

## Mobile Programming dengan Android

Yudi Wibisono ([yudi@upi.edu](mailto:yudi@upi.edu) atau [yudi1975@gmail.com](mailto:yudi1975@gmail.com) )

Ilmu Komputer Universitas Pendidikan Indonesia ([cs.upi.edu](http://cs.upi.edu))

versi (BETA): 11 April 2011



<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Modul ini bebas dicopy, didistribusikan, ditransmit dan diadaptasi/modifikasi/diremik dengan syarat tidak untuk komersial, pembuat asal tetap dicantumkan dan hasil modifikasi dishare dengan lisensi yang sama.

TBD PERBAIKAN MODUL: .....	2
Pengantar Tentang Modul.....	3
Sekilas Android.....	3
Fitur-fitur.....	3
Instalasi Lingkungan Pengembangan .....	4
Aplikasi Pertama: Hello World.....	6
Aplikasi Kedua: Hello xxx.....	10
Widget.....	17
Text View.....	17
Button.....	18
ImageButton dan ImageView .....	18
EditText.....	20
CheckBox.....	21
RadioButton .....	22
ListView.....	25
Dialog.....	26
TBD.....	26
Layout Manager .....	27
RelativeLayout.....	27
TableLayout .....	30
Frame Layout.....	32
Kombinasi Layout.....	33
Sub Activity .....	35
Menyimpan Data.....	45
Shared Preferences.....	45
SQLite .....	46
File .....	51

## TBD PERBAIKAN MODUL:

- perlu ditambahkan sebelum membahas widget, bahwa semua widget merupakan turunan dari view, sehingga atribut yang dimiliki view juga diturunkan (terutama background) juga width dan height, wrap\_content, fill\_parent, padding
- ListView
- Dialog
- Advanced:
  - SeekBar
  - ScrollView
  - TabHost , TabWidget
  - ViewPager, SlidingDrawer
- Menu
- Intent

## Pengantar Tentang Modul

Modul praktikum ini dibuat untuk praktikum matakuliah Mobile Computing di Ilmu Komputer UPI. Diasumsikan pengguna modul ini telah mengenal dasar-dasar bahasa Java dan XML.

## Sekilas Android

Android adalah software platform yang open source untuk mobile device. Android berisi sistem operasi, middleware dan aplikasi-aplikasi dasar. Basis OS Android adalah kernel linux 2.6 yang telah dimodifikasi untuk mobile device.

Android versi 1.0 direlease 23 Sep 08, versi 1.1 pertama kali digunakan di mobile phone (T1). Versi 1.5 (Cupcake) muncul April 09 yang mendukung soft keyboard, 1.6 (Donut) Sept 09 dan versi terakhir:

- **2.0/2.1 (Eclair)** memperbaiki user interface, HTML5 dan Exchange ActiveSync 2.5.
- **2.2 (Froyo)**, mempercepat kinerja dengan Just In Time compiler dan Chrome V8 JavaScript engine, Wi-Fi hotspot tethering dan support Adobe Flash.
- **2.3 (Gingerbread)**, memperbaiki user interface, soft keyboard, copy/paste features dan support ke Near Field Communication (NFC).
- **2.4 (Gingerbread)**, an update to 2.3, akan direlease April 2011
- **3.0 (Honeycomb)**, untuk tablet yang menggunakan layar lebih besar, multicore processors dan hardware acceleration untuk graphics.
- **Ice-cream sandwich**, kombinasi 2.3 Gingerbread and 3.0 Honeycomb, akan muncul di pertengahan 2011.

## Fitur-fitur

Fitur-fitur android adalah sebagai berikut:

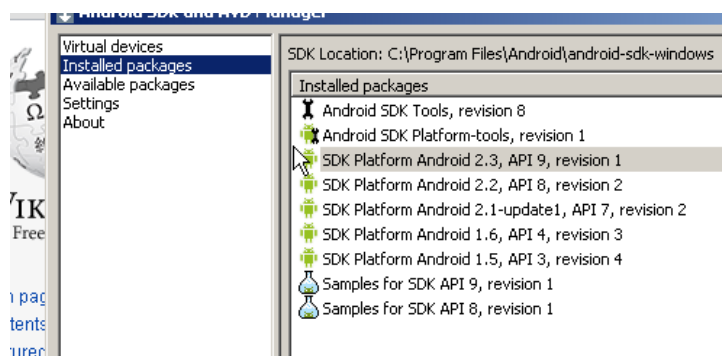
- **Menulis file** di SD card.
- **Database** dengan SQLite.
- **Application framework** berbasis komponen yang memudahkan reuse.
- **Dalvik virtual machine** dioptimisasi untuk menjalankan program Java di mobile devices
- **Integrated browser** berbasis WebKit engine
- **Optimized graphics** tersedia 2D graphics library; dan OpenGL ES 1.0 untuk 3D graphics
- **Media support** untuk MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF
- **GSM Telephony** fasilitas telepon.
- **Bluetooth, EDGE, 3G, dan WiFi**
- **Kamera, GPS, compass dan accelerometer**
- **Multitouch**

## Instalasi Lingkungan Pengembangan

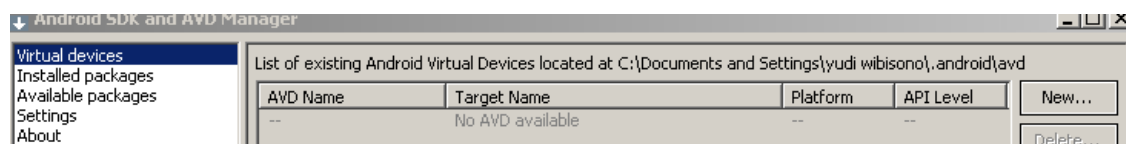
Untuk membuat app Android, dibutuhkan Android SDK (*Software Development Kit*) yang diperoleh di <http://developer.android.com/> sedangkan untuk IDE (*Integrated Development Environment*) dapat dipilih Netbeans, Eclipse atau bahkan langsung dengan command line. Dalam tutorial ini hanya digunakan IDE Eclipse.

