berbasis *Android* Rekomendasi *Travelling* Tempat Wisata Alam di Pulau Timor – NTT.

2.2.1 Travelling

Traveling dalam bahasa Indonesia adalah aktivitas melancong; berpindah dalam satu tempat ketempat lainnya dengan berbagai alasan, seperti bisnis, liburan, dan sebagainya. Tetapi menurut Herajeng Gustiyu dalam bukunya yang berjudul *The Backpacker's Notes*, *traveling* adalah lebih kepada bagaimana beradaptasi dengan lingkungan baru. Didalam buku tersebut juga dibahas 3 hal dasar yang penting dari membuat perencanaan *traveling* dan menentukan biaya yang tepat diantaranya adalah transportasi, akomodasi dan makanan [1].

2.2.2 Wisata

Wisata adalah bepergian secara bersama-sama dengan tujuan untuk bersenang-senang, menambah pengetahuan, dan lain-lain. Selain itu juga dapat diartikan sebagai bertamasya atau piknik. Pariwisata adalah suatu kegiatan perjalanan yang dilakukan dengan tujuan liburan atau rekreasi. Pengertian tempat wisata adalah suatu tempat yang digunakan untuk melakukan kegiatan wisata. Tempat wisata alam dapat berupa pantai, gunung, dan lain-lain, sedangkan tempat wisata bangunan dapat berupa peninggalan sejarah, museum, dan lain-lain. Sedangkan Wisatawan adalah orang yang melakukan kegiatan wisata [2].

2.2.3 Sistem

Sistem adalah sekumpulan komponen atau elemen yang saling berkaitan satu dengan lainnya dalam melakukan kegiatan bersama untuk memudahkan aliran suatu informasi atau materi untuk menghasilkan suatu tujuan yang sudah direncanakan [3].

2.2.3.1 Syarat – syarat Sistem

Pada pembangunan suatu sistem harus memenuhi beberapa persyaratan, diantaranya :

1. Sistem harus dibentuk untuk menyelesaikan suatu masalah.

2. Elemen sistem harus mempunyai rencana yang akan ditetapkan.
3. Adanya keterkaitan dari setiap elemen sistem.
4. Unsur dari proses (arus informasi, energi dan material) lebih penting dari pada elemen sistem.
5. Tujuan organisasi lebih penting dari pada tujuan elemen.

2.2.3.2 Elemen – elemen Sistem

Pada sebuah sistem memiliki beberapa elemen, yaitu : Tujuan, Masukan, Proses, Keluaran, Batas, Mekanisme Pengendalian dan Umpan Balik serta Lingkungan.

2.2.3.3 Klasifikasi Sistem

Klasifikasi sistem merupakan suatu komponen yang memiliki kesatuan yang sama, satu dengan lainnya, system yang memiliki tujuan akhir yang berbeda untuk setiap kasus atau perkara yang terjadi dalam setiap sistem. Maka sistem tersebut dapat diklasifikasikan menjadi beberapa sistem, diantaranya :

1. *Deterministik Sistem*

Sistem operasi (input atau output) yang didalamnya dapat ditentukan atau diketahui dengan pasti.

2. *Probabilistik Sistem*

Sistem input dan prosesnya dapat didefinisikan, tetapi output yang dihasilkan tidak dapat ditentukan dengan pasti, sistem ini memiliki sedikit kesalahan atau penyimpangan terhadap ramalan suatu jalannya sistem.

3. *Open Sistem*

Sistem yang dapat menyesuaikan diri dengan lingkungannya sehingga dapat meneruskan eksistensinya.

4. *Closed Sistem*

Sistem yang prosesnya tidak mengalami pertukaran materi,energi atau informasi dengan lingkungan di luar sistem.

5. *Relatively Closed Sistem*

Sistem yang bersifat tertutup tetapi dalam pengoperasiannya dapat menerima pengaruh dari luar yang sudah didefinisikan dalam batasan tertentu.

6. *Artificial Sistem*

Sistem yang dapat menyerupai kejadian alam atau dapat menirukan kejadian-kejadian yang ada di alam.

7. *Natural Sistem*

Sistem yang dibentuk dari kejadian dalam alam.

8. *Manned Sistem*

Sistem yang menjelaskan tingkah laku manusia dalam keikutsertaannya.

2.3 **Android**

Android dikembangkan oleh Google bersama *Open Handset Alliance* (OHA). *Open handset Alliance* merupakan aliansi perangkat selular terbuka yang terdiri dari 47 perusahaan Hardware, Software dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. Application Programming Interface (API) yang disediakan menawarkan akses ke hardware, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga [4].

2.3.1 **Android SDK (Software Development Kit)**

Sesuai dengan namanya merupakan kumpulan tools-tools yang diperlukan untuk mengembangkan aplikasi Android. Seperti pada java yang dapat dijalankan pada multiplatform, SDK Android pun tersedia untuk beberapa platform seperti Windows, Mac OS dan Linux [5].

Adapun aplikasi SDK yang didalamnya terdapat beberapa folder sebagai berikut :

1. Tools

Berisikan tools-tools yang diperlukan dalam membangun aplikasi Android seperti emulator, Android Virtual Device (AVD), SDK Manager, ddms dan banyak lagi untuk memudahkan membangun aplikasi Android. Semua tools ini dapat di update mengikuti setiap perkembangan platform Android.

2. Platforms

Direktori ini ditunjukkan untuk meletakkan versi platform-platform Android yang telah kita download dari SDK Manager. Seperti Android 1.5, 1.6, 2.0 dan seterusnya.

3. Add-ons

Direktori ini nantinya berisikan tools tambahan untuk Android SDK, yang fungsinya untuk menambahkan library yang akan digunakan untuk pengembangan aplikasi. Seperti jika ingin mengembangkan aplikasi menggunakan map dari google maka harus memerlukan platform Android yang sudah terdapat library map. Semua ini dapat di download melalui SDK Manager.

Selain folder-folder diatas, akan terdapat file tambahan jika mendownload beberapa platform dan tool dalam folder Android SDK tersebut. Seperti *Docs*, *platform tools*, dan *samples*.

