

# Membuat Aplikasi Android Untuk Pemula

Disusun oleh:  
Lhuqita Fazry





# Daftar Isi

<b>DAFTAR ISI.....</b>	<b>III</b>
<b>1. MENGENAL ANDROID STUDIO .....</b>	<b>1</b>
1.1. INSTALASI JAVA DEVELOPMENT KIT (JDK) .....	1
1.2. INSTALASI ANDROID STUDIO .....	2
1.2. JENDELA KERJA ANDROID STUDIO .....	2
1.3. ANDROID SDK MANAGER.....	3
1.4. ANDROID VIRTUAL DEVICE MANAGER .....	4
1.5. GENYMOTION .....	7
1.6. ANDROIDMANIFEST.XML.....	11
1.7. JAVA DAN XML.....	12
<b>2. SEKILAS TENTANG JAVA.....</b>	<b>14</b>
2.1. APA ITU JAVA?.....	14
2.2. NAMA VARIABEL .....	14
2.3. OBJEK DAN KELAS .....	14
2.4. TIPE DATA PADA JAVA .....	15
2.5. PEWARISAN ( <i>INHERITANCE</i> ) .....	16
<b>3. MEMBUAT DAN MENJALANKAN PROJEK PERTAMA.....</b>	<b>17</b>
3.1. MEMBUAT PROJEK BARU .....	17
3.2. STRUKTUR PROJEK ANDROID STUDIO .....	21
3.3. MENJALANKAN PROJEK PADA ANDROID VIRTUAL DEVICE .....	22
3.4. MENJALANKAN PROJEK PADA GENYMOTION .....	23
3.5. MENJALANKAN PROJEK PADA PERANGKAT ANDROID .....	23
<b>4. MENGENAL ACTIVITY .....</b>	<b>24</b>
4.1. APA ITU ACTIVITY? .....	24
4.2. ACTIVITY LIFECYCLE .....	28
4.3. APLIKASI MULTISCREEN .....	29
<b>5. RESOURCE .....</b>	<b>30</b>
5.1. JENIS RESOURCE .....	30
5.2. RESOURCE LAYOUT .....	31
5.3. RESOURCE TEKS .....	32
5.4. RESOURCE ARRAY TEKS .....	33
5.5. RESOURCE DRAWABLE.....	34
5.6. RESOURCE MENU .....	35
<b>6. VIEW.....</b>	<b>38</b>
6.1. MENGENAL VIEW DAN VIEWGROUP.....	38
6.2. TEXTVIEW .....	40
6.3. EDITTEXT .....	42
6.4. BUTTON .....	43
6.5. IMAGEVIEW.....	45
6.6. IMAGEBUTTON .....	46

6.7. SPINNER .....	47
6.8. CHECKBOX.....	49
6.9. AUTOCOMPLETETEXTVIEW .....	50
6.10. TOGGLEBUTTON .....	52
6.11. RADIOBUTTON .....	53
6.12. RADIOGROUP.....	55
6.13. PROGRESSBAR.....	57
<b>7. LAYOUT.....</b>	<b>59</b>
7.1. ATRIBUT LAYOUT .....	59
7.2. FRAME LAYOUT .....	62
7.3. LINEARLAYOUT.....	63
7.4. RELATIVELAYOUT.....	65
<b>8. DIALOG .....</b>	<b>69</b>
8.1. TOAST.....	69
8.2. SNACK BAR .....	69
8.3. ALERT DIALOG .....	69
8.4. PROGRESS DIALOG .....	69
8.5. ACTIVITY SEBAGAI DIALOG.....	69
<b>9. MENU.....</b>	<b>70</b>
9.1. MENU SEBAGAI ACTION .....	70
9.2. POPUP MENU .....	70
9.3. CONTEXT MENU .....	70
<b>10. SHARED PREFERENCES .....</b>	<b>71</b>
10.1. MENYIMPAN DATA.....	71
10.2. MENGAMBIL DATA.....	71
10.3. MEMBUAT KELAS HELPER.....	71
<b>11. SQLITE .....</b>	<b>72</b>
11.1. SQLITEHELPER .....	72
11.2. MELAKUKAN QUERY DATA .....	72
11.3. MENAMBAH DATA.....	72
11.4. MENGUPDATE DATA.....	72
11.5. MENGHAPUS DATA.....	72
<b>12. LIST VIEW.....</b>	<b>73</b>
12.1. APA ITU LIST VIEW? .....	73
12.2. DEFAULT ADAPTER .....	73

# 1. Mengenal Android Studio

Android Studio merupakan software pengembang yang digunakan untuk membuat sebuah aplikasi android. Pada buku ini, versi yang akan digunakan adalah Android Studio 1.5.1. Android Studio dapat digunakan pada berbagai sistem operasi komputer termasuk Windows, Linux dan Mac. Untuk menginstall Android Studio pada komputer, maka kita harus menginstall Java Development Kit (JDK) terlebih dahulu.

## 1.1. Instalasi Java Development Kit (JDK)

Pada buku ini, versi JDK yang akan kita gunakan adalah JDK 7. JDK7 tersedia gratis pada website *Oracle*. Anda dapat mengunduh file instalasi JDK7 melalui tautan berikut:

[www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html](http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html)

Pada halaman unduh JDK7, anda dapat melihat beberapa pilihan file untuk anda download. Pilihlah salah satu file yang sesuai dengan sistem operasi yang anda gunakan. Sebelum itu, jangan lupa untuk menyetujui lisensi dengan menekan button *Accept License Agreement*.

Java SE Development Kit 7u79		
You must accept the <a href="#">Oracle Binary Code License Agreement for Java SE</a> to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux x86	130.4 MB	<a href="#">jdk-7u79-linux-i586.rpm</a>
Linux x86	147.6 MB	<a href="#">jdk-7u79-linux-i586.tar.gz</a>
Linux x64	131.69 MB	<a href="#">jdk-7u79-linux-x64.rpm</a>
Linux x64	146.4 MB	<a href="#">jdk-7u79-linux-x64.tar.gz</a>
Mac OS X x64	196.89 MB	<a href="#">jdk-7u79-macosx-x64.dmg</a>
Solaris x86 (SVR4 package)	140.79 MB	<a href="#">jdk-7u79-solaris-i586.tar.Z</a>
Solaris x86	96.66 MB	<a href="#">jdk-7u79-solaris-i586.tar.gz</a>
Solaris x64 (SVR4 package)	24.67 MB	<a href="#">jdk-7u79-solaris-x64.tar.Z</a>
Solaris x64	16.38 MB	<a href="#">jdk-7u79-solaris-x64.tar.gz</a>
Solaris SPARC (SVR4 package)	140 MB	<a href="#">jdk-7u79-solaris-sparc.tar.Z</a>
Solaris SPARC	99.4 MB	<a href="#">jdk-7u79-solaris-sparc.tar.gz</a>
Solaris SPARC 64-bit (SVR4 package)	24 MB	<a href="#">jdk-7u79-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	18.4 MB	<a href="#">jdk-7u79-solaris-sparcv9.tar.gz</a>
Windows x86	138.31 MB	<a href="#">jdk-7u79-windows-i586.exe</a>
Windows x64	140.06 MB	<a href="#">jdk-7u79-windows-x64.exe</a>

Setelah proses unduhan selesai, kemudian lanjutkan dengan proses instalasi sesuai dengan sistem operasi pada komputer yang anda miliki. Berikut adalah tutorial proses instalasi JDK7 pada komputer dengan sistem operasi Mac OS.

## 1.2. Instalasi Android Studio

Android Studio tersedia secara gratis pada website resmi android. Anda dapat mengunjungi tautan berikut ini untuk mengunduh file instalasi Android Studio.

**[developer.android.com/sdk/index.html](https://developer.android.com/sdk/index.html)**

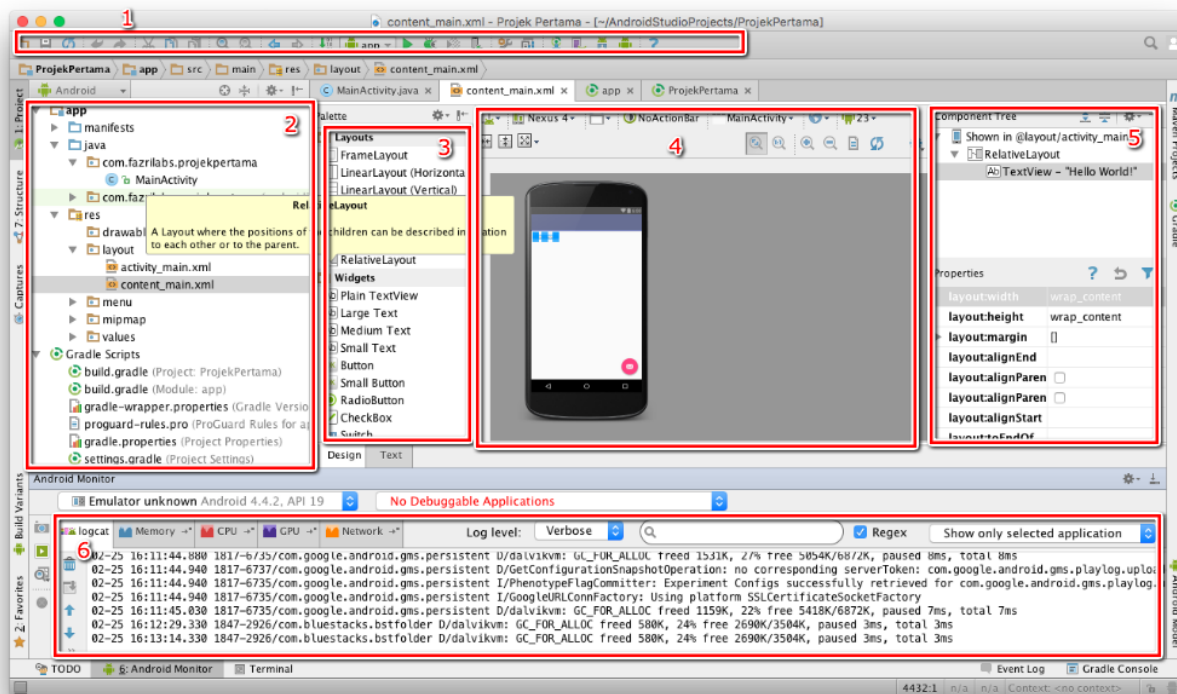
Berikut adalah tampilan dari halaman unduh Android Studio. Untuk mengunduh, klik tombol *Download Android Studio* yang berwarna hijau.



Setelah file instalasi berhasil diunduh dengan baik. Maka lanjutkan dengan proses instalasi dengan melakukan klik ganda pada file instalasi tersebut.

## 1.2. Jendela Kerja Android Studio

Pada sub bab ini, kita akan membahas mengenai beberapa fungsi dasar dari Android Studio yang akan membantu kita dalam memahami bagaimana cara menggunakan software Android Studio. Tampilan standar dari Android Studio tertera seperti gambar di bawah ini.



szoter.com

Gambar di atas merupakan tampilan standar dari Android Studio. Berikut adalah keterangan dari gambar di atas.


1. *Toolbar*: Merupakan lokasi beberapa tombol cepat yang paling sering digunakan oleh pengembang.
2. *Project*: Merupakan tampilan folder dari proyek sehingga pengembang dapat melakukan navigasi ke berbagai file proyek dengan cepat.
3. *Palette*: Merupakan lokasi widget UI, anda dapat langsung men-drag dan drop widget ke windows preview yang berada di sebelah kanan.
4. *Preview*: Preview dari tampilan aplikasi.
5. *Component Tree & Properties*: Merupakan editor dari sebuah widget.
6. *Monitor*: Untuk melihat error dan warning.

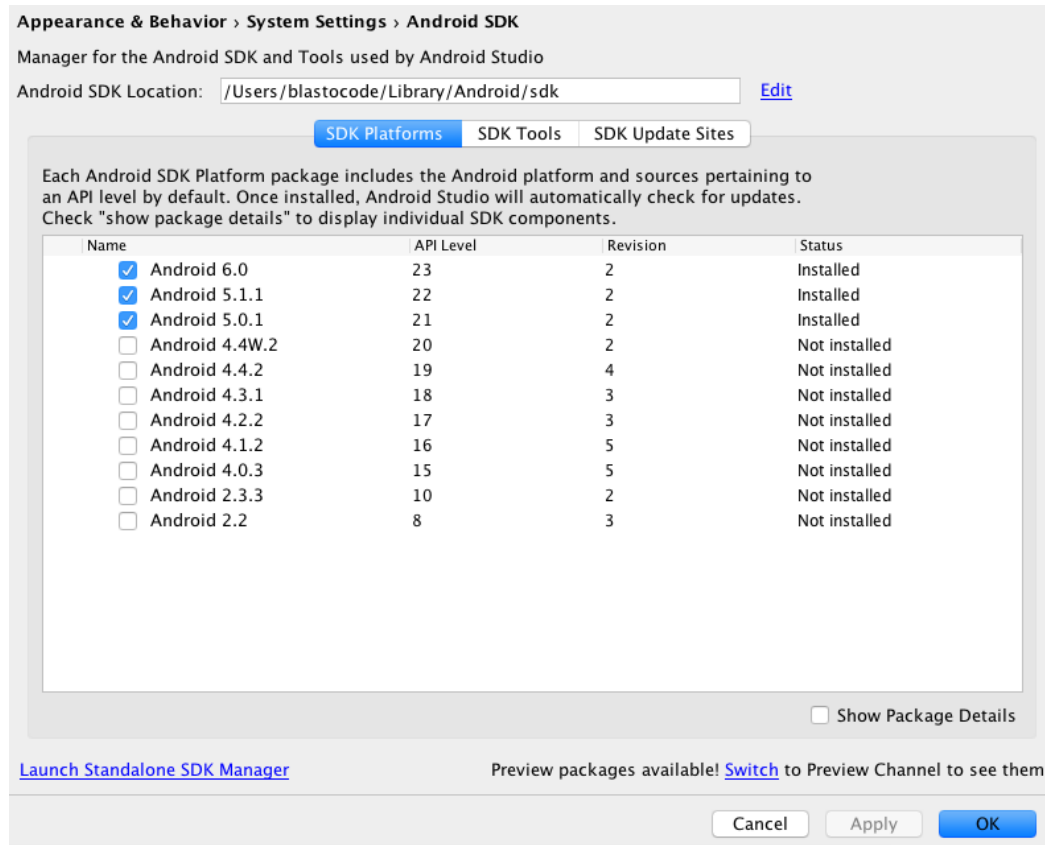
### 1.3. Android SDK Manager

Sejak pertama kali dirilis hingga saat ini, Google telah merilis beberapa versi android mulai dari *froyo*, *gingerbread*, hingga yang terbaru adalah *marshmallow*. Versi terbaru dibuat untuk melengkapi dan menyempurnakan versi sebelumnya. Oleh karena itu, pada versi-versi terbaru terdapat penambahan fitur atau mungkin saja penggantian fitur dengan yang lebih canggih. Hal tersebut berdampak pula dari sisi pembuatan aplikasi.

Seiring dengan rilisnya versi baru dari android, maka Google pun merilis Android SDK untuk android versi tersebut. Oleh karena itu, akan kita temukan variasi SDK mulai

dari Android 2.2 hingga Android 6.0. Untuk dapat menggunakan SDK tersebut kita harus mengunduhnya terlebih dahulu. Android SDK Manager merupakan *tool* yang dapat memudahkan kita dalam mengunduh Android SDK serta beberapa *library* pendukung.

Android SDK Manager dapat dijalankan melalui Android Studio dengan menekan tombol  pada *toolbar*. Berikut adalah tampilan dari Android SDK Manager.




Pada Android SDK Manager kita dapat mencentang versi SDK yang ingin kita unduh kemudian menekan tombol *OK*. Versi SDK yang kita pilih akan segera diunduh. Pastikan komputer anda memiliki koneksi internet. Setelah proses unduhan selesai, maka kita sudah bisa menggunakan versi SDK tersebut pada proyek.

