

# Percobaan 9

## Basic Widget

### 9.1 Pokok Bahasan

- XML *Layout*
- *Basic Widget*

### 9.2 Tujuan

Dengan praktikum ini mahasiswa diharapkan dapat:

- Memahami pembuatan *user interfaces* dengan menggunakan XML *layout*
- Memahami *handling user interface event*

### 9.3 Software yang dibutuhkan

- Jdk
- ADT
- Eclipse
- SDK

### 9.4 Dasar Teori

GUI Android merupakan objek-objek dari kelas View dan ViewGroup. Kelas View adalah dasar dari sub kelas yang disebut “*widgets*” sedangkan ViewGroup, melayani untuk sub kelas yang disebut “*layouts*”.

#### A. *Basic Widget*

Pada dasarnya *widget* memiliki fungsi yang sama yaitu untuk berinteraksi dengan pengguna. Tujuannya yaitu untuk memberikan suatu proses pada aplikasi hasil dari aksi yang dilakukan pengguna (memilih menu, *setting*, menekan tombol *Start* atau *Exit*). Namun tipe dari interaksi dengan pengguna itu terdapat beberapa macam yaitu untuk melakukan suatu proses, untuk melakukan suatu pengeditan, ataupun untuk menampilkan informasi saja.

## TextView

Pada android terdapat *class* khusus untuk menampilkan *Text* yaitu menggunakan *class* *TextView*, pada aplikasi “Hello World” yang kita buat kita menggunakan *class* ini untuk menampilkan *text*-ya. sebagai contoh penggunaan *TextView* kita akan menggunakan beberapa fitur seperti penggunaan *Custom Font*, *text color*, *text size* dll.

Tabel 1. *Properties* yang bisa digunakan pada *TextView*

Text	digunakan untuk mengeset text yang akan ditampilkan
Text Color	digunakan untuk mengeset warna text e.g: #FF0000 (warna merah)
Text Size	digunakan untuk mengeset ukuran text yang digunakan, disarankan untuk menggunakan satuan "sp" e.g: 20sp
Text Style	digunakan untuk mengeset tampilan text apakah dibold atau diitalic atau tampilan standar dan bisa kombinasinya
Gravity	digunakan untuk mengatur tata letak/alignment penulisan text terhadap ukuran objek <i>TextView</i>
Layout Gravity	digunakan untuk mengatur tata letak/alignment penulisan text terhadap ukuran objek <i>ViewGroup/Layout</i> nya dan terhadap objek <i>View</i> lainnya

## Button

Button adalah *subclass* dari *TextView* dan biasanya digunakan untuk melakukan aksi clicking pada GUI. Adapun format dari tampilan *button* hampir mirip dengan *TextView*.



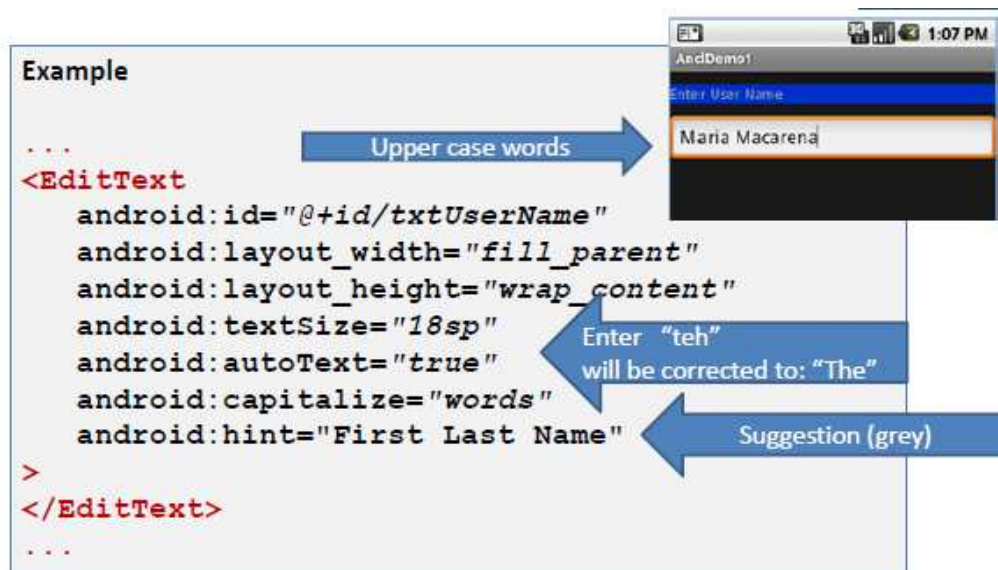
Gambar 1. Contoh tampilan *Button*

## EditText

*EditText* adalah perluasan dari *TextView* yang bisa di-update. Fungsi yang biasanya sering digunakan adalah:

```
TextBox.setText("nilai tertentu")
```

```
TextBox.getText().toString()
```



Gambar 3. Contoh tampilan *EditText*

## CheckBox

*CheckBox* adalah tipe yang spesifik dari *button* yang bisa *checked* atau *unchecked* salah satu atau beberapa *button* sekaligus.



Gambar 4. Contoh tampilan *CheckBox*

**RadioButton**

RadioButton adalah tipe spesifik dari button yang bisa *checked* atau *unchecked* salah satu saja, dan tidak bisa beberapa button sekaligus.



Gambar 5. Contoh tampilan *radiobutton* dan *checkbox*

## B. XML Layout

Sebuah *layout* merupakan arsitektur UI(*User Interface*) untuk suatu *Activity* atau *Widget*. Android mengatur *layout* melalui *ViewGroup* dan *Layout-Parameter Object*. *ViewGroup* adalah suatu *View* spesial yang dapat berisi *view-view* lainnya (disebut *child view*) dan juga memberikan akses ke *Layout*. Walaupun *layout* tidak benar-benar menggambar sesuatu pada layar, tetapi *layout* merupakan *parent container* dari *control* yang akan kita gunakan dalam membuat *user interface*(UI).

## 9.5 Tugas Pendahuluan

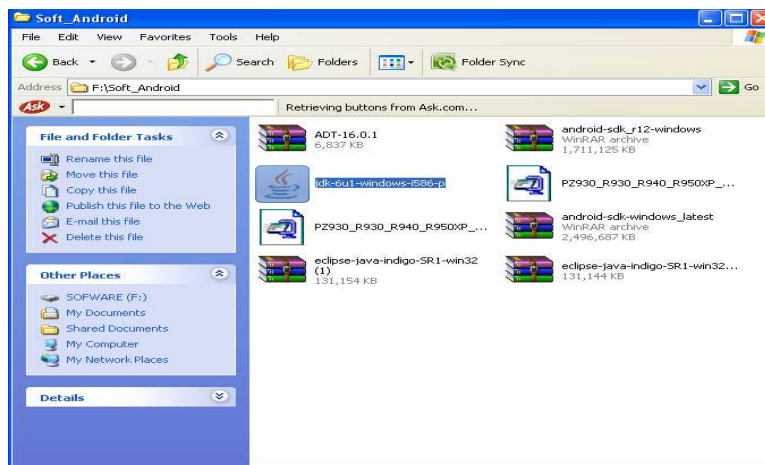
1. Apa yang dimaksud dengan widget?
2. Apa kegunaan dari XML Layout!

## 9.6 Percobaan

### A. Instalasi Android

Langkah-langkah untuk instalasi android meliputi instalasi jdk-6u1-windows-i586-p,serta instalasi eclipse.

