

# MATERI 1

## PENDAHULUAN

- Alasan Membangun Aplikasi Android
  - Dasar Pemrograman Android
  - Fitur Perangkat Keras Android
  - Fitur Perangkat Lunak Android
- 

### 1.1. Alasan Membangun Aplikasi Android

Terdapat beberapa alasan mengapa Android berkembang dengan pesat dan kenapa *developer* sebaiknya membangun Aplikasi Android.

#### a. *Market Share*

*Developer* memiliki kesempatan untuk mengembangkan aplikasi untuk pasar yang relatif baru dan sedang berkembang pesat. Keberadaan *Android Market* akan menempatkan aplikasi yang dibuat oleh *developer* langsung kepada pengguna. Pengguna tidak perlu mencari melalui internet untuk menemukan dan meng-*install* aplikasi yang dibuat oleh *developer*. Pengguna cukup membuka *Android Market* yang sudah ter-install di perangkat android pengguna dan mengakses aplikasi yang pengguna butuhkan dan meng-*install* ke dalam perangkat android pengguna.

#### b. *Time to Market*

Terdapat Android APIs (*Application Programming Interfaces*) yang memudahkan *developer* untuk membangun aplikasi dengan mudah dan waktu yang singkat.

#### c. *Open Platform*

Sistem operasi Android merupakan platform terbuka. Sehingga tidak merujuk pada suatu perusahaan *hardware* atau suatu *provider*. Platform terbuka android memungkinkan perkembangan *market* dengan sangat cepat, karena semua perusahaan *hardware* dan *provider* dapat membuat dan menjual perangkat android. *Source code* android dapat di akses melalui at <http://source.android.com>, untuk digunakan dan dimodifikasi sesuai kebutuhan perusahaan *hardware*, *provider* atau *developer* aplikasi.

#### d. *Cross Compatibility*

Android dapat berjalan di berbagai perangkat dengan ukuran dan resolusi layar yang berbeda. Android memiliki fitur yang membantu *developer* untuk mengembangkan aplikasi yang *compatible* untuk berbagai perangkat (*cross-compatible application*). Google memiliki *feature detection* yang mengatur agar aplikasi yang dikembangkan oleh *developer* hanya berjalan di perangkat yang *compatible*. Sebagai contoh, jika suatu aplikasi

membutuhkan kamera depan (*front-facing camera*) maka hanya perangkat android dengan kamera depan yang akan dapat melihat aplikasi tersebut di *Android Market*.

Suatu perangkat android harus mengikuti suatu aturan untuk mendapatkan sertifikat kesesuaian terhadap *Android Market* untuk memastikan aplikasi dapat berjalan diatasnya, diantaranya keberadaan: (a) Kamera, (b) Kompas, (c) GPS (*Global Positioning System*), (d) *Bluetooth Transceiver*.

e. *Mashup Capability*

*Mashup Capability* merupakan kemampuan untuk mengkombinasikan dua atau lebih layanan untuk mengembangkan suatu aplikasi. Sebagai contoh, *developer* dapat membuat *mashup* dengan menggunakan kamera dan GPS, sehingga terbangun aplikasi pengambilan photo dengan dilengkapi secara tepat lokasi pengambilannya.

Beberapa contoh lain *mashup* yang mungkin dikembangkan dalam suatu aplikasi, yaitu:

1. *Geolocation dan Social Networking*

Suatu aplikasi yang secara otomatis melakukan *tweet* posisi pengguna setiap 10 menit.

2. *Geolocation dan Gaming*

*Location-based gaming* merupakan model game yang saat ini populer. Game memiliki *background services* untuk mengecek posisi pengguna dan membandingnya dengan posisi pengguna yang lain di area yang sama. Jika jarak antara pengguna kurang dari 1 km misal, maka akan diberikan notifikasi dan memungkinkan mereka untuk bertarung melalui game tersebut.

## 1.2. Dasar Pemrograman Android

Membangun Aplikasi Android sangat mudah, *default* untuk bahasa pemrograman yang digunakan oleh Android adalah java. Tidak semua fitur java yang ada digunakan akan tetapi sebagian dari java yang sering disebut dengan *Dalvik Virtual Machine*. Beberapa bagian kecil dari *framework* Android menggunakan bahasa XML untuk *scripting*-nya.

a. *Activity*

*Activity* merupakan *container* untuk *User Interface (UI)*. Sebuah Aplikasi Android terbangun dari satu atau beberapa *Activity*.

b. *Intents*

*Intent* merupakan sistem pesan utama yang menjalankan Android. *Intent* terdiri dari *Action* yang harus dijalankan (Tampil, Ubah, Dial, dll) dan Data.

*Intent* digunakan untuk memulai aktivitas dan komunikasi antar bagian dari sistem Android. Suatu aplikasi dapat mengirimkan atau menerima *intent*.

1. Mengirimkan pesan dengan *Intent*

Ketika suatu aplikasi mengirimkan suatu *intent*, aplikasi mengirimkan pesan memberitahukan Android untuk menjalankan sesuatu. Misal: memberitahukan Android untuk menjalankan suatu *Activity* dari aplikasi atau menjalankan aplikasi lain.

## 2. Mendaftarkan *Intent Receiver*

Mengirimkan pesan dengan *Intent* bukan berarti sesuatu selanjutnya akan terjadi secara otomatis. Aplikasi harus mendaftarkan *intent receiver* yang memperhatikan *intent* dan memberitahukan Android apa yang harus dilakukan.

Contoh, jika pengguna menekan lama suatu *image* di *image gallery*, maka akan memunculkan *context menu* yang berkaitan dengan *image sharing*. Dikarenakan terdapat beberapa *receivers* yang didaftarkan berkaitan dengan image sharing (*email, messaging, Bluetooth, dll*) maka context menu dimunculkan untuk memberikan opsi keputusan kepada pengguna.

### c. *Cursorless Controls*

Perangkat Android menggunakan jari pengguna sebagai *input*. Jika pada computer biasa digunakan pointer mouse, kemudian *left-click* untuk berinteraksi dengan suatu tombol misal, maka pengguna Android dapat menggunakan jarinya langsung ke tombol. Untuk menggantikan *right-click*, pengguna Android dapat menekan jarinya agak lama ke layar, kemudian *context menu* akan muncul.

### d. *Views dan Widgets*

*View* merupakan elemen dasar UI (*User Interface*). Sebagai contoh area kotak pada layar yang bertanggung jawab untuk tampilan dan menerima *event* (*event handling*). Beberapa contoh view antara lain: (1) ContextMenu, (2) Menu, (3) View, (4) SurfaceView .

*Widgets* adalah elemen UI yang lebih canggih. Merupakan control untuk interaksi antara sistem dengan pengguna. Beberapa contoh *Widgets* antara lain: (1) Button, (2) CheckBox, (3) DatePicker, (4) DigitalClock, (5) Gallery, (6) FrameLayout, (7) ImageView, (8) RelativeLayout, (9) PopupWindow.

Untuk mempelajari *widgets* lebih lanjut dapat dilihat melalui url:

<http://developer.android.com/reference/android/widget/package-summary.html>

### e. *Asynchronous Calls*

Android memiliki sebuah class yang *AsyncTask* yang memungkinkan aplikasi menjalankan beberapa operasi pada waktu yang bersamaan, tanpa harus mengatur bagaimana *thread* berjalan secara khusus. *AsyncTask* memungkinkan *developer* untuk membangun model program yang bersih untuk proses-proses asinkron.

Proses asinkron biasa digunakan untuk proses-proses yang membutuhkan waktu lama, misal: *Network Communication (Internet)*, *Media Processing*, dan berbagai proses lain yang mengharuskan pengguna menunggu. Jika pengguna harus menunggu, maka dapat digunakan proses asinkron untuk menampilkan UI yang memberitahukan pengguna apa yang terjadi.

### f. *Background Services*

*Services* merupakan aplikasi yang berjalan di belakang dan tidak terlalu penting memiliki UI, sebagai contoh: anti-virus. Sebagian besar pemutar musik (*music player*) dari Android

Market berjalan sebagai *Background Services*, yangmana pengguna dapat mendengarkan musik sambil mengecek e-mail atau melakukan tugas lain yang membutuhkan penggunaan tampilan layar.

### 1.3. Fitur Perangkat Keras Android

Perangkat Android memiliki beberapa fitur perangkat keras didalamnya, yang dapat dimanfaatkan *developer* dalam membangun aplikasi.

#### a. Touchscreen

Perangkat Android memiliki fitur layar sentuh (*touchscreen*) yang memberikan beberapa kemungkinan bagi pengguna untuk berinteraksi dengan aplikasi dengan menggunakan jari. Pengguna dapat melakukan *swipe*, *flip*, *drag*, dan *pinch* untuk *zoom*. Android juga mendukung *multitouch* yang berarti keseluruhan layar dapat disentuh dengan satu atau lebih jari pada saat yang bersamaan.

#### b. GPS

Sistem operasi Android mendukung GPS yang memungkinkan *developer* untuk mengakses lokasi pengguna. Contoh aplikasi yang memanfaatkan GPS adalah Aplikasi Peta (*Map*) yang menunjukkan lokasi pengguna dan memberikan petunjuk untuk menuju suatu lokasi.

#### c. Accelerometer

Android mendukung *Accelerometer*, yaitu perangkat yang digunakan untuk mengukur percepatan. *Accelerometer* dapat memberitahukan apabila suatu Perangkat Android bergerak, atau terguncang, atau berbalik arah posisinya.

#### d. SD Card

Android memiliki fitur yang memungkinkan pengguna atau aplikasi untuk mengakses (menyimpan atau membuka) file pada SD Card. SD Card merupakan media penyimpanan medium yang digunakan Perangkat Android dan beberapa perangkat *mobile* lain non Android sebagai media penyimpanan.

### 1.4. Fitur Perangkat Lunak Android

Android memiliki banyak fitur perangkat lunak yang dapat digunakan oleh *developer* dalam mengembangkan aplikasi. Beberapa fitur populer yang akan dijabarkan disini.

#### a. Internet

Kemampuan akses internet pada Android memberikan banyak keunggulan. Berbagai informasi secara real-time dapat diperoleh dengan mudah dengan internet. Contoh, sebagai pengguna, dapat menggunakan internet untuk melihat jadwal pemutaran film bioskop, cuaca suatu area, jadwal penerbangan dan lainnya. Sebagai *developer*, dapat menggunakan internet untuk akses secara real-time kepada data, *update data*. *Developer* juga dapat menggunakan internet untuk menyimpan berbagai asset untuk kemudian

digunakan suatu aplikasi, seperti dilakukan Pandora dan YouTube. Dengan internet dapat dibangun model aplikasi yang disebut *client-server computing*. Contoh lain, aplikasi peta, mengakses data peta dan GPS dari web server.

b. *Audio dan Video Support*

Sistem operasi Android memungkinkan developer menyertakan audio dan video dalam aplikasi dengan mudah. Berbagai standar format audio dan video didukung.

c. *Contact*

Android memungkinkan akses ke *contacts* yang tersimpan di Perangkat Android. *Developer* dapat menggunakan fitur ini untuk menampilkan *contacts* dalam cara baru yang berbeda. Hal lain yang dapat dilakukan adalah membangun aplikasi yang menggabungkan antara *contacts* dengan GPS, yang memberikan notifikasi kepada pengguna jika pengguna berada di dekat alamat satu *contact* yang ada.

d. *Security*

Android memungkinkan aplikasi untuk melakukan banyak hal. Akan tetapi Android juga menyiapkan mekanisme keamanan berupa *permission* berkaitan dengan beberapa tugas. Contoh: Download *image* dan menyimpannya di SD Card, maka harus disetujui terlebih dahulu *permission* untuk mengakses SD Card.

e. Google APIs

Sistem operasi Android memungkinkan dengan tidak terbatas membuat panggilan telepon, mengorganisasi *contacts* atau meng-*install* aplikasi. *Developer* dapat mengintegrasikan peta (*map*) ke dalam suatu aplikasi dengan menggunakan Maps API yang mengandung Map Widgets. Berbagai fitur dapat ditambahkan dengan Maps API, antara lain: (1) Menampilkan suatu lokasi di peta, (2) Mendapatkan panduan navigasi, (3) Komunikasi data antara aplikasi dengan *clouds*.

### 1.5. Menjadi Developer Android

Semua perangkat lunak untuk membangun aplikasi Android bisa didapatkan dengan GRATIS. Hal tersebut salah satu keunggulan membangun aplikasi Android.

Menjadi developer Android bukan merupakan pekerjaan yang sulit, yang utama yang perlu ditanyakan terhadap diri adalah:

- 1) Apakah anda ingin membangun Aplikasi Android ?
- 2) Apakah anda suka perangkat lunak pengembang yang gratis ?
- 3) Apakah anda memiliki komputer atau laptop untuk digunakan ?

### 1.6. Framework Android

Android sepenuhnya open source, sehingga semua developer yang membutuhkan dapat mengakses untuk menggunakan dan memodifikasi kode program Android. Source code android dapat di download melalui <http://source.android.com>.

Android dibangun diatas open source linux kernel 2.6. Kernel linux tersebut dipilih karena menyediakan fitur utama untuk membangun sistem operasi Android, diantaranya:

- 4) Security Model: Linux kernel menangani keamanan antara aplikasi dan sistem
- 5) Memory Management: Linux kernel menangani manajemen memori untuk developer
- 6) Process Management: Linux kernel mengatur proses dengan baik, mengalokasikan resource untuk proses sesuai dengan kebutuhan
- 7) Network Management: Linux kernel juga menangani jaringan komunikasi
- 8) Driver Model: Perusahaan perangkat keras dapat mengembangkan drivers perangkat mereka secara mandiri untuk berjalan di atas linux



Diatas kernel linux tersebut, framework Android dibangun dengan berbagai fitur. Fitur-fitur tersebut diadopsi dari berbagai project opensource. Beberapa fitur Android framework diantaranya:

- 9) Android Run Time: Mengandung inti library java dan dalvik virtual machine
- 10) Open GL (Graphics Library): Merupakan API (Application Program Interface) yang digunakan untuk menghasil graphics 2D dan 3D, bersifat cross-language dan cross-platform
- 11) Webkit: merupakan mesin web browser opensource yang menyediakan fungsionalitas untuk menampilkan web konten
- 12) SQLite: merupakan opensource mesin database yang di rancang untuk sistem embedded
- 13) Media frameworks: library yang digunakan untuk menjalankan dan merekam audio serta video
- 14) Secure Socket Layer (SSL): library ini bertanggung jawab terhadap keamanan internet





Untuk pemanfaatan fitur-fitur dalam Android framework diatas, disediakan berbagai library dalam Application Framework yang dapat digunakan oleh developer, diantaranya:

- 15) Activity Manager: Mengelola siklus hidup activity
- 16) Telephony Manager: Menyediakan akses ke telephone service
- 17) View System: Menangani view dan layout yang membangun User Interface (UI)
- 18) Location Manager: menemukan lokasis secara geografi



