



```
    "Name": "Abdominal guarding"
  },
  {
    "ID": 238,
    "Name": "Anxiety"
  },
  {
    "ID": 974,
    "Name": "Aggressiveness"
  },
  {
    "ID": 54,
    "Name": "Reduced appetite"
  },
  {
    "ID": 131,
    "Name": "Increased appetite"
  },
  {
    "ID": 250,
    "Name": "Breathing-related pains"
  },
  ...
]
```



```

    "Icd" : "A04.0",
    "IcdName" : "Epistaxis",
    "Accuracy": 90
  },
  "Specialisation":[
    {
      "ID": 15,
      "Name": "General practice",
      "SpecialistID": 3
    },
    {
      "ID": 32,
      "Name": "Otolaryngology",

```

```

      "SpecialistID": 49
    }
  ]
},
{
  "Issue":{
    "ID": 649,
    "Name": "Lung bleeding",
    "ProfName": "Hemoptysis"
    "Icd" : "A04.1",
    "IcdName" : "Haemoptysis",
    "Accuracy": 18
  }
}

```

```
    },  
    "Specialisation": [  
      {  
        "ID": 15,  
        "Name": "General practice",  
        "SpecialistID": 3  
      },  
      {  
        "ID": 19,  
        "Name": "Internal medicine",  
        "SpecialistID": 4  
      }  
    ]  
  }  
]
```

## 2.2 Aplikasi

Aplikasi dapat diartikan sebagai suatu program berbentuk perangkat lunak yang berjalan pada suatu sistem tertentu yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia. Selain pengertian tersebut ada banyak pengertian dari para ahli mengenai kata aplikasi adalah sebagai berikut:

### a. Ali Zaki dan Smitdev Community

Menurut Ali Zaki dan *Smitdev Community*, Aplikasi merupakan komponen yang bermanfaat sebagai media untuk menjalankan pengolahan data ataupun

berbagai kegiatan lainnya seperti pembuatan ataupun pengolahan dokumen dan file.

b. Sri Widianti

Menurut Sri Widianti, Aplikasi merupakan sebuah *software* (perangkat lunak) yang bertugas sebagai *front end* pada sebuah sistem yang dipakai untuk mengelolah berbagai macam data sehingga menjadi sebuah informasi yang bermanfaat untuk penggunaanya dan juga sistem yang berkaitan.

c. Harip Santoso

Menurut Harip Santoso, Aplikasi merupakan sebuah kelompok file (*class, form, report*) yang ditujukan sebagai pengeksekusi aktivitas tertentu yang saling berkaitan seperti contohnya aplikasi *payroll* dan aplikasi *fixed asset*.

d. Yuhefizer

Menurut Yuhefizar, Aplikasi adalah program yang sengaja dibuat dan dikembangkan sebagai pemenuh kebutuhan penggunaanya dalam menjalankan suatu pekerjaan tertentu.

e. Hengky W.Pranama

Menurut Hengky W. Pramana, pengertian aplikasi adalah satu unit perangkat lunak yang sengaja dibuat untuk memenuhi kebutuhan akan berbagai aktivitas ataupun pekerjaan, seperti aktivitas perniagaan, periklanan, pelayanan masyarakat, game, dan berbagai aktivitas lainnya yang dilakukan oleh manusia. [3]

### 2.3 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak bergerak layar seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android.inc, dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open*

*Handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler.