#### I. Instalasi jdk-6u1-windows-i586-p



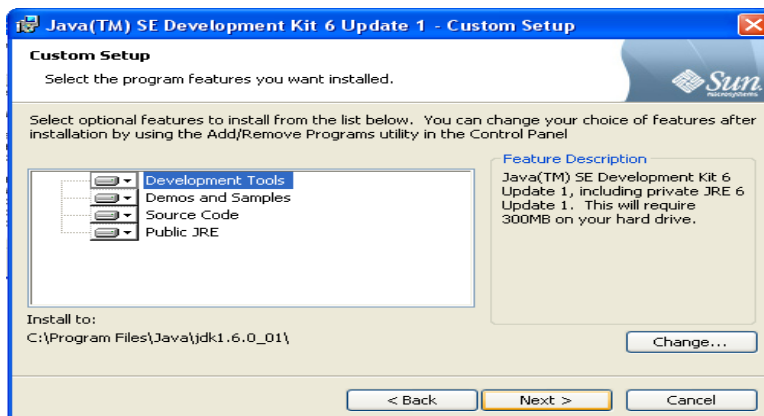
Gambar 1. Icon jdk-6-ul-windows-i586-p

Klik *accept* pada *license agreement*



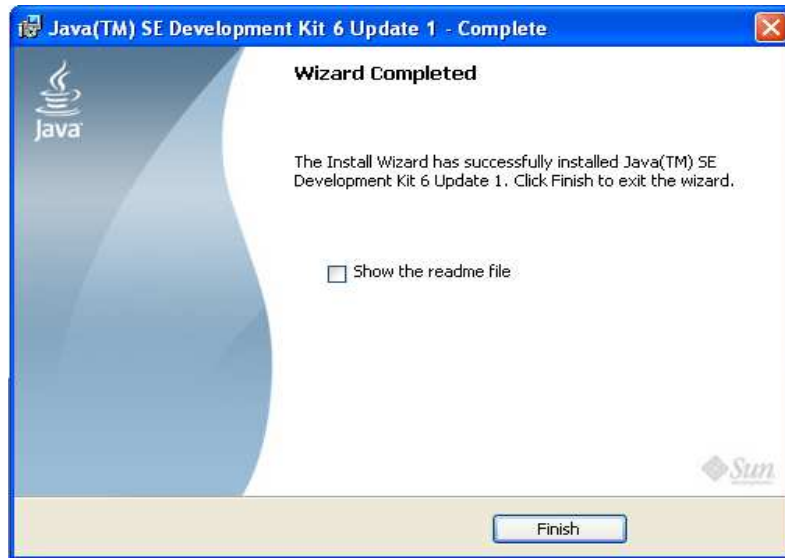
Gambar 2. License agreement

Pada proses instalasi klik *next*



Gambar 2. Proses Instalasi

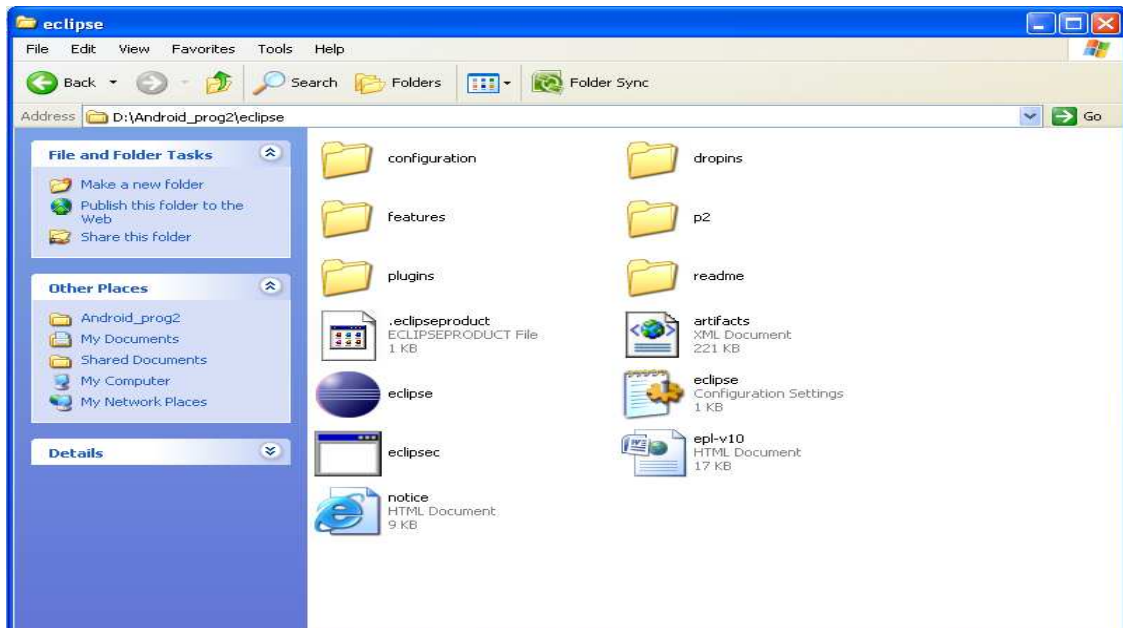
Setelah proses instalasi selesai klik *finish*



Gambar 3. Proses Instansi selesai

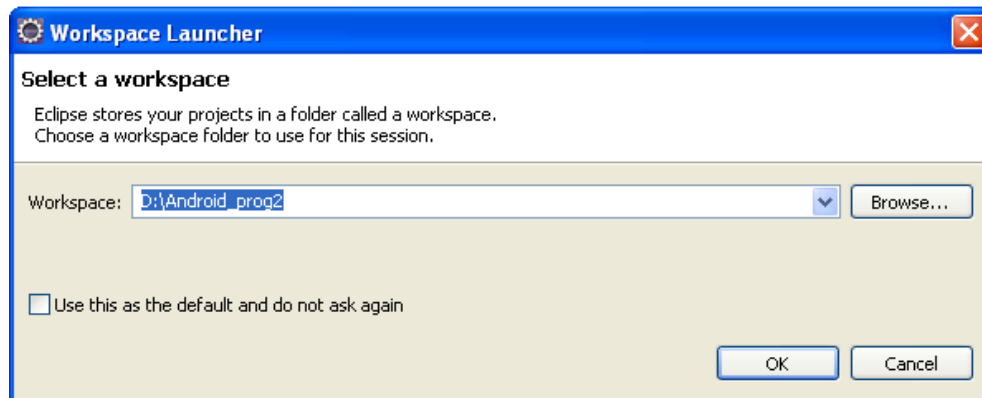
## II. Instalasi eclipse

Untuk melakukan instalasi ini, dibutuhkan 3 komponen pendukung yang meliputi eclipse, ADT dan SDK. *Extract* file eclipse dan SDK kemudian lakukan langkah-langkah instalasi. Jalankan eclipse dengan cara *double* klik *icon* eclipse seperti pada Gambar 4.



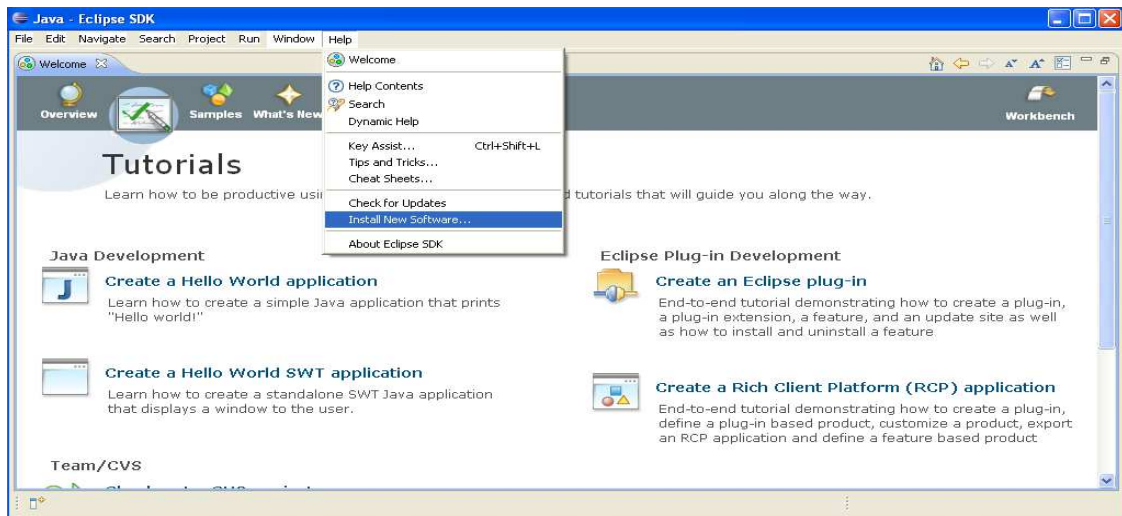
Gambar 4. *Icon* eclipse

Setelah muncul tampilan *Select a workspace*, maka dipilih tempat untuk menyimpan *Workspace* atau *folder* yang diinginkan. Kemudian klik **ok**