### 1.7. Open Handled Alliance (OHA)

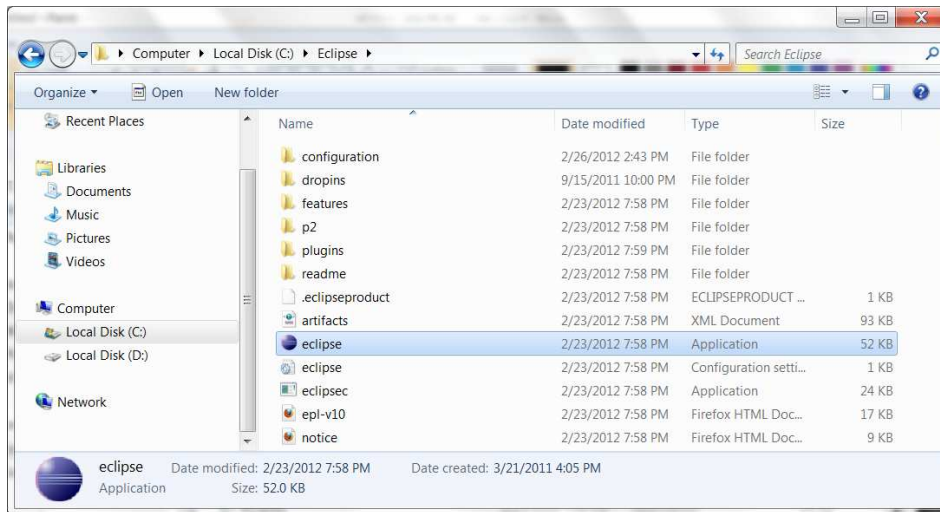
Open Handled Alliance (OHA) berdiri November 2007 dengan 34 anggota yang dimotori oleh google. Saat ini OHA telah memiliki 71 anggota, diantaranya: T-Mobile, Sprint, LG, Motorola, HTC, NVidia, and Texas Instruments

### 1.8. Perangkat Lunak Pengembangan

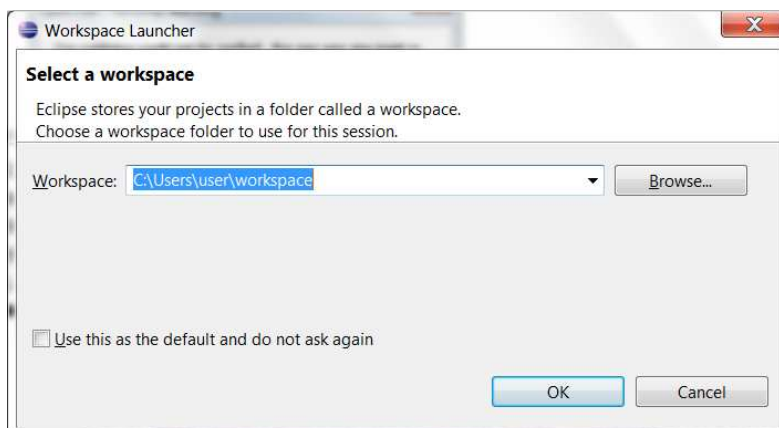
- Java JDK → <http://java.sun.com/javase/downloads/index.jsp>
- Android SDK → <http://developer.android.com/sdk/index.html>
- Eclipse IDE → [www.eclipse.org/downloads](http://www.eclipse.org/downloads)
- Android ADT → <https://dl-ssl.google.com/android/eclipse/>

## 1.9. Menginstall Eclipse

### 1. *Double-Click* pada shortcut Eclipse.exe untuk menjalankan Eclipse



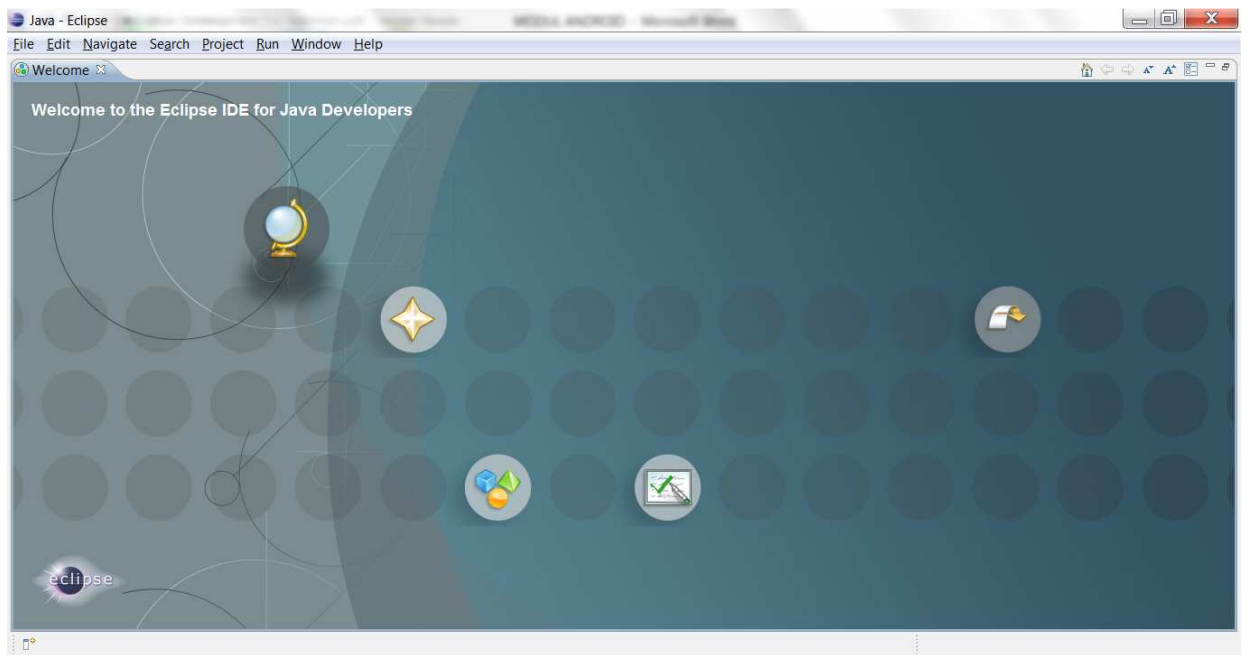
### 2. Jika dibutuhkan konfigurasi *workspace*



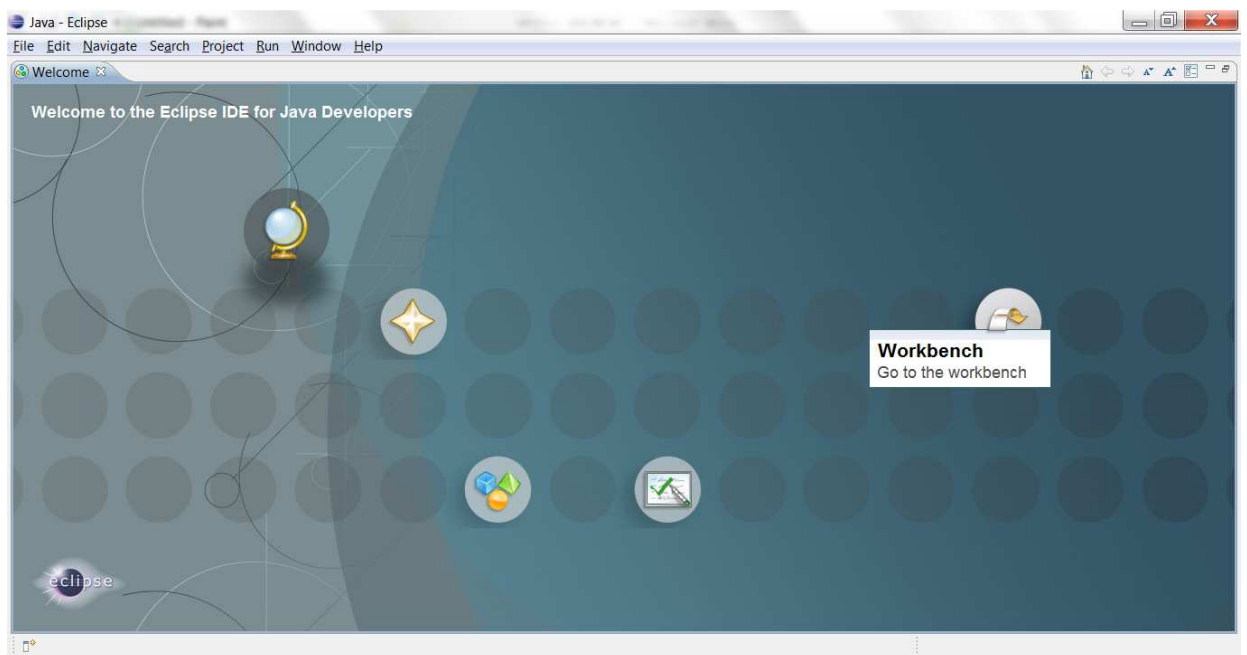
Jika anda bekerja dengan beberapa *project* dianjurkan untuk menggunakan *workspace* yang berbeda untuk masing-masing *project*.



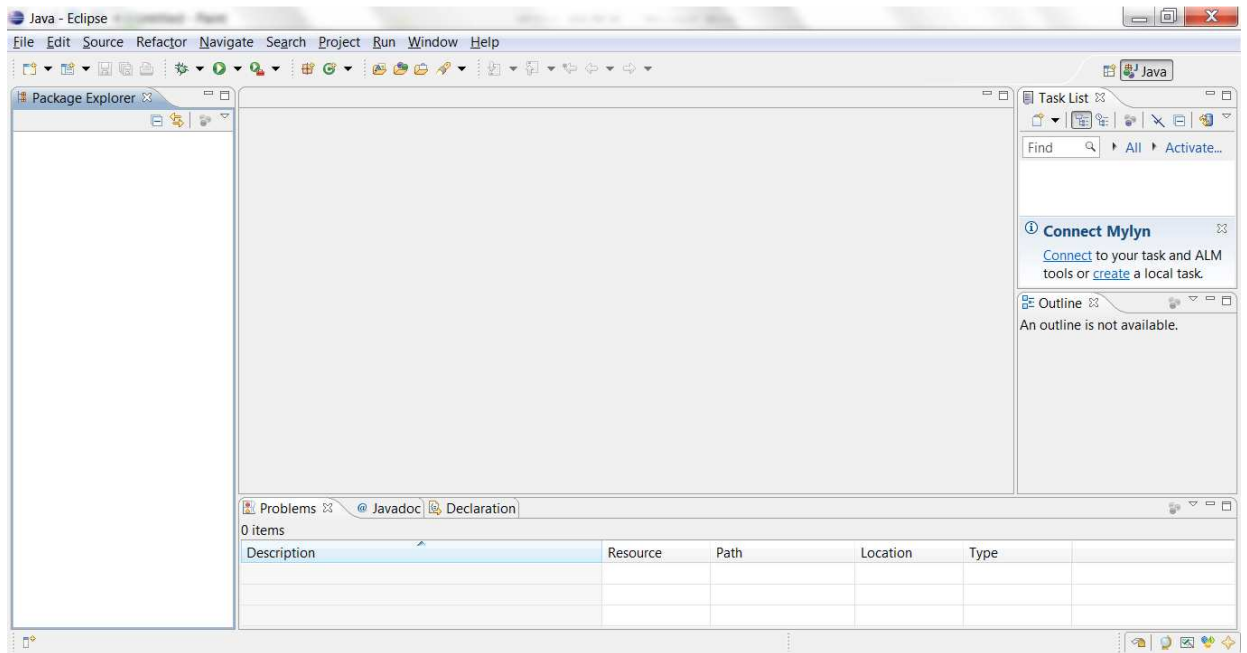
### 3. Tampilan awal Eclipse



### 4. Click pada icon untuk menuju workbench.



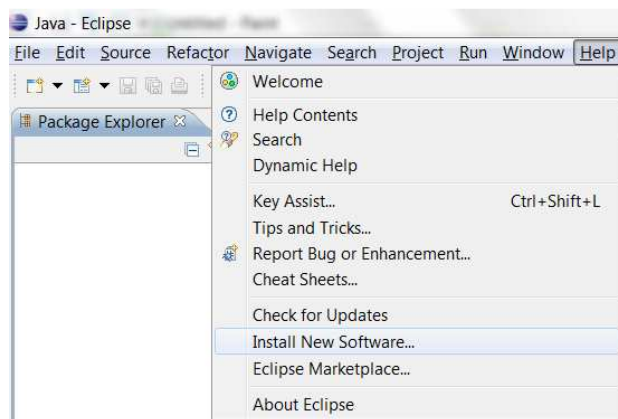
## 5. Workbench



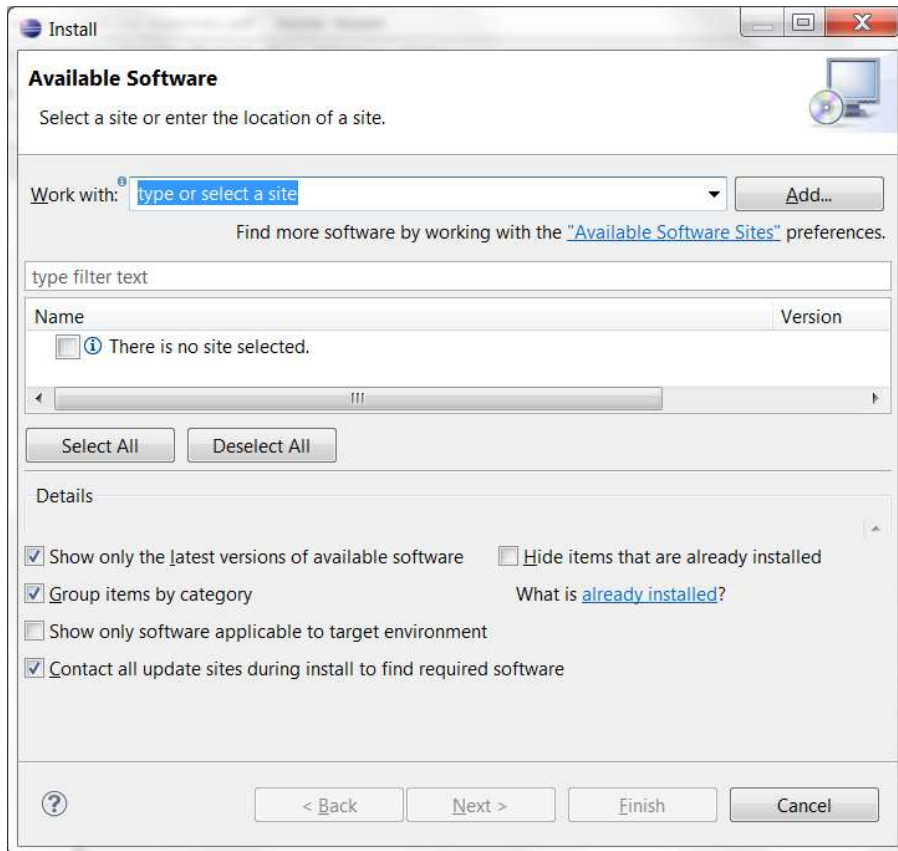
### 1.10. Mengkonfigurasi Eclipse dengan ADT

Untuk dapat menggunakan Eclipse dalam membangun Aplikasi Android. Terlebih dahulu harus dilakukan konfigurasi *Android Development Tools* (ADT) dalam Eclipse.