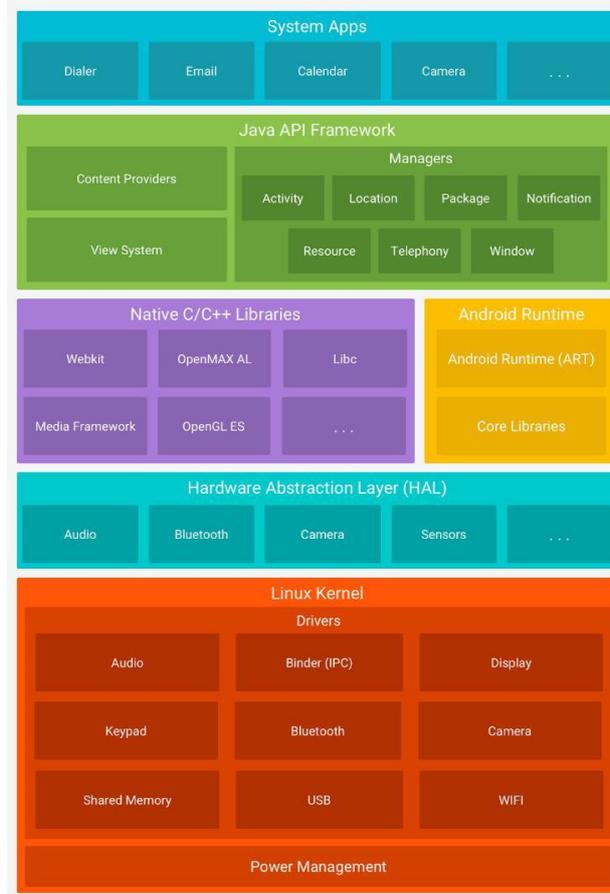
Android merupakan sistem operasi yang terbuka (*open source*) yang mana berarti jika pihak Google memperbolehkan dan membebaskan bagi pihakmanapun untuk dapat mengembangkan sistem operasi tersebut. Bahkan anda sendiripun juga dapat mengemangkan sistem android yang memang sesuai dengan keinginn anda. Sistem android memiliki gudang aplikasi dn game yaitu Google Playstore, yang mana disini bisa mendownload serta menggunakan aplikasi atau game yang terdapat di Google Playstore sepuasnya dengan menggunakan perangkat seluler dengan sistem Android. [1]

Di dunia terdapt dua jenis distributor sistem android. Pertama yang mendapat dukungan penuh dari Google dan Google Mail Service (GMS), seperti yang terdapat pada *smartphone* Nexus keluaran google yang bekerja sama dengan vendor *smartphone* HTC dan kedua adaah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD), jenis yang kedua ini banyak terdapat pada *smartphone* android di pasaran. Android sangat berkembang mengikuti perkembangan zaman, berikut merupakan versi-versi yang terdapat pada android:

**Tabel 2.3 Tabel Versi Android**

Nama Versi	Versi	Tahun rilis
No Codename	1.0	23 Sep-08
Petit Four	1.1	9 February 2009
Cupcake	1.5	27 Apr-09
Donut	1.6	15 Sep-09
Eclair	2.0	26 Okt-09
Froyo	2.2	20 Mei-10
Gingerbread	2.3	06 Des-10
Honeycomb	3.0	22 Feb-11
Ice Cream Sandwich	4.0	18 Okt-11
Jelly Bean	4.1	09 Jul-12
Kitkat	4.4	31 Okt-13
Lolipop	5.0	12 Nov-14
Marshmallow	6.0	05 Okt-15
Nougat	7.0	22 Agu-16
Oreo	8.0	21 Agu-17
Pie	9.0	6 Agu 2018
Q	10.0	7 Agu 2019

Selain itu Sistem Operasi Android memiliki beberapa komponen utama yang disebut dengan Arsitektur Platform Android. Berikut adalah diagram komponen-komponen utama dari platform Android:



**Gambar 2-1** Arsitektur Komponen Android

Berikut ini adalah penjelasan dari setiap komponen utama dari Arsitektur pada Komponen Android pada Gambar 2.2:

1. System Apps

System Apps sendiri adalah aplikasi Default yang sudah ada pada sistem operasi Android kita. Aplikasi ini biasanya adalah Dialer yang biasa kita gunakan untuk menelfon, SMS yang kita gunakan untuk mengirim pesan teks, Mail untuk mengirim dan menerima E-mail. Aplikasi ini biasanya tidak bisa di uninstal tanpa aplikasi bantuan dan setiap System Apps akan berbeda tergantung vendor yang memunculkannya.

2. Java API Framework

Java API Framework adalah kumpulan Library pada Java yang digunakan untuk mengembangkan aplikasi Android. dalam Java API Framework

terdapat 3 komponen utama yaitu Content Providers, View System, dan Managers. Pada Managers sendiri memiliki sub komponen sendiri yaitu Activity, Location, Package, Notification, Resource, Telephony, Window.

### 3. Native C/C++ Libraries

Ini adalah support yang diberikan oleh Android untuk para developer yang menggunakan bahasa C dan C++ untuk mengembangkan aplikasinya. Beberapa library yang dapat kita kembangkan adalah OpenGL ES, Webkit, dll.