Gambar 5. *Workspace launcher*

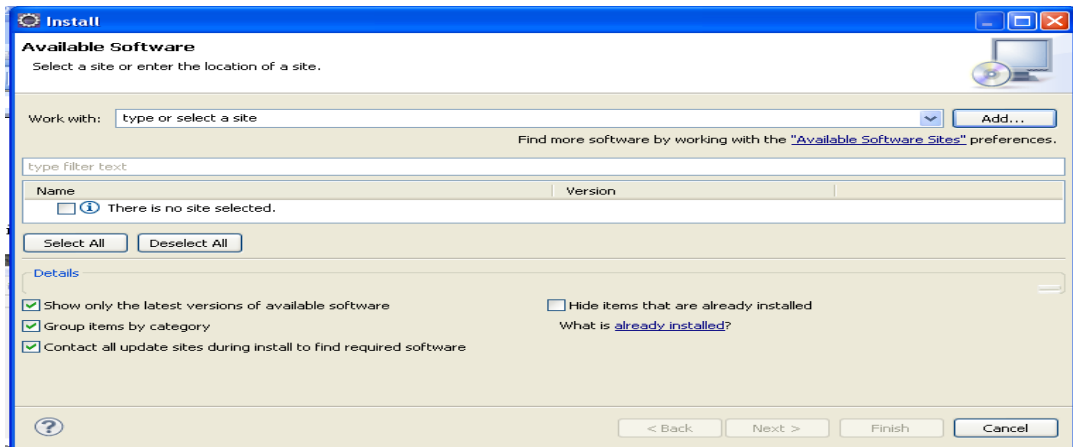
Pada menu *help* pilih *install new software*



Gambar 6. *Install new software*

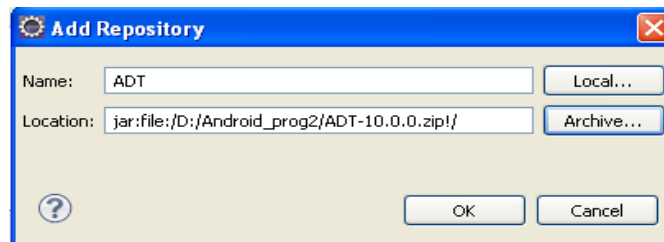


Pada pilihan *available software* klik *add* yang terletak di pojok kanan atas.



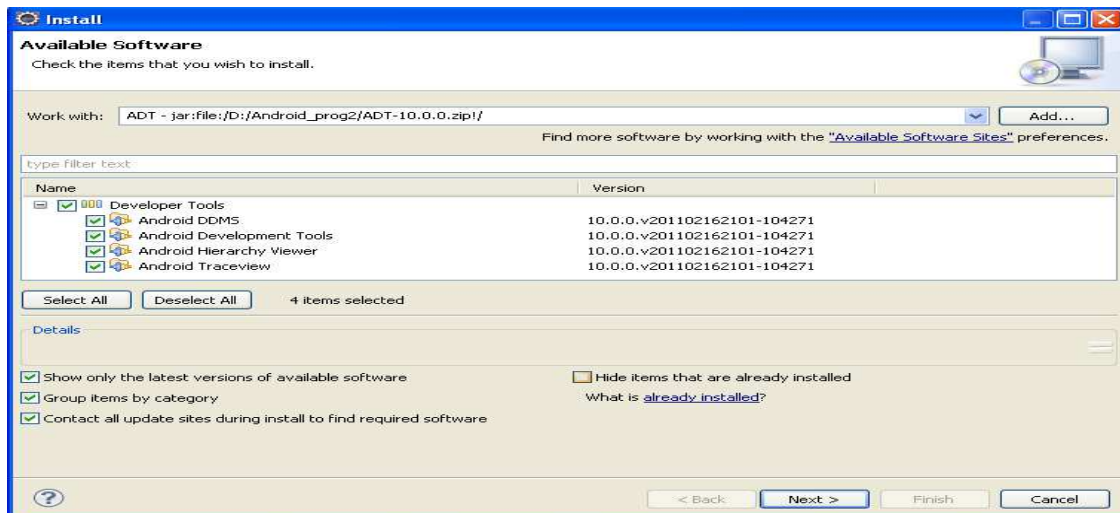
Gambar 7. *Available software*

Pada tampilan *Add Repository* isikan *Name* : **ADT** serta tambahkan lokasi dimana *file* ADT berada, kemudian klik *ok*.



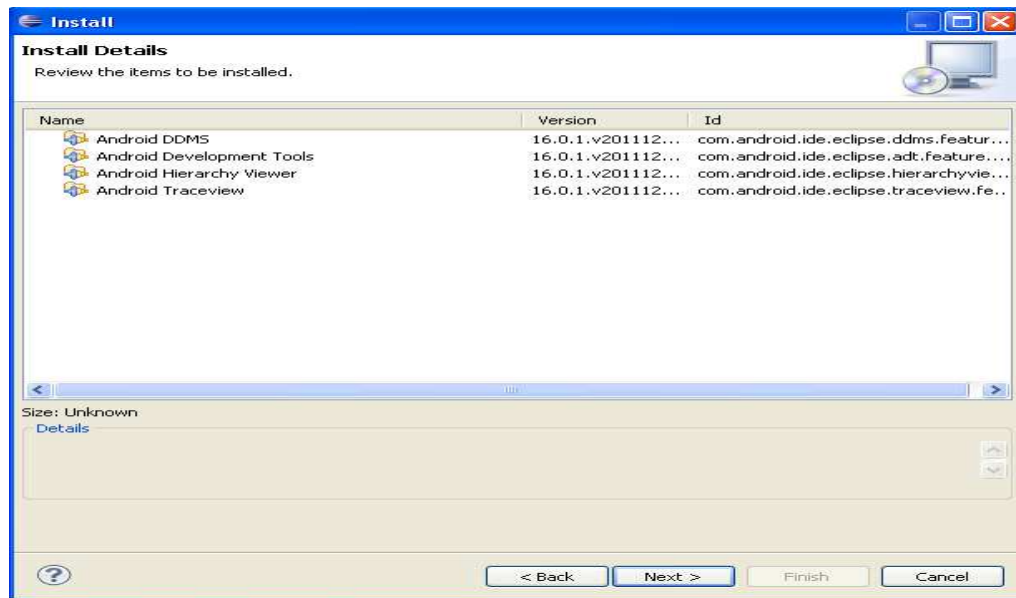
Gambar 8. *Add Repository*

Pada tampilan *Available Software* centang pada item *Developer Tools* lalu klik *Next*