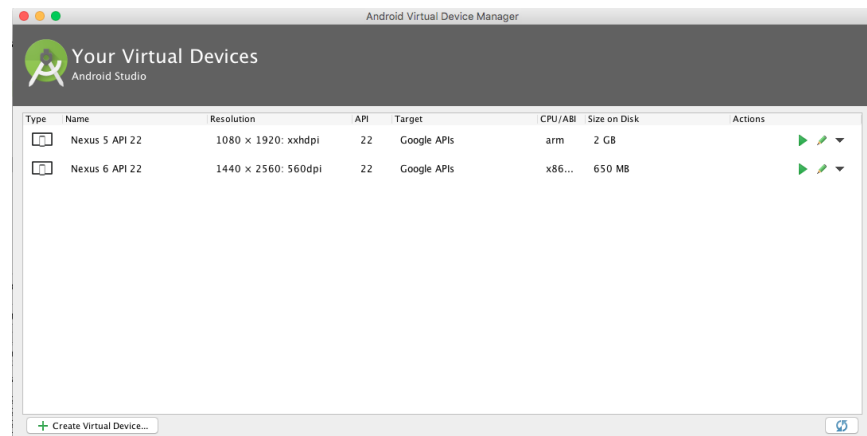
#### 1.4. Android Virtual Device Manager

Untuk melakukan pengetesan aplikasi, kita dapat menggunakan perangkat virtual yang telah tersedia pada Android Studio. Dengan perangkat virtual, kita dapat dengan mudah melakukan tes aplikasi tanpa harus menggunakan ponsel sesungguhnya.

Sebelum kita dapat menjalankan aplikasi pada perangkat virtual, kita harus membuatnya terlebih dahulu. Berikut adalah langkah-langkah untuk membuat sebuah perangkat virtual.

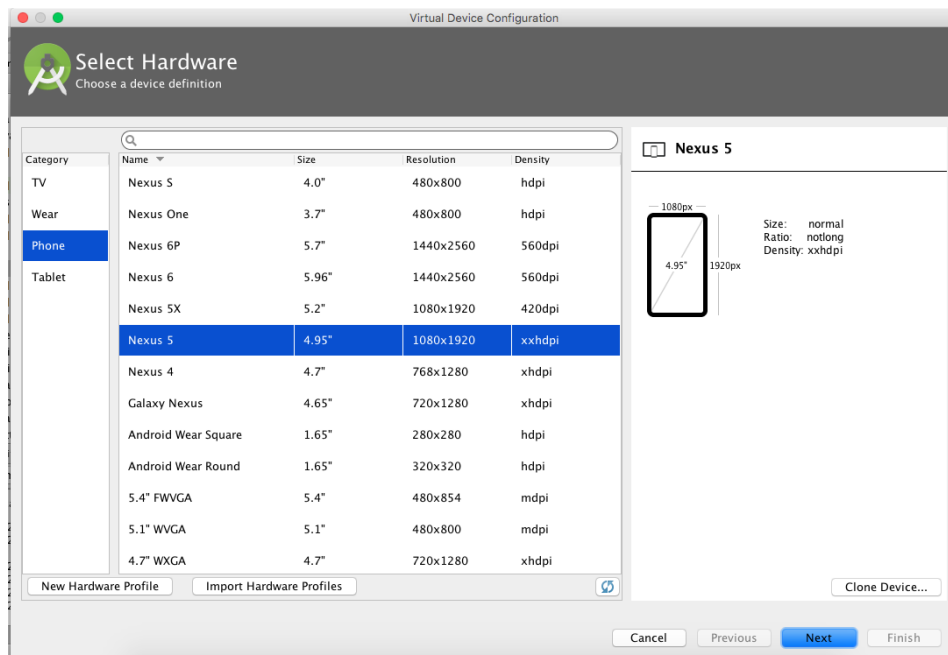


1. Klik tombol  pada *toolbar* maka akan muncul jendela *Android Virtual Device Manager*.

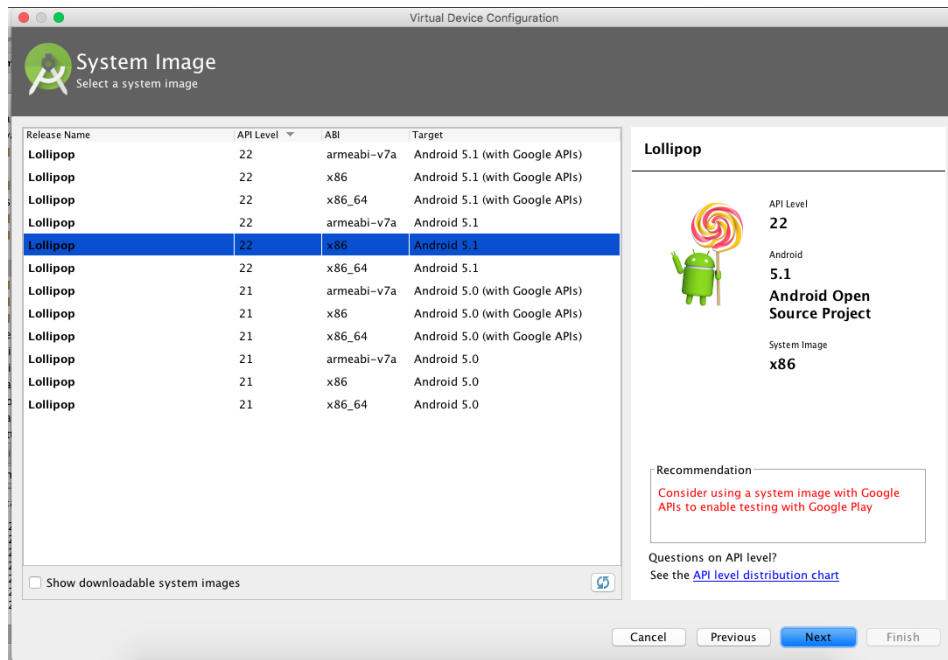


Semua perangkat virtual yang telah kita buat akan muncul pada jendela tersebut.

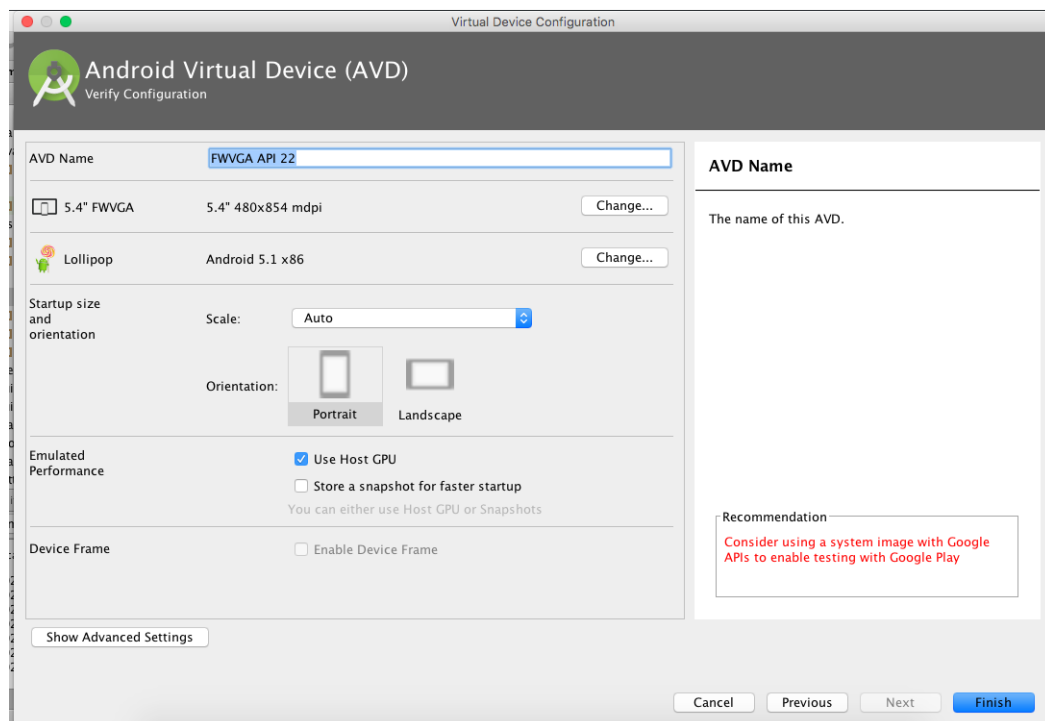
2. Kemudian klik tombol *Create Virtual Device* maka akan muncul jendela *Virtual Device Configuration*. Pada jendela ini, kita dapat mengatur konfigurasi dari perangkat virtual yang akan kita buat.




3. Pilih kategori dan jenis perangkat virtual kemudian tekan tombol *Next* maka akan muncul jendela pemilihan versi android untuk perangkat virtual yang akan kita gunakan.



- Pilih versi android yang akan anda pakai kemudian klik tombol *Next* maka akan muncul jendela verifikasi konfigurasi. Anda dapat melakukan konfigurasi tambahan pada jendela tersebut. Kemudian klik tombol *Finish* maka proses pembuatan perangkat virtual telah selesai. Perangkat virtual yang baru saja kita buat sudah bisa kita pergunakan untuk pengetesan aplikasi.



- Untuk memulai mengetes aplikasi, anda perlu menjalankan perangkat virtual yang baru saja anda buat dengan meng-klik tombol  pada jendela *Android Virtual Device Manager*. Silahkan tunggu beberapa saat, perangkat virtual akan segera aktif.



### 1.5. GenyMotion

Walaupun kita bisa menggunakan perangkat virtual pada Android Studio dengan mudah, akan tetapi perangkat virtual tersebut sangat lambat dibandingkan dengan ponsel sungguhan. Kelambatan perangkat virtual tersebut terjadi baik pada saat perangkat dinyalakan(*booting*) maupun ketika dioperasikan. Kurangnya performa perangkat virtual ini membuat proses tes aplikasi menjadi terhambat.


Sebagai alternatif, kita bisa menggunakan GenyMotion sebagai ganti perangkat virtual bawaan Android Studio. GenyMotion menawarkan solusi yang lebih baik dari perangkat virtual yang sudah ada di Android Studio. GenyMotion bisa digunakan secara gratis untuk keperluan pribadi sedangkan untuk keperluan komersial anda bisa menggunakannya dengan harga sekitar \$136 per tahun.

Untuk mendownload GenyMotion, silahkan mengunjungi alamat website berikut:

**[www.genymotion.com/download](http://www.genymotion.com/download)**

Anda diharuskan untuk membuat akun sebelum dapat mengunduh GenyMotion. Untuk membuat akun baru di GenyMotion, silahkan klik tombol *Create an account*.

## Sign in



Username
  
  
  
Password
  
  
[Forgotten your password?](#)

☒ Remember me on this computer

Silahkan isi data-data yang diperlukan pada formulir *Sign up*. Isilah *username*, *email address* dan *password* dengan data yang benar. Untuk field *company size* bisa diisi dengan *personal use* sedangkan field *usage type* bisa diisi *development*.

## Sign up

Username

Company size

Email Address

Usage type

Password

☒ Get our latest news & updates
 ☐ I accept terms of the [privacy statement](#)

Setelah semua *field* diisi dengan baik, jangan lupa mencentang kotak *I accept terms of the privacy statement* kemudian dilanjutkan dengan meng-klik tombol *Create an account*. Maka akan muncul halaman konfirmasi berikut:

## Account creation

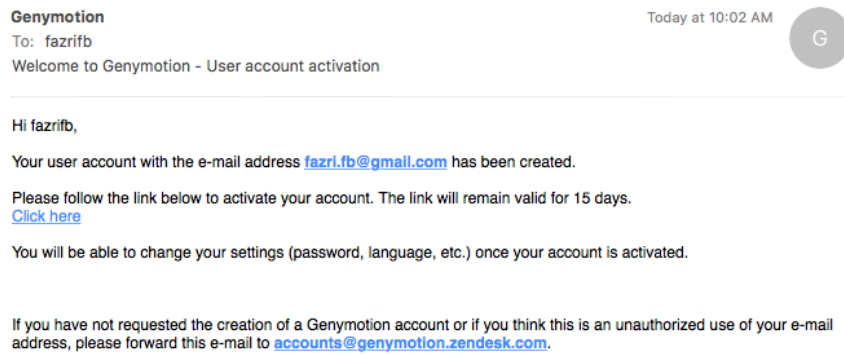
Thank you for registering on the Genymotion website. You will receive a confirmation e-mail containing an activation link at this address:

fazri.fb@gmail.com

If you have not received the confirmation e-mail, please check your spam folder and verify that your e-mail address is correct. If you try to sign in, you will be able to request the confirmation e-mail again.

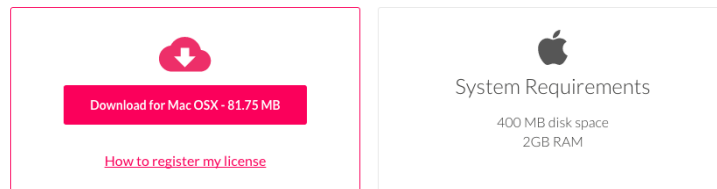
Pada tahap ini, pembuatan akun GenyMotion telah berhasil. GenyMotion telah mengirimkan sebuah email yang berisi tautan untuk mengaktifkan akun yang baru saja

anda buat. Silahkan cek email anda, email yang anda terima kira-kira seperti gambar di bawah.



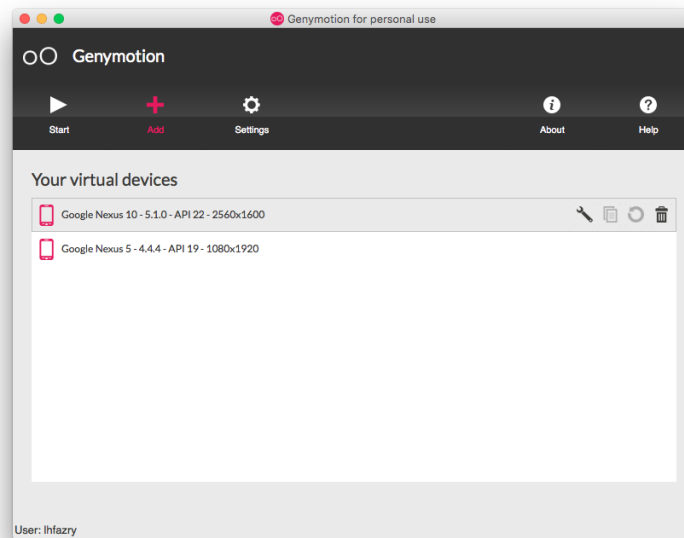
Klik pada tautan *Click here*. Sampai disini, akun anda telah berhasil diaktivasi. Sekarang anda bisa mengunjungi kembali halaman untuk mengunduh GenyMotion pada tautan [www.genymotion.com/download](http://www.genymotion.com/download). Kemudian pada halaman unduhan tersebut, klik tombol *Download for Mac OSX*. Label dari tombol tersebut tergantung sistem operasi yang anda gunakan.

### Download Genymotion

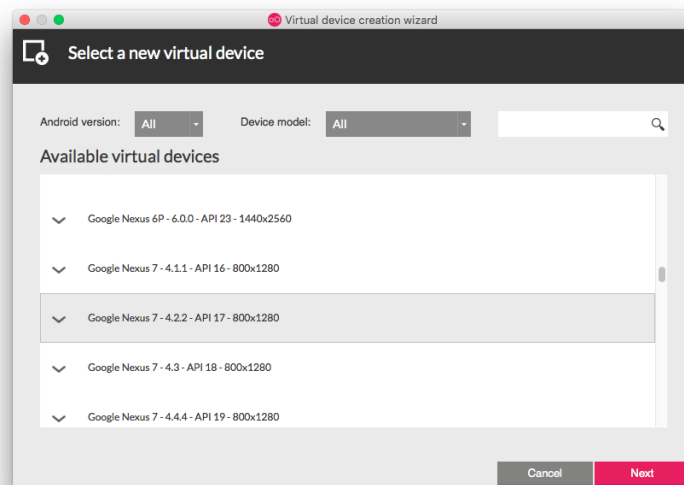


Setelah proses unduhan selesai, silahkan lanjutkan dengan proses instalasi.