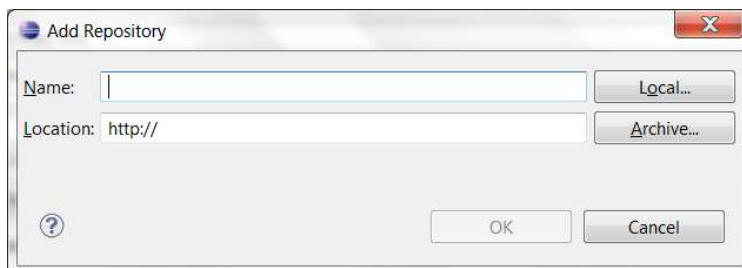
1. Jalankan Eclipse (jika belum berjalan)
2. Pilih pada menu bar → *Help* → *Install New Software*



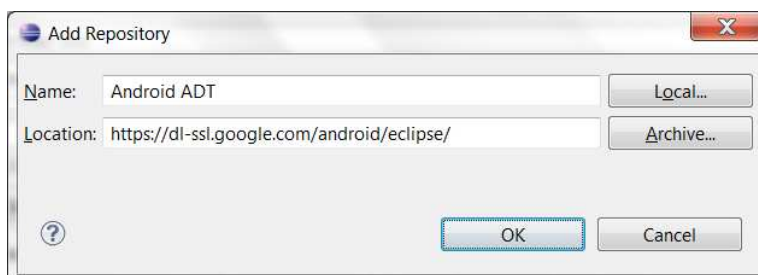
3. Muncul Install window pops-up, yang memungkinkan untuk menginstall plug-in baru ke dalam Eclipse.



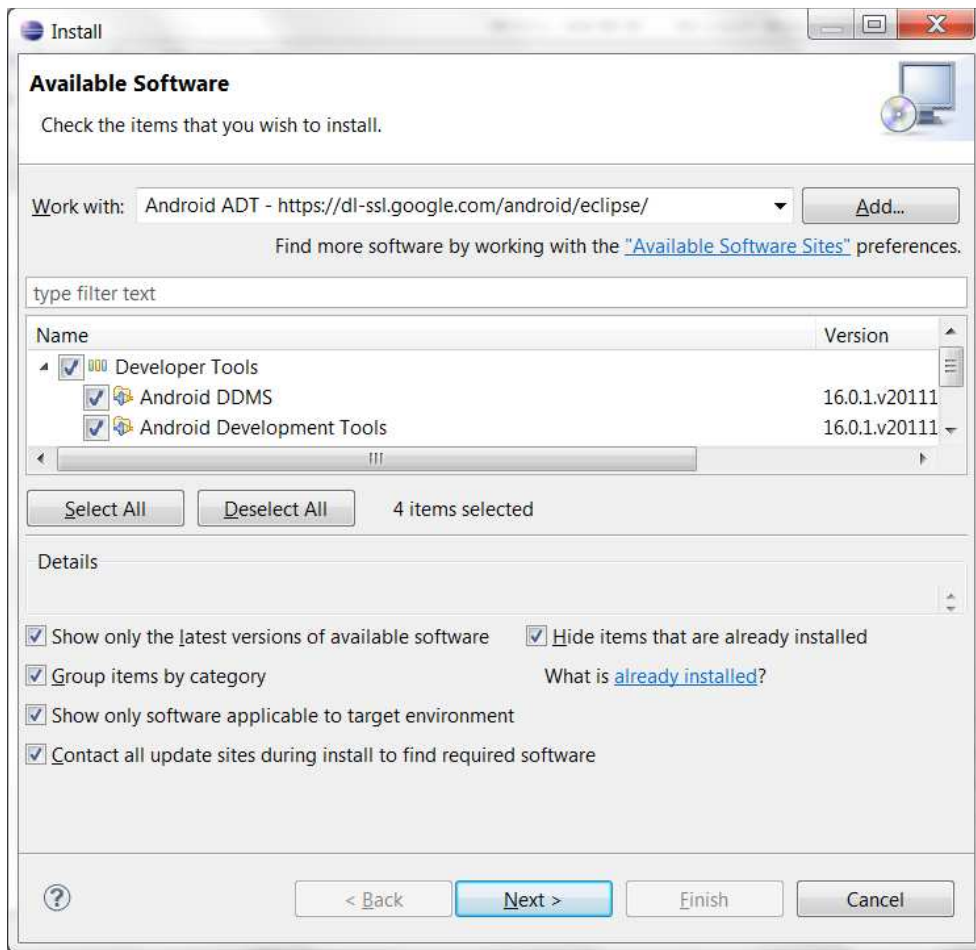
4. Klik Button Add, untuk menambahkan



5. Tuliskan nama pada Name Field, misal: **Android ADT**
6. Pada Location Field, tuliskan: <https://dl-ssl.google.com/android/eclipse/>

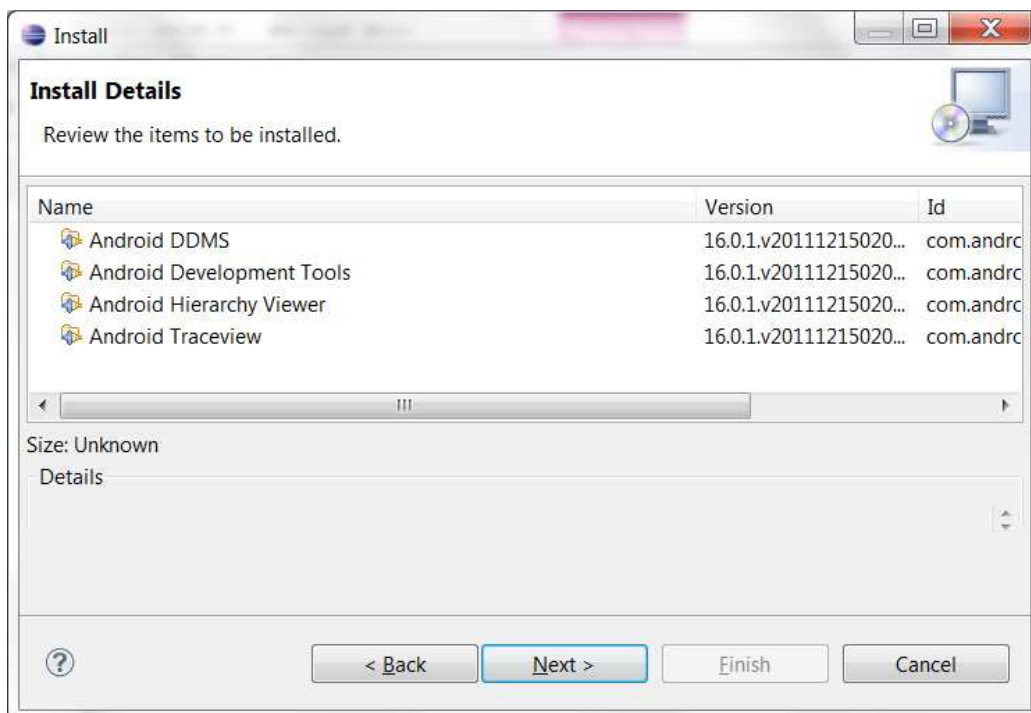


## 7. Tampil:

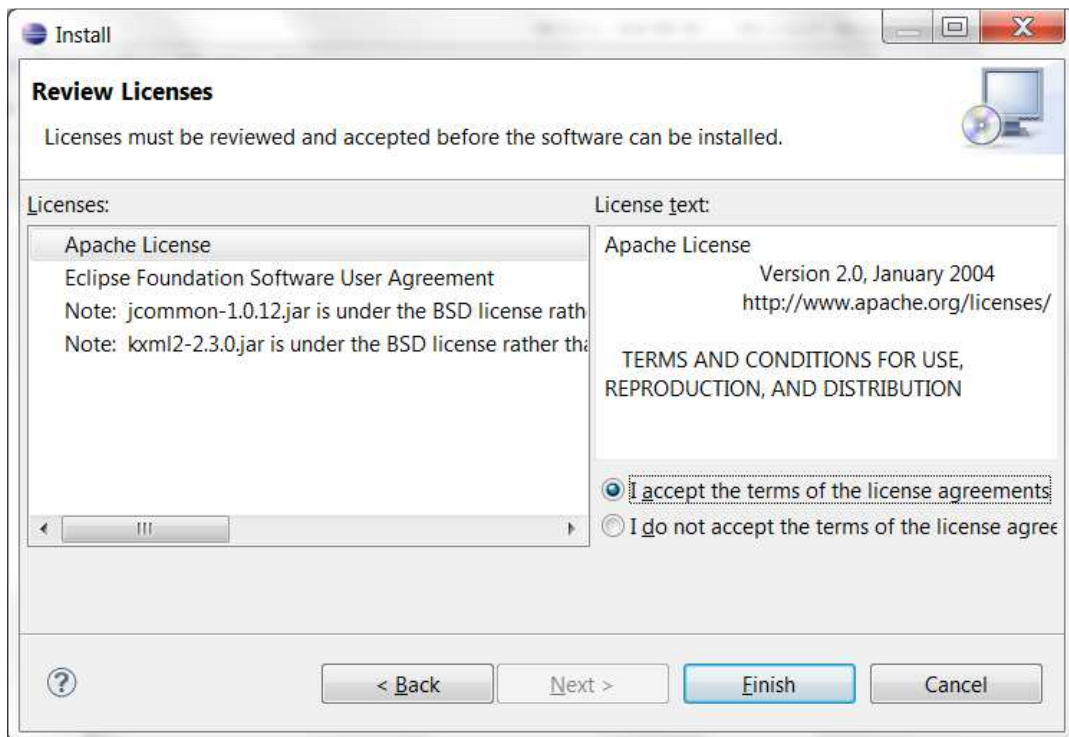


## 8. Select All, klik next> untuk install

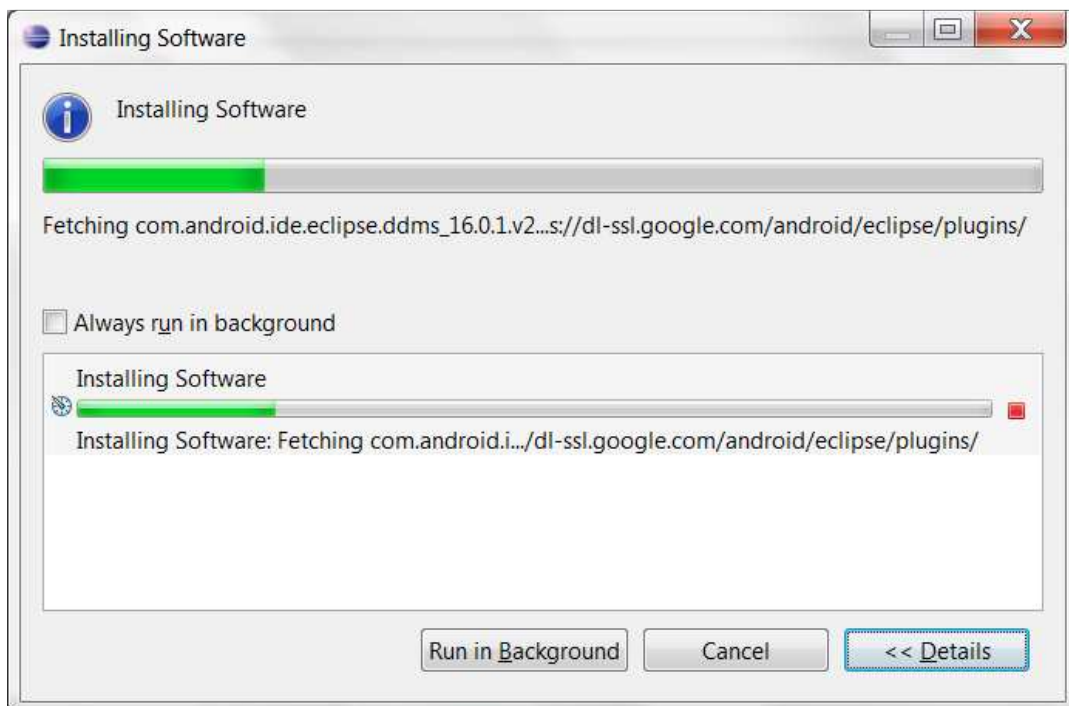
## 9. Review, klik next>



#### 10. License Agreement, Klik Finish

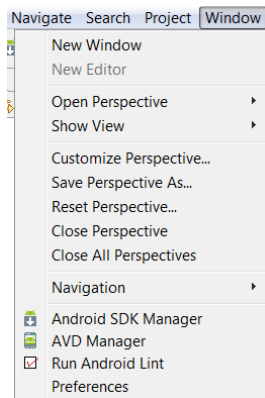


#### 11. Install dalam akan berjalan, plugin ADT akan terinstall

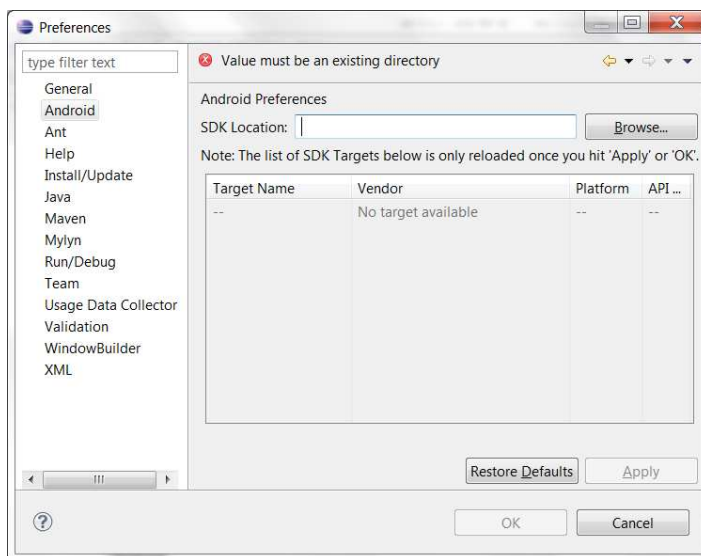


## 1.11. Mengkonfigurasi lokasi SDK

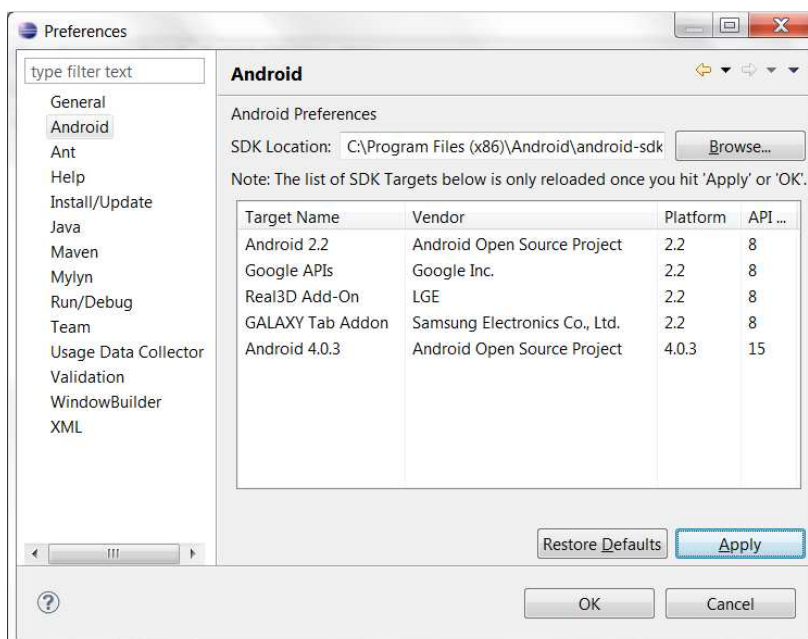
1. Pada menubar, pilih window → preferences