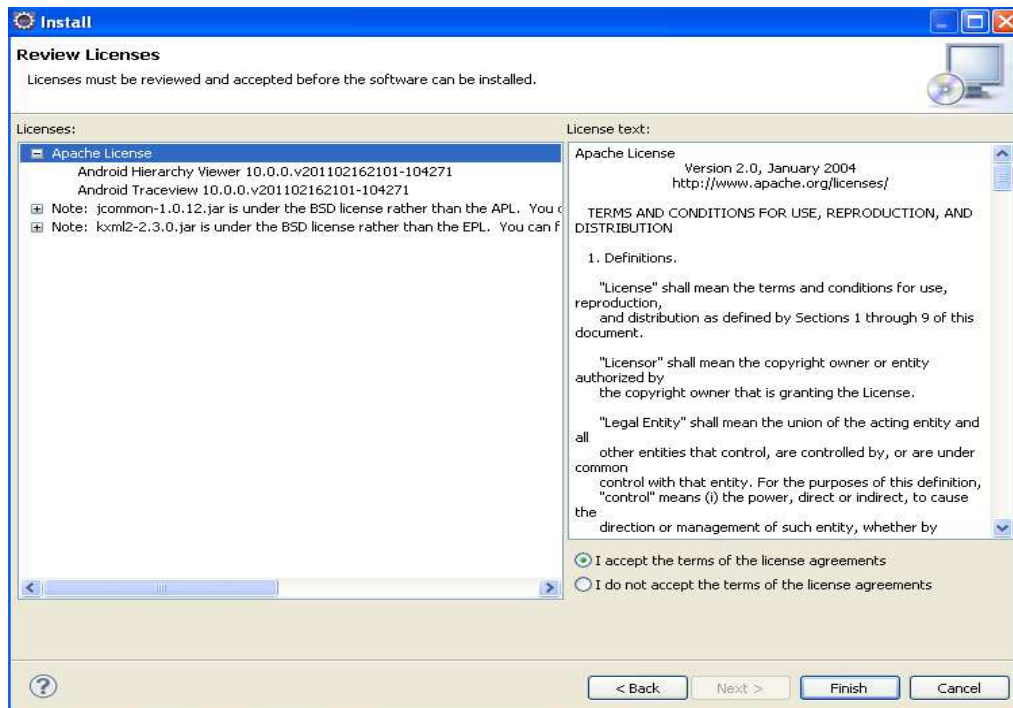
Gambar 9. *Developer Tool*

Pada *install detail* klik *next*



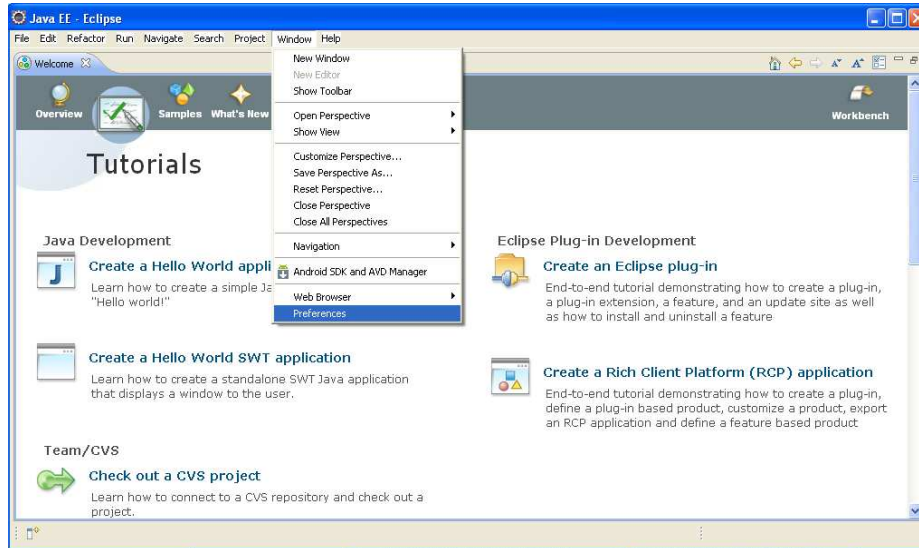
Gambar 10. Install Detail

Pada *review license* pilih *accept*, kemudian *finish* jika telah selesai.



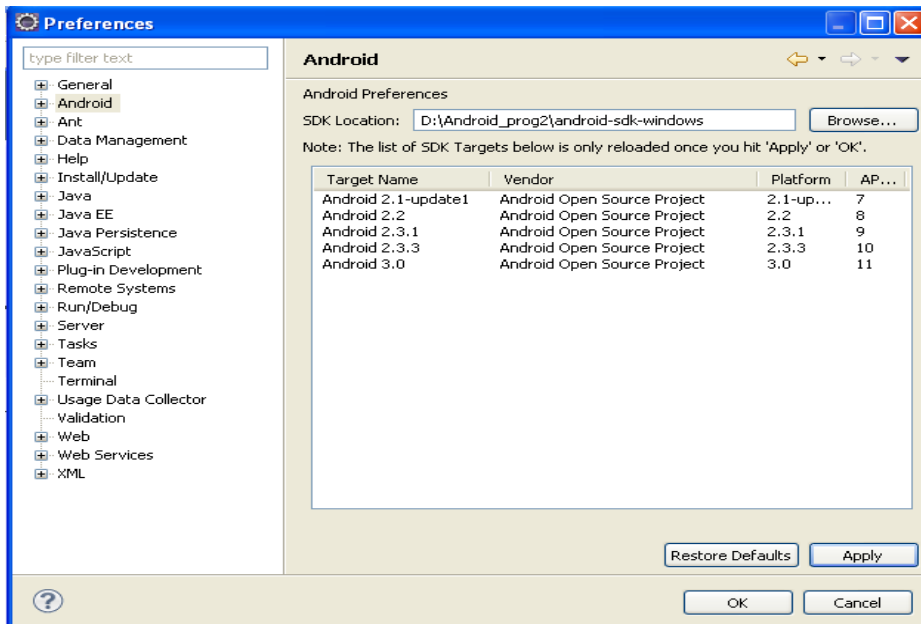
Gambar 11. Review License

Setelah proses instalasi selesai, maka *restart eclipse*. Kemudian lakukan *Setting plug in SDK*, pada tampilan Java EE-Eclipse pilih *preferences*.



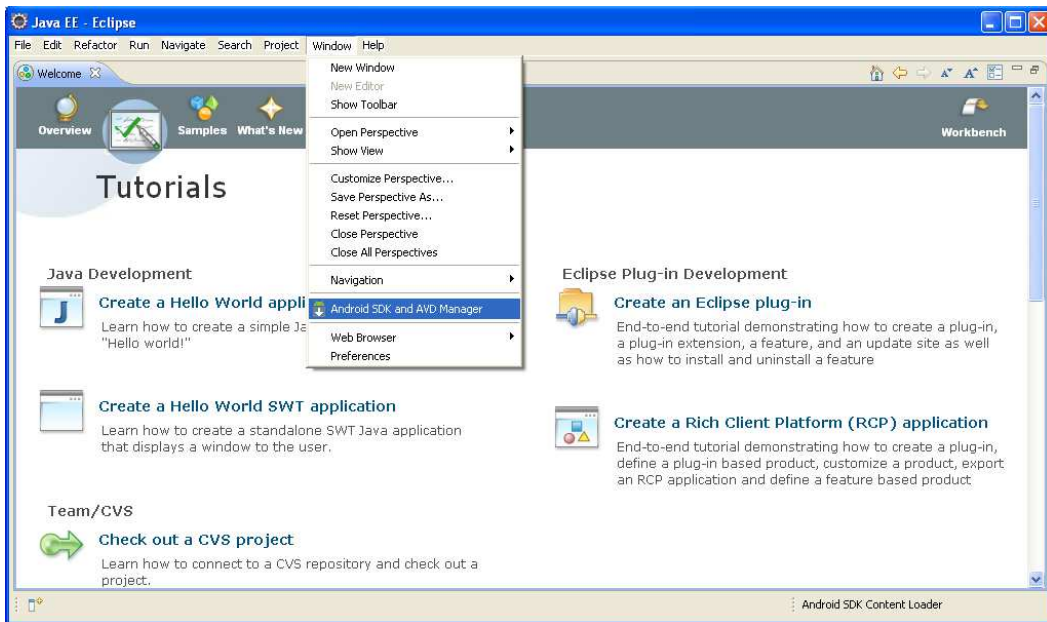
Gambar 12. Java EE-Eclipse

Pada tampilan *preferences* pilih *Android* lalu masukan file *android-sdk-windows* kemudian pilih *Apply* dan ok.