Setelah Android SDK didownload dan diinstall, maka secara otomatis Android SDK dan AVD Manager akan dijalankan (atau melalui Start → Program → Android SDK Tools). Tersedia beberapa platform SDK dan dibutuhkan koneksi internet untuk mendownloadnya. Dalam modul ini yang digunakan adalah SDK Platform 2.3 API 9.

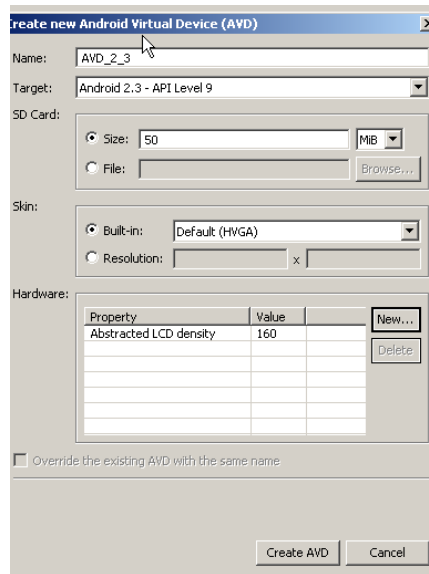
Untuk instalasi Android SDK tanpa koneksi internet, silahkan lihat: <http://yudiwbs.wordpress.com/2011/02/13/cara-menginstall-sdk-android-tanpa-internet/>



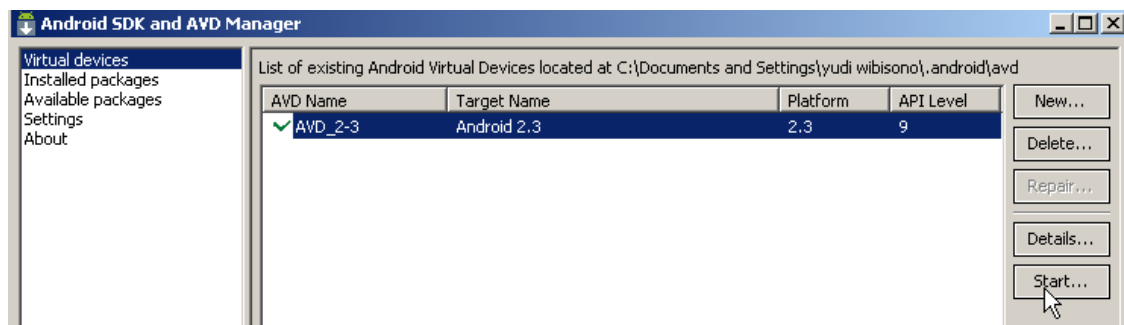
Setelah platform selesai di-install, selanjutnya akan dibuat Virtual Devices, emulator android yang dijalankan di desktop. Masih di SDK And AVD manager, pilih virtual devices, lalu New



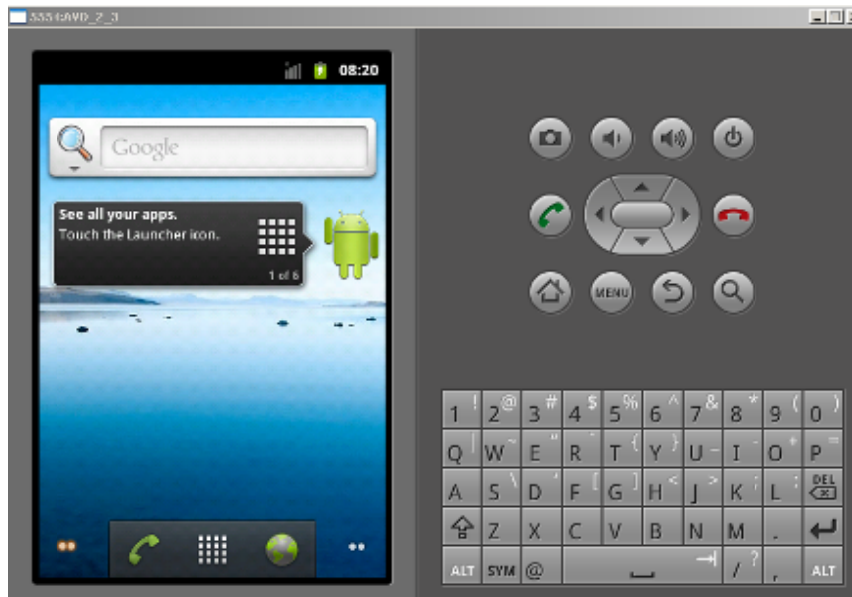
Isi nama, target dan ukuran SD card, lalu klik “Create AVD”



Pilih AVD yang baru Anda buat lalu klik Start dan kemudian Launch. Tunggu 1-5 menit (tergantung kemampuan komputer yang Anda miliki)

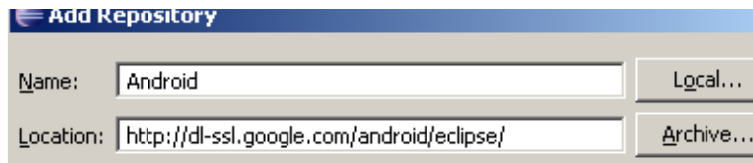


Emulator ini membutuhkan waktu cukup lama dan memori yang besar. Ambil secangkir kopi dan silahkan tunggu sampai layar berikut muncul (jika yang muncul masih tulisan Android, tunggu). Setelah emulator dijalankan, sebaiknya jangan ditutup lagi untuk menghemat waktu.