### 4. Android Runtime (ART)

Untuk Android dengan versi 5.0 keatas sudah memiliki komponen Android Runtime, ART akan menggantikan Dalvik Virtual Machine (VM). Dengan menggunakan ART proses pada start up akan lebih cepat dibandingkan Dalvik VM. Fitur dari ART sendiri adalah :

- a. Kompilasi Ahead-of-Time (AOT) dan Just-in-Time (JIT)
- b. Optimasi Garbage Collection (GC)
- c. Support Debugging yang lebih bagus

### 4. Hardware Abstraction Layer (HAL)

HAL akan menyediakan interface untuk mengekspos kemampuan hardware ke tingkat Java API Framework yang lebih tinggi. HAL terdiri dari beberapa modul library yang mengimplementasikan interface setiap hardware yang berbeda misal kamera atau bluetooth atau yang lainnya.

### 5. Linux Kernel

Fondasi platform Android adalah kernel Linux. Contohnya adalah Android Runtime (ART) yang bergantung pada kernel Linux untuk fungsional dasar seperti threading dan manajemen memori tingkat rendah.

## 2.4 Web Service

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (service)

yang disediakan oleh suatu sistem yang menyediakan web service. Web service menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler.

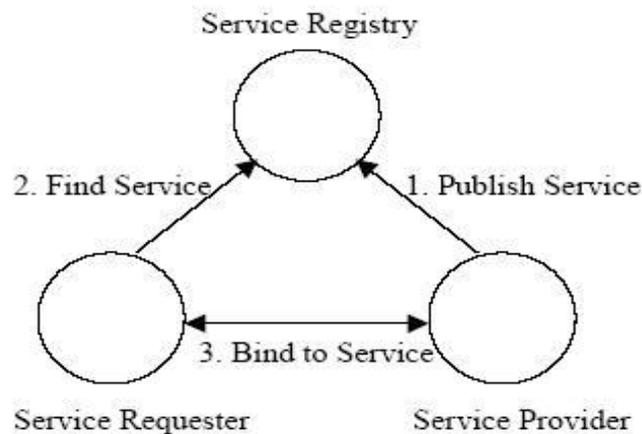
Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam Web Service dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya. [14]Beberapa alasan mengapa digunakannya web service adalah sebagai berikut:

1. Web service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. Web service memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. Web service cukup diupload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. Web service berjalan di port 80 yang merupakan protokol standar HTTP,dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall.

#### **2.4.1 Arsitektur Web Service**

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

1. Service Provider: Berfungsi untuk menyediakan layanan/service dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia.
2. Service Registry: Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/service yang telah di-register.
3. Service Requestor: Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.



**Gambar 2-2 Arsitektur Web Service**

## 2.5 Global Positioning System (GPS)

GPS merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunanya dimana lokasinya berada (secara global) di permukaan bumi yang berbasis satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital. Dimanapun pengguna tersebut berada, maka GPS bisa membantu menunjukkan arah. Layanan GPS ini tersedia secara gratis. Awalnya gps hanya digunakan untuk kepentingan militer, tapi pada tahun 1980-an dapat digunakan untuk kepentingan sipil. GPS dapat digunakan dimanapun juga dalam 24 jam, posisi unit GPS akan ditentukan berdasarkan titik-titik koordinat latitude dan longitude. [4]

### 2.5.1 Cara Kerja GPS

Bagian yang paling penting dalam sistem navigasi GPS adalah beberapa satelit yang berada di orbit bumi atau yang sering kita sebut di ruang angkasa. Satelit GPS saat ini berjumlah 24 unit yang semuanya dapat memancarkan sinyal ke bumi yang lalu dapat ditangkap oleh alat penerima sinyal tersebut atau GPS Tracker. Selain satelit terdapat 2 sistem lain yang saling berhubungan, sehingga jadilah 3 bagian penting dalam sistem GPS. Ketiga bagian tersebut terdiri dari: GPS Control Segment (Bagian Kontrol), GPS Space Segment (bagian angkasa), dan GPS User Segment (bagian pengguna).

- GPS Control Segment

Control segment GPS terdiri dari lima stasiun yang berada di pangkalan Falcon Air Force, Colorado Springs, Ascension Island, Hawaii, Diego Garcia dan Kwajalein. Kelima stasiun ini adalah mata dan telinga bagi [GPS](#). Sinyal-sinyal dari satelit diterima oleh bagian kontrol, kemudian dikoreksi, dan dikirimkan kembali ke satelit. Data koreksi lokasi yang tepat dari satelit ini disebut data ephemeris, yang kemudian nantinya dikirimkan ke alat navigasi yang dimiliki.