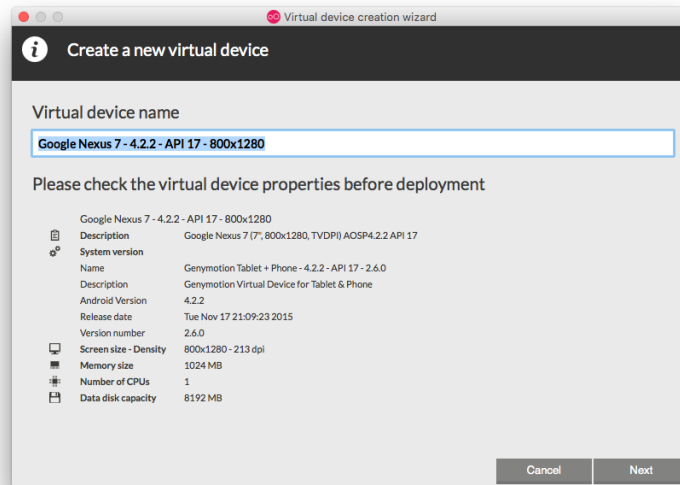
Setelah proses instalasi selesai kemudian jalankan GenyMotion maka akan muncul jendela utama GenyMotion seperti tampak pada gambar di bawah.



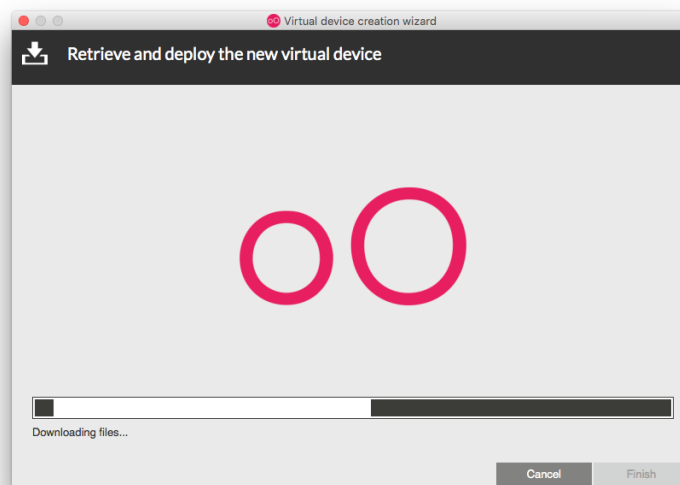
Sama halnya dengan *Android Virtual Device*, di GenyMotion anda juga harus menambahkan perangkat virtual baru sebelum bisa melakukan tes. Silahkan klik tombol *Add* untuk menambah perangkat virtual baru pada GenyMotion. Maka akan muncul jendela *Virtual device creation wizard*.



Silahkan pilih jenis perangkat yang ingin anda buat lalu klik tombol *Next*. Kemudian muncul jendela selanjutnya, lanjutkan saja dengan meng-klik tombol *Next*.



GenyMotion akan mengunduh beberapa file yang dibutuhkan setelah itu proses pembuatan perangkat virtual pada GenyMotion telah selesai.



Untuk menjalankan perangkat virtual, silahkan klik ganda pada perangkat virtual tersebut pada jendela utama GenyMotion.

## 1.6. AndroidManifest.xml

AndroidManifest.xml merupakan sebuah file xml yang menyimpan informasi-informasi penting mengenai aplikasi android. Setiap aplikasi android harus memiliki file tersebut. Beberapa informasi yang tersimpan di dalam file tersebut adalah sebagai berikut:

1. Identitas: Informasi ini berupa nama package dari sebuah aplikasi dan harus unik di dalam Google Play Store, misal: com.fazrilab.adzkar.

2. Deklarasi *permission*. Setiap aplikasi yang akan mengakses fitur sensitif pada smartphone maka perlu mendaftarkan permission di file AndroidManifest.xml. Jika tidak, maka android tidak akan mengizinkan penggunaan fitur tersebut. Misal akses ke kamera, gps, sms, koneksi internet, dan lain-lain.
3. Deklarasi activity. Semua activity yang dibuat, harus dicantumkan disini.

Berikut adalah contoh sebuah file AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.fazrilabs.projekpertama">
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

## 1.7. Java Dan XML

Untuk dapat membuat sebuah aplikasi android, anda harus memahami dua bahasa pemrograman sekaligus, yaitu Java dan XML (*Extensible Markup Language*). Walaupun tidak harus mencapai tingkat mahir, akan tetapi pemahaman mendasar mengenai kedua bahasa tersebut sangat dibutuhkan oleh seorang pengembang aplikasi android. Kita akan mempelajari pokok konsep Java pada bab berikutnya. Pada sub bab ini kita akan mencoba membahas sedikit mengenai konsep xml.

Perhatikan contoh kode xml sederhana berikut:

```
<?xml version="1.0" encoding="UTF-8"?>
<surat>
    <kepada>Ahmad</kepada>
    <dari>Zuhair</dari>
    <judul>Peringat</judul>
    <pesan>Jangan lupa janji minggu ini!</pesan>
</surat>
```



Jika kita perhatikan contoh kode xml di atas, kita dapat melihat bahwa xml merupakan sebuah sarana untuk menyimpan informasi. Setiap potongan informasi dimulai dengan tanda pembuka dan tanda penutup. Tanda pembuka dan tanda penutup dapat kita definisikan sendiri sesuai kebutuhan dengan syarat keduanya harus sama.

Contoh di atas menunjukkan informasi mengenai sebuah surat. Di dalam tanda pembuka `<surat>`, kita temukan beberapa potongan informasi yang kesemuanya diawali dengan tanda pembuka yang baru. Kita dapat menyimpulkan bahwa informasi surat tersebut adalah ditujukan kepada *Ahmad* yang dikirim oleh *Zuhair* dengan judul *Pengingat* dan pesannya adalah *Jangan lupa janji minggu ini*. Masing-masing dari potongan informasi tersebut ditutup dengan tanda penutup yang sama dengan tanda pembuka masing-masing potongan informasi.

Android menggunakan xml untuk mendefinisikan berbagai macam *resource* termasuk *layout*, *string*, *menu*, *drawable* dan lain-lain. Pembahasan lebih lanjut mengenai penggunaan xml pada android akan kita bahas pada bab selanjutnya.

## 2. Sekilas Tentang Java

Pada sub sebelumnya, kita telah memperkenalkan tentang xml dan menyinggung sedikit bagaimana xml digunakan untuk membuat aplikasi android. Pada bab ini kita akan membahas mengenai bahasa pemrograman Java yang merupakan bahasa pemrograman utama untuk mengembangkan aplikasi android.

### 2.1. Apa Itu Java?

Java merupakan bahasa pemrograman yang pertama kali dikembangkan oleh James Gosling di perusahaan Sun Microsystems. Perusahaan tersebut kemudian diakusisi oleh Oracle. Sintaks pada Java banyak mengadopsi dari bahasa pemrograman C dan C++. Oleh karena itu, kita akan banyak menemukan kemiripan sintaks antara Java dan C/C++. Saat ini Java merupakan salah satu bahasa pemrograman paling populer. Jumlah pengguna Java hingga saat ini mencapai jumlah 10 juta orang.

### 2.2. Nama Variabel

Variabel merupakan tempat sebuah informasi disimpan sementara di memory. Variabel dalam java dideklarasikan menggunakan format **type\_data nama\_variable**. Misal **int gear**, dberarti mendeklarasikan sebuah variabel dengan tipe data **int** dan nama variabel adalah **gear**.

Berikut adalah beberapa aturan yang perlu anda ketahui ketika memberikan nama variabel.

1. Nama variabel adalah *case sensitive*. Jadi nama variable **gear** dengan **Gear** adalah dua variabel yang berbeda.
2. Nama variabel bisa mengandung angka, huruf kecil dan besar, \$, dan \_.
3. Jika nama variabel terdiri dari satu kata maka semua harus menggunakan huruf kecil. Misal **gear**, **book**, **node**, etc. Jika lebih dari dua kata, maka setiap kata berikutnya harus diawali dengan huruf besar. Misal **currentUser**, **maxGear**, **minDistance**, etc.

### 2.3. Objek Dan Kelas

Objek dalam pemrograman Java dapat diartikan segala sesuatu yang memiliki sifat dan perilaku. Sedangkan kelas adalah abstraksi yang menjelaskan sifat dan perilaku dari objek tersebut. Kelas juga dapat bermakna sebuah tipe data kustom selain dari tipe data yang sudah ada. Sebuah kelas didefinisikan menggunakan keyword **class**. Perhatikan contoh definisi sebuah kelas berikut:

```
public class Sepeda {
    public String produsen;
    public String merek;
    public int jumlahBan;
    public boolean spion;
}
```

Potongan kode Java di atas adalah untuk mendefinisikan sebuah kelas dengan nama kelas Sepeda. Kelas tersebut harus disimpan dalam sebuah file dengan nama *Sepeda.java*. Kelas tersebut memiliki beberapa variabel kelas antara lain produsen, merek, jumlahBan dan spion.

## 2.4. Tipe Data Pada Java

Di dalam Java, terdapat dua jenis tipe data.

1. Tipe data primitif
2. Tipe data object/class

Tipe data primitif merupakan tipe data sederhana dan sudah terdapat secara built in di dalam Java. Ada delapan tipe data primitif pada Java.

1. **byte**
2. **short**
3. **int**
4. **long**
5. **float**
6. **double**
7. **boolean**
8. **char**

Sedangkan tipe data object merupakan tipe data yang didefinisikan menggunakan *keyword* **class**. Misalkan tipe data Employee, People, dan lain-lain. Programmer dapat mendefinisikan tipe data tersebut sesuai dengan kebutuhan masing-masing. Perhatikan contoh berikut:

```
public class Employee {
    private String name;
    private String wife;
    private int age;
    private int numberOfChildren;
}
```

File tersebut harus disimpan dengan nama *Employee.java*. Yang harus anda perhatikan adalah bahwa, nama kelas harus sama persisi dengan nama file, dalam hal ini adalah Employee. Untuk mendefinisikan sebuah variable dengan tipe Employee, anda harus menggunakan *keyword* **new**. Contoh di bawah ini adalah potongan kode Java untuk

mendefinisikan sebuah variabel dengan tipe Employee dan cara mengakses field dari object tersebut.

```
Employee employee = new Employee();  
employee.name = "Burhan";  
employee.wife = "Sarah";  
employee.age = 45;  
employee.numberOfChildren = 5;
```

## 2.5. Pewarisan (*Inheritance*)

Sebuah kelas dapat mewarisi kelas yang lain jika antara kedua kelas tersebut memiliki kesamaan pada beberapa sifat dan perilaku. Cara mendefinisikan kelas yang mewarisi kelas lain adalah dengan menggunakan *keyword* **extends**. Kelas yang mewarisi kelas lain disebut dengan kelas turunan. Perhatikan contoh kelas di bawah ini:

```
public class Ontel extends Sepeda {  
    public boolean lampu;  
}
```

Potongan kode Java di atas adalah untuk mendefinisikan sebuah kelas dengan nama kelas Ontel. Kelas tersebut merupakan kelas turunan dari kelas Sepeda yang telah didefinisikan sebelumnya. Kelas Ontel memiliki sebuah variabel kelas, yaitu lampu. Akan tetapi karena kelas Ontel adalah kelas turunan dari kelas Sepeda, maka kelas Ontel juga memiliki variabel kelas yang dimiliki oleh kelas Sepeda, yaitu produsen, merek, jumlahBan dan spion.

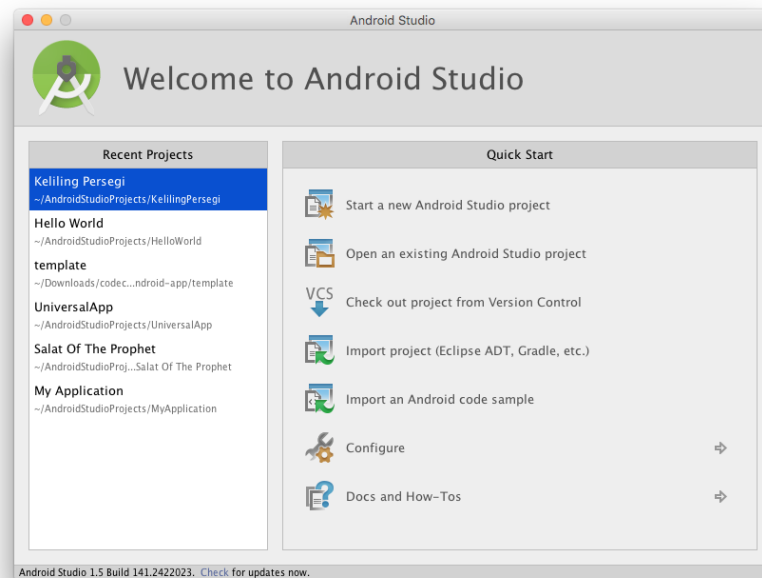
### 3. Membuat Dan Menjalankan Proyek Pertama

Pada bab ini, kita akan mencoba membuat proyek android pertama menggunakan Android Studio. Proyek yang akan kita buat merupakan proyek sederhana yakni proyek "Hello World". Tujuan dari bab ini adalah untuk memberi gambaran umum bagaimana membuat sebuah proyek menggunakan Android Studio sebelum kita membicarakan lebih banyak mengenai konsep-konsep di android itu sendiri.

#### 3.1. Membuat Proyek Baru

Untuk membuat proyek baru pada Android Studio, silahkan mengikuti langkah-langkah berikut:

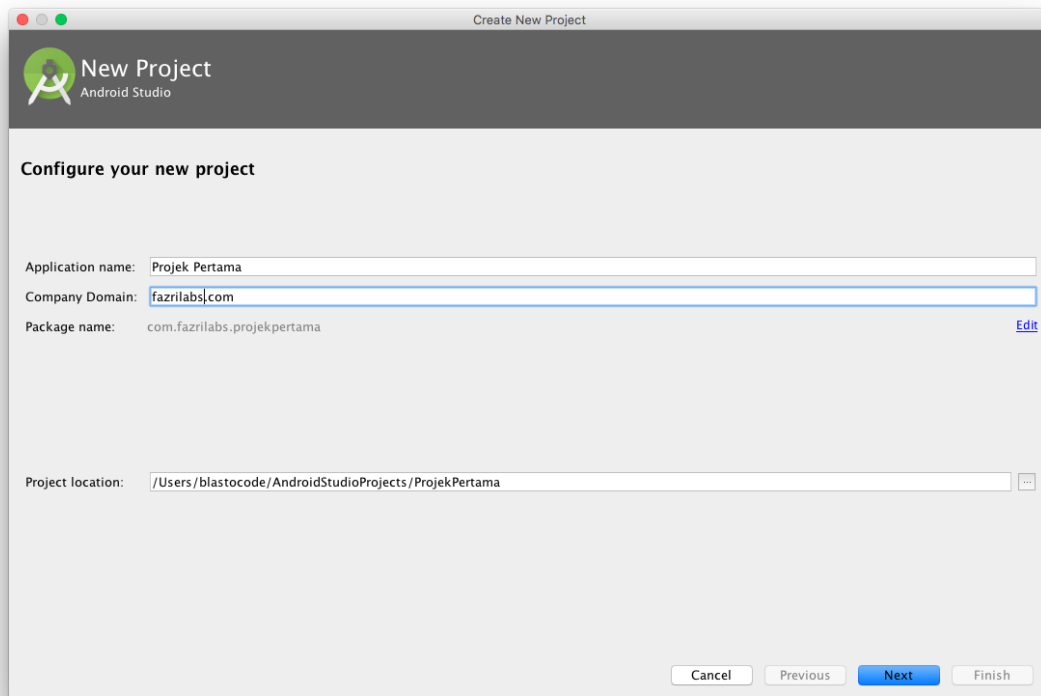
1. Buka aplikasi Android Studio. Silahkan tunggu beberapa saat sampai muncul jendela Selamat Datang berikut:



Jendela di atas terdiri atas dua bagian, bagian pertama adalah Recent Projects. Pada bagian ini kita dapat melihat semua proyek-proyek yang telah kita buat sebelumnya menggunakan Android Studio. Tentu saja bagian ini akan kosong jika anda belum pernah membuat proyek sebelumnya. Bagian kedua adalah Quick Start. Pada bagian ini terdapat beberapa tombol cepat untuk memulai menggunakan Android Studio.

Dari beberapa tombol cepat tersebut, kita hanya perlu menggunakan tombol *Start a new Android Studio project*. Tombol-tombol yang lainnya belum kita perlukan untuk saat ini.