2. Pilih Android pada tab di kiri



3. Atur lokasi SDK, Kemudian OK





### **1.12. Pustaka**

Felker D.; Dobbs J, 2011, *Android Application Development For Dummies*, Wiley Publishing Inc., USA

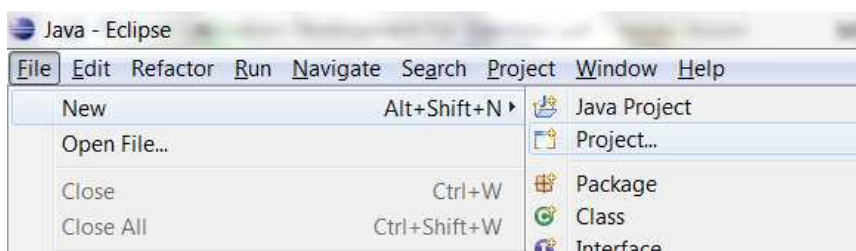
## MATERI 2

### MEMBANGUN APLIKASI ANDROID

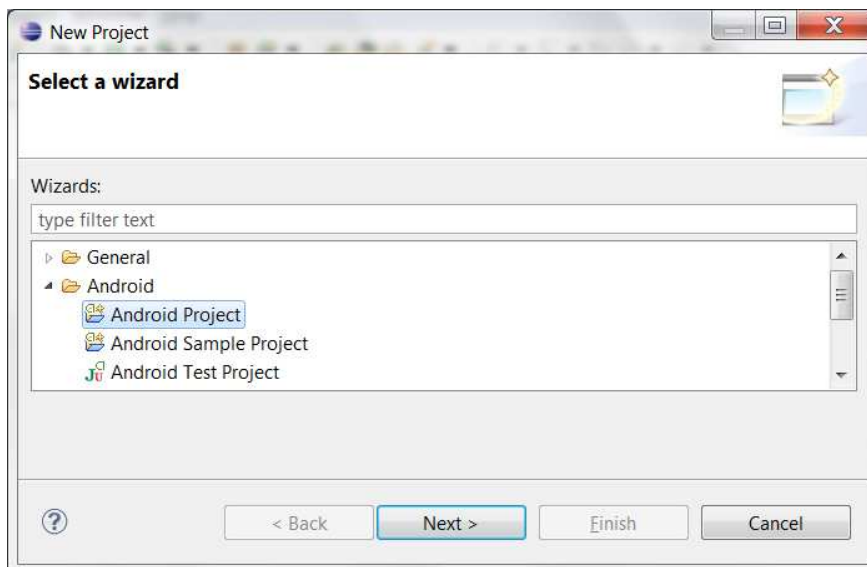
- **Membangun Project Baru di Eclipse**
  - **Mengkonfigurasi Emulator**
  - **Konfigurasi dan Menjalankan Aplikasi**
  - **Memahami Anatomi dari Project**
- 

#### 2.1. Membangun Project Baru di Eclipse

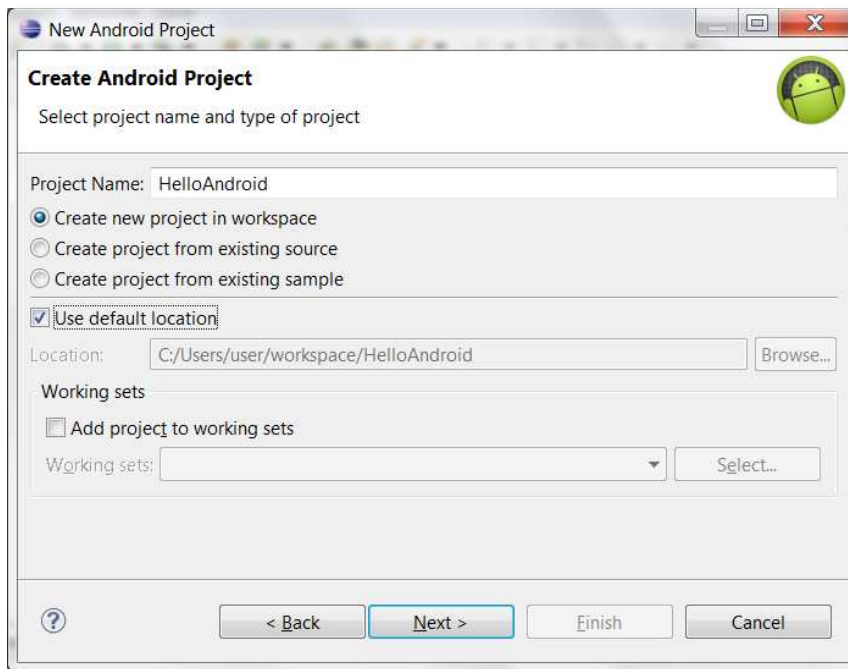
1. Dalam Eclipse, Pilih File → New → Project



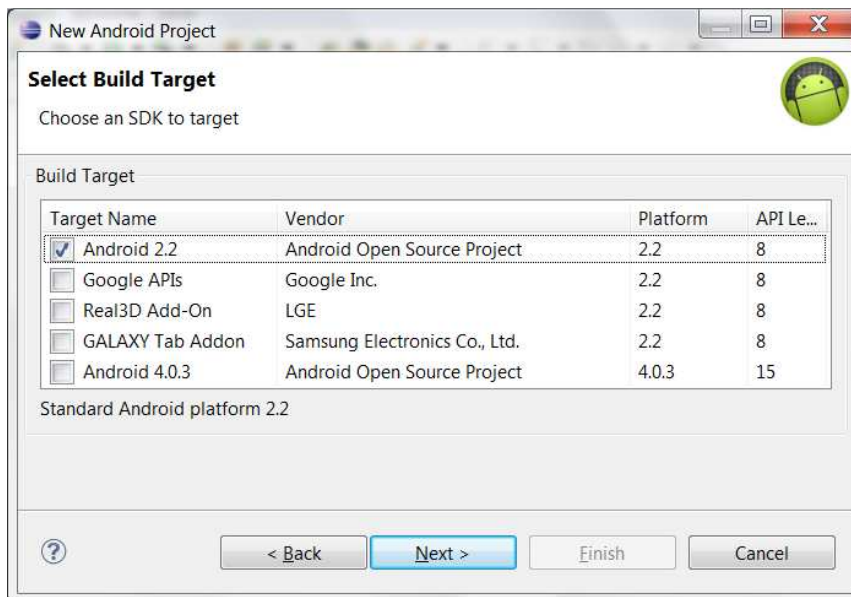
2. Pilih Android → Android Project → Next



3. Pada Project Name, tulis: **HelloAndroid**

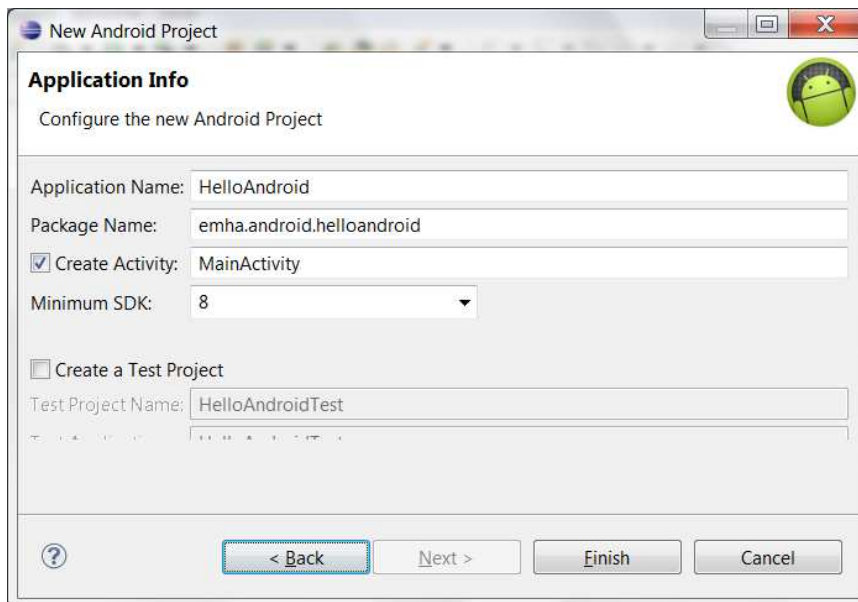


4. Untuk Build Target → Pilih Android 2.2

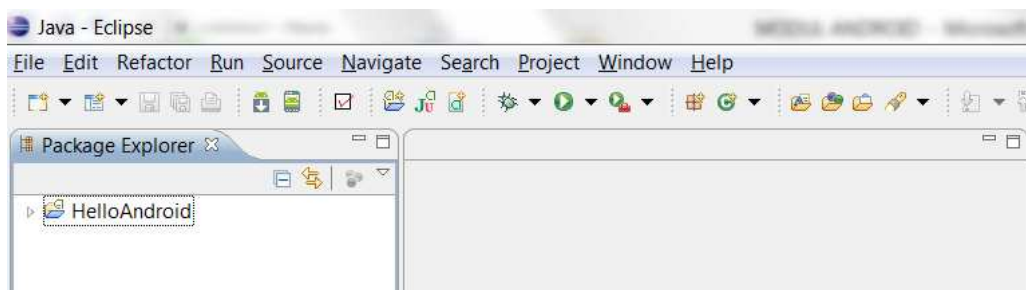


5. Isikan untuk Application Info:

**Application Name** : **HelloAndroid**  
**Package Name** : **emha.android.helloandroid**  
**Create Activity** : **MainActivity**  
**Minimum SDK** : **8**  
**Klik Finish**



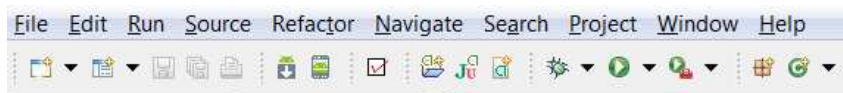
6. Didapatkan tampilan pada Package Explorer sbb:



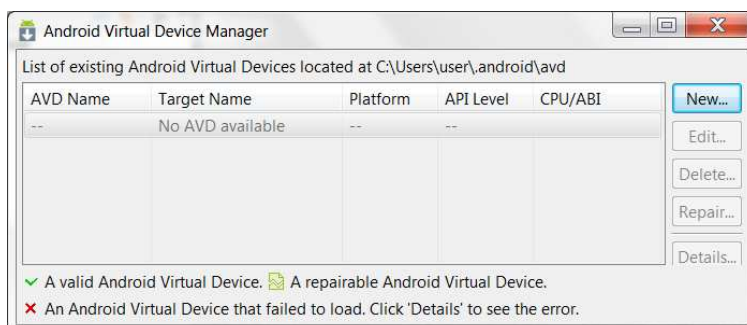
## 2.2. Mengkonfigurasi Emulator

Untuk melihat apakah aplikasi yang dibangun pada kondisi berjalan baik atau tidak, dapat dicoba dijalankan dengan emulator. Developer harus tahu bagaimana men-set up emulator dengan beberapa konfigurasi yang berbeda. Pertama harus membuat Android Virtual Device (AVD) yang disebut juga emulator. AVD merupakan Perangkat Virtual Android yang memiliki fitur seperti Perangkat Android sesungguhnya.

1. Buka AVD Manager, pada toolbar klik ikon  (Opens Android Virtual Device Manager)



2. Muncul kotak dialog sbb:

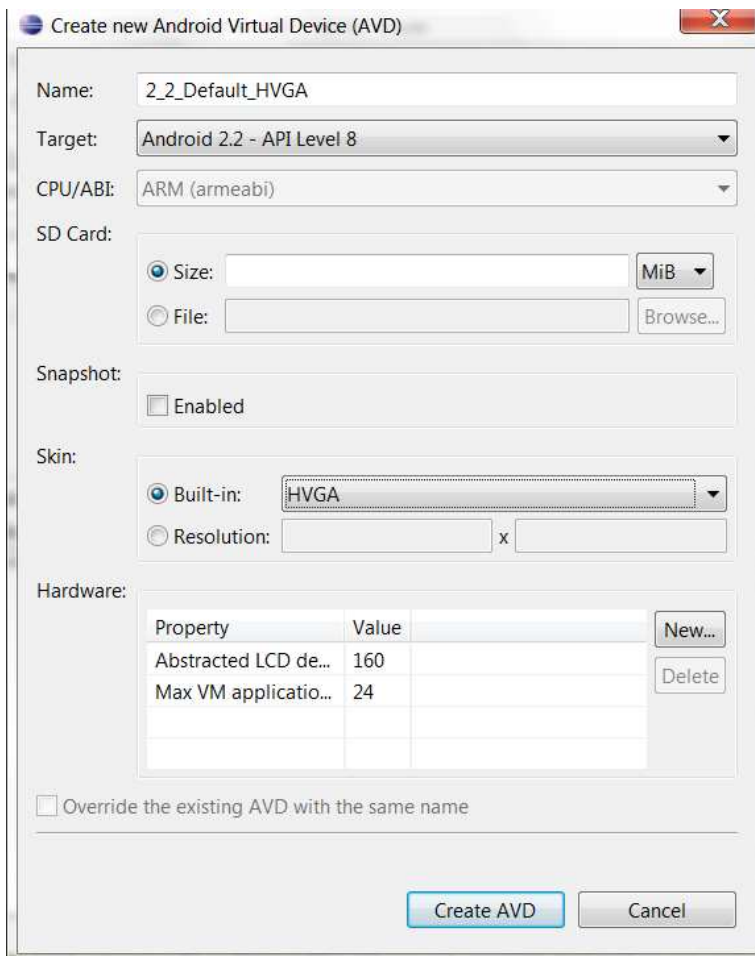


### 3. Klik New

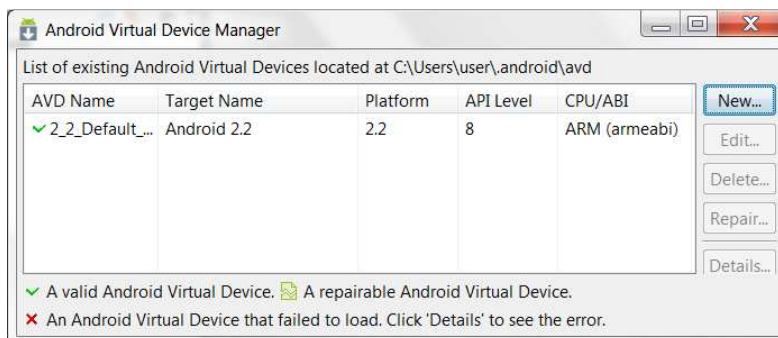
Kemudian isikan :

Name : 2\_2\_Default\_HVGA  
Target : Android 2.2 – API Level 8  
SD Card : <sementara kosongkan>  
Skin : HVGA  
Hardware : <sementara abaikan>

Klik Button → Create AVG



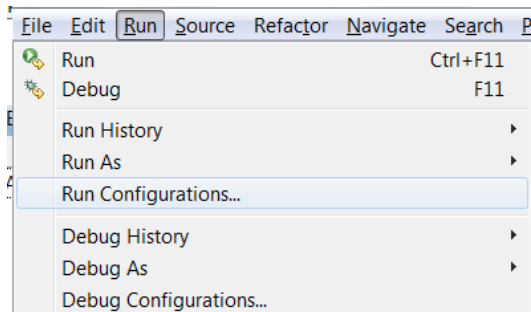
Didapatkan hasil sebagai berikut :



