



**MODUL PANDUAN  
MOBILE COMPUTING SYSTEM  
UNIVERSITAS GUNADARMA  
ATA 2016-2017**



**LABORATORIUM LANJUT SISTEM KOMPUTER  
UNIVERSITAS GUNADARMA**

## DAFTAR ISI

<b>BRIEFING .....</b>	<b>3</b>
<b>BAB 1 ANDROID BASIC APPLICATION .....</b>	<b>31</b>
<b>BAB 2 ANDROID MULTIMEDIA .....</b>	<b>42</b>
<b>BAB 3 ANDROID ANIMATION DAN GAME .....</b>	<b>48</b>
<b>BAB 4 ANDROID NETWORK PROGRAMMING .....</b>	<b>76</b>
<b>BAB 5 ANDROID SQLite DATABASE .....</b>	<b>83</b>
<b>BAB 6 ANDROID CLIENT-SERVER USING JSON .....</b>	<b>89</b>
<b>BAB 7 ANDROID GOOGLE MAPS API V2 .....</b>	<b>109</b>
<b>BAB 8 ANDROID SOCKET PROGRAMMING .....</b>	<b>118</b>

**B**

# BRIEFING



**B****BRIEFING****PENGENALAN JAVA & ANDROID****B.1 Sekilas Tentang Mobile Computing System**

Gambar B.1 Computing System

*mobile computing* adalah seperangkat benda atau teknologi yang memiliki teknologi secanggih komputer dan mampu melakukan komunikasi walaupun user atau pengguna dari alat tersebut sedang melakukan perpindahan.

**1. Perkembangan *Mobile Computing***

1. Personal Computer
2. Networking, LAN, dan WLAN
3. Laptop, handphone, dan alat komunikasi lainnya
4. Mobile computing

**B.2 ANDROID**

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android memiliki empat karakteristik sebagai berikut:

### **1. Terbuka**

Android merupakan sistem operasi yang bersifat *open source* atau dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul.

### **2. Dengan Versi yang sama, android pada tiap smarthphone dapat berbeda**

Setiap pabrikan smarthphone biasanya menggunakan tampilan yang menjadi ciri khas dari smarthphone tersebut sehingga interface dan fitur yang dimiliki berbeda-beda.

### **3. Harga relatif murah**

Smartphone yang menggunakan android sekarang ini relatif murah. Ini dikarenakan jumlah produsen android semakin banyak dan bersaing.

### **4. Pengembangan aplikasi yang cepat dan mudah**

Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan library yang diperlukan dan tools yang dapat digunakan untuk membangun aplikasi yang semakin baik.

Google Inc. sepenuhnya membangun Android dan menjadikannya bersifat terbuka (open source) sehingga para pengembang dapat menggunakan Android tanpa mengeluarkan biaya untuk lisensi dari Google dan dapat membangun Android tanpa adanya batasan-batasan. Android Software Development Kit (SDK) menyediakan alat dan Application Programming Interface (API) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java.

#### **B.2.1 Sejarah Android**

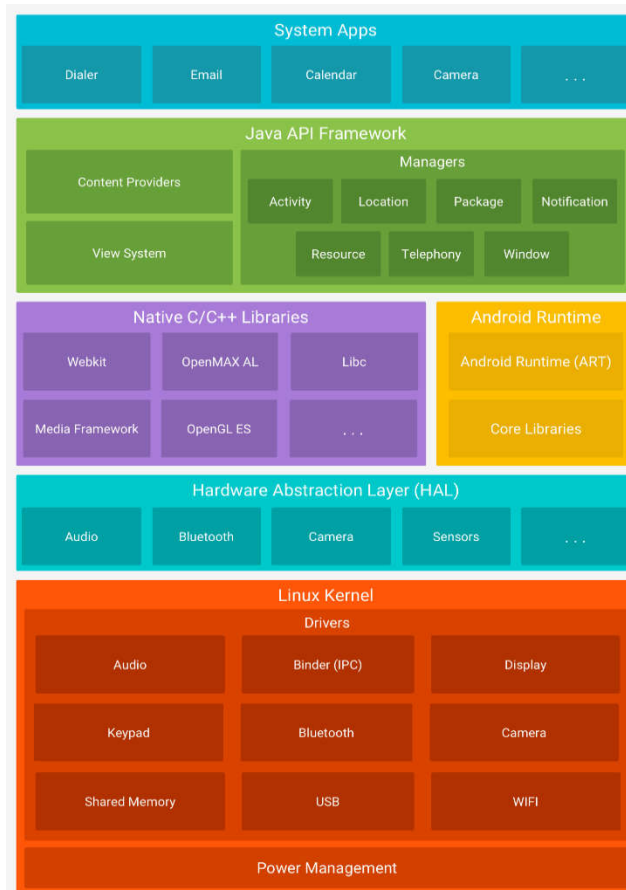
Android adalah sistem operasi berbasis kernel Linux yang pada awalnya dikembangkan oleh Android, Inc, yang didukung Google finansial dan kemudian dibeli pada tahun 2005.

Android ini diresmikan pada tahun 2007 seiring dengan berdirinya Open Handset Alliance-konsorsium hardware, software, dan perusahaan telekomunikasi yang ditujukan untuk memajukan standar perangkat selular.

Smartphone yang tersedia untuk publik pertama kalinya yang menjalankan Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008.

### B.2.2 Arsitektur Platform

Android adalah tumpukan perangkat lunak berbasis Linux *open source* yang dibuat untuk berbagai perangkat. Diagram berikut menunjukkan komponen besar dari platform Android.



Gambar B.2 Arsitektur Sistem Android

#### 1. Linux Kernel

Fondasi platform Android adalah kernel Linux. Dengan menggunakan kernel Linux akan memungkinkan Android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen perangkat untuk mengembangkan driver perangkat keras untuk kernel yang sudah cukup dikenal.

#### 2. Hardware Abstraction Layer (HAL)

Hardware Abstraction Layer (HAL) menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras di perangkat ke kerangka kerja Java API yang lebih tinggi. HAL

terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth. Bila API kerangka kerja melakukan panggilan untuk mengakses perangkat keras, maka sistem Android memuat modul pustaka untuk komponen perangkat keras tersebut.

### **3. Android Runtime**

Untuk perangkat yang menjalankan Android versi 5.0 (API level 21) atau yang lebih tinggi, setiap aplikasi menjalankan proses masing-masing dengan tahap Android Runtime (ART). ART digunakan untuk menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi file DEX.

Beberapa fitur utama ART mencakup:

1. Kompilasi mendahului waktu (AOT) dan tepat waktu (JIT)
2. Pengumpulan sampah (GC) yang dioptimalkan
3. Dukungan debug yang lebih baik, mencakup profiler sampling terpisah, pengecualian diagnostik mendetail dan laporan kerusakan dan kemampuan untuk mengatur titik pantau guna memantau bidang tertentu.

### **4. Native C/C++ Libraries**

Banyak komponen dan layanan sistem Android inti seperti ART dan HAL dibuat dari kode asli yang memerlukan pustaka asli yang tertulis dalam C dan C++. Platform Android memungkinkan kerangka kerja Java API mengekspos fungsionalitas beberapa pustaka asli pada aplikasi.

Jika ingin mengembangkan aplikasi yang memerlukan kode C atau C++ bisa menggunakan Android NDK untuk mengakses beberapa pustaka platform asli langsung dari kode asli.

### **5. Java API Framework**

Keseluruhan rangkaian fitur pada Android OS tersedia melalui API yang ditulis dalam bahasa Java. API ini membentuk elemen dasar yang diperlukan untuk membuat aplikasi Android, komponen dan layanan sistem modular, yang menyertakan berikut ini:

1. Tampilan Sistem yang kaya dan luas bisa digunakan untuk membuat UI aplikasi, termasuk daftar, kisi, kotak teks, tombol, dan bahkan browser web yang dapat disematkan
2. Pengelola Sumber Daya, memberikan akses ke sumber daya bukan kode seperti string yang dilokalkan, grafik, dan file layout
3. Pengelola Notifikasi yang mengaktifkan semua aplikasi guna menampilkan lansiran khusus pada bilah status
4. Pengelola Aktivitas yang mengelola daur hidup aplikasi dan memberikan back-stack navigasi yang umum
5. Penyedia Materi yang memungkinkan aplikasi mengakses data dari aplikasi lainnya, seperti aplikasi Kontak, atau untuk berbagi data milik sendiri

Developer memiliki akses penuh ke API Framework yang sama dengan yang digunakan oleh aplikasi sistem Android.

## **6. System Apps**

Aplikasi sistem berfungsi sebagai aplikasi untuk pengguna dan memberikan kemampuan kunci yang dapat diakses oleh developer dari aplikasi mereka sendiri. Misalnya, jika aplikasi yang dibuat ingin mengirimkan pesan SMS, tidak perlu membangun fungsionalitas tersebut sendiri—sebagai gantinya bisa menjalankan aplikasi SMS mana saja yang telah dipasang guna mengirimkan pesan kepada penerima yang telah ditetapkan.

## **7. Library**

Library berisi sekumpulan instruksi untuk mengarahkan perangkat android dalam menangani berbagai tipe data.

Berikut ini adalah beberapa kegunaan dari Library :

1. Surface Manager: Mengolah tampilan Windows Pada Layar
2. SGL: Grafik 2 Dimensi
3. Open GL|ES: Grafik 3 Dimensi maupun 2 Dimensi
4. Media Framework: Menunjang perekaman dari berbagai macam format audio, video, dan gambar
5. Free Type: Penerjemah Font
6. WebKit: Mesin Browser



7. libc (System C libraries)
8. SQLite: Database
9. Open SSL: Sekuritas Jaringan

a. Android Libraries

Berikut ini beberapa Library Android yang tersedia untuk Android Developer adalah sebagai berikut :

1. android.app - memberikan akses ke model aplikasi dan fondasi dari semua aplikasi android
2. android.content - memfasilitasi aksien konten, mempublikasikan dan mengirim pesan antar aplikasi dan komponen aplikasi.
3. android.database Digunakan untuk mengakses data yang dipublikasikan oleh content providers, juga termasuk kelas manajemen basis data SQLite.
4. android.opengl antarmuka Java ke Api penerjemah OpenGL ES 3D Graphic
5. android.os Menyediakan akses aplikasi ke OS system service standar seperti perpesanan, system service, dan IPC ( Inter Process Communication ).
6. android.text Digunakan untuk menyajikan dan memanipulasi teks yang ditampilkan pada perangkat.
7. android.view Dasar dari pembangunan user interface aplikasi.
8. android.widget Koleksi dari beberapa user interface yang telah dibuat dan dapat langsung digunakan seperti tombol, label, tabel, view, layout manager, tombol pilihan, dll.
9. android.webkit Sekumpulan kelas untuk memungkinkan kemampuan web browsing yang akan dibangun ke dalam aplikasi.

### **B.2.3 Perkembangan Android**

Android mengalami perkembangan yang sangat cepat, android sudah dikembangkan sebanyak 22 kali, seperti berikut ini :



Gambar B.3 Android Version

1. Android 1.0 Alpha
2. Android 1.1 Beta
3. Android 1.5 Cupcake
4. Android 1.6 Donut
5. Android 2.0 Eclair
6. Android 2.0.1 Eclair
7. Android 2.1 Eclair
8. Android 2.2 Froyo
9. Android 2.3 Gingerbread
10. Android 2.3.3 Gingerbread
11. Android 3.0 Honeycomb
12. Android 3.1 Honeycomb
13. Android 3.2 Honeycomb
14. Android 4.0 Ice Cream Sandwich
15. Android 4.0.3 Ice Cream Sandwich
16. Android 4.1 Jelly Bean
17. Android 4.2 Jelly Bean
18. Android 4.3 Jelly Bean
19. Android 4.4 KitKat
20. Android 5.0 Lollipop
21. Android 6.0 Marshmallow
22. Android 7.0 Nougat

## **B.2.4 Fitur-fitur Android**

Berikut ini berbagai fitur dari sistem operasi Android:

1. Rancangan handset. Platform disesuaikan dengan kebutuhan VGA (Video Graphics Adapter) yang lebih besar, library grafis 2D dan 3D yang berdasarkan pada spesifikasi OpenGL ES 1.0 serta layout smartphone yang tradisional.
2. Penyimpanan. Android menggunakan software database SQLite sebagai penyimpanan data.
3. Konektivitas. Android mendukung berbagai teknologi konektivitas, seperti GSM (Global System for Mobile Communications) /EDGE (Enhanced Data rates for GSM Evolution), CDMA (Code Division Multiple Access), EV-DO (Evolution-Data Optimized), UMTS (Universal Mobile Telecommunications System), bluetooth dan Wi-Fi (Wireless Fidelity).
4. Pesan. Android mendukung pengiriman pesan dalam bentuk SMS (Short Message Service) dan MMS (Multimedia Messaging Service).
5. Web browser. Web browser yang tersedia di Android berdasarkan pada framework aplikasi open source WebKit.
6. Dukungan Java. Software yang ditulis dalam bahasa Java dapat dikompilasi dan dieksekusi pada mesin virtual Dalvik, yang merupakan implementasi dari VM (Virtual Machine) yang dirancang khusus untuk penggunaan perangkat bergerak.
7. Dukungan media. Android mendukung beberapa format audio/video seperti: H.263, H.264 (dalam kontainer 3GP atau MP4), MPEG-4 SP, AMR, AMR-WB (dalam kontainer 3GP), AAC, HE-AAC (dalam kontainer MP4 atau 3GP), MP3, MIDI, OGG Vorbis, WAV, JPEG, PNG, GIF dan BMP.
8. Dukungan hardware lainnya. Android mendukung penggunaan kamera, layar sentuh, GPS (Global Positioning System), accelerometers, magnetometers, akselerasi 2D bit blits (dengan orientasi hardware, scaling, konversi format piksel) dan akselerasi grafis 3D.
9. Lingkungan Development. Android menyediakan perangkat emulator, debugger, memori dan profil kinerja yang tersedia pada plugin Eclipse IDE (Integrated Development Environment).
11. Market. Android Market adalah sebuah katalog aplikasi yang dapat di-download dan diinstal pada telepon seluler secara online, tanpa menggunakan PC (Personal Computer).

12. Multi-touch. Android memiliki dukungan bawaan untuk multi-touch yang tersedia pada handset terbaru, seperti HTC Hero. Pada awalnya fitur tersebut dinonaktifkan pada level kernel (mungkin untuk menghindari pelanggaran paten terhadap teknologi layar sentuh Apple). Sejak Google merilis update untuk Nexus One dan juga berencana untuk merilis update untuk Motorola Droid yang memungkinkan multi-touch.

### **B.2.5 Pemrograman Java Android**

Java merupakan bahasa pemrograman tingkat tinggi (high level), artinya bahasa ini mudah dipahami oleh manusia karena menggunakan bahasa sehari-hari manusia. Java pada awalnya dibuat oleh perusahaan Sun Microsystems, yang sekarang dimiliki oleh perusahaan Oracle. Bahasa java mirip dengan bahasa C++, namun dikembangkan agar mudah dipahami dan dipakai. Java menggunakan konsep OOP (Object Oriented Programming), artinya semua inti program dari Java yaitu mengacu ke sebuah Obyek. Java termasuk platform independent, yang artinya bisa dijalankan dengan berbagai Macam OS (Sistem Operasi) dan arsitektur komputer. Seperti pada perangkat Android dari Google dan Beberapa Aplikasi Desktop seperti program AntiVirus, program media player, Aplikasi Enterprise dll.

#### **Kelebihan Java**

1. Bersifat portable dan platform independent.
2. Memiliki garbage collection yang dapat deallocasi memori secara otomatis.
3. Menghilangkan pewarisan berganda yang terdapat pada C.
4. Mengurangi pointer aritmatik.
5. Mengurangi kerancuan antara pemberian nilai pada conditional statement.

#### **Kelemahan**

1. Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain dalam Java.
2. Mudah didekompilasi.
3. Penggunaan memori yang banyak.

### **B.2.5.1 Konsep OOP**

#### **1. Class**

Merupakan sebuah kerangka atau bentuk awal, yang berfungsi untuk tempat menaruh, dan mendeksripsikan variabel, method dari sebuah obyek. Penamaan nama class diprogram, harus sama dengan nama di struktur dengan extention .java.

#### **2. Object**

Merupakan sebuah representasi dan instance dari class. Object adalah sebuah inti dan wujud real dari sebuah Class. Contoh: dari class binatang, memiliki obyek yaitu kucing, ayam, kelinci.

#### **3. Attribute**

Merupakan sebuah unsur data yang ada di class, biasanya terdiri dari sebuah data, variabel, prototipe, dan field. Contoh: Manusia memiliki attribute nama, tinggi, berat, umur, dan jenis kelamin.

#### **4. Method**

Merupakan perilaku dari sebuah class. Contoh: Manusia mempunyai method Berjalan, Berlari, Loncat, Berbicara, dan Tidur.

#### **5. Encapsulation**

Mekanisme membungkus suatu data, agar tidak dapat diakses oleh class lain.

#### **6. Inheritance (Pewarisan)**

Suatu proses dimana, suatu class yang bisa disebut super class, dapat mewarisi sifat turunan kedalam sub class.

#### **7. Polymorphism**

Suatu kemampuan yang dimiliki sebuah method, yang memiliki nama sama, namun dengan perilaku yang berbeda-beda.

#### **B.2.5.1.1 Modifier**

Merupakan sebuah ijin hak akses untuk penggunaan suatu atribut atau method, pada suatu package dan class. Terdapat empat 4 jenis modifier yaitu:

1. Public : dapat diakses dari class yang sama, package yang sama, subclass, dan package lain.
2. Protected: dapat diakses dari class yang sama, package yang sama, subclass, tapi tidak dapat diakses dari package lain.
3. Privatae: hanya dapat diakses oleh class yang sama.
4. No Modifiers: dapat diakses dari class dan package yang sama.

#### **B.2.5.1.2 Struktur program java**

```
package namaPackage;  
import namaPackage;  
/**Komentor dalam java */  
modifier class namaKelas {  
    //deklarasi variable  
    modifier methodConstructor() {  
        //statement  
    }  
    public static void main() {  
        /*  
        *Statement1  
        *Statement2  
        */  
    }  
}
```

Penjelasan :

1. Package, cara pengelompokan dan pengorganisasian class-class dalam satu library

2. Import, perintah untuk mengimpor class tertentu anda dapat menuliskan nama class.
3. Komentar dalam java, catatan yang ditulis bersama kode program yang berguna sebagai catatan atau keterangan dari suatu kode, sehingga dapat digunakan sebagai referensi dalam pembuatan dokumentasi.
4. Identifier java, suatu tanda yang berupa huruf, kata, atau gabungan kata yang digunakan sebagai nama variable, method, class, dan sebagainya.
5. Deklarasi Class, mempertimbangkan dimana dan bagaimana class tersebut digunakan.
6. Modifier, keyword java yang berfungsi mengatur hubungan antar class, method, dan variable.
7. Deklarasi method, bagian program yang menjelaskan tingkah laku dari objek yang akan diinstan.

### B.2.6 SDK

Android SDK (Software Development Kit) merupakan suatu alat, platform, dan komponen lain yang dibutuhkan untuk pengembangan aplikasi android, Android studio telah menyediakan manajer untuk android SDK yang dapat digunakan untuk men-download SDK menggunakan SDK Manajer.

### B.2.7 Layout

Layout mendefinisikan struktur visual untuk antarmuka pengguna, seperti UI sebuah [aktivitas](#) atau [widget aplikasi](#). Layout dapat dideskripsikan dengan dua cara:

1. **Deklarasikan elemen UI dalam XML.** Android menyediakan sebuah kosakata XML sederhana yang sesuai dengan kelas dan subkelas View, seperti halnya untuk widget dan layout.
2. **Buat instance elemen layout saat waktu proses.** Aplikasi Anda bisa membuat objek View dan ViewGroup (dan memanipulasi propertinya) lewat program.

Keuntungan mendeklarasikan UI dalam XML adalah karena hal ini memungkinkan untuk memisahkan penampilan aplikasi dari kode yang mengontrol perilakunya dengan lebih baik. Mendeklarasikan layout dalam XML akan mempermudah dalam memvisualisasikan struktur UI,

sehingga lebih mudah men-debug masalahnya. Karena itu, dokumen ini berfokus pada upaya mengajari cara mendeklarasikan layout dalam XML.

Inilah layout XML yang menggunakan [LinearLayout](#) vertikal untuk menyimpan [TextView](#) dan [Button](#):

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

Gambar B.4 Source Code Text View Dan Button

#### B.2.6.1 Parameter Layout

Semua grup tampilan berisi lebar dan tinggi (`layout_width` dan `layout_height`), dan masing-masing tampilan harus mendefinisikannya. Banyak `LayoutParams` yang juga menyertakan margin dan border opsional.

Anda bisa menetapkan lebar dan tinggi dengan ukuran persis, meskipun Anda mungkin tidak ingin sering-sering melakukannya. Lebih sering, Anda akan menggunakan salah satu konstanta ini untuk mengatur lebar atau tinggi:

- *wrap\_content* menjadikan tampilan menyesuaikan sendiri ukurannya dengan dimensi yang dibutuhkan oleh materinya.
- *match\_parent* menjadikan tampilan memiliki besar yang akan diperbolehkan oleh kelompok tampilan induknya.

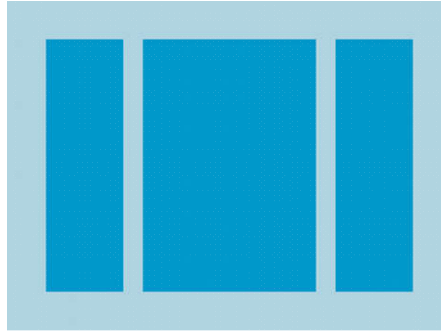
Secara umum, menetapkan lebar dan tinggi layout dengan satuan mutlak seperti piksel tidaklah disarankan. Sebagai gantinya, menggunakan pengukuran relatif seperti satuan piksel yang tidak bergantung pada kerapatan (*dp*), *wrap\_content*, atau *match\_parent*, adalah sebuah pendekatan yang lebih baik, karena membantu memastikan bahwa aplikasi Anda akan ditampilkan dengan benar pada berbagai ukuran layar perangkat.



### B.2.6.2 Layout Umum

#### 1. Linier Layout

LinearLayout adalah sekelompok tampilan yang menyejajarkan semua anak dalam satu arah, secara vertikal atau horizontal.



Gambar B.5 Linier Layout

#### 2. Relative Layout

RelativeLayout adalah sebuah layout dimana posisi dari sebuah komponen (simbol, text, dsb) letaknya bisa diatur terhadap komponen lainnya.

#### 3. Tampilan WEB

Menampilkan laman web.

### B.2.7 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi Android. Selain merupakan editor kode dan alat pengembang, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya:

1. Sistem pembuatan berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat penguji dan kerangka kerja yang ekstensif

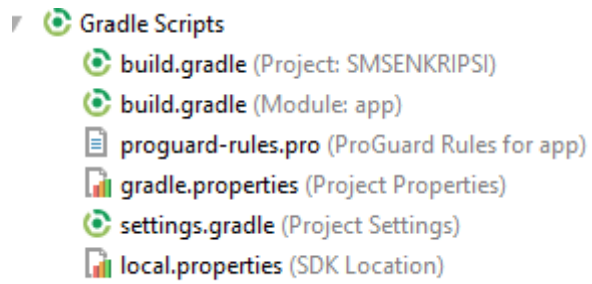
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

### B.2.8.1 Gradle

Aplikasi Android dibangun menggunakan Open Source Gradle Build System. Secara default **Gradle** adalah build tools yang digunakan pada Android Studio, untuk mencompile-menjalankan project Aplikasi yang sedang kalian kembangkan , berdasarkan settingan atau konfigurasi yang berada di bagian Gradle Script.. Android Plugin untuk Gradle berjalan secara independent (bebas) di Android Studio. Yang berarti bisa melakukan build Aplikasi Android di dalam Android Studio, serta juga dapat membuild lewat Command Line menggunakan Gradle. Serta mempunyai fitur yang spesifikasinya diperuntukan untuk Aplikasi Android ,Menentukan build tipe ,flavors ,Signing configurations ,Menambahkan library project ,dan masih banyak lagi.

Bagi seorang developer Android baik pemula ataupun sudah mahir , sangat penting sekali mempelajari cara menggunakan Gradle dengan baik dan benar. Karna saat mengkonfigurasi gradle merupakan tahap awal dan tahap akhir , saat mengembangkan Aplikasi di Android Studio. Contoh **tahap awal** : yaitu ketika ingin mensetting minimum SDK API yang akan digunakan ,dan minimum requirement OS yang compatible dengan Aplikasi dan mensetting library-library yang akan digunakan dll. **tahap akhir** : menentukan version name dari Aplikasi dan settingan lainnya, saat Aplikasi sudah ready dan bersih dari bug-bug yang ada , dan siap untuk di publikasikan di Google Play Store.

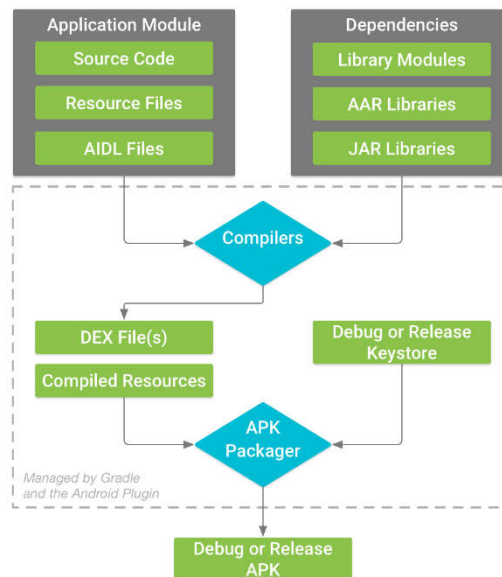
Banyak juga masalah yang terjadi oleh developer Android , saat mengembangkan Aplikasinya ,dikarena konfigurasi file di Gradle yang tidak benar. Saat membuat dan mengembangkan project Aplikasi Android, pasti akan melihat di bagian struktur project Android , di sebelah kiri yang bernama Gradle Script.