2. Pada jendela Selamat Datang, klik tombol *Start a new Android Studio project* sehingga muncul jendela *Create New Project* berikut:

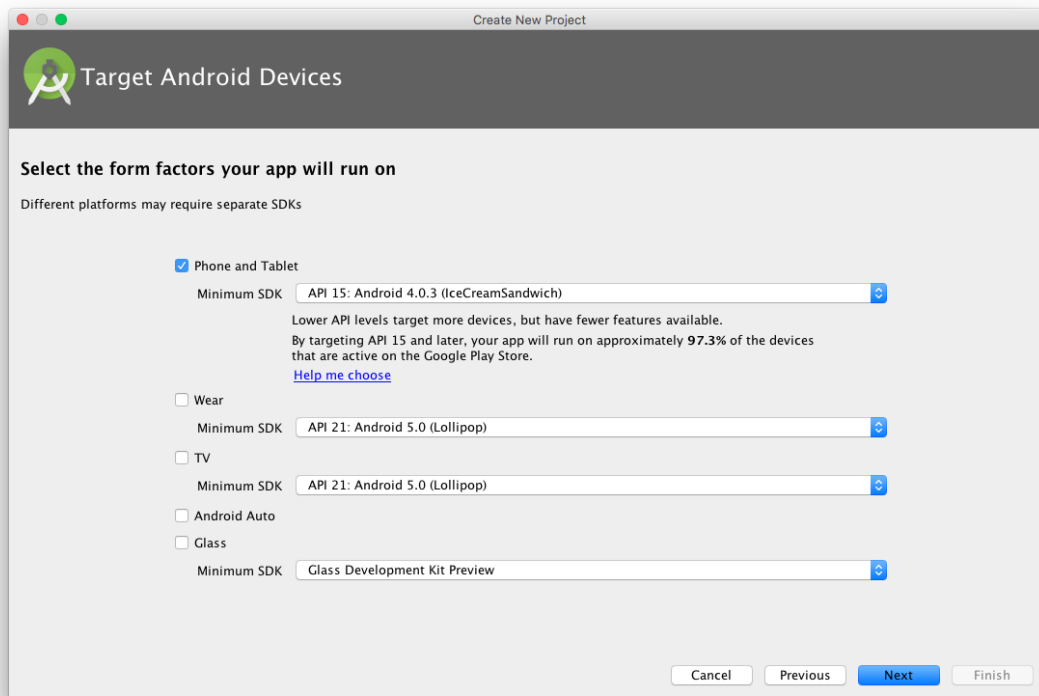


The screenshot shows the 'Create New Project' dialog in Android Studio. The dialog has a title bar with 'Create New Project'. Below the title bar is a header with the Android Studio logo and 'New Project' text. The main area is titled 'Configure your new project'. It contains four input fields: 'Application name' with the value 'Projek Pertama', 'Company Domain' with the value 'fazrilabs.com', 'Package name' with the value 'com.fazrilabs.projekpertama', and 'Project location' with the value '/Users/blastocode/AndroidStudioProjects/ProjekPertama'. There is an 'Edit' link next to the package name field. At the bottom, there are four buttons: 'Cancel', 'Previous', 'Next', and 'Finish'. The 'Next' button is highlighted in blue.

Ada beberapa *field* yang harus kita isi, isilah dengan data berikut:

Field	Nilai	Keterangan
<i>Application name</i>	<b>Projek Pertama</b>	Merupakan nama dari aplikasi yang akan kita buat dan akan muncul dibawah icon aplikasi ketika aplikasi ini diinstall ke ponsel. Field ini harus diisi dan bisa menggunakan spasi
<i>Company Domain</i>	<b>com.fazrilabs</b>	Umumnya menggunakan domain website dari perusahaan pembuat aplikasi. Gabungan dari <i>company domain</i> dan <i>application name</i> akan secara otomatis digunakan pada <i>package name</i> yang merupakan identitas unik dari sebuah aplikasi android di Google Play Store.
<i>Project location</i>	<i>biarkan default</i>	Lokasi penyimpanan projek pada drive

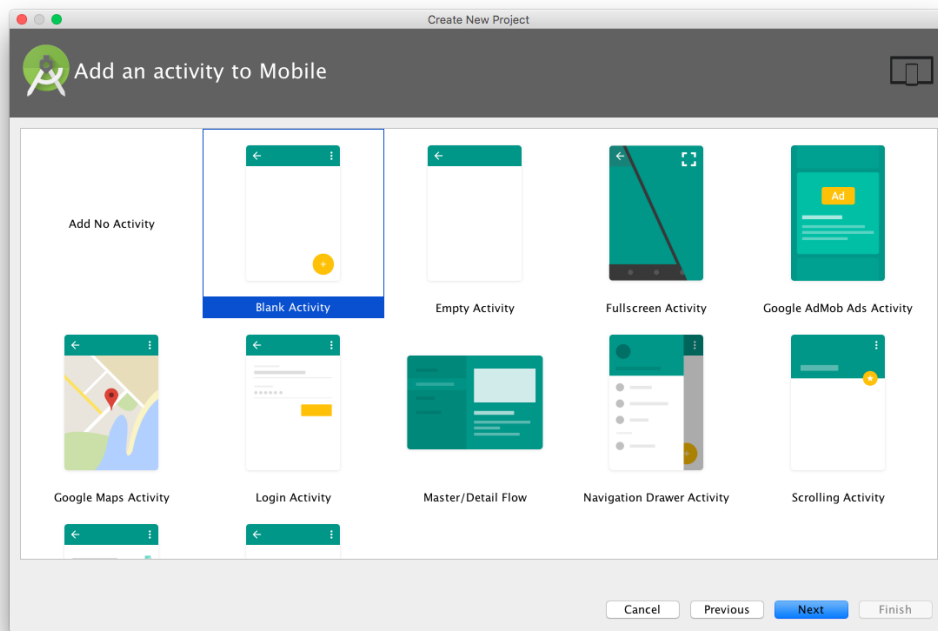
3. Setelah semua field terisi dengan baik, maka lanjutkan dengan menekan tombol *Next* maka akan muncul jendela *Target Android Devices* berikut:



Pada jendela ini kita bisa mengatur target ponsel yang akan menggunakan aplikasi yang akan kita buat. Untuk saat ini hanya centang pilihan *Phone and Tablet* karena kita hanya akan membuat aplikasi tersebut berjalan pada ponsel dan tablet. Pilihan lainnya untuk sementara bisa diabaikan.

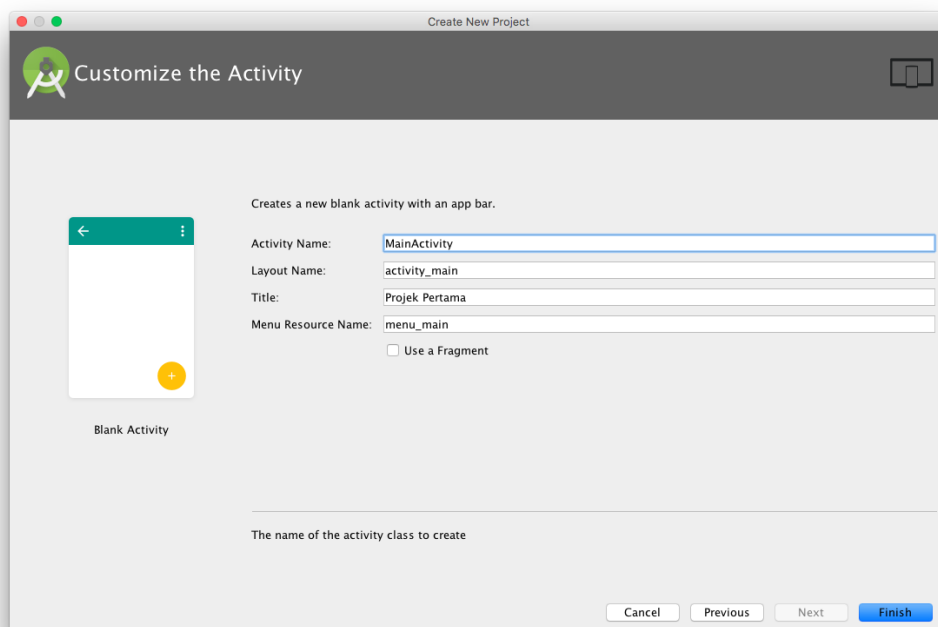
Selain itu kita juga bisa mengatur *Minimum SDK* untuk masing-masing target. Dengan melakukan konfigurasi pada *Minimum SDK* kita bisa mengatur aplikasi hanya dapat berjalan jika ponsel target terinstall android dengan versi minimal sesuai dengan versi yang kita atur pada pilihan tersebut. Misal kita set *API 15 (Ice Cream Sandwich)*, maka aplikasi tersebut tidak akan bisa diinstall pada ponsel dengan android di bawah *Ice Cream Sandwich*. Selain itu, aplikasi tersebut juga tidak akan muncul di *Google Play Store* jika kita mencari melalui ponsel dengan android di bawah *Ice Cream Sandwich*.

4. Kemudian klik tombol *Next* maka akan muncul jendela *Add an activity to Mobile*.



Jendela ini memungkinkan kita untuk menambah *activity* baru ke dalam proyek. Untuk penjelasan mengenai *activity* akan dijabarkan pada bab selanjutnya. Untuk sementara pilih *Blank Activity*.

5. Kemudian klik tombol *Next* maka akan muncul jendela *Customize the Activity*.



Pada jendela tersebut kita bisa mengatur beberapa konfigurasi *activity* yang akan kita buat. Ada beberapa *field* yang harus kita isi. Isilah dengan data-data berikut:

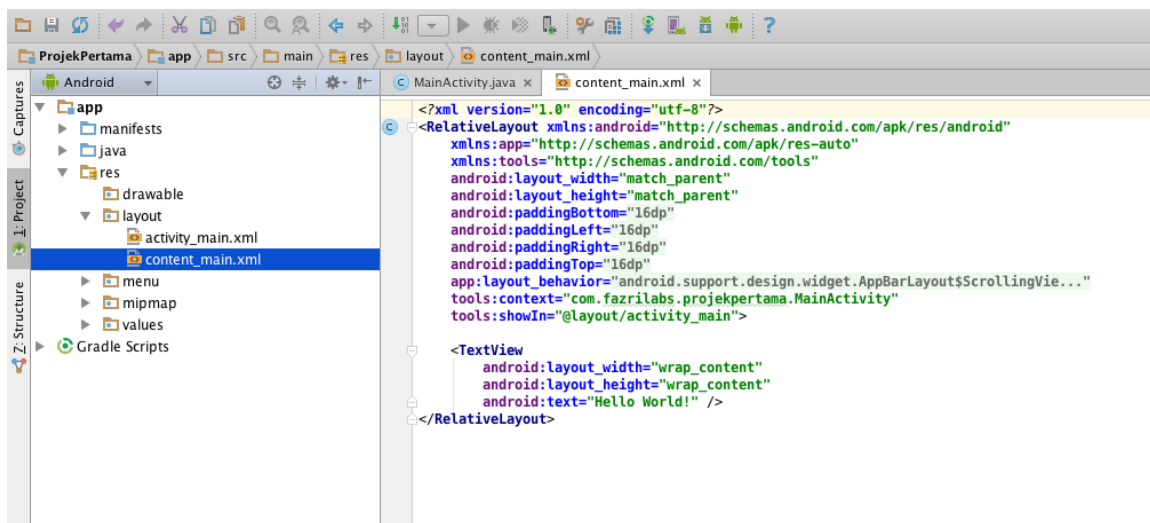
Field	Nilai	Keterangan
-------	-------	------------



<i>Activity Name</i>	<b>MainActivity</b>	Merupakan nama <i>class</i> dari <i>activity</i> yang akan dibuat
<i>Layout Name</i>	<b>activity_main</b>	Merupakan nama <i>layout</i> untuk <i>activity</i> yang akan dibuat
<i>Title</i>	<b>Projek Pertama</b>	Merupakan judul dari <i>activity</i>
<i>Menu Resource Name</i>	<b>menu_main</b>	Merupakan nama file untuk <i>resource</i> dari menu

Untuk sementara pastikan *checkbox Use a Fragment* tidak dicentang.

6. Kemudian klik tombol *Finish*. Silahkan tunggu sebentar, Android Studio akan membuatkan file-file proyek yang dibutuhkan selama beberapa saat. Pastikan komputer anda terkoneksi dengan internet karena Android Studio akan mendownload beberapa library ke dalam proyek melalui gradle. Jika proses telah selesai, maka akan muncul jendela kerja berikut.



### 3.2. Struktur Proyek Android Studio


Ketika kita membuat sebuah proyek baru, maka Android Studio akan secara otomatis membuatkan beberapa folder dan file dalam sebuah struktur yang cukup rapih. Hal tersebut dapat mempercepat kita memulai sebuah proyek baru. Berikut adalah struktur sebuah proyek pada Android Studio.

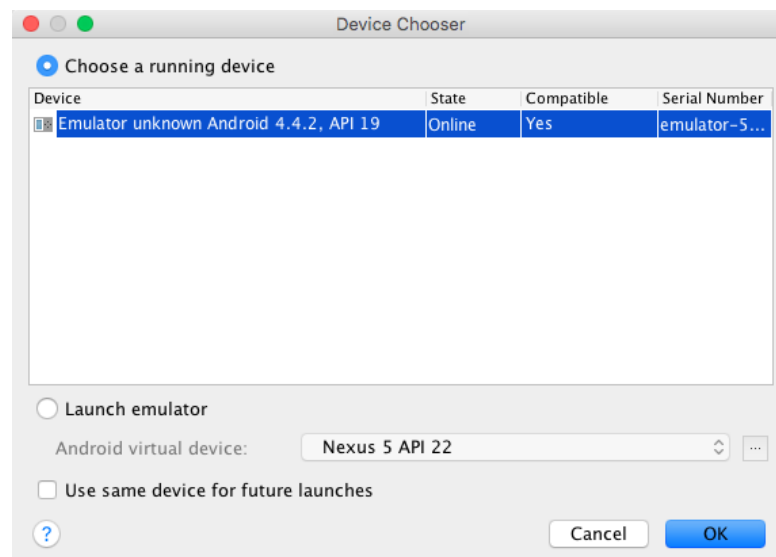
- a. *manifests*: Folder tempat meletakkan file AndroidManifest.xml
- b. *java*: Folder tempat meletakkan file-file Java
- c. *res:drawable*: Folder tempat meletakkan file-file gambar
- d. *res:layout*: Folder tempat meletakkan file-file xml untuk layout
- e. *res:menu*: Folder tempat meletakkan file-file xml untuk menu
- f. *res:mipmap*: Folder tempat meletakkan file-file gambar untuk icon aplikasi
- g. *res:values*: Folder tempat meletakkan file-file xml yang lain.

Kita akan membahas lebih lanjut tentang bagaimana kode-kode tersebut bekerja pada sub bab berikutnya. Untuk saat ini kita akan langsung mencoba menjalankan proyek tersebut untuk melihat hasil akhirnya.

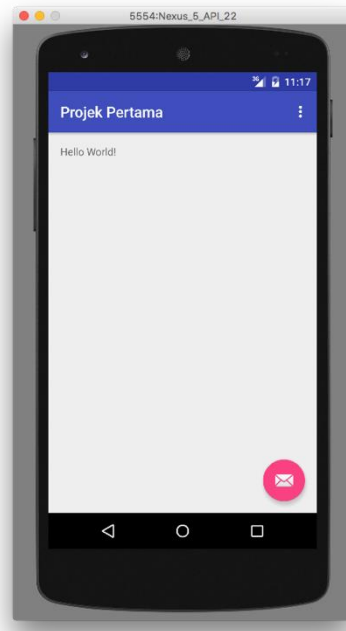
### 3.3. Menjalankan Projek Pada Android Virtual Device

Sebelum menjalankan proyek pada perangkat virtual android, anda harus sudah membuat sebuah perangkat virtual di Android Studio. Jika belum, anda bisa mengikuti panduan pembuatan perangkat virtual pada sub bab 1.4.

Setelah perangkat virtual selesai dibuat kemudian nyalakan perangkat tersebut. Lalu klik tombol  pada toolbar untuk menjalankan proyek. Kemudian akan muncul jendela *Device Chooser*.



Pilih perangkat virtual yang sedang berjalan kemudian klik tombol *OK*. Maka Android Studio akan menginstal aplikasi proyek ke dalam perangkat virtual. Berikut adalah tampilan setelah proyek berhasil diinstal ke perangkat.



### 3.4. Menjalankan Projek Pada GenyMotion

Untuk menjalankan projek menggunakan GenyMotion, anda hanya perlu mengaktifkan salah satu perangkat virtual pada GenyMotion. Setelah salah satu perangkat virtual aktif, maka Android Studio akan otomatis mendeteksi keberadaan perangkat tersebut.