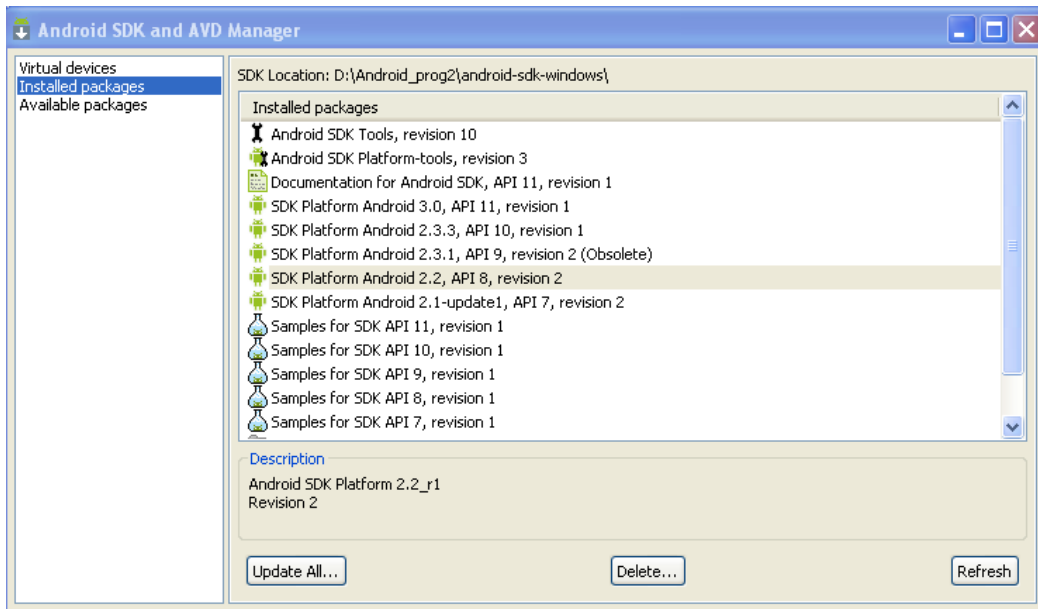
Gambar 13. Android Preferences

Pada tampilan **Java EE-Eclipse** pilih **Android SDK dan AVD Manager**



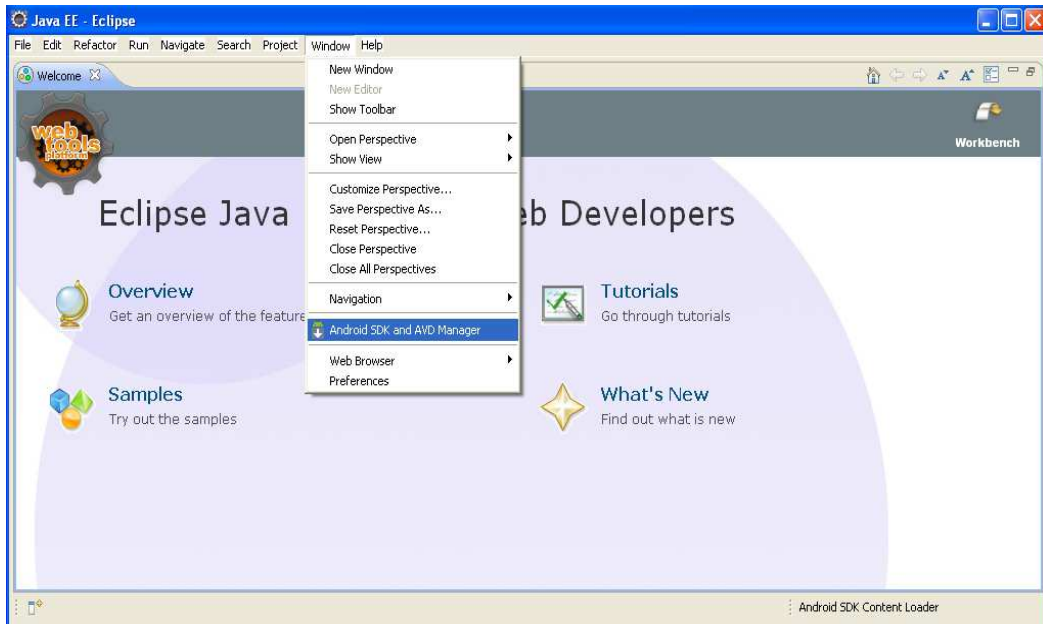
Gambar 14. Java EE-Eclipse

Pada tampilan **Android SDK dan AVD Manager** pilih **Installed packages** untuk mengecek *plug in* SDK .



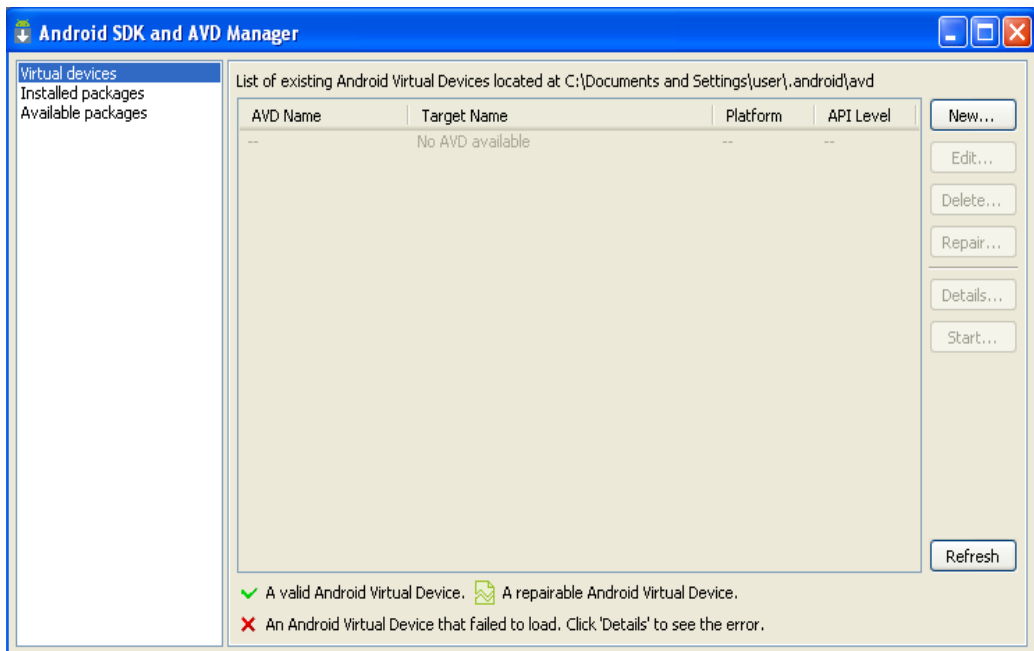
Gambar 15. Android SDK dan AVD Manager

Untuk pembuatan *Android Virtual device*, Pada tampilan Java EE-Eclipse dipilih **Android SDK dan AVD Manager**.