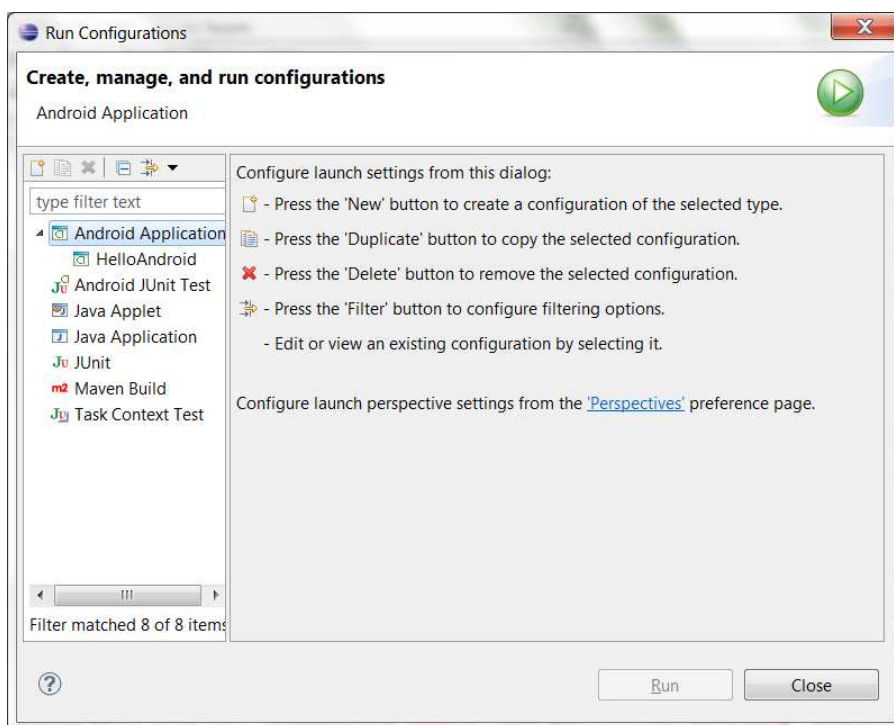
## 2.3. Menjalankan Project Aplikasi Android dengan Emulator

1. Sebelum project aplikasi dijalankan, atur terlebih dahulu **run configuration**

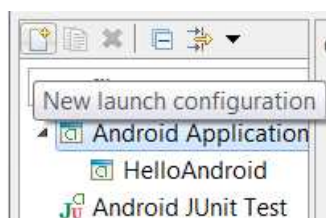
Pada MenuBar → Run → Run Configuration



2. Pilih Android Application



3. Kemudian klik ikon new

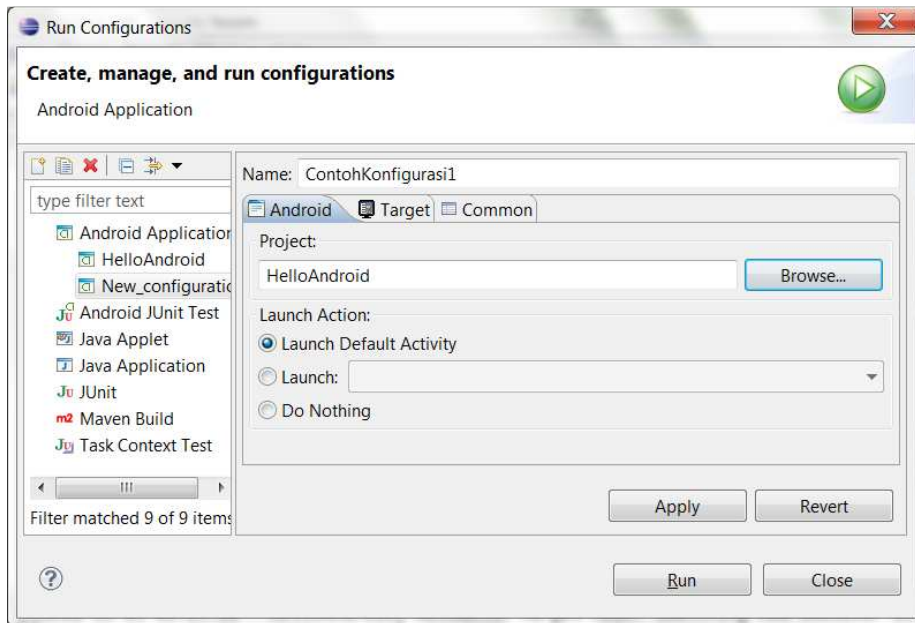




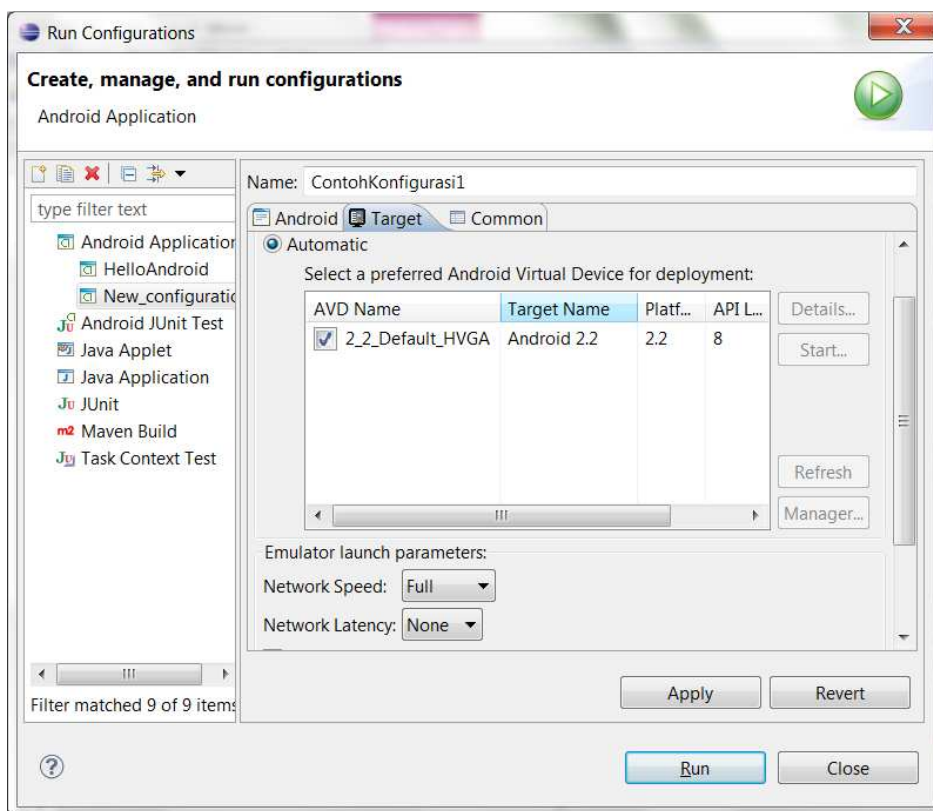
4. Masukkan Nama Konfigurasi dan Pilih Project yang akan dijalankan dengan konfigurasi ini

Name : ContohKonfigurasi1

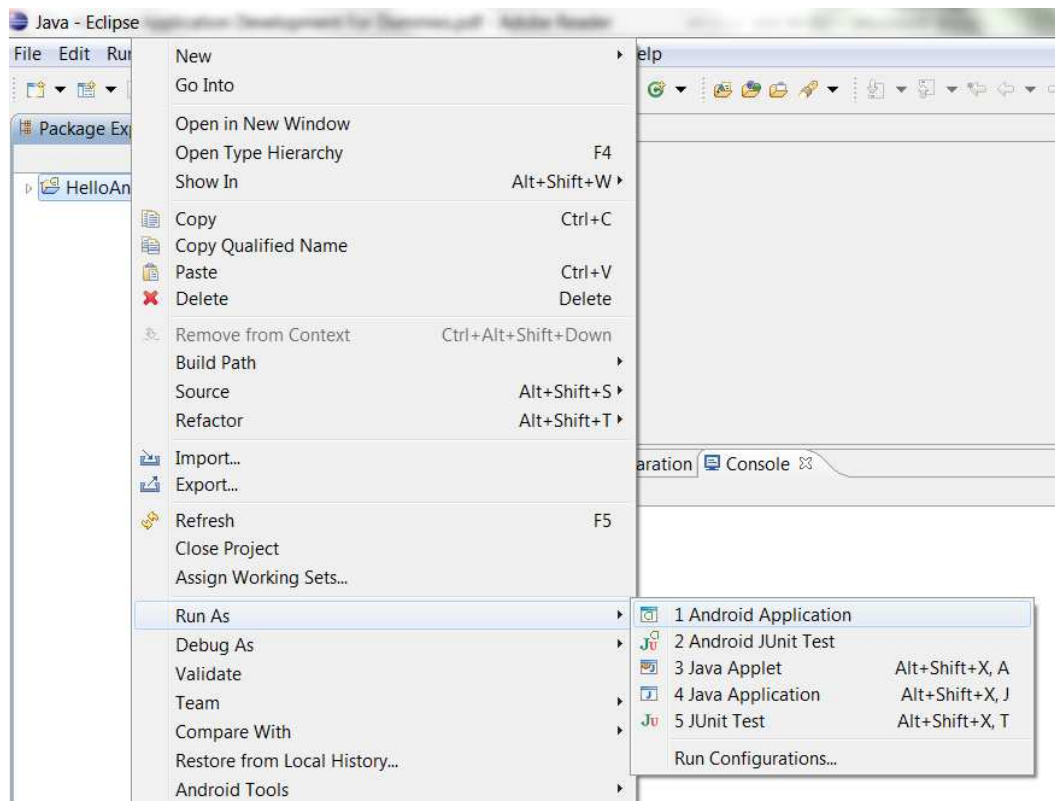
Project : HelloAndroid



Pada Tab Target : Pilih Automatic kemudian AVD → 2\_2\_Default\_HVGA

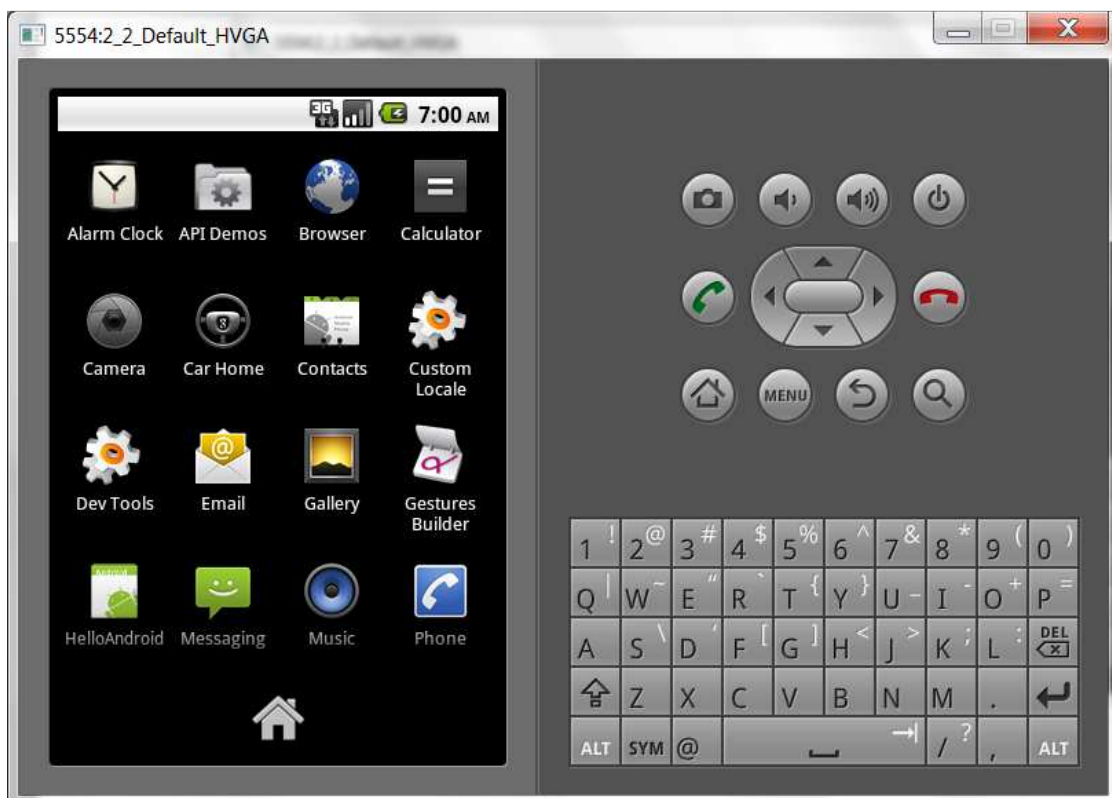


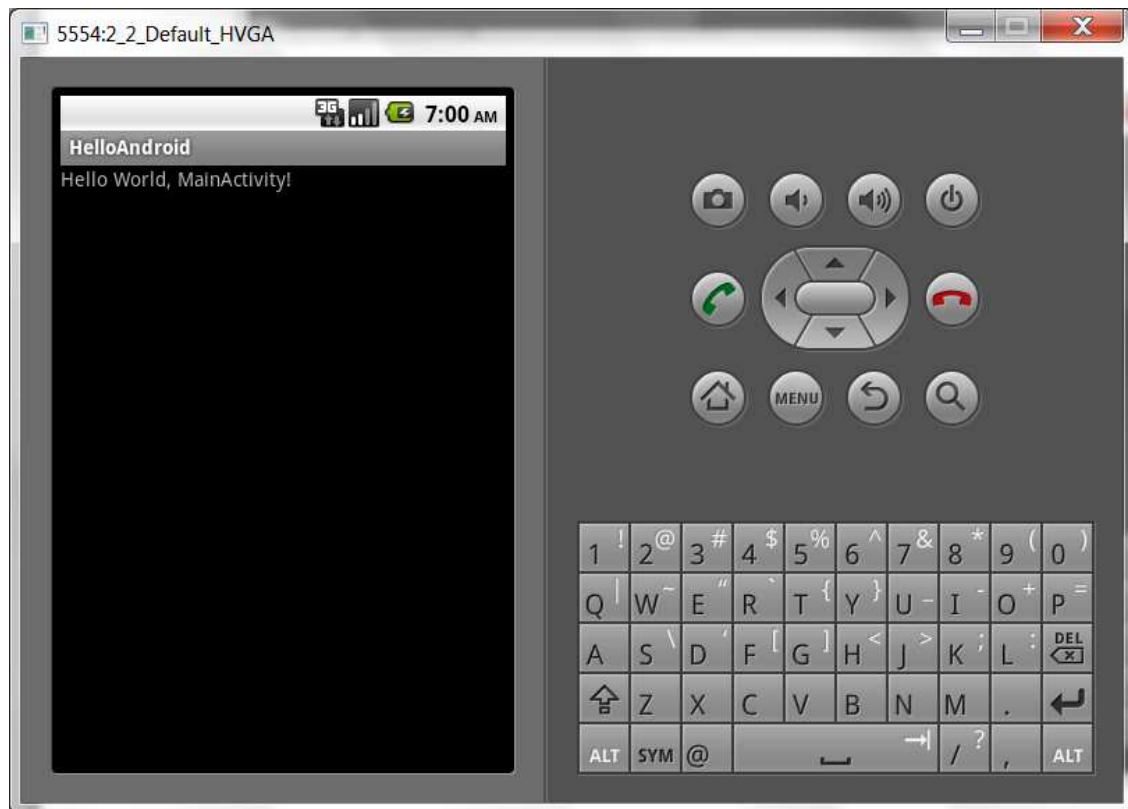
## 5. Jalankan



## 6. Tampilan AVD

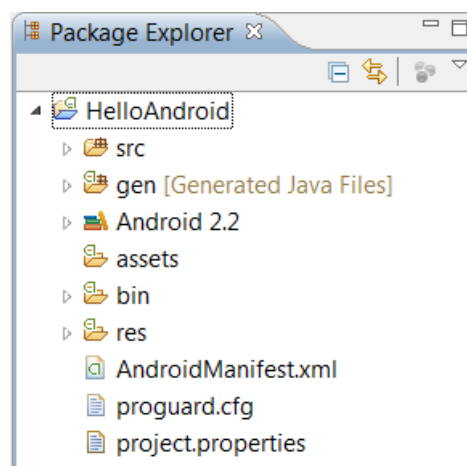






## 2.4. Struktur Project Android

Pada Project Explorer terlihat detail struktur dari suatu Project Android



Terdapat beberapa folder dibawah Nama Project, yaitu:

- 19) src  
merupakan folder untuk source
- 20) gen  
merupakan folder misterius. Folder ini berikan file yang di generate oleh ADT
- 21) Android Version (misal: Android 2.2)
- 22) assets

merupakan folder untuk menempatkan berbagai file asset (data) yang dimiliki project (contoh: file data dari SQLite). Data di folder ini dapat diakses melalui `AssetManager` atau metode `getAssets()`.