Terutama bagi yang belum memiliki Android, emulator ini dapat digunakan untuk bereksplorasi.

Dengan asumsi Anda telah memiliki JDK, langkah selanjutnya adalah menginstall Eclipse (<http://www.eclipse.org/downloads/>) dan dilanjutkan dengan plugin ADT (*Android Development Tools*). Untuk menginstall ADT, masuk ke Eclipse, pilih menu Help → Install New Software, klik ADD lalu isi seperti berikut

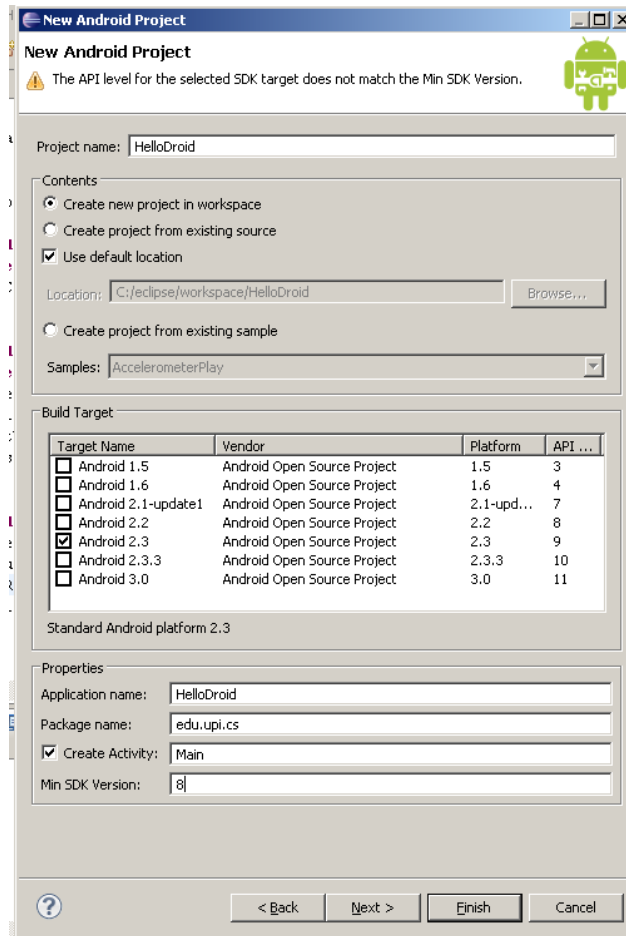


Kemudian ikuti petunjuk selanjutnya dan restart Eclipse.

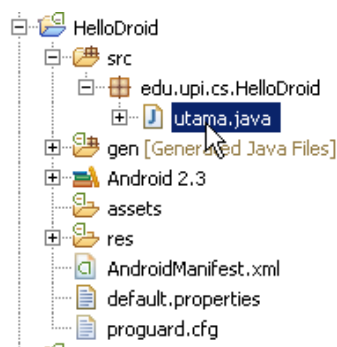
## Aplikasi Pertama: Hello World

Langkah pertama tentunya membuat aplikasi hello world terlebih dulu.

Untuk memulai, jalankan Eclipse, pilih File → New → Other (ctrl-N), pilihlah Android project.



Lihat dibagian Package Explorer (panel disebelah kiri Eclipse), ADT telah membuatkan direktory src, gen, Android, assest, res dan seterusnya (akan dijelaskan kemudian). Expand lalu klik utama.java



Terlihat ADT telah menyiapkan activity (anggap activity itu semacam Form). Untuk sementara kita tidak sentuh terlebih dulu.

```

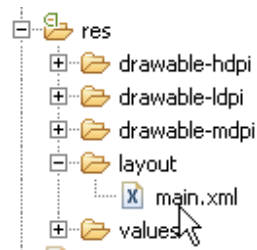
package edu.upi.cs.HelloDroid;

import android.app.Activity;

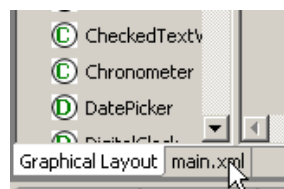
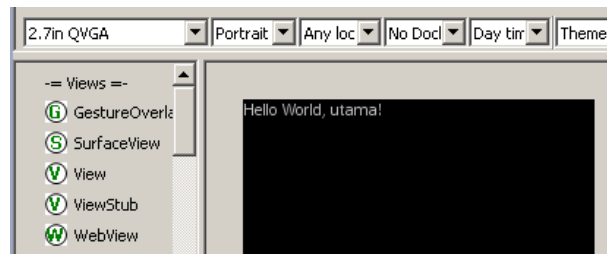
public class utama extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}

```

Kembali ke Package Explorer, lihat res → layout, klik main.xml. File xml ini berisi definisi layout komponen user interface, misalnya tombol, edit box dan seterusnya.



Secara default telah dibuatkan satu komponen yaitu text view. Klik tab main.xml



Ganti isi android:text menjadi “Hello Android”

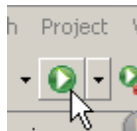


```

<?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="Hello Android"
    />
</LinearLayout>

```

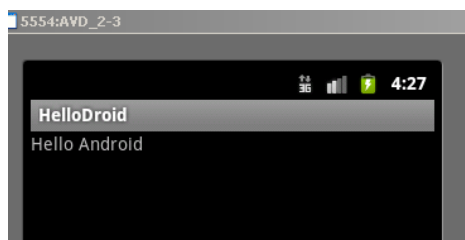
Jalankan project dengan mengklik icon run atau ctrl-F11, lalu pilih Android Application. Kalau emulator telah ditutup sebelumnya, maka perlu ditunggu agak lama.