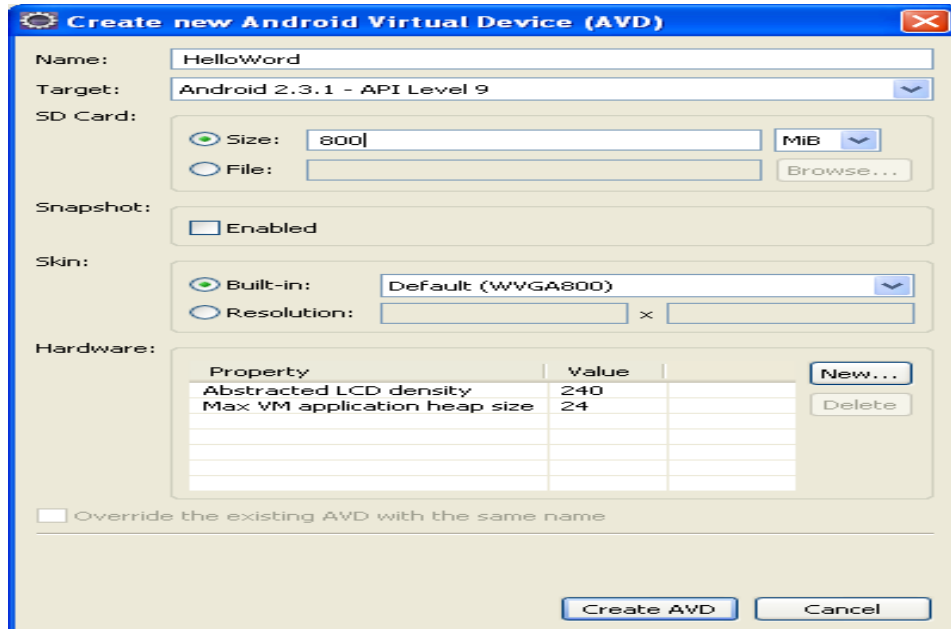
Gambar 16. Java EE-Eclipse untuk pembuatan Android Virtual Device

Pada tampilan **Android SDK dan AVD Manager** pilih *new*



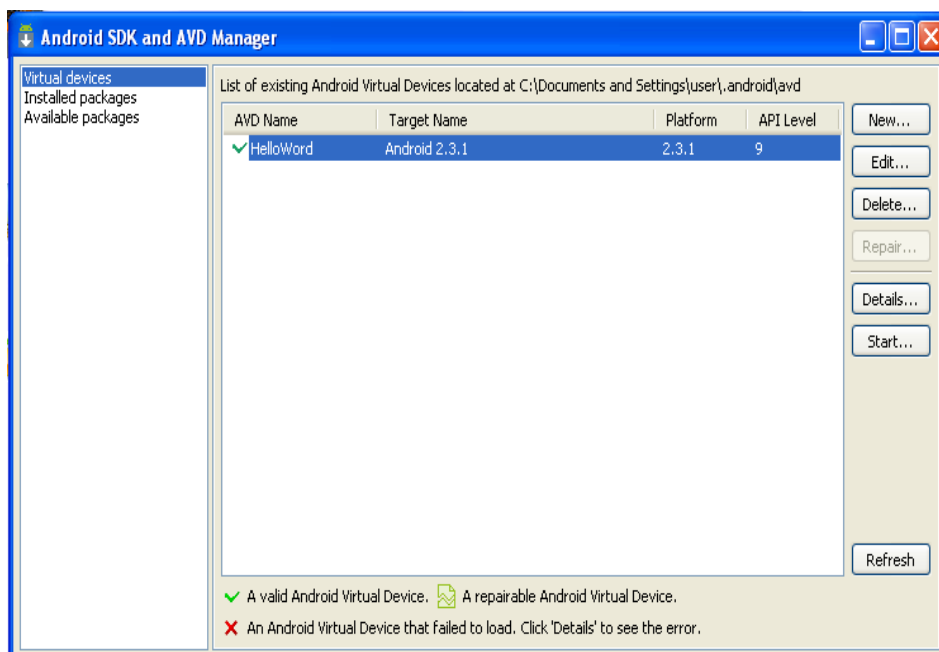
Gambar 17. Android SDK dan AVD Manager untuk pembuatan *virtual device*

Pada tampilan *Create new AVD* dipilih *Name* dan *Target* kemudian dipilih *Create AVD*

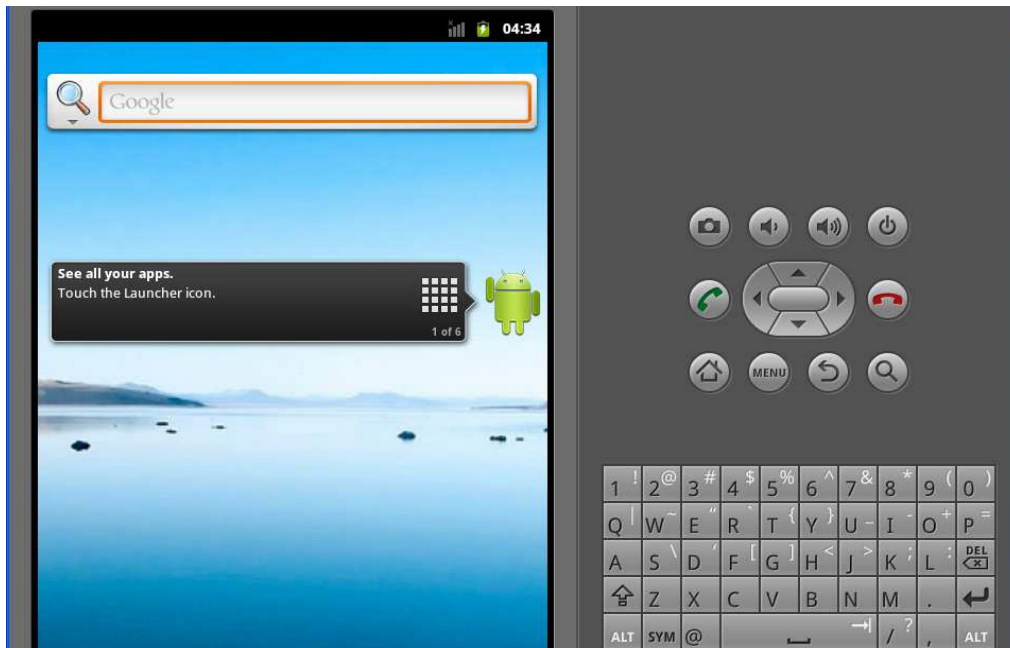


Gambar 18. Pemberian nama dan target untuk pembuatan *virtual device*

Pada tampilan android SDK dan AVD Manager pilih AVD Name yang telah dibuat kemudian klik *start* dan *launch*



