BAB 2

LANDASAN TEORI

2.1 Monitoring

Monitoring adalah pemantauan atau dapat dijelaskan sebagai sesuatu yang ingin diketahui. Monitoring akan memberikan informasi bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, monitoring umumnya dilakukan untuk tujuan tertentu.

2.2 Batas Kecepatan Berkendara

Menurut Peraturan Menteri Perhubungan Nomor 111 Tahun 2015 Pasal 3. Menjelaskan mengenai batasan kecepatan berkendara. Peraturan yang dimaksud adalah sebagai berikut.

1. Setiap jalan memiliki batas kecepatan paling tinggi yang ditetapkan secara nasional.
2. Batas kecepatan paling tinggi sebagaimana dimaksud pada ayat 1 meliputi:
    a. batas kecepatan jalan bebas hambatan;
    b. batas kecepatan jalan antarkota;
    c. batas kecepatan jalan pada kawasan perkotaan; dan
    d. batas kecepatan jalan pada kawasan pemukiman.
3. Untuk jalan bebas hambatan sebagaimana dimaksud pada ayat 2 huruf a ditetapkan batas kecepatan paling rendah.
4. Batas kecepatan sebagaimana yang dimaksud pada ayat 2 dan ayat 2 ditetapkan:

a. paling rendah 60 (enam puluh) kilometer per jam dalam kondisi arus bebas dan paling tinggi 100 (seratus) kilometer per jam untuk jalan bebas hambatan;
b. paling tinggi 80 (delapan puluh) kilometer per jam untuk jalan antar kota;
c. paling tinggi 50 (lima puluh) kilometer per jam untuk kawasan perkotaan; dan
d. paling tinggi 30 (tiga puluh) kilometer per jam untuk kawasan permukiman.

5. Batas kecepatan paling tinggi dan batas kecepatan paling rendah sebagaimana dimaksud pada ayat 4 harus dinyatakan dengan rambu lalu lintas.

2.3 Android

Android adalah sebuah sistem operasi untuk *smartphone* dan tablet. Sistem operasi dapat diilustrasikan sebagai jembatan atau penghubung antar piranti *(device)* dan penggunanya, sehingga pengguna dapat berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*

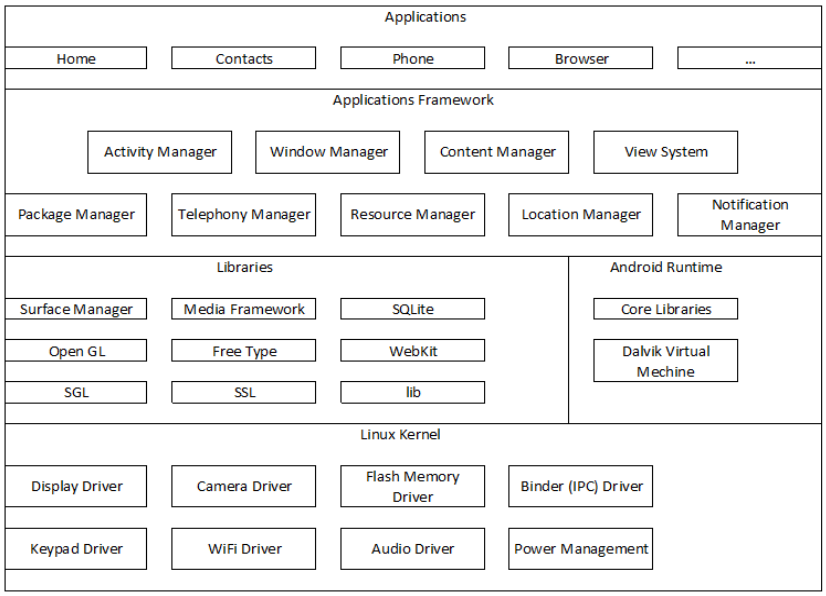
Android merupakan sistem operasi yang bersifat *open source*. Disebut *open source* karena *source code* dari sistem operasi Android dapat dilihat, di-*download* dan dimodifikasi secara bebas. Paragdigma *open source* ini memudahkan pengembangan teknologi Android, karena semua pihak yang tertarik dapat memberikan kontribusi, baik pada pengembangan sistem operasi maupun aplkasi.[1]