Jika pada emulator hanya halaman depan yang muncul, drag slide lock ke sebelah kanan



Jangan diclose emulator ini, Eclipse selanjutnya akan menggunakan emulator yang sudah terbuka ini dan tidak perlu menjalankan yang baru.

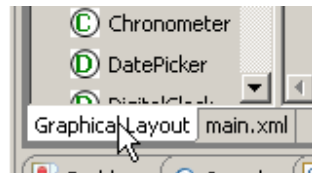


Dapat dilihat bahwa aspek user interface disimpan dalam XML di direktori res/layout yang terpisah dengan code, sehingga memudahkan programmer (mengapa?).

## Aplikasi Kedua: Hello xxx

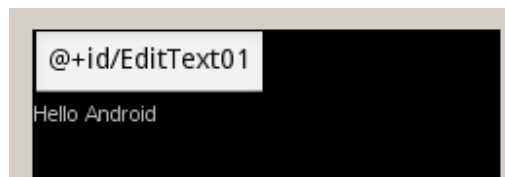
Selanjutnya kita akan memodifikasi program ini dan menambahkan masukan nama dari user, lalu setelah user menekan tombol, akan keluar “Hello, [nama]. Terimakasih”

Pertama, melalui project explorer kembali XML user interface main.xml. Pilih tab “Graphical Layout”

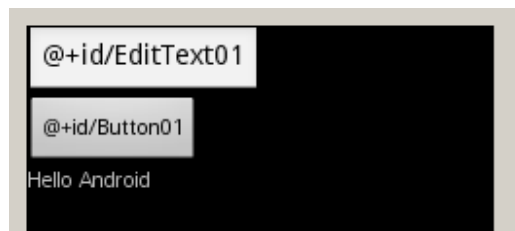


Komponen-komponen yang berada di bagian kiri, sering disebut “widget” atau “view”.

Pertama tambahkan EditText, drag EditText ini diatas TextView.



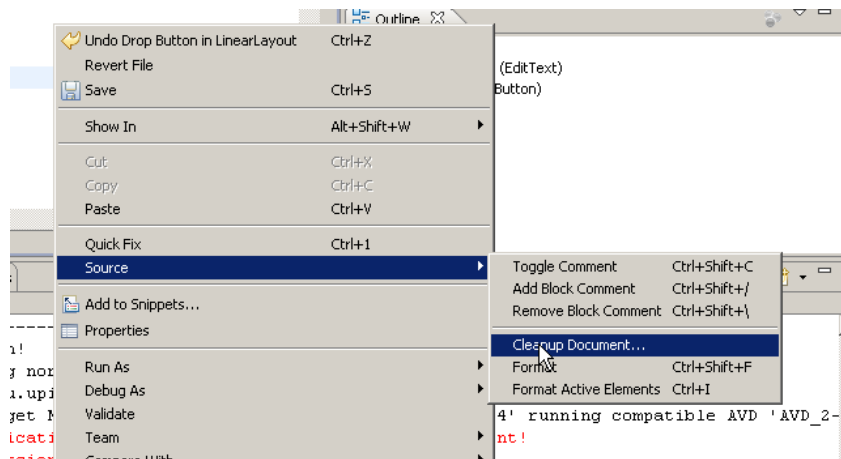
Lalu drag button diantara dua widget tersebut.



Coba anda run (ctrl-F11) untuk melihat hasil dari program ini. Jangan lupa, **jangan close** emulatoornya, supaya cepat.

Sekarang kita harus memberikan ID kepada setiap widget agar dapat dihubungkan dengan program.

Kembali ke main.xml, terlihat XML yang ditambahkan Eclipse tidak rapi, untuk merapkannya klik kanan dibagian editor XML, pilih Source → Cleanup Document.



Ganti android:text dan android:id untuk EditText dan Button. Atribut android:text adalah string pada komponen (caption atau label), sedangkan android:id adalah id yang dapat digunakan program untuk berhubungan dengan widget ini.

```

android:layout_height="fill_parent">
<EditText android:text="" android:id="@+id/etNama"
    android:layout_width="wrap_content" android:layout_hei
<Button android:text="Sapa!" android:id="@+id/bSapa"

```

Demikian juga id untuk TextView, anda perlu menambahkan "android:id=".

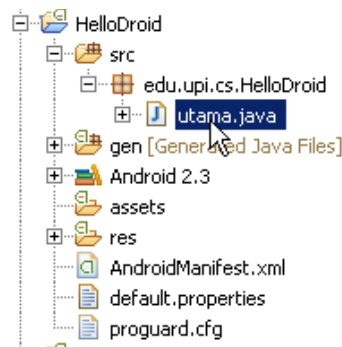
```

<TextView android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:text="Hello Android" android:id="@+id/tvSalam"/>

```

Sekarang kita perlu menambahkan code agar saat tombol diklik app akan mengeluarkan respon.

Melalui package explorer, kembali ke utama.java.