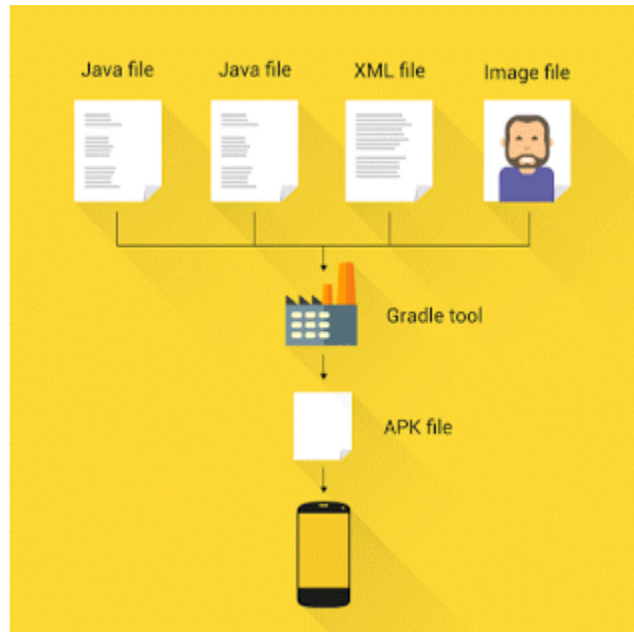
Gambar B.6 Lokasi Gradle Script Android Studio

## 1. Proses Building

Proses dimana melibatkan banyak tools, lalu mengconvert project ke dalam Android Package (APK). Bisa dilihat prosesnya pada gambar di bawah ini:



Gambar B.7 Pengaturan Gradle untuk Android Plugin



Gambar B.8 Proses Kerja Gradle

## 2. setting.gradle

*include ':app'*

Salah satu keunggulan kita menggunakan Gradle ,kita bisa membuat sebuah **multi-project**. Jadi code **include** menunjukan project yang ada di dalam aplikasi, di atas hanya ada project 'app'(secara default atau bawaan). Pada bagian setting.gradle ini bisa menambahkan library project turunan, yang digunakan untuk mendukung project utama dari Aplikasi yang dibuat.

### 3. **build.gradle** (Project :Nama Project)

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.

buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.5.0'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

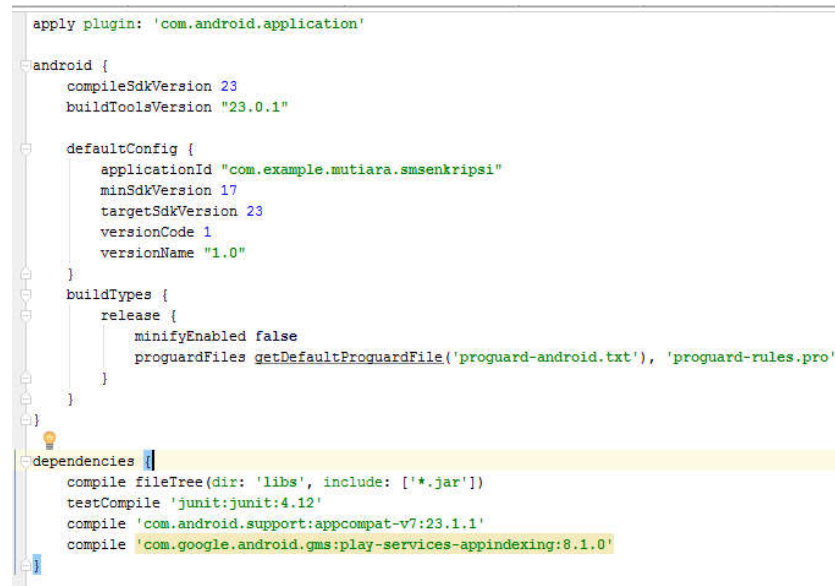
Gambar B.9 build.gradle

Secara default atau bawaan, Gradle tidak memasukan fungsi-fungsi Android. Namun, Google menyediakan Android Plugin untuk Gradle, yang dimana membantu programmer dengan mudah mengkonfigurasi file project Android.

Seperti yang telah dilihat di atas merupakan konfigurasi file top level, yang dimana bisa menambahkan pilihan konfigurasi secara umum untuk project turunan /modules. Pada code block **buildscript** , digunakan untuk mendownload plugin yang di perlukan Gradle ,seperti code diatas secara default ,digunakan untuk mengunduh repository **jcenter**.

Pada code **allprojects** berfungsi untuk menunjukan bahwa project top level atau project turunan yang di setting secara default, akan mendownload plugin atau library Java yang ada di repository **jcenter**. Nah selanjutnya pada code **task clean** ini dengan tipe delete, ini menunjukan bahwa direktori build akan dihapus pada project root atau utama, karna secara default ini berada pada build di top level.

#### 4. build.gradle (Module :app)



```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.1"

    defaultConfig {
        applicationId "com.example.mutiara.smsenkripsi"
        minSdkVersion 17
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }

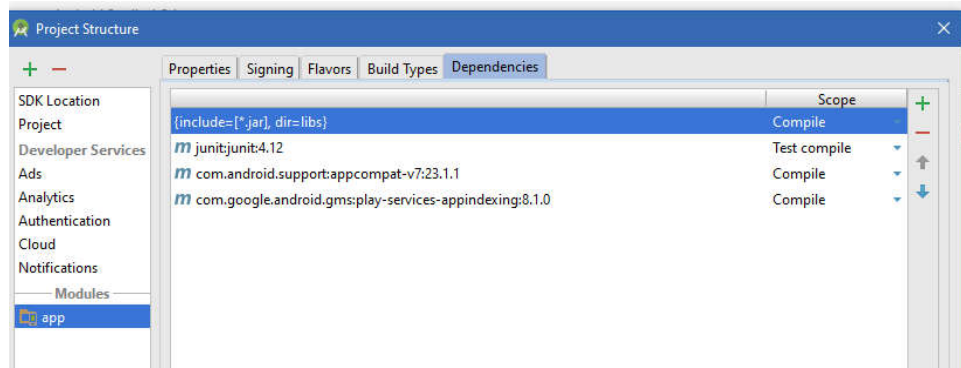
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.google.android.gms:play-services-appindexing:8.1.0'
```

Gambar B.10 build.gradle (module:app)

Seperti yang telah dilihat diatas, merupakan konfigurasi file app level , pada code **apply** digunakan untuk menambahkan plugin Android ke dalam build sistem. Pada block code **dependencies** terdapat 4 baris code. Baris pertama yaitu file tree dependency , yang dimana baris pertama maksudnya adalah semua file yang mempunyai extension *.jar* , di dalam *lib* folder akan di compile. Baris kedua menunjukan bahwa Gradle mendownload Junit versi 4.12, yang dimana fungsi code **testCompile** , akan menyediakan class-class didalam *app/src/androidTest/Java*, dan juga *src/test/java* pada struktur project, yang akan digunakan untuk unit test. Baris ketiga menunjukan bahwa Gradle menambahkan appcompat-v7 jar versi 23.1.1 dari Android Support Library. Baris keempat menunjukan gradle menambahkan play-services appindexing dari google.android.gms.

Untuk versi UI nya juga dapat disetting di Android Studio, pada bagian **File > Project Structure** Lalu akan tampil seperti ini :



Gambar B.11 Project Structure

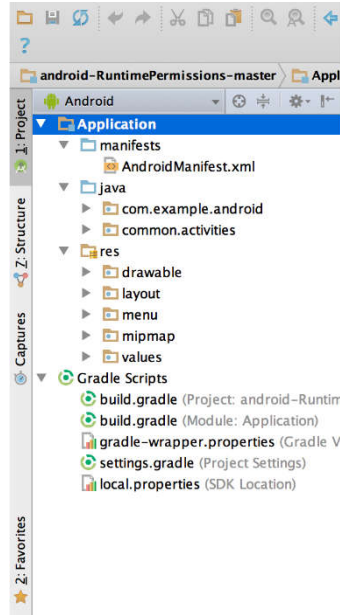
### B.2.8.2 Project Structure

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup:

1. Modul aplikasi Android
2. Modul Library
3. Modul Google App Engine

Secara default, Android Studio menampilkan file proyek dalam tampilan proyek Android. Tampilan ini diatur menurut modul untuk memberi akses cepat ke file sumber kunci proyek. Semua file versi terlihat di bagian atas di bawah **Gradle Scripts** dan masing-masing modul aplikasi berisi folder berikut:

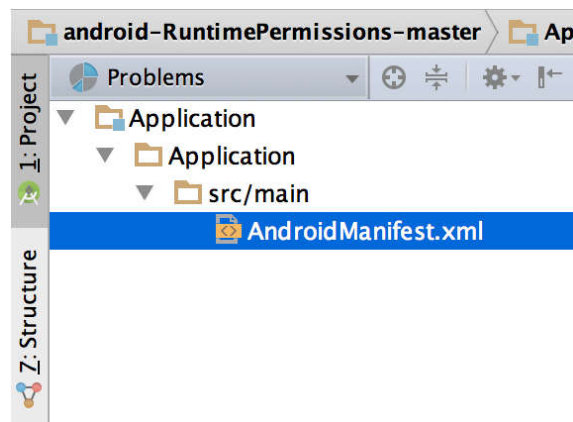
1. **manifests**: Berisi file AndroidManifest.xml.
2. **java**: Berisi file kode sumber Java, termasuk kode pengujian JUnit.
3. **res**: Berisi semua sumber daya bukan kode, seperti tata letak XML, string UI, dan gambar bitmap.



Gambar B.12 Modul App

Untuk melihat struktur file sebenarnya dari proyek ini, pilih **Project** dari menu tarik turun **Project** (dalam gambar 12, struktur ditampilkan sebagai **Android**).

Bisa juga menyesuaikan tampilan file proyek untuk berfokus pada aspek tertentu dari pengembangan aplikasi yang dibuat. Misalnya, memilih tampilan **Problems** dari tampilan proyek yang akan menampilkan tautan ke file sumber yang berisi kesalahan pengkodean dan sintaks yang dikenal, misalnya tag penutup elemen XML tidak ada dalam file tata letak.

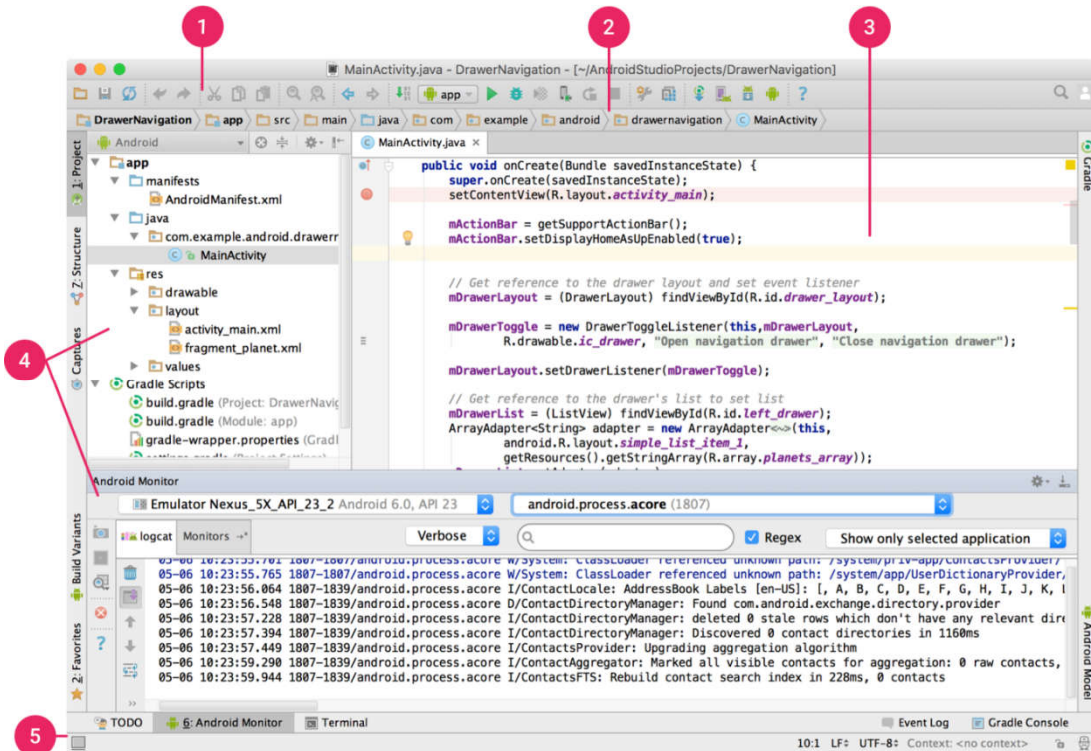


Gambar B.13 Tampilan Problem

### B.2.8.3 Antarmuka Pengguna

Jendela utama Android Studio terdiri dari beberapa bidang logika yang diidentifikasi dalam gambar dibawah ini





Gambar B.14 User GUI


1. **Bilah alat** memungkinkan untuk melakukan berbagai jenis tindakan, termasuk menjalankan aplikasi dan meluncurkan alat Android.
2. **Bilah navigasi** akan membantu menavigasi di antara proyek dan file yang dibuka untuk pengeditan. Di sini tampilan struktur yang terlihat tampak lebih ringkas daripada jendela **Project**.
3. **Jendela editor** merupakan tempat membuat dan mengubah kode. Bergantung pada jenis file saat ini, editor dapat berubah. Misalnya, ketika melihat file tata letak, editor akan menampilkan Layout Editor.
4. **Jendela alat** memberi akses ke tugas-tugas spesifik seperti pengelolaan proyek, penelusuran, kontrol versi, dan banyak lagi, bisa diluaskan dan juga diciutkannya.
5. **Bilah status** menampilkan status proyek dan IDE itu sendiri, serta setiap peringatan atau pesan.

Bisa menata jendela utama untuk memberi ruang layar yang lebih luas dengan menyembunyikan atau memindahkan bilah alat dan jendela alat. Juga bisa menggunakan pintasan keyboard untuk mengakses sebagian besar fitur IDE. Dapat menelusuri seluruh kode sumber, basis data, tindakan, elemen antarmuka pengguna, dan seterusnya setiap saat dengan menekan tombol Shift

dua kali, atau mengeklik kaca pembesar di sudut kanan atas dari jendela Android Studio. Ini akan sangat berguna misalnya saat mencoba menemukan tindakan IDE tertentu yang dilupakan cara memicunya.

#### B.2.8.4 Jendela Alat

Android Studio mengikuti konteks dan secara otomatis memunculkan jendela alat yang relevan saat programmer bekerja. Secara default, alat yang tersering dipakai akan disematkan ke bilah jendela alat di tepi jendela aplikasi.

1. Untuk meluaskan atau menciutkan jendela alat, klik nama alat di bilah jendela alat. Juga bisa menyeret, menyematkan, melampirkan, dan melepaskan jendela alat.
2. Untuk kembali ke tata letak jendela alat default saat ini, klik **Window > Restore Default Layout** atau sesuaikan tata letak default dengan mengeklik **Window > Store Current Layout as Default**.
3. Untuk menampilkan atau menyembunyikan bilah jendela alat, klik ikon jendela  di sudut kiri bawah jendela Android Studio.
4. Untuk menemukan jendela alat spesifik, arahkan ke atas ikon jendela dan pilih jendela alat tersebut dari menu.

Bisa juga menggunakan pintasan keyboard untuk membuka jendela alat. Tabel 1 mencantumkan pintasan jendela paling umum.

Jendela Alat	Windows dan Linux	Mac
Proyek	<b>Alt+1</b>	<b>Command+1</b>
Kontrol Versi	<b>Alt+9</b>	<b>Command+9</b>
Menjalankan	<b>Shift+F10</b>	<b>Control+R</b>
Debug	<b>Shift+F9</b>	<b>Control+D</b>
Pemantauan Android	<b>Alt+6</b>	<b>Command+6</b>
Mengembalikan ke Editor	<b>Esc</b>	<b>Esc</b>

Menyembunyikan Semua Jendela Alat	<b>Control+Shift+F12</b>	<b>Command+Shift+F12</b>
-----------------------------------	--------------------------	--------------------------

**Tabel 1.** Pintasan keyboard ke beberapa jendela alat yang bermanfaat.

Jika ingin menyembunyikan semua bilah alat, jendela alat, dan tab editor, klik **View > Enter Distraction Free Mode**. Ini akan mengaktifkan *Distraction Free Mode*. Untuk keluar dari Distraction Free Mode, klik **View > Exit Distraction Free Mode**. Bisa juga menggunakan *Speed Search* untuk menelusuri dan memfilter di dalam sebagian besar jendela alat dalam Android Studio. Untuk menggunakan Speed Search, pilih jendela alat lalu ketik kueri penelusuran.

Untuk tip selengkapnya, lihat Pintasan Keyboard.

#### B.2.8.5 Pelengkapan Kode

Android Studio memiliki tiga jenis pelengkapan kode, yang bisa diakses memakai pintasan keyboard.

**Tabel 2.** Pintasan keyboard untuk pelengkapan kode.

<b>Tipe</b>	<b>Keterangan</b>	<b>Windows dan Linux</b>	<b>Mac</b>
Pelengkapan Dasar	Menampilkan saran dasar untuk variabel, jenis, metode, ekspresi, dan seterusnya. Jika Anda memanggil pelengkapan dasar dua kali secara berturut-turut, Anda melihat lebih banyak hasil, termasuk anggota pribadi dan anggota statis yang tidak diimpor.	<b>Control+Spasi</b>	<b>Control+Spasi</b>
Pelengkapan Cerdas	Menampilkan opsi relevan berdasarkan konteks. Pelengkapan cerdas	<b>Control+Shift+Spasi</b>	<b>Control+Shift+Spasi</b>

	mengetahui tipe yang diharapkan dan alur data. Jika Anda memanggil Pelengkapan Cerdas dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk rantai.		
Pelengkapan Pernyataan	Membantu Anda melengkapi pernyataan saat ini, menambahkan tanda kurung, tanda kurung siku, tanda kurung kurawal, pemformatan, dsb.	<b>Control+Shift+Enter</b>	<b>Shift+Command+Enter</b>

Melakukan perbaikan cepat dan menunjukkan tindakan langsung dengan menekan **Alt+Enter**. Untuk informasi selengkapnya tentang pelengkapan kode, lihat Pelengkapan Kode.

#### B.2.8.6 Navigasi

Berikut beberapa tip untuk membantu menjelajahi di dalam Android Studio.

1. Beralih antar file yang baru-baru ini diakses memakai tindakan *Recent Files*. Tekan **Control+E** (**Command+E** pada Mac) untuk memunculkan tindakan Recent Files. Secara default, akses yang terakhir dipilih. Bisa juga mengakses jendela alat mana saja melalui kolom kiri dalam tindakan ini.
2. Lihat struktur file saat ini menggunakan tindakan *File Structure*. Munculkan tindakan File Structure dengan menekan **Control+F12** (**Command+F12** pada Mac). Menggunakan tindakan ini, bisa menavigasi dengan cepat ke bagian mana pun dari file Anda saat ini.
3. Telusuri dan navigasikan ke kelas spesifik di proyek menggunakan tindakan *Navigate to Class*. Munculkan tindakan dengan menekan **Control+N** (**Command+O** pada Mac). Navigasikan ke Kelas yang mendukung ekspresi canggih, termasuk CamelHumps, jalur, baris menavigasi ke, nama tengah pencocokan, dan banyak lagi. Jika memanggilnya dua

kali berturut-turut, hasil dari kelas proyek akan ditampilkan.

4. Navigasikan ke file atau folder menggunakan tindakan "Navigate to File\*". Munculkan tindakan Navigate to File dengan menekan **Control+Shift+N** (**Command+Shift+O** pada Mac). Untuk menelusuri folder dan bukan file, tambahkan / di akhir ekspresi.
5. Navigasikan ke metode atau bidang menggunakan tindakan *Navigate to Symbol*. Munculkan tindakan Navigate to Symbol dengan menekan **Control+Shift+Alt+N** (**Command+Shift+Alt+O** pada Mac).
6. Temukan semua bagian dari kode yang merujuk kelas, metode, bidang, parameter, atau pernyataan di posisi kursor saat ini dengan menekan **Alt+F7**.

### B.2.8.7 Gaya dan Pemformatan

Saat mengedit, Android Studio otomatis menerapkan pemformatan dan gaya seperti yang ditetapkan dalam setelan gaya kode. Dapat juga menyesuaikan setelan gaya kode dengan bahasa pemrograman, termasuk menetapkan konvensi untuk tab dan inden, spasi, pembungkusan dan tanda kurung kurawal, dan baris kosong. Untuk menyesuaikan setelan gaya kode, klik **File > Settings > Editor > Code Style (Android Studio > Preferences > Editor > Code Style** pada Mac.)

Meski IDE otomatis menerapkan pemformatan saat programmer bekerja, juga dapat secara eksplisit memanggil tindakan *Reformat Code* dengan menekan **Control+Alt+L** (**Opt+Command+L** pada Mac), atau inden otomatis semua baris dengan menekan **Control+Alt+I** (**Alt+Option+I** pada Mac).

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    mActionBar = getSupportActionBar();  
    mActionBar.setDisplayHomeAsUpEnabled(true);  
}
```

Gambar B.15 Kode sebelum pemformatan.

```
} public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    mActionBar = getSupportActionBar();  
    mActionBar.setDisplayHomeAsUpEnabled(true);  
    // Get reference to the drawer layout and set event listener
```

Formatted 7 lines  
Show reformat dialog: ⌘⇧L

**Gambar B.16** Kode sesudah pemformatan.

#### **B.2.8.8 Dasar-Dasar Kontrol Versi**

Android Studio mendukung berbagai versi sistem kontrol, termasuk Git, GitHub, CVS, Mercurial, Subversion, dan Penyimpanan Google Cloud Source. Setelah mengimpor aplikasi ke dalam Android Studio, gunakan opsi menu Android Studio VCS untuk mengaktifkan dukungan VCS bagi sistem kontrol versi yang diinginkan, membuat penyimpanan, mengimpor file baru ke dalam kontrol versi, dan melakukan pengoperasian kontrol versi lainnya:

1. Dari menu Android Studio **VCS**, klik **Enable Version Control Integration**.
2. Dari menu tarik-turun, pilih sistem kontrol versi yang terkait dengan akar proyek, lalu klik **OK**.

Menu VCS sekarang menunjukkan sejumlah opsi kontrol versi berdasarkan sistem yang dipilih.

#### **B.2.8.9 GridView Layout**

GridViewLayout merupakan salah satu class dari android.widget. Yang dapat menampilkan data dalam bentuk grid. Data yang ditampilkan dapat berupa teks, gambar, dan lain-lain.

#### **B.2.8.10 ScrollView**

ScrollView adalah salah satu jenis komponen yang digunakan pada user interface di layout.xml. ScrollView berbeda dengan LinearLayout dan RelativeLayout, ScrollView berfungsi untuk memuat suatu layar agar dapat ditampilkan secara penuh, serta akan menambahkan scrollbar di pojok kanan layar.

# I

## ANDROID BASIC APPLICATION



# 1

## BAB 1

### ANDROID BASIC APPLICATION

#### 1. 1. Komponen-komponen pada Aplikasi Android



Gambar 1.1. Komponen Aplikasi Android

Komponen aplikasi adalah bagian penting dari sebuah aplikasi Android. Komponen ini harus terhubung dengan `AndroidManifest.xml` file yang menggambarkan setiap komponen dari aplikasi dan bagaimana mereka berinteraksi.

Ada empat komponen utama yang dapat digunakan dalam aplikasi Android:

Table 1.1. Komponen utama dalam aplikasi Android

Komponen	Deskripsi
Activity	Untuk Menampilkan antarmuka aplikasi
Service	Untuk menjalankan proses aplikasi di belakang layar



Broadcast Receiver	Untuk komunikasi antar activity, aplikasi dan device
Content Provider	Untuk manajemen data dan database

### 1.1.1 Activity

Suatu activity akan menyajikan user interface (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bisa jadi hanya memiliki satu activity, tetapi umumnya aplikasi mempunyai banyak activity, tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut.

Satu activity biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai user interface saat aplikasi diperlihatkan kepada user. Untuk pindah dari satu activity ke activity lain kita dapat melakukannya dengan satu event, misalnya klik tombol, memilih opsi atau menggunakan triggers tertentu.

Secara hirarki sebuah windows activity dinyatakan dengan method `Activity setContentView()`. Content view adalah objek yang berada pada root hirarki. Sebuah kegiatan mewakili satu layar dengan user interface. Sebagai contoh, sebuah aplikasi email mungkin memiliki satu aktivitas yang menunjukkan daftar email baru, kegiatan lain untuk menulis email, dan aktivitas lain untuk membaca email.

Activity diimplementasikan sebagai subclass dari class Activity sebagai contoh:

```
public class MainActivity extends Service {
    }
```

### 1. 1. 2. Service

Service tidak memiliki Graphic User Interface (GUI), tetapi service berjalan secara background, sebagai contoh dalam memainkan music, service mungkin memainkan music ataumengambil data dari jaringan, tetapi setiap service harus berada dalam kelas induknya.

Misalnya media player sedang memutar playlist dari list yang ada, aplikasi ini akan memiliki dua atau lebih activity yang memungkinkan user untuk memilih lagu misalnya, atau menulis sms sambil player sedang jalan.

Untuk menjaga music tetap dijalankan, activity player dapat menjalankan service. Service dijalankan pada thread utama dari proses aplikasi. Service diimplementasikan sebagai subclass dari class Service sebagai contoh:

```
public class MyService extends Service {  
    }  
}
```

### 1. 1. 3. Broadcast Receiver

Broadcast receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh broadcast seperti notifikasi zona waktu berubah, baterai low, gambar telah selesai diambil oleh kamera, atau perubahan referensi bahasa yang digunakan.

Aplikasi juga dapat menginisiasi broadcast misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah di unduh ke perangkat dan siap untuk digunakan.

Broadcast receiver tidak memiliki user interface (UI), tetapi memiliki sebuah activity untuk merespon sebuah notifikasi yang mereka terima, atau mungkin menggunakan notification manager untuk memberi tahu kepada pengguna, seperti lampu latar atau vibrating (getaran perangkat). Sebuah broadcast receiver diimplementasikan sebagai subclass dari BroadcastReceiver dan setiap broadcast receiver disampaikan sebagai object Intent.

```
public class MyReceiver extends BroadcastReceiver {  
    public void onReceive(context,intent) {  
    }  
}
```

### 1. 1. 4. Content Providers

ContentProvider adalah penyedia konten dari satu aplikasi ke aplikasi lain atas perintah tertentu. Perintah tersebut ditangani oleh sebuah method dari class ContentResolver. Data dapat disimpan dalam sistem file, database atau di tempat lainnya. ContentProvider

diimplementasikan sebagai subclass dari class `ContentProvider` dan harus menerapkan satu set standar API yang memungkinkan aplikasi lain untuk melakukan transaksi.