### 3.5. Menjalankan Projek Pada Perangkat Android

Untuk menjalankan projek pada perangkat android, anda harus memiliki sebuah perangkat android yang anda hubungkan dengan komputer anda. Jika anda menggunakan Mac, maka Android Studio otomatis akan mendeteksi keberadaan perangkat android anda. Akan tetapi jika Windows atau Linux, anda harus menginstall dulu driver dari perangkat android tersebut.

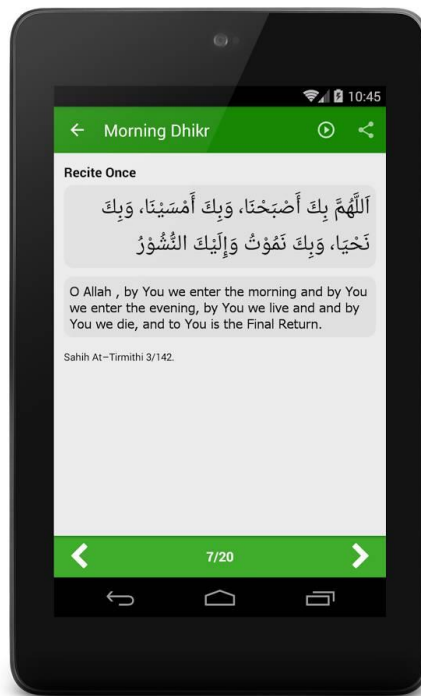
## 4. Mengenal Activity

### 4.1. Apa Itu Activity?

Secara sederhana, sebuah *activity* pada aplikasi android mewakili sebuah layar yang tampak dalam aplikasi tersebut. Di dalam layar tersebut bisa terdapat beberapa komponen *user interface* baik berupa *TextView*, *Button*, *EditText*, *Menu* dan komponen-komponen yang lain. Perhatikan layar utama dari sebuah aplikasi berikut.



Layar aplikasi tersebut dibentuk oleh sebuah *activity*. Dalam layar tersebut terdapat 3 buah tombol. Apabila salah satu tombol diklik, maka akan berganti ke layar kedua.



Layar kedua pada aplikasi tersebut merupakan sebuah *activity* yang lain. Jadi dapat kita simpulkan bahwa sebuah aplikasi android bisa terdiri dari satu atau lebih *activity*. Keseluruhan *activity* secara bersama-sama membangun sebuah aplikasi yang sempurna.

Sebuah *activity* merupakan sebuah kelas yang ditulis menggunakan bahasa pemrograman Java. Berikut ini adalah contoh *activity* dari Proyek Pertama yang telah kita buat pada bab sebelumnya. Silahkan buka proyek pertama anda dan buka file *MainActivity* pada folder *java/com.fazrilabs.projekpertama*.

```
package com.fazrilabs.projekpertama;
```

```
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
```

```
        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
```

```

fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
    }
});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}
}

```

Nama kelas dari *activity* di atas adalah MainActivity yang merupakan turunan dari kelas AppCompatActivity. Tampilan layar pada *activity* tersebut didefinisikan dalam file *main\_activity.xml*. Anda dapat membuka file tersebut pada folder *res/layout*.

```

<?xml version="1.0" encoding="utf-8" ?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="com.fazrilabs.projekpertama.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```

        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        android:src="@android:drawable/ic_dialog_email" />

</android.support.design.widget.CoordinatorLayout>

```

Sekilas kode dalam file *content\_main.xml* tampak rumit, akan tetapi yang perlu anda perhatikan hanya pada bagian baris **<include layout="@layout/content\_main" />**. Baris ini menunjukkan bahwa file xml tersebut mengikutsertakan isi dari file *content\_main*. Anda dapat membuka file *content\_main.xml* pada folder yang sama.

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.fazrilabs.projekpertama.MainActivity"
    tools:showIn="@layout/activity_main">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

</RelativeLayout>

```





Tanda panah pada bagan di atas menunjukkan alur dari sebuah proses. Jadi ketika sebuah activity dijalankan, maka urutan fungsi yang akan dipanggil oleh android secara berurutan adalah *onCreate*, *onStart*, *onResume*.

#### 4.3. Aplikasi Multiscreen

## 5. Resource

Android Studio mengorganisir file-file selain kode java dalam sebuah folder yang disebut *res*. Di dalam folder tersebut kita dapat meletakkan berbagai macam file pendukung seperti teks, image, audio dan file yang lain. Umumnya file tersebut merupakan sebuah file xml yang menyimpan berbagai macam informasi penting seperti kumpulan teks, warna, dimensi dan lain-lain. File-file tersebut dinamakan *resource*. Setiap *resource* dapat digunakan dalam kode java dengan menggunakan format *R.jenis\_resource.nama\_resource*. Misal *R.string.app\_name*, *R.drawable.logo* dan lain sebagainya. Selain digunakan dalam kode java, *resource* jg bisa digunakan oleh *resource* yang lain.

Penggunaan *resource* dapat mempermudah pengembang aplikasi dalam mengatur karakteristik aplikasi tanpa harus mengubah kode java. Sebagai contoh sederhana, ketika akan mengatur judul dari sebuah *activity*, kita bisa saja menuliskan kode berikut.

```
setTitle("Judul Pertama");
```

Walaupun cara tersebut bisa dilakukan dan tidak menyebabkan kesalahan, akan tetapi cara yang direkomendasikan adalah dengan menggunakan *resource* teks sehingga kode di atas akan menjadi

```
setTitle(R.string.app_name);
```

Dalam hal ini *R.string.app\_name* merupakan sebuah *resource* teks yang bisa kita definisikan dalam sebuah file *string.xml* yang kita letakkan pada folder */res/values*. Cara mendefinisikan resource string adalah sebagai berikut

```
<resources>
  <string name="app_name">Judul Pertama</string>
</resources>
```

Ketika aplikasi dijalankan, maka *resource* *R.string.app\_name* akan diubah menjadi teks Judul Pertama. Keunggulan menggunakan cara ini adalah semua teks tertata dengan rapih dalam satu file xml sehingga memudahkan ketika akan mencarinya tanpa harus mencari di dalam kode java. Selain itu kita bisa melakukan lokalisasi bahasa ke bahasa lain dengan mudah tanpa harus merubah banyak pada kode java. Misal bahasa Perancis, Indonesia, Malaysia dan bahasa lainnya.

### 5.1. Jenis Resource

Ada beberapa jenis resource yang dapat digunakan antara lain *resource* teks, *resource* array dari teks, *resource drawable*, *resource* warna (*color*), *resource* dimensi dan *resource* layout. Pembahasan mengenai masing-masing *resource* dan cara penggunaannya akan dijabarkan pada sub bab berikutnya.

## 5.2. Resource Layout

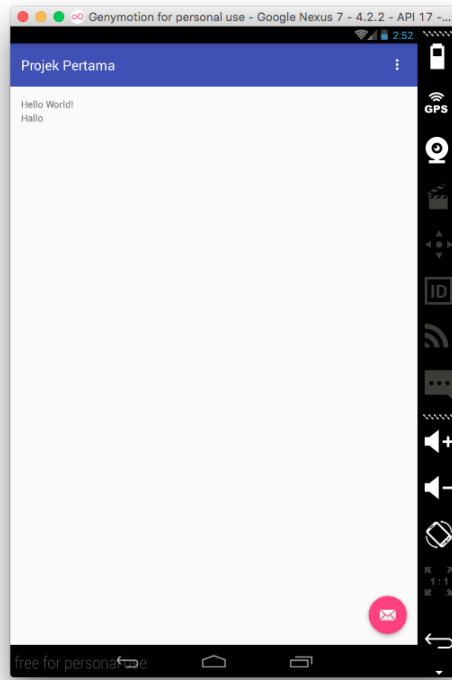
Resource layout merupakan file xml yang membentuk tatap muka aplikasi. Di dalam resource tersebut anda dapat mengatur dan menyusun berbagai macam komponen tatap muka seperti tombol, gambar, label, teks input dan lain sebagainya. Anda juga dapat mengontrol bagaimana komponen-komponen tersebut tersusun. Resource layout harus diletakkan di dalam folder `/res/layout`. Berikut adalah contoh sebuah file layout dengan nama `content_main.xml`.

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.fazrilabs.projekpertama.MainActivity"
    tools:showIn="@layout/activity_main">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hallo" />
</LinearLayout>
```

Untuk dapat menggunakan layout tersebut, anda harus mendaftarkannya pada activity dengan cara memanggil metode `setContentView(R.layout.content_main)`. Gambar di bawah menunjukkan hasil pratinjau dari layout di atas.



Pembahasan tentang layout akan dijelaskan lebih lanjut pada bab berikutnya.

### 5.3. Resource Teks

Ketika mengembangkan sebuah program atau aplikasi, baik itu aplikasi komputer atau aplikasi ponsel, anda akan banyak menggunakan teks seperti memberi label pada tombol, teks pada menu, teks pada judul aplikasi, teks pada pop pesan dan lain sebagainya. Teks tersebut merupakan teks statis. Pada aplikasi android juga berlaku hal tersebut. Untuk kemudahan pengelolaan teks pada aplikasi android maka teks bisa kita pisahkan dari kode java dan diletakkan dalam sebuah file xml. File ini diletakkan dalam folder */res/values*.

Anda bisa mendefinisikan banyak teks pada file *string.xml*. Masing-masing teks memiliki nama sehingga kita bisa menggunakan teks tersebut pada kode java. Berikut ada contoh dari sebuah file *string.xml*.

```
<resources>
  <string name="app_name">Projek Pertama</string>
  <string name="action_settings">Settings</string>
  <string name="action_add">Add</string>
  <string name="action_delete">Delete</string>
  <string name="action_edit">Edit</string>
</resources>
```

Setelah anda mendefinisikan teks pada file *string.xml*, anda bisa menggunakan teks tersebut dalam beberapa cara berikut:

1. Penggunaan *resource* teks dalam *resource* yang lain

*Resource* teks dapat digunakan dalam *resource* yang lain dengan menggunakan format *@string/name\_string*. Berikut ini adalah contoh potongan kode dari sebuah *resource layout* yang didalamnya menggunakan *resource* teks.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello_world" />
```

## 2. Penggunaan resource teks dalam kode java

Resource teks dapat digunakan dalam kode java dengan menggunakan format berikut *R.string.nama\_string*. Berikut adalah potongan kode java yang menggunakan *resource* teks.

```
String appName = getResources().getString(R.string.app_name);
```

Potongan kode di atas bermakna "ambil *resource* teks dengan nama *app\_name* pada file *string.xml* dan simpan ke dalam variabel *appName*". Selain dengan cara di atas beberapa metode dalam sebuah *activity* bisa langsung menerima input dalam format *R.string.nama\_string*. Perhatikan contoh potongan kode java berikut:

```
setTitle(R.string.app_name);
```

## 5.4. Resource Array Teks

*Array* teks merupakan kumpulan dari beberapa teks yang disimpan dalam sebuah variabel bertipe *string-array*. Contoh dari *array* adalah kumpulan teks nama negara, kumpulan teks nama warna, kumpulan teks nama sekolah dan lain sebagainya. Anda bisa mendefinisikan kumpulan teks tersebut dalam sebuah *array* pada file *string.xml*. Berikut adalah cara mendefinisikan kumpulan teks dalam bentuk *array* pada file *string.xml*.

```
<string-array name="countries">
    <item>Indonesia</item>
    <item>Malaysia</item>
    <item>Jepang</item>
    <item>Singapura</item>
    <item>Thailand</item>
</string-array>
```

Potongan kode adalah contoh cara mendefinisikan kumpulan teks nama negara dalam sebuah *array* yang diberi nama **countries**. *Resource array* teks hanya bisa digunakan dalam kode java. Untuk menggunakan *resource array* teks, anda bisa menggunakan format *R.array.nama\_array*. Berikut adalah contoh potongan kode java yang menggunakan *resource array* teks.

```
String[] countries = getResources().getStringArray(R.array.countries);
```

Potongan kode java di atas bermakna "ambil *resource array* teks dengan nama ***countries*** pada file *string.xml* dan simpan ke dalam variabel *countries*".

## 5.5. Resource Drawable

Ketika anda mengembangkan sebuah aplikasi android, tentu saja tidak terlepas dari penggunaan gambar. Anda mungkin saja menggunakan gambar sebagai *splash screen*, logo perusahaan, icon pada sebuah tombol dan lain sebagainya. Dalam android jenis resource gambar disebut drawable. Anda dapat meletakkan gambar-gambar yang butuhkan tersebut di dalam folder */res/drawable*.

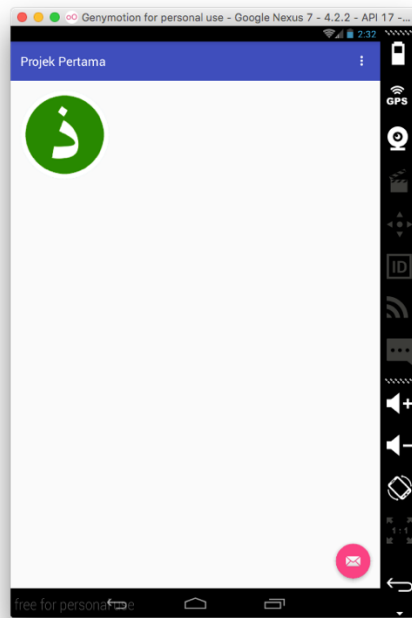
Resource drawable merupakan jenis resource untuk gambar, baik yang dibuat menggunakan editor gambar seperti photoshop ataupun yang dibuat menggunakan file xml pada Android Studio. File gambar yang digunakan bisa bertipe png, jpg dan gif. Yang perlu anda perhatikan adalah bahwa nama file gambar tidak boleh menggunakan spasi.

Resource drawable dapat digunakan dalam file resource yang lain dengan menggunakan format *@drawable/nama\_gambar*. Hal yang perlu anda perhatikan adalah *nama\_gambar* dalam format tersebut tidak boleh mengandung ekstensi. Misalkan anda mempunyai sebuah gambar dengan nama *adzkar.png*. Letakkan file gambar tersebut pada folder */res/drawable*. Setelah itu anda bisa menggunakan gambar tersebut di dalam sebuah file resource layout dengan cara sebagai berikut:

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/adzkar"/>
</RelativeLayout>
```

Perhatikan bahwa penggunaan resource drawable di atas dilakukan tanpa menuliskan ekstensinya, yaitu ***@drawable/adzkar*** bukan ***@drawable/adzkar.png***. Berikut adalah pratinjau dari layout tersebut.



Walaupun ukuran layar perangkat android berbeda-beda, akan tetapi android akan secara otomatis menyesuaikan ukuran gambar dengan ukuran layar. Android memberikan keleluasan kepada pengembang aplikasi untuk menyediakan gambar berbeda untuk ukuran layar yang berbeda. Jadi untuk sebuah gambar, anda bisa menyediakan beberapa versi ukuran, android akan secara otomatis mengambil gambar yang sesuai dengan ukuran layar. Untuk keperluan tersebut, anda harus membuat folder tambahan di dalam folder *res*, yaitu:

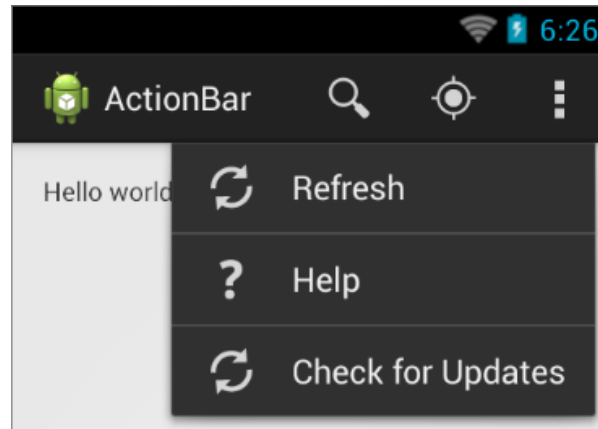
1. *drawable-ldpi* (low)
2. *drawable-mdpi* (medium)
3. *drawable-hdpi* (high)
4. *drawable-xhdpi* (extra-high)
5. *drawable-xxhdpi* (extra-extra-high)

Untuk mempermudah anda memahami konsep di atas, perhatikan contoh variasi ukuran gambar dari *adzkar.png*.

mdi	hdpi	xhdpi	xxhdpi
48 x 48	72 x 72	96 x 96	144 x 144

## 5.6. Resource Menu

Salah satu aspek penilaian pengguna aplikasi terhadap kualitas sebuah aplikasi adalah *user friendly*. Dengan kata lain, pengguna akan lebih menyukai aplikasi yang mudah ketika digunakan. Salah satu faktor terpenting agar aplikasi yang anda buat mudah digunakan adalah dengan membuat menu. Menu merupakan bagian penting dari aplikasi android. Menu dapat mempermudah user dalam melakukan navigasi ke berbagai halaman dalam aplikasi. Gambar di bawah menunjukkan sebuah menu.