Pertama kita harus membuat event yang dipanggil saat tombol diklik. Import dua class

```
import android.view.View;
import android.view.View.OnClickListener;
```

**TBD: Ganti dengan android:onClick saja, lebih mudah.**

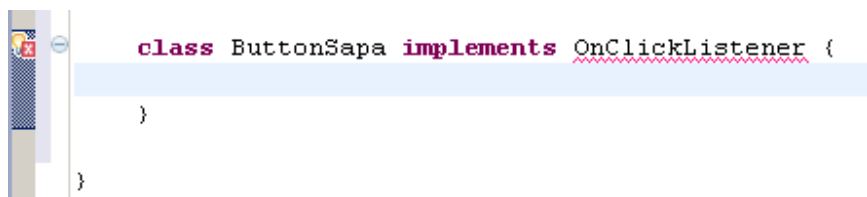
Lalu **di dalam kelas utama**, buatlah class ButtonSapa

```
public class utama extends Activity {
    class ButtonSapa implements OnClickListener {
        @Override
        public void onClick(View v) {
        }
    }
}
```

#### Tips:

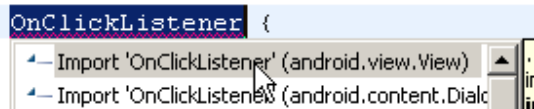
Cara yang lebih mudah untuk menambahkan impor: anda langsung menyetikkan code, dan jika terjadi error, pilih opsi untuk menambahkan impor otomatis.

Misalnya, jika anda menyetikkan onClickListener dan class tersebut belum diimpor maka akan muncul error seperti ini. Lalu klik gambar x kecil-merah dibagian kiri



```
class ButtonSapa implements OnClickListener {
}
}
```

lalu pilih “impor ...”, maka secara otomatis impor akan ditambahkan oleh Eclipse.



```
OnClickListener {
  Import 'OnClickListener' (android.view.View)
  Import 'OnClickListener' (android.content.DialogInterface)
```

Lengkapi isi method onClick

```
class ButtonSapa implements OnClickListener {
    @Override
    public void onClick(View v) {
        //hubungkan komponen
        EditText etNama = (EditText) findViewById(R.id.etNama);
        TextView tvSalam = (TextView) findViewById(R.id.tvSalam);

        //ambil string nama
        String strNama= etNama.getText().toString();

        //set salam
        tvSalam.setText("Halo "+strNama+" selamat menggunakan android");
    }
}
```

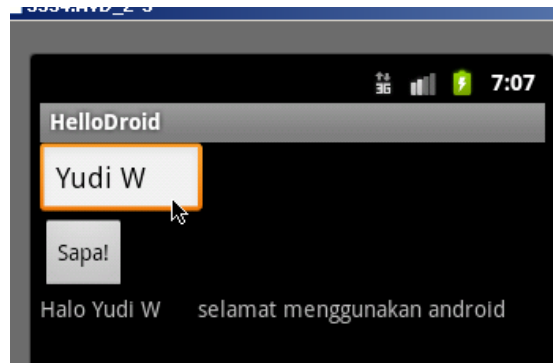
Sedangkan pada bagian import ditambahkan (biasanya sudah secara otomatis)

```
import android.widget.EditText;
import android.widget.TextView;
import edu.upi.cs.HelloDroid.R;
```

Selanjutnya kita perlu mengeset Onclik-nya button, tambahkan dua baris code ini dibagian onCreate.

```
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    Button btSapa= (Button) findViewById(R.id.btSapa);
    btSapa.setOnClickListener(new ButtonSapa());
}
```

Jalankan program (ctrl-F11)



### Catatan:

Pada contoh diatas, event handler dibuat dengan membuat class baru. Ini membuat app sedikit lebih boros dari sisi memori. Alternatif lain adalah dengan mengimplement handler di kelas utama.

Pertama, tambahkan deklarasi di kelas utama bahwa class ini akan mengimplements method OnClickListener

```
public class Utama extends Activity implements OnClickListener {
```

Lalu, implementasikan method onClick di class Utama.

```
@Override
public void onClick(View arg0) {
    // TODO Auto-generated method stub
    EditText edNama = (EditText) findViewById (R.id.etNama);
    TextView tvSalam = (TextView) findViewById (R.id.tvSalam);
    String strName = edNama.getText().toString();
    tvSalam.setText("Halo "+strName);
}
}
```

Lalu set di btSapa class Utama ini (this).

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

    Button btSapa = (Button) findViewById (R.id.btSapa);
    btSapa.setOnClickListener(this);
}
}
```

Alternatif ketiga adalah dengan menggunakan anonymous inner classes. Cara ini lebih ringkas, tapi tetap menggunakan memori lebih banyak karena tetap membuat class baru dan membuat code lebih sulit dibaca.

Dengan cara ini, anda tidak perlu membuat method terpisah, tetapi langsung diisi saat `setOnClickListener`.

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

        Button btSapa = (Button) findViewById (R.id.bSapa);
        btSapa.setOnClickListener(new OnClickListener() {
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                EditText edNama = (EditText) findViewById (R.id.etNama);
                TextView tvSalam = (TextView) findViewById (R.id.tvSalam);
                String strName = edNama.getText().toString();
                tvSalam.setText("Halo "+strName);
            }
        });
    }
}
```

Alternatif keempat adalah dengan memanfaatkan property android.onClick. Pada main.xml set atribut android.onClik dengan nama method yang akan menangani event tersebut.

```
<Button android:text="Sapa!" android:id="@+id/bSapa"
        android:onClick="bSapaClick"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
/>
```

Kemudian buat satu method public dengan satu parameter View.

```
public class Utama extends Activity {
    /** Called when the activity is first created. */

    public void bSapaClick(View view) {
        EditText edNama = (EditText) findViewById (R.id.etNama);
        TextView tvSalam = (TextView) findViewById (R.id.tvSalam);
        String strName = edNama.getText().toString();
        tvSalam.setText("Halo .. "+strName);
    }
}
```

Cara ini paling sederhana, tetapi programmer harus memastikan nama method pada XML cocok dengan method di program. Jika nama method tidak sama, maka tidak akan muncul kesalahan pada saat program dcompile, tapi akan menyebabkan kesalahan pada saat runtime.

Latihan:

Buatlah program untuk menghitung luas sebuah persegi panjang. Input adalah panjang dan lebar. Output adalah luasnya (panjang kali lebar). Gunakan syntax berikut untuk mengubah tipe string ke double.

```
double dblPanjang = Double.valueOf(strPanjang);
```



## Widget

Pada modul sebelumnya kita telah menggunakan input box, button dan label. Komponen user interface ini disebut widget. Beberapa widget dasar adalah: TextView, Button, Image, EditText, CheckBox, RadioButton.

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

Detil property dan event untuk setiap widget dapat dilihat di:

<http://developer.android.com/reference/android/widget/package-summary.html> (scroll sedikit ke bawah, di bagian classes)

### Text View

Untuk menampilkan label. Beberapa contoh property dari TextView adalah android:textSize, android:textStyle, android:textColor.

```
<TextView
    android:text="Nama:" android:id="@+id/TextView01"
    android:textSize="30sp"
    android:textColor="#FF0000"
    android:textStyle="bold"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

Hasilnya:



Catatan: pada atribut textSize, direkomendasikan menggunakan ukuran sp. sp singkatan dari scaled-pixel yang memperhitungkan kepadatan resolusi dan juga preferensi ukuran font dari pengguna.

## Button

Button merupakan turunan dari TextView sehingga yang berlaku di textView juga berlaku di button. Tambahan property yang penting adalah onClick (lihat pembahasannya di halaman 13)

```
<Button android:text="KlikSaya"
        android:id="@+id/bSaya"
        android:onClick="bSayaClick"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
/>
```

Hasilnya (perhatikan penggunaan fill\_parent untuk atribut layout\_width):

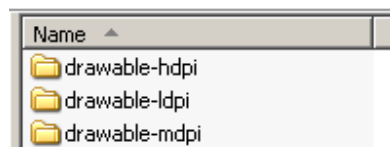


Catatan: dp yang digunakan sebagai satuan panjang merupakan singkatan dari *density-independent pixel*. Nilai real pixel untuk 1 dp berubah-ubah sesuai dengan resolusi. 1 dp pada resolusi 160 pixel berarti 1 pixel. Jika resolusinya 320 pixel, maka 1 dp = 2 pixel demikian seterusnya.

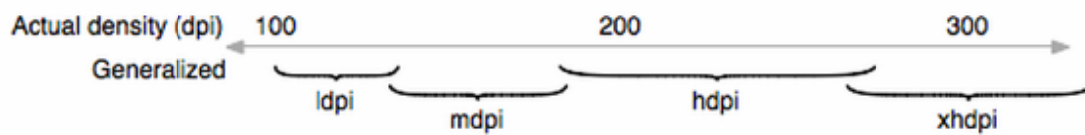
## ImageButton dan ImageView

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

Image diletakkan di direktori `/[projectand]/res/drawable-hdpi`, `/[projectand]/res/drawable-ldpi` dan `/[projectand]/res/drawable-mdpi`



ldpi untuk low density (resolusi rendah), mdpi untuk medium dan hdpi untuk high density.

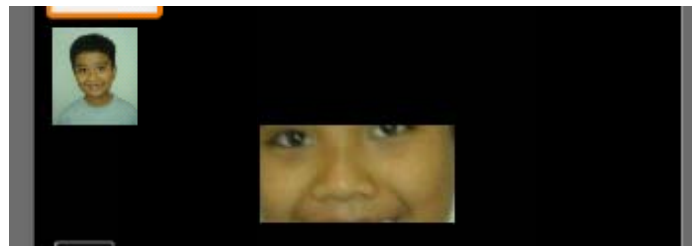


Contoh, jika sebuah file gambar furqon2.jpg diletakkan di direktori drawable (hdpi sampai dengan ldpi). Maka contoh deklarasi widget imageView dan imageButton adalah

```
<ImageView
    android:id="@+id/ImageView01"
    android:src="@drawable/furqon2"
    android:layout_width="50dp"
    android:layout_height="50dp"
/>

<ImageButton
    android:adjustViewBounds="false"
    android:src="@drawable/furqon"
    android:id="@+id/ImageButton01"
    android:onClick="bSapaClick"
    android:layout_height="50dp"
    android:layout_width="100dp"
    android:layout_gravity="center_horizontal"
/>
```

dan hasilnya

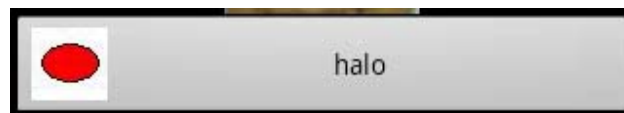


Untuk menambahkan image bersama-sama dengan button, dapat digunakan komponen button (**bukan** ImageButton) lalu set atribut drawableLeft dengan image yang diinginkan.

Contoh deklarasi UML-nya. Gunakan ukuran image yang sesuai.

```
<Button
    android:adjustViewBounds="false"
    android:drawableLeft="@drawable/circle"
    android:id="@+id/ImageButton03"
    android:onClick="bSapaClick"
    android:layout_height="wrap_content"
    android:layout_width="fill_parent"
    android:layout_gravity="center_horizontal"
    android:text="halo"
/>
```

dan hasilnya



Coba ganti `drawableLeft` dengan `drawableRight`, `drawableTop` atau `drawableBottom`.

### ***EditText***

`EditText` digunakan untuk menerima input dari pengguna. Beberapa atribut yang berguna adalah `android:password`, `android:maxLine`, `android:maxLength`, `android:minWidth` dan `android:phoneNumber`. Contoh:

```
<EditText
    android:text=""
    android:id="@+id/EditText01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:maxLines="1"
    android:maxLength="100dp"
    android:minWidth="100dp"
    android:password="true"
/>
```

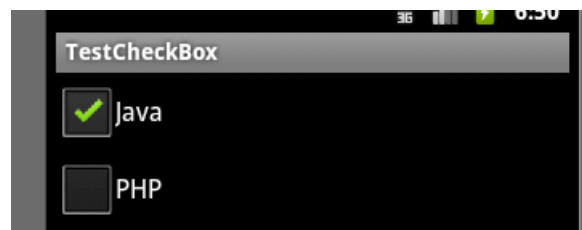
## CheckBox

User dapat memilih lebih dari satu pilihan dengan checkbox.

Contoh deklarasi checkbox

```
<CheckBox
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/cbJava"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Java"
/>
<CheckBox
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/cbPHP"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="PHP"
```

Hasilnya

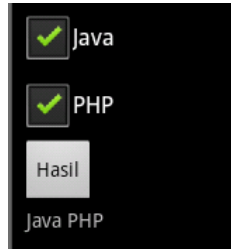


Sedangkan program untuk mendapatkan nilai suatu checkbox di check atau tidak adalah sebagai berikut.

Tambahkan Button dan TextView di form untuk menampilkan hasil pilihan user.

```
<Button android:text="Hasil"
    android:onClick = "klikHasil"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:id="@+id/bHasil">
</Button>
<TextView
    android:text="Hasilnya:"
    android:id="@+id/tvHasil"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</TextView>
```

sehingga hasilnya seperti ini



Tambahkan di activity code sebagai berikut, perhatikan penggunaan isChecked untuk mengambil nilai apakah user meng-check pilihan:

```
public class Main extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
  
    public void klikHasil(View V) {  
        TextView tvHasil = (TextView) findViewById(R.id.tvHasil);  
        CheckBox cbJava = (CheckBox) findViewById(R.id.cbJava);  
        CheckBox cbPHP = (CheckBox) findViewById(R.id.cbPHP);  
        String s="";  
        if (cbJava.isChecked()) {  
            s = s + " Java";  
        }  
        if (cbPHP.isChecked()) {  
            s = s + " PHP";  
        }  
        tvHasil.setText(s);  
    }  
}
```

### **RadioButton**

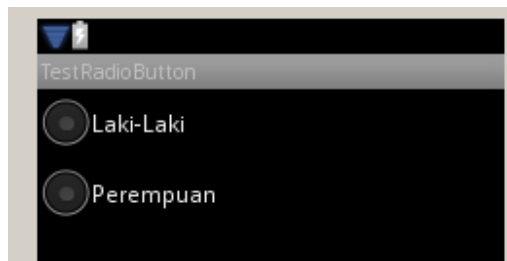
Pada radioButton, hanya satu yang boleh aktif (mutual exclusive) di dalam satu group yang disebut radioButton. Contoh deklarasinya:

```

<RadioGroup
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/rgJenisKelamin"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
>
<RadioButton android:id="@+id/rLaki2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Laki-Laki" />
<RadioButton android:id="@+id/rPerempuan"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Perempuan" />
</RadioGroup>

```

Hasilnya:



Catatan: Anda dapat mengeset atribut android:orientation pada RadioGroup menjadi horizontal agar radio button tersusun secara horizontal.

Contoh program untuk mengambil hasilnya adalah sebagai berikut:

Tambahkan button dan textView seperti pada halaman 18 untuk kasus checkBox, sehingga form menjadi (letakkan button dan TextView di dalam RadioGroup):

```

<Button
    android:text="Hasil"
    android:id="@+id/bHasil"
    android:onClick="klikHasil"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:text="TextView"
    android:id="@+id/tvHasil"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</RadioGroup>

```



Tambahkan method `klikHasil` sebagai berikut. Perhatikan penggunaan `getCheckedRadioButtonId` yang mengambil `idRadio` yang dipilih oleh pengguna:

```

public void klikHasil(View v) {
    RadioGroup rgJenisKelamin = (RadioGroup) findViewById(R.id.rgJenisKelamin);
    TextView tvHasil = (TextView) findViewById(R.id.tvHasil);
    int id = rgJenisKelamin.getCheckedRadioButtonId();
    if (id==R.id.rLaki2) {
        tvHasil.setText("Laki-laki");
    } else if (id==R.id.rPerempuan) {
        tvHasil.setText("Perempuan");
    } else {
        tvHasil.setText("Tidak diisi");
    }
}

```



## ***ListView***

## **Dialog**

TBD

## Layout Manager

Untuk meletakkan komponen (view) pada layar, android menggunakan layout manager. Ada empat jenis layout manager. LinearLayout, TableLayout, RelativeLayout dan FrameLayout. Sebelumnya kita telah menggunakan LinearLayout yang meletakkan view secara berurutan. Coba lihat main.xml kembali di direktori resource

```
<?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">
```

## RelativeLayout

RelativeLayout mengatur posisi secara relatif berdasarkan posisi widget yang lain atau parentnya. Contoh dua widget dibawah. EditText posisinya relatif terhadap textView nama.



Deklarasinya adalah sebagai berikut:

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

    <TextView android:id="@+id/label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nama: "
        android:paddingTop="15px"/>

    <EditText
        android:id="@+id/entry"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@id/label"
        android:layout_alignBaseline="@id/label"/>

</RelativeLayout>
```

Perhatikan penggunaan atribut `layout_toRightOf` yang menyatakan bahwa `EditText` berada disebelah label. Sedangkan atribut `layout_baseLine` digunakan agar `TextView` dan `EditText` aligned. Jika `layout_baseLine` dihapus efeknya akan sebagai berikut (`TextView` terlalu tinggi posisinya):

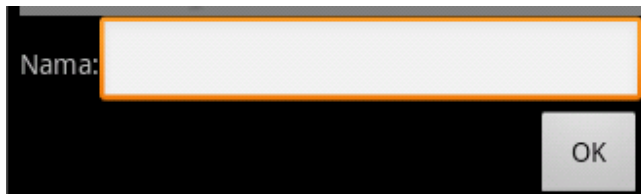


Beberapa atribut lain yang mengatur posisi widget dibandingkan dengan widget lain adalah:

- `android:layout_above`
- `android:layout_below`
- `android:layout_toLeftOf`
- `android:layout_toRightOf`.