```
public class MyContentProvider extend Content
Provider{
public void onCreate(){} }
```

## 1. 2.   Komponen-komponen pada Layout Android

Layout merupakan salah satu komponen user interface pada aplikasi Android yang berperan sebagai suatu grup yang terstruktur dari beberapa komponen user interface (view). Layout ini menggabungkan beberapa view seperti Button dan TextView atau yang lainnya yang disusun menjadi sebuah halaman aplikasi Android.

Komponen user interface ini disebut widget. Beberapa widget dasar yang akan dibahas adalah: TextView, Button, Image, EditText, ListView. Semua merupakan turunan dari kelas View, sehingga sering juga disebut View. Untuk widget yang menerima input, disebut juga input control. (kadang memang membingungkan untuk urusan penamaan di Android ini)

Setiap widget memiliki property atau atribut yang mengatur bagaimana widget itu ditampilkan, seperti tinggi dan lebar widget. Property ini dapat diatur melalui xml layout, property editor atau melalui app. Beberapa widget memiliki beberapa event yang ter-trigger berdasarkan aksi dari pengguna, misalnya event click pada button.

### 1. 2. 1.    TextView

Komponen TextView ini dapat memunculkan tulisan di layar tetapi ketika program sudah dijalankan, pengguna tidak dapat mengubah tulisan tersebut. Tampilan atau isi TextView juga dapat diedit melalui properties textview yang berada di kolom sebelah kanan, diantaranya :

Table 1.2. Text View

Tools	Fungsi

Id	berfungsi sebagai penamaan unix dari textview
Layout_width	untuk mengatur lebarnya layout textview
Layout_margin	untuk mengatur tingginya layout textview
Text	untuk menuliskan isi ext yang akan ditampilkan
Background	untuk mengatur warna background tulisan atau bentuk tulisan yang ditampilkan
TextSize	untuk mengatur ukuran text
Text color	untuk mengatur warna tulisan yang akan ditampilkan

### 1. 2. 2. EditText

Komponen EditText ini merupakan subclass dari TextView karena dapat memunculkan tulisan di layar, akan tetapi pada EditText pengguna dapat mengubah tulisan tersebut. EditText juga digunakan untuk menerima input dari pengguna. Pada palette telah disediakan berbagai jenis EditText diantaranya :

- a. Plain Text
- b. Person Name
- c. Password
- d. Password (Numeric)
- e. E-mail
- f. Phone
- g. Postal Address
- h. Multiline Text
- i. Time
- j. Date
- k. Number
- l. Number (Signed)
- m. Number (Decimal)
- n. AutoCompleteTextView

o. MultiAutoCoplatformTextView

### 1. 2. 3. Button

Button adalah tombol yang dapat menampilkan, memanggil atau mengirim data. Button merupakan turunan dari TextView sehingga yang berlaku di TextView juga berlaku di button. Tambahan property yang penting adalah onClick. Pada class ini hanya dapat menangani event ketika button di tekan.

### 1. 2. 4. ImageButton dan ImageView

ImageButton adalah turunan dari button, gunakan widget ini jika Anda ingin menggunakan image sebagai pengganti tulisan pada button. Sedangkan ImageView dapat digunakan untuk menampilkan image.



Gambar 1.2 Icon

Berbeda dengan program pada desktop, app di Android dapat berjalan di device dengan berbagai resolusi. Oleh karena itu, perlu menyiapkan gambar untuk berbagai kategori resolusi. Hal ini untuk mencegah gambar yang digunakan terlihat pecah atau terlalu kecil pada device dengan resolusi berbeda. Ada lima jenis resolusi: mdpi untuk medium (160dpi), hdpi untuk high density (240dpi) dan xhdpi untuk ekstra high density (320dpi) serta xxhdpi untuk resolusi 480dpi dan xxxhdpi untuk 640.

### 1. 2. 5. ListView

ListView berfungsi untuk menampilkan list dari item-item secara vertikal. Daftar item secara otomatis dimasukkan ke dalam list menggunakan Adapter yang dapat mengambil konten atau isi dari source/sumber seperti array atau database.

### 1. 2. 6. Message

Message box adalah pesan yang muncul dalam bentuk popup apa bila suatu peristiwa atau event terjadi. Messages box yang di kenal di android terbagi dua jenis yaitu :

a) Toast

Adalah message box yang muncul dalam waktu sebentar dan kemudian hilang tanpa berinteraksi dengan pengguna, biasa ini merupakan fokus dari activity yang active.

b) Alert

Adalah message box yang mempunyai interaksi dengan user, biasa terdiri dari 3 bagian yaitu:

- setMessage() berisi tulisan pesan/message
- setTitle()/setIcon(), konfigurasi title/icon dari dialog box.
- setPositiveButton(), setNegativeButton(), setNeutralButton(), berisi posisi serta metode kliknya.

## 1.3 Mengaktifkan Komponen

Tiga dari empat tipe komponen—aktivitas, layanan, dan penerima siaran—diaktifkan oleh pesan asinkron yang disebut intent. Intent saling mengikat setiap komponen saat waktu proses (sebagai pembawa pesan yang meminta aksi dari komponen lain), baik komponen itu milik aplikasi sendiri atau milik aplikasi lain.

Intent dibuat dengan objek Intent, yang mendefinisikan pesan untuk mengaktifkan komponen tertentu atau komponen tipe komponen tertentu—masing-masing intent bisa eksplisit atau implisit.

Untuk aktivitas dan layanan, intent mendefinisikan aksi yang akan dilakukan (misalnya, untuk "melihat" atau "mengirim" sesuatu) dan mungkin menetapkan URI data untuk ditindak lanjuti (salah satu hal yang mungkin perlu diketahui oleh komponen yang akan dimulai). Misalnya, intent mungkin menyampaikan permintaan suatu aktivitas untuk menampilkan gambar atau membuka laman web. Dalam beberapa kasus dapat memulai aktivitas untuk menerima hasil, dalam hal ini, aktivitas juga akan mengembalikan hasil dalam Intent (misalnya Anda bisa mengeluarkan intent agar pengguna bisa memilih kontak pribadi dan memintanya dikembalikan

kepada Anda—intent yang dikembalikan menyertakan URI yang menunjuk ke kontak yang dipilih).

Untuk penerima siaran, intent hanya mendefinisikan pengumuman yang sedang disiarkan (misalnya, siaran untuk menunjukkan baterai perangkat hampir habis hanya menyertakan string aksi yang menunjukkan "baterai hampir habis").

Tipe komponen lainnya dan penyedia materi, tidak diaktifkan oleh intent. Melainkan diaktifkan saat ditargetkan oleh permintaan dari ContentResolver. Resolver materi menangani semua transaksi langsung dengan penyedia materi sehingga komponen yang melakukan transaksi dengan penyedia tidak perlu dan sebagai gantinya memanggil metode pada objek ContentResolver. Ini membuat layer abstraksi antara penyedia materi dan komponen yang meminta informasi (demi keamanan).

Ada beberapa metode terpisah untuk mengaktifkan masing-masing tipe komponen:

1. memulai aktivitas (atau memberinya pekerjaan baru) dengan meneruskan Intent ke `startActivity()` atau `startActivityForResult()` (bila ingin aktivitas mengembalikan hasil).
2. memulai layanan (atau memberikan petunjuk baru ke layanan yang sedang berlangsung) dengan meneruskan Intent ke `startService()`. Atau Anda bisa mengikat ke layanan dengan meneruskan Intent ke `bindService()`.
3. memulai siaran dengan meneruskan Intent ke metode-metode seperti `sendBroadcast()`, `sendOrderedBroadcast()`, atau `sendStickyBroadcast()`.
4. Anda bisa melakukan kueri ke penyedia materi dengan memanggil `query()` pada ContentResolver.

## **1.4 File Manifest**

Sebelum sistem Android bisa memulai komponen aplikasi, sistem harus mengetahui bahwa komponen memang ada dengan membaca file `AndroidManifest.xml` aplikasi (file "manifest"). Aplikasi Anda harus mendeklarasikan semua komponennya dalam file ini, yang harus menjadi akar dari direktori proyek aplikasi.

Manifes melakukan banyak hal selain mendeklarasikan komponen aplikasi, seperti:

1. Mengidentifikasi izin pengguna yang diperlukan aplikasi, seperti akses Internet atau akses-baca ke kontak pengguna.
2. Mendeklarasikan API Level minimum yang diperlukan aplikasi, berdasarkan API yang digunakan aplikasi.
3. Mendeklarasikan fitur perangkat keras dan perangkat lunak yang diperlukan aplikasi, seperti kamera, layanan Bluetooth, atau layar multisentuh.
4. Pustaka API aplikasi perlu ditautkan (selain Android framework API), seperti pustaka Google Maps.
5. Dan lainnya

## 1.5 Pengaplikasian sederhana aplikasi android

### 1.5.1 Sequential Search

Sequential Search merupakan salah satu metode pencarian (searching) yang digunakan untuk mencari sebuah data yang biasa disebut key, secara bertahap (sequence) dari data awal sampai data akhir, yang berasal dari kumpulan data, dan tidak harus data-data tersebut terurut. Yang dimaksud key ini adalah sebuah bagian dari kumpulan data, untuk perbandingan selama proses pencarian berlangsung. Misal kita ingin mencari angka atau key 4, dari tipe data *integer* pada variable array numbers { 2,6,8,4,5,}, Maka proses pencarian *Sequential Search*, akan dimulai dari index 0, 1, 2 yang dimana dari 2,6,8, ketika sudah ketemu dan cocok, maka proses akan dihentikan.

Contoh Algoritma dari **Sequential Search**

INPUT: List of size N. Target value T
OUTPUT: Position of T in the list I
BEGIN
1. Set FOUND to false
Set I to 0
2. While ( $I \leq N$ ) and (FOUND is false)
If List [I] = T
FOUND = true
Else



```
I=I+1  
END  
3. If FOUND is false  
T is not present in List.  
END
```

### 1.5.2 Simple Browse

Browser adalah salah satu Aplikasi yang sangat populer ,karena kita bisa mengakses dan menjelajah internet lewat Aplikasi ini , banyak digunakan untuk platform Desktop ataupun untuk platform Mobile. Khususnya untuk perangkat Android ,Browser yang Sangat Populer digunakan di kalangan user (pengguna) ,seperti contohnya , Chrome ,Firefox, UC Browser ,Opera dll. Nah yang pasti tiap Aplikasi Browser ,memiliki kelebihan masing-masing ,entah itu dari segi kecepatan ,performa ,fitur untuk hemat kouta dll.

Simple Browser memanfaatkan library Webkit, dengan beberapa fitur yang disematkan yaitu refresh, stp, go bak, dan forward.

# 2

## ANDROID MULTIMEDIA



### 2.1 Multimedia

Multimedia adalah suatu sarana (media) yang didalamnya terdapat perpaduan (kombinasi) berbagai bentuk elemen informasi, seperti teks, graphics, animasi, video, interaktif maupun suara sebagai pendukung untuk mencapai tujuannya yaitu menyampaikan informasi atau sekedar memberikan hiburan bagi target audiens-nya. Multimedia sering digunakan dalam dunia hiburan seperti game. Kata multimedia itu sendiri berasal dari kata multi (Bahasa Latin) yang berarti banyak dan kata media (Bahasa Latin) yang berarti sesuatu yang dipakai untuk menyampaikan sesuatu.

Multimedia dapat dikategorikan menjadi 2 macam, yaitu multimedia linier dan multimedia interaktif. Multimedia linier adalah suatu multimedia yang tidak dilengkapi dengan alat pengontrol apapun yang dapat dioperasikan oleh pengguna. Multimedia ini berjalan sekuensial (berurutan / lurus), contohnya TV dan film. Sedangkan multimedia interaktif adalah suatu multimedia yang dilengkapi dengan alat pengontrol (atau alat bantu berupa komputer, mouse, keyboard dan lain-lain) yang dapat dioperasikan oleh pengguna, sehingga pengguna dapat memilih apa yang diinginkan untuk proses selanjutnya. Contohnya seperti aplikasi game. Multimedia interaktif menggabungkan dan mensinergikan semua media yang terdiri dari teks, grafik, audio, dan interaktivitas (rancangan).

Multimedia dapat disajikan dalam beberapa metode, antara lain :

1. Berbasis kertas (Paper-based), contoh : Buku, Majalah, Brosur.
2. Berbasis cahaya (Light-based), contoh : Slideshows, Transparansi.
3. Berbasis suara (Audi-based), contoh : CD Players, Tape recorder, Radio.
4. Berbasis gambar bergerak (Moving-image-based), contoh : Televisi, VCR (Video Cassete Recorder, Film).

5. Berbasis Digital (Digitally-based), contoh : Komputer).

Multimedia dimanfaatkan juga dalam dunia pendidikan dan bisnis. Di dunia pendidikan, multimedia digunakan sebagai media pengajaran, baik dalam kelas maupun secara sendiri-sendiri. Di dunia bisnis, multimedia digunakan sebagai media profil perusahaan, profil produk, bahkan sebagai media kios informasi dan pelatihan dalam sistem e-learning.

Multimedia Programmer ialah Programmer atau orang yang menguasai penggunaan teknologi dan pengembangan aplikasi berbasis multimedia. Atau orang yang berkecimpung di bidang Multimedia Programming. Contoh aplikasi pengolah multimedia yaitu Corel Draw, Adobe Photoshop, Power Sound Editor, Adobe After Effect, dan lain-lain.

Android Multimedia Programming ialah pemrograman multimedia yang berbasis di platform android. Untuk membuat aplikasi multimedia di android dapat dibuat dengan aplikasi Android Studio, Eclipse, NetBeans, Andromo, AppInventor, dll.

### **2.1.1 Audio**

Audio atau kita kenal dengan istilah suara adalah kompresi mekanikal atau gelombang longitudinal yang merambat melalui medium. Medium atau zat perantara ini dapat berupa zat cair, padat, gas. Jadi, gelombang bunyi dapat merambat misalnya di dalam air, batu bara, atau udara. Kebanyakan suara adalah merupakan gabungan berbagai sinyal, tetapi suara murni secara teoritis dapat dijelaskan dengan kecepatan osilasi atau frekuensi yang diukur dalam Hertz (Hz) dan amplitudo atau kenyaringan bunyi dengan pengukuran dalam desibel. Manusia mendengar bunyi saat gelombang bunyi, yaitu getaran di udara atau medium lain, sampai ke gendang telinga manusia. Batas frekuensi bunyi yang dapat didengar oleh telinga manusia kira-kira dari 20 Hz sampai 20 kHz pada amplitudo umum dengan berbagai variasi dalam kurva responsnya. Suara di atas 20 kHz disebut ultrasonik dan di bawah 20 Hz disebut infrasonik.

Socrates pernah berujar bahwa suara adalah imitasi terbaik bagi pikiran maka suara adalah media terbaik untuk menyampaikan informasi. Bagi Socrates text adalah imitasi dari suara, dengan demikian sebagai penyampai pikiran text bukanlah media yang ideal karena ia

hanyalah imitasi dari suatu imitasi. Pendapat Socrates mungkin ada benarnya karena suara adalah media yang secara natural telah dimiliki oleh manusia sehingga suara adalah media yang paling alami. Guru di kelas pun lebih banyak mengandalkan suara baik ketika memberikan materi atau melakukan motivasi bagi siswa-siswanya. Jika untuk percakapan secara langsung audio adalah media yang simpel dan alami maka tidak demikian halnya ketika digunakan di dalam komputer. Penggunaan suara di dalam komputer berlangsung belakangan sesudah penggunaan text.

Kelebihan suara di dalam multimedia pembelajaran :

1. Sangat cocok bila digunakan sebagai media untuk memberikan motivasi.
2. Untuk materi- materi tertentu suara sangat cocok karena mendekati keadaan asli dari materi (misal pelajaran mengenai mengenal suara-suara binatang)
3. Membantu pembelajar fokus pada materi yang dipelajari karena pembelajar cukup mendengarkan tanpa melakukan aktivitas lain yang menuntut konsentrasi . Bandingkan dengan pembelajar yang melihat teks di layar komputer. Dalam hal ini pembelajar melakukan multi aktivitas yakni : membaca teks pada layar (yang tidak semudah membaca pada buku), mencari kata-kata kunci (keyword) dari materi, dan menggerakkan tangan, seperti melakukan klik mouse untuk menggulung layar saat ingin melihat bagian teks yang tak terlihat pada layar.

Kelemahan audio :

1. Memerlukan tempat penyimpanan yang besar di dalam komputer.
2. Memerlukan software dan hardware yang spesifik (dan mungkin mahal) agar suara dapat disampaikan melalui komputer.

### **2.1.2 Video**

Video adalah teknologi untuk menangkap, merekam, memproses, mentransmisikan dan menata ulang gambar bergerak. Biasanya menggunakan film seluloid, sinyal elektronik, atau media digital. Berkaitan dengan “penglihatan dan pendengaran” Aplikasi video pada multimedia mencakup banyak aplikasi yaitu:

1. Entertainment: roadcast TV, VCR/DVD recording
2. Interpersonal: video telephony, video conferencing
3. Interactive: windows

Digital video adalah jenis sistem video recording yang bekerja menggunakan sistem digital dibandingkan dengan analog dalam hal representasi videonya. Biasanya digital video direkam dalam tape, kemudian didistribusikan melalui optical disc, misalnya VCD dan DVD.

Video digital memiliki keuntungan:

1. Interaktif Video digital disimpan dalam media penyimpanan random contohnya magnetic/optical disk. Sedangkan video analog menggunakan tempat penyimpanan sekuensial, contohnya magnetic disc/kaset video. Video digital dapat memberikan respon waktu yang cepat dalam mengakses bagian manapun dari video.
2. Mudah dalam proses edit
3. Kualitas: sinyal analog dari video analog akan mengalami penurunan kualitas secara perlahan karena adanya pengaruh kondisi atmosfer. Sedangkan video digital kualitasnya dapat diturunkan menggunakan teknik kompresi.
4. Transmisi dan distribusi mudah karena dengan proses kompresi, maka video digital dapat disimpan dalam CD, ditampilkan pada web, dan ditransmisikan melalui jaringan.

Untuk membuat project video sederhana dalam android, hanya dibutuhkan source xml berupa widget “VideoView”, dan sebuah file java untuk mendeklarasikan widget VideoView yang digunakan. Untuk video ada 2 cara untuk menampilkannya di emulator : pertama, sama seperti ingin memunculkan suara, yaitu diletakkan didalam folder raw. Kedua, upload file video ke dalam emulator yang akan digunakan untuk melihat hasil output video tersebut. Pada project ini kita akan lakukan pada cara pertama karena caranya lebih mudah.

### **2.1.3 Gambar**

Gambar adalah hasil yang terlihat diam baik berwarna maupun hitam-putih yang dihasilkan oleh kamera yang merekam suatu objek atau kejadian atau keadaan pada waktu

tertentu. Selain itu foto dan potret juga sering digunakan sebagai kiasan. Misalnya: “Foto/potret masa silam itu sering muncul kembali dalam benaknya”. Dalam hal ini, foto/ potret berarti bayangan, gambaran, atau kenangan.

Selain definisi diatas dan makna kiasan yang dapat dikiaskan dengan kata foto. Secara kategorisasi foto juga harus dibedakan menjadi beraga. Kategorisasi ini bertujuan untuk memudahkan pembuatan dan pemanfaatannya, sesuai dengan standar kualitas bagi masing-masing keperluan. Ada banyak sekali kategori foto, antara lain: foto keluarga, foto dokumentasi, foto resmi, foto salon, foto seni, foto kriminal, foto kedokteran (foto sinar X/rontgen), foto infra merah, foto satelit, foto udara, foto mikro, foto jurnalistik, dan lain-lain.

Selain itu, ada pula kategori foto berdasarkan ukuran. Misalnya pas foto, foto seluruh badan, foto KTP, paspor, foto postcard, dan lain-lain. Ada pula perbedaan sesuai dengan jenis kameranya. Misalnya foto analog (dengan film) dan foto digital pada kamera smartphone. Ada dua cara untuk menampilkan suatu gambar, menggunakan widget ImageView atau menggunakan ImageButton. ImageButton adalah turunan dari button, gunakan widget ini jika Anda ingin menggunakan image sebagai pengganti tulisan pada button. Sedangkan ImageView dapat digunakan untuk menampilkan image.

Berbeda dengan program pada desktop, app di Android dapat berjalan di device dengan berbagai resolusi. Oleh karena itu, perlu menyiapkan gambar untuk berbagai kategori resolusi. Hal ini untuk mencegah gambar yang digunakan terlihat pecah atau terlalu kecil pada device dengan resolusi berbeda. Ada lima jenis resolusi: mdpi untuk medium (160dpi), hdpi untuk high density (240dpi) dan xhdpi untuk ekstra high density (320dpi) serta xxhdpi untuk resolusi 480dpi dan xxxhdpi untuk 640.

# 3

## ANDROID ANIMATION & GAME





### 3.1 Pengertian Animasi

Animasi berasal dari bahasa latin yaitu “anima” yang berarti jiwa, hidup, semangat. Sedangkan karakter adalah orang, hewan maupun objek nyata lainnya yang dituangkan dalam bentuk gambar 2D maupun 3D. sehingga karakter animasi secara dapat diartikan sebagai gambar yang memuat objek yang seolah-olah hidup, disebabkan oleh kumpulan gambar itu berubah beraturan dan bergantian ditampilkan. Objek dalam gambar bisa berupa tulisan, bentuk benda, warna dan spesial efek.

Pengertian Animasi Menurut Ibiz Fernandes dalam bukunya *Macromedia Flash Animation & Cartooning: A creative Guide*, animasi definisikan sebagai berikut : “Animasi adalah sebuah proses merekam dan memainkan kembali serangkaian gambar statis untuk mendapatkan sebuah ilusi pergerakan.”

Berdasarkan arti harfiah, Animasi adalah menghidupkan, yaitu usaha untuk menggerakkan sesuatu yang tidak bisa bergerak sendiri. Secara garis besar, animasi komputer dibagi menjadi dua kategori, yaitu:

- a. *Computer Assisted Animation*, animasi pada kategori ini biasanya menunjuk pada system animasi 2 dimensi, yaitu mengkomputerisasi proses animasi tradisional yang menggunakan gambaran tangan. Komputer digunakan untuk pewarnaan, penerapan virtual kamera dan penataan data yang digunakan dalam sebuah animasi.
- b. *Computer Generated Animation*, pada kategori ini biasanya digunakan untuk animasi 3 dimensi dengan program 3D seperti 3D Studio Max, Maya, Autocad dll.

Secara garis besar, animasi computer dibagi menjadi dua kategori, yaitu:

- a. Computer Assisted Animation, animasi pada kategori ini biasanya menunjuk pada system animasi 2 dimensi, yaitu mengkomputerisasi proses animasi tradisional yang menggunakan gambaran tangan. Computer digunakan untuk pewarnaan, penerapan virtual kamera dan penataan data yang digunakan dalam sebuah animasi.
- b. Computer Generated Animation, pada kategori ini biasanya digunakan untuk animasi 3 dimensi dengan program 3D seperti 3D Studio Max, Maya, Autocad dll.

### **3.2 Animasi Secara Multimedia**

Menurut Rosch definisi animasi adalah kombinasi dari computer dan video, menurut McCornick multimedia secara umum merupakan kombinasi tiga elemen, yaitu suara, gambar dan teks. Menurut Turban, dkk multimedia adalah kombinasi dari paling sedikit dua media input atau output dari data, media ini berupa audio (suara, musik), animasi, video, teks, grafik, dan gambar.

Menurut Robin dan Linda multimedia merupakan alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, audio, dan gambar video. Definisi lain dari multimedia yaitu dengan menempatkan dalam konteks seperti yang dilakukan Hofstetter, multimedia adalah pemanfaatan computer untuk membuat dan menggabungkan teks, grafik, audio, gambar bergerak (video dan animasi) dengan menggabungkan link dan tools yang memungkinkan navigasi, berinteraksi, berkreasi, dan berkomunikasi.

### **3.3 Prinsip Dasar Animasi**

#### **a. Jenis – jenis animasi**

Dilihat dari tehnik pembuatannya animasi yang ada saat ini dapat dikategorikan menjadi 3, yaitu:

1. Animasi Stop-motion (*Stop Motion Animation*)
2. Animasi Tradisional (*Traditional animation*)
3. Animasi Komputer (*Computer Graphics Animation*)

### 1. Animasi *Stop-Motion*

Stop-motion animation sering pula disebut claymation karena dalam perkembangannya, jenis animasi ini sering menggunakan clay (tanah liat) sebagai objek yang digerakkan. Teknik stop-motion animation merupakan animasi yang dihasilkan dari pengambilan gambar berupa obyek (boneka atau yang lainnya) yang digerakkan setahap demi setahap. Dalam pengerjaannya teknik ini memiliki tingkat kesulitan dan memerlukan kesabaran yang tinggi.

### 2. Animasi Tradisional (*Traditional animation*)

Tradisional animasi adalah teknik animasi yang paling umum dikenal sampai saat ini. Dinamakan tradisional karena teknik animasi inilah yang digunakan pada saat animasi pertama kali dikembangkan. Tradisional animasi juga sering disebut cel animation karena teknik pengerjaannya dilakukan pada celluloid transparent yang sekilas mirip sekali dengan transparansi OHP yang sering kita gunakan. Pada pembuatan animasi tradisional, setiap tahap gerakan digambar satu persatu di atas cel.

### 3. Animasi Komputer

Sesuai dengan namanya, animasi ini secara keseluruhan dikerjakan dengan menggunakan komputer. Dari pembuatan karakter, mengatur gerakan “pemain” dan kamera, pemberian suara, serta special efeknya semuanya dikerjakan dengan komputer.

Dengan animasi komputer, hal-hal yang awalnya tidak mungkin digambarkan dengan animasi menjadi mungkin dan lebih mudah. Sebagai contoh perjalanan wahana ruang angkasa ke suatu planet dapat digambarkan secara jelas, atau proses terjadinya tsunami.

Perkembangan teknologi komputer saat ini, memungkinkan orang dengan mudah membuat animasi. Animasi yang dihasilkan tergantung keahlian yang dimiliki dan software yang digunakan.

### 3.4 Syarat yang diperhatikan saat membuat animasi

#### 1. *Timing* (Waktu)

Ini menentukan apakah gerakan tersebut alami atau tidak. Misalkan gerakan orang berjalan terlalu lambat, sedangkan latar belakang terlalu cepat bergerak. Atau bola yang memantul ke tanah, tetapi sebelum memantul, efek suara pantulan sudah terdengar lebih dahulu. Jadi timing ini lebih kepada sinkronisasi antara elemen-elemen animasi. Grim Natwick, seorang animator Disney pernah berkata, “Animasi adalah tentang timing dan spacing”. Timing adalah tentang menentukan waktu kapan sebuah gerakan harus dilakukan, sementara spacing adalah tentang menentukan percepatan dan perlambatan dari bermacam-macam jenis gerak.

Contoh Timing: Menentukan pada detik seberapa sebuah obyek/karakter berjalan sampai ke tujuan atau berhenti.

Contoh Spacing: Menentukan kepadatan gambar (yang pada animasi akan berpengaruh pada kecepatan gerak)

#### 2. *Ease In* dan *Ease Out* (Percepatan dan Perlambatan)

Prinsip ini juga paling banyak digunakan dalam animasi. Ketika bola di lempar ke atas, gerakan tersebut harus semakin lambat. Dan bola jatuh akan semakin cepat. Atau ketika mobil berhenti, pemberhentian tersebut harus secara perlahan-lahan melambat, tidak bisa langsung berhenti. *Slow In* dan *Slow Out* menegaskan bahwa setiap gerakan memiliki percepatan dan perlambatan yang berbeda-beda. *Slow in* terjadi jika sebuah gerakan diawali secara lambat kemudian menjadi cepat. *Slow out* terjadi jika sebuah gerakan yang relatif cepat kemudian melambat.

#### 3. *Arcs* (Lengkungan)

Banyak hal tidak bergerak secara garis lurus. Bola saja dilempar tidak akan pernah lurus, pasti ada sedikit pergeseran. Jadi usahakan gerakan objek anda tidak sempurna, agak “dirusak” sedikit sehingga terlihat alami.

Pada animasi, sistem pergerakan tubuh pada manusia, binatang, atau makhluk hidup lainnya bergerak mengikuti pola/jalur (maya) yang disebut Arcs. Hal ini memungkinkan mereka bergerak secara 'smooth' dan lebih realistik, karena pergerakan mereka mengikuti suatu pola yang berbentuk lengkung (termasuk lingkaran, elips, atau parabola). Sebagai contoh, Arcs ditunjukkan pada lintasan tangan saat melempar bola dan lintasan gerak bola di udara.

#### **4. Follow Through and Overlapping Action (Gerakan penutup sebelum benar-benar diam).**

Follow through adalah tentang bagian tubuh tertentu yang tetap bergerak meskipun seseorang telah berhenti bergerak. Misalnya, rambut yang tetap bergerak sesaat setelah melompat. Overlapping action secara mudah bisa dianggap sebagai gerakan saling-silang. Maksudnya, adalah serangkaian gerakan yang saling mendahului (overlapping).

Contoh : Kelinci yang melompat. Sesaat setelah melompat telinganya masih bergerak-gerak meskipun gerakan utama melompat telah dilakukan. Prinsip ini ingin menggambarkan perilaku karakter sebelum menyelesaikan suatu tindakan. Misalkan saat seseorang melempar bola, gerakan setelah melempar bola (Follow Through) tersebut adalah menunjukkan mimik muka senang karena puas telah melempar bola. Kemudian yang disebut Overlapping action adalah gerakan baju atau rambut yang bergerak akibat gerakan tersebut.

Jadi animasi bukan sekedar asal bergerak, tetapi membuatnya hidup dengan hal-hal detail seperti ini. Banyak yang sangat detail bisa menggambar karakter, tetapi banyak yang gagal dalam menganimasikan karena karakter yang digambar terlalu rumit untuk dianimasikan.

#### **5. Secondary Action (Gerakan Pelengkap)**

Ini bukanlah gerakan yang sebenarnya, misalkan saat di ruang tunggu dokter, ada tokoh utama yang sedang membaca, tetapi di latar belakang ada pemeran pendukung seperti orang merokok, sedang mengobrol atau apapun yang membuatnya terlihat alami. Secondary action adalah gerakan-gerakan tambahan yang dimaksudkan untuk memperkuat gerakan utama supaya sebuah animasi tampak lebih realistik. Secondary action tidak dimaksudkan untuk menjadi 'pusat perhatian' sehingga mengaburkan atau mengalihkan perhatian dari gerakan utama. Kemunculannya lebih berfungsi memberikan emphasize untuk memperkuat gerakan utama

Contoh: Ketika seseorang sedang berjalan, gerakan utamanya tentu adalah melangkahakan kaki sebagaimana berjalan seharusnya. Namun sambil berjalan ‘seorang’ figur atau karakter animasi dapat sambil mengayun-ayunkan tangannya. Gerakan mengayun-ayunkan tangan inilah yang disebut secondary action untuk gerakan berjalan.

## **6. Squash and Strecth (Kelenturan suatu objek)**

Bola yang ketika jatuh agak sedikit gepeng menunjukkan kelenturan bola tersebut. Atau ketika orang melompat dan jatuh, kakinya agak sedikit lentur. Squash and strecth adalah upaya penambahan efek lentur (plastis) pada objek atau figur sehingga seolah-olah ‘memuai’ atau ‘menyusut’ sehingga memberikan efek gerak yang lebih hidup. Penerapan squash and stretch pada figur atau benda hidup (misal: manusia, binatang, creatures) akan memberikan ‘enhancement’ sekaligus efek dinamis terhadap gerakan/action tertentu, sementara pada benda mati (misal : gelas, meja, botol) penerapan squash and stretch akan membuat mereka (benda-benda mati tersebut) tampak atau berlaku seperti benda hidup.

Contoh ketika sebuah bola dilemparkan. Pada saat bola menyentuh tanah maka dibuat seolah-olah bola yang semula bentuknya bulat sempurna menjadi sedikit lonjong horizontal, meskipun kenyataannya keadaan bola tidak selalu demikian. Hal ini memberikan efek pergerakan yang lebih dinamis dan ‘hidup’.

## **7. Exaggeration (Melebih-lebihkan)**

Animasi bisa dilebih-lebihkan dengan musik, latar belakang atau gambar. Orang digambarkan dengan mata besar yang menunjukkan keterkejutan. Ini bisa kita lihat di film-film kartun jepang, bagaimana orang berlari tetapi ada gambar seekor elang besar sebagai latarnya untuk menunjukkan kecepatan lari orang tersebut.

Exaggeration merupakan upaya mendramatisir animasi dalam bentuk rekayasa gambar yang bersifat hiperbolis. Dibuat sedemikian rupa sehingga terlihat sebagai bentuk ekstrimitas ekspresi tertentu dan biasanya digunakan untuk keperluan komedik. Seringkali ditemui pada film-film animasi anak-anak (segala usia) seperti Tom & Jerry, Donald Duck, Mickey Mouse, Sinchan, dsb. Contoh : Tubuh Donald duck melayang mengikuti sumber asap saat hidung Donald cuck mencium aroma masakan/makanan lezat.

## **8. Straight Ahead and Pose to Pose**

Prinsip Straight-ahead mengacu kepada teknik pembuatannya, yaitu dengan teknik frame by frame, digambar satu per satu. Walt Disney yang mempunyai ratusan animator dari berbagai mancanegara menggunakan teknik ini sehingga animasi terlihat sangat halus dan detail. Bagi Anda yang mempunyai dana terbatas jangan coba-coba menggunakan teknik ini karena pengerjaannya akan lama dan butuh tenaga animator yang banyak. Ujung-ujungnya dana bisa habis sebelum film animasi selesai dikerjakan.

Pose to pose menggunakan teknik keyframe, seperti tween motion di flash. Ini cocok untuk mereka yang dananya terbatas dan butuh pengerjaan cepat. Tetapi ingat, karakter yang dibuat jangan terlalu detail dan rumit karena akan menyulitkan pengerjaan animasi. Sederhana saja sehingga karakter tersebut mudah digerakkan. Animasi jepang paling banyak menggunakan teknik ini seperti Sinchan dan The Powerpuff Girls.

Dari sisi resource dan pengerjaan, ada dua cara yang bisa dilakukan untuk membuat animasi. Yang pertama adalah Straight Ahead Action, yaitu membuat animasi dengan cara seorang animator menggambar satu per satu, frame by frame dari awal sampai selesai seorang diri. Teknik ini memiliki kelebihan: kualitas gambar yang konsisten karena dikerjakan oleh satu orang saja. Tetapi memiliki kekurangan yaitu waktu pengerjaan yang lama. Yang kedua adalah Pose to Pose, yaitu pembuatan animasi oleh seorang animator dengan cara menggambar hanya pada keyframe-keyframe tertentu saja, selanjutnya in-between atau interval antar keyframe digambar/dilanjutkan oleh asisten/animator lain. Cara kedua ini memiliki waktu pengerjaan lebih cepat karena melibatkan lebih banyak sumber daya sehingga lebih cocok diterapkan pada industri animasi.

## **9. Anticipation ( Gerakan Pendahulu )**

Anticipation boleh juga dianggap sebagai persiapan/awalan gerak atau ancang-ancang. Seseorang yang bangkit dari duduk harus membungkukkan badannya terlebih dahulu sebelum benar-benar berdiri. Pada gerakan melompat, seseorang yang tadinya berdiri harus ada gerakan ‘membungkuk’ terlebih dulu sebelum akhirnya melompat.

Gerakan ini bertujuan untuk menjelaskan gerakan utama. Misalkan gerakan utama adalah orang terpeleset dan jatuh ke kolam renang. Sebelum itu, ada elemen-elemen yang ditunjukkan sebelum itu seperti ada kulit pisang di lantai, kemudian ada gerakan air di kolam renang, orang berjalan dengan mimik ciek. Gerakan-gerakan antisipasi bertujuan agar penonton memahami apa yang akan terjadi berikutnya. Jadi tidak langsung membuat orang tersebut jatuh ke kolam renang tanpa penonton tahu apa penyebabnya. Ini yang kadang dilupakan oleh orang, jadi seakan-akan penonton mengerti jalan pikiran sang animator. Oleh karena itu, film animasi yang dibuat banyak keganjilan dan keanehan karena kurang pahamiannya animator dengan prinsip ini.

## **10. Staging (Bidang Gambar)**

Staging dalam animasi meliputi bagaimana ‘lingkungan’ dibuat untuk mendukung suasana atau ‘mood’ yang ingin dicapai dalam sebagian atau keseluruhan scene. Biasanya berkaitan dengan posisi kamera pengambilan gambar. Posisi kamera bawah membuat karakter terlihat besar dan menakutkan, kamera atas membuat karakter tampak kecil dan bingung sedangkan posisi kamera samping membuat karakter tampak lebih dinamis dan menarik. Staging adalah sudut pengambilan gambar seperti memperbesar muka tokoh untuk memperlihatkan kesedihannya, mengambil dari jarak jauh untuk memperlihatkan kemewahan suatu rumah, mengambil dari atas untuk memberi kesan ada seseorang yang mengintip, dan sebagainya.

## **11. Personality (Penjiwaan Karakter)**

Membuat sedetail mungkin kepribadian seorang tokoh misalkan tanggal lahir, hobi, sifat baik dan jahat. Penjiwaan karakter akan membuat penonton mengenali karakter tersebut.



## 12. Appeal (Daya Tarik Karakter)

Appeal berkaitan dengan keseluruhan look atau gaya visual dalam animasi. Kita bisa dengan mudah mengidentifikasi gaya animasi buatan Jepang dengan hanya melihatnya sekilas. Kita juga bisa melihat style animasi buatan Disney atau Dreamworks cukup dengan melihatnya beberapa saat. Hal ini karena mereka memiliki appeal atau gaya tersendiri dalam pembuatan karakter animasi.

Daya tarik karakter tersebut harus bisa mempengaruhi emosi penonton. Misalkan tampannya yang bodoh sehingga membuat penonton tertawa atau tampang yang tak berdosa sehingga membuat penonton merasa kasihan.

### 3.5 Animasi Pada Android

#### 1. Animasi Tween

Tween Animation merupakan animasi yang diterapkan pada aplikasi berbasiskan widget, yang menerima input form biasa, lalu menggerakkannya sehingga menghasilkan (suatu) animasi. Perbedaan antara Tween dengan Frame Animations adalah pada Tween, Anda tidak perlu memerlukan banyak gambar yang terdefinisi sebelumnya, karena animasi ini akan menggerakkan satu gambar dengan pola-pola tetap yang dimiliki API Android

Sebuah animasi tween dapat melakukan serangkaian transformasi sederhana (posisi, ukuran, rotasi, dan transparansi) pada isi dari objek view. Jadi, jika dimiliki objek textview, objek tersebut dapat dipindahkan, diputar, maupun mengecilkan teks. Jika memiliki gambar latar belakang, gambar latar belakang akan berubah bersama dengan teks.

Animasi Tween dapat dikategorikan sebagai berikut :

- Scale animation  
animasi perubahan skala (besar-kecil, tinggi-rendah)
- Rotate animation  
perubahan perputaran objek
- Translate animation  
perpindahan objek animasi

- Alpha animation  
perubahan transparansi

File animasi XML disimpan di res/anim file harus memiliki elemen root : <alpha>, <scale>, <rotate>, <translate>, atau <set> yang dapat menggabungkan beberapa elemen animasi. Secara default, semua instruksi animasi diterapkan secara bersamaan. Untuk membuat mereka terjadi secara berurutan, harus ditentukan atribut start off set.

Contoh code animasi pada directory res :

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator">
    <scale
        android:fromXScale="1"
        android:toXScale="1"
        android:fromYScale="0.1"
        android:toYScale="1.0"
        android:duration="500"
        android:pivotX="50%"
        android:pivotY="50%"
        android:startOffset="100" />
    </set>
```