Membuat menu dalam aplikasi android sangatlah mudah. Anda cukup mendefinisikan menu tersebut sebagai sebuah file xml yang diletakkan di dalam folder */res/menu*. Berikut adalah contoh sebuah resource menu dengan nama file *menu\_main.xml*.

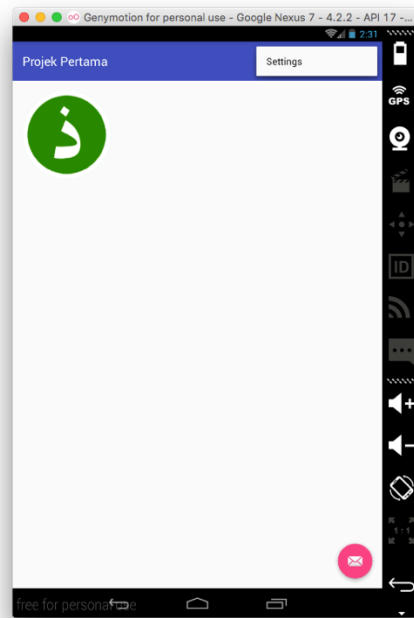
```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.fazrilabs.projekpertama.MainActivity">
    <item
        android:id="@+id/action_settings"
        android:title="@string/action_settings"
        app:showAsAction="never" />
</menu>
```

Maksud kode xml di atas adalah mendaftarkan sebuah menu dengan id **action\_settings** dan judul **@string/action\_settings** yang merujuk pada resource teks. Sedangkan potongan kode **app:showAsAction="never"** maksudnya adalah bahwa menu tersebut adalah popup menu. Kemudian anda perlu mendaftarkan menu tersebut pada sebuah activity agar ditampilkan pada activity yang bersangkutan.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
```

Hasilnya adalah sebagai berikut:





Untuk mengetahui ketika menu tersebut diklik adalah dengan cara meng-*override* metode pada `onOptionsItemSelected` activity.

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.action_settings) {
        // lakukan sesuatu disini
        return true;
    }

    return super.onOptionsItemSelected(item);
}
```

## 6. View

View merupakan salah satu materi yang penting untuk dipelajari. Setiap aplikasi android pasti akan menggunakan view sebagai salah satu komponennya. Memahami setiap view beserta konfigurasinya sangat membantu seorang pengembang dalam mengembangkan aplikasi android.

### 6.1. Mengenal View dan ViewGroup

Secara sederhana, view dapat kita artikan sebagai komponen tatap muka atau biasa disebut dengan *user interface*. Masing-masing komponen memiliki fungsi dan tujuan yang berbeda. Akan tetapi semuanya bersinergi satu sama lain untuk membentuk satu tujuan utama. Pengembang aplikasi dapat menyusun berbagai macam view dalam satu layar untuk suatu tujuan yang mereka inginkan.

Sedangkan viewgroup atau disebut juga *parents* merupakan jenis view yang dapat menampung view-view yang lain sebagai anak-anaknya. Contoh viewgroup adalah LinearLayout, RelativeLayout, FrameLayout, dan lain-lain. Masing-masing viewgroup memiliki karakteristik berbeda-beda sesuai dengan maksud dan tujuannya. ViewGroup akan dibahas lebih mendetail pada bab berikutnya ketika membahas tentang layout.

Berikut adalah beberapa atribut view yang perlu anda pahami:

#### 1. *id*

Apabila anda ingin mengakses sebuah view melalui kode java, maka anda harus memberikan atribut **android:id** pada view tersebut. Nilai dari atribut tersebut adalah sembarang teks dengan format **@+id/idview**. Nilai dari **android:id** harus unik dalam 1 file xml. Perhatikan contoh kode xml berikut:

```
<?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">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"/>
</LinearLayout>
```

Kode xml di atas mendefinisikan sebuah *button* dengan **android:id** adalah **button1**. Untuk mengakses button tersebut melalui kode java, gunakan format **R.id.idview**. Perhatikan potongan kode java berikut:

```
Button button1 = (Button) findViewById(R.id.button1);
```

Selain berguna untuk mengakses dalam kode java, pemberian id juga berguna ketika anda ingin mengakses view tersebut dalam file xml yang sama. Teknik ini biasa dipakai dalam RelativeLayout. Perhatikan contoh kode xml berikut:

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"/>

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button1"
        android:text="Button 1"/>
</RelativeLayout>
```

Pada kode xml di atas, didefinisikan dua buah button dalam sebuah RelativeLayout. Id dari kedua button tersebut masing-masing **button1** dan **button2**. Button2 mengacu kepada button1 dengan menggunakan atribut **android:layout\_below="@+id/button1"**. Pembahasa detil mengenai RelativeLayout akan dibahas pada bab berikutnya.

## 2. *layout\_width* dan *layout\_height*

Setiap view harus memiliki minimal dua atribut berikut, yaitu **android:layout\_width** dan **android:layout\_height**. Kedua atribut tersebut masing-masing mendefinisikan lebar dan tinggi dari view. Berikut adalah nilai yang bisa anda gunakan untuk mengisi kedua atribut tersebut.

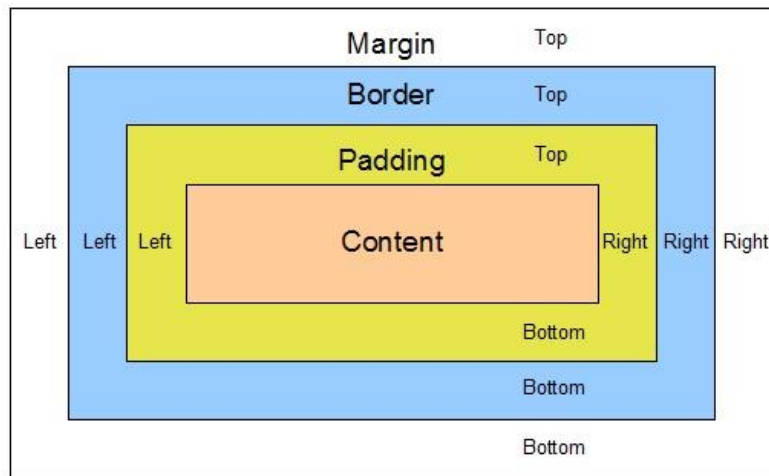
Nilai	Keterangan
<b>wrap_content</b>	Ukuran menyesuaikan dengan ukuran konten view tersebut
<b>match_parent</b>	Ukuran menyesuaikan dengan ukuran kontainer

Selain kedua nilai tersebut, anda juga bisa memberikan sembarang nilai dengan menggunakan satuan **dp** (density-independent pixels). Berbeda dengan piksel (*px*), **dp** merupakan satuan abstrak yang berdasarkan pada kerapatan piksel dari sebuah layar. Berikut adalah contoh potongan kode xml yang menggunakan **dp**.

```
<TextView
    android:layout_width="150dp"
    android:layout_height="20dp"
    android:text="@string/hello_world" />
```

## 3. *margin* dan *padding*

*Margin* adalah jarak antara batas (*border*) view dengan *parent* atau view lainnya. Sedangkan *padding* adalah jarak antara batas view dengan konten view tersebut. Gambar di bawah mengilustrasikan bagaimana margin dan padding terletak pada sebuah view.



Untuk memberikan padding dan margin pada sebuah view, anda cukup memberikan nilai padding dan margin tersebut dalam satuan **dp**. Perhatikan penggunaan margin dan padding di bawah ini.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:layout_margin="5dp"
    android:text="@string/hello_world" />
```

Potongan kode xml di atas adalah pendefinisian sebuah view *TextView*. Atribut **android:padding="10dp"** maksudnya adalah memberikan padding sebesar **10dp** pada setiap sisi view. Sedangkan atribut **android:layout\_margin="5dp"** adalah untuk memberikan margin sebesar **5dp** pada setiap sisi view.

Android mengizinkan anda untuk mengatur margin dan padding hanya pada sisi tertentu. Tabel berikut merangkum beberapa atribut untuk keperluan tersebut.

Atribut	Keterangan
<b>android:paddingTop</b>	Mengatur padding atas view
<b>android:paddingRight</b>	Mengatur padding kanan view
<b>android:paddingBottom</b>	Mengatur padding bawah view
<b>android:paddingLeft</b>	Mengatur padding kiri view
<b>android:layout_marginTop</b>	Mengatur margin atas view
<b>android:layout_marginRight</b>	Mengatur margin kanan view
<b>android:layout_marginBottom</b>	Mengatur margin bawah view
<b>android:layout_marginLeft</b>	Mengatur margin kiri view

## 6.2. TextView

TextView merupakan view yang berfungsi untuk menampilkan teks. Teks yang ditampilkan merupakan teks yang tidak dapat diedit oleh user. Berikut adalah beberapa atribut TextView yang penting untuk dipelajari:

Atribut	Keterangan
<b>android:id</b>	Mendefinisikan ID unik untuk TextView
<b>android:gravity</b>	Mengatur penyalarsan ( <i>alignment</i> ) teks pada TextView
<b>android:text</b>	Teks yang akan ditampilkan
<b>android:textAllCaps</b>	Menampilkan teks dalam huruf besar semua
<b>android:textColor</b>	Mengatur warna teks dalam format <i>#rgb</i>
<b>android:textSize</b>	Mengatur ukuran teks dalam <i>dp</i>
<b>android:textStyle</b>	Mengatur gaya teks: <i>normal</i> , <i>italic</i> , or <i>bold</i>
<b>android:typeface</b>	Mengatur <i>typeface</i> : <i>normal</i> , <i>sans</i> , <i>serif</i> or <i>monospace</i>

Selain atribut di atas, masih banyak atribut-atribut yang lain yang tidak memungkinkan untuk ditampilkan disini. Untuk melihat atribut yang lebih lengkap, anda dapat melihat dokumentasi yang terdapat pada situs *android.com*.

Berikut adalah contoh pemakain TextView:

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/text1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="hello_world"
        android:textColor="#3ebc3e"
        android:textStyle="bold"
        android:layout_centerInParent="true"
        android:textSize="50dp"/>

</RelativeLayout>
```

Hasil dari kode xml di atas adalah seperti pada gambar di bawah ini:



Berikut adalah contoh kode java untuk mengakses TextView dan merubah teks yang muncul menjadi "Belajar Android":

```
TextView textView1 = (TextView) findViewById(R.id.textView1);
textView1.setText("Belajar Android");
```

### 6.3. EditText

EditText merupakan view untuk menerima input dari user. EditText adalah sangat penting jika anda ingin membuat sebuah form yang mengharuskan user menginput beberapa teks. Berikut adalah

Atribut	Keterangan
<b>android:id</b>	Mendefinisikan ID unik untuk EditText
<b>android:autoText</b>	Otomatis memperbaiki kesalahan pengejaan
<b>android:hint</b>	Label petunjuk pengisian ketika EditText kosong
<b>android:inputType</b>	Jenis tipe data yang dapat dimasukan: Phone, Date, Time, Password dan lainnya

Berikut adalah contoh penggunaan atribut di dalam EditText:

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```

android:layout_width="match_parent"
android:layout_height="match_parent">

<EditText
    android:id="@+id/editText1"
    android:layout_width="200dp"
    android:layout_height="50dp"
    android:hint="Silahkan input nama"
    android:layout_centerInParent="true"/>

</RelativeLayout>

```

Berikut adalah pratinjau dari kode di atas:



Berikut ada contoh cara mengakses EditText melalui kode Java dan mengambil nilai yang telah dimasukkan oleh user.

```

EditText editText = (EditText) findViewById(R.id.editText1);
String text = editText.getText().toString();

```

#### 6.4. Button

Button merupakan view yang dapat diklik/ditap oleh user. Biasanya digunakan untuk memberikan user akses terhadap aksi tertentu. Sebuah Button memiliki label yang dapat menjelaskan kepada user fungsi dari Button tersebut. Teks pada Button dapat diatur melalui atribut. Berikut adalah beberapa atribut penting dari Button:

Atribut	Keterangan
<b>android:id</b>	Mendefinisikan ID unik dari Button
<b>android:text</b>	Teks yang akan ditampilkan pada Button

Berikut adalah contoh penggunaan Button:

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

Kode xml di atas akan menghasilkan layout seperti pada gambar di bawah ini:





Untuk mengakses Button melalui kode java dan mengetahui ketika Button tersebut ditap, silahkan perhatikan contoh di bawah ini:

```
Button button = (Button) findViewById(R.id.button1);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // button diklik oleh user
    }
});
```

## 6.5. ImageView

ImageView merupakan view yang dapat digunakan untuk menampilkan gambar. Anda dapat menampilkan gambar dengan tipe png dan jpg. Sebelum anda dapat menggunakan gambar pada ImageView, anda harus meletakkan gambar tersebut pada folder */res/drawable*. Berikut beberapa atribut ImageView yang penting untuk dipahami:

Atribut	Keterangan
<b>android:id</b>	Mendefinisikan ID unik dari ImageView
<b>android:scaleType</b>	Mengatur jenis skala yang digunakan, antara lain: <i>center</i> , <i>center_crop</i> , <i>center_inside</i> , <i>fit_center</i> , <i>fit_end</i> , <i>fit_start</i> , <i>fit_xy</i> , <i>matrix</i> .
<b>android:src</b>	Image yang akan ditampilkan

Perhatikan contoh penggunaan ImageView dalam layout berikut:

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:scaleType="center"
        android:src="@drawable/adzkar"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

File xml di atas akan menghasilkan tampilan layout seperti pada gambar di bawah ini:



## 6.6. ImageButton

ImageButton merupakan Button yang dapat menampilkan gambar sebagai ganti teks. Dengan menggunakan image, sebuah ImageButton akan tampak lebih menarik oleh user dibandingkan Button biasa. Berikut adalah beberapa atribut penting dari ImageButton:

Atribut	Keterangan
<b>android:id</b>	Mendefinisikan ID unik dari ImageButton
<b>android:src</b>	Image yang akan ditampilkan

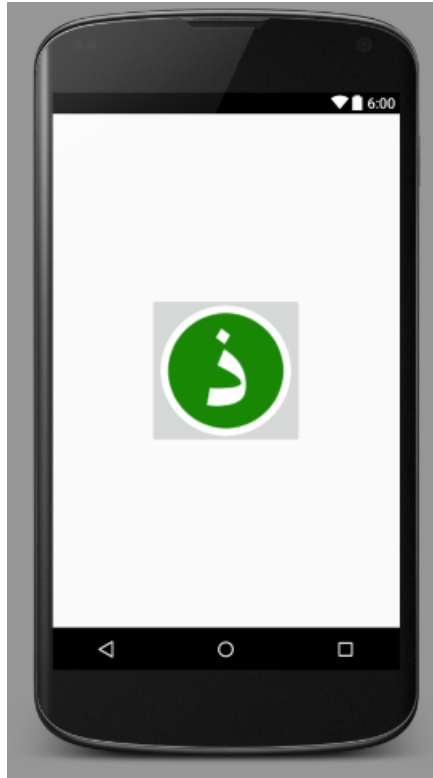
Perhatikan contoh penggunaan ImageButton di bawah ini:

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageButton
        android:id="@+id/imageButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/adzkar"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

Kode xml di atas akan menghasilkan tampilan seperti pada gambar di bawah ini:



Cara mengakses ImageButton melalui kode java hampir mirip dengan Button. Perhatikan contoh potongan kode java di bawah ini:

```
ImageButton imageButton = (ImageButton) findViewById(R.id.imageButton1);
imageButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // imageButton diklik oleh user
    }
});
```

## 6.7. Spinner

Spinner merupakan view yang mempunyai drop down menu sehingga user dapat memilih salah satu item dalam drop down menu tersebut. Berikut adalah beberapa atribut Spinner yang penting untuk diketahui:

Atribut	Keterangan
<b>android:id</b>	Mendefinisikan ID unik dari ImageButton
<b>android:entries</b>	Array teks yang akan dipilih oleh user
<b>android:prompt</b>	Hint untuk user