#### 23) res

merupakan folder untuk menempatkan berbagai resource yang digunakan oleh project, misal: file xml, icon atau picture.

Selain folder-folder diatas, terdapat beberapa folder lain misal folder: *bin*, *libs*, dan *referenced libraries*.

Folder *bin* tidak tampak dikarenakan di *hidden*. Sementara folder *libs* dan *referenced libraries* tidak akan muncul sampai developer menambahkan *third-party library* dan direferensi dalam project.

File *AndroidManifest.xml* membantu developer mengidentifikasi komponen yang di *build* dan dijalankan oleh aplikasi.

*Project.properties* membantu developer mengidentifikasi *default properties* dari project android.

## 2.5. Pustaka

Felker D.; Dobbs J, 2011, *Android Application Development For Dummies*, Wiley Publishing Inc., USA

## MATERI 3

### MEMBANGUN GRAPHICAL USER INTERFACE

- **Membangun Aplikasi Pengalih Mode *Silent***
  - **Merancang *Layout***
  - **Membangun Antarmuka Pengguna**
  - **Menambahkan *Image* dan *Button***
- 

1. Buatlah sebuah project android baru dengan aturan sebagai berikut:

Project Name : PengalihModeSilent

Application Name : PengalihModeSilent

Contents : default

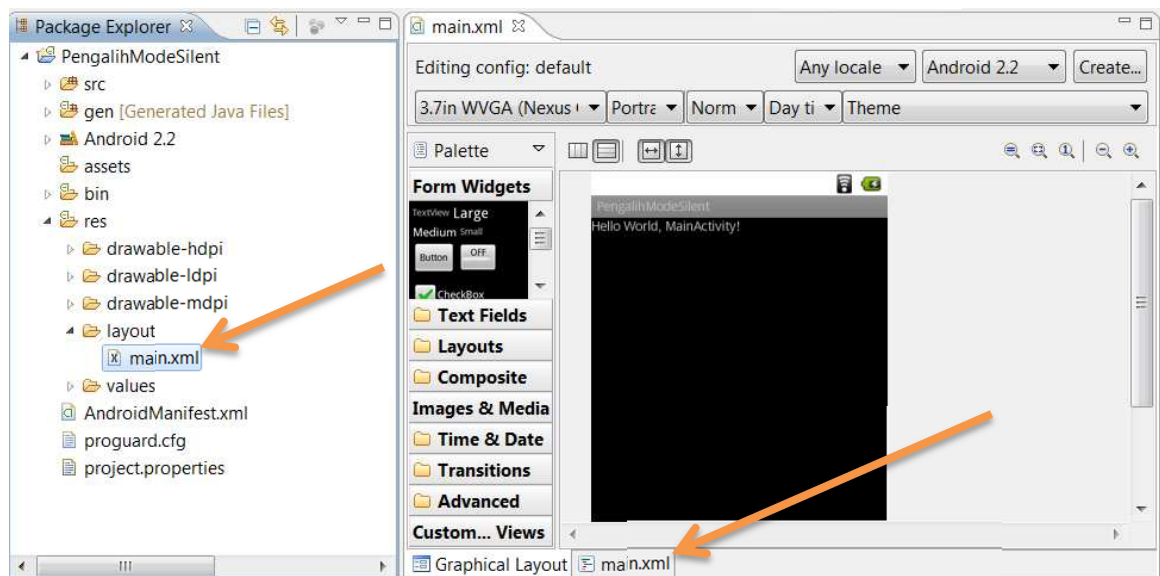
Build Target : Android 2.2

Package Name : emha.android.pengalihmodesilent

Create Activity : MainActivity

Min SDK Version : 8

2. Buka file main.xml di folder res → layout



Default-nya yang akan muncul adalah Visual Designer dari main.xml, klik pada tab main.xml dibagian bawah.



```

*main.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

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

</LinearLayout>

```

## Default Deklarasi XML

```
<?xml version="1.0" encoding="utf-8"?>
```

Baris pertama menunjukkan deklarasi file xml. Memberikan informasi tipe file tersebut terhadap editor.

## Tipe Layout

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

```

Baris selanjutnya diatas menunjukkan tipe Layout yang digunakan, dalam hal ini adalah LinearLayout (akan dibahas lebih lanjut nanti).

## View

```

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

```

Baris selanjut menunjukkan blok untuk view, yaitu blok dasar pembangun *user interface*. Dalam hal ini adalah sebuah TextView.

3. Dari main.xml, hapus bagian view (TextView) yang ada. Sehingga main.xml secara keseluruhan akan menjadi sbb:

```

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

</LinearLayout>

```

4. Tambahkan (tarik) image yang akan digunakan kedalam folder res → drawable-mdpi



phone\_on.png → untuk mode telepon regular

phone\_silent.png → untuk mode telepon silent

5. Tambahkan image ke dalam Layout

Dilakukan dengan mengetikkan kode tambahan di main.xml sehingga menjadi sbb:

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

    <ImageView
        android:id="@+id/phone_icon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:src="@drawable/phone_on" />

</LinearLayout>
```

6. Tambahkan (tarik) icon untuk aplikasi ke project, res → drawable-mdpi



7. Tambahkan button ke dalam main.xml, menjadi sbb:

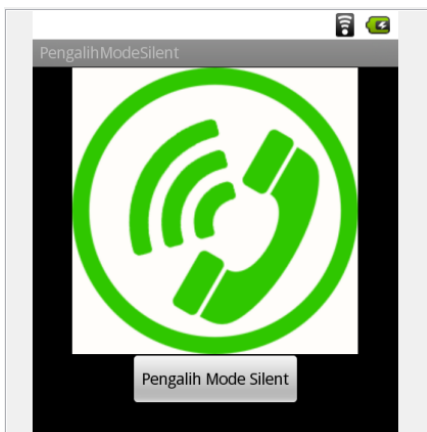
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <ImageView
        android:id="@+id/phone_icon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:src="@drawable/phone_on" />

    <Button
        android:id="@+id/toggleButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Pengalih Mode Silent" />

</LinearLayout>
```

8. Coba tampilkan Visual Designer



9. Atur Run Configuration

10. Jalankan Aplikasi PengalihModeSilent



11. Buka file MainActivity.java di folder src → emha.android.pengalihmodesilent



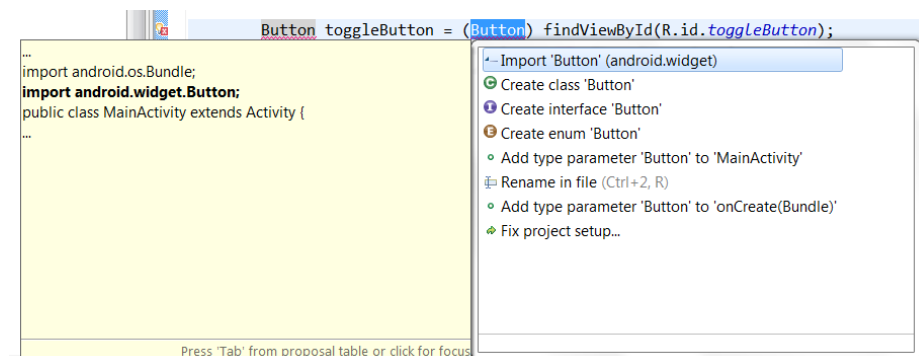
12. Kita akan mencoba membuat event handler pertama, yang menangani jika button di klik  
Tambahkan kode program sbb :

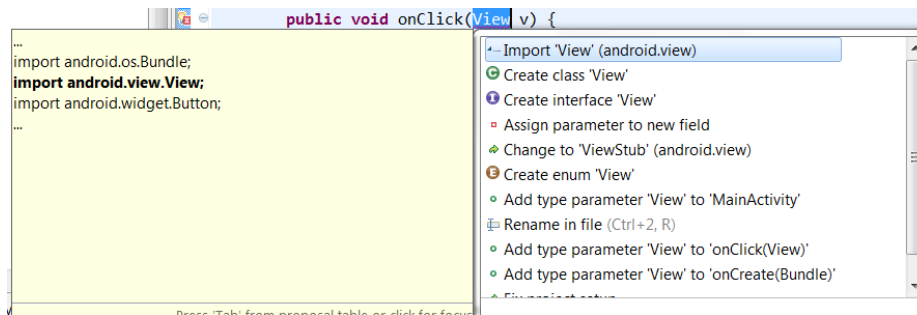


13. Tanda cross merah di kiri pinggir menunjukkan terdapat error pada baris kode bersangkutan.

14. Hal ini terjadi dikarenakan object Button dan View belum dikenali.

Pada tanda cross merah, klik kemudian klik import untuk menambahkan library





15. Pada bagian atas kode, akan terdapat beberapa tambahan import sbb:

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {
```

16. Untuk selanjutnya agak manajemen kode program menjadi lebih baik, maka kita ubah, dengan menempatkan kode program dengan fungsi tertentu dalam satu paket method

```
package emha.android.pengalihmodesilent;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        setButtonClickListener();
    }

    private void setButtonClickListener() {
        Button toggleButton = (Button) findViewById(R.id.toggleButton);
        toggleButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // TODO Auto-generated method stub
            }
        });
    }
}
```

17. Untuk selanjutnya kita akan coba bekerja dengan audio service dan pengaturan mode. Lengkapi kode program menjadi sbb:

```
package emha.android.pengalihmodesilent;

import android.app.Activity;

public class MainActivity extends Activity {

    private AudioManager mAudioManager;
    private boolean statusModeSilent;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        setButtonClickListener();
        mAudioManager = (AudioManager) getSystemService(AUDIO_SERVICE);
        checkIfPhoneIsSilent();
    }

    private void checkIfPhoneIsSilent() {
        int ringerMode = mAudioManager.getRingerMode();
        if (ringerMode == AudioManager.RINGER_MODE_SILENT) {
            statusModeSilent = true;
        } else {
            statusModeSilent = false;
        }
    }

    private void setButtonClickListener() {
        Button toggleButton = (Button) findViewById(R.id.toggleButton);
        toggleButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                if (statusModeSilent) {
                    // Ubah kembali ke mode normal
                    mAudioManager.setRingerMode(AudioManager.RINGER_MODE_NORMAL);
                    statusModeSilent = false;
                } else {
                    // Ubah ke mode silent
                    mAudioManager.setRingerMode(AudioManager.RINGER_MODE_SILENT);
                    statusModeSilent = true;
                }
                // Now toggle the UI again
                pindahUI();
            }
        });
    }
}
```



```

private void pindahUI() {
    ImageView imageView = (ImageView) findViewById(R.id.phone_icon);
    Drawable imgTampil;
    if (statusModeSilent) {
        imgTampil = getResources().getDrawable(R.drawable.phone_silent);
    } else {
        imgTampil = getResources().getDrawable(R.drawable.phone_on);
    }
    imageView.setImageDrawable(imgTampil);
}

protected void onResume() {
    super.onResume();
    checkIfPhoneIsSilent();
    pindahUI();
}
}

```

18. Jalankan aplikasi



Klik Button



## Pustaka

Felker D.; Dobbs J, 2011, *Android Application Development For Dummies*, Wiley Publishing Inc., USA

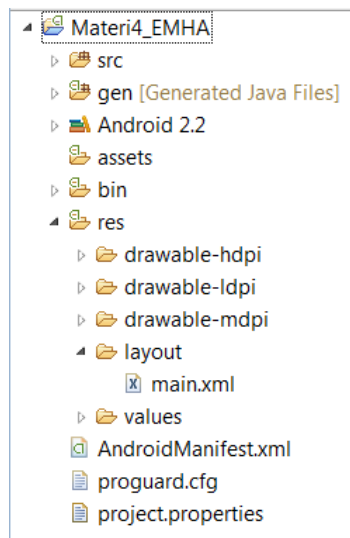
## MATERI 4

### MEMBANGUN GRAPHICAL USER INTERFACE

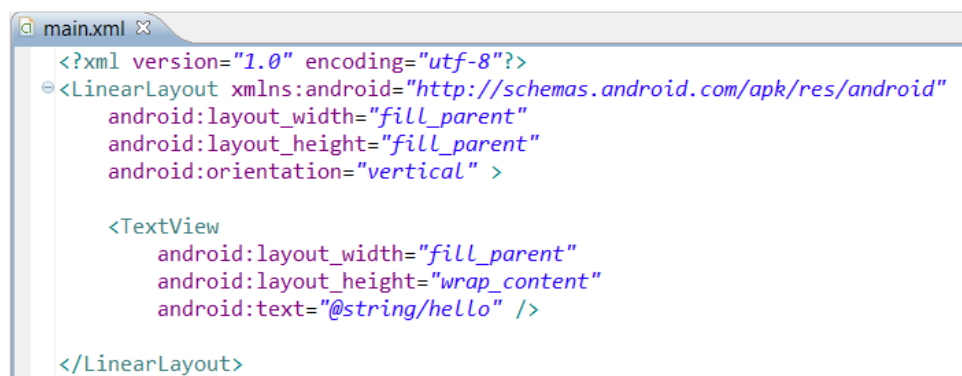
- Bekerja dengan TextView
  - Bekerja dengan Button
  - Bekerja dengan Image
  - Bekerja dengan CheckBox
  - Bekerja dengan RadioButton
- 

1. Buatlah sebuah project android baru dengan aturan sebagai berikut:

Project Name : Materi4\_NIMANDA  
Application Name : Materi4\_NIMANDA  
Contents : default  
Build Target : Android 2.2  
Package Name : emha.android.materi4  
Create Activity : MainActivity  
Min SDK Version : 8