## 2. Animasi Frame

Ini adalah animasi tradisional dalam arti bahwa hal itu dibuat dengan urutan gambar yang berbeda dimainkan dalam urutan, seperti rol film. Teknik ini memerlukan banyak gambar dengan perubahan-perubahan tertentu untuk menghasilkan suatu animasi. Sebagai contoh teknik animasi frame yang sering kita temui sehari-harinya adalah pada roll film(bioskop). Biasanya, bioskop-bioskop memutar film dengan memutar frame film yang sangat banyak yang digulung dalam roll film, mirip pada roll film kamera.

Kelas animation drawable adalah dasar untuk animasi frame. Lebih sederhana dilakukan dengan file XML yang berisi daftar frame yang membentuk animasi. Seperti

animasi tween di atas, file XML untuk jenis animasi termasuk ke dalam res/drawable. Dalam hal ini, petunjuk urutan dan durasi untuk setiap frame animasi.

### 3.6 Komponen yang dibutuhkan dalam merancang animasi pada android.

#### 1. Widget

Widget merupakan objek yang ada pada pemrograman Android. Objek widget ini berfungsi sebagai interface dan media untuk berinteraksi dengan pengguna. Berikut beberapa widget yang diperlukan dalam merancang suatu aplikasi animasi pada android.

- Button

Objek button merupakan objek yang sering digunakan dalam membuat aplikasi, apakah itu berbasis web atau desktop. Button biasanya digunakan untuk awal memulai sebuah activity dengan melakukan aksi tekan button, klik kanan button dan sebagainya.

- TextView

Objek textview fungsinya hampir sama dengan *label* (jika pada pemrograman Visual Basic). Seperti halnya label maka text view berfungsi sebagai penamaan saja pada objek dalam Android.

Atribut	Keterangan	Nilai
android:textsize	Menampilkan ukuran teks	Sembarang nilai
android:fontstyle	Mengatur font style	Bold, italic
android:typeface	Mengatur jenis huruf yang digunakan	Sans, serif, monospace dan lainnya
Android:textcolor	Mengatur warna text	Sembarang warna

- RadioButton

Objek radio button fungsinya hampir sama dengan check box, namun pada radio button hanya dapat memilih satu opsi saja.

- ImageButton

Widget yang sering digunakan dalam pembuatan aplikasi android yang sebenarnya hanya button biasa hanya saja bisa diletakkan gambar/*image* di atasnya agar lebih menarik.

- Drop-Down

Widget ini biasa juga disebut combo box. Menu yang akan ditampilkan akan disusun dalam bentuk-bentuk drop-down. Dalam penggunaanya widget drop-down membutuhkan satu widget text view dan satu spinner.

## 2. Layout

- Linear Layout

Layout yang digunakan untuk pengaturan widget dalam format horizontal atau vertical. Jenis layout ini cukup populer dan sering digunakan dalam pembuatan aplikasi android. Layout ini akan mengatur keseluruhan widget dalam posisi horizontal atau vertical sesuai dengan nilai *orientation* yang digunakan.

- Table Layout

Layout yang digunakan untuk pengaturan widget dalam format table/tabular. Jika anda pernah menggunakan bahasa HTML maka layout ini sama dengan tag `<table></table>`.

- Relative Layout

Layout yang digunakan untuk pengaturan widget yang jarak antar-widget atau jarak widget dengan container diatur secara relative

- Absolute Layout

Layout yang digunakan untuk pengaturan widget sesuai dengan titik tertentu pada layar, atau titik koordinat tertentu pada layar. Pada widget yang kita gunakan, tinggal menambahkan property *android:layout\_x* dan *android:layout\_y*.

- Frame Layout

Layout jenis ini akan menempatkan widget dalam satu tampilan tunggal. Frame layout ini akan ditempatkan pada kiri atas dari layout keseluruhan

## 3. Canvas

Sebuah aplikasi yang memerlukan pengontrolan animasi grafis maupun diinginkan penggambaran harus dilakukan melalui Canvas. Canvas akan bekerja sebagai antarmuka ke permukaan sebenarnya grafis akan digambar. Melalui Canvas penggambaran akan dilakukan ke bitmap dan diletakkan ke window.

Pemanggilan *method* `onDraw ()` akan menyediakan Canvas sehingga dapat menempatkan gambar di atasnya. Selain itu, Canvas dapat juga diperoleh dengan memanggil *method* `SurfaceHolder.lockCanvas()` ketika ingin menggunakan objek `SurfaceView`. Kelas Canvas telah memiliki *method* yang dapat digunakan seperti `drawText ()`, `drawCircle ()`, dan banyak lagi

- Paint

Paint berfungsi sebagai alat bantu untuk melakukan penggambaran. Paint mengontrol warna, transparansi, dan ukuran brush untuk objek yang dilukis pada Canvas. Selain itu dapat digunakan mengontrol font, ukuran, dan gaya saat menggambar teks.

- Path

Kelas Path memegang seperangkat perintah menggambar vector seperti grafis , empat persegi panjang, dan kurva.

- Bitmap

Bitmap dapat diibaratkan seperti kertas yang dapat digambari dan kelas yang menangani sebenarnya dari pixel gambar.

- Drawable

Drawable merupakan sesuatu yang ingin digambar misalnya persegi atau gambar. Walaupun tidak semua yang ingin digambar merupakan bagian dari subclass ini misalnya text. Drawable merupakan objek yang dapat membuat sendiri pada canvas. Drawable dapat mengambil berbagai bentuk :

- ✓ Bitmap File

Sebuah file grafis bitmap (png, Jpeg, Jpg atau GIF)

- ✓ Nine-Patch File

File PNG dengan daerah elastis untuk mengizinkan gambar Resize berdasarkan konten.

- ✓ State List

XML file yang referensi grafis bitmap yang berbeda untuk state list yang berbeda, misalnya : untuk menggunakan gambar yang berbeda ketika tombol ditekan.

- ✓ Color

Sebuah sumber yang didefinisikan dalam XML yang menentukan segi empat warna, dengan sudut opsional bulat.

✓ Shape

Sebuah file XML yang menentukan bentuk geometris, termasuk warna dan gradient

Ada tiga cara untuk mendefinisikan Drawable, yaitu menggunakan gambar yang disimpan pada folder resources proyek, menggunakan file xml yang mendefinisikan property drawable, atau menggunakan constructor kelas tersebut.

- OpenGL

OpenGL terlalu besar untuk dimuat pada perangkat mobile. Jadi Android mengimplementasikan subset dari OpenGL yang disebut OpenGL for Embedded Systems (OpenGL ES). Android menyediakan pelayanan untuk grafis 3D melalui OpenGL api. Untuk menggunakannya dengan meng-extend kelas **GLSurfaceView**. SurfaceView merupakan subclass dari view yang menawarkan permukaan gambar ke Thread sekunder aplikasi, sehingga aplikasi tidak perlu menunggu sampai sistem view siap untuk menggambar.

**GLSurfaceView** adalah kelas baru yang diperkenalkan di Android 1.5 dan sangat menyederhanakan penggunaan OpenGL pada Android. Hal ini menyambungkan OpenGL ES ke sistem **view** dan **activity life cycle**. Menangani pemilihan frame buffer format piksel yang tepat dan mengelola terpisah rendering Thread untuk mengaktifkan animasi halus. Dalam OpenGL ES di Android, gambar dipisahkan keluar menuju kelas **Render** yang bertanggung jawab untuk menginisialisasi dan menggambar seluruh layar.

**GLRenderer** mengimplementasikan antarmuka **GLSurfaceView**. **Renderer**, yang memiliki tiga metode. *Method onSurfaceCreated*, yang akan dipanggil ketika Surface OpenGL ( seperti Canvas di 2D biasa) dibuat atau diciptakan kembali. OpenGL memiliki banyak pilihan yang dapat diaktifkan atau di nonaktifkan dengan **glEnable ()** dan **glDisable ()**.

Yang paling umum digunakan adalah :

- `GL_BLEND` : Menghaluskan nilai warna yang masuk dengan nilai-nilai warna yang telah di buffer.
- `GL_CULL_FACE` : Mengabaikan polygon berdasarkan belokan-nya (searah jarum jam atau berlawanan) dalam jendela koordinat.
- `GL_DEPTH_TEST` : Melakukan perbandingan mendalam, dan update kedalam buffer. Apabila ukuran pixels lebih jauh hasilnya dari apa yang telah digambarkan maka akan diabaikan oleh compiler.
- `GL_LIGHTi` : Menyertakan nomor cahaya saat menentukan sebuah kecerahan dan warna objek
- `GL_LIGHTING` : Menghidupkan perhitungan pencahayaan dan material.
- `GL_LINE_SMOOTH` : Membuat garis antialiased.
- `GL_MULTISAMPLE` : Multisampling untuk antialiasing dan efek lainnya.
- `GL_POINT_SMOOTH` : Sebagai Draw poin antialiased.
- `GL_TEXTURE_2D` : Menggunakan tekstur dalam menggambar surface.

Semua pilihan di atas secara default berada dalam keadaan *disable* kecuali untuk `GL_DITHER` dan `GL_MULTISAMPLE`.

### 3.3 Pemrograman Android.Animation

Android menawarkan beberapa alat animasi berbeda. Animasi yang didasarkan pada kelas `Android.view.animation.Animation` : `RotateAnimation`, `TranslateAnimation`, `ScaleAnimation`, dan `RotateAnimation`. Animasi ini digunakan antar pasangan View. Tipe kedua merupakan subkelas dari `android.graphics.drawable.Animation`. `Drawable` bisa dimasukkan ke dalam latar belakang widget apapun untuk menyediakan berbagai macam efek.

- Deklarasi *Package*

Class-class ini menyediakan fungsionalitas untuk sistem animasi properti, yang memungkinkan untuk menghidupkan properti obyek dari jenis apapun. `Int`, `float`, nilai, warna, dan heksadesimal didukung secara default. User dapat menghidupkan jenis lain dengan mengatakan sistem bagaimana menghitung nilai untuk tipe tertentu dengan `TypeEvaluator` dan user dapat mengatur berbagai jenis interpolators (terkandung



dalam `android.view.animation` ), tentukan `keyframes` atau animasi kelompok untuk bermain secara berurutan atau bersamaan (dengan `AnimatorSet` ) untuk lebih mengontrol perilaku animasi yang dibuat.

- Deklarasi *Interface*

<a href="#"><u>Animator.AnimatorListener</u></a>	Pendengar animasi menerima pemberitahuan dari animasi.
<a href="#"><u>Animator.AnimatorPauseListener</u></a>	Pendengar jeda menerima pemberitahuan dari animasi saat animasi paused atau resumed.
<a href="#"><u>LayoutTransition.TransitionListener</u></a>	Antarmuka ini digunakan untuk mendengarkan awal dan akhir acara untuk transisi.
<a href="#"><u>TimeAnimator.TimeListener</u></a>	Pelaksanaan dari interface ini dapat mengatur user sebagai pembaruan pendengar untuk TimeAnimator contoh, untuk menerima callback pada setiap frame animasi untuk menerima total nya sejak animator mulai dan waktu delta sejak frame terakhir.
<a href="#"><u>TimeInterpolator</u></a>	Waktu interpolator mendefinisikan laju perubahan animasi.
<a href="#"><u>TypeEvaluator</u> &lt;T&gt;</a>	Antarmuka untuk digunakan dengan fungsi <code>setEvaluator</code> (TypeEvaluator).
<a href="#"><u>ValueAnimator.AnimatorUpdateListener</u></a>	Pelaksanaan dari interface ini dapat menambah user sebagai pembaruan pendengar ke ValueAnimator contoh untuk menerima callback pada setiap frame animasi, setelah nilai frame saat ini telah dihitung untuk ValueAnimator .

- Deklarasi *Classes*

<a href="#"><u>Animator</u></a>	Ini adalah super class untuk kelas yang menyediakan dukungan dasar untuk animasi yang dapat dimulai, berakhir, dan memiliki AnimatorListeners ditambahkan kepada user.
---------------------------------	--

<a href="#"><u>Animator Inflater</u></a>	Kelas ini digunakan untuk instantiate animator file XML menjadi obyek Animator.
<a href="#"><u>Animator Listener Adapter</u></a>	Kelas adaptor ini menyediakan implementasi kosong metode dari Animator.AnimatorListener .
<a href="#"><u>AnimatorSet</u></a>	Kelas ini memainkan set Animator objek dalam urutan tertentu.
<a href="#"><u>AnimatorSet. Builder</u></a>	Objek Builder adalah kelas utilitas untuk memfasilitasi menambahkan animasi ke AnimatorSet bersama dengan hubungan antara berbagai animasi.
<a href="#"><u>ArgbEvaluator</u></a>	Evaluator ini dapat digunakan untuk melakukan jenis interpolasi antara nilai integer yang mewakili warna ARGB.
<a href="#"><u>Bidirectional Type Converter</u></a> <T, V>	Kelas dasar abstrak yang digunakan jenis mengkonversi T untuk tipe V yang lain dan kembali lagi.
<a href="#"><u>FloatArray Evaluator</u></a>	Evaluator ini dapat digunakan untuk melakukan jenis interpolasi antara nilai-nilai float [ ].
<a href="#"><u>FloatEvaluator</u></a>	Evaluator ini dapat digunakan untuk melakukan jenis interpolasi antara nilai float.
<a href="#"><u>IntArray Evaluator</u></a>	Evaluator ini dapat digunakan untuk melakukan jenis interpolasi antara nilai-nilai int [ ] .
<a href="#"><u>IntEvaluator</u></a>	Evaluator ini dapat digunakan untuk melakukan jenis interpolasi antara nilai int.
<a href="#"><u>Keyframe</u></a>	Kelas ini memegang sepasang waktu/ nilai animasi.

<a href="#"><u>Layout Transition</u></a>	Kelas ini memungkinkan animasi otomatis pada perubahan tata letak dalam objek ViewGroup.
<a href="#"><u>Object Animator</u></a>	Ini subclass dari ValueAnimator menyediakan dukungan untuk menghidupkan properti pada objek sasaran.
<a href="#"><u>PointF Evaluator</u></a>	Evaluator ini dapat digunakan untuk melakukan jenis interpolasi antara nilai PointF.
<a href="#"><u>Property ValuesHolder</u></a>	Kelas ini menyimpan informasi tentang properti dan nilai-nilai yang properti yang harus mengambil selama animasi.
<a href="#"><u>RectEvaluator</u></a>	Evaluator ini dapat digunakan untuk melakukan jenis interpolasi antara nilai Rect.
<a href="#"><u>StateList Animator</u></a>	Memungkinkan user menentukan sejumlah animator yang akan berjalan terpasang tergantung pada View milik yang ditarik.
<a href="#"><u>TimeAnimator</u></a>	Kelas ini menyediakan mekanisme callback sederhana untuk pendengar yang disinkronkan dengan semua animator lain dalam sistem.
<a href="#"><u>TypeConverter&lt;T, V&gt;</u></a>	Kelas dasar abstrak yang digunakan jenis mengkonversi T untuk tipe V lain.
<a href="#"><u>Value Animator</u></a>	Kelas ini menyediakan mesin waktu sederhana untuk menjalankan animasi yang menghitung nilai animasi dan mengatur mereka pada objek sasaran.

### 3.4 Definisi Game

Menurut Romi Satria Wahono game merupakan aktifitas terstruktur atau semi terstruktur yang biasanya bertujuan untuk hiburan dan kadang dapat digunakan sebagai sarana pendidikan. Karakteristik game yang menyenangkan, memotivasi, membuat kecanduan dan kolaboratif membuat aktifitas ini digemari oleh banyak orang.

### 3.5 Game Platforms

Game secara umum banyak dibuat di beberapa platform sebagai berikut:

#### » **Arcade**

Arcade merupakan game dengan sistem coin operated, sampai saat ini platform jenis ini masih sering digunakan. Permulaan platform ini dipelopori oleh Atari. Yang termasuk dalam platform ini adalah :

1. Sega
2. Namco
3. Video Game

#### » **Personal Computer**

Game yang berjalan pada platform ini sangat mendominasi pasar, ada beberapa Operating System yang biasa dipakai untuk membuat game pada platform ini:

##### 1. Windows

Memanfaatkan kemampuan Windows untuk tampilan yang sangat baik dengan kemampuan DirectX yang disediakan Microsoft menjadikan para pengembang game umumnya bermain di platform ini. Tools dan software untuk membuat game sangat banyak tersedia di platform Windows. Baik dengan atau tanpa programming ada di platform ini, dan karena PC adalah peralatan yang banyak dimiliki oleh pengguna rumahan maka kebanyakan pembuat game mengembangkan game pada platform ini.

##### 2. Linux

Linux walaupun terkenal stabil, masih sedikit keraguan untuk mengembangkan game pada platform ini, karenanya masih sedikit game yang dibuat di Linux. Tapi kemudahan lain adalah tersedianya aplikasi yang gratis sehingga walaupun sedikit komunitasnya bisa membuat game yang tidak kalah dari platform Windows. Mungkin hanya kalah bersaing dari jumlah pengembang dan waktu saja, salah satu game engine yang bisa berjalan pada platform ini adalah Torque Engine 3D dengan pemrograman C++.

##### 3. Dos

Masa keemasan platform ini sudah selesai. Dulu hampir semua game komersil berjalan di platform ini. Namun kini sudah tidak dikembangkan lagi karena perpindahan Operating System yang beralih ke Windows. Namun masih banyak resources di internet yang masih menyediakan pembahasan dan programming dengan DOS.

#### 4. Mac

Mac adalah sistem operasi keluaran Apple yang stabil dan fiturnya sangat baik. Sebenarnya Windows menirunya. Tapi nasibnya tidak terlalu beruntung. Kelebihan dari platform ini adalah fitur grafik dan suaranya yang sangat baik. Jika kita mencari di internet ada beberapa tools dan software yang bisa kita gunakan untuk membuat game di platform ini, tapi terbatas baik jumlah maupun jenis game yang bisa dibuat.

#### » **Hand-held dan Mobile Device (Cell phone)**

Platform ini dipilih karena kepraktisannya. Walaupun tidak sehebat game di komputer, tetapi kemudahan dan jumlah peralatan yang nyaris digunakan untuk alat komunikasi akan membuat platform ini menjadi pilihan pengembang. Umumnya menggunakan java akan tetapi sekarang mulai banyak game mobile yang dikembangkan dengan menggunakan flashlite pada platform ini. Yang termasuk dalam platform hand held ini adalah:

1. Game Watch
2. Gameboy
3. Gameboy Advance
4. N-Gage
5. Gameboy DS
6. PSP

#### » **Console**

Mesin Console seperti Playstation dan Xbox menjadi saingan bagi platform Windows. Pada awalnya mesin console duluan menjual hiburan ini kepada pemakainya, komputer muncul belakangan. Namun kekuatan pasar awal mesin console dan kemudahan peralatannya menjadikan tetap menjadi pesaing platform PC walaupun

sekarang platform PC (Windows) sudah menjadi nomor satu. Pemrograman di mesin console hampir sama dengan platform PC namun kita kurang mendapatkan kemudahan karena umumnya sulit mengembangkan game di platform ini secara individu karena tools dan lisensi ada pada studio pengembang dan harganya sangat mahal demi menjaga mutu dan hak cipta mereka.

### **3.6 Game Genres**

Jenis-jenis game lebih dikenal dengan istilah genre. Samuel Henry mendefinisikan genre sebagai format atau gaya dari sebuah game. Format sebuah game bisa murni sebuah genre atau campuran (hybrid) dari beberapa genre lain dengan maksud membuat unsur permainan lebih bervariasi dan menantang berbagai jenis game beredar dipasaran, beberapa diantaranya adalah:

#### **» Maze Game**

Jenis game ini adalah jenis game yang paling awal muncul. Contoh yang paling populer di Indonesia adalah game pacman dan digger. Pada maze game ini pemain hanya mengitari maze (lorong-lorong yang berhubungan) dan memakan beberapa item untuk menambah tenaga dan kekebalan misalnya. Pemain juga memiliki musuh yang selalu mengejar. Ketika pemain mendapat kekebalan, pemain bisa berbalik mengejar musuh. Permainan ini sederhana tetapi mengasyikkan. Mode permainan ini yang menjadi dasar bagi permainan 3D sekarang. Jika dulu bentuknya 2D maka sekarang menjadi bentuk 3D.

#### **» Board Game**

Jenis game ini sama dengan game board tradisional, seperti monopoly (selain platform PC juga ada pada platform konsol). Sampai saat ini tidak ada variasi yang memunculkan gameplay atau perubahan desain dari versi tradisional ke versi elektronik. Versi elektronik benar-benar hanya memindahkan versi tradisional ke layar komputer. Variasi yang ada hanyalah memindahkan versi 2D menjadi versi 3D (misalnya seperti dalam game catur). Terkadang disisipkan variasi film intro atau animasi lainnya. Umumnya game ini lebih menekankan kepada kemampuan komputer untuk menjadi lawan tanding bagi pemain. Ini melibatkan kemampuan AI (Artificial Intelligence) yang andala untuk bisa menjadikan game ini menantang pemain dengan baik.

#### » **Puzzle Game**

Game ini memberikan tantangan kepada pemainnya dengan cara menjatuhkan sesuatu dari sisi sebelah atas ke bawah. Pemain harus menyusun sedemikian rupa dan tidak ada yang tersisa ketika susunan di atasnya sudah akan dibuat. Susunan ini dilakukan secepat dan sebaik mungkin. Semakin lama akan semakin cepat dan semakin banyak obyek yang jatuh. Contoh yang populer dari jenis ini adalah tetris.

#### » **Fighting Game**

Sesuai dengan namanya game ini mengetengahkan pertarungan. Pada awalnya bersifat 2D namun pada akhirnya banyak mengadopsi sistem 3D disertai animasi. Game ini memberikan kesempatan kepada pemain untuk mengkombinasikan berbagai gerakan dalam pertarungan. Ada yang mengadopsi permainan bela diri, atau ada juga gerakan yang liar. Terkadang musuh bukan manusia tetapi makhluk yang tidak masuk akal sama sekali. Contoh yang populer di Indonesia adalah Street Fighter 2, Samuraidown, Virtua Fighter, Kungfu dan sebagainya.

#### » **Racing Game**

Game balapan, game ini memberikan permainan lomba kecepatan dari kendaraan yang dimainkan oleh pemain. Bisa di dalam arena balap atau diluar arena balap. Beberapa contoh game yang terkenal seperti Need For Speed Underground dan Toca Race Driver.

#### » **Action Game**

Memasuki gua bawah tanah, melompati bebatuan di antara lahar, bergelayutan dari pohon satu ke pohon lain, bergulat dengan ular sambil mencari kunci untuk membuka pintu kuil legendaris atau sekedar mencari telepon umum untuk mendapatkan misi berikutnya, itulah beberapa dari banyak hal yang karakter pemain harus lakukan dan lalui dalam video game jenis ini. Menurut penulis, game jenis ini sudah berkembang jauh hingga menjadi genre campuran action beatem up juga, dan sekarang, di tahun 2000-an, jenis ini cenderung untuk memiliki visual 3d dan sudut pandang orang ketiga. Tomb rider, grand theft auto dan prince of persia termasuk didalamnya.

### » **Strategi Game**

Kebalikan dari game jenis action yang berjalan cepat dan perlu refleks secepat kilat, video game jenis strategi, layaknya bermain catur, justru lebih memerlukan keahlian berpikir dan memutuskan setiap gerakan secara hati-hati dan terencana. Game jenis strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan, sampai kendaraan, bahkan hingga pembangunan berbagai bangunan, pabrik dan pusat pelatihan tempur, tergantung dari tema ceritanya. Pemain game strategi melihat dari sudut pandang lebih meluas dan lebih ke depan dengan waktu permainan yang biasanya lebih lama dan santai dibandingkan game action. Unsur-unsur permainannya biasanya berkisar sekitar prioritas pembangunan, peletakan pasukan, mencari dan memanfaatkan sumber daya (uang, besi, kayu, minyak, dll), hingga ke pembelian dan peng-upgrade-an pasukan atau teknologi.

### » **Game Simulasi kendaraan**

Game jenis ini memberikan pengalaman atau interaktifitas sedekat mungkin dengan kendaraan yang aslinya, meskipun terkadang kendaraan tersebut masih eksperimen atau bahkan fiktif, tapi ada penekanan khusus pada detil dan pengalaman realistik menggunakan kendaraan tersebut.

## **3.7 Konsep pengembangan game**

Pengembangan game biasanya dilakukan oleh seorang developer, bisa 1 orang atau 1 perusahaan besar. Biasanya, game komersial berskala besar dibuat oleh tim pengembang dalam sebuah perusahaan yang mengkhususkan pada game komputer atau konsol. Video game modern umumnya menghabiskan biaya mulai dari USD \$1.000.000 sampai lebih dari \$20.000.000 untuk dikembangkan. Pengembangan biasanya didanai oleh sebuah penerbit. Sebuah game modern bisa memakan waktu 1-3 tahun untuk dikembangkan, meskipun ada pengecualian lainnya. Proses pengembangan game bervariasi tergantung pada perusahaan dan proyek. Namun perkembangan game komersial biasanya meliputi tahapan sebagai berikut.

### **1. Pre-Production**



Fase awal perkembangan game sering ditandai oleh rendahnya kualitas grafis. Hal ini benar terutama dari berbagai permainan prototipe. Produsen game bekerja selama pra-produksi yang umumnya terkait dengan perencanaan jadwal, anggaran & perkiraan tugas dalam tim. Melakukan produser ini bertujuan untuk membuat rencana produksi yang padat sehingga produksi dapat dimulai bila diperlukan tanpa penundaan.

## 2. Production

Produksi mainstream biasanya didefinisikan sebagai periode waktu ketika proyek sepenuhnya dikerjakan. Programmer menulis banyak kode baru, artis game mengembangkan aset permainan seperti sprite, unsur permainan model 3D. Sound insinyur mengembangkan efek suara dan komposer musik untuk mengembangkan game. Desainer menciptakan tingkat lanjutan dan kenyamanan mata saat memainkan game tersebut dan penulis menulis dialog untuk cutscenes dan NPC.

Sementara itu, desainer game menerapkan dan memodifikasi desain game untuk mencerminkan visi sebuah game. Fitur dan level seringkali dihapus atau ditambahkan. Gaya seni yang ada bisa dikembangkan dan backstory mungkin dirubah. Platform baru mungkin saja menargetkan game dengan demografis baru. Semua perubahan ini harus didokumentasikan dan tersebar ke seluruh tim. Sebagian besar perubahan terjadi sebagai pembaruan dokumen desain.

Dari sudut pandang waktu, level pertama dari game bisa memakan waktu yang lama untuk dikembangkan. Sebagai desainer dan artis menggunakan alat-alat untuk membangun level, mereka meminta fitur dan perubahan yang lebih cepat dan memungkinkan untuk pengembangan dengan kualitas yang lebih tinggi. Memperkenalkan fitur baru dapat menyebabkan level lama menjadi usang, sehingga level awal bisa dikembangkan berkali-kali. Karena lingkungan dinamis pada pengembangan game, desain level awal dapat juga berubah seiring berjalannya waktu. Hal ini tidak jarang menghabiskan lebih dari dua belas bulan pada satu level game yang dikembangkan selama 3 tahun. Level selanjutnya dapat dikembangkan jauh lebih cepat sebagai set fitur yang lebih lengkap dan visi permainan yang lebih jelas dan lebih stabil.



### 3. Testing (pengujian)

Staf pengujian paling banyak diandalkan pada akhir proyek karena mereka tidak hanya perlu untuk menguji fitur baru yang ditambahkan, level dan perbaikan bug, tapi mereka juga perlu melakukan pengujian regresi untuk memastikan bahwa fitur-fitur yang telah ada ditempatnya selama berbulan-bulan bisa beroperasi dengan benar. Hal ini juga sering ketika fitur dan level sedang selesai pada level tertinggi, sehingga ada lebih banyak materi baru yang akan diuji daripada waktu lainnya dalam proyek. Pengujian regresi merupakan salah satu tugas paling penting yang dibutuhkan untuk pengembangan software yang efektif.

### 4. Penyelesaian

Setelah game dibuat, beberapa developer akan memberikan waktu kompensasi untuk anggota tim (mungkin sampai satu atau dua minggu) untuk mengkompensasi lembur dalam menyelesaikan permainan, meskipun kompensasi ini bukanlah suatu standar.

### 5. Pemeliharaan

Konsol game dulu dianggap 100% lengkap sehingga tidak mungkin adanya masalah atau bug. Namun, dengan pengenalan konsol yang bisa online seperti Xbox 360, PlayStation 3 dan Wii sebagian besar game menerima patch dan perbaikan setelah karena adanya bug dan masalah-masalah kecil, seperti game komputer.

Sementara konsol game dapat dikembangkan untuk kumpulan komponen yang berhingga, game komputer dapat memiliki konflik dengan berbagai konfigurasi hardware pengguna. Developer mencoba untuk menjelaskan konfigurasi yang paling umum, tetapi tidak dapat mengantisipasi semua sistem permainan yang mungkin mereka coba. Hal ini merupakan penerapan umum untuk pengembang game komputer untuk merilis patch. Patch ini digunakan untuk dikirimkan kepada pengguna melalui floppy disk, tetapi sekarang umumnya tersedia untuk didownload melalui situs web pengembang.

### 6. Durasi

Game modern bisa memakan waktu 1-3 tahun untuk menyelesaikannya. Lamanya suatu pengembangan bergantung pada sejumlah faktor, seperti genre, skala, platform pengembangan dan jumlah aset.

Sebagai contoh, sebuah game puzzle yang sederhana dengan menggunakan grafik 2D akan jauh lebih sedikit memakan waktu untuk dikembangkan daripada game role-playing komputer 3D. Pertimbangan lainnya adalah penggunaan mesin game middleware. Mengembangkan mesin 3D dari bawah ke atas membutuhkan waktu lebih banyak daripada menggunakan paket middleware yang ada Cots (commercial, off-the-shelf) seperti Gamebryo atau RenderWare. Sebagai contoh, Gas Powered Games mengembangkan sebuah mesin 3D kustom untuk permainan mereka Dungeon Siege. Pengembangannya memakan waktu 3 tahun. Firaxis menggunakan mesin game Gamebryo untuk permainan mereka Sid Meier's Pirates! yang dikembangkan selama 2 tahun kurang.

### **3.8 Perbedaan Game iOS dengan Game Android**

Faktor aplikasi merupakan hal yang paling menonjol ketika membandingkan iOS dengan Android. Saat ini iOS telah memiliki aplikasi termasuk didalamnya terdapat game dalam jumlah besar pada iTunes mereka yang merupakan sumber penghasilan dari si iOS. walaupun awalnya si Android memiliki lebih sedikit jumlah aplikasi maupun game yang ditawarkan kepada konsumen dibandingkan iOS, tapi kini pengguna Android tidak perlu khawatir karna saat ini banyak developer yang bersaing dalam pembuatan aplikasi dan game Android alhasil kini game Android lebih banyak jumlahnya dibandingkan dengan game iOS, selain itu terdapat kata yang amat disukai banyak konsumen yaitu kata "freedom".

Dengan Android kita dapat melakukan instalasi game Android dari sebuah halaman web (tentunya dengan melakukan download) ataupun bisa selain share game buatan sendiri untuk langsung dicoba di smartphone Android masing-masing. Tidak bisa melakukan hal yang sama seperti Android jika kita bergelut dengan si iOS. Di iOS instalasi game atau aplikasi lain tidak dapat dilakukan jikalau tidak dari iStore, para developer pun tidak mendapat akses untuk melakukan modifikasi fungsi iOS itu sendiri.

#### ***Game Android***

1. Dibuat dengan Foundation berbasis Java.
2. Bersifat Ekonomis.
3. Perangkat dapat dibeli dengan harga yang relative murah.

4. Terdapat market atau forum yang sifatnya gratis.

### ***Game iOS***

1. Dibuat dengan Foundation berbasis Objective-C
2. Bersifat Komersial.
3. Perangkat yang dipasarkan diperuntukan untuk kalangan menengah ke atas.
4. Tidak memiliki market gratis melainkan iStore yang statusnya berbayar.

# 4

## ANDROID NETWORK PROGRAMMING



Wi-Fi (/icon wa fa /, juga ditulis Wifi atau WiFi) adalah sebuah teknologi terkenal yang memanfaatkan peralatan elektronik untuk bertukar data secara nirkabel (menggunakan gelombang radio) melalui sebuah jaringan komputer, termasuk koneksi Internet berkecepatan tinggi. Wi-Fi Alliance mendefinisikan Wi-Fi sebagai "produk jaringan wilayah lokal nirkabel (WLAN) apapun yang didasarkan pada standar Institute of Electrical and Electronics Engineers (IEEE) 802.11". Meski begitu, karena kebanyakan WLAN zaman sekarang didasarkan pada standar tersebut, istilah "Wi-Fi" dipakai dalam bahasa Inggris umum sebagai sinonim "WLAN". Sebuah alat yang dapat memakai Wi-Fi (seperti komputer pribadi, konsol permainan video, telepon pintar, tablet, atau pemutar audio digital) dapat terhubung dengan sumber jaringan seperti Internet melalui sebuah titik akses jaringan nirkabel. Titik akses (atau hotspot) seperti itu mempunyai jangkauan sekitar 20 meter (65 kaki) di dalam ruangan dan lebih luas lagi di luar ruangan. Cakupan hotspot dapat mencakup wilayah seluas kamar dengan dinding yang memblokir gelombang radio atau beberapa mil persegi — ini bisa dilakukan dengan memakai beberapa titik akses yang saling tumpang tindih.

Sejarah WI-FI Sejarah teknologi 802.11 berawal pada putusan Komisi Komunikasi Federal AS tahun 1985 yang merilis pita GSM untuk pemakaian tanpa lisensi. Pada tahun 1991, NCR Corporation bersama AT&T menemukan pendahulu 802.11 yang ditujukan untuk sistem kasir. Produkproduk nirkabel pertama berada di bawah nama WaveLAN. Vic Hayes dijuluki "Bapak Wi-Fi". Ia terlibat dalam perancangan standar pertama IEEE. Sejumlah besar paten oleh banyak perusahaan memakai standar 802.11.

Pada tahun 1992 dan 1996, organisasi Australia CSIRO mendapatkan paten untuk sebuah metode yang kelak dipakai di Wi-Fi untuk menghapus gangguan sinyal. Pada bulan April 2009, 14 perusahaan teknologi setuju membayar \$250 juta kepada CSIRO karena melanggar paten-paten mereka.= Ini mendorong Wi-Fi disebut-sebut sebagai temuan Australia, meski hal ini telah menjadi topik sejumlah kontroversi. CSIRO memenangkan gugatan senilai \$220 juta atas

pelanggaran paten Wi-Fi tahun 2012 yang meminta firma-firma global di Amerika Serikat membayar hak lisensi kepada CSIRO senilai \$1 miliar. Tahun 1999, Wi-Fi Alliance dibentuk sebagai sebuah asosiasi dagang untuk memegang merek dagang Wi-Fi yang digunakan oleh banyak produk.

Nama

Istilah Wi-Fi, pertama dipakai secara komersial pada bulan Agustus 1999, dicetuskan oleh sebuah firma konsultasi merek bernama Interbrand Corporation. Wi-Fi Alliance mempekerjakan Interbrand untuk menentukan nama yang "lebih mudah diucapkan daripada 'IEEE 802.11b Direct Sequence'". Belanger juga mengatakan bahwa Interbrand menciptakan Wi-Fi sebagai plesetan dari Hi-Fi (high fidelity); mereka juga merancang logo Wi-Fi. Wi-Fi Alliance membuat slogan iklan asal-asalan "The Standard for Wireless Fidelity" dan sempat menggunakannya sesaat setelah merek Wi-Fi diciptakan. Karena slogan tersebut, orang-orang salah mengira bahwa Wi-Fi merupakan singkatan dari "Wireless Fidelity" meski kenyataannya bukan. Logo yin-yang Wi-Fi menandakan sertifikasi interoperabilitas suatu produk.

Sertifikasi Wi-Fi IEEE tidak menguji peralatan untuk memenuhi standar mereka. Badan nirlaba Wi-Fi Alliance didirikan tahun 1999 untuk mengisi celah ini — untuk menetapkan dan mendorong standar interoperabilitas dan kompatibilitas mundur, serta mempromosikan teknologi jaringan wilayah lokal nirkabel. Pada 2010, Wi-Fi Alliance terdiri dari lebih dari 375 perusahaan di seluruh dunia. Wi-Fi Alliance mendorong pemakaian merek Wi-Fi kepada teknologi yang didasarkan pada standar IEEE 802.11 dari Institute of Electrical and Electronics Engineers. Ini meliputi koneksi jaringan wilayah lokal nirkabel (WLAN), konektivitas alat-ke-alat (seperti Wi-Fi Peer to Peer atau Wi-Fi Direct), jaringan wilayah pribadi (PAN), jaringan wilayah lokal (LAN), dan bahkan sejumlah koneksi jaringan wilayah luas (WAN) terbatas.

Perusahaan manufaktur dengan keanggotaan Wi-Fi Alliance, yang produknya berhasil melewati proses sertifikasi, berhak menandai produk tersebut dengan logo Wi-Fi. Secara spesifik, proses sertifikasi memerlukan pemenuhan standar radio IEEE 802.11, standar keamanan WPA dan WPA2, dan standar autentikasi EAP. Sertifikasi opsionalnya meliputi pengujian standar draf IEEE 802.11, interaksi dengan teknologi telepon seluler pada peralatan konvergen, dan fitur-fitur keamanan, multimedia, dan penghematan tenaga.[23] Tidak semua peralatan Wi-Fi dikirim untuk mendapatkan sertifikasi. Kurangnya sertifikasi Wi-Fi tidak berarti bahwa sebuah alat tidak kompatibel dengan alat Wi-Fi lainnya. Jika alat tersebut memenuhi



syarat atau setengah kompatibel, Wi-Fi Alliance tidak perlu berkomentar terhadap penyebutannya sebagai sebuah alat Wi-Fi,[butuh rujukan] meskipun secara teknis hanya alat yang bersertifikasi yang disetujui. Istilah seperti Super Wi-Fi, yang dicetuskan oleh Komisi Komunikasi Federal (FCC) AS untuk mendeskripsikan rencana jaringan pita TV UHF di Amerika Serikat, dapat disetujui atau tidak.



Gambar 4.1 Logo Wifi

Penggunaan Agar terhubung dengan LAN Wi-Fi, sebuah computer atau device android perlu dilengkapi dengan pengontrol antarmuka jaringan nirkabel. Gabungan komputer dan pengontrol antarmuka disebut stasiun. Semua stasiun berbagi satu saluran komunikasi frekuensi radio. Transmisi di saluran ini diterima oleh semua stasiun yang berada dalam jangkauan. Perangkat keras tidak memberitahu pengguna bahwa transmisi berhasil diterima dan ini disebut mekanisme pengiriman terbaik. Sebuah gelombang pengangkut dipakai untuk mengirim data dalam bentuk paket, disebut "bingkai Ethernet". Setiap stasiun terus terhubung dengan saluran komunikasi frekuensi radio untuk mengambil transmisi yang tersedia.

Spesifikasi Wi-Fi dirancang berdasarkan spesifikasi IEEE 802.11. Sekarang ini ada empat variasi dari 802.11, yaitu:

1. 802.11a
2. 802.11b
3. 802.11g
4. 802.11n

Spesifikasi b merupakan produk pertama Wi-Fi. Variasi g dan n merupakan salah satu produk yang memiliki penjualan terbanyak pada 2005.