Untuk mendemonstrasikan penggunaan Spinner, langkah pertama anda perlu mendefinisikan array teks pada resource *string.xml*.

```

<string name="country_prompt">Pilih negara</string>
<string-array name="countries">
    <item>Indonesia</item>
    <item>Malaysia</item>
    <item>Jepang</item>
    <item>Singapura</item>
    <item>Thailand</item>
</string-array>

```

Setelah itu anda dapat menggunakan array teks tersebut untuk mengisi atribut **android:entries** pada Spinner.

```

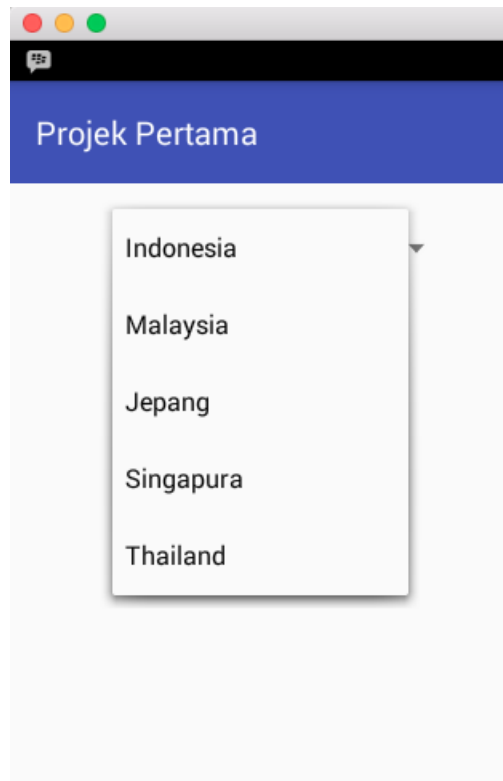
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="200dp"
        android:layout_height="50dp"
        android:entries="@array/countries"
        android:prompt="@string/country_prompt"
        android:layout_centerInParent="true">

    </Spinner>
</RelativeLayout>

```

Berikut adalah tampilan dari Spinner tersebut:



Berikut adalah potongan kode java untuk mengakses Spinner dan mengetahui kapan sebuah item dipilih oleh user:

```
final Spinner spinner = (Spinner) findViewById(R.id.spinner1);
spinner.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        String selectedCountry = spinner.getSelectedItem().toString();
    }
});
```

## 6.8. CheckBox

CheckBox merupakan jenis input yang memiliki kondisi on dan off. CheckBox biasa digunakan dalam sebuah form sebagai field yang memerlukan jawaban ya atau tidak. Berikut adalah beberapa atribut CheckBox yang penting:

Atribut	Keterangan
<b>android:id</b>	Mendefinisikan ID unik dari CheckBox
<b>android:text</b>	Teks yang akan ditampilkan
<b>android:drawableRight</b>	Gambar yang akan ditampilkan di sebelah kanan teks
<b>android:drawableBottom</b>	Gambar yang akan ditampilkan di sebelah bawah teks

Perhatikan contoh penggunaan CheckBox pada layout di bawah ini:

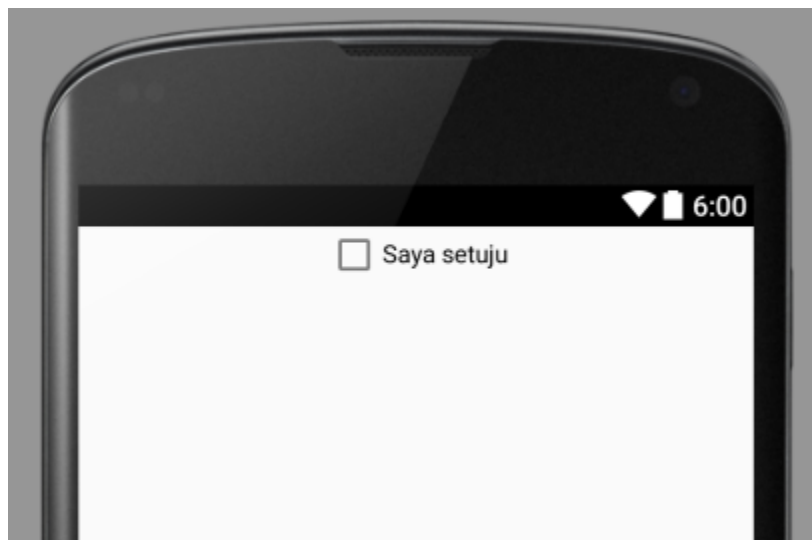
```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <CheckBox
        android:id="@+id/checkBox1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Saya setuju"
        android:layout_centerHorizontal="true"/>
</RelativeLayout>

```

Kode xml di atas akan menghasilkan tampilan seperti pada gambar di bawah ini:



Untuk mengetahui kapan sebuah CheckBox diklik oleh user, kita dapat menggunakan metode `setOnCheckedChangeListener`. Perhatikan potongan kode java di bawah ini:

```

CheckBox checkBox = (CheckBox) findViewById(R.id.checkBox1);
checkBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        // CheckBox diklik
    }
});

```

## 6.9. AutoCompleteTextView

AutoCompleteTextView merupakan view yang mirip dengan EditText, hanya saja AutoCompleteTextView akan menampilkan drop down menu berupa teks prediksi ketika

user mengetik. Berikut adalah beberapa atribut AutoCompleteTextView yang penting untuk dipahami:

Atribut	Keterangan
<b>android:id</b>	Mendefinisikan ID unik dari AutoCompleteTextView
<b>android:completionThreshold</b>	Jumlah minimal karakter yang harus diketika sebelum drop down menu muncul

Untuk mendemonstrasikan penggunaan AutoCompleteTextView, perhatikan contoh di bawah. Definisikan sebuah resource array teks berikut pada file *string.xml*:

```
<string-array name="countries">
  <item>Afrika Selatan</item>
  <item>Arab Saudi</item>
  <item>Afganistan</item>
  <item>Albania</item>
  <item>Aljazair</item>
  <item>Amerika Serikat</item>
  <item>Indonesia</item>
  <item>India</item>
  <item>Malaysia</item>
  <item>Jepang</item>
  <item>Singapura</item>
  <item>Thailand</item>
</string-array>
```

Kemudian buat layout berikut:

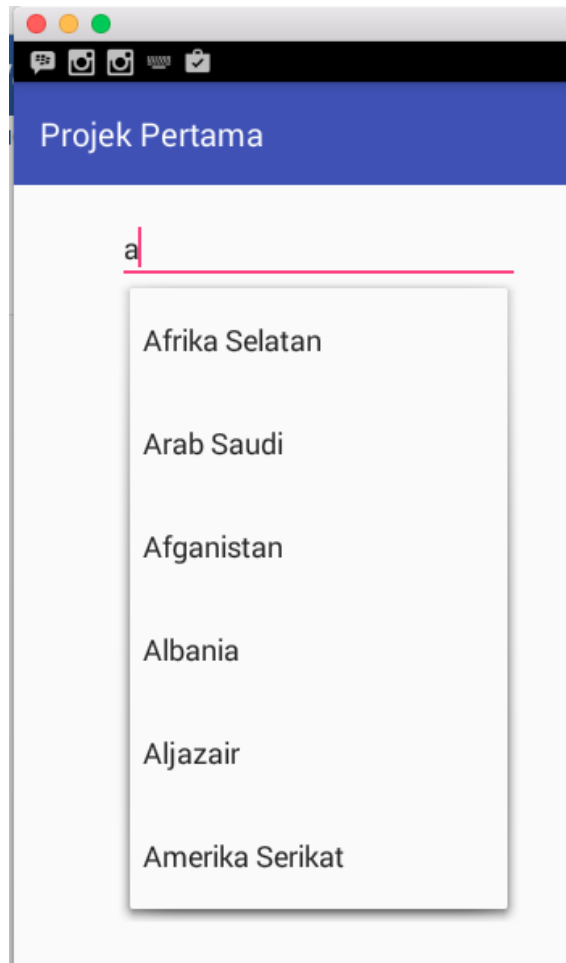
```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <AutoCompleteTextView
    android:id="@+id/autoCompleteTextView1"
    android:layout_width="250dp"
    android:layout_height="wrap_content"
    android:completionThreshold="1"
    android:layout_centerHorizontal="true" />
</RelativeLayout>
```

Kemudian, anda harus mengkonfigurasi AutoCompleteTextView melalui kode java.

```
AutoCompleteTextView autoCompleteTextView = (AutoCompleteTextView)
findViewById(R.id.autoCompleteTextView1);
String[] countries = getResources().getStringArray(R.array.countries);
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
  android.R.layout.simple_dropdown_item_1line, countries);
autoCompleteTextView.setAdapter(adapter);
```

Berikut adalah tampilan dari layout di atas:



#### 6.10. ToggleButton

ToggleButton merupakan Button yang memiliki kondisi on atau off. Hampir mirip dengan CheckBox, hanya saja penampilan ToggleButton lebih mirip dengan Button. Berikut adalah beberapa atribut penting dari ToggleButton:

Atribut	Keterangan
<b>android:id</b>	Mendefinisikan ID unik dari ToggleButton
<b>android:textOff</b>	Teks yang akan ditampilkan ketika kondisi Off
<b>android:textOn</b>	Teks yang akan ditampilkan ketika kondisi On
<b>android:checked</b>	Jika true, maka kondisi On, sebaliknya Off

Perhatikan contoh penggunaan ToggleButton berikut:

```
<?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">
```



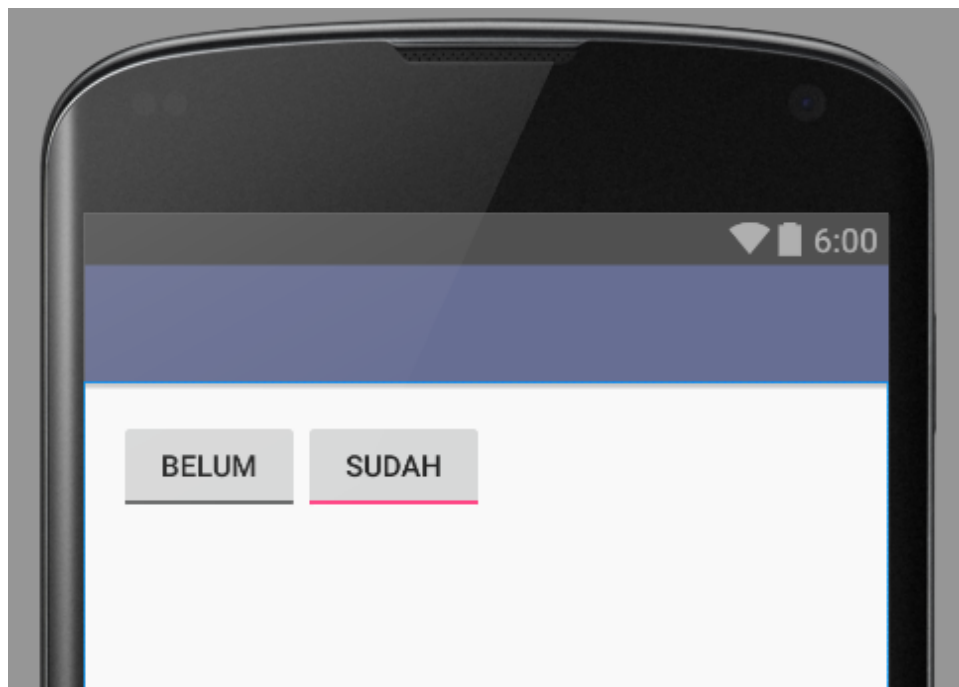
```

<ToggleButton
    android:id="@+id/toggleButton1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOff="Belum"
    android:textOn="Sudah"/>

<ToggleButton
    android:id="@+id/toggleButton2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOff="Belum"
    android:textOn="Sudah"
    android:checked="true"/>
</LinearLayout>

```

Hasil dari kode xml di atas adalah seperti pada gambar di bawah ini:



Untuk mengakses ToggleButton melalui kode java dan mengambil kondisinya, gunakan fungsi isChecked. Perhatikan contoh berikut:

```

ToggleButton toggleButton = (ToggleButton) findViewById(R.id.toggleButton1);
boolean isChecked = toggleButton.isChecked();

```

### 6.11. RadioButton

RadioButton merupakan view yang memiliki dua kondisi, dalam kondisi tercentang atau tidak. Berbeda dengan CheckBox, RadioButton tidak memiliki sifat toggle, artinya user

hanya bisa melakukan aksi centang, tanpa bisa mengembalikan ke kondisi semula. Berikut adalah beberapa atribut RadioButton yang penting:

Atribut	Keterangan
<b>android:id</b>	Mendefinisikan ID unik dari RadioButton
<b>android:text</b>	Teks yang akan ditampilkan
<b>android:textColor</b>	Warna teks dengan format #rgb
<b>android:checked</b>	Jika true, maka kondisi tercentang, sebaliknya tidak tercentang

Perhatikan contoh penggunaan RadioButton berikut:

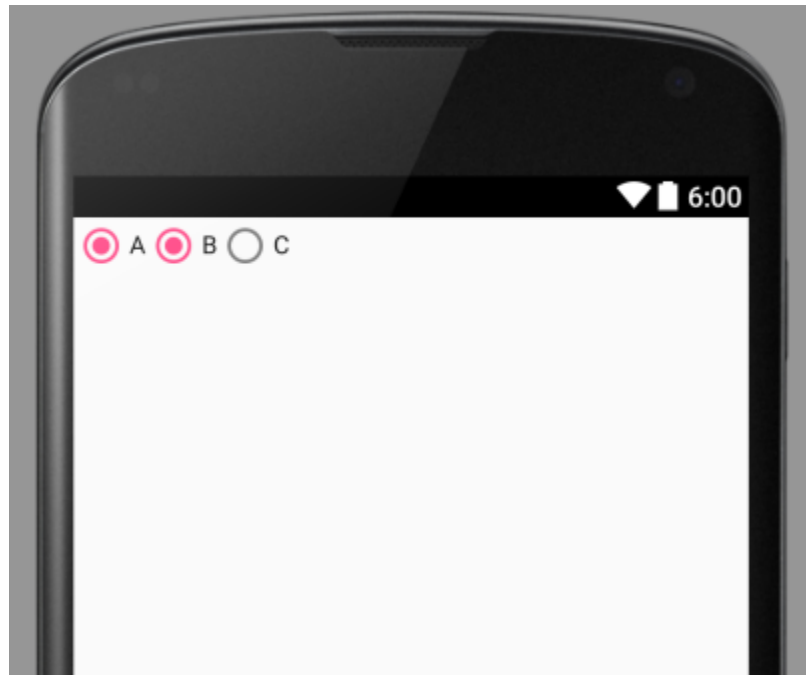
```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="A"
        android:checked="true"
        android:id="@+id/radioButton1" />

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="B"
        android:checked="true"
        android:id="@+id/radioButton2" />

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="C"
        android:id="@+id/radioButton3" />
</LinearLayout>
```

Berikut adalah tampilan dari kode xml di atas:



Untuk mengakses RadioButton melalui kode java dan mengambil kondisinya, gunakan fungsi isChecked. Perhatikan contoh berikut:

```
RadioButton radioButton = (RadioButton) findViewById(R.id.radioButton1);
boolean isChecked = radioButton.isChecked();
```

## 6.12. RadioGroup

RadioGroup merupakan view untuk meletakkan beberapa RadioButton. Bila sebuah RadioButton di dalam RadioGroup diklik, maka RadioButton yang lain akan otomatis tidak tercentang. Berikut adalah beberapa atribut RadioButton yang penting:

Atribut	Keterangan
<b>android:id</b>	Mendefinisikan ID unik dari RadioGroup
<b>android:checkedButton</b>	ID dari RadioButton yang tercentang

Perhatikan contoh penggunaan RadioGroup berikut:

```
<?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="horizontal">

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
```

```
<RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="A"
    android:checked="true"
    android:id="@+id/radioButton1" />

<RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="B"
    android:checked="true"
    android:id="@+id/radioButton2" />

<RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="C"
    android:id="@+id/radioButton3" />
</RadioGroup>
</LinearLayout>
```

Hasil dari layout di atas adalah seperti pada tampilan di bawah ini:



Untuk mengakses RadioGroup melalui kode java dan mengambil nilai yang tercentang gunakan fungsi `getCheckedRadioButtonId`. Perhatikan contoh potongan kode java berikut:

```
RadioGroup radioGroup = (RadioGroup) findViewById(R.id.radioGroup);  
int selectedRadioButtonId = radioGroup.getCheckedRadioButtonId();
```