- GPS Space Segment

Space Segment adalah terdiri dari sebuah jaringan satelit yang terdiri dari beberapa satelit yang berada pada orbit lingkaran yang terdekat dengan tinggi nominal sekitar 20.183 km di atas permukaan bumi. Sinyal yang dipancarkan oleh seluruh satelit tersebut dapat menembus awan, plastik dan kaca, namun tidak bisa menembus benda padat seperti tembok dan rapatnya pepohonan. Terdapat 2 jenis gelombang yang hingga saat ini digunakan sebagai alat navigasi berbasis satelit. Masing-masingnya adalah gelombang L1 dan L2, dimana L1 berjalan pada frekuensi 1575.42 MHz yang bisa digunakan oleh masyarakat umum, dan L2 berjalan pada frekuensi 1227.6 Mhz dimana jenis ini hanya untuk kebutuhan militer saja.

- GPS User Segment

User segment 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 Control Segment).

## 2.6 Location Based Services (LBS)

*Location based services* adalah layanan berbasis lokasi atau istilah umum yang sering digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang pengguna gunakan. Layanan ini menggunakan teknologi *global positioning service* (GPS) dan *cell-based location* dari Google. Selain itu, LBS tersebut terdiri dari beberapa komponen di antaranya *mobile devices*, *communication network*, *position component*, dan

*service and content provider. Mobile devices* merupakan komponen yang sangat penting. Piranti *mobile* tersebut diantaranya adalah *smartphone*, PDA, dan lainnya yang dapat berfungsi sebagai alat navigasi atau seperti halnya alat navigasi berbasis GPS.

### **2.6.1 Cara Kerja LBS**

LBS apps biasanya ketika pertama dibuka langsung akan mencari tahu lokasi user saat ini. Setelah Lokasi diketahui maka melalui internet aplikasi akan melakukan request ke Google Maps sebagai penyedia API. Request yang dilakukan berupa request peta berdasarkan lokasi user saat ini, dan ketika request itu berhasil diterima, maka Google Maps akan segera merespon aplikasi dengan memberikan akses peta kepada aplikasi. Selain itu Ketika lokasi berhasil diketahui, maka melalui internet juga aplikasi akan melakukan request ke web service yang kita buat sendiri. Web Service yang kita buat berguna untuk mengambil data pada database, request dari aplikasi tadi diubah menjadi query oleh web service.

Ketika Database yang berbentuk SQL ini berhasil menerima lalu memproses query, maka database akan memberikan data SQL sesuai dengan yang diminta oleh Web Service. Sekali lagi Web Service mengubah data SQL yang diterimanya menjadi JSON format yang akan lebih mudah dimengerti oleh aplikasi *smartphone*. Dan ketika JSON yang dikirimkan web service diterima oleh aplikasi, maka aplikasi akan menampilkan data sesuai dengan lokasi user berada. Dengan konsep database yang terpusat ini, maka akan mengurangi kemungkinan duplikasi data. Selain itu juga mudah untuk melakukan updating pada data, ketika data di server di update, maka data yang diterima oleh user juga otomatis akan terupdate.

### **2.7 Google Maps API**

*Google Maps* adalah layanan gratis yang diberikan oleh Google dan sangat populer. Google maps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, google maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. Google menyediakan berbagai API yang sangat berguna bagi pengembang web maupun aplikasi desktop untuk memanfaatkan berbagai fitur yang disediakan oleh Google seperti misalnya: AdSense, Search Engine, Translation maupun Youtube. [5]

*Google Maps API* adalah sebuah layanan (*service*) yang diberikan oleh *Google* kepada para pengguna untuk memanfaatkan *Google Map* dalam mengembangkan aplikasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis *services* yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi *enterprise* di dalam websitenya. Pengguna dapat memanfaatkan layanan-layanan yang ditawarkan oleh *Google Maps* setelah melakukan registrasi dan mendapatkan *Google Maps API Key*. *Google* menyediakan layanan ini secara gratis kepada pengguna di seluruh dunia.

## 2.8 Google Maps Direction API

*Google Maps Directions API* adalah layanan yang menghitung arah antar lokasi menggunakan permintaan *HTTP*. Dengan *Google Maps Direction API* pengembang dapat membangun fitur untuk mencari arah untuk beberapa moda transportasi, termasuk angkutan umum, mengemudi, berjalan atau bersepeda. *Google Maps Directions API* bisa mengembalikan arah multi-bagian menggunakan serangkaian titik jalan.

Layanan ini biasanya didesain untuk menghitung arah alamat statis (sudah diketahui sebelumnya) untuk penempatan konten aplikasi pada peta; layanan ini tidak didesain untuk merespons input pengguna secara real-time, misalnya. Untuk perhitungan arah dinamis (misalnya, dalam elemen antarmuka pengguna).