2.3.1 Arsitektur Android

Tentunya Android mempunyai arsitektur tersendiri, arsitektur Android dapat digambarkan seperti pada Gambar 2.1. Arsitektur Android dapat dijelaskan sebagai berikut.

a. *Application* dan *Widgets*

Application dan *Widgets* merupakan *layer* yang berhubungan dengan aplikasi-aplikasi inti yang berjalan pada Android OS. Seperti klien *email*, kalender, *browser*, peta, kontak, dan lain-lain. Semua aplikasi ini dibuat dengan menggunakan bahasa Java.



Gambar 2. 1 Arsitektur Android [2]

b. *Applications Frameworks*

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

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notivication Manager*
5. *Activity Manager*

c. *Libraries*

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

d. Android *RunTime*

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

1. *Core Libraries*: berfungsi untuk menerjemahkan bahasa Java/C.
2. *Dalvik Birtual Machine*: sebuah mesin virtual yang dioptimalkan untuk menjalankan fungsi-fungsi pada Android secara efisien.

e. 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 lainnya. Linux Kernel ini berfungsi sebagai jembatan penghubung antara perintah-perintah yang dibuat terhadap hardware yang digunakan pada Android. [2]

2.3.2 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. *Activites*

   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 suatu *activity* ke *activity* lain dapat dilakukan dengan suatu *event* seperti klik tombol.

2. *Service*

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

3. *Broadcast Receiver*

   Broadcast Receiver adalah komponen aplikasi yang menanggapi *system-wide broadcast announcements.* Contohnya adalah memunculkan notifikasi ketika adanya aplikasi baru yang ter-*install*. *Broadcast Receiver* juga tidak memiliki tampilan, tetapi dapat

menjalankan suatu *activity* atau menampilkan notifikasi di Notifikasi Bar.

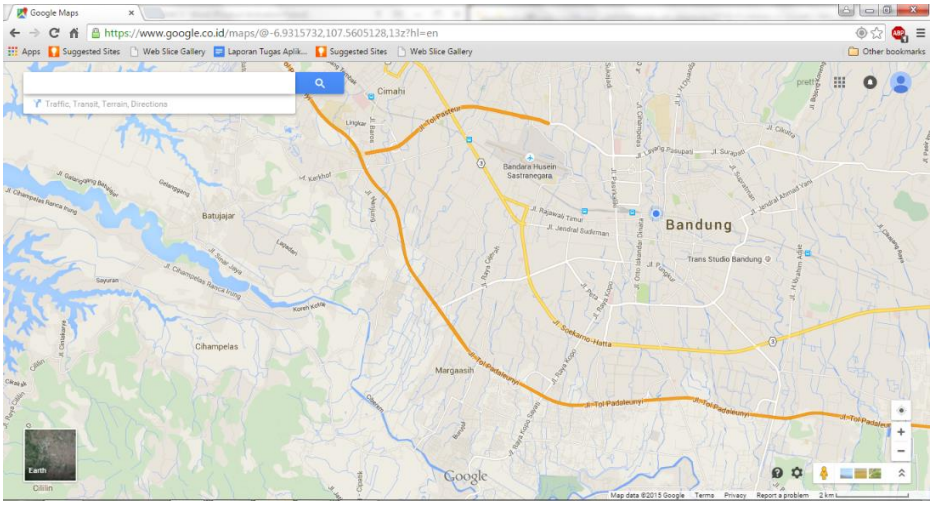
4. *Intent*

   Intent merupakan pesan *Asynchronous* yang memungkinkan aplikasi untuk memberi *request* secara fungsional dari komponen yang berbeda-beda didalam sistem Android. [3]

2.4 Google *Maps*

Google *Maps* merupakan sebuah layanan gratis dari Google mengenai berbagai macam teknologi pemetaan. Google *Maps* menampilkan peta dunia secara digital sehingga dapat dengan mudah mengetahui suatu lokasi tertentu. Google *Maps* menyediakan beberapa keunggulan seperti peta jalan, menunjukan rute untuk suatu perjalanan yang diinginkan baik itu melalui jalur mobil sampai pejalan kaki.