2. Buka main.xml di folder layout

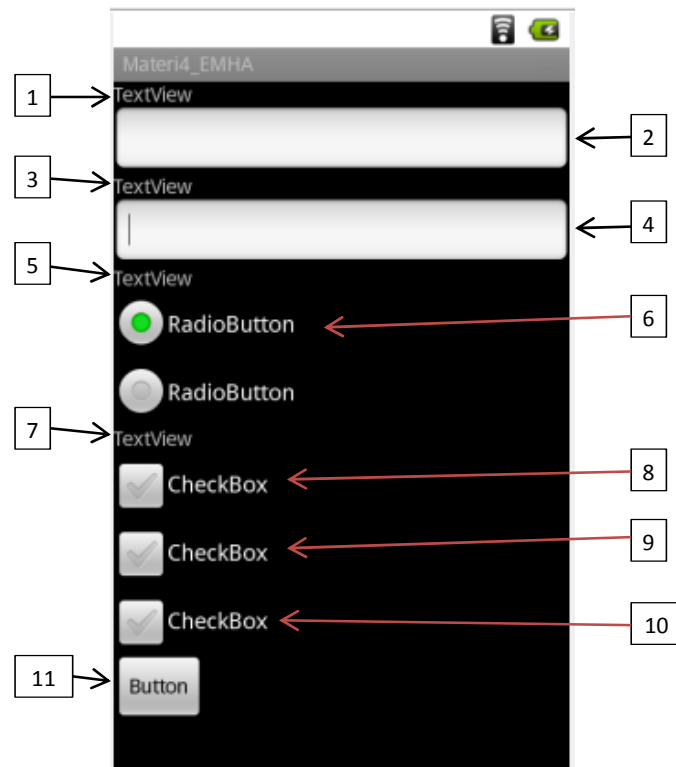


3. Hapus TextView yang sudah ada sehingga main.xml menjadi sbb:

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

    </LinearLayout>
```

4. Melalui Graphical Layout, tambahkan view-view sebagai berikut:



5. Atur property semua view sbb:

No	Tipe View	Properti	Nilai
1	TextView	id	@+id/textNama
		text	Nama Lengkap
		typeFace	monospace
		textSize	24dp
		textStyle	Bold
2	EditText (Plain Text)	id	@+id/editNama
		typeFace	monospace
		textSize	24dp
3	TextView	id	@+id/textEmail
		text	Email
		typeFace	monospace

		textSize	24dp
		textStyle	Bold
4	EditText (E-Mail)	id	@+id/editEmail
		typeFace	Monospace
		textSize	24dp
5	TextView	id	@+id/textJenisKelamin
		text	Jenis Kelamin
		typeFace	monospace
		textSize	24dp
		textStyle	Bold
6	RadioGroup		
	a. RadioButton	id	radioLaki
		text	Laki-Laki
	b. RadioButton	Id	radioPerempuan
		text	Perempuan
7	TextView	id	textHobi
		text	Hobi
		typeFace	monospace
		textSize	24dp
		textStyle	Bold
8	CheckBox	Id	checkBaca
		text	Membaca
9	CheckBox	id	checkMemancing
		text	Memancing
10	CheckBox	id	checkBersepeda
		text	Bersepeda
11	Button	id	btnOK
		text	OK

6. menu.xml akan berubah menjadi sbb:

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

    <TextView
        android:id="@+id/textNama"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nama Lengkap"
```

```

        android:typeface="monospace"
        android:textSize="24dp"
        android:textStyle="bold"/>

<EditText
    android:id="@+id/editNama"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="24dp"
    android:typeface="monospace"/>

<TextView
    android:id="@+id/textEmail"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Email"
    android:typeface="monospace"
    android:textSize="24dp"
    android:textStyle="bold"/>

<EditText
    android:id="@+id/editEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="24dp"
    android:typeface="monospace"
    android:inputType="textEmailAddress" >
    <requestFocus />
</EditText>

<TextView
    android:id="@+id/textJenisKelamin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Jenis Kelamin"
    android:typeface="monospace"
    android:textSize="24dp"
    android:textStyle="bold"/>

<RadioGroup
    android:id="@+id/radioGroup1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <RadioButton
        android:id="@+id/radioLaki"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="Laki-Laki" />

    <RadioButton
        android:id="@+id/radioPerempuan"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Perempuan" />
</RadioGroup>

```

```

<TextView
    android:id="@+id/textHobi"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hobi" android:textSize="24dp"/>

<CheckBox
    android:id="@+id/checkBaca"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Membaca" />

<CheckBox
    android:id="@+id/checkMemancing"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Memancing" />

<CheckBox
    android:id="@+id/checkBersepeda"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Bersepeda" />

<Button
    android:id="@+id/btnOK"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="OK" />

</LinearLayout>

```

7. Coba jalankan aplikasi, akan didapatkan tampilan sbb:



Akan tampak bahwa, ukuran layar secara vertical tidak cukup untuk menampilkan seluruh obyek view yang ada. Solusinya adalah dengan menambahkan Scroll untuk menggulung layar kearah vertical.



## 8. Tambahkan ScrollView, dan atur struktur dalam main.xml

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

    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <LinearLayout
            android:id="@+id/linearLayout1"
            android:layout_width="match_parent"
            android:layout_height="match_parent" >
            <LinearLayout>

            </ScrollView>

            <TextView
                android:id="@+id/textNama"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Nama Lengkap"
                android:typeface="monospace"
                android:textSize="24dp"
```

Ubah menjadi :

```
<?xml version="1.0" encoding="utf-8"?>
    <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/scrollView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:orientation="vertical" >

            <TextView
                android:id="@+id/textNama"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Nama Lengkap"
                android:typeface="monospace"
                android:textSize="24dp"
                android:textStyle="bold"/>

            ...

        </LinearLayout>

    </ScrollView>
```

9. Tambahkan kode program ke dalam btnOK, pada mainActivity

Jika di klik OK, maka akan memunculkan identitas ke Form

```
package emha.android.materi4;

import android.app.Activity;
import android.os.Bundle;
import android.text.style.LineHeightSpan.WithDensity;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.RadioButton;

public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        final EditText editNama = (EditText) findViewById(R.id.editNama);

        final EditText editEmail = (EditText)
            findViewById(R.id.editEmail);

        final RadioButton radioLaki = (RadioButton)
            findViewById(R.id.radioLaki);

        final RadioButton radioPerempuan = (RadioButton)
            findViewById(R.id.radioPerempuan);

        final CheckBox checkBaca = (CheckBox)
            findViewById(R.id.checkBaca);

        final CheckBox checkMemancing = (CheckBox)
            findViewById(R.id.checkMemancing);

        final CheckBox checkBersepeda = (CheckBox)
            findViewById(R.id.checkBersepeda);

        Button btnOK = (Button) findViewById(R.id.btnOK);

        btnOK.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                editNama.setText("Harun Al Rasyid");
                editEmail.setText("harun@amikom.ac.id");
                radioLaki.setChecked(true);
                checkBaca.setChecked(true);
                checkMemancing.setChecked(false);
                checkBersepeda.setChecked(true);
            }

        });
    }
}
```

10. Jalankan aplikasi



11. Klik button OK



## MATERI 5

### MEMAHAMI RESOURCES DAN APP WIDGET

- Bekerja dengan Resources
  - Bekerja dengan App Widgets
- 

Android memiliki beberapa resources, antara lain:

#### 1) *Layouts*

*Layout* merupakan bagian yang mendefinisikan bagaimana keadaan view dalam aplikasi. File layout defaultnya berada di direktori `res/layouts`. ADT secara default akan membuat file `main.xml` pada direktori tersebut.

LinearLayout	Layout yang menyusun bagian-bagiannya dalam satu baris
RelativeLayout	Layout yang posisi dari bagian-bagiannya dapat di deskripsikan dalam relasi satu dgn yg lainnya atau dgn induknya.
FrameLayout	Layout ini dirancang untuk memblokir area pada screen untuk menampilkan satu komponen. Developer dapat menambahkan beberapa anak bagian, tetapi semua akan dipatok ke kiri atas screen. Anak bagian dari layout akan digambarkan dalam stack, yang terakhir akan berada di atas stack.
TableLayout	Layout yang menyusun anak bagiannya dalam baris dan kolom

#### 2) *Strings*

#### 3) *Images*

#### 4) *Dimensions*

Merupakan unit ukuran numerik:

- dp (*density-independent pixels*)
- sp (*scale-independent pixels*)
- pt (*points*)
- px (*pixels*)
- mm (*millimeters*)
- in (*inches*)

## 5) Styles

*Styles* di android sangat mirip dengan Cascading Style Sheets (CSS) di pengembangan web. *Style* merupakan kumpulan dari property yang dapat diterapkan pada View secara mandiri, Activity, atau keseluruhan aplikasi (melalui file manifest). *Style* mendukung *inheritance*. Contoh property style antara lain: *font size*, *font color*, dan *screen background*.

## 6) Themes

Theme merupakan *Style* yang diterapkan pada keseluruhan Activity atau Aplikasi. Jika *Style* diterapkan sebagai Theme dalam suatu Aplikasi, maka semua View dan Activity dalam Aplikasi akan Meng-Inherits setting dari *Style*.

## 7) Values

*Value* dapat mengandung berbagai tipe untuk aplikasi, diantaranya:

- **Bool** : merupakan nilai Boolean. Disimpan dalam file xml, biasanya di lokasi `res/values/<filename>.xml`;  
Contoh: `res/values/bools.xml`
- **Integer**: merupakan nilai Integer. Disimpan dalam file xml, biasanya di lokasi `res/values/<filename>.xml`;  
Contoh: `res/values/integers.xml`
- **Integer Array**: merupakan Array Integer. Disimpan dalam file xml, biasanya di lokasi `res/values/<integers>.xml`; Yang mana `integers` merupakan `<nama file>`
- **Typed Arrays**: merupakan tipe yang digunakan untuk membuat Array dari berbagai resource yang tidak sejenis. Biasanya disimpan di `res/values/<filename>.xml`;  
Misal `res/values/types.xml`

## 8) Menus

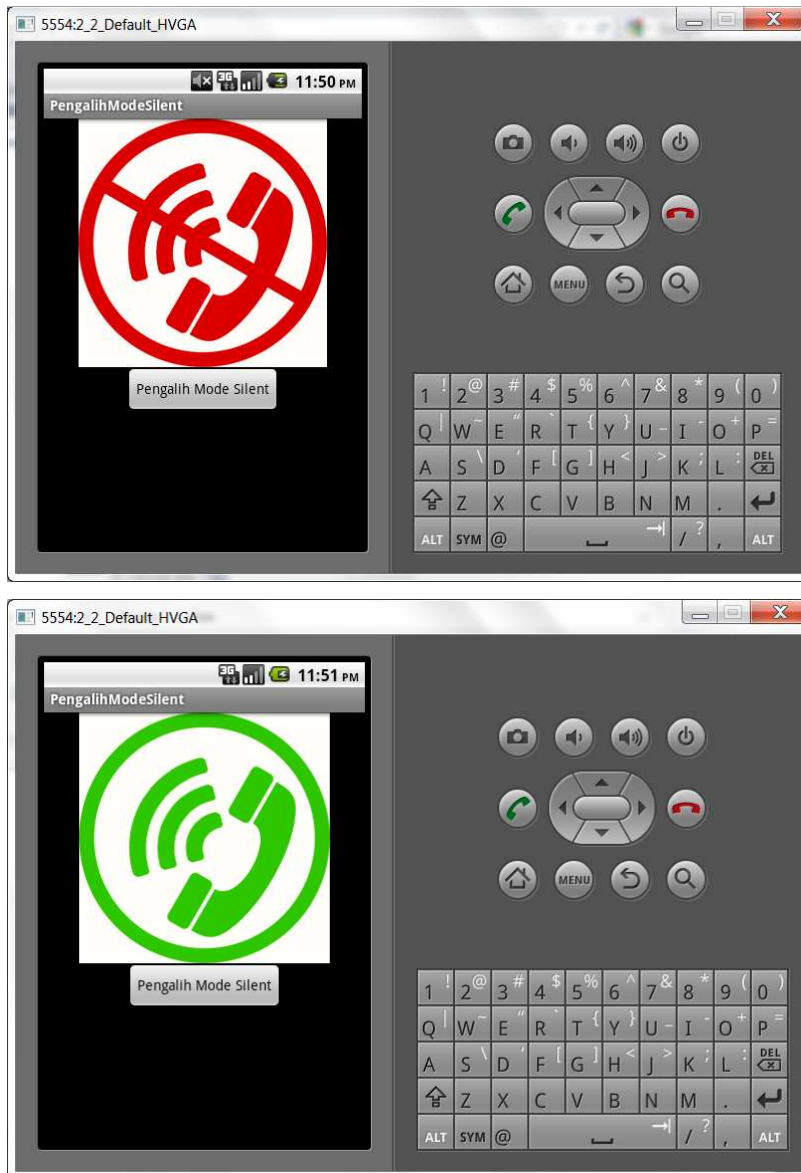
*Menu* dapat didefinisikan melalui kode maupun XML. Cara awal untuk mendefinisikan menu adalah melalui XML. Biasanya menu diletakkan di direktori `/menus`. Setiap menu memiliki file `.xml` sendiri.

## 9) Colors

File *colors* biasanya diletakkan di `values/colors.xml`. File ini memungkinkan developer untuk mendefinisikan nama color, misal: `login_screen_font_colors`, yang berarti color dari font yang digunakan pada halaman login.

## LATIHAN

1. Buka kembali project **PengalihModeSilent** yang pernah dibuat



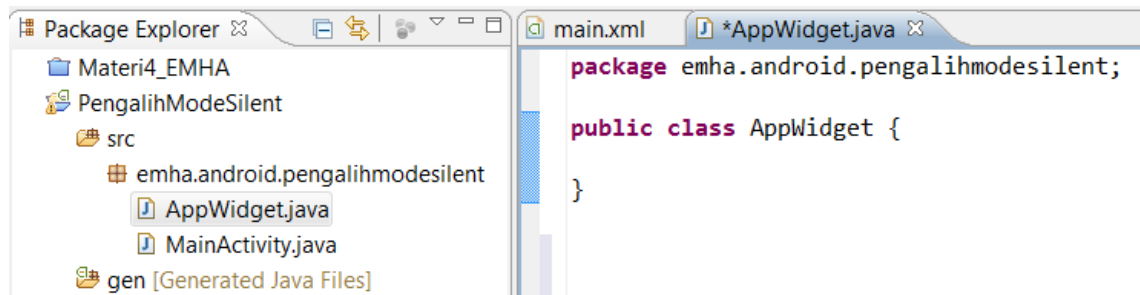
2. Pada sesi latihan ini akan coba di buat Home Screen Widget untuk Aplikasi **PengalihMode Silent**. Beberapa class yang akan digunakan:
  - Intent
  - BroadcastReceiver
  - AppWidgetProvider
  - IntentService
  - AppWidgetProviderInfo

Home Screen Widget dalam android merupakan miniature aplikasi yang dapat dilekatkan pada aplikasi lain seperti pada Home Screen. Hal ini juga di sebut App Widgets. App Widgets dapat menerima input dari pengguna melalui event click dan dapat meng-update

diri mereka sendiri melalui jadwal yg regular. App Widget diimplementasikan pada *home screen* dengan *long-pressing*.

Untuk membuat Aplikasi PengalihModeSilent lebih bermanfaat, maka akan digunakan Home Screen Widget, sehingga pengguna bisa menambahkan aplikasi ini ke home screen perangkat mereka.

3. Tambahkan sebuah class baru pada project anda (**Aplikasi PengalihModeSilent**), pada package **emha.android.pengalihmodesilent**, dengan nama: AppWidget.java



4. Lengkapi kode program menjadi seperti berikut ini:

```
package emha.android.pengalihmodesilent;

import android.appwidget.AppWidgetManager;
import android.appwidget.AppWidgetProvider;
import android.content.Context;
import android.content.Intent;

public class AppWidget extends AppWidgetProvider {

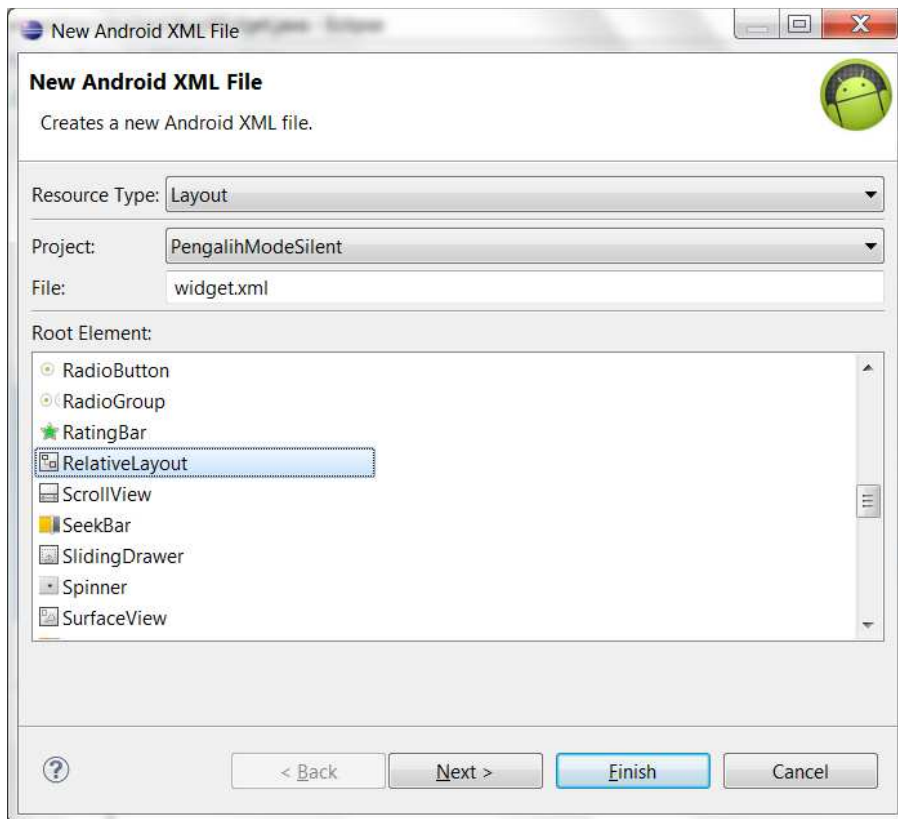
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction() == null) {
            //Lakukan sesuatu
        } else {
            super.onReceive(context, intent);
        }
    }

    @Override
    public void onUpdate(Context context, AppWidgetManager
        appWidgetManager, int[] appWidgetIds) {

        //Lakukan Sesuatu
    }
}
```



5. Pada layout, buat sebuah **layout baru** dengan nama **widget.xml**



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

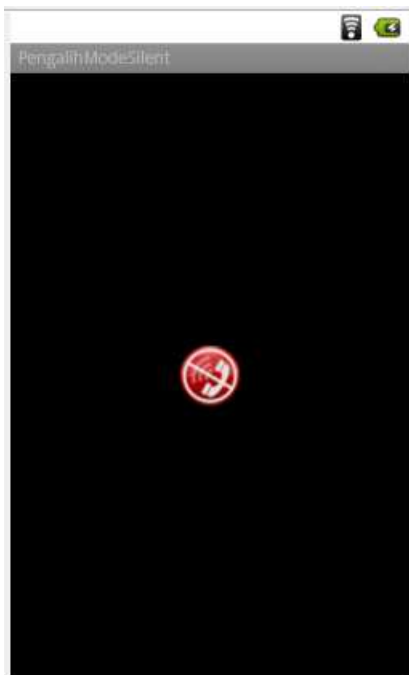
</RelativeLayout>
```

6. Tambahkan sebuah **ImageView**, dan atur properties sehingga didapat kondisi widget.xml sebagai berikut:

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

    <ImageView
        android:id="@+id/kondisiPhone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/icon"
        android:clickable="true"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```



7. Lengkapi kode program pada src/ → widget.java, menjadi sbb:

```
package emha.android.pengalihmodesilent;

import android.app.Activity;
import android.app.IntentService;
import android.app.PendingIntent;
import android.appwidget.AppWidgetManager;
import android.appwidget.AppWidgetProvider;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.media.AudioManager;
import android.widget.RemoteViews;

public class AppWidget extends AppWidgetProvider {

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction() == null) {
            //Lakukan sesuatu
            context.startService(new Intent(context, ToggleService.class));
        } else {
            super.onReceive(context, intent);
        }
    }

    @Override
    public void onUpdate(Context context, AppWidgetManager
        appWidgetManager, int[] appWidgetIds) {
        //Lakukan Sesuatu
        context.startService(new Intent(context, ToggleService.class));
    }
}
```

```

public static class ToggleService extends IntentService {
    public ToggleService() {
        super("AppWidget$ToggleService");
    }

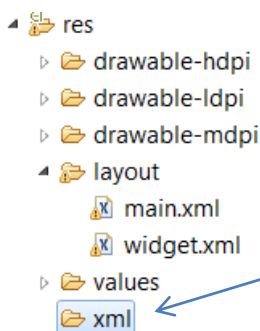
    @Override
    protected void onHandleIntent(Intent intent) {
        ComponentName me=new ComponentName(this, AppWidget.class);
        AppWidgetManager mgr=AppWidgetManager.getInstance(this);
        mgr.updateAppWidget(me, buildUpdate(this));
    }

    private RemoteViews buildUpdate(Context context) {
        RemoteViews updateViews = new
            RemoteViews(context.getPackageName(),R.layout.widget);
        AudioManager audioManager =
            (AudioManager)context.getSystemService(Activity.AUDIO_SERVICE);
        if (audioManager.getRingerMode() ==
            AudioManager.RINGER_MODE_SILENT) {
            updateViews.setImageViewResource(R.id.kondisiPhone,
                R.drawable.phone_on);
            audioManager.setRingerMode(AudioManager.RINGER_MODE_NORMAL);
        } else {
            updateViews.setImageViewResource(R.id.kondisiPhone,
                R.drawable.phone_silent);
            audioManager.setRingerMode(AudioManager.RINGER_MODE_SILENT);
        }
        Intent i = new Intent(this, AppWidget.class);
        PendingIntent pi = PendingIntent.getBroadcast(context, 0, i,0);
        updateViews.setOnClickPendingIntent(R.id.kondisiPhone, pi);
        return updateViews;
    }
}

```

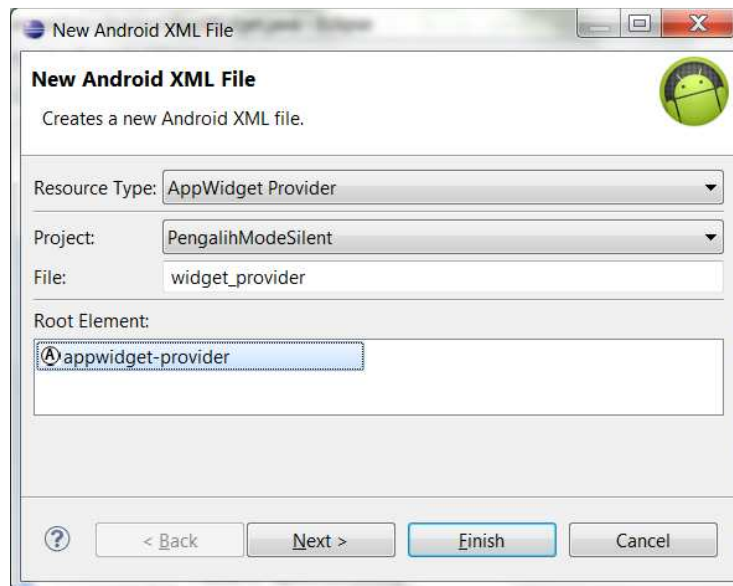
## 8. Mengatur App Widget Metadata

- a. Pada direktori res/
- b. Buat satu direktori baru, New → Folder, beri nama: xml

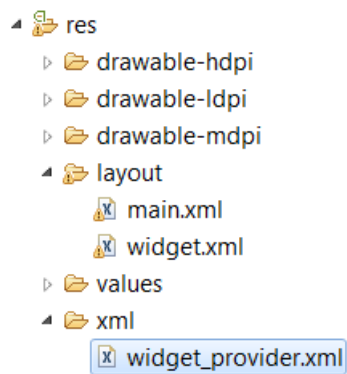


- c. Dalam folder xml, buat sebuah new android xml file, dengan nama: **widget\_provider.xml**

d. Tipe file tersebut adalah AppWidgetProvider



e. Klik Finish



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

</appwidget-provider>
```

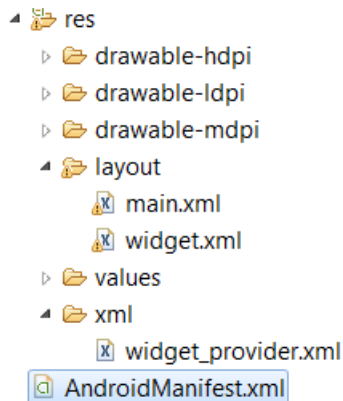
9. Lengkapi widget\_provider.xml menjadi sbb:

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="79px"
    android:minHeight="79px"
    android:updatePeriodMillis="1800000"
    android:initialLayout="@layout/widget" >

</appwidget-provider>
```

10. Langkah selanjutnya, mendaftarkan komponen baru ke manifest

a. Buka : AndroidManifest.xml



Akan tampak source xml sbb:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="emha.android.pengalihmodesilent"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="8" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

b. Tambahkan elemen xml sehingga menjadi sbb (yg diblok orange adalah kode yg baru):

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="emha.android.pengalihmodesilent"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="8" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
```

```

        android:name=".MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

    <receiver
        android:name=".AppWidget"
        android:icon="@drawable/icon"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        </intent-filter>
        <meta-data
            android:name="android.appwidget.provider"
            android:resource="@xml/widget_provider" />
        </receiver>

    <service android:name=".AppWidget$ToggleService" />
</application>
</manifest>

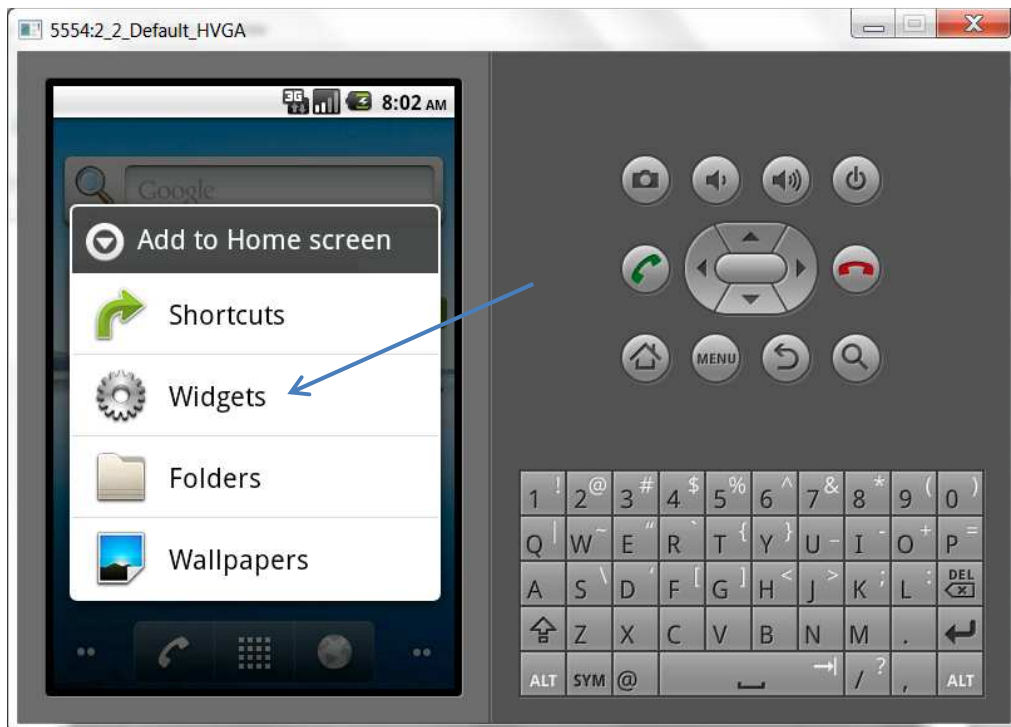
```

## 11. Selanjutnya, Meletakkan Widget di atas Home Screen

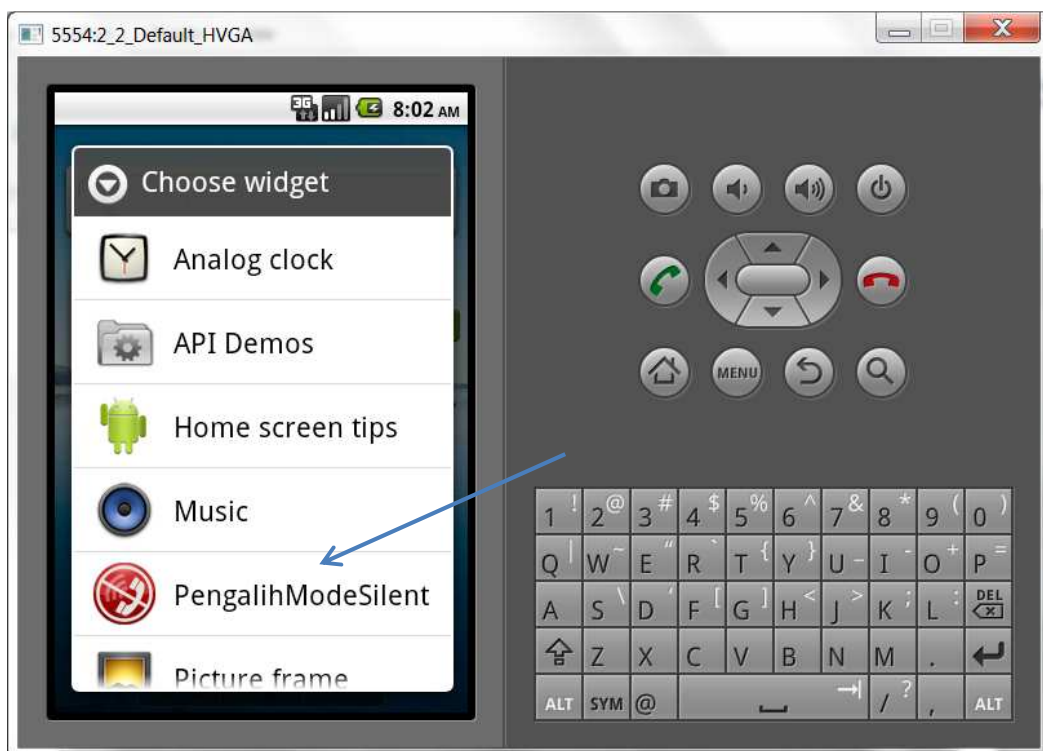
### a. Klik lama di home screen



- b. Akan muncul pop up, pilih **widgets**



- c. Kemudian Pilih **PengalihModeSilent**





12. Coba jalankan aplikasi, klik icon widgets PengalihModeSilent di home screen