Tabel Spesifikasi Wifi

Spesifikasi Wi-Fi			
Spesifikasi	Kecepatan	Frekuensi Band	Cocok dengan
<u>802.11b</u>	11 Mb/s	~2.4 GHz	b
<u>802.11a</u>	54 Mb/s	~5 GHz	a
<u>802.11g</u>	54 Mb/s	~2.4 GHz	b, g
<u>802.11n</u>	100 Mb/s	~2.4 GHz	b, g, n

Di banyak bagian dunia, frekuensi yang digunakan oleh Wi-Fi, pengguna tidak diperlukan untuk mendapatkan izin dari pengatur lokal (misal, Komisi Komunikasi Federal di A.S.). 802.11a menggunakan frekuensi yang lebih tinggi dan oleh sebab itu daya jangkauannya lebih sempit, lainnya sama. Versi Wi-Fi yang paling luas dalam pasaran AS sekarang ini (berdasarkan dalam IEEE 802.11b/g) beroperasi pada 2.400 GHz sampai 2.483,50 GHz. Dengan begitu mengijinkan operasi dalam 11 channel (masing-masing 5 MHz), berpusat di frekuensi berikut:

1. Channel 1 - 2,412 GHz;
2. Channel 2 - 2,417 GHz;
3. Channel 3 - 2,422 GHz;
4. Channel 4 - 2,427 GHz;
5. Channel 5 - 2,432 GHz;
6. Channel 6 - 2,437 GHz;
7. Channel 7 - 2,442 GHz;
8. Channel 8 - 2,447 GHz;
9. Channel 9 - 2,452 GHz;
10. Channel 10 - 2,457 GHz;
11. Channel 11 - 2,462 GHz

Secara teknis operasional, Wi-Fi merupakan salah satu varian teknologi komunikasi dan informasi yang bekerja pada jaringan dan perangkat WLAN (wireless local area network). Dengan kata lain, Wi-Fi adalah sertifikasi merek dagang yang diberikan pabrikan kepada perangkat telekomunikasi (internet) yang bekerja di jaringan WLAN dan sudah memenuhi kualitas kapasitas interoperasi yang dipersyaratkan. Teknologi internet berbasis Wi-Fi dibuat dan dikembangkan sekelompok insinyur Amerika Serikat yang bekerja pada Institute of Electrical and Electronics Engineers (IEEE) berdasarkan standar teknis perangkat bernomor 802.11b, 802.11a dan 802.16. Perangkat Wi-Fi sebenarnya tidak hanya mampu bekerja di jaringan WLAN, tetapi juga di jaringan Wireless Metropolitan Area Network (WMAN). Karena perangkat dengan standar teknis 802.11b diperuntukkan bagi perangkat WLAN yang digunakan di frekuensi 2,4 GHz atau yang lazim disebut frekuensi ISM (Industrial, Scientific dan Medical). Sedang untuk perangkat yang berstandar teknis 802.11a dan 802.16 diperuntukkan bagi perangkat WMAN atau juga disebut Wi-Max, yang bekerja di sekitar pita frekuensi 5 GHz. Wi-fi Hardware Hardware Wi-Fi yang ada di pasaran saat ini ada berupa :

1. PCI
2. USB
3. PCMCIA
4. Compact Flash

Port (Jaringan Komputer) Dalam protokol jaringan TCP/IP, sebuah port adalah mekanisme yang mengizinkan sebuah komputer untuk mendukung beberapa sesi koneksi dengan komputer lainnya dan program di dalam jaringan. Port dapat mengidentifikasi aplikasi dan layanan yang menggunakan koneksi di dalam jaringan TCP/IP. Sehingga, port juga mengidentifikasi sebuah proses tertentu di mana sebuah server dapat memberikan sebuah layanan kepada klien atau bagaimana sebuah klien dapat mengakses sebuah layanan yang ada dalam server. Port dapat dikenali dengan angka 16-bit (dua byte) yang disebut dengan Port Number dan diklasifikasikan dengan jenis protokol transport apa yang digunakan, ke dalam Port TCP dan Port UDP. Karena memiliki angka 16-bit, maka total maksimum jumlah port untuk setiap protokol transport yang digunakan adalah 65536 buah. Dilihat dari penomorannya, port UDP dan TCP dibagi menjadi tiga jenis, yakni sebagai berikut:

1. Well-known Port: yang pada awalnya berkisar antara 0 hingga 255 tapi kemudian diperlebar untuk mendukung antara 0 hingga 1023. Port number yang termasuk ke dalam well-known port, selalu merepresentasikan layanan jaringan yang sama, dan ditetapkan oleh Internet Assigned Number Authority (IANA). Beberapa di antara port-port yang berada di dalam range Well-known port masih belum ditetapkan dan diresevasikan untuk digunakan oleh layanan yang bakal ada pada masa depan. Well-known port didefinisikan dalam RFC 1060.
2. Registered Port: Port-port yang digunakan oleh vendor-vendor komputer atau jaringan yang berbeda untuk mendukung aplikasi dan sistem operasi yang mereka buat. Registered port juga diketahui dan didaftarkan oleh IANA tapi tidak dialokasikan secara permanen, sehingga vendor lainnya dapat menggunakan port number yang sama. Range registered port berkisar dari 1024 hingga 49151 dan beberapa port di antaranya adalah Dynamically Assigned Port.
3. Dynamically Assigned Port: merupakan port-port yang ditetapkan oleh sistem operasi atau aplikasi yang digunakan untuk melayani request dari pengguna sesuai dengan kebutuhan. Dynamically Assigned Port berkisar dari 1024 hingga 65536 dan dapat digunakan atau dilepaskan sesuai kebutuhan.

**5**

## **ANDROID SQLite DATABASE**



### 5.1. Pengertian SQLite

SQLiteDatabase merupakan interface yang ada pada system operasi Android yang digunakan untuk membuat relational database. SQLite menyokong implementasi dari SQL yang kaya untuk apapun yang dibutuhkan oleh aplikasi mobile. Setiap aplikasi dapat memiliki databasenya sendiri dengan pengaturan lengkap.

Dengan penggunaan SQLite, dapat dibuat sebuah database untuk aplikasi yang digunakan untuk menyimpan dan mengatur data aplikasi terstruktur. Secara default, semua database bersifat private yang hanya diakses oleh aplikasi yang membuatnya. Desain database dengan baik termasuk normalisasi cukup penting untuk mengurangi redundancy.

Didalam database terdapat 3 kategori pengelolaannya yaitu :

#### 1. Data Definition Language (DDL)

Data Definition Language (DDL) merupakan sub bahasa SQL yang digunakan untuk membangun kerangka database. Ada tiga perintah yang termasuk dalam DDL, yaitu:

- **CREATE** : Perintah ini digunakan untuk membuat, termasuk di antaranya membuat database baru, tabel baru, view baru, dan kolom.
- **ALTER** : Perintah ini digunakan untuk mengubah struktur tabel yang telah dibuat. Pekerjaannya mencakup mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom, maupun memberikan atribut pada kolom.
- **DROP** : Perintah ini digunakan untuk menghapus database dan tabel.

## 2. Data Manipulation Language (DML)

Data Manipulation Language (DML) merupakan sub bahasa SQL yang digunakan untuk memanipulasi data dalam database yang telah terbuat. Perintah yang digunakan, di antaranya:

- **INSERT** : Perintah ini digunakan untuk menyisipkan atau memasukkan data baru ke dalam tabel. Penggunaannya setelah database dan tabel selesai dibuat.
- **SELECT** : Perintah ini digunakan untuk mengambil data atau menampilkan data dari satu tabel atau beberapa tabel dalam relasi. Data yang diambil dapat kita tampilkan dalam layar prompt MySQL secara langsung maupun ditampilkan pada tampilan aplikasi.
- **UPDATE** : Perintah ini digunakan untuk memperbarui data lama menjadi data terkini. Jika Anda memiliki data yang salah atau kurang up to date dengan kondisi sekarang, maka dapat diubah isi datanya menggunakan perintah UPDATE.
- **DELETE** : Perintah ini digunakan untuk menghapus data dari tabel. Biasanya data yang dihapus merupakan data yang sudah tidak diperlukan lagi. Pada saat menghapus data, perintah yang telah dijalankan tidak dapat digagalkan, sehingga data yang telah hilang tidak dapat dikembalikan lagi.

## 3. Data Control Language (DCL)

Data Control Language (DCL) merupakan sub bahasa SQL yang digunakan untuk melakukan pengontrolan data dan server databasenya. Perintah DCL, di antaranya:

- **GRAND** : Perintah ini digunakan untuk memberikan hak/ijin akses oleh administrator (pemilik utama) server kepada user (pengguna biasa). Hak akses tersebut berupa hak membuat (CREATE), mengambil (SELECT), menghapus (DELETE), mengubah (UPDATE), dan hak khusus berkenaan dengan sistem databasenya.
- **REVOKE** : Perintah ini memiliki kegunaan terbalik dengan GRAND, yaitu untuk menghilangkan atau mencabut hak akses yang telah diberikan kepada user oleh administrator.

#### ❖ Cursor dan Content Value

Objek Cursor merupakan nilai kembalian apabila dilakukan *Query* di Android. Cursor merupakan penunjuk dari serangkaian hasil data.cursor melayani pengaturan posisi pada kumpulan hasil dari *query* database. Beberapa method yang ada pada kelas `cursor`.

Query	Fungsi
<b>moveToFirst ()</b>	Memindahkan cursor ke baris pertama dari hasil <i>Query</i>
<b>moveToNext ()</b>	Memindahkan cursor ke baris selanjutnya
<b>moveToPrevious ()</b>	Memindahkan cursor ke baris sebelumnya
<b>getCount ()</b>	Mengembalikan jumlah baris pada kumpulan hasil
<b>getColumnName ()</b>	Mengembalikan nama dari indeks kolom yang ditentukan
<b>moveToPosition ()</b>	Memindahkan cursor ke baris yang ditentukan
<b>getPosition ()</b>	Mengembalikan posisi cursor saat ini

ContentValues digunakan untuk menambahkan baris baru ke tabel.

#### ❖ SQLiteAssetHelper

SQLiteAssetHelper adalah kelas abstrak yang diimplementasikan untuk mengkoneksikan database, jadi database dan table bisa dibuat dari aplikasi sqlite expert.

#### ❖ SQLiteOpenHelper

SQLiteOpenHelper adalah kelas abstrak yang diimplementasikan untuk membuat, membuka dan meningkatkan database dengan meng-Override method `onCreate ()` dan



onUpgrade (). Method onCreate () dipanggil otomatis ketika aplikasi dijalankan pertama kali, biasanya digunakan untuk membuat database. Jika aplikasi yang dibuat merupakan versi terbaru dan database pada phone perlu diperbarui, maka gunakan method onUpgrade ().

#### ❖ Query Database untuk Mendapatkan Data

Setiap query database dikembalikan sebagai cursor. Untuk mengeksekusi query pada database menggunakan method query ().

**namaSQLiteDatabase.query**

(Tabel, Kolom, Seleksi, Argument Seleksi, groupBy, having, orderBy)

Jika ingin digunakan sintaks dari bahasa SQL, selain method tersebut dapat juga digunakan method.rawQuery ().

**namaSQLiteDatabase.rawQuery**

(perintah SQL, argument seleksi)

Penggunaan method method ini akan mengembalikan nilai sebagai hasil cursor. Cursor ini sebagai penunjuk dari hasil query yang telah dilakukan. Untuk mengambil nilai dari cursor, pertama menggunakan method moveToPosition (indeks baris) memindahkan cursor ke posisi baris yang diinginkan. Kemudian ambil data yang ingin diperoleh menggunakan method get 'jenis data' (indeks kolom). Jenis data merupakan jenis dari data yang ingin diperoleh misalnya string, int, float. Untuk memperoleh jenis data dari kolom tersebut.

#### ❖ Query untuk Insert, Edit dan Delete

Kelas SQLiteDatabase menyediakan perintah untuk mengeksekusi perintah insert, edit, dan delete. Selain itu, method yang disediakan memungkinkan untuk melakukan suatu perintah SQL yang valid pada tabel database yang diinginkan.

##### 1. Insert

Untuk membuat baris baru, dengan membangun sebuah objek ContentValues dan menggunakan method put () untuk memberikan nilai bagi setiap kolom. Insert baris baru dengan melewati objek ContentValues menjadi method insert () pada database target bersama dengan nama tabel.

**Sintaks :**

**namaSQLiteDatabase.insert**

(Table, Nilai Inisial, Objek ContentValues)

## 2. Edit

Memperbarui baris juga menggunakan ContentValues. Pertama buat objek ContentValues baru, menggunakan method put () untuk menetapkan nilai baru untuk setiap kolom yang ingin diperbarui. Panggil method update () pada database, dengan melewati nama tabel, nilai-nilai diperbarui objek dan klausa where yang menentukan baris yang akan diperbarui.

**Sintaks :**

**namaSQLiteDatabase.update**

(table, objek ContentValues, klausa where, argument where)

## 3. Delete

Menghapus baris dapat dilakukan dengan menggunakan method delete () pada sebuah database, menentukan nama tabel dan baris mana yang ingin dihapus dengan menggunakan klausa where.

**Sintaks :**

**namaSQLiteDatabase.delete**

(table, klausa where, argument where)

Klausa where ini dapat langsung digunakan dengan langsung menulis argumen pada klausa tersebut sedangkan apabila ingin menggunakan argument where dapat digunakan.

**Sintaks :**

**namaSQLiteDatabase.delete**

(table, "namatabel=?", argument where)

Selain menggunakan method tersebut, dapat digunakan method execSQL (). Method ini digunakan untuk mengeksekusi perintah SQL yang diberikan.

**Sintaks :**

**namaSQLiteDatabase.execSQL**

(Perintah SQL).

# 6

## ANDROID CLIENT-SERVER USING JSON



## 6.1 Pengertian Client Server

Client merupakan sembarang sistem atau proses yang melakukan suatu permintaan data atau layanan ke server sedangkan server ialah, sistem atau proses yang menyediakan data atau layanan yang diminta oleh client.

Client-Server adalah pembagian kerja antara server dan client yg mengakses server dalam suatu jaringan. Jadi arsitektur client-server adalah desain sebuah aplikasi terdiri dari client dan server yang saling berkomunikasi ketika mengakses server dalam suatu jaringan.

### 1. Servis (layanan)

- Hubungan antara proses yang berjalan pada mesin yang berbeda
- Pemisahan fungsi berdasarkan ide layanannya
- Server sebagai provider, client sebagai konsumen

**2. Sharing resources (sumber daya):** Server bisa melayani beberapa client pada waktu yang sama, dan meregulasi akses bersama untuk share sumber daya dalam menjamin konsistensinya.

**3. Asymmetrical protocol (protokol yang tidak simetris ):** Many-to-one relationship antara client dan server. Client selalu menginisiasikan dialog melalui layanan permintaan, dan server menunggu secara pasif request dari client.

**4. Transparansi lokasi:** Proses yang dilakukan server boleh terletak pada mesin yang sama atau pada mesin yang berbeda melalui jaringan. Lokasi server harus mudah diakses dari client.

## **5. Mix-and-Match : Perbedaan server client platforms**

**6. Pesan berbasiskan komunikasi;** Interaksi server dan client melalui pengiriman pesan yang menyertakan permintaan dan jawaban.

**7. Pemisahan interface dan implementasi:** Server bisa diupgrade tanpa mempengaruhi client selama interface pesan yang diterbitkan tidak berubah.

- Client Server System
- Client Server Application

## **6.2 Fungsi Client Server**

Dalam konteks basis data, client mengatur interface berfungsi sebagai workstation tempat menjalankan aplikasi basis data. Client menerima permintaan pemakai, memeriksa sintaks dan generate kebutuhan basis data dalam SQL atau bahasa yang lain. Kemudian meneruskan pesan ke server, menunggu response dan bentuk response untuk pemakai akhir. Server menerima dan memproses permintaan basis data kemudian mengembalikan hasil ke client.

Proses-proses ini melibatkan pemeriksaan otorisasi, jaminan integritas, pemeliharaan data dictionary dan mengerjakan query serta proses update. Selain itu juga menyediakan kontrol terhadap concurrency dan recovery.

Ada beberapa keuntungan jenis arsitektur ini adalah :

- Memungkinkan akses basis data yang besar
- Menaikkan kinerja
- Jika client dan server diletakkan pada komputer yang berbeda kemudian CPU yang berbeda dapat memproses aplikasi secara paralel. Hal ini mempermudah merubah mesin server jika hanya memproses basis data.
- Biaya untuk hardware dapat dikurangi
- Hanya server yang membutuhkan storage dan kekuatan proses yang cukup untuk menyimpan dan mengatur basis data
- Biaya komunikasi berkurang

- Aplikasi menyelesaikan bagian operasi pada client dan mengirimkan hanya bagian yang dibutuhkan untuk akses basis data melewati jaringan, menghasilkan data yang sedikit yang akan dikirim melewati jaringan
- Meningkatkan kekonsistenan
- Server dapat menangani pemeriksaan integrity sehingga batasan perlu didefinisikan dan validasi hanya di satu tempat, aplikasi program mengerjakan pemeriksaan sendiri
- Map ke arsitektur open-system dengan sangat alami

Berikut ini adalah ringkasan fungsi client-server

### 1. Client

- Mengatur user interface
- Menerima dan memeriksa sintaks input dari pemakai
- Memproses aplikasi
- Generate permintaan basis data dan memindahkannya ke server
- Memberikan response balik kepada pemakai
- Menyediakan akses basis data secara bersamaan
- Menyediakan kontrol recovery

## 6.3 Aplikasi client server

Istilah arsitektur mengacu pada desain sebuah aplikasi, atau dimana komponen yang membentuk suatu system ditempatkan dan bagaimana mereka berkomunikasi.

Macam-macam arsitektur aplikasi Client-Server beserta kelebihan dan kekurangannya yaitu:

### 1. Standalone (one-tier)

Pada arsitektur ini semua pemrosesan dilakukan pada mainframe. Kode aplikasi, data dan semua komponen sistem ditempatkan dan dijalankan pada host. Walaupun computer client dipakai untuk mengakses mainframe, tidak ada pemrosesan yang terjadi pada mesin ini, dan karena mereka “dump- client” atau “dump-terminal”. Tipe model ini, dimana semua pemrosesan terjadi secara terpusat, dikenal sebagai

berbasis-host. Sekilas dapat dilihat kesalahan pada model ini. Ada dua masalah pada komputasi berbasis host: Pertama, semua pemrosesan terjadi pada sebuah mesin tunggal, sehingga semakin banyak user yang mengakses host, semakin kewalahan jadinya. Jika sebuah perusahaan memiliki beberapa kantor pusat, user yang dapat mengakses mainframe adalah yang berlokasi pada tempat itu, membiarkan kantor lain tanpa akses ke aplikasi yang ada.

Pada saat itu jaringan sudah ada namun masih dalam tahap bayi, dan umumnya digunakan untuk menghubungkan terminal dump dan mainframe. Namun keterbatasan yang dikenakan pada user mainframe dan jaringan telah mulai dihapus.

**Keuntungan arsitektur standalone (one-tier):**

- Sangat mudah
- Cepat dalam merancang dan mengaplikasikan

**Kelemahan arsitektur standalone (one-tier):**

- Skala kecil
- Susah diamankan
- Menyebabkan perubahan terhadap salah satu komponen diatas tidak mungkin dilakukan, karena akan mengubah semua bagian.
- Tidak memungkinkan adanya re-usable component dan code.
- Cepat dalam merancang dan mengaplikasikan

## **2. Client/Server (two tier)**

Dalam model client/server, pemrosesan pada sebuah aplikasi terjadi pada client dan server. Client/server adalah tipikal sebuah aplikasi two-tier dengan banyak client dan sebuah server yang dihubungkan melalui sebuah jaringan.

Aplikasi ditempatkan pada computer client dan mesin database dijalankan pada server jarak-jauh. Aplikasi client mengeluarkan permintaan ke database yang mengirimkan kembali data ke client-nya.

Model Two-tier terdiri dari tiga komponen yang disusun menjadi dua lapisan : client (yang meminta service) dan server (yang menyediakan service).

**Tiga komponen tersebut yaitu :**

1. User Interface. Adalah antar muka program aplikasi yang berhadapan dan digunakan langsung oleh user.
2. Manajemen Proses.
3. Database. Model ini memisahkan peranan user interface dan database dengan jelas, sehingga terbentuk dua lapisan.

**Kelebihan dari model client/server :**

- Mudah
- Menangani Database Server secara khusus
- Relatif lebih sederhana untuk di develop dan diimplementasikan.
- Lebih cocok diterapkan untuk bisnis kecil.

Server database berisi mesin database, termasuk tabel, prosedur tersimpan, dan trigger (yang juga berisi aturan bisnis). Dalam system client/server, sebagian besar logika bisnis biasanya diterapkan dalam database.

**Server database menangani :**

- Manajemen data
- Keamanan
- Query, trigger, prosedur tersimpan
- Penangan kesalahan

Arsitektur client/server merupakan sebuah langkah maju karena mengurangi beban pemrosesan dari komputer sentral ke computer client. Ini berarti semakin banyak user bertambah pada aplikasi client/server, kinerja server file tidak akan menurun dengan



cepat. Dengan client/server user dari berbagai lokasi dapat mengakses data yang sama dengan sedikit beban pada sebuah mesin tunggal.

Namun masih terdapat kelemahan pada model ini. Selain menjalankan tugas-tugas tertentu, kinerja dan skalabilitas merupakan tujuan nyata dari sebagian besar aplikasi.

**Kekurangan dari model client/server :**

- Kurangnya skalabilitas
- Koneksi database dijaga
- Tidak ada keterbaharuan kode
- Tidak ada tingkat menengah untuk menangani keamanan dan transaksi skala kecil.
- Susah di amankan.
- Lebih mahal.

### **3. Three Tier**

Arsitektur Three Tier merupakan inovasi dari arsitektur Client Server. Pada arsitektur Three Tier ini terdapat Application Server yang berdiri di antara Client dan Database Server. Contoh dari Application server adalah IIS, WebSphere, dan sebagainya.

Application Server umumnya berupa business process layer, dimana bisa dikembangkan menggunakan PHP, ASP.Net, maupun Java. Sehingga kita menempatkan beberapa business logic kita pada tier tersebut. Arsitektur Three Tier ini banyak sekali diimplementasikan dengan menggunakan Web Application. Karena dengan menggunakan Web Application, Client Side (Komputer Client) hanya akan melakukan instalasi Web Browser. Dan saat komputer client melakukan inputan data, maka data tersebut dikirimkan ke Application Server dan diolah berdasarkan business process-nya. Selanjutnya Application Server akan melakukan komunikasi dengan database server.

Biasanya, implementasi arsitektur Three Tier terkendala dengan network bandwidth. Karena aplikasinya berbasis web, maka Application Server selalu mengirimkan Web Application-nya ke computer Client. Jika kita memiliki banyak sekali client, maka bandwidth yang harus disiapkan akan cukup besar, Sedangkan network bandwidth biasanya memiliki limitasi. Oleh karena itu biasanya, untuk mengatasi masalah ini, Application Server ditempatkan pada sisi client dan hanya mengirimkan data ke dalam database server. Konsep model three-tier adalah model yang membagi fungsionalitas ke dalam lapisan-lapisan, aplikasiaplikasi mendapatkan skalabilitas, keterbaharuan, dan keamanan.

#### **Kelebihan arsitektur Three Tier :**

- Segala sesuatu mengenai database terinstalasikan pada sisi server, begitu pula dengan pengkonfigurasian. Hal ini membuat harga yang harus dibayar lebih kecil.
- Apabila terjadi kesalahan pada salah satu lapisan tidak akan menyebabkan lapisan lain ikut salah
- Perubahan pada salah satu lapisan tidak perlu menginstalasi ulang pada lapisan yang lainnya dalam hal ini sisi server ataupun sisi client.
- Skala besar.
- Keamanan dibelakang firewall.
- Transfer informasi antara web server dan server database optimal.
- Komunikasi antara system-sistem tidak harus didasarkan pada standart internet, tetapi dapat menggunakan protocol komunikasi yang lebih cepat dan berada pada tingkat yang lebih rendah.
- Penggunaan middleware mendukung efisiensi query database dalam SQL di pakai untuk menangani pengambilan informasi dari database.

#### **Kekurangan arsitektur Three Tier :**

- Lebih susah untuk merancang
- Lebih susah untuk mengatur
- Lebih mahal

#### **4. Multi Tier**

Arsitektur Multi Tier adalah suatu metode yang sangat mirip dengan Three Tier. Bedanya, pada Multi Tier akan diperjelas bagian UI (User Interface) dan Data Processing. Yang membedakan arsitektur ini adalah dengan adanya Business Logic Server. Database Server dan Business Logic Server merupakan bagian dari Data Processing, sedangkan Application Server dan Client/Terminal merupakan bagian dari UI. Business Logic Server biasanya masih menggunakan bahasa pemrograman terdahulu, seperti COBOL. Karena sampai saat ini, bahasa pemrograman tersebut masih sangat mumpuni sebagai business process.

Multi-tier architecture menyuguhkan bentuk three – tier yang diperluas dalam model fisik yang terdistribusi. Application server dapat mengakses Application server yang lain untuk mendapat data dari Data server dan mensuplai servis ke client Application.

#### **Kelebihan arsitektur Multi Tier :**

- Dengan menggunakan aplikasi multi-tier database, maka logika aplikasi dapat dipusatkan pada middle-tier, sehingga memudahkan untuk melakukan control terhadap client-client yang mengakses middle server dengan mengatur seting pada dcomcnfg.
- Dengan menggunakan aplikasi multi-tier, maka database driver seperti BDE/ODBC untuk mengakses database hanya perlu diinstal sekali pada middle server, tidak perlu pada masing-masing client.
- Pada aplikasi multi-tier, logika bisnis pada middle-tier dapat digunakan lagi untuk mengembangkan aplikasi client lain, sehingga mengurangi besarnya program untuk mengembangkan aplikasi lain. Selain itu meringankan beban pada tiap-tiap mesin karena program terdistribusi pada beberapa mesin.

- Memerlukan adaptasi yang sangat luas ruang lingkupnya apabila terjadi perubahan sistem yang besar.

#### **Kekurangan arsitektur Multi Tier :**

- Program aplikasi tidak bisa mengquery langsung ke database server, tetapi harus memanggil prosedur-prosedur yang telah dibuat dan disimpan pada middle-tier.
- Lebih mahal

#### **Keunggulan Client/Server**

- Kecepatan akses lebih tinggi
- Sistem keamanan & administrasi lebih baik
- Sistem backup data lebih baik

#### **Kelemahan Client/Server**

- Biaya lebih mahal
- Dibutuhkan komputer dengan spesifikasi khusus untuk menjadi server
- Ketergantungan terhadap server, jika server terganggu maka keseluruhan jaringan terganggu

## **6.4 Kriptography**

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Menurut Bruce Schneier dalam bukunya "Applied Cryptography", kriptografi adalah ilmu pengetahuan dan seni menjaga message-message agar tetap aman (secure).

Konsep kriptografi sendiri telah lama digunakan oleh manusia misalnya pada peradaban Mesir dan Romawi walau masih sangat sederhana. Prinsip-prinsip yang mendasari kriptografi yakni:

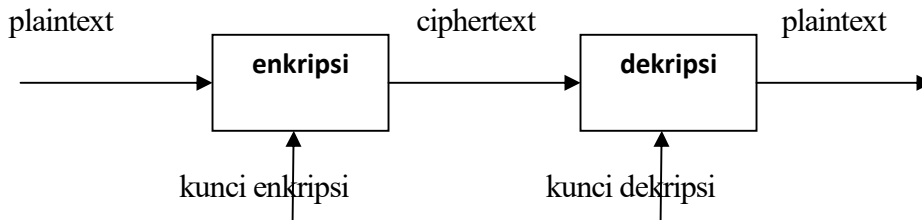
- *Confidelity* (kerahasiaan) yaitu layanan agar isi pesan yang dikirimkan tetap rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima / pihak-pihak memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatu algoritma matematis yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.
- *Data integrity* (keutuhan data) yaitu layanan yang mampu mengenali/mendeteksi adanya manipulasi (penghapusan, perubahan atau penambahan) data yang tidak sah (oleh pihak lain).
- *Authentication* (keotentikan) yaitu layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/informasi.
- *Non-repudiation* (anti-penyangkalan) yaitu layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dirinya).

Berbeda dengan kriptografi klasik yang menitikberatkan kekuatan pada kerahasiaan algoritma yang digunakan (yang artinya apabila algoritma yang digunakan telah diketahui maka pesan sudah jelas "bocor" dan dapat diketahui isinya oleh siapa saja yang mengetahui algoritma tersebut), kriptografi modern lebih menitikberatkan pada kerahasiaan kunci yang digunakan pada algoritma tersebut (oleh pemakainya) sehingga algoritma tersebut dapat saja disebarluaskan ke kalangan masyarakat tanpa takut kehilangan kerahasiaan bagi para pemakainya.

Berikut adalah istilah-istilah yang digunakan dalam bidang kriptografi :

- **Plaintext** (M) adalah pesan yang hendak dikirimkan (berisi data asli).
- **Ciphertext** (C) adalah pesan ter-enkrip (tersandi) yang merupakan hasil enkripsi.
- **Enkripsi** (fungsi E) adalah proses pengubahan *plaintext* menjadi *ciphertext*.
- **Dekripsi** (fungsi D) adalah kebalikan dari enkripsi yakni mengubah *ciphertext* menjadi *plaintext*, sehingga berupa data awal/asli.
- **Kunci** adalah suatu bilangan yang dirahasiakan yang digunakan dalam proses enkripsi dan dekripsi.