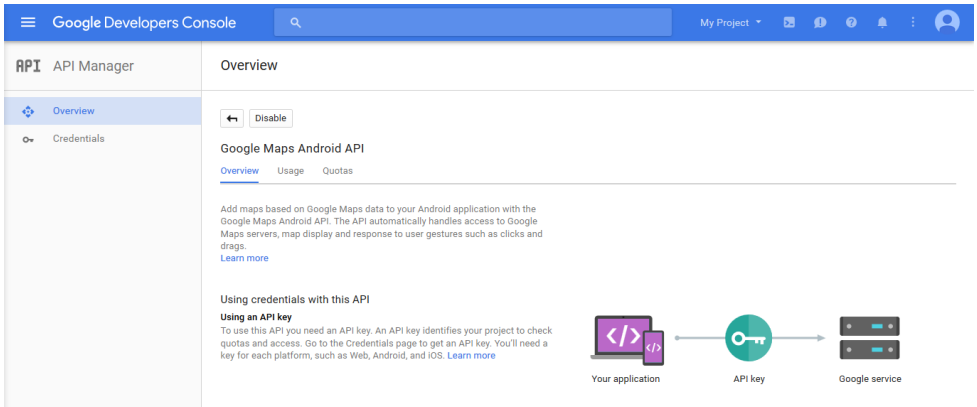
Google *Maps* dibuat dengan menggunakan kombinasi dari gambar peta, *database*, serta objek-objek interaktif yang dibuat menggunakan bahasa pemograman *Hyper Text Markup Language* (HTML), Java dan beberapa bahasa pemograman lainnya. Gambar-gambar peta yang muncul pada layar merupakan hasil komunikasi dari pengguna dengan *database* pada *web server* Google untuk menampilkan gambar yang diminta. [4]



Gambar 2. 2 Google *Maps*

2.5 Google *Maps Application Programing Interface* (Google *Maps* API)

Google *Maps* API merupakan sebuah layanan yang memungkinkan Google *Maps* agar dapat diintegrasikan kedalam aplikasi yang memang membutuhkan fitur dari Google *Maps* ini. Dengan menggunakan Google *Maps* API ini memudahkan para pengembang dalam membuat aplikasi yang membutuhkan teknologi peta didalamnya. Dengan menggunakan Google *Maps* API ini sudah tersedia semua yang dibutuhkan terkait teknologi peta *digital*. [4]



Gambar 2. 3 Google *Maps* API

2.6 *Global Positioning System* (GPS)

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

a. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
b. Lokasi. GPS memberikan informasi lokasi dalam tiga dimensi.
    1. *Latitude*
    2. *Longitude*
    3. *Elevasi*
c. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
d. Arah perjalanan. GPS dapat menunjukkan arah tujuan.
e. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver*.
f. 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 berinteraksi satu sama lain, kemudian pengguna adalah alat navigasi yang digunakan seperti perangkat *mobile* yang kini sudah memiliki fitur GPS

didalamnya. GPS biasanya digunakan untuk menunjukan suatu lokasi yang berada 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. [5]

2.6.1 Akurasi *Global Positioning System* (GPS)

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

Mengapa tingkat akurasi yang terlihat bisa berubah-ubah? Kadang terlihat 10 meter, 15 meter atau 5 meter. Ada beberapa hal yang mempengaruhi tingkat akurasi tersebut, antara lain.

a. Kesalahan *Ephemeris*. Terjadi jika satelit tidak dapat mentransmisikan posisi diorbit dengan tepat.
b. Keadaan *lonosphere. Lonosphere* berada pada jarak sekitar 43-50 mil di atas permukaan bumi. Satelit yang melewati *ionosphere* akan menjadi lambat dikarenakan adanya *plasma* (gas dengan tingkat kepadatan rendah). Walaupun GPS *receiver* berusaha untuk mengkoreksi/memperbaiki faktor keterlambatan yang terjadi tetap saja aktivitas tertentu dari *plasma* bisa menyebabkan kesalahan perhitungan.
c. Keadaan *Troposphere*. *Troposphere* adalah bagian terendah dari *atmosfer* sampai dengan ketinggian sekitar 11 mil dari permukaan tanah. Variasi pada temperatur, tekanan dan kelembaban bisa menyebabkan perbedaan kecepatan penerima gelombang radio.