### 6.13. ProgressBar

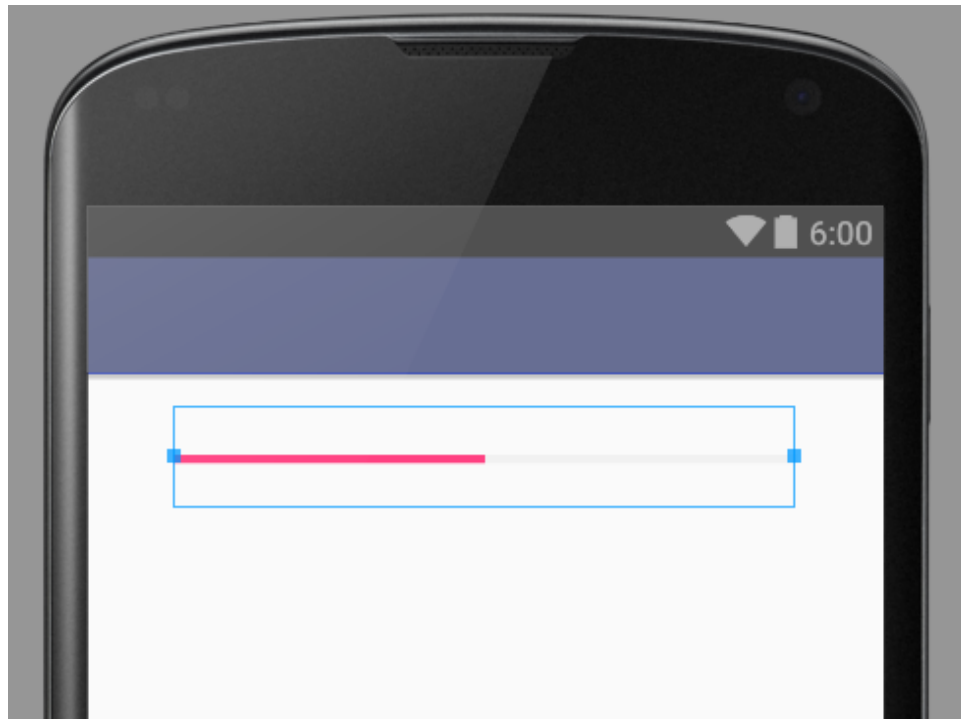
ProgressBar merupakan view yang berfungsi untuk menampilkan visualisasi kemajuan (*progress*) dari sebuah proses. Umumnya proses-proses yang perlu untuk ditampilkan progres-nya adalah proses-proses yang membutuhkan waktu yang tidak singkat. Misalnya, proses upload/download. Berikut adalah beberapa atribut ProgressBar yang perlu anda ketahui:

Atribut	Keterangan
<code>android:id</code>	Mendefinisikan ID unik dari ProgressBar
<code>android:progress</code>	Nilai progress saat ini
<code>android:max</code>	Nilai maksimum progress

Perhatikan contoh penggunaan ProgressBar di bawah ini:

```
<?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">  
  
    <ProgressBar  
        android:id="@+id/progressBar"  
        style="?android:attr/progressBarStyleHorizontal"  
        android:layout_width="300dp"  
        android:layout_height="50dp"  
        android:progress="50"  
        android:max="100"  
        android:layout_gravity="center_horizontal" />  
</LinearLayout>
```

Layout di atas menghasilkan tampilan seperti di bawah ini:



Untuk mengakses ProgressBar melalui kode java dan mengatur nilai progress gunakan metode `setProgress`. Perhatikan contoh berikut:

```
ProgressBar progressBar = (ProgressBar) findViewById(R.id.progressBar);  
progressBar.setProgress(50);
```

## 7. Layout

Pada bab sebelumnya telah dibahas sedikit tentang pengertian viewgroup. Sebuah viewgroup bisa memiliki beberapa view lain sebagai anak-anaknya. Tujuan utama kenapa beberapa view ditempatkan dalam sebuah viewgroup adalah agar memudahkan dalam pengaturan tata letak dan penyelarasan (*alignment*) dari view-view tersebut. Teknik pengaturan tersebut disebut layout.

### 7.1. Atribut Layout

Ketika sebuah *view* diletakkan dalam sebuah *viewgroup*, maka ada beberapa atribut yang mempengaruhi karakteristik layout mereka, diantaranya:

1. *gravity* dan *layout\_gravity*

Atribut *gravity* dan *layout\_gravity* adalah dua atribut yang mengatur penyelarasan (*alignment*) view dalam sebuah layout. Kedua atribut tersebut hanya berpengaruh bila digunakan dalam *LinearLayout* dan *RelativeLayout*. Atribut *gravity* merupakan penyelarasan konten view terhadap batas view. Sedangkan *layout\_gravity* merupakan penyelarasan view terhadap *parent*-nya.

Kedua atribut tersebut dapat anda isi dengan satu atau gabungan dari nilai berikut:

Nilai	Keterangan
<b>top</b>	Sejajar atas
<b>bottom</b>	Sejajar bawah
<b>left</b>	Sejajar kiri
<b>right</b>	Sejajar kanan
<b>center</b>	Sejajar tengah
<b>center_vertical</b>	Sejajar vertikal
<b>center_horizontal</b>	Sejajar horizontal

Perhatikan contoh potongan kode xml berikut:

```
<?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" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#e3e2ad"
        android:orientation="vertical" >
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:textSize="24sp"
    android:text="gravity=" />
```

```
<TextView
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:background="#bcf5b1"
    android:gravity="left"
    android:text="left" />
```

```
<TextView
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:background="#aacaff"
    android:gravity="center_horizontal"
    android:text="center_horizontal" />
```

```
<TextView
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:background="#bcf5b1"
    android:gravity="right"
    android:text="right" />
```

```
<TextView
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:background="#aacaff"
    android:gravity="center"
    android:text="center" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:background="#d6c6cd"
    android:orientation="vertical" >
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:textSize="24sp"
```



```

        android:text="layout_gravity=" />
<TextView
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:layout_gravity="left"
    android:background="#bcf5b1"
    android:text="left" />

<TextView
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:layout_gravity="center_horizontal"
    android:background="#aacaff"
    android:text="center_horizontal" />

<TextView
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:layout_gravity="right"
    android:background="#bcf5b1"
    android:text="right" />

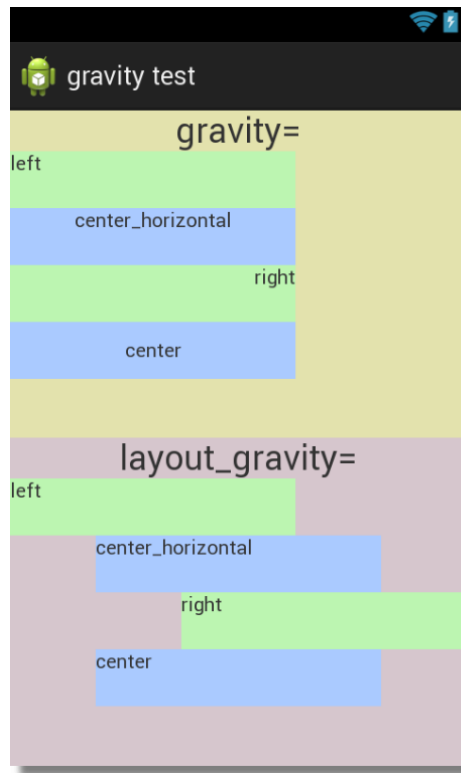
<TextView
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:layout_gravity="center"
    android:background="#aacaff"
    android:text="center" />

</LinearLayout>

</LinearLayout>

```

Pratinjau dari kode xml di atas adalah sebagai berikut:



## 2. *layout\_weight*

Atribut *layout\_weight* merupakan pembobotan terhadap lebar atau tinggi dari sebuah view. Atribut tersebut hanya berpengaruh ketika parent dari view adalah *LinearLayout*. Atribut ini akan dibahas lebih lanjut pada sub bab pembahasan *LinearLayout*.

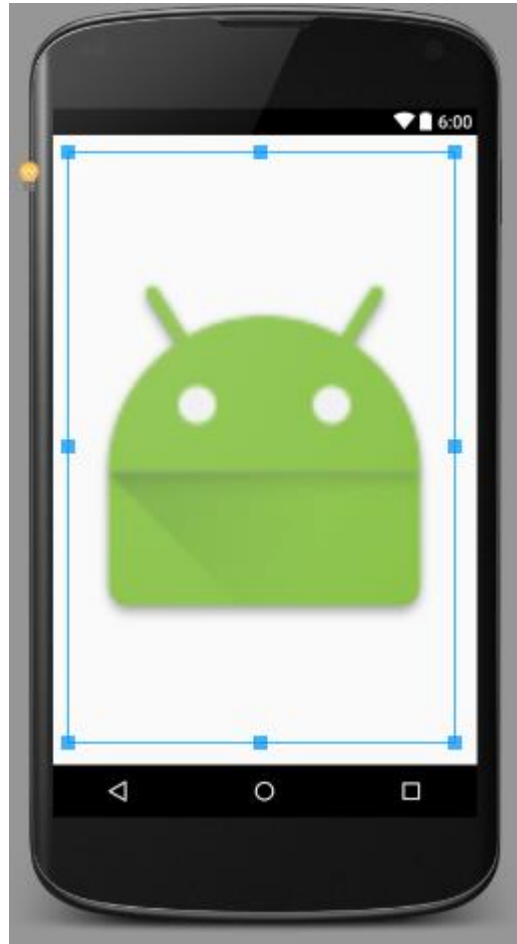
## 7.2. Frame Layout

Frame layout merupakan viewgroup yang hanya dapat menampilkan satu view dalam satu saat. Jika anda menambahkan beberapa view ke dalam frame layout maka view-view tersebut akan saling tindih satu sama lain. Frame layout tidak memiliki mekanisme internal untuk mengatur view-view di dalamnya mengikuti suatu aturan tertentu. Dalam prakteknya, frame layout banyak digunakan ketika hanya ada satu view sebagai anaknya. Perhatikan contoh kode xml berikut:

```
<?xml version="1.0" encoding="utf-8" ?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin">
```

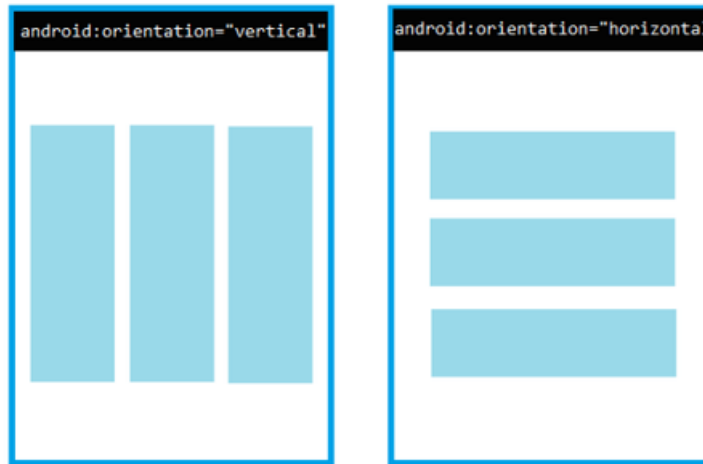
```
<ImageView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:src="@mipmap/ic_launcher"/>  
</FrameLayout>
```

Pratinjau dari kode xml tersebut adalah sebagaimana terlihat dalam gambar di bawah ini:



### 7.3. LinearLayout

LinearLayout merupakan viewgroup yang dapat meletakkan view-view di dalamnya dalam posisi bertumpuk. Posisi tumpukan bisa berarah vertikal maupun horizontal. Arah posisi tersebut ditentukan oleh sebuah atribut yaitu **android:orientation**. Nilai dari atribut tersebut adalah **vertical** atau **horizontal**. Gambar di bawah mengilustrasikan posisi view anak pada arah horizontal dan vertikal.



View anak dari sebuah LinearLayout dapat mengandung bobot. Bobot tersebut dapat berarti seberapa lebar atau tinggi view anak di dalam layout. Bobot tersebut dapat diatur menggunakan atribut **android:layout\_weight**. Semakin besar nilai bobot tersebut semakin besar pula lebar atau tinggi dari view anak memenuhi sisa ruang dari *parent*.

Perhatikan kode xml berikut

```
<?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">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button 1"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="5"
        android:text="Button 2"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="3"
        android:text="Button 3"/>
</LinearLayout>
```

Kode xml di atas mendefinisikan 3 buah button di dalam sebuah LinearLayout dengan orientasi horizontal. Ketiga button tersebut masing-masing memiliki nilai **android:layout\_weight** sebesar 1, 5 dan 3. Hasil dari kode xml di atas adalah seperti pada gambar di bawah:



#### 7.4. RelativeLayout

RelativeLayout merupakan viewgroup yang menampilkan beberapa view anak yang berada di dalamnya dalam posisi relatif. Relatif dalam hal ini bermakna relatif terhadap *parent* ataupun relatif terhadap view yang lain dalam satu *parent* (*sibling*). Berikut adalah beberapa atribut yang penting untuk dipahami ketika bekerja menggunakan RelativeLayout

Atribut	Keterangan
<a href="#">layout_below</a>	Memposisikan view di bawah view dengan ID yang diacu
<a href="#">layout_above</a>	Memposisikan view di atas view dengan ID yang diacu
<a href="#">layout_toLeftOf</a>	Memposisikan view di kiri view dengan ID yang diacu
<a href="#">layout_toRightOf</a>	Memposisikan view di kanan view dengan ID yang diacu
<a href="#">layout_alignParentTop</a>	Jika <b>true</b> , posisikan view sejajar dengan batas atas <i>parent</i>
<a href="#">layout_alignParentRight</a>	Jika <b>true</b> , posisikan view sejajar dengan batas kanan <i>parent</i>
<a href="#">layout_alignParentBottom</a>	Jika <b>true</b> , posisikan view sejajar dengan batas bawah <i>parent</i>
<a href="#">layout_alignParentLeft</a>	Jika <b>true</b> , posisikan view sejajar dengan batas kiri <i>parent</i>

<b>layout_centerInParent</b>	Jika <b>true</b> , posisikan view di tengah-tengah <i>parent</i>
<b>layout_centerHorizontal</b>	Jika <b>true</b> , posisikan view di tengah <i>parent</i> secara horizontal
<b>layout_centerVertical</b>	Jika <b>true</b> , posisikan view di tengah <i>parent</i> secara vertikal

Kode xml di bawah ini menggunakan beberapa atribut di atas dalam sebuah RelativeLayout:

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:text="Button 1"/>
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:text="Button 2"/>
    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentBottom="true"
        android:text="Button 3"/>
    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:text="Button 4"/>
    <Button
        android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Button 5"/>

</RelativeLayout>

```

Hasil pratinjau dari kode xml di atas menghasilkan layout seperti pada gambar di bawah ini:





## 8. Dialog

Dalam sebuah aplikasi, Dialog banyak digunakan untuk menyampaikan pesan kepada user. Misal pesan bahwa proses download telah selesai. Dengan adanya dialog, aplikasi yang anda buat akan lebih user friendly.

Di android ada beberapa jenis dialog yang dapat anda gunakan.

### 8.1. Toast

Toast merupakan sebuah popup sederhana yang dapat digunakan untuk menampilkan pesan. Toast biasa dipergunakan untuk memberi notifikasi kepada user bahwa sebuah proses telah selesai. Misal menghapus item, mengubah judul, dll sesuai dengan keperluan. Toast akan otomatis hilang setelah 2 atau 3 detik.

Untuk menampilkan Toast, gunakan kode berikut dalam sebuah Activity:

### 8.2. Snack Bar

### 8.3. Alert Dialog

### 8.4. Progress Dialog

### 8.5. Activity Sebagai Dialog

## **9. Menu**

9.1. Menu Sebagai Action

9.2. Popup Menu

9.3. Context Menu

## **10. Shared Preferences**

10.1. Menyimpan Data

10.2. Mengambil Data

10.3. Membuat Kelas Helper

## 11. SQLite

- 11.1. SQLiteHelper
- 11.2. Melakukan Query Data
- 11.3. Menambah Data
- 11.4. Mengupdate Data
- 11.5. Menghapus Data

## **12. List View**

12.1. Apa itu List View?

12.2. Default Adapter