Untuk menggunakan *Google Maps Direction API* ini, pengembang dapat memanggil URL :

```
https://maps.googleapis.com/maps/api/directions/outputFormat?pa  
r
```

### Gambar 2. 3 Coding Pemanggilan API Direction

Dalam hal ini, *outputFormat* bisa berupa salah satu nilai berikut:

- a. JSON (disarankan) menunjukkan keluaran dalam JavaScript Object Notation (JSON)
- b. XML menunjukkan keluaran berupa XML

## 2.9 TypeScript

[TypeScript](#) adalah bahasa pemrograman berbasis JavaScript yang menambahkan fitur strong-typing & konsep pemrograman OOP klasik ( class, interface). Di dalam dokumentasinya, TypeScript disebut sebagai super-set dari JavaScript, artinya semua kode JavaScript adalah kode TypeScript juga. Kompiler TypeScript menterjemahkan (transpile) sintaks TypeScript ke dalam JavaScript standar yang sudah kita kenal. Tentunya untuk sintaks/konsep OOP belum didukung di JavaScript hanya dipakai oleh TypeScript Compiler (TSC) untuk memverifikasi kode TypeScript yang kita tulis & nggak ada di file JavaScript hasil kompilasi. Bukan berarti konsep ini nggak berguna, justru sebaliknya adanya fitur ini membuat kita bisa menulis aplikasi yang kompleks dengan relatif lebih mudah tanpa perlu pusing mikirin dukungan browser (hasilnya toh tetap JavaScript). Contoh dibawah merupakan coding sederhana dari penggunaan Typescript:

```
// class define in TypeScript

class VirtualPageTracker extends
  Tracker { private
  virtualPageName: string = '';
  constructor(name) {

    super(name);

  }

  getName(): void {

    return this.virtualPageName;
  }
}
```

**Gambar 2. 4 Contoh Coding Typescript**

## 2.10 PHP

PHP merupakan singkatan dari *Hypertext Processor*, yang digunakan sebagai bahasa script dalam pengembangan web yang disisipkan pada dokumen HTML. Pengguna PHP memungkinkan web dapat dibuat dinamis sehingga *maintenance web* tersebut menjadi lebih mudah dan efisien. [6]

Contoh sebuah program PHP:

```

<!DOCTYPE html>
<html>
<head>
<title>Biodata Saya</title>
</head>
<body>
<br>
<?php
Echo "Nama      =          Sugih Purnama"
Echo "NIM       =          10115478"
Echo "Kampus =          Universitas Komputer Indo
      ?>
<br></body></html>

```

**Gambar 2. 5 Contoh Koding PHP**

## 2.11 Java

Java sebagai salah satu bahasa pemrograman yang sudah berumur dari era 1990-an, kian berkembang dan melebarkan dominasinya di berbagai bidang. Salah satu penggunaan terbesar Java adalah dalam pembuatan aplikasi *native* untuk Android. Selain itu Java pun menjadi pondasi bagi berbagai bahasa pemrograman seperti Kotlin, Scala, Clojure, Groovy, JRuby, Jython, dan lainnya yang memanfaatkan Java Virtual Machine sebagai rumahnya.

Java pun akrab dengan dunia saintifik dan akademik. Cukup banyak akademisi di Indonesia yang menggunakan Java sebagai alat bantu untuk menyelesaikan dengan berbagai topik yang didominasi kecerdasan buatan, *data mining*, *enterprise architecture*, aplikasi *mobile*, dan lainnya. Di dunia *web*

*development* sendiri, Java memiliki berbagai *web framework* unggulan seperti Spring, Play Framework, Spark, Jakarta Struts, dan Java Server Pages. [7]

Contoh sebuah program Java:

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Perkenalkan nama  
        saya");  
    }  
}
```

**Gambar 2. 6 Contoh Coding Java**

## 2.12 JSON

JSON adalah format pertukaran data yang bersifat ringan, disusun oleh Douglas Crockford. Fokus JSON adalah pada representasi data di *website*. JSON dirancang untuk memudahkan pertukaran data pada situs dan merupakan perluasan dari fungsi-fungsi *javascript*. [17] JSON dipakai untuk dua hal; yang pertama untuk format transfer data antar device/OS/bahasa pemrograman yang berbeda dan sebagai media penyimpanan data seperti MongoDB. JSON dalam artikel ini lebih menekankan pada transfer data. Sebenarnya selain JSON, format transfer data antar sistem bisa dilakukan XML, plaintext ataupun HTML, namun 3 format tersebut punya banyak kekurangan, terutama dari sisi ukuran. Saingan terdekat dari JSON adalah XML, dalam layanan Web API atau webservice, sebagian layanan menggunakan JSON, sebagian yang lain menggunakan format XML. Dua format ini bisa saling menggantikan.

## 2.13 HTML

HTML adalah singkatan dari Hypertext Markup Language. Disebut hypertext karena di dalam HTML sebuah text biasa dapat berfungsi lain, kita dapat membuatnya menjadi link yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan meng-klik text tersebut. Kemampuan text inilah yang dinamakan hypertext, walaupun pada implementasinya nanti tidak hanya

text yang dapat dijadikan link. Disebut Markup Language karena bahasa HTML menggunakan tanda (mark), untuk menandai bagian-bagian dari text. Misalnya, text yang berada di antara tanda tertentu akan menjadi tebal, dan jika berada di antara tanda lainnya akan tampak besar. Tanda ini di kenal sebagai HTML tag.

Berikut contoh dari HTML adalah sebagai berikut:

```
<html>
  <head>
    <title> Latihan HTML </title>
  </head>
  <body>
    <body bgcolor=#00FFFF>
      <h1> Belajar Website </h1>
      <hr width=100% align=center size=5%
        color="blue">
```

**Gambar 2. 7 Contoh Coding HTML**

## 2.14 CSS

CSS adalah kependekan dari Cascading Style Sheet. CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web supaya lebih elegan dan menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh World Wide Web Consortium atau W3C pada tahun 1996. Awalnya, CSS dikembangkan di SGML pada tahun 1970, dan terus dikembangkan hingga saat ini. CSS telah mendukung banyak bahasa markup seperti HTML, XHTML, XML, SVG (Scalable Vector Graphics) dan Mozilla XUL (XML User Interface Language).

## 2.15 MySQL

Menurut Hariyanto MySQL merupakan DBMS(*Database Management System*) kecil yang kompak, cocok untuk aplikasi berbasis web keperluan minimal dan menengah. MySQL dikelola, didistribusikan dan didukung oleh MySQL AB. MySQL AB adalah perusahaan komersial yang didirikan oleh pengelola MySQL. [15]

MySQL merupakan perangkat lunak *open source* yang artinya perangkat lunak tersebut dapat digunakan dan dikembangkan oleh siapapun. Setiap orang dapat mengunduh aplikasi tersebut dari internet dan tanpa dikenakan biaya apapun. Dan jika kita mau kita bisa mempelajari *source code* dari MySQL dan memodifikasinya sesuai dengan kebutuhan kita. [6]

Contoh sebuah coding MySQL:

```
Create database db_mahasiswa;
```

**Gambar 2. 8 Contoh Coding Membuat Database**

```
create table mahasiswa(  
id_mhs int(4) auto_increment  
primary key,  
  
nim varchar(12),
```

**Gambar 2. 9 Contoh Coding Membuat Tabel**

```
insert into  
mahasiswa(nim,nama_depan,nama_belakang)
```

**Gambar 2. 10 Contoh Koding Menambah Data**

```
select * from mahasiswa;
```

**Gambar 2. 11 Contoh Koding Memlihat Data**

```
delete from mahasiswa where nim =10115478
```

**Gambar 2. 12 Contoh Koding Menghapus Data**

```
update mahasiswa set nama_belakang = 'FadhilKaplale'
where nim = '10113083';
```

**Gambar 2. 13 Contoh Koding Mengubah Data**

```
drop table
```

**Gambar 2. 14 Contoh Tabel Menghapus Tabel**

### 2.15.1 Data Definition Language (DDL)

Perintah dalam SQL yang pertama adalah perintah DDL. DDL sendiri merupakan kependekan dari apa yang dikenal dengan nama Data Definition Language. Apabila diartikan secara harafiah, Data Definition Language berarti merupakan sebuah bahas SQL yang digunakan untuk mendefinisikan suatu data. Secara teoritis, 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.

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

**Gambar 2. 15 Contoh Perintah Dalam sql**

### 2.15.2 Data Control Language (DCL)

Perintah kedua yang ada pada struktur bahasa SQL dalam manajemen basis data adalah DCL/ DCL merupakan kependekan dari data control language. Apabila diartikan secara harafiah, maka tentu saja perintah ini adalah sebuah bahasa yang digunakan untuk mengontrol data dan juga objek di dalam sebuah database.

Perintah DCL ini merupakan perintah yang fungsinya berperan untuk melakukan proses manipulasi user pada sebuah database, dan juga menyangkut hak akses dari sebuah database. Jadi, dengan menggunakan perintah ini, kita dapat menentukan, siapa saja yang berhak untuk mendapatkan akses ke dalam database yang kita miliki. Terdapat beberapa perintah *DCL* pada MySQL sebagai berikut :

- a. *GRANT* berfungsi untuk memberikan hak akses pengguna ke database.
- b. *REVOKE* berfungsi untuk menghilangkan hak akses yang telah diberikan dengan perintah *GRANT*.