d. Kesalahan Waktu. Kesalahan waktu dari GPS *receiver* yang tidak presisi dapat menimbukan ketidakakurasian.
e. 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.
f. Buruknya Sinyal Satelit. Keadaan sekitar atau keadaan lingkungan dapat juga menyebabkan GPS sulit untuk menerima data satelit. [6]

2.6.2 Android dan *Global Positioning System* (GPS)

Saat ini teknologi berkembang dengan sangat pesat terutama pada tekonologi komunikasi. Dulu ponsel hanya sekedar digunakan untuk menelpon dan mengirim *Short Message Service* (SMS) atau pesan singkat saja. Salah satu ponsel saat ini adalah ponsel dengan sistem operasi Android.

Dengan standarisasi fitur dan *hardware* yang dimiliki, saat ini banyak orang yang menyukai ponsel dengan sistem operasi Android. Bukan hanya canggih karena adanya fitur *Multimedia Messaging Service* (MMS), radio atau internet berkecepatan tinggi tapi juga karena dilengkapinya fitur teknologi satelit di dalamnya. 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.

2.7 *Location Based Service* (LBS)

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. [7]

2.7.1 Komponen *Location Based Service*

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.

2. *Communication Network*

   Jaringan seluler yang mengirimkan data pengguna dan permintaan layanan.

3. *Positioning Component*

   Untuk dapat mengolah layanan harus menentukan lokasi pengguna. Posisi pengguna ini dapat diperoleh menggunakan jaringan komunikasi atau dengan menggunakan *Global Positioning System* (GPS).

4. *Service and Application Provider*

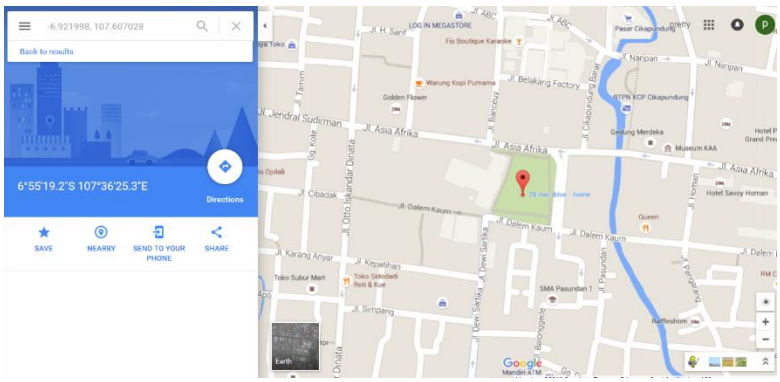
   Penyedia layanan pengguna seluler yang bertanggung jawab untuk memproses suatu layanan.

5. *Data and Content Provider*

   Penyedia layanan informasi data yang dapat diminta oleh pengguna. [8]

2.7.2 *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 Bandung. *Latitude* untuk kota Bandung adalah -6.921998. *Latitude* -6, maksudnya berada dibawah garis khatulistiwa.

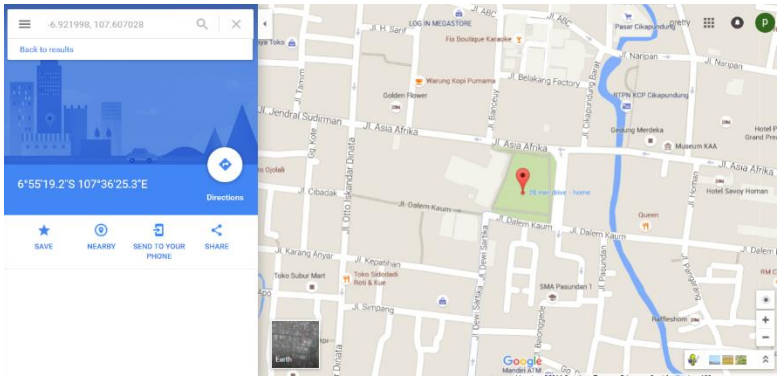


Gambar 2. 4 *Latitude* Kota Bandung

2.7.3 *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 Bandung. *Longitude* kota Bandung adalah 107.607028. Karena logitude 0 berada di Ghana, sedangkan arah +180 berada di Hawai maka posisi *logitude* untuk kota Bandung adalah +107 derajat.



Gambar 2. 5 *Longitude* Kota Bandung

2.8 Web *Service*

Web *service* merupakan aplikasi sekumpulan data (*database*), perangkat lunak (*software*) atau bagian dari perangkat lunak yang dapat diakses secara remote dan berbagai piranti dengan sebuah perantara tertentu. Web service dapat diidentifikasikan dengan menggunakan URL (*Uniform Resource Locator*) seperti web pada umumnya. Namun yang membedakan web *service* dengan web pada umumnya adalah interaksi yang diberikan oleh web *service*. URL web *service* hanya mengandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna untuk membangun sebuah fungsi – fungsi tertentu dari aplikasi.

Web *service* dapat diartikan juga sebuah metode pertukaran data tanpa memperhatikan dimana sebuah *database* ditanamkan. Sehingga web *service* dapat menjadi sebuah jembatan penghubung antara berbagai sistem yang ada.

2.9 Basis data (*Database*)

Basis data atau *database* adalah kumpulan data yang saling berelasi yang disimpan dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat mengorganisasikan data, menghindari duplikasi data. Perangkat lunak yang digunakan untuk mengelola dan memanggil (*Query*) basis data disebut *Database Management System* (DBMS).

2.10 Jaringan Internet

Internet adalah suatu jaringan komputer global yang terbentuk dari jaringan-jaringan komputer lokal dan regional yang memungkinkan komunikasi data antar komputer yang terhubung ke jaringan tersebut. Internet awalnya merupakan suatu rencana dari Departemen Pertahanan Amerika Serikat *(US Departement of Defense)* pada sekitar tahun 1960. Dimulai dari suatu proyek yang dinamakan *Advance Research Project Agency Network* atau ARPANET yang melakukan riset tentang cara menghubungkan komputer satu dengan komputer lainnya agar bisa saling berkomunikasi. [9]

2.11 MySQL

MySQL adalah aplikasi SQL *database server* yang *multi user*. Oleh karena itu, MySQL digunakan dalam pembuatan aplikasi ini yang digunakan sebagai *database server* untuk menyimpan data lokasi yang dikirim oleh masing-masing

telepon seluler. Ada beberapa operasi dasar dalam MySQL yang lebih dikenal dengan operasi *Creat Read Update Delete* (CRUD).

1. *Create*

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

   INSERT INTO *table_name*

   VALUES (*value1*,*value2*,*value3*,...);

   Untuk yang kedua adalah menginputkan data dengan mencantumkan nama kolom yang akan diisikan dengan isi didalamnya. *Query*-nya adalah sebagai berikut.

   INSERT INTO *table_name* (*column1*,*column2*,*column3*,...)

   VALUES (*value1*,*value2*,*value3*,...);

2. *Read*

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

   SELECT *column_name*,*column_name*

   FROM *table_name*;

3. *Update*

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

   UPDATE *table_name*

   SET *column1=value1*,*column2=value2*,...

WHERE *some_column=some_value*;

4. *Delete*

   Delete adalah operasi untuk menghapus data yang ada di dalam *table*. Querynya adalah sebagai berikut.

   DELETE FROM *table_name*

   WHERE *some_column=some_value*; [10]

2.12 Android Studio

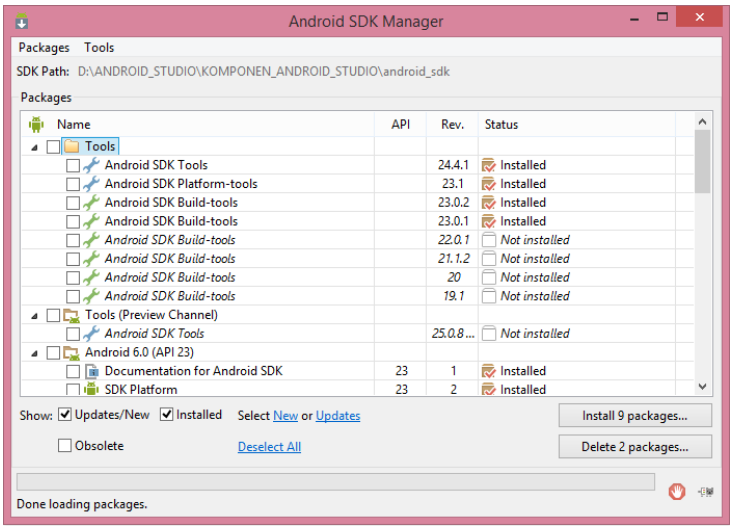
Android Studio merupakan sebuah IDE *(Integreted Development Environment)* untuk platform Android. Android Studio termasuk kedalam *tools* yang bersifat *open source*. Android Studio ini mendukung pengembangan berbasis *Gradle*, dapat melakukan perbaikan dengan cepat, memiliki sebuah *layout editor* 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. 6 Android Studio

2.13 *Software Development Kit* (SDK)

Software Development Kit (SDK) adalah suatu *kit* atau *library* dari bahasa pemograman untuk pengembangan atau pembangunan suatu perangkat lunak dan biasanya SDK terdiri 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.

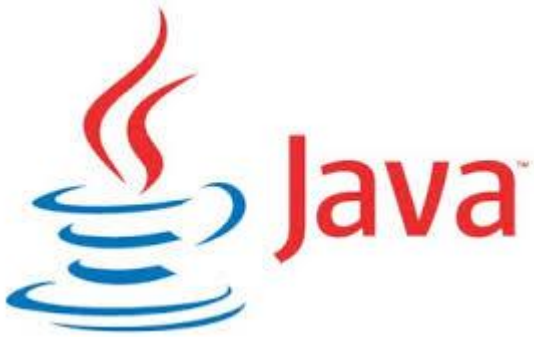


Gambar 2. 7 *Software Developmrnt Kit* (SDK)

2.14 Java *Development Kit* (JDK)

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 benar-benar digunakan untuk menjalankan program Java. Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber kompiler 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. 8 Java *Development Kit* (JDK)

2.15 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.

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 pemograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemograman 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).

2. Larik

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

3. Nilai

Nilai (*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.

4. String

String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam *string* dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string*. *String* sangat mirip dengan *string* C atau Java.

5. Angka

Fungsi Angka pada JSON sangat mirip dengan fungsi angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan. [11]



Gambar 2. 9 JSON

2.16 Java

Java adalah suatu teknologi di dunia *software* komputer, yang merupakan suatu bahasa pemograman, dan sekaligus suatu *platform*. Sebagai bahasa pemogranan, Java dikenal sebagai bahasa pemograman tingkat tinggi. Java mudah dipelajari, terutama bagi programer yang telah mengenal C/C++. Java merupakan bahasa pemograman berorientasi objek yang merupakan paragdigma pemograman masa depan. Sebagai bahasa pemograman Java dirancang menjadi handal dan aman. Java juga dirancang agar dapat dijalankan di semua platform. Dan juga dirancang untuk menghasilkan aplikasi – aplikasi dengan performansi yang terbaik, seperti aplikasi *database*. Sedangkan Java bersifat *neutralarchitecture*, karena Java *Compiler* yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras yang disebut sebagai Java *Bytecode*. Di bawah ini contoh beserta penjelasan dari sintak dasar pada pemograman Java. [12]

```
package com.googlemap;
```
⟶ 1

```
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;        → 2
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
public class MainActivity extends FragmentActivity {        → 3
 private GoogleMap map;
 private LocationManager lm;
 private Double lattitude, longitude;
 private boolean gps_enabled = false;
 private boolean network_enabled = false;
 @Override
 protected void onCreate(Bundle savedInstanceState) {        → 4
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

Penjelasan pada setiap bagian.

1. *Package* atau paket dari kode program. Dengan adanya *package* ini kode program dapat dibuat lebih teratur.
2. *Import* adalah kumpulan dari *library* yang digunakan pada kode program.
3. Kelas yang menyimpan informasi sifat-sifat kode program sesuai apa yang diinginkan
4. *Protected void onCreate()*, merupakan kata kunci dimana aplikasi Java pertama kali akan memanggil method ini.

2.17 *Hypertext Preprocessor* (PHP)

Hypertext Preprocessor (PHP) adalah bahasa pemrograman *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 mendistribusikannya secara bebas. Dibawah ini penjelasan dari beberapa fungsi dasar yang terdapat di PHP.

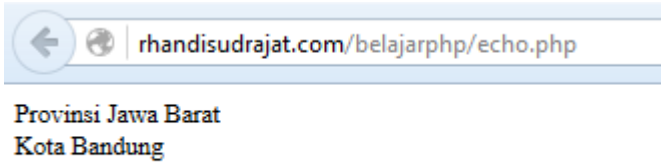
1. Fungsi *echo*

   echo() berfungsi untuk menampilkan satu atau lebih data string kedalam internet *browser*. Fungsi *echo()* memiliki keunikan tersendiri, dimana bisa menyediakan lebih dari satu buah parameter, dengan parameter dasarnya adalah $arg1. Semua data diparameter yang ada bisa di masukan secara langsung ke dalam fungsi *echo(),* atau dengan bantuan variable terlebih dahulu.

   Contoh sederhana dari penggunaan fungsi *echo().*