Kriptografi itu sendiri terdiri dari dua proses utama yakni proses enkripsi dan proses dekripsi. Seperti yang telah dijelaskan di atas, proses enkripsi mengubah *plaintext* menjadi *ciphertext* (dengan menggunakan kunci tertentu) sehingga isi informasi pada pesan tersebut sukar dimengerti.



Algoritma kriptografi berdasarkan jenis kunci yang digunakan dapat dibedakan menjadi dua jenis yaitu :

- Algoritma *simetris*

Dimana kunci yang digunakan untuk proses enkripsi dan dekripsi adalah kunci yang sama

- Algoritma *asimetris*

Dimana kunci yang digunakan untuk proses enkripsi dan dekripsi menggunakan kunci yang berbeda.

Sedangkan berdasarkan besar data yang diolah dalam satu kali proses, maka algoritma kriptografi dapat dibedakan menjadi dua jenis yaitu :

- Algoritma *block cipher*

Informasi/data yang hendak dikirim dalam bentuk blok-blok besar (misal 64-bit) dimana blok-blok ini dioperasikan dengan fungsi enkripsi yang sama dan akan menghasilkan informasi rahasia dalam blok-blok yang berukuran sama.

- Algoritma *stream cipher*

Informasi/data yang hendak dikirim dioperasikan dalam bentuk blok-blok yang lebih kecil (byte atau bit), biasanya satu karakter persatuan persatuan waktu proses, menggunakan transformasi enkripsi yang berubah setiap waktu.

## 6.5 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.

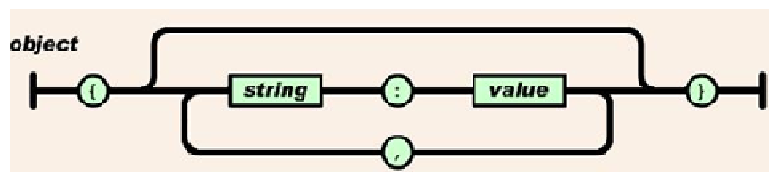
JSON terbuat dari dua struktur:

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 pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas 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

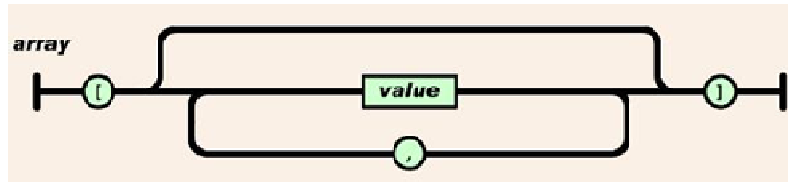
Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



#### 2. Larik

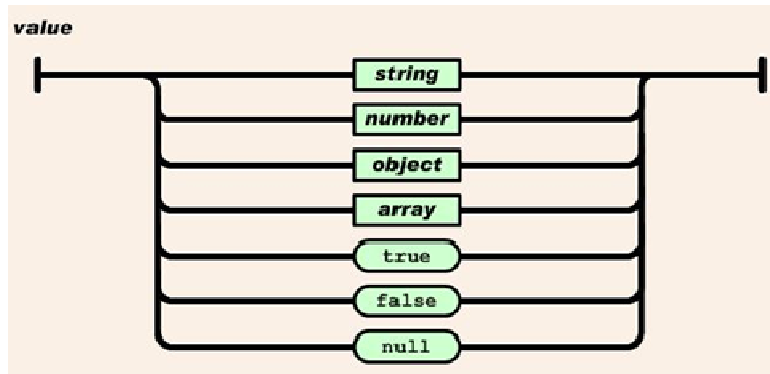
Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [ (kurung kotak buka) dan diakhiri dengan ] (kurung kotak tutup). Setiap nilai dipisahkan oleh

(koma).



### 3. Nilai

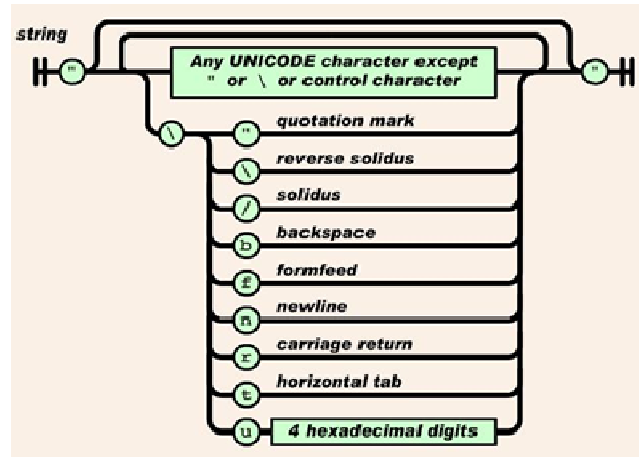
Nilai (value) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau true atau false atau null, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



### 4. String

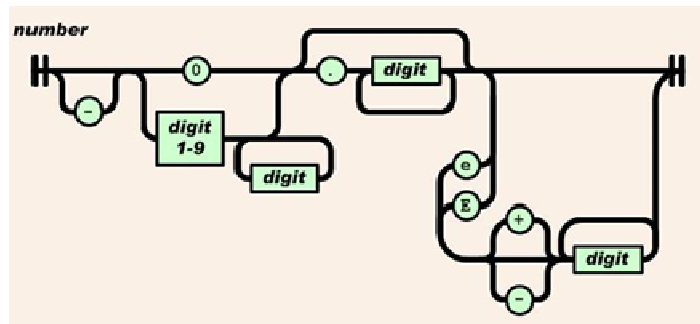
String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan backslash escapes "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.





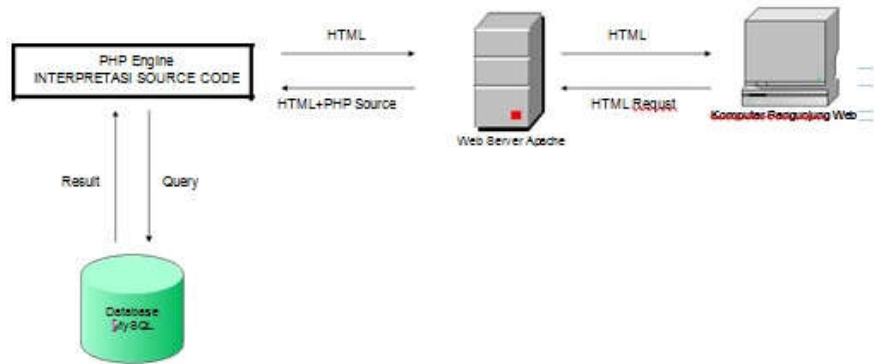
## 5. Angka

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



## 6.6 PHP

PHP adalah bahasa skrip dalam server (server-side embedded scripting language). Artinya, PHP bekerja di dalam HTML dengan tugas membuat isi dokumen sesuai permintaan. Alur kerja PHP itu sendiri dapat kita lihat pada gambar berikut:



Pada gambar di atas tampak alur kerja engine PHP. Pada awalnya, sebuah komputer pengunjung web melakukan HTTP request terhadap halaman tertentu. Jika halaman yang di-request tersebut adalah halaman PHP, maka Web Server Apache akan meneruskan halaman PHP tersebut ke PHP engine. PHP engine akan melakukan interpretasi terhadap source PHP dalam halaman tersebut, dengan bantuan dari database jika perlu. Setelah interpretasi selesai, maka PHP akan mengembalikan hasilnya yang berupa HTML murni (tanpa adanya source PHP sama sekali) kepada Web Server Apache. Setelah itu Web Server Apache akan meneruskannya kepada komputer pengunjung web tadi dalam bentuk HTML yang dapat ditampilkan oleh internet browser. PHP tidak hanya didesain untuk Web Server Apache dan database MySQL. Bahkan PHP tidak hanya bisa dipakai untuk halaman web.

PHP dapat berjalan pada hampir semua web server yang mendukung konsep CGI-binary atau ISAPI module, di antaranya Personal Web Server (PWS) 3 ke atas, Internet Information Service (IIS) 3 ke atas, Apache, OmniHTTPd 2 ke atas, Oreilly Website Pro, Xitami, dan Netscape Enterprise Server. Koneksi database pada PHP amat mudah, cukup dengan menggunakan 2 atau 3 baris perintah saja. Fungsi-fungsi untuk database cukup lengkap, dan database yang didukung amat banyak, di antaranya MySQL, PostgreSQL, dBase, DB++, FrontBase, ODBC, Sybase-CT, Oracle, Informix, dan Interbase. PHP memiliki empat kelebihan utama yang menarik minat banyak pengguna. Kelebihan utama PHP tersebut diringkas dalam 4P berikut:

1. **Practicality.** PHP dibuat dengan menitikberatkan pada kepraktisan. Hasilnya, PHP adalah bahasa pemrograman minimalis, dilihat dari segi kebutuhan pengguna dan kebutuhan sintaks.

2. Power. PHP memiliki banyak kemampuan, mulai dari kemampuan untuk terhubung dengan basis data, membuat halaman web dinamis, membuat dan memanipulasi berkas gambar, Flash dan PDF, berkomunikasi dengan bermacam protokol seperti IMAP dan POP3, dan masih banyak lagi.
3. Possibility. PHP dapat menyediakan lebih dari satu solusi untuk suatu masalah
4. Price. PHP selalu dirilis kepada publik tanpa ada batasan penggunaan, modifikasi, atau redistribusi.

## 6.7 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

### 1. Sejarah MySQL

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level ISAM database engine dengan indexing. Pada saat itu Monty bekerja pada perusahaan bernama TcX di Swedia. TcX pada tahun 1994 mulai mengembangkan aplikasi berbasis web, dan berencana menggunakan UNIREG sebagai sistem database. Namun sayangnya, UNIREG dianggap tidak cocok untuk database yang dinamis seperti web. TcX kemudian mencoba mencari alternatif sistem database lainnya, salah satunya adalah mSQL (miniSQL). Namun mSQL versi 1 ini juga memiliki kekurangan, yaitu tidak mendukung indexing, sehingga performanya tidak terlalu bagus.

Dengan tujuan memperbaiki performa mSQL, Monty mencoba menghubungi David Hughes (programmer yang mengembangkan mSQL) untuk menanyakan apakah ia tertarik mengembangkan sebuah konektor di mSQL yang

dapat dihubungkan dengan UNIREG ISAM sehingga mendukung indexing. Namun saat itu Hughes menolak, dengan alasan sedang mengembangkan teknologi indexing yang independen untuk mSQL versi 2. Dikarenakan penolakan tersebut, David Hughes, TcX (dan juga Monty) akhirnya memutuskan untuk merancang dan mengembangkan sendiri konsep sistem database baru. Sistem ini merupakan gabungan dari UNIREG dan mSQL (yang source codenya dapat bebas digunakan). Sehingga pada May 1995, sebuah RDBMS baru, yang dinamakan MySQL dirilis.

David Axmark dari Detron HB, rekanan TcX mengusulkan agar MySQL di 'jual' dengan model bisnis baru. Ia mengusulkan agar MySQL dikembangkan dan dirilis dengan gratis. Pendapatan perusahaan selanjutnya di dapat dari menjual jasa "support" untuk perusahaan yang ingin mengimplementasikan MySQL. Konsep bisnis ini sekarang dikenal dengan istilah Open Source. Pada tahun 1995 itu juga, TcX berubah nama menjadi MySQL AB, dengan Michael Widenius, David Axmark dan Allan Larsson sebagai pendirinya. Titel "AB" dibelakang MySQL, adalah singkatan dari "Aktiebolag", istilah PT (Perseroan Terbatas) bagi perusahaan Swedia.

#### **Kelebihan :**

1. Berlisensi GPL dan Multi Platform.
2. Dapat diintegrasikan dengan beberapa bahasa Pemrograman seperti .Net, Java, Python, Perl yang merupakan bahasa pemrograman yang paling dominan di kalangan programmer.
3. Mendukung ODBC untuk sistem operasi Windows sehingga bisa digunakan aplikasi yang berjalan di windows.
4. Bisa dijalankan pada spesifikasi hardware yang rendah karena lebih hemat resource memory (dibandingkan database lain) sehingga mudah digunakan untuk bahan pembelajaran.
5. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari 20 bahasa meskipun bahasa indonesia belum termasuk didalamnya.

**Kekurangan :**

1. Banyak mengklaim kurang support terhadap pemrograman Visual/Desktop, sehingga sedikit yang menggunakan untuk aplikasi visual.
2. Karena berlisensi GPL sehingga sulit mendapatkan update untuk problem yang urgent, sehingga perusahaan skala menengah keatas lebih memilih RDBMS berlisensi dan disupport seperti Oracle dan MS SQL Server
3. Sangat diragukan dalam menangani data skala besar, karena ada beberapa opini yang pro dan kontra terhadap kemampuan MySQL terhadap pengolahan data yang besar.

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. 'Performance tuning', MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

# 7

## ANDROID GOOGLE MAPS API V2





## BAB 7

# ANDROID GOOGLE MAPS API V2

### 7.1. Google Maps

Kehadiran akan Google Maps pada saat ini sudah dapat dikatakan sebagai kebutuhan bagi masyarakat banyak. Hal tersebut timbul karena seseorang jika ingin berpergian ke suatu tempat yang dinilai asing atau baru, mereka pasti memanfaatkan Google Map app untuk memberikan petunjuk arah agar sampai ditujuan.

Google Maps merupakan sebuah layanan peta dunia virtual berbasis web yang dikembangkan oleh Google. Layanan ini gratis dan dapat dengan mudah ditemukan di <http://maps.google.com>. Google Maps menawarkan peta yang dapat digeser (panned), diperbesar (zoom in), diperkecil (zoom out), dapat diganti dalam beberapa mode (map, satelit, hybrid dan lain-lain), fitur pencarian rute (routing), penunjuk arah dari suatu objek peta ke objek yang lain (direction) dan juga pencari tempat (place). Google Maps akan menampilkan peta sebuah wilayah yang dibuat secara digital. Sehingga, cukup mudah untuk dapat mengetahui peta sebuah wilayah, cukup dengan masuk ke situs google map tersebut dan menekan tombol atau mengetikkan sebuah alamat. Maka, lokasi yang dicari atau diinginkan akan segera ditemukan tanpa menunggu waktu lama.

Google map dapat disertai pada third-party website (website pengguna) dengan menggunakan Google Maps API. Layanan Google Maps ini merangkum peta jalanan, perencanaan rute jalan untuk pejalan kaki, mobil, sepeda dan juga transportasi umum. Selain itu Google Maps juga digunakan sebagai penanda lokasi / alamat dari badan usaha / organisasi di beberapa negara di dunia. Data – data peta yang ada pada Google Maps tersebut tidak terupdate secara Real-Time, namun Google menyimpan data dari gambar satelit tersebut pada database utama dan kebanyakan gambar yang ada pada layanan tersebut diambil kurang dari 3 tahun lamanya.

#### 7.1.1. Google Map View

Google Maps menyediakan gambar udara dari satelit dengan resolusi tinggi dari seluruh dunia. Namun, tidak semua daerah pada gambar satelit tersebut memiliki resolusi



yang sama, area yang berpenduduk rendah biasanya tampil dengan detail gambar yang kurang. Walaupun Google menggunakan satelit dalam pengambilan gambar tersebut, kadangkala gambar perkotaan dengan resolusi tinggi diambil dari pesawat terbang di ketinggian 240 meter sampai 460 meter, namun tetap kebanyakan berasal dari satelit.

Google Maps menyediakan beberapa pilihan bagi kita sebagai developer untuk memilih view yang akan digunakan. Beberapa pilihan tersebut diantaranya Satellite, RoadMap, Hybrid, dan Terrain. Masing – masing view tersebut digunakan sesuai dengan kebutuhan kita sebagai developer.

### **7.1.2. Directions**

Google Maps menyediakan layanan perencanaan rute dalam fitur “Get Directions” pada aplikasi website. Terdapat 4 tipe mode transportasi yang dapat digunakan dalam area yaitu driving, public Transit, walking dan bicycling. Dalam Google Street View, hal hal seperti parkir, putaran arah dan jalan satu arah dapat dilihat sebelum memulai rute.

### **7.1.3 Polylines**

Google Maps menyediakan layanan polylines, dimana layanan ini berfungsi untuk menyempurnakan layanan direction. Layanan ini digunakan secara bersamaan dengan direction dimana saat kita sebagai developer telah menentukan direction dari app yang kita buat maka tugas polylines adalah menghubungkan 2 rute atau lebih dengan cara memberikan warna terhadap rute terkait.

## **7.2. Google Maps Untuk Perangkat Mobile**

Pada Oktober 2005, Google memperkenalkan sebuah aplikasi berbasis java bernama Google Maps for Mobile, yang dijalankan pada ponsel atau perangkat mobile berbasis Java, terdapat banyak fitur dari versi web yang juga disertai pada aplikasi mobile tersebut.

Pada tanggal 4 Oktober 2009, Google Maps Navigation dirilis oleh Google untuk Android 2.0 Éclair pada Motorola Droid dengan fitur voice commands, traffic reports dan street view support. Pada Maret 2011, Vice President of Location Service Google, Marrison Mayer mengumumkan bahwa layanan peta Google telah digunakan oleh 150 juta pengguna.

Pada bulan Juni 2012, Apple mengumumkan bahwa akan mengganti aplikasi Google Maps dengan aplikasi layanan peta dari Apple sendiri dari iOS 6. Namun, pada Desember 2012, Google mengumumkan ketersediaan Google Maps untuk perangkat iOS Apple di Apple App Store. Hanya berlangsung 1 jam sejak dirilis, Google Maps iOS menjadi top free app di App Store.

### 7.3. Google Maps Android API v2

Untuk membuat aplikasi android baru yang menggunakan Google Maps Android API v2, dibutuhkan beberapa tahap. Beberapa dari langkah langkah tersebut hanya akan dilakukan sekali saja, tetapi beberapa informasi akan menjadi referensi penting untuk pengembangan aplikasi kedepannya. Berikut adalah langkah langkah proses untuk menambahkan peta di Aplikasi Android :

1. Install [Android SDK](#).
2. Download dan konfigurasi Google Play Service SDK yang telah disertai dengan Google Maps Android API melalui SDK Manager. Jika akan menggunakan Google Maps Mobile SDK untuk Business maka harus men-download versi Google Maps Mobile SDK for Business static library.
3. Men-generate kode API key. Untuk mendapatkan kode API Key, developer harus mendaftarkan proyek pada Google APIs Console, dan mendapatkan tanda certificate untuk aplikasi.
4. Menambahkan beberapa konfigurasi penting pada Android application's manifest
5. Menambahkan fragment peta ke dalam aplikasi.
6. Mempublikasi aplikasi.

#### Menspesifikasi Permissions Pada Android Manifest

Untuk dapat menggunakan akses Google Maps pada aplikasi android, maka developer harus menspesifikasikan permissions yang dibutuhkan pada aplikasi, dengan menambahkan tag `<uses-permission>` elements sebagai anak dari tag `<manifest>` element. Berikut adalah sintaks umumnya :

```
<uses-permissionandroid:name="permission_name"/>
```

Sebagai contoh, agar aplikasi dapat mengakses internet maka butuh menambahkan request Internet permission, seperti berikut:

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

Selain digunakan untuk bagian dari aplikasi, permission juga diperlukan untuk dapat menggunakan Google Maps Android API. Berikut adalah beberapa permission yang perlu disertakan di AndroidManifest.XML agar dapat menggunakan Google Maps Android API :

1. **android.permission.INTERNET**, Digunakan oleh API untuk mendownload proyeksi peta dari Google Maps Server dengan Internet.
2. **android.permission.ACCESS\_NETWORK\_STATE**, Mengizinkan API untuk mengecek status koneksi perangkat untuk dapat mengalokasikan data data yang akan didownload.
3. **com.google.android.providers.gsf.permission.READ\_GSERVICES**, Memberikan Akses kepada API untuk mengakses layanan Google berbasis web.
4. **android.permission.WRITE\_EXTERNAL\_STORAGE**, Mengizinkan API untuk menyimpan cache map tile data padamedia penyimpanan external perangkat.
5. **com.scopproject.maps.permission.MAPS\_RECEIVE**, Mengizinkan bahwa aplikasi untuk menerima fragment dari Google Maps.
6. **android.permission.ACCESS\_COARSE\_LOCATION**, Mengizinkan API untuk dapat menggunakan WiFi ataupun paket data untuk mengirimkan lokasi perangkat.
7. **android.permission.ACCESS\_FINE\_LOCATION**, Mengizinkan API menggunakan Global Positioning System (GPS) untuk mengirim lokasi perangkat dari area yang lebih detail.

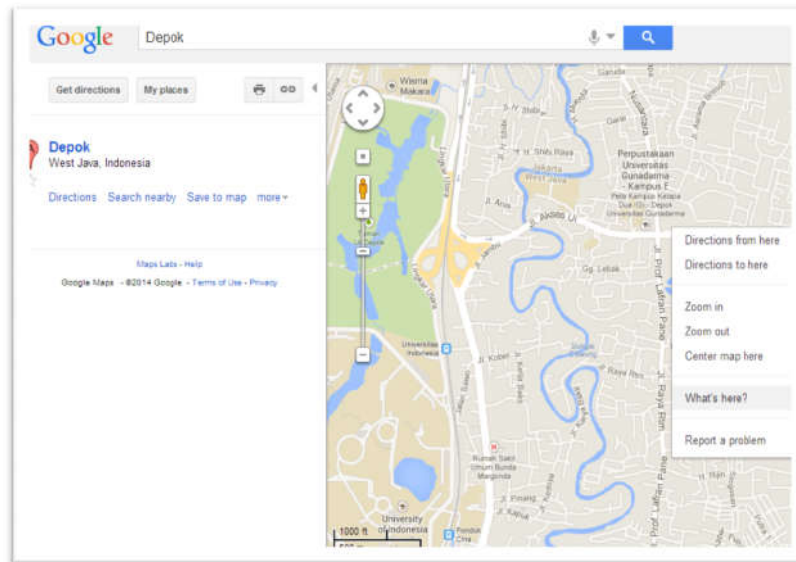
#### 7.4.1. Geo Point

Geo Point adalah sebuah kelas tetap yang mewakili sepasang lintang dan bujur, disimpan sebagai angka integer dari microdegrees. Geo Point biasanya digunakan dalam aplikasi pemetaan, untuk memanggilnya dapat digunakan perintah import sebagai berikut:

```
import com.google.android.maps.GePoint;
```

GePoint bisa didapatkan salah satunya dengan cara melihat pada pada Google Maps. Misalkan arah tujuan menuju ke Indonesia. Dan disini akan diambil

GeoPoint untuk kota Depok dengan titik Universitas Gunadarma. Klik kanan, lalu pilih What's here?



Gambar 7.1 Mendapatkan Geo Point dari Google Maps

Sehingga pada jendela sebelah kiri akan muncul angka GeoPoint dari titik tersebut, atau pada field searching. Untuk Universitas Gunadarma kita mendapatkan GeoPoint dengan deretan angka, -6.36806, 106.833029.

#### 7.4.2. Map Fragment

Map Fragment adalah sub class dari Google Library Maps yang sebelumnya diimport. Fungsi dari Map Fragment adalah sebagai wadah untuk menampilkan maps pada layout xml Android. Map Fragment memiliki constructor yang memang telah tergenerate secara otomatis saat developer mendeklarasikannya.

#### 7.4.3. Map View

Map View adalah sub class dari class View yang berfungsi untuk menempatkan map pada view di Android. Map view ini lebih kepada sintak yang digunakan pada activity Android bukan ke pada komponen layout seperti point 7.4.2. Sebuah view data direpresentasikan sebagai bentuk persegi dalam area layar, dan membentuk blok – blok pada aplikasi android dengan widget – widget. Seperti halnya MapFragment, Map View ini akan menampung Google Maps object.

#### 7.4.4. Tipe – Tipe Google Maps Android API

Ada berbagai macam tipe maps yang tersedia oleh Google Maps Android API. Tipe peta pada dasarnya merepresentasikan peta. Contohnya, sebuah atlas biasanya terdapat peta politik yang focus untuk menunjukkan batas dari negara-negara berdasarkan warna dan peta jalan yang fokus untuk menunjukkan jalan pada suatu kota atau negara.

Google Maps Android API menyediakan berbagai macam peta, yaitu sebagai berikut :

1. Normal

Tipe peta jalanan. Menampilkan fokus jalanan, tempat - tempat penting dan tempat alam seperti sungai. Nama jalan juga ditampilkan.

2. Hybrid

Foto gambaran satelit peta jalanan. Hanya jalan dan nama jalan yang ditampilkan.

3. Satellite

Foto gambaran jalan dari data satelit, namun jalan dan nama jalan tidak ditampilkan.

4. Terrain

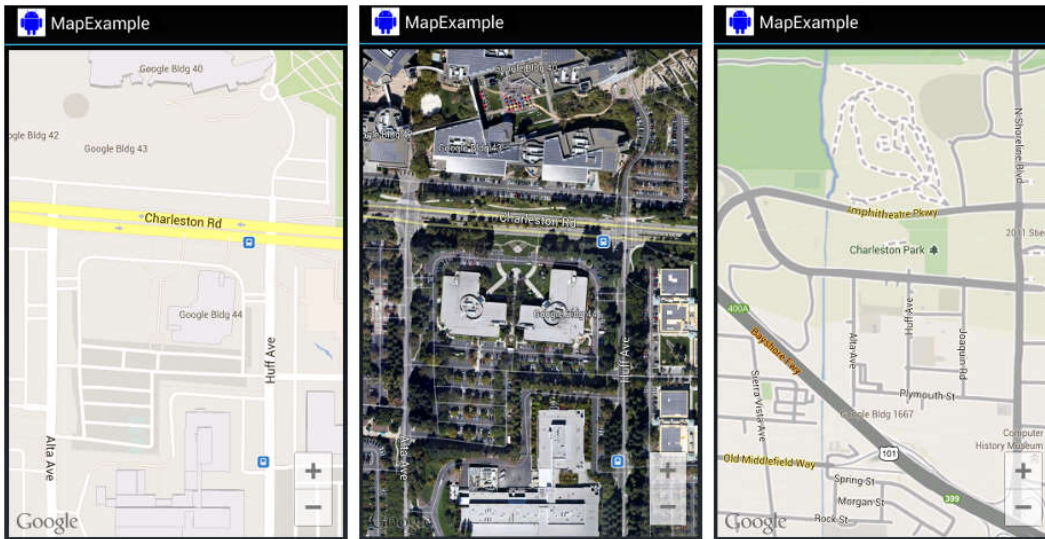
Foto topografi dari jalan, peta terdapat warna-warna, corak garis dan label. Beberapa jalan dan nama jalan ditampilkan.