Untuk mengatur posisi ada tambahan lima atribut: `android:layout_alignTop`, `android:layout_alignBottom`, `android:layout_alignLeft`, `android:layout_alignBaseline`

Contoh jika kita ingin menambahkan button sehingga posisinya sebagai berikut:



Maka deklarasi buttonnya adalah sebagai berikut:

```
<Button
    android:id="@+id/ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/entry"
    android:layout_alignRight="@id/entry"
    android:text="OK" />
```

Artinya button tersebut dibawah dan disebelah kanan `EditText`.

Dengan `RelativeLayout`, widget juga dapat diatur berdasarkan parent.

- `android:layout_alignParentTop`
- `android:layout_alignParentBottom`

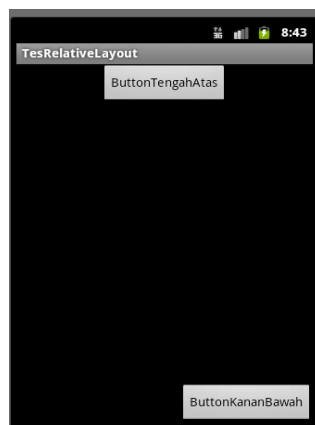
- android:layout\_alignParentLeft
- android:layout\_alignParentRight
- android:layout\_centerHorizontal
- android:layout\_centerVertical
- android:layout\_centerInParent

Contoh:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:layout_width="match_parent"
    android:id="@+id/relativeLayout1"
    android:layout_height="wrap_content"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ButtonTengahAtas"
        android:layout_centerHorizontal="true"
        android:layout_alignParentTop="true">
    </Button>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ButtonKananBawah"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true">
    </Button>
</RelativeLayout>
```

Hasilnya:



Silahkan bereksperimen dengan berbagai setting atribut layout.

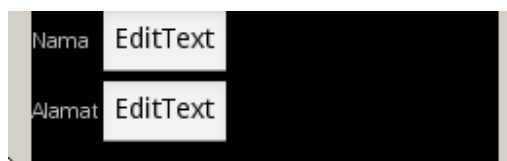
## TableLayout

TableLayout meletakkan komponen dalam bentuk tabel (baris dan kolom), mirip seperti Table yang digunakan di HTML untuk mengatur layout. Pengguna mendefinisikan baris dengan tag <TableRow> dan secara otomatis sistem mendefinisikan kolomnya. Contoh deklarasi berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout android:id="@+id/tableLayout1"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
>
    <TableRow>
        <TextView android:text="Nama" android:id="@+id/textView1"
            android:layout_width="wrap_content" android:layout_height="wrap_content" />
        <EditText android:text="EditText" android:id="@+id/editText1"
            android:layout_width="wrap_content" android:layout_height="wrap_content" />
    </TableRow>
    <TableRow>
        <TextView android:text="Alamat" android:id="@+id/textView1"
            android:layout_width="wrap_content" android:layout_height="wrap_content" />
        <EditText android:text="EditText" android:id="@+id/editText1"
            android:layout_width="wrap_content" android:layout_height="wrap_content" />
    </TableRow>
</TableLayout>
```

Hasilnya adalah sebagai berikut (bayangkan sebagai sebuah tabel dengan dua kolom dan dua baris).

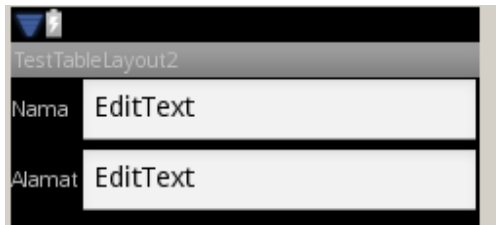
	Kolom 1
Kolom 0	



Misalnya kita ingin agar EditText sampai ke bagian belakang, maka tambahkan android:stretchColumns="1" pada TableLayout. Maka kolom 1 akan terisi penuh.

```
<TableLayout android:id="@+id/tableLayout1"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:stretchColumns="1">
```

Hasilnya:



Selanjutnya kita ingin menambahkan button.. Button ini mengisi dua kolom, mulai dari kolom 0 sampai dengan 1, untuk mengekspresikan hal tersebut, gunakan atribut `android:layout_span`

```
<Button
    android:text="Button"
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_span="2"
/>
```

Lalu tambahkan `TextView` di **kolom 1** (ingat kolom dimulai dari 0). Ini bisa dilakukan dengan menggunakan atribut `android:layout_column`. Tanpa atribut ini, `TextView` akan muncul di kolom 0

```
<TextView
    android:layout_column="1"
    android:text="(Petunjuk: xxxx xx xxx x xxx)"
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >
</TextView>
```



## Frame Layout

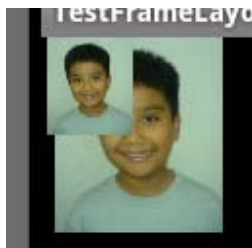
FrameLayout digunakan untuk menampilkan satu komponen. Jika ada lebih dari satu widget, maka akan ditampilkan secara bertumpuk. Kita bisa menambahkan layout lain sebagai child di dalam frame layout.