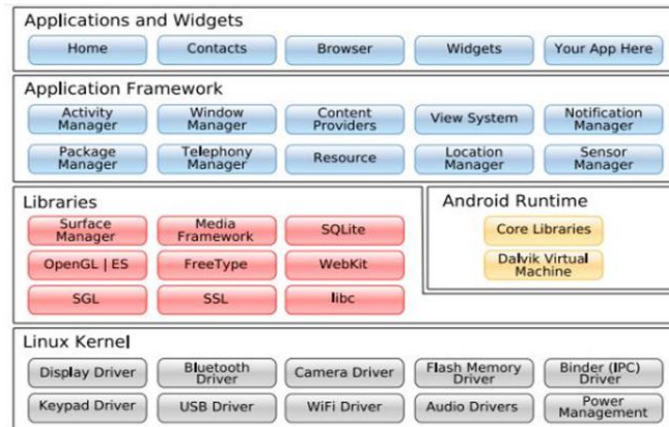
2.3.2 Android Development Tools

Merupakan plugin Eclipse yang disediakan oleh google untuk membantu dan memudahkan *Developer* mengembangkan aplikasi Android. Dengan adanya ADT membuat project Android, mendesain tampilan aplikasi, menambahkan komponen-komponen yang berasal dari Android Framework, *debug* aplikasi menggunakan tools-tools pada SDK Android hingga membuat *file installer* menjadi lebih mudah berkat plugin ini. Untuk proses penginstallan ADT dapat dilakukan dengan 2 cara Pertama secara Offline dengan mendownload file ADT secara manual pada situs Android dan kedua Secara online dengan menggunakan eclipse maupun Android studio [5].

2.3.3 Arsitektur Android

Arsitektur pada sistem Android memiliki 5 tingkat lapisan layer. Setiap lapisan layer tersebut menghimpun beberapa program yang yang mendukung

fungsi-fungsi spesifik dari sistem operasi. Pemisahan layer tersebut bertujuan untuk memberikan abstraksi sehingga memudahkan pengembang aplikasi, yaitu dapat dilihat pada Gambar 2.2 :



Sumber : <https://developer.android.com/guide/platform/index.html?hl=id>

Gambar 2. 2 Arsitektur Android

1) Layer Applicatoins and Widgets

Pada layer Application dan Widget inilah berjalan aplikasi-aplikasi ini. Seperti Email, SMS, Kalender, Browser, Peta, Kontak dan lain-lain. Bahasa Java digunakan untuk membuat aplikasi-aplikasi tersebut. Aplikasi yang kita buat akan berada di layer ini.

2) Layer Applications Framework

Application Framework adalah dimana beradanya komponen-komponen yang digunakan para pembuat aplikasi, untuk mengembangkan aplikasi mereka. Berikut contoh-conoh komponen yang masuk di dalam Applications Framework :

- a) *View*
- b) *Content Provider*
- c) *Resource Manager*
- d) *Notification Manager*
- e) *Activity Manager*

3) Layer Libraries

Pada layer Libraries inilah kita bisa temukan fitur-fitur dari Android. Untuk mengimplentasikan aplikasi biasanya mengakses libraries ini. Libraries ada dua,

yaitu libraries media dimana ini memutar video dan audio, dan libraries untuk menjalankan tampilan, seperti libraries graphic, libraries SQLite untuk support data base dan masih banyak library lainnya.

4) Android Run Time

Dilayer inilah aplikasi android dapat berjalan. Android RunTime dibagi menjadi 2 bagian yaitu:

- 1) *Core Libraries* : berfungsi menterjemahkan bahasa Java dan C.
- 2) *Dalvik Virtual Machine* : berfungsi sebagai virtual mesin berbasis register yang bertugas mengoptimalkan jalannya fungsi-fungsi di Android agar lebih efisien.

5) Linux Kernal

Pada Linux Linux Kernel adalah layer dimana inti dari system operasi Android itu berada. Berisi file-file system yang mengatur sistem processing, memory, resource, drivers, dan sistem-sistem operasi Android lainnya. Linux Kernel yang digunakan Android adalah Linux Kernel release 2.6. Apa maksud penggunaan kembali object dari elemen-elemen penyusun system operasi, ini maksudnya komponen yang terdapat pada aplikasi Android kita bisa gunakan ketika membutuhkannya [4].

2.3.4 Android Life Cycle

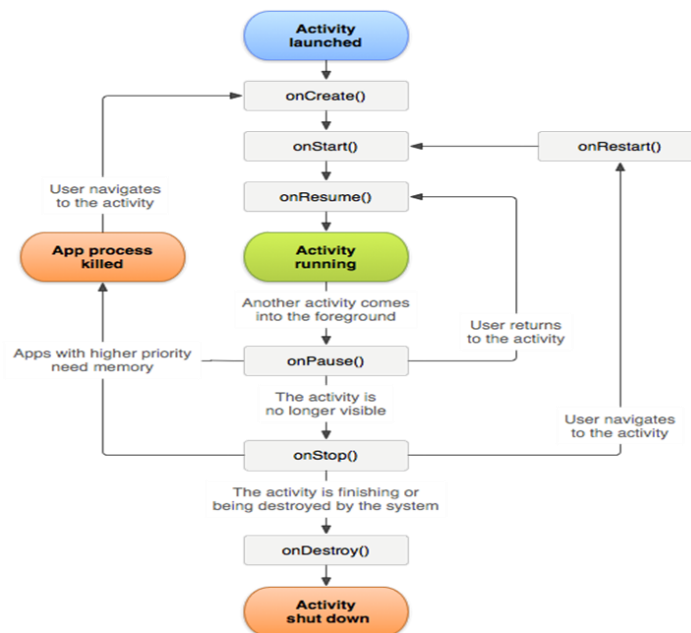
Aplikasi android terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android yang berbeda. Komponen Aplikasi Android, klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan [4].

Pada bahasa pemrograman c, c++, dan java, program dimulai dengan function main(). Hal ini sangat mirip, pada aplikasi android, program dimulai dengan method callback onCreate(). Urutan method callback dari mulai activity sampai berakhirnya [6].

Tabel 2. 2 Komponen Aplikasi Android

Functionality	Java Base Class	Example
Focused thing a user can do	Activity	Edit a note, play a game
Background process	Service	Play music, update weather icon
Receive messages	BroadcastReceiver	Trigger alarm upon event
Store and retrieve data	ContentProvider	Open a phone contact

Terdapat beberapa *method* dalam *Android Life Cycle* yang dapat dilihat pada Gambar 2.3 Android Life Cycle yaitu sebagai berikut :



Sumber : <https://developer.android.com/guide/components/activities/activity-lifecycle.html#alc>

Gambar 2. 3 Android Life Cycle

Tabel 2. 3 Method dalam Android Life Cycle

Method	Description
onCreate()	Method ini pertama kali dipanggil ketika activity pertama dimulai.
onStart()	Method ini dipanggil ketika activity sudah terlihat pada user.
onResume()	Method ini dipanggil ketika activity mulai berinteraksi dengan user.

onPause()	Method ini Dipanggil ketika activity berhenti sementara tidak menerima inputan user dan tidak mengeksekusi kode apapun.
onStop()	Method ini dipanggil ketika activity sudah tidak terlihat pada user.
onDestroy()	Method ini dipanggil sebelum sebuah activity di matikan.
onRestart()	Method ini dipanggil setelah activity berhenti dan ditampilkan ulang oleh user.

Namun, hanya beberapa dari *state* atau *method* tersebut yang menjadi statis diantaranya:

1. *Resumed*

Resumed terjadi ketika aplikasi berjalan setelah state paused. State ini akan menjalankan perintah program yang ditulis pada method `onResume()`.