Gambar 19. Proses *start* dan *launch*

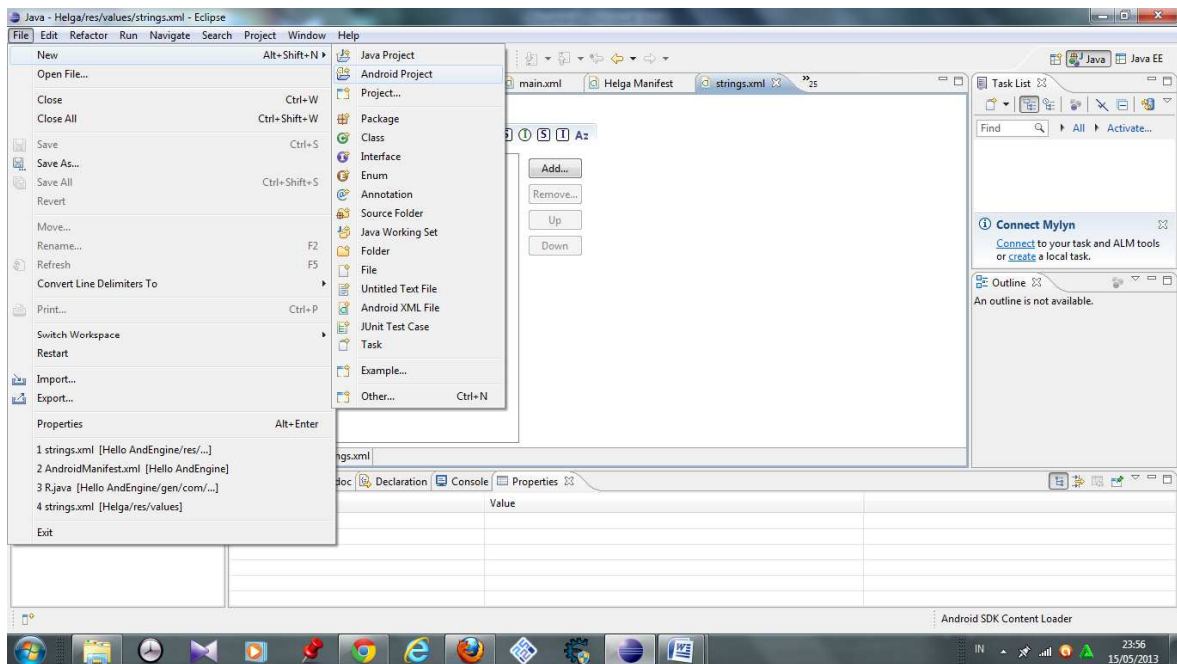


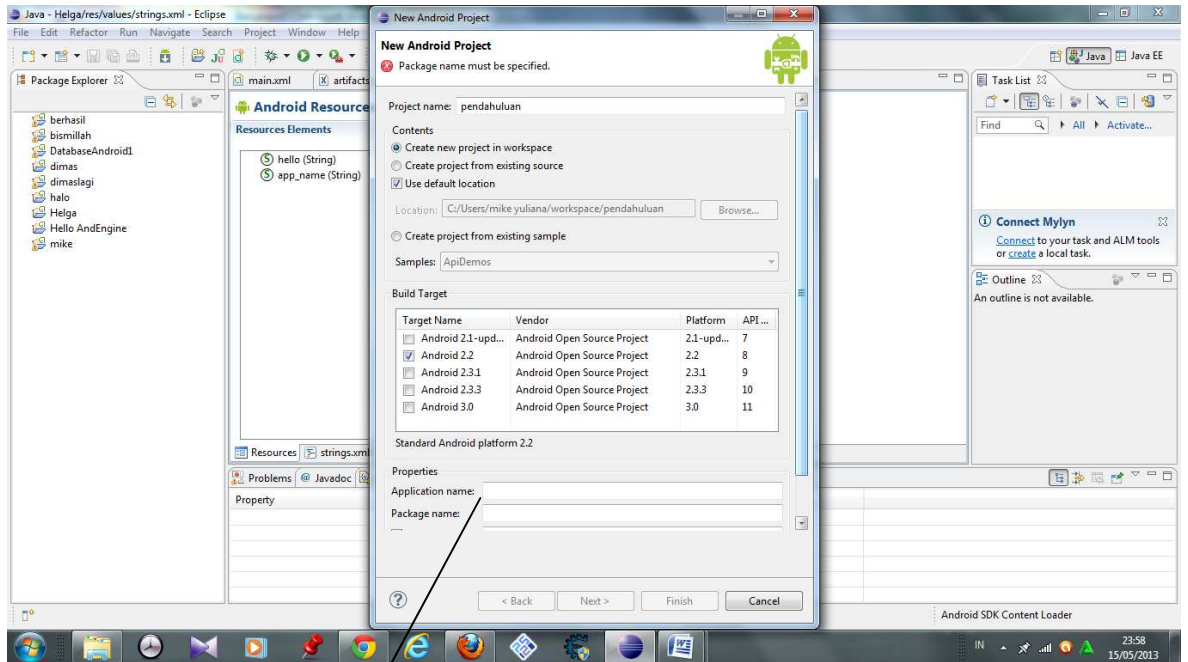
Gambar 20. Android *Virtual Device* siap digunakan

## B. Pembuatan Android Project

1. Buatlah program Hello world, dengan mengikuti langkah-langkah dibawah ini:

- Pilih File → new → Android Project





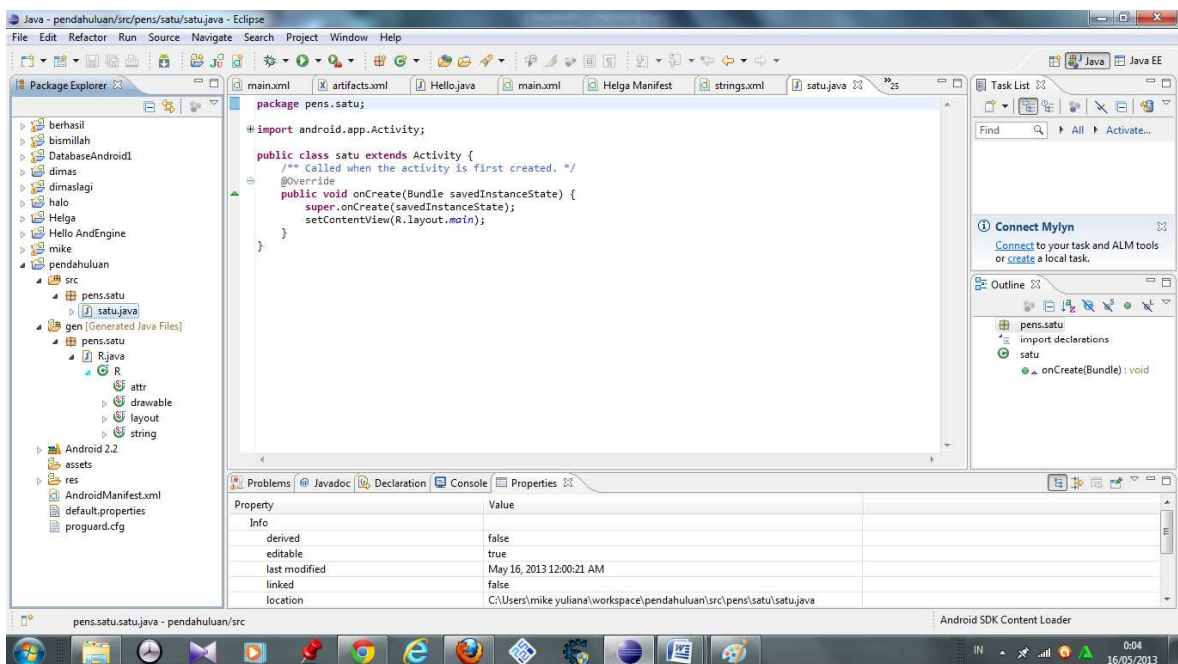
Properties

Application name:

Package name:

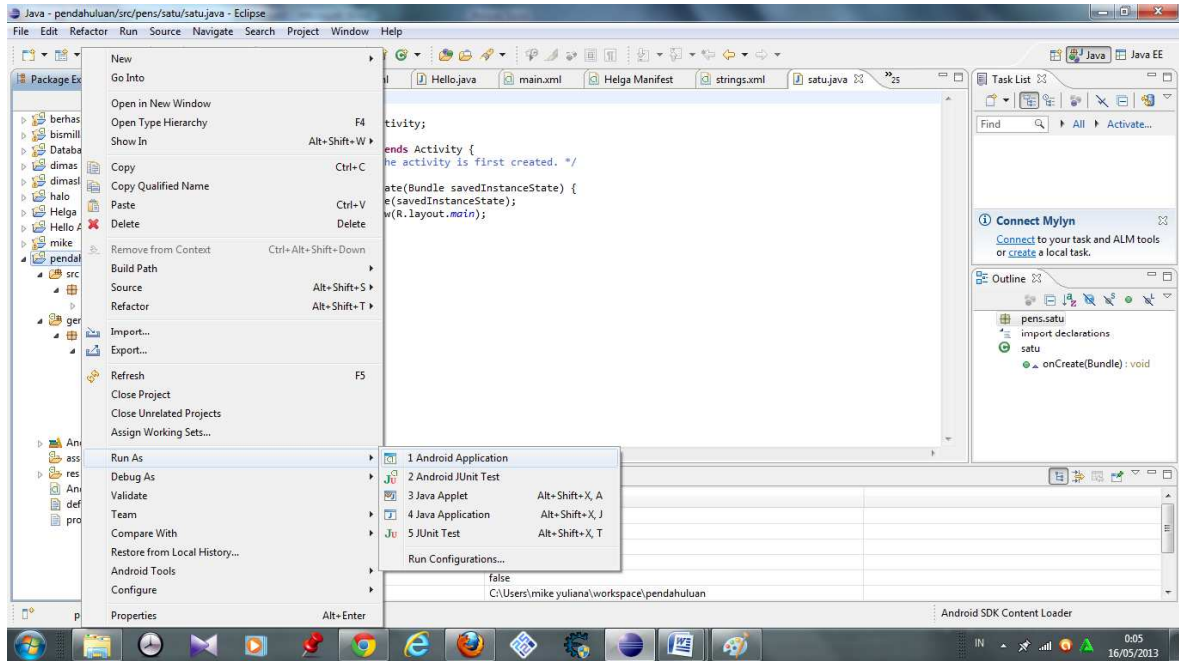
Create Activity:

Min SDK Version:

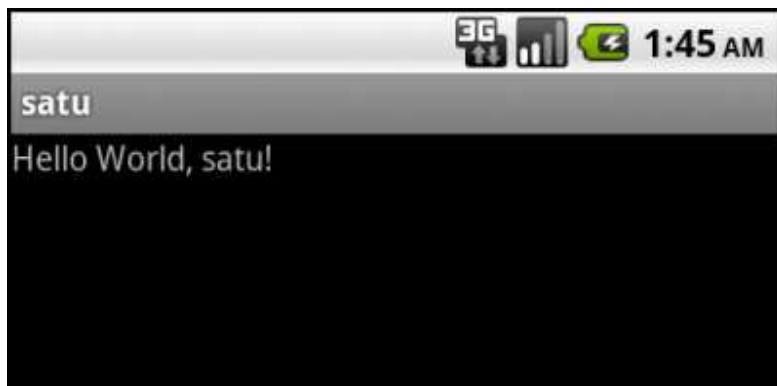




- Untuk menjalankan project, klik kanan pada project dan run as → android application



Sehingga didapat tampilan seperti dibawah ini:



Rubahlah bagian main.XML dan tunjukkan tampilan yang dihasilkan!

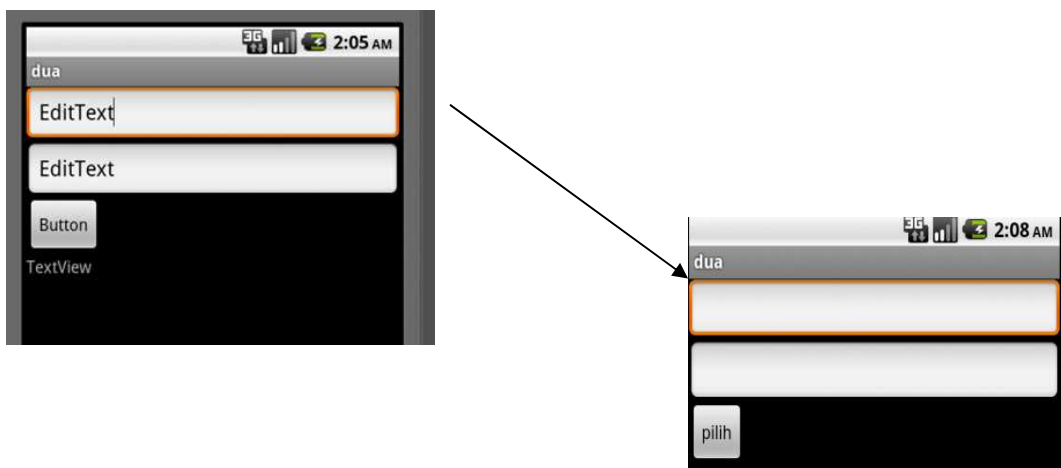
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"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

2. Buatlah program perkalian dengan mengikuti langkah-langkah dibawah ini:

Pada bagian main.xml pilih Graphical Layout



Pilih EditText, Button serta TextView



### main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<EditText android:id="@+id/editText1"
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:text="">
</EditText>
<EditText android:id="@+id/editText2"
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:text="">
</EditText>
<Button android:text="pilih"
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</Button>
<TextView android:text=""
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</TextView>
</LinearLayout>
```

### dua.java

```
package pens.dua;

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

```

public class dua extends Activity {
    EditText harga, jumlah;
    Button hitung;
    TextView total;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        harga=(EditText) findViewById(R.id.editText1);
        jumlah=(EditText) findViewById(R.id.editText2);
        hitung=(Button) findViewById(R.id.button1);
        total=(TextView) findViewById(R.id.textView1);
        hitung.setOnClickListener(new klik());
    }

    class klik implements Button.OnClickListener{
        String harga1,jumlah1;
        int harga_ok,jumlah_ok,total_ok;
        public void onClick(View v){
            harga1=harga.getText().toString();
            jumlah1=jumlah.getText().toString();
            harga_ok=Integer.parseInt(harga1);
            jumlah_ok=Integer.parseInt(jumlah1);
            total_ok=harga_ok*jumlah_ok;
            total.setText(total_ok+"");
        }
    }
}

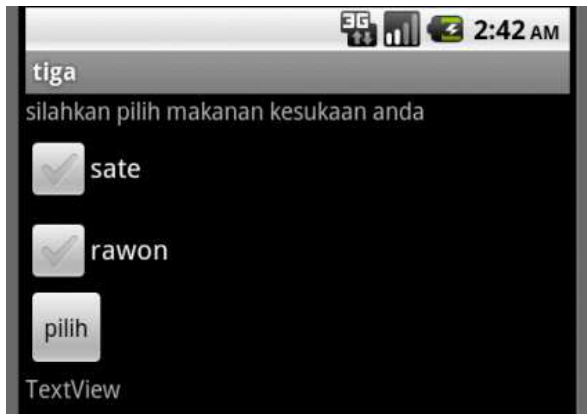
```

Tampilan yang dihasilkan:



3. Buatlah program pemilihan menu makanan dengan mengikuti langkah-langkah dibawah ini:

Pilih CheckBox, TextView dan Button



#### main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="silahkan pilih makanan kesukaan anda";
    />
<CheckBox android:text="sate"
    android:id="@+id/checkBox1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</CheckBox>
<CheckBox android:text="rawon"
    android:id="@+id/checkBox2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</CheckBox>
<Button android:text="pilih"
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</Button>
<TextView android:text="TextView"
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</TextView>
</LinearLayout>
```

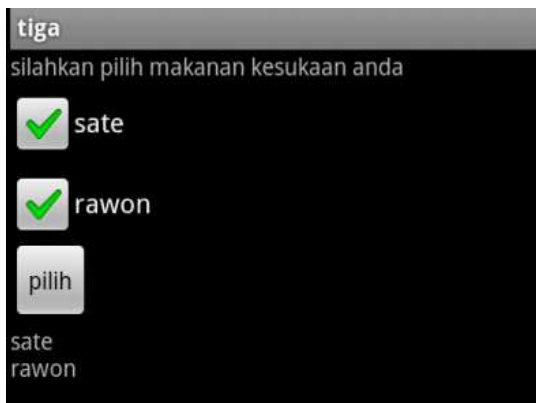
tiga.java

```
import pens.tiga.R;
import pens.tiga.tiga.klik;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.TextView;

public class tiga extends Activity {
    CheckBox sate, rawon;
    Button pilih;
    TextView hasil;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        sate=(CheckBox) findViewById(R.id.checkBox1);
        rawon=(CheckBox) findViewById(R.id.checkBox1);
        pilih=(Button) findViewById(R.id.button1);
        hasil=(TextView) findViewById(R.id.textView1);
        pilih.setOnClickListener( new klik());
    }

    class klik implements Button.OnClickListener{
        String a="";
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            if(sate.isChecked())
                a=a+"sate\n";
            if(rawon.isChecked())
                a=a+"rawon\n";
            hasil.setText(a);
        }
    }
}
```

Tampilan yang dihasilkan:



### 9.7 Tugas Laporan Resmi

1. Rubahlah CheckBox pada percobaan 3 menjadi RadioButton, amati perbedaan yang terjadi !
2. Buatlah program sehingga didapatkan tampilan seperti dibawah ini:

