

# Praktikum 8

## Dasar Pemrograman Android

(Instalasi Java JDK, Eclipse IDE & Android SDK)

### Pokok Bahasan

- Konsep Pemrograman Bahasa Java pada Android
- Deklarasi Pemrograman Java pada Android
- Penggunaan import dan resource

### Tujuan Belajar

- Mengetahui cara instalasi Java JDK, Eclipse IDE, dan Android SDK.
- Mengenal tentang konsep pemrograman java untuk Android
- Mengenal pemrograman berorientasi obyek untuk Android

### Pendahuluan

Android merupakan adalah perangkat lunak terdiri dari OS beserta tool dan library lainnya yang dikembangkan oleh perusahaan Google, dan bersifat Open Source. Saat ini disediakan Android Software Development Kit (SDK) sebagai alat bantu dan API diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java, sehingga mudah digunakan untuk belajar membuat aplikasi. Fitur-fitur Android antara lain:

- Mesin virtual Dalvik dioptimalkan untuk perangkat mobile
- Integrated browser berdasarkan engine open source WebKit
- Grafis yang dioptimalkan dan didukung oleh perpustakaan grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware)
- SQLite untuk penyimpanan data
- Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
- Kamera, GPS, kompas, dan accelerometer (hardware tergantung)
- Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin.

### Framework Aplikasi

- Pengembang memiliki akses penuh framework API yang sama yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (reuse) dengan mudah.

- Setiap aplikasi dapat memanfaatkan kemampuan ini dan aplikasi yang lain mungkin akan memanfaatkan kemampuan ini (sesuai dengan batasan keamanan yang didefinisikan oleh framework). Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna.
- Semua aplikasi yang merupakan rangkaian layanan dan sistem, termasuk:
  - View Set kaya dan extensible yang dapat digunakan untuk membangun aplikasi, termasuk daftar, grids, kotak teks, tombol, dan bahkan sebuah embeddable web.
  - Content Provider yang memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak telp) atau dari data mereka sendiri.
  - Resource Manager, yang menyediakan akses ke kode sumber non-lokal seperti string, gambar, dan tata letak file
  - Notifikasi Manager yang memungkinkan semua kustom aplikasi untuk ditampilkan dalam alert status bar
  - Activity Manager yang mengelola siklus hidup aplikasi dan menyediakan navigasi umum backstack

### Android Runtime

- Android terdiri dari satu set core libraries yang menyediakan sebagian besar fungsi yang tersedia dalam core libraries dari bahasa pemrograman Java. Setiap menjalankan aplikasi Android sendiri dalam proses, dengan masing-masing instance dari mesin virtual Dalvik (Dalvik VM).
- Dalvik dirancang agar perangkat dapat menjalankan multiple VMs secara efisien. Mesin
- Virtual Dalvik dieksekusi dalam Dalvik executable (.dex), sebuah format yang dioptimalkan untuk memori yang kecil. Dalvik VM berbasis, berjalan dan dikompilasi oleh compiler bahasa Java yang telah ditransformasikan ke dalam .dex format yang disertakan oleh tool "dx".
- Dalvik VM bergantung pada kernel Linux untuk berfungsi , seperti threading dan manajemen memori tingkat rendahnya .

### Linux Kernel

- Android bergantung pada Linux Versi 2.6 untuk inti sistem pelayanan seperti keamanan, manajemen memori, proses manajemen, susunan jaringan, dan driver model.
- Kernel juga bertindak sebagai lapisan yang abstrak antara hardware dan software stacknya.

### Android ADT

- Android Development Tools (ADT) merupakan plugin untuk Eclipse IDE yang didesain dengan kemampuan yang lengkap dan terintegrasi untuk membangun aplikasi Android.
- ADT dapat meningkatkan kemampuan yang dimiliki Eclipse to agar dapat digunakan untuk membuat project aplikasi, membuat uer interface, menambahkan beberapa package yang berkaitan untuk membangun aplikasi dengan framework application user interface (API) pada Android.

- Disamping itu kita bias melakuka debug menggunakan Android SDK tools, meskipun sudah dalam bentuk file aplikasi yang telah jadi .apk dalam rangka menggunakan aplikasi Android yang telah dibuat.

### Praktek 1: Instalasi Java JDK, Eclipse IDE, dan Android SDK

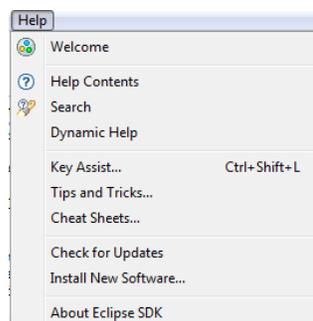
1. Periksa PC masing-masing, dan pastikan telah bekerja dengan baik.
2. Periksa apakah beberapa perangkat lunak yang diperlukansudah tersedia antara lain:
  - o OS Windows XP/7
  - o Eclipse IDE for Java (122 MB)
  - o JDK 6 atau 7 ( kalau sudah ada gak perlu install lg) 70 MB
  - o Android Development Tool Plugin (4,8 MB)
  - o Android SDK (34 MB)

#### **Berikut adalah cara instalasi Android:**

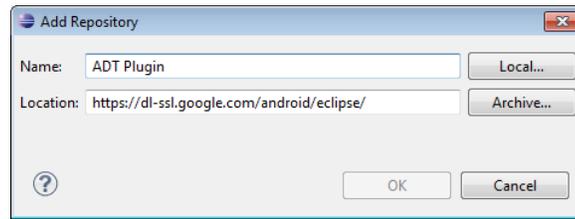
1. Instal JDK, apabila sudah pernah menginstall maka tidak perlu menginstall lagi.
2. Ekstrak Eclipse IDE yang sudah di download. Kemudian ekstrak pada directory yang mudah diakses. Jika telah siap maka jalankan Eclipse, dan akan ditampilkan logo seperti berikut:



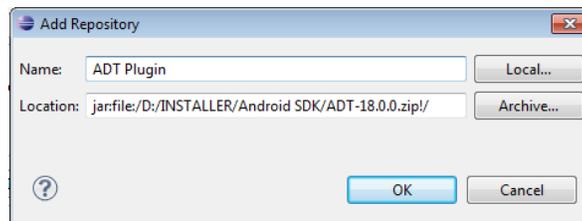
3. Kemudian pilih menu **Help > Install New Software...**



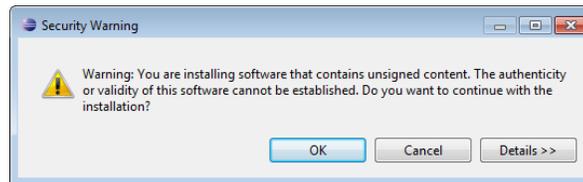
4. Klik tombol **Add...** di kiri atas, lalu muncul **Add Repository**, disini kita bisa masukkan ADTnya secara online dengan memasukan alamat repository Android (<https://dl-ssl.google.com/android/eclipse/>).



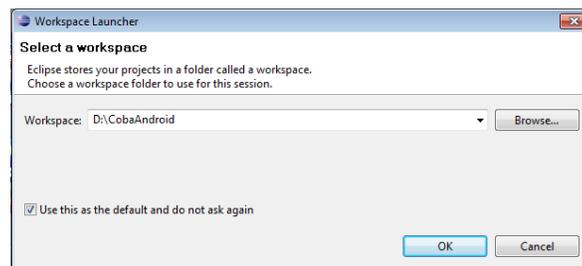
5. Jika secara offline dengan menekan tombol **Archive** dan arahkan ke file yang telah kita download sebelumnya, seperti berikut:



Jika terdapat Warning seperti gambar dibawah maka tekan OK.



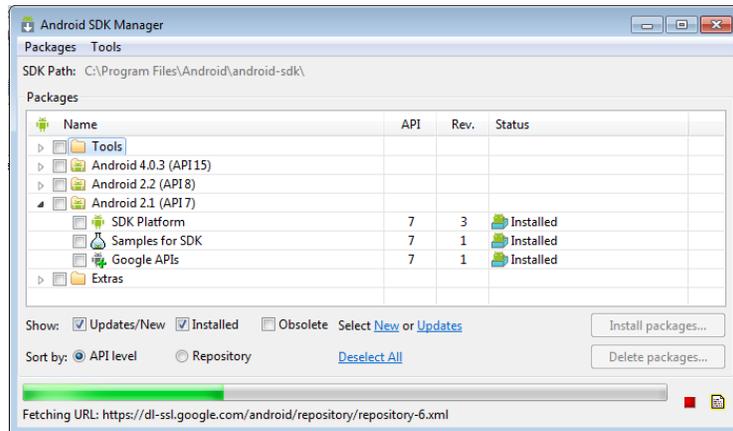
6. Setelah proses diatas selesai dilakukan, maka selanjutnya jalankan eclipse dan tentukan directori **Workspace**, seperti contoh dibawah.



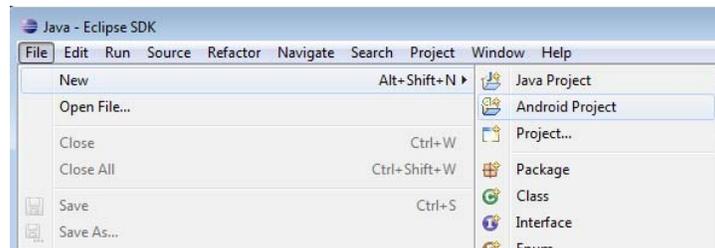
7. Setelah proses diatas selesai dilakukan, maka selanjutnya pilih **Window > Preferences > Android**, lalu browse ke folder SDK Android lalu tekan **Apply**, tunggu beberapa saat, lalu tekan OK.

8. Pilih lagi **Window > Android SDK and AVD Manager**, nah disini kita harus mendownload lagi file yang kita butuhkan, misalnya ingin buat aplikasi yang

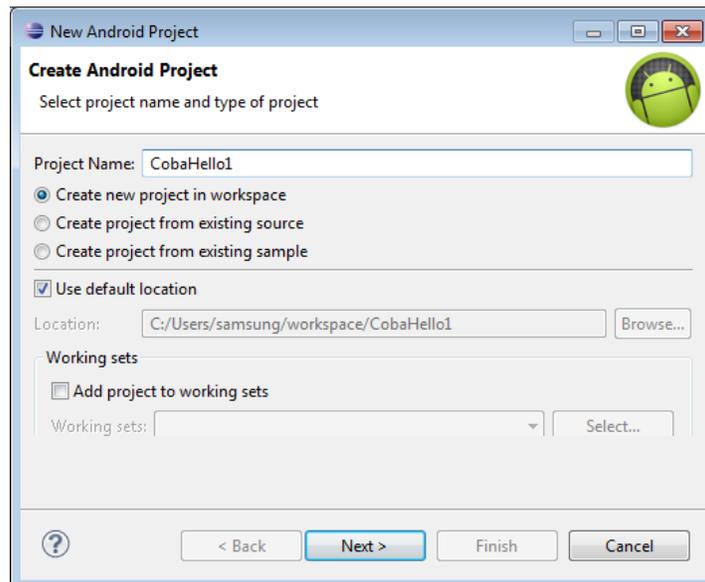
berjalan di Android 2.1, maka download lah **SDK Platform Android 2.1**, lebih baik kita download sama **Sample** dan **Documentation** agar memudahkan kita untuk mencari dokumentasi dan belajar dari source code yang sudah ada.



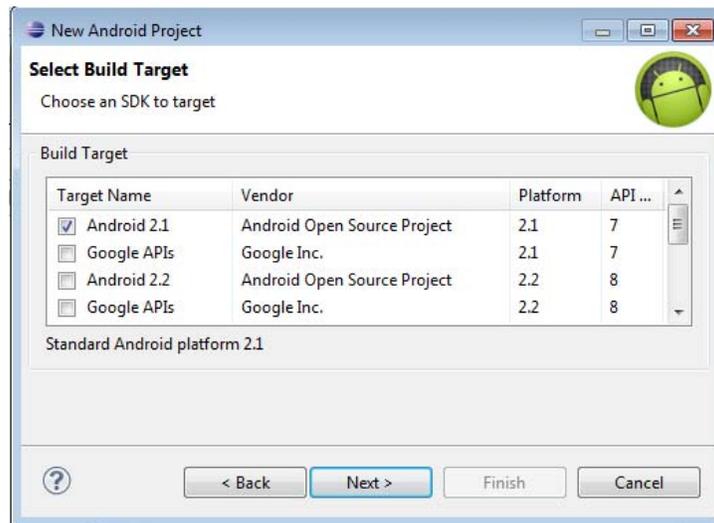
9. Bila langkah-langkah diatas telah dilakukan dengan benar maka dapat dikatakan selesai instalasinya, Selanjutnya langsung bisa memulai membuat aplikasi dengan memilih **File > New > Android Project**.



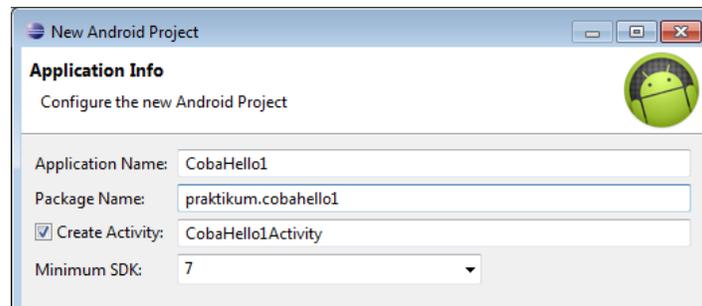
10. Pada contoh ini akan membuat **Android Project** dengan nama **CobaHello1**.



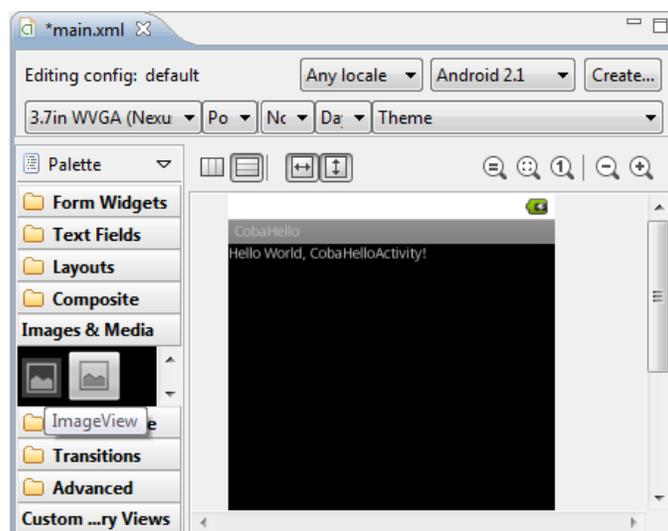
11. Kemudian pilih **Android Target** sesuai dengan HP yang dimiliki.



12. Kemudian pilih minimum SDK yang sesuai dengan HP yang dimiliki.



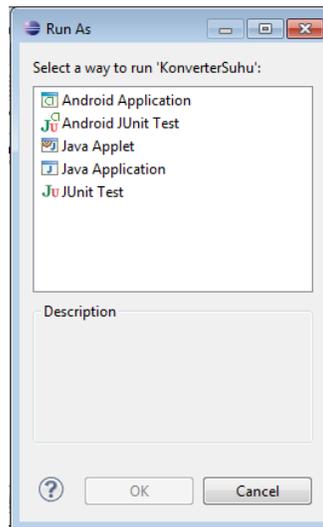
13. Kemudian periksa project yang telah dibuat dengan menampilkan Graphical Layout dari main.xml seperti gambar dibawah:



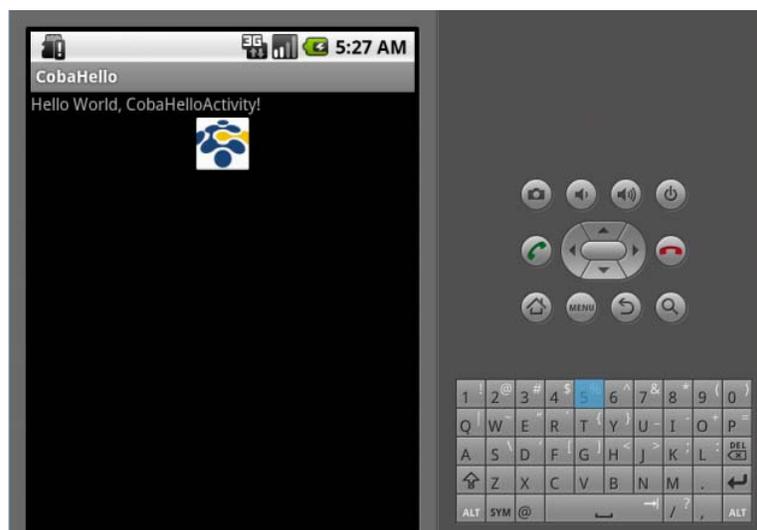
14. Lalu tambahkan gambar icon seperti berikut ini:



15. Jika telah selesai maka click button Run dan akan ditampilkan menu seperti berikut, lalu pilih Android Application.

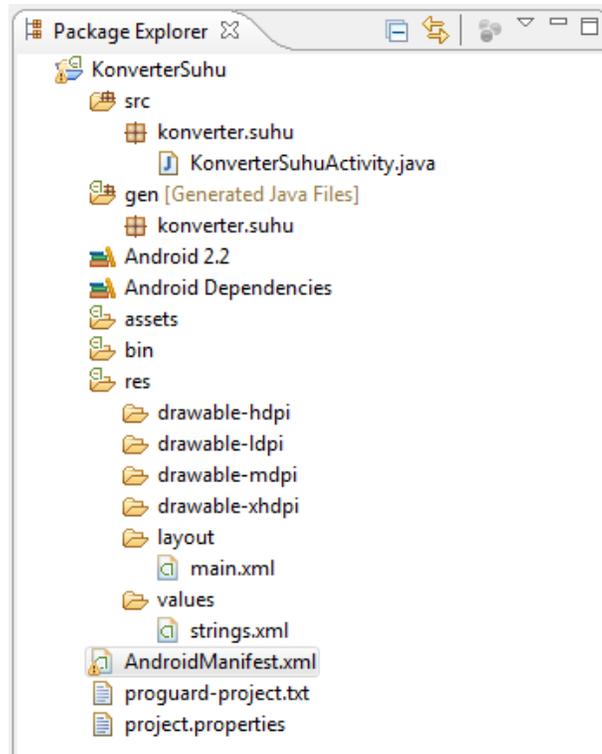


16. Jika tidak ada kesalahan maka akan ditampilkan pada emulator seperti terlihat dibawah ini:



## Praktek 2 : Program Konverter Suhu

1. Pertama-tama butlah project baru, misalnya dengan nama **KonverterSuhu**. Selanjutnya pada *Package Explorer* akan terlihat susunan seperti berikut:



2. Langkah kerja aplikasi Android pertama kali akan memeriksa file **AndroidManifest.xml** pada project kita. Dalam file tersebut, sistem android akan mencari sebuah element `<activity>` yang ditetapkan sebagai **LAUNCHER**, ditandai dengan memiliki elemen `<category>` dengan atribut `android:name="android.intent.category.LAUNCHER"`.
3. Kita lihat file **AndroidManifest.xml** dalam aplikasi **KonverterSuhu**, seperti berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="konverter.suhu"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="7" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name="konverter.suhu.KonverterSuhuActivity"
            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>
```

4. Kemudian edit file *KonverterSuhuActivity.java* seperti contoh dibawah ini. File inilah yang mengatur logic dari aplikasi, berdasarkan informasi yang ada di *AndroidManifest.xml*, maka file inilah yang akan dijalankan pertama kali ketika user ingin menjalankan aplikasi.

```
package konverter.suhu;

import konverter.suhu.R;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.Toast;

public class KonverterSuhuActivity extends Activity {
    private EditText text;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        text = (EditText) findViewById(R.id.editText1);
    }

    // method untuk click button
    public void myClickHandler(View view) {
        switch (view.getId()) {
            case R.id.button1:
                RadioButton celsiusButton = (RadioButton)
                findViewById(R.id.radio0);
                RadioButton fahrenheitButton = (RadioButton)
                findViewById(R.id.radio1);
                if (text.getText().length() == 0) {
                    Toast.makeText(this, "Input salah!",
                    Toast.LENGTH_LONG).show();

                    return;
                }

                float inputValue =
                Float.parseFloat(text.getText().toString());
                if (celsiusButton.isChecked()) {

                    text.setText(String.valueOf(konversiFahrenheitToCelsius(inputValue)));
                    celsiusButton.setChecked(false);
                    fahrenheitButton.setChecked(true);
                } else {

                    text.setText(String.valueOf(konversiCelsiusToFahrenheit(inputValue)));
                    fahrenheitButton.setChecked(false);
                    celsiusButton.setChecked(true);
                }
                break;
            }
        }

        // konversi ke celsius
        private float konversiFahrenheitToCelsius(float fahrenheit) {
            return ((fahrenheit - 32) * 5 / 9);
        }

        // konversi ke fahrenheit
        private float konversiCelsiusToFahrenheit(float celsius) {
            return ((celsius * 9) / 5) + 32;
        }
    }
}
```

- Class **KonverterSuhuActivity** diatas merupakan sebuah class turunan dari class **Activity**. Metode **onCreate** pada class diatas akan dipanggil ketika activity ini ingin dijalankan (ditampilkan di layar). Yang perlu kita perhatikan disini adalah pada baris **setContentView(R.layout.main)**; Artinya, **file xml** yang akan digunakan menjadi tampilan activity ini adalah pada file dengan nama **main.xml** dalam folder **res/layout/** pada project kita.
- Kemudian edit file **main.xml** seperti contoh dibawah. File inilah yang mengatur tampilan, sesuai dengan parameter yang diberikan ke metode **setContentView()**.

```

<?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:background="@color/myColor"
    android:orientation="vertical" >

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

    <EditText
        android:id="@+id/editText1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal|numberSigned" >
    </EditText>

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

        <RadioButton
            android:id="@+id/radio0"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:checked="true"
            android:text="@string/celsius" >
        </RadioButton>

        <RadioButton
            android:id="@+id/radio1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/fahrenheit" >
        </RadioButton>
    </RadioGroup>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="myClickHandler"
        android:text="@string/calc" >
    </Button>

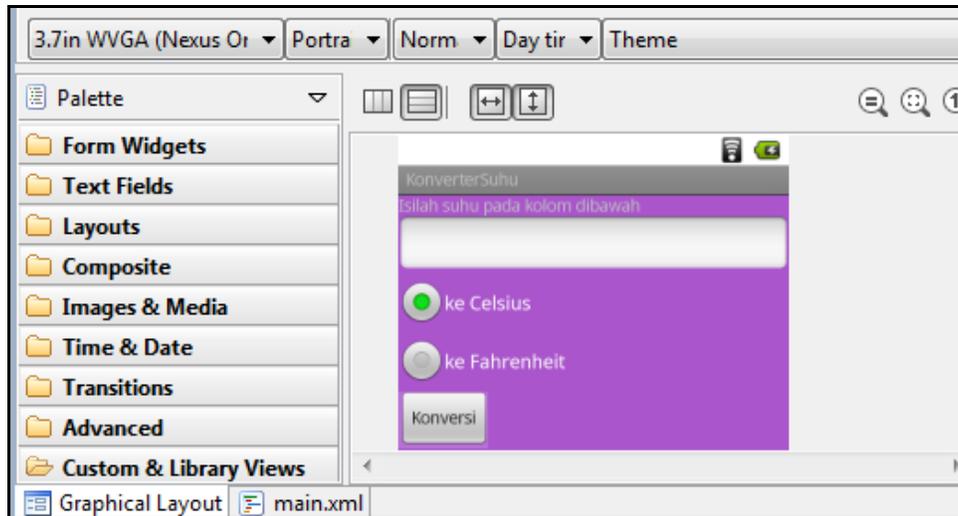
</LinearLayout>

```

- LinearLayout** merupakan format layout yang paling sering digunakan dalam aplikasi Android. Format ini menempatkan menampilkan komponen UI (button, textview, dll) secara berurutan berdasarkan penulisannya. Seperti namanya, **TextView** digunakan untuk menampilkan text. Text yang ditampilkan langsung ditulis atau merupakan referensi ke variable string dalam **res/values/strings.xml**.

Dalam contoh di atas, text yang akan ditampilkan adalah string dengan nama "hello".

8. Pada Graphical Layout file *main.xml* akan terlihat seperti contoh berikut:



9. Kemudian edit file *string.xml* tempat mendeklarasikan beberapa string, seperti contoh berikut:

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

    <color name="myColor">#AA55CC</color>
    <string name="app_name">KonverterSuhu</string>
    <string name="hello">Isilah suhu pada kolom dibawah</string>
    <string name="myClickHandler">myClickHandler</string>
    <string name="celsius">ke Celsius</string>
    <string name="fahrenheit">ke Fahrenheit</string>
    <string name="calc">Konversi</string>

</resources>
```

10. Jika proses edit telah selesai semua, maka click **Run**, lalu tunggu beberapa saat, maka pada emulator akan terlihat seperti contoh berikut:



Cara kerja aplikasi diatas adalah sebagai berikut:

1. Ketika pertama kali aplikasi dijalankan (activity dibuat), method *onCreate()* dieksekusi. Di dalam method ini, terdapat pengaturan tampilan layar dan pengambilan ID dari text field.
2. Dalam Android, tampilan text field direpresentasikan oleh class *editText1*. Setelah method *onCreate()* dieksekusi, selanjutnya Android mengeksekusi method *onResume()*.
3. Karena activity baru diluncurkan pertama kalinya, belum ada nilai yang tersimpan di *inputValue* sehingga variabel *text* bernilai null, dan jika akan diisi dengan cara meng-click didalam *textfield* maka akan ditampilkan keypad Android pada sisi bawah.
4. Ketika aplikasi akan dieksekusi dengan memilih radio button “ke Fahrenheit”, maka nilai yang diisikan kedalam *text* akan dikonversikan ke Fahrenheit. Selanjutnya dengan meng-click button “konversi” maka inilah saat yang pas untuk menyimpan nilai dari textfield atau menyimpan konten dari textfield.
5. Jika kita lihat kode di dalam *KonverterSuhuActivity.java*, maka kita akan melihat proses penyimpanan konten *text* dan diberikan ke method *konversiCelsiusToFahrenheit()* untuk dikonversikan. Selanjutnya hasil kalkulasi akan disimpan dalam *text* sebagai nilai hasil kalkulasi, dan nilai *text* inilah yang selanjutnya ditampilkan ke *textfield*.