2. *Paused*

Dalam keadaan ini aktivitas yang terjadi dihentikan secara sementara tetapi masih terlihat oleh pengguna karena terdapat proses yang memiliki prioritas lebih tinggi seperti panggilan telepon. Aplikasi tidak dapat menjalankan perintah apapun ataupun menampilkan apapun dalam state ini.

3. *Stopped*

Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan service di background.

State lain seperti Create and Started bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode life cycle callback berikutnya. Artinya, setelah sistem `onCreate()` dipanggil, dengan cepat sistem akan memanggil method `onStart()`, kemudian diikuti oleh `onResume()` [7].

2.4 Web Server

Web Server adalah sebuah perangkat lunak yang bertugas menerima permintaan client melalui port HTTP maupun HTTPS dan merubahnya ke dalam format HTML. Terdapat beberapa format selain HTML yaitu PHP atau ASP, tetapi

format – format tersebut hanyalah berfungsi untuk menghubungkan HTML dengan *database* [8].

Web server sebagai sebuah software aplikasi yang dapat teridentifikasi oleh URL dan memiliki interface yang didefinisikan, dideskripsikan, dan dimengerti oleh XML atau JSON dan juga mendukung interaksi langsung dengan software aplikasi yang lain dengan menggunakan *message* berbasis XML atau JSON melalui protokol.

2.5 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan Bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin.

Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Saat ini java Merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web. Berikut ini adalah contoh syntax dari Bahasa Pemograman Java.

```
public class HelloWorld {
    public static void main(String[]
args) {
    System.out.println("Hello,World");
    }
}
```

2.6 PHP

PHP adalah singkatan dari PHP: *Hypertext PreProcessors*, PHP saat ini adalah bahasa pemrograman interpreter yang paling banyak digunakan saat ini untuk pengembangan *web server*. PHP (*Hypertext Preprocessor*), merupakan

bahasa pemrograman pada sisi *server* yang memperbolehkan programmer menyisipkan perintah – perintah perangkat lunak *web server* (Apache, IIS, atau lain sebagainya) akan dieksekusi sebelum perintah itu dikirim oleh halaman ke browser yang me-*request*-nya. Sebagai bahasa pemrograman untuk tujuan umum, kode PHP diproses oleh aplikasi penerjemah dalam modus baris - baris perintah modus dan melakukan operasi yang diinginkan sesuai sistem operasi untuk menghasilkan keluaran program di channel *output* standar [9].

2.7 MySQL

MySQL adalah sebuah sistem manajemen database relasi (*relational database management system*) yang bersifat “terbuka” (*open source*). Terbuka maksudnya adalah MySQL boleh di-download oleh siapa saja, baik versi kode program aslinya maupun versi binernya dan dapat digunakan secara (relatif) gratis baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi computer [10].

2.7.1 Sejarah MySQL

Sejarah MySQL yang merupakan hasil buah pikiran dari Michael “Monty” Widenius, David Axmark, dan Alan Larson dimulai tahun 1995. Mereka bertiga kemudian mendirikan perusahaan bernama MySQL AB di Swedia. Tujuan awal ditulisnya program MySQL adalah untuk mengembangkan aplikasi web yang digunakan salah satu klien MySQL AB.

2.7.2 Penggunaan Database MySQL

Dalam penggunaan database MySQL, setiap perintah yang diketikkan disebut *query*. Perintah MySQL dapat dikategorikan menjadi 3 sub perintah, yaitu DDL (*Data Definition Language*), DML (*Data Manipulation Language*), dan DCL (*Data Control Language*).

2.7.2.1 DDL (Data Definition Language)

Perintah dalam *SQL* yang pertama adalah perintah DDL. DDL sendiri merupakan singkatan dari *Data Definition Language*. DDL dapat berarti sebuah perintah yang berhubungan dengan pendefinisian dari suatu struktur database. Terdapat beberapa perintah DDL pada MySQL sebagai berikut :

1. *CREATE* berfungsi untuk membuat database baru, tabel baru, view baru dan kolom.
2. *ALTER* berfungsi untuk mengubah struktur tabel. Seperti mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom maupun memberikan atribut pada kolom.
3. *DROP* berfungsi untuk menghapus database dan tabel.
4. *TRUNCATE* berfungsi untuk Menghapus semua catatan dari tabel.
5. *COMMENT* berfungsi untuk Menambahkan komentar pada data.
6. *RENAME* berfungsi untuk mengubah nama obyek.

Adapun contoh sintaks dari perintah DDL pada MySQL sebagai berikut :

```
CREATE DATABASE NILAI;
ALTER TABLE Mahasiswa ADD (NoTelp char(8));
DROP INDEX nama_index ;
TRUNCATE TABLE table_barang;
RENAME table_barang to table_barang_gudang;
```

Gambar 2. 4 Contoh Sintaks dari Perintah DDL pada MySQL

2.7.2.2 DML (Data Manipulation Language)

Data Manipulation Language (DML) ialah perintah yang digunakan untuk mengelola/memanipulasi data dalam database. Terdapat beberapa perintah *DML* pada MySQL sebagai berikut :

1. *SELECT* berfungsi untuk mengambil/menampilkan data dari database.
2. *INSERT* berfungsi untuk memasukkan data ke dalam tabel.
3. *UPDATE* berfungsi untuk memperbaiki data dalam tabel.
4. *DELETE* berfungsi untuk menghapus data dari tabel.
5. *CALL* berfungsi untuk memanggil subprogram PL / SQL atau Java.
6. *EXPLAIN PLAN* berfungsi untuk menjelaskan jalur akses ke data.
7. *LOCK TABLE* berfungsi untuk mengunci tabel.

Adapun contoh sintaks dari perintah DML pada MySQL sebagai berikut :

```

INSERT INTO mahasiswa VALUES("08052926", "Frenky","70");
SELECT nama_mahasiswa FROM mahasiswa;
DELETE FROM mahasiswa;
UPDATE mahasiswa SET nama_mahasiswa='ujang'
WHERE nim='08052926';

```

Gambar 2. 5 Contoh Sintaks dari Perintah DML pada MySQL

2.7.2.3 DCL (Data Control Language)

Data Control Language (DCL) ialah perintah yang digunakan untuk melakukan pengontrolan data dan server databasenya. Terdapat beberapa perintah *DCL* pada MySQL sebagai berikut :

1. *GRANT* berfungsi untuk memberikan hak akses pengguna ke database.
2. *REVOKE* berfungsi untuk menghilangkan hak akses yang telah diberikan dengan perintah *GRANT*.

Adapun contoh sintaks dari perintah DCL pada MySQL sebagai berikut :

```

GRANT all privileges on * to nm_user@localhost
identified by 'nm_passwd' with grand option;

REVOKE all on nm_db.nm_tbl from nm_user@localhost
identified by 'nm_passwd';

```

Gambar 2. 6 Contoh Sintaks dari Perintah DCL pada MySQL

2.8 GPS (Global Positioning System)

GPS adalah sebuah sistem navigasi berbasis radio yang menyediakan informasi berupa koordinat posisi, kecepatan dan waktu kepada pengguna dengan bantuan sinkronisasi satelit. Sistem ini didesain untuk memberikan posisi dan kecepatan tiga dimensi serta informasi mengenai waktu, secara kontinyu di seluruh dunia tanpa tergantung waktu dan cuaca, kepada banyak orang secara simultan. Sistem ini menggunakan satelit yang berfungsi sebagai pengirim sinyal yang berisi informasi koordinat lokasi, kecepatan, arah dan waktu pada alat penerima sinyal GPS (*receiver*) dipermukaan bumi [7].

2.8.1 Arsitektur GPS

Sistem ini menggunakan sejumlah satelit yang berada di orbit bumi, yang memancarkan sinyal ke bumi dan ditangkap oleh alat penerima di permukaan bumi. Arsitektur GPS terdiri dari tiga bagian yaitu *Space Segment*, *Control Segment* dan *User Segment*.

1. Space Segment

Space Segment adalah bagian yang terdiri dari kumpulan-kumpulan satelit diluar angkasa yang berada pada orbit bumi. Jarak antara satelit dengan permukaan bumi biasanya sekitar 20.000 km, diatas permukaan bumi. Satelit-satelit ini diatur sedemikian rupa sehingga GPS receiver menerima sedikitnya data dari 6 satelit yang berbeda, namun berada dalam jangkauan orbitnya. Sinyal satelit dapat melewati awan, kaca maupun plastik, tetapi tidak dapat melewati gedung dan gunung. Satelit mempunyai jam atom, dan juga akan memancarkan informasi berupa waktu/jam saat ini. Data ini dipancarkan dengan kode *pseudo-random*. Masing-masing satelit memiliki kodenya sendiri-sendiri. Nomor kode ini biasanya akan ditampilkan pada alat navigasi atau GPS receiver, maka dengan nomor kode tersebut pengguna dapat mengidentifikasi sinyal satelit yang sedang diterima alat tersebut. Pada umumnya, yang pertama lebih dikenal dengan sebutan L1 pada 1575.42 MHz. Sinyal L1 ini yang akan diterima oleh alat navigasi. Satelit juga mengeluarkan gelombang L2 pada frekuensi 1227.6 Mhz. Gelombang L2 ini digunakan untuk tujuan militer dan bukan untuk umum.

2. Control Segment

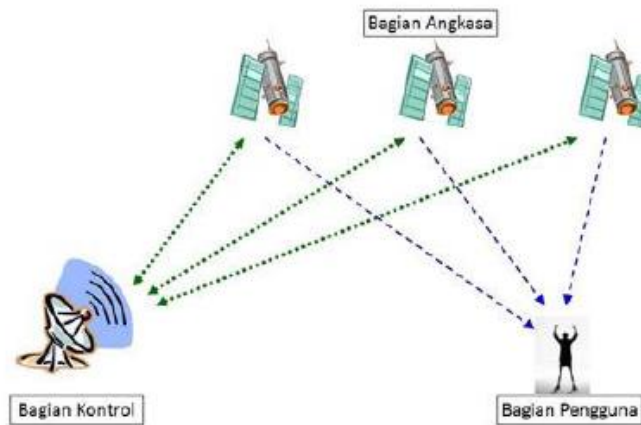
Segment sistem control GPS berfungsi mengontrol dan memantau operasional semua satelit GPS dan memastikan bahwa semua satelit berfungsi sebagai mana mestinya. Secara lebih spesifik tugas utama dari segment system control GPS adalah :

- a) Secara kontinyu memantau dan mengontrol sistem satelit.
- b) Menentukan dan menjaga waktu system GPS.

- c) Memprediksi empheris satelit serta karakteristik jam satelit.
- d) Secara periodik meremajakan (*update*) *navigation message* dari setiap satelit.
- e) Melakukan manuver satelit agar tetap berada dalam orbitnya, atau melakukan relokasi untuk menggantikan satelit yang tidak “sehat”, seandainya diperlukan.

3. User Segment

Segment user terdiri dari antenna dan *prosesor receiver* yang menyediakan *positioning*, kecepatan dan ketepatan waktu ke pengguna. Bagian ini menerima data dari satelit-satelit melalui sinyal radio yang dikirimkan setelah mengalami koreksi oleh stasiun pengendali GPS (*GPS Control Segment*).



Sumber : nusantride.com

Gambar 2. 7 Cara Kerja GPS

2.9 LBS (Location Based Service)

Location Based Service merupakan layanan informasi yang dapat diakses menggunakan *mobile devices*, yang dilengkapi kemampuan untuk mengetahui keberadaan lokasi dari si pengguna perangkat dan kemampuan memberikan informasi mengenai layanan yang tersedia berdasarkan lokasi mereka pada saat itu. Menurut Schiller J, *Location Based Service* dapat didefinisikan sebagai “layanan yang mengintegrasikan lokasi perangkat mobile atau posisi dengan informasi lain sehingga dapat memberikan nilai tambah bagi pengguna” [11].

Layanan berbasis lokasi dapat diklasifikasikan ke dalam empat jenis yaitu *Maps and Navigation*, *Tracking Services*, *Information Services* dan *Application* seperti yang digambarkan dalam gambar berikut :



Sumber : <http://www.telecomcircle.com/2009/06/introduction-to-lbs/>

Gambar 2. 8 Klasifikasi LBS

2.9.1 Latitude (Garis Lintang)

Latitude adalah garis yang melintang di antara kutub utara dan kutub selatan yang menghubungkan sisi bagian bumi dari timur dan barat. Garis lintang memiliki posisi membentangi bumi seperti halnya garis khatulistiwa tetapi dengan kondisi nilai tertentu. Latitude dibedakan menjadi dua wilayah yaitu lintang utara dan selatan dimana nilai koordinat di bagian utara selalu positif dan koordinat selatan selalu negatif [12].

2.9.2 Longitude (Garis Bujur)

Longitude adalah garis bujur yang menghubungkan antara sisi utara dan sisi selatan permukaan bumi. Garis bujur digunakan untuk mengukur sisi barat dan timur koordinat suatu titik di belahan bumi. Longitude dibedakan menjadi dua wilayah yaitu bujur timur dan bujur barat dimana nilai koordinat yang berada di timur selalu positif dan nilai koordinat dibagian barat selalu negatif [12].

2.10 MapBox

MapBox Service adalah penyedia layanan pemetaan seperti halnya Google Maps, untuk data pemetaan sendiri MapBox menggunakan *base tiles* dari OpenStreetMaps dan juga dari beberapa sumber lainnya seperti NASA,

DigitalGlobe, dan USGS. MapBox sendiri *default*-nya menyediakan 4 jenis tipe maps, yaitu *MapBox Street, Emerald, Light and Dark*, dan *Satellite*.

2.10.1 Cara Mendapatkan API key MapBox

Berikut langkah-langkah mendapatkan *Key* API MapBox :

1. Masuk ke halaman <https://mapbox.com>
2. Klik **Get started for free** dan membuat akun baru dengan mendaftarkan email dan *password* Anda.
3. Klik **Continue**, untuk mengaktifkan MapBox
4. Verifikasi akan dikirim ke alamat email Anda.
5. Klik dapatkan akses token. Jika Anda sudah memiliki token, Anda dapat menggunakan token tersebut.

MapBox API adalah sebuah webservice yang menyediakan informasi tentang geocoding dan directions dari dua buah titik node awal ke titik node tujuan. Mapbox digunakan pada Aplikasi ini untuk menunjukkan peta lokasi tempat wisata dan pengguna, berikut ini cara-cara mengakses Mapbox pada Android Studio. Langkah-langkah dapat dilihat sebagai berikut :

1. Dependencies & Access Token

Akses token :

```
<com.mapbox.mapboxsdk.views.MapView
    Android : id="@+id/mapview"
    Android : layout_widht="match_parent"
    Android : layout_height="match_parent"
    Mapbox : access_token="
    ANd9GcRZmQN9d67QKqLpskv1ThV11A-
    AYIfWtkel3xKdmk3409Z2X 87"/>
```

Untuk membuat aplikasi pemetaan menggunakan MapBox, pertama yang harus dilakukan adalah mendaftar terlebih dahulu pada <https://mapbox.com/> untuk mendapatkan akses Token, yang nantinya digunakan sebagai *API Key* pada aplikasi yang dibangun. Jika sudah, maka membuat project baru di

Android Studio dan memasukkan *dependencies* seperti berikut pada file *build-gradle* :

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.0.1'
    compile 'com.android.support:design:23.0.1'
    compile ('com.mapbox.mapboxsdk:mapbox-android-
sdk:2.2.0@aar'){
        transitive=true
    }
}
```

2. App Permissions

Setelah itu, MapBox menggunakan beberapa permission yang harus ditambah pada Android Manifest di aplikasi yang dibangun.

Permission INTERNET dan ACCESS_NETWORK_STATE digunakan untuk menampilkan peta digital secara online. Kemudian permission LOCATION digunakan apabila ingin mendeteksi lokasi pengguna.

```
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"
/>
<uses-permission
android:name="android.permission.INTERNET"/>
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"
/>
<uses-permission
android:name="android.permission.FINE_LOCATION"/>
```

3. Setting Up Maps

Untuk men-setting maps pada main layout pastikan menambahkan elemen MapView dari MapBox, disertai dengan access token yang didapat :

```

Main Activity Setup Maps :

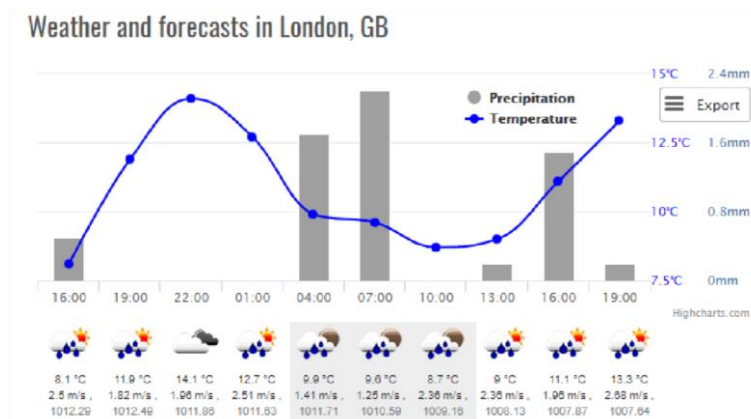
private void setMap(Bundle savedInstanceState) {
    mapView = (MapView) findViewById(R.id.mapview);
    /* set map style */
    mapView.setStyleUrl(Style.EMERALD);
    /* set user location */
    mapView.setMyLocationEnabled(true);

    mapView.setMyLocationTrackingMode(MyLocationTracking
        .TRACKING_NONE);
    /* animate to center */
    mapView.setCenterCoordinate(new LatLngZoom(-
        10.1296736, 123.6707237, 10f), true);
    /* custom marker icon */
    SpriteFactory spriteFactory = new
    SpriteFactory(mapView);
    Drawable drawable =
    ContextCompat.getDrawable(this,
    R.drawable.ic_marker_strip);
    Sprite icon =
    spriteFactory.fromDrawable(drawable);
    mapView.addMarker(new MarkerOptions()
        .position(new LatLng(-10.1296736,
        123.6707237))
        .title("To Destination")
        .icon(icon));
    /* standard icon */
    mapView.addMarker(new MarkerOptions()
        .position(new LatLng(-10.1296736,
        123.6707237))
        .title("Tempat Wisata")
        .snippet("Pantai Lasiana"));
    mapView.onCreate(savedInstanceState);
}

```

2.11 OpenWeatherMaps

OpenWeatherMap adalah layanan penyedia data cuaca, termasuk data prakiraan dan data historis terkini untuk para pengembang layanan web dan aplikasi *mobile*. Untuk sumber data, OpenWeatherMap menggunakan layanan siaran *meteorologi*, data mentah dari stasiun cuaca bandara, data mentah dari stasiun radar, dan data mentah dari stasiun cuaca resmi lainnya.



Sumber gambar: <https://openweathermap.org/>

Gambar 2. 9 Contoh Data Cuaca dari OpenWeather

2.11.1.1 Cara Mendapatkan API Key OpenWeather

Berikut langkah-langkah cara mendapatkan *Key API* OpenWeatherMap :

1. Mendapatkan *API key* dengan cara mendaftarkan akun pada alamat *website* : https://home.openweathermap.org/users/sign_up

Sumber Gambar : https://home.openweathermap.org/users/sign_up

Gambar 2. 10 Halaman Daftar Akun

2. Aktivasi API *key* untuk versi gratis dan membutuhkan waktu 10 menit.
3. Setelah mendapatkan API *key* lakukan pemanggilan melalui *link* :
<http://api.openweathermap.org/data/2.5/forecast?id=524901&APPID={APIKEY}> adalah API *key* unik yang telah didapatkan.

Adapun cara kerja system yang akan dibangun adalah sebagai berikut :

1. Pengguna mendaftar ke situs resmi Openweather API untuk mendapatkan *Key*, dimana *Key* ini berfungsi sebagai izin untuk mengakses layanan Openweather.
2. Pengguna mempersiapkan pembangunan aplikasi dengan menggunakan Android Studio sebagai *software* pembangunan template yang telah disediakan dan dapat dimodifikasi penggunaanya.
3. System akan memeriksa kesamaan *Key* yang terdapat pada aplikasi dan *Key* dari Openweather, jika *Key* berbeda akan mendampirkan pesan error.
4. Jika *Key* sesuai akan menampilkan peta digital dan pengguna dapat menggunakan aplikasi.

2.11.1.2 Cara Kerja Open Weather

Cara Kerja Openweather pada Aplikasi yang akan dibangun digunakan untuk memprediksi perkiraan cuaca 3-5 hari kedepan. Untuk menggunakan bootstrap pada project yang dibangun dapat menggunakan link CDN dengan catatan server atau komputer yang digunakan harus terkoneksi ke internet.

Sisipkan kode berikut pada index.blade.php dibagian <head> :

```
<link
rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/boo
tstrap.min.css" integrity="sha384-
9gVQ4dYFwwWSjIDZnLEWnxCjeSWFphJiwGPXr1jddIhOegiulFwO5qRGvFXOdJZ4
" crossorigin="anonymous">
```

Instalasi Icon Weather

Sama seperti bootstrap, penggunaan icon weather menggunakan CDN dengan menyisipkan kode berikut pada bagian head index.blade.php :


```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/weather-
icons/2.0.9/css/weather-icons.min.css">
```

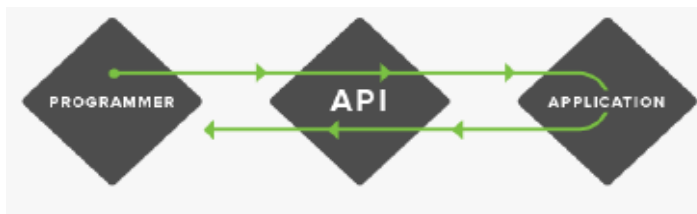
JSON Format

```
<?xml version="1.0" encoding="UTF-8" ?>
<city>
  <id>2057087</id>
  <name>Kupang</name>
  <coord>
    <lon>123.5833</lon>
    <lat>-10.1667</lat>
  </coord>
  <country>ID</country>
  <population>282396</population> </city>
<cod>200</cod>
<message>11.447274</message>
<cnt>7</cnt> <list>
  <dt>1534388400</dt>    <temp>
    <day>299.15</day>
    <min>297.63</min>
    <max>299.15</max>
    <night>297.71</night>
    <eve>299.15</eve>
    <morn>299.15</morn> </temp>
  <pressure>1012.58</pressure>
  <humidity>84</humidity>
  <weather>
    <id>500</id>
    <main>Rain</main>
    <description>light rain</description>
    <icon>10d</icon> </weather>
  <speed>4.95</speed>
  <deg>115</deg>
  <clouds>64</clouds>
</list>
<list>
```

2.12 API (Application Programming Interface)

API adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal. Pada konteks web, API merupakan pemanggilan fungsi lewat *Hyper Text Transfer Protocol* (HTTP) dan mendapatkan respon berupa *Extensible Markup Language* (XML) atau *JavaScript Object Notation* (JSON). Pemanggilan fungsi ke suatu situs tertentu akan menghasilkan respon yang berbeda kepada pengguna untuk membangun aplikasi *enterprise* di dalam *website*-nya [13].

Dengan API, pemanggilan yang bolak-balik antar aplikasi diatur melalui web service. Web service adalah kumpulan standar teknis dan protokol, termasuk XML (*Extensible Markup Language*), bahasa umum yang digunakan oleh aplikasi-aplikasi tersebut selama berkomunikasi di internet. API dan *web service* sepenuhnya bekerja di belakang layar. Dengan demikian, API adalah standar komunikasi yang dibuka oleh perusahaan software, agar dapat dimanfaatkan oleh pengembang pihak ketiga untuk mendesain aplikasi yang memanfaatkan layanan mereka dengan mudah [13].

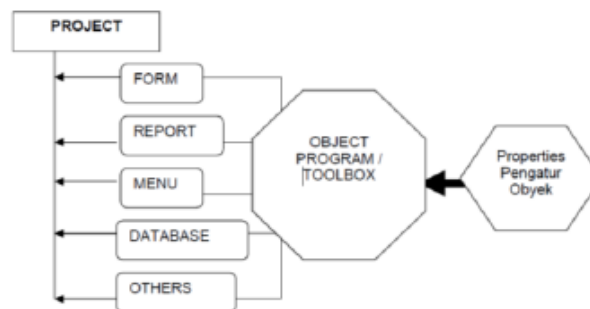


Sumber : *sproutsocial.com*

Gambar 2. 11 Ilustrasi API

2.13 OOP (Objek Oriented Programming)

Pemrograman berorientasi objek atau *object oriented programming* merupakan suatu pendekatan pemrograman yang menggunakan *object* dan *class*. Dalam OOP, setiap bagian dari program adalah *object*. Sebuah *object* mewakili suatu bagian program yang akan diselesaikan [14].



Sumber : E-journal Infokam (2013) [14]

Gambar 2. 12 Model Pemrograman OOP

2.14 UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah bahasa yang digunakan untuk menspesifikasikan, memvisualisasikan, dan mendokumentasikan artifak dari sebuah sistem berorientasi objek yang dalam tahap pengembangan. UML menyediakan standar defacto dalam analisis sistem berorientasi objek dan desain yang didasarkan dari pengalaman pengguna [15]. Dalam membangun block UML ada 3 hal yang harus diperhatikan, yaitu *object* (memodelkan konsep), *relationship* (mengkoneksikan *object*), dan *diagram grouping* (saling mengkoneksikan antara *object* dan *relationship*).

Karena aplikasi yang akan dibangun ini beorientasi obyek maka untuk perancangan konsep pembangunan aplikasi ini menggunakan pendekatan sistem OOP (*Object Oriented Programming*) yang divisualisasikan dengan UML. Didalam UML memiliki beberapa jenis diagam diantaranya adalah *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Sequence Diagram* sebagai berikut :

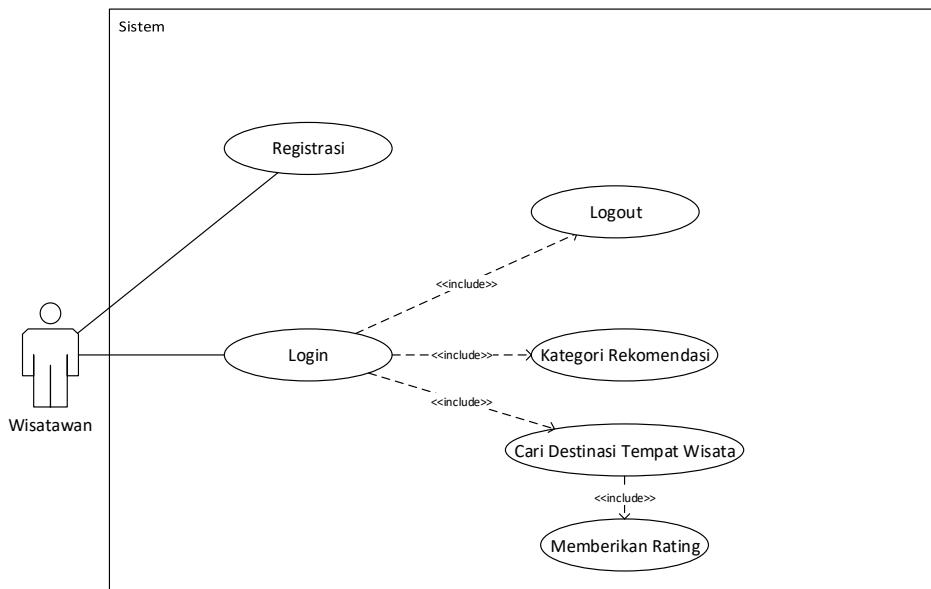
1. Use Case Diagram

Use Case Diagram adalah gambaran graphical dari beberapa atau semua actor, use case, dan interaksi diantaranya yang memperkenalkan suatu sistem. Use case diagram tidak menjelaskan secara detail tentang penggunaan use case, tetapi hanya memberi gambaran singkat hubungan antara usecase, aktor, dan sistem.

Sebuah *use case* dapat di *include* oleh lebih dari suatu *use case* lain, shingga duplikasi fungsionalitas dapat dihindari dengan menarik keluar

fungsionalitas yang *common*. Use case juga dapat meng *extend use case* lain dengan behaviornya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa *use case* yang satu merupakan spesialis dari yang lain. Diagram *use case* memiliki beberapa *element* yang digunakan untuk menggambarkan rancangan dan penghubung antar komponen, baik *actor* dengan *use case* atau *use case* dengan *use case* yang lainnya [15]. Untuk lebih jelasnya berikut ini adalah contoh *Use Case Diagram*.

Berikut ini adalah contoh gambar perancangan *Use Case Diagram* :



Gambar 2. 13 Contoh Use Case Diagram

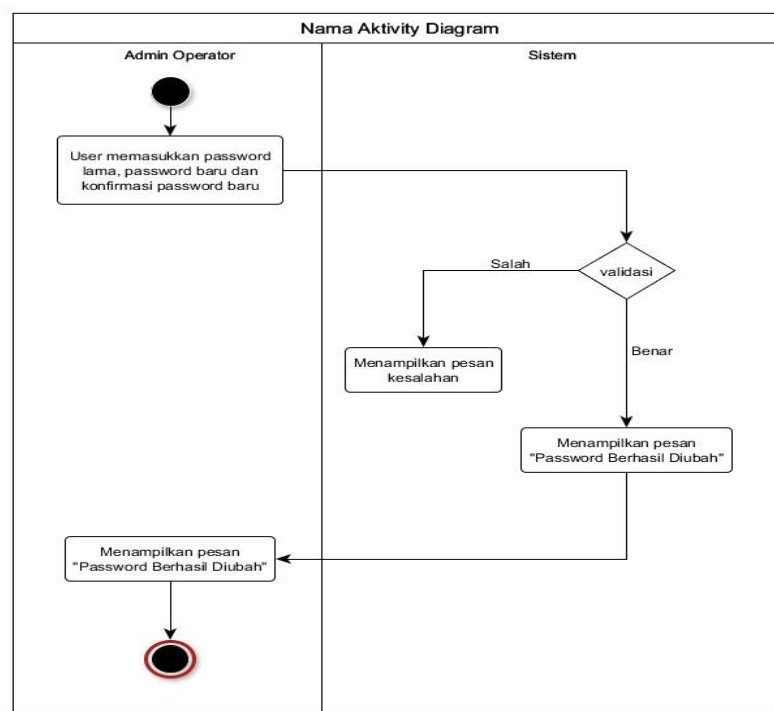
2. Activity Diagram

Activity Diagram atau Diagram Aktivitas digunakan untuk mendokumentasikan alur kerja pada sebuah sistem yang akan dibangun, yang dimulai dari pandangan *business level* hingga ke operasional *level*, adapun penjelasannya antara lain :

1. *Activity diagram* merupakan cara lain untuk memodelkan aliran suatu kejadian.
2. Berfungsi untuk menggambarkan *workflow* atau aliran kerja proses bisnis.

3. *Activity diagram* menggambarkan berbagai aliran aktivitas dalam sebuah sistem yang sedang dirancang bagaimana masing aliran berawal, decision yang mungkin terjadi, dan bagaimana alur berakhir.
4. *Activity diagram* dapat menggambarkan proses *parallel* yang mungkin dapat terjadi pada beberapa eksekusi [15].

Berikut ini adalah contoh Gambar dari *Activity Diagram* :

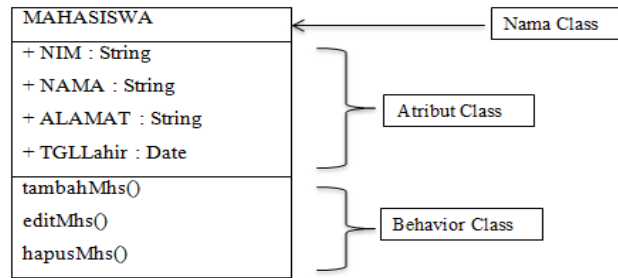


Gambar 2. 14 Contoh Activity Diagram

3. Class Diagram

Class Diagram merupakan elemen-elemen terpenting dalam sistem berorientasi objek, kelas mendeskripsikan satu blok pembangun sistem. *Class Diagram* memiliki fitur yang memodelkan multiplisitas, ketampakan, penanda, *polymorphism*, dan karakter-karakter lainnya. UML menggunakan istilah fitur sebagai istilah umum yang meliputi *property* dan operasi sebuah kelas Terdapat empat bagian masing-masing bagan kelas, yaitu : nama, atribut, perilaku, dan keterangan kelas.

Berikut ini adalah contoh gambar perancangan *Class Diagram*.

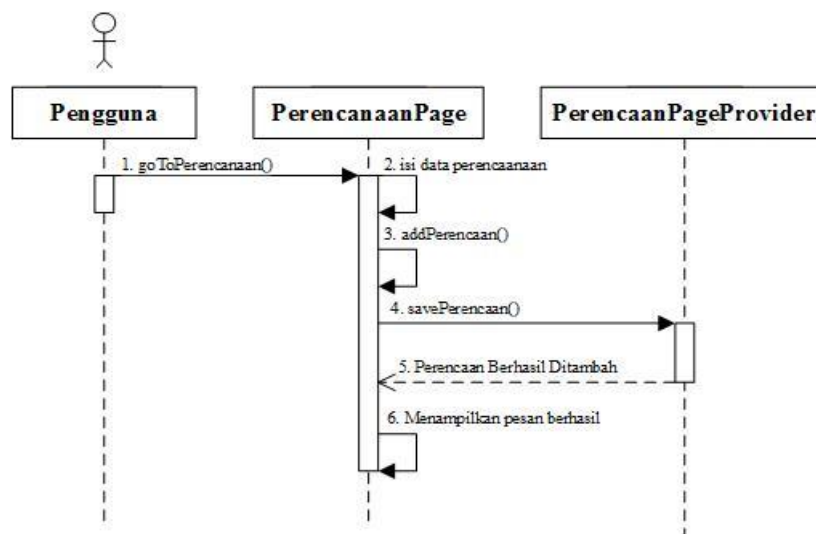


Gambar 2. 15 Contoh Class Diagram

4. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan Output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *Output* apa yang dihasilkan.

Masing-masing *object*, termasuk *actor*, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message [15]. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk *object boundary*, *controller* dan *persistent entity*.



Gambar 2. 16 Contoh Sequence Diagram

2.15 JSON (JavaScript Object Notation)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [16].

JSON terdiri dari dua struktur yaitu :

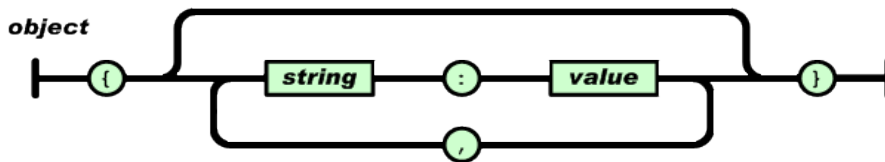
1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman modern mendukung struktur data dalam bentuk yang sama maupun berlainan [16].

Hal ini disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini. JSON menggunakan bentuk sebagai berikut :

1. Objek

Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).

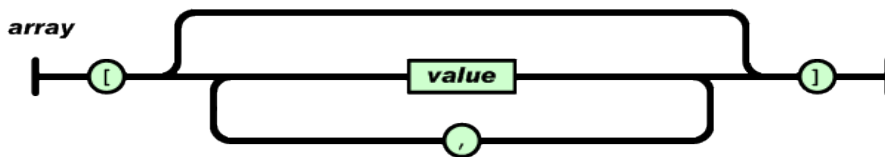


Sumber : <http://www.json.org/json-id.html>

Gambar 2. 17 Bentuk Objek dari JSON

2. Array

Array adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).

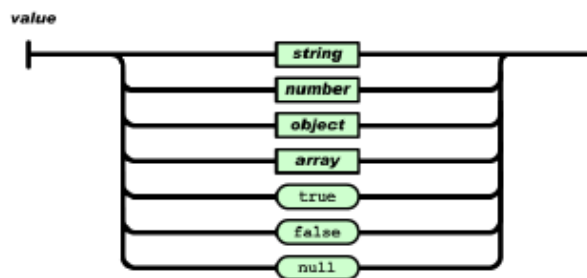


Sumber : <http://www.json.org/json-id.html>

Gambar 2. 18 Bentuk Array dari JSON

3. Value

Value dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau **true** atau **false** atau **null**, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



Sumber : <http://www.json.org/json-id.html>

Gambar 2. 19 Bentuk Value dari JSON

2.16 Kamus Data

Kamus Data merupakan fakta tentang data dan kebutuhan-kebutuhan informasi dari sistem informasi. Dengan menggunakan data dictionary, analisis sistem dapat mendefinisikan data yang mengalir dalam sistem dengan lengkap. Contoh dari kamus data dapat dilihat pada Tabel 2.4 Contoh Kamus Data.

Tabel 2. 4 Contoh Kamus Data

Nama	Data Nama dari data yang digunakan
Arus Data	Daftar pada poin proses apa data digunakan
Keterangan	Uraian singkat dari data yang digunakan
Struktur Data	Daftar komponen data yang ada pada data yang digunakan
Deskripsi	Jenis data dalam representasi komputer untuk masing-masing data.

2.17 Kuesioner Skala Likert

Skala Likert adalah sebuah tipe skala psikometri yang menggunakan angket dan menggunakan skala yang lebih luas dalam penelitian survei. Metode rating yang dijumlahkan (*summated rating*) populer juga dengan nama penskalaan Model Likert. Dalam pendekatan ini tidak diperlukan adanya kelompok panel penilai (*Judging Group*) dikarenakan nilai skala setiap pernyataan tidak akan ditentukan oleh derajat favorabelnya masing-masing, akan tetapi ditentukan oleh distribusi respons setuju atau tidak setuju dari sekelompok responden yang bertindak sebagai kelompok uji coba [17].

Dalam skala Likert terdapat lima kategori yang digunakan. Kelima kategori tersebut masuk ke dalam pernyataan positif dan negatif seperti tabel 2.5 berikut :

Tabel 2. 5 Lima Kategori Skala Likert

NO	Pernyataan Positive	Pernyataan Negative
1	Sangat Tidak Setuju	Sangat Setuju
2	Tidak Setuju	Setuju
3	Ragu-ragu	Ragu-ragu
4	Setuju	Tidak Setuju
5	Sangat Setuju	Sangat Tidak Setuju

2.18 Metode Pengujian

Metode pengujian adalah cara atau teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan. Pengujian perangkat lunak adalah proses untuk mencari kesalahan pada setiap *item* perangkat lunak, mencatat hasilnya, mengevaluasi setiap aspek pada setiap komponen (*system*) dan mengevaluasi aktivitas-aktivitas dari perangkat lunak [18]. Perangkat lunak dapat diuji dengan dua cara, yaitu sebagai berikut :

1. Pengujian *Black-Box*

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk program.

Tahap Pengujian *BlackBox (Fungsional)*

- 1) Fungsi-fungsi yang salah atau hilang.
- 2) Kesalahan *interface*.
- 3) Kesalahan struktur data atau akses database eksternal.
- 4) Kesalahan performa.
- 5) Kesalahan inisialisasi atau terminasi.

Teknik pengujian *black-box* terdiri dari 10 jenis diantaranya *Equivalence Partitioning, Boundary Value Analysis/Limit Testing, Comparison Testing, Sample Testing, Robustness Testing, Behavior Testing, Requirement Testing, Performance*

Testing, Endurance Testing, Cause-Effect Relationship Testing. Salah satunya yang akan dibahas adalah *Equivalence partitioning*.

Equivalence partitioning merupakan metode ujicoba *black-box* yang membagi domain input dari program menjadi beberapa kelas data dari kasus uji coba yang dihasilkan. Kasus uji penanganan single yang ideal menemukan sejumlah kesalahan (misalnya : kesalahan pemrosesan dari seluruh data karakter) yang merupakan syarat lain dari suatu kasus yang dieksekusi sebelum kesalahan umum diamati.