```
<?php
$Data1    ="Provinsi Jawa Barat";
$Data2    ="Kota Bandung";
echo ("$Data1");
echo ("<BR>n");
echo ("$Data1");
?>
```

   Hasil yang akan ditampilkan adalah sebagai berikut.

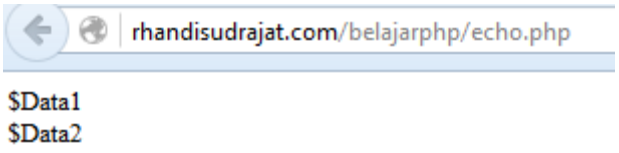


Gambar 2. 10 Fungsi *echo()* dengan kutip ganda (“)

Perbedaan fungsi *echo()* ketika menggunakan tanda kutip ganda (") dengan ketika menggunakan tanda kutip tunggal(') adalah ketika menggunakan tanda kutip ganda, maka semua *variable* yang terdapat didalam data *string* yang akan diproses oleh *echo()*, seluruh data akan diproses terlebih dahulu sehingga semua data dalam *variable* akan dimasukkan ke dalam hasil keluarannya. Sedangkan fungsi *echo()* jika menggunakan tanda petik tunggal ('), semua isi variable tidak akan diproses lebih lanjut sehingga hasil keluarannya hanya menampilkan nama *variable*-nya saja (jika didalamnya ada nama variable) contohnya seperti berikut. [13]

```
<?php
//Data dimasukkan ke dalam Variabel-Variabel.
$Data1    ="Provinsi Jawa Barat";
$Data2    ="Kota Bandung";

echo ('$Data1');
echo ("<BR>n");
echo ('$Data2');
?>
```



Gambar 2. 11 Fungsi echo() dengan petik tunggal (‘)

2. Fungsi ($)

Dengan PHP kita bisa mengambil sebuah nilai atau mengisi nilai pada sebuah *variabel* yang ditandai dengan dollar ($), simbol ini merupakan simbol *variabel* dalam PHP. Selain melakukan 2 hal diatas, kita juga bisa membuat sebuah *variabel* dari sebuah nilai yang ada pada *variabel*, artinya jika var bernilai satu, var adalah *variabel* dan satu adalah nilai, dan jika kita memanggil *variabel* var maka yang keluar adalah satu. Dalam PHP kita bisa membuat sebuah *variabel* yang bernama satu dengan nilai yang bisa kita tentukan sendiri. Berikut penjelasannya.

a. $var=”satu”;

   Mengisi nilai *variabel* $var dengan *string* “satu”.

b. echo $var;

   Output: satu.

c. $$var=1;

   Membuat sebuah *variabel* yaitu satu dengan nilai integer 1.

d. echo $satu;

   Output: 1.