```
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
```

**Gambar 2. 18 Contoh Perintah Dalam DCL**

### 2.15.3 Data Management Language (DML)

Perintah terakhir di dalam sebuah struktur bahasa SQL adalah perintah DML. DML merupakan kependekan dari Data management language. Dari kepanjangannya saja sudah dapat terlihat, bahwa yang dimaksud dengan DML

adalah sebuah perintah yang digunakan untuk melakukan proses manipulasi dan pengeditan terhadap data – data yang sudah ada di dalam sebuah data base. Pada dasarnya, fungsi utama dari penggunaan DML di dalam sebuah struktur bahasa SQL adalah untuk melakukan proses pengolahan isi dari data di dalam tabel dan database. Terdapat beberapa perintah *DML* pada MySQL sebagai berikut :

- a. *SELECT* berfungsi untuk mengambil/menampilkan data dari database.
- b. *INSERT* berfungsi untuk memasukkan data ke dalam tabel.
- c. *UPDATE* berfungsi untuk memperbarui data dalam tabel.
- d. *DELETE* berfungsi untuk menghapus data dari tabel.
- e. *CALL* berfungsi untuk memanggil subprogram PL / SQL atau Java
- f. *EXPLAIN PLAN* berfungsi untuk menjelaskan jalur akses ke data