#### **7.4.5. Cara Mengubah Map Type**

Untuk mengubah tipe peta pada Google Maps API Android adalah dengan memanggil objek GoogleMap dengan method setMapType dan kemudian mengisi salah satu tipe konstanta dari tipe peta. Sebagai contoh, untuk menampilkan peta tipe hybrid adalah dengan cara sebagai berikut :

```
// deklarasi objek GoogleMap
GoogleMapmap;
...
// men-Set tipe peta menjadi "hybrid"
map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
```

Setelah itu jalankan aplikasi maka akan tampak perbeaan tipe peta tersebut, seperti pada gambar dibawah ini



Gambar 7.2 Perbedaan Tipe Peta Google Maps

#### 7.4.6. Marker Manager

Google Maps API sebetulnya sudah menyediakan komponen Marker Manager yang berfungsi untuk mengoptimalkan penampilan marker ketika kita menavigasikan peta namun ada banyak juga Marker Manager yang dibuat oleh orang lain yang lebih bagus. Dengan menggunakan komponen ini objek marker pada peta di Google Maps dapat diatur secara otomatis mana yang perlu ditampilkan dan mana yang tidak secara otomatis. Namun untuk optimalisasi proses penginputan data dari database masih harus dibuat manual, karena jika kita memasukkan data berjumlah besar seperti 500 marker sekaligus akan membuat page load terlalu lama, namun navigasi peta tetap lancar karena telah diatur oleh Marker Manager. Marker adalah simbol yang menandakan suatu lokasi wisata pada peta yang ditampilkan Google Maps. Google Maps API juga mendukung membuat tanda menggunakan Google Map Marker. Tanda Google Map Marker diciptakan melalui kontribusi dari seluruh dunia, dan area yang belum tercakup oleh Google Maps.

#### 7.5 Volley Library

Volley library merupakan suatu library yang dikhususkan untuk para developer android dengan tujuan mempermudah serta mempercepat proses dan skematik dalam melakukan getting dan parsing data pada aplikasi android yang tengah didevelop. Volley sendiri dikembangkan langsung oleh official developer android yaitu Google.

## **7.6 DirectionFinder**

Pada folder Modules terdapat beberapa file java berupa :

### **1. DirectionFinder**

Berfungsi untuk menemukan arah dari tempat tujuan. pengukuran arah dari mana sinyal yang diterima ditransmisikan. Hal ini dapat merujuk ke radio atau bentuk lain dari komunikasi nirkabel, termasuk radar deteksi sinyal dan pemantauan (ELINT / ESM). Dengan menggabungkan informasi arah dari dua atau lebih sesuai spasi penerima (atau penerima seluler tunggal), sumber penularan mungkin berada melalui triangulasi

### **2. DirectionFinderListener dan Route**

Sama halnya seperti Direction Finder, kelebihanannya DirectionFinderListener bisa mendapatkan apa yang diminta dengan perintah suara. Sedangkan Route adalah beberapa arah yang bisa dilalui dengan cepat tanpa terkandala pada satu masalah kedepannya.

### **3. Distance dan Duration**

Adalah jarak yang ditempuh untuk mendapatkan perhitungan waktu, jarak antara dua lokasi, tempat asal dan tujuan. Informasi yang dikembalikan berdasarkan rute yang disarankan antara titik awal dan titik akhir, sesuai perhitungan Google Maps API, dan terdiri dari baris-baris berisi nilai duration dan distance untuk setiap pasang.

# 8

## ANDROID SOCKET PROGRAMMING







## BAB 8

# ANDROID SOCKET PROGRAMMING

### 8.1. Sejarah Socket

Socket adalah mekanisme komunikasi yang memungkinkan terjadinya pertukaran data antar program atau proses baik dalam satu mesin maupun antar mesin. Gaya pemrograman socket sendiri berawal dari sistem Unix BSD yang terkenal dengan kepeloporannya pada bidang penanganan jaringan, sehingga sering disebut BSD Socket. Socket pertama kali diperkenalkan di sistem Unix BSD versi 4.2 tahun 1983 sebagai kelanjutan dari implementasi protokol TCP/IP yang muncul pertama kali pada sistem Unix BSD 4.1 pada akhir 1981. Hampir setiap variant Unix dan Linux mengadopsi BSD Socket. Pada lingkungan Unix, socket memberikan keleluasaan pemrograman gaya Unix yang terkenal dengan ideologinya, Semua di Unix/Linux adalah file. Komunikasi antar program dapat berlangsung lewat penggunaan deskriptor file standar Unix dengan bantuan socket.

Keunggulan dari penggunaan socket adalah anda dapat melakukan komunikasi antar proses/program melalui jaringan berbasis yang TCP/IP tentunya, bahkan dengan program lain yang berjalan pada platform non-unix seperti Microsoft Windows, sepanjang program tersebut berbicara dalam protokol transfer yang sama. Fasilitas-fasilitas yang disediakan oleh mesin unix seperti rlogin, ssh, ftp, dan lain-lain menggunakan socket sebagai sarana komunikasi mereka. Socket dibentuk dan digunakan dengan cara yang berbeda. Komunikasi socket terutama diciptakan untuk tujuan menjembatani komunikasi antara dua buah program yang dijalankan pada mesin yang berbeda. Jangan khawatir, ini tentu saja berarti dua program pada mesin yang sama dapat juga saling berkomunikasi. Kelebihan lain dari komunikasi socket adalah mampu menangani banyak klien sekaligus (multiple clients).

#### 8.1.1. Jenis Socket

Ada dua golongan socket di Unix yang paling umum dipakai yaitu:

##### 1. Socket Lokal atau AF\_UNIX

Socket Lokal adalah socket yang melakukan komunikasi dengan perantara sebuah file yang biasanya diletakkan pada direktori /tmp atau /usr/tmp ataupun /var/tmp. Socket semacam ini digunakan umumnya terbatas untuk komunikasi antar aplikasi dalam satu mesin.

## **2. Socket Networking atau AF\_INET**

Socket Networking ditujukan untuk komunikasi antar aplikasi antar mesin dalam lingkungan jaringan TCP/IP. Identifikasi socket dilakukan dengan sebuah service identifieryaitu berupa nomor port TCP/IP yang dapat di sambung oleh client.

Socket Networking memiliki beberapa jenis, yang paling umum digunakan yaitu:

### **1. Socket Stream atau SOCK\_STREAM**

Stream adalah socket komunikasi full-duplexberbasis aliran (stream) data. Pada model komunikasi Socket Stream, koneksi dua aplikasi harus dalam kondisi tersambung dengan benar untuk dapat bertukar data. Ini dapat dianalogikan seperti komunikasi telepon. Jika sambungan telepon di salah satu titik putus, maka komunikasi tidak dapat terjadi. Koneksi model seperti ini akan menjamin data dapat dipertukarkan dengan baik, namun memiliki kelemahan dalam hal penggunaan jalur data yang relatif besar dan tidak boleh terputus.

### **2. Socket Datagram atau SOCK\_DGRAM**

Socket Datagram berkomunikasi dengan cara yang berbeda. Socket ini tidak membutuhkan koneksi yang tersambung dengan benar untuk mengirimkan dan menerima data. Model koneksi semacam ini tidak dapat menjamin data dapat dipertukarkan dengan baik, namun memiliki keunggulan dalam hal penggunaan jalur data yang minimal. Socket Datagram dapat dianalogikan dengan komunikasi yang terjadi pada kelas, misalnya pada saat guru melakukan broadcastingmateri pelajaran untuk diterima oleh setiap murid. Tidak ada yang dapat menjamin materi pelajaran dapat diterima oleh semua murid dengan baik, kecuali diterapkan metoda rechecking. Rechecking ini dapat dilakukan baik oleh guru maupun murid. Guru bertanya untuk memastikan jawaban dari murid benar, atau murid bertanya untuk memastikan kebenaran materi yang diterimanya. Socket Datagram pun menggunakan metoda ini untuk menjamin pengiriman data dapat dilakukan dengan baik.

## **8.1.2. Sekilas Tentang Socket, TCP Dan UDP**

### **1. Mengenal Socket**

Pengertian socket adalah interface pada jaringan yang menjadi titik komunikasi antarmesin pada Internet Protocol, dan tentunya tanpa komunikasi ini, tidak akan ada pertukaran data dan informasi jaringan.

Socket terdiri dari elemen-elemen utama sebagai berikut:

1. Protokol.
2. Local IP.
3. Local Port.
4. Remote IP.
5. Remote Port.

Dalam komunikasi antara dua pihak, tentunya harus digunakan kesepakatan aturan dan format yang sama agar komunikasi dapat dimengerti. Seperti halnya dua orang yang menggunakan bahasa yang sama, maka bahasa di sini berfungsi sebagai protokol. Protokol yang digunakan dalam socket dapat menggunakan TCP ataupun UDP. Contoh komunikasi sederhana adalah komunikasi antara komputer A dan komputer B. Baik komputer A maupun komputer B harus memiliki identitas unik, yang direpresentasikan oleh IP masing-masing. Komunikasi yang terjadi melalui port, sehingga baik komputer A maupun komputer B harus memiliki port yang dapat diakses satu sama lain.

### **2. TCP dan UDP**

Pemrograman socket adalah cara untuk menggunakan komponen/API (Application Programming Interface) socket untuk membuat sebuah aplikasi. Aplikasi socket umumnya terdiri dari dua kategori berdasarkan pengiriman datanya, yaitu:

- a. Datagram socket (menggunakan UDP).
- b. Stream socket (menggunakan TCP).

Terdapat perlakuan yang berbeda antara UDP dan TCP, walaupun sama-sama berfungsi sebagai protokol pertukaran data. UDP tidak memerlukan proses koneksi terlebih dahulu untuk dapat mengirimkan data, paket-paket data yang dikirimkan UDP bisa jadi melalui rute yang berbeda-beda, sehingga hasil yang diterima bisa jadi tidak berurutan. Contohnya jika aplikasi socket pengirim mengirimkan berturut-turut pesan 1, pesan 2, dan pesan 3, maka aplikasi socket penerima belum tentu mendapatkan pesan yang berurutan

dimulai dari pesan 1, pesan 2, dan terakhir pesan 3. Bisa saja pesan 2 terlebih dulu diterima, menyusul pesan-pesan yang lain, atau berbagai kemungkinan lainnya.

Bahkan, dapat terjadi pesan yang dikirimkan tidak sampai ke penerima karena kegagalan pengiriman paket data. Tidak demikian halnya dengan stream socket yang menggunakan TCP. Jenis ini mengharuskan terjadinya koneksi terlebih dahulu, kemudian mengirimkan paket-paket data secara berurutan, penerima juga dijamin akan menerima data dengan urutan yang benar, dimulai dari data pertama yang dikirimkan hingga data terakhir. TCP dapat menangani data yang hilang, rusak, terpecah, ataupun terduplikasi.

Dari sekilas perbedaan ini, kita dapat menarik kesimpulan bahwa aplikasi socket yang menggunakan TCP memerlukan pertukaran data dua arah yang valid. Sedangkan, aplikasi socket yang menggunakan UDP lebih memprioritaskan pada pengumpulan data. Karena itu aplikasi socket dengan TCP sering diterapkan untuk aplikasi chat, transfer file, ataupun transaksi-transaksi penting. Sedangkan aplikasi socket dengan UDP cocok diterapkan untuk aplikasi monitoring jaringan, game online, dan aplikasi-aplikasi broadcast.

### **3. Socket Programming**

Socket adalah sebuah cara untuk berkomunikasi dengan program atau node lain menggunakan file deskriptor. Di UNIX (dimana socket diciptakan) kita sering mendengar slogan: “everything is a file”, jadi untuk berkomunikasi dengan program atau node lain semudah kita membaca dan menulis file deskriptor. Antarmuka socket dan file adalah mirip, jikapada file kita membukanya dengan `open()` sedangkan pada socket kita menggunakan `socket()`. Pada file deskriptor yang menjadi tujuan adalah sebuah file, sedangkan pada socket adalah computer atau node lain. Intinya ketika kita telah terhubung dengan `socket()`, maka antarmukanya sama saja dengan sebuah file. Sebuah abstraksi perangkat lunak yang digunakan sebagai suatu “terminal” dari suatu hubungan antara dua mesin atau proses yang saling berinterkoneksi

Penggunaan socket programming memungkinkan adanya komunikasi antara client dan server. Salah satu contoh sederhana penggunaan socket programming adalah pembuatan program untuk chatting. Program tersebut sebenarnya merupakan bentuk aplikasi berupa komunikasi antara client dan server. Ketika seorang user (client) melakukan koneksi ke chat server, program akan membuka koneksi ke port yang diberikan, sehingga server perlu membuka socket pada port tersebut dan “mendengarkan” koneksi yang datang. Socket sendiri merupakan gabungan antara host-address dan port address. Dalam hal ini socket digunakan untuk komunikasi antara client dan server.

Socket merupakan fasilitas IPC (Inter Proses Communication) untuk aplikasi jaringan. Agar suatu socket dapat berkomunikasi dengan socket lainnya, maka socket butuh diberi suatu alamat unik sebagai identifikasi. Alamat socket terdiri atas Alamat IP dan Nomer Port. Contoh alamat socket adalah 192.168.29.30: 3000, dimana nomer 3000 adalah nomer portnya. Alamat IP dapat menggunakan alamat Jaringan Lokal (LAN) maupun alamat internet. Jadi socket dapat digunakan untuk IPC pada LAN maupun Internet.

### **8.1.3. Port dan Winsock**

#### **1. Port**

Salah satu elemen penting yang digunakan dalam aplikasi socket adalah port. Port merupakan sebuah koneksi data virtual yang digunakan aplikasi untuk bertukar data secara langsung. Terdapat banyak port di dalam sebuah sistem komputer dengan fungsinya masingmasing. Sebagai contoh, dalam mengirim e-mail digunakan service SMTP yang umumnya menggunakan port 25. Sementara service POP3 untuk menerima e-mail menggunakan port 110, port 80 digunakan untuk HTTP, port 443 digunakan untuk HTTPS, dan seterusnya.

Nomor-nomor port dikategorikan dalam tiga jenis sebagai berikut:

##### **a. Well-known ports.**

Merupakan port yang telah digunakan secara internal oleh sistem Windows, misalnya port untuk koneksi Internet, service FTP, dan seterusnya. Port yang telah digunakan ini adalah port 0 sampai dengan port 1023.

##### **b. Registered ports.**

Port ini dapat digunakan dalam aplikasi Anda, range-nya adalah port 1024 hingga port 49151, cukup banyak port yang tersedia yang bebas Anda pilih sehingga Anda tidak perlu khawatir kekurangan port untuk aplikasi Anda.

##### **c. Dynamic/Private ports.**

Dari port 49152 sampai dengan port 65535.

#### **2. Winsock**

Untuk pemrograman aplikasi socket berbasis Windows, maka komponen API yang sering digunakan adalah Winsock (Win-dows Socket API) yang mendukung interface standar TCP/IP, yang merupakan protokol jaringan paling populer saat ini (contoh protokol jaringan

yang lain adalah NetBIOS, IPX dari Novell, AppleTalk dari Apple, dan sebagainya). Pengertian TCP/IP (TCP over IP) mungkin dapat menjadi sedikit rancu jika diartikan TCP/IP hanya mengizinkan pengiriman TCP (dan tidak UDP), padahal seperti yang telah kita bahas, pengiriman socket dapat melalui TCP maupun UDP.

Pengertian TCP/IP di sini sebenarnya digunakan untuk menunjukkan teknologi jaringan/Internet, termasuk di dalamnya adalah UDP. Jika Anda menggunakan UDP, dapat juga disebut sebagai UDP/IP (UDP over IP), tetapi umumnya istilah ini jarang digunakan dan istilah TCP/IP telah mencakup, baik TCP maupun UDP.

Pada bahasa pemrograman visual seperti Visual Basic/Delphi, Anda dapat menggunakan control Winsock yang telah disediakan untuk mengembangkan aplikasi socket. Walaupun kita akan mencontohkan aplikasi socket dalam environment Windows, Anda tidak perlu khawatir jika aplikasi socket yang menggunakan Winsock tidak dapat berkomunikasi dengan aplikasi socket berbasis Unix/Linux, karena komunikasi tetap dapat terjadi selama aplikasi tersebut menggunakan protokol jaringan yang sama.

Bagi Anda yang terpaksa hanya menggunakan satu komputer, dapat memanfaatkan alamat local host atau 127.0.0.1 yang mengizinkan dua aplikasi berjalan pada satu mesin komputer dan berkomunikasi satu sama lain.

## 8.2. Model Aplikasi Client Server



Gambar 8.1 Model Client Server

Untuk membuat aplikasi socket yang sederhana diperlukan dua aplikasi. Yaitu, pertama adalah aplikasi server yang akan menerima data, sedangkan aplikasi kedua adalah aplikasi client yang mengirimkan data pada server.

Baik aplikasi server dan aplikasi client mendefinisikan port yang sama sebagai jalur komunikasi. Objek socket pada sisi client dan server berbeda sedikit. Pada sisi aplikasi server, suatu socket server dibentuk dan melakukan operasi listen/menunggu. Operasi ini

pada intinya menunggu permintaan koneksi dari sisi client. Sedangkan pada sisi client, dibentuk suatu socket biasa.

Pada saat socket client, informasi alamat socket server dilewatkan sebagai argumen dan socket client akan otomatis mencoba meminta koneksi ke socket server. Pada saat permintaan koneksi client sampai pada server, maka server akan membuat suatu socket biasa. Socket ini yang nantinya akan berkomunikasi dengan socket pada sisi client. Setelah itu socket server dapat kembali melakukan listen untuk menunggu permintaan koneksi dari client lainnya. Langkah ini umumnya hanya dilakukan jika aplikasi server mengimplementasikan multithreading.

Setelah tercipta koneksi antara client dan server, maka keduanya dapat saling bertukar pesan. Salah satu atau keduanya kemudian dapat mengakhiri komunikasi dengan menutup socket. Untuk protokol UDP, perbedaannya adalah socket di sisi server sama dengan socket di sisi client, dan tidak ada operasi listen pada sisi server. Kemudian saat paket data dikirimkan, alamat socket penerima harus disertakan sebagai argumen.

### **8.3. Kelas Tambahan**

Kelas `java.net.Socket` Kelas ini digunakan untuk melakukan koneksi ke suatu mesin atau proses yang dituju. Kelas `java.net.ServerSocket` Kelas ini berfungsi untuk listen koneksi yang sudah terjadi dan siap menerima permintaan dari proses lain. Kelas ini digunakan ketika pembuatan aplikasi server yang bersifat Connection Oriented. Ketika dibuat obyek baru `ServerSocket`, untuk dapat menerima dan mengirim melalui stream perlu dibuat `InputStream` dan `OutputStream`. `ServerSocket` siap menerima permintaan client ketika menjalankan method `accept()` dan akan mengakhiri koneksi ketika menjalankan method `close()`

Kelas `java.net.ServerSocket` digunakan oleh Server untuk listen koneksi, sedangkan `java.net.Socket` digunakan oleh Client untuk inisialisasi koneksi.

Setelah client terkoneksi ke server dengan menggunakan Socket, maka `ServerSocket` akan mengembalikan status server ke client melalui koneksi yang terbentuk sebelumnya.

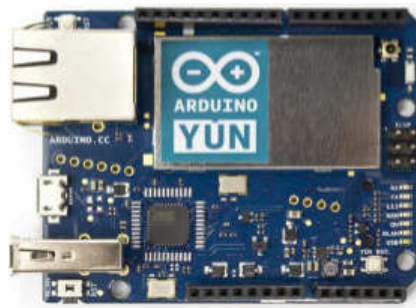
### **8.4. Menggunakan Socket**

Komunikasi menggunakan socket biasa disebut sebagai connection oriented, yaitu komunikasi dimana saluran akan dipertahankan terkoneksi dari awal hingga akhir pengiriman data. Karena komunikasi ini melibatkan server dan klien.

Untuk membangun socket server diperlukan 5 langkah berikut:

1. Menciptakan Objek ServerSocket Konstruktor dari ServerSocket memerlukan port number (1024-65535) sebagai argument, Misalnya : `ServerSocket soket = new ServerSocket(1234);`  
Pada contoh diatas berarti server akan menunggu permintaan dari klien melalui port 1234.
2. Server diperintahkan untuk menunggu Server akan menunggu permintaan “tanpa batas waktu” melalui kelas ServerSocket. `Socket link = soket.accept();`
3. Siapkan sarana input dan output stream Metorde `getInputStream` dan `getOutputStream` dari kelas Socket digunakan untuk menyediakan sarana input dan output. Pada aplikasi non-GUI dapat digunakan objek `Scanner` untuk proses input dan objek `PrintWriter` untuk proses output.  
`Scanner input = new Scanner(link.getInputStream());` Dan `PrintWriter output = new PrintWriter(link.getOutputStream(),true);`
4. Mengirim dan menerima data Setelah sarana input dan output tersedia maka membaca input dari klien dan mengirim output ke klien dapat dilakakukan dengan mudah.  
`output.println(“Menunggu data..”);` `String input = input.nextLine();`
5. Menutup koneksi Memakai fungsi `close` dari kelas socket `link.close();`

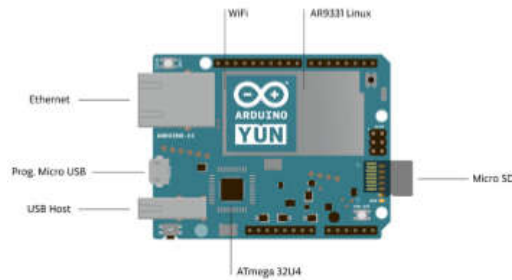
## 8.5. Arduino Yun



Gambar 8.2 Arduino Yun

Arduino Yun adalah mikrokontroler berbasis ATmega32u4 dan Atheros AR9331. Prosesor Atheros mendukung distribusi Linux berbasis OpenWRT bernama OpenWRT-Yun. Arduino Yun memiliki beberapa fitur seperti sambungan Ethernet, dukungan WiFi, Port USB, slot kartu micro-SD, 20 digital input / output pin ( dimana 7 dapat digunakan sebagai output PWM dan 12 sebagai input analog ), osilator kristal 16 MHz , koneksi micro USB, header ICSP , dan 3 tombol reset.

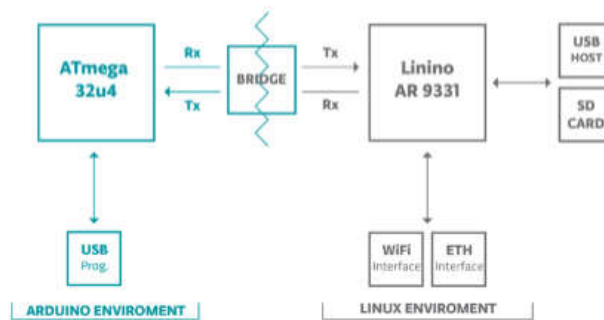




Gambar 8.3 Fitur-fitur Arduino Yun

Arduino Yun membedakan dirinya dari papan Arduino lain dalam beberapa hal, seperti dapat berkomunikasi dengan Linux dan menawarkan jaringan komputer yang kuat dengan kemudahan Arduino. Selain perintah linux seperti Curl, pada arduino yun dapat ditulis shell dan python.

Library Bridge pada contoh sketch arduino memfasilitasi komunikasi antara dua prosesor, memberikan kemampuan pada arduino untuk menjalankan skrip shell, berkomunikasi dengan antarmuka jaringan, dan menerima informasi dari prosesor AR9331. USB host, interface jaringan dan kartu SD tidak terhubung ke prosesor 32U4, tapi terhubung ke prosesor AR9331. Dengan adanya library Bridge, memungkinkan Arduino untuk berinteraksi dengan peripheral-periferal tersebut.



Gambar 8.4 Komunikasi Dua Prosesor Pada Arduino Yun

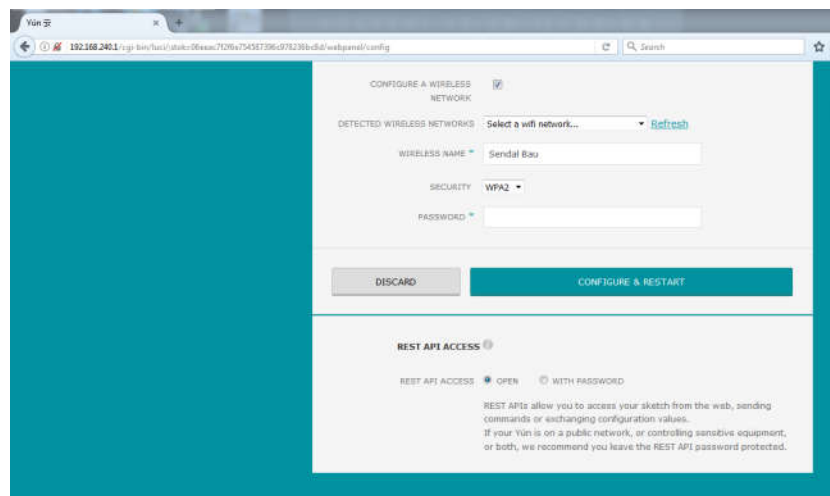
#### A. Listing program Bridge pada Arduino IDE

Pada listing program ini menunjukkan bagaimana cara kerja dari library Bridge untuk mengakses pin digital dan analog di board Arduino Yun melalui panggilan REST API. Ini menunjukkan bagaimana program dapat membuat api sendiri ketika dipanggil melalui browser.

API (*Application Programming Interface*) adalah sekumpulan perintah, fungsi, dan protocol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk

sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi lain.

Saat menjalankan, perintah ini pastikan komputer di jaringan yang sama dengan Arduino YUN. Bila di upload pada *board*, anda dapat meminta nilai pada pin, menulis nilai untuk pin, dan mengkonfigurasi pin sebagai input atau output. Ketika password REST pada arduino yun dimatikan, dengan menggunakan browser dengan struktur URL Berikut.



Gambar 8.5 mematikan password REST API Yun.

Untuk mengakses fungsi rest api pada yun dapat menggunakan ip address atau menggunakan nama dari arduino yun tersebut contohnya sebagai berikut:

```
http://192.168.240.1/arduino/digital/13: call digitalRead(13);  
http://192.168.240.1/arduino/digital/13/1: call digitalWrite(13,1);  
http://myArduinoYun.local/arduino/analog/13/125: call analogWrite(13,125);  
http://myArduinoYun.local/arduino/analog/13: call analogRead(13);  
http://myArduinoYun.local/arduino/mode/13/input: pinMode(13,INPUT);  
http://myArduinoYun.local/arduino/mode/13/output: pinMode(13,OUTPUT);  
#include <Bridge.h>  
#include <BridgeServer.h>  
#include <BridgeClient.h>
```

Dari listing program diatas menjelaskan tentang fungsi program menyertakan library untuk masing-masing Bridge, BridgeServer, BridgeClient. Dan untuk difungsikan juga permintaan dari REST. REST (Representational State Transfer) adalah salah satu jenis web service yang menerapkan konsep perpindahan antar state. State disini dapat digambarkan seperti jika browser meminta suatu halaman web, maka server akan mengirimkan state

halaman web yang sekarang ke browser. Bernavigasi melalui link-link yang disediakan sama halnya dengan mengganti state dari halaman web, begitu pula REST bekerja dengan bernavigasi melalui link-link.