```
INSERT INTO mahasiswa VALUES ("10113083", "Fadhil", "99");
SELECT nama_mahasiswa FROM mahasiswa;
DELETE FROM mahasiswa'
```

**Gambar 2. 17 Contoh Perintah Dalam DML**

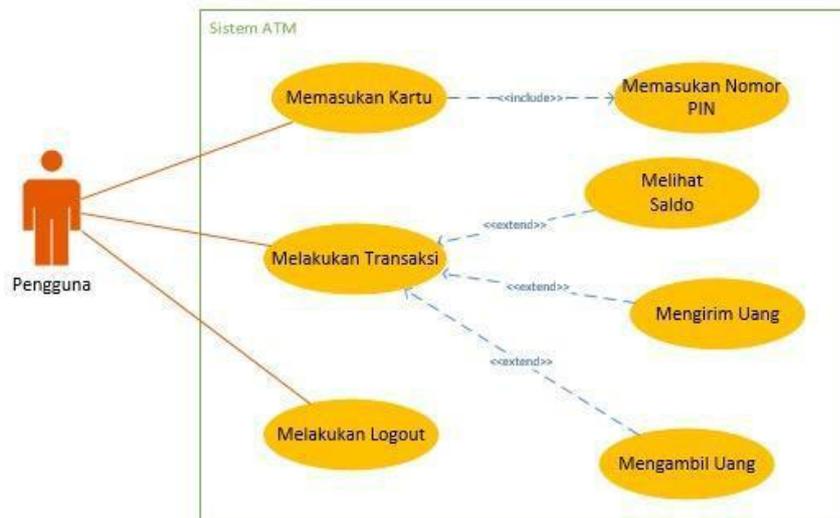
## 2.16 UML

Rancangan yang digunakan adalah UML (*Unified Modelling Language*), UML adalah sebuah bahasa yang didasarkan grafik/gambar untuk memvisualisasi, menspesifikasi, membangun dan pendokumentasi dari sebuah sistem pengembangan software berbasis OOP (*Object Oriented Programing*). UML tidak hanya merupakan sebuah bahasa programan 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 OPP database. [5]

### 2.16.1 Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) aplikasi yang akan dibuat. Usecase mendeskripsikan sebuah interaksi antara suatu

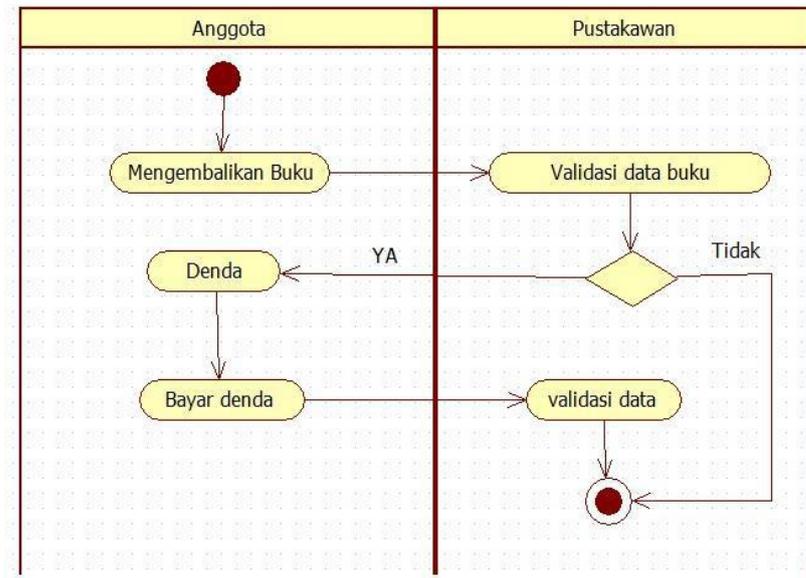
atau lebih actor dengan aplikasi yang akan dibuat . secara kasar, usecase digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah aplikasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. [7]



**Gambar 2.17 Contoh Use Case Diagram**

### 2.16.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Dari pertama sampai akhir, diagram ini menunjukkan langkah-langkah dalam proses kerja sistem yang kita buat. Sebagai contoh, langkah – langkah memasak air. Tetapi kita akan menjelaskannya dengan bentuk grafik. Struktur diagram ini juga mirip dengan flowchart. [7]

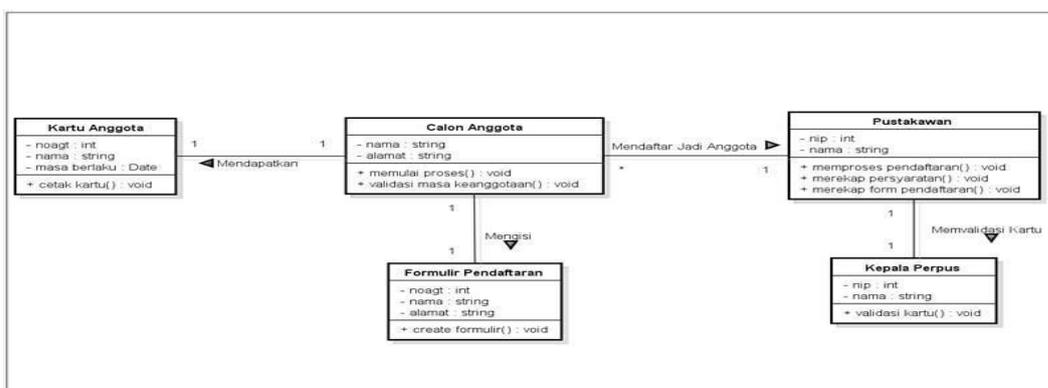


**Gambar 2.19** Contoh Activity Diagram

### 2.16.3 Class Diagram

*Class Diagram* menggambarkan struktur statis dari kelas dalam sistem dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class Diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem. Diagram Kelas menggambarkan struktur sistem dari segi pendefinisian kelas yang akan dibuat untuk membangun sistem.

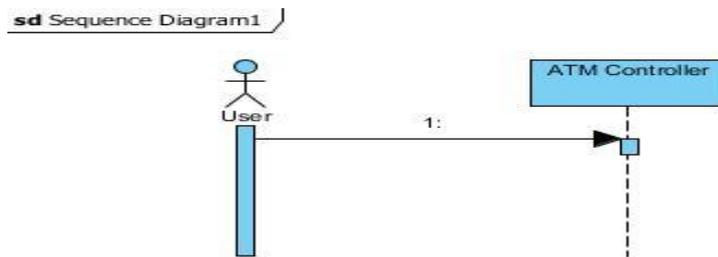
Contoh *Class Diagram* antara lain sebagai berikut:



**Gambar 2. 20** Contoh Class Diagram

### 2.16.4 Sequence Diagram

*Sequence Diagram* atau diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. [7] Berikut merupakan komponen-komponen yang digunakan untuk perancangan sequence diagram:



**Gambar 2. 21 Contoh Sequence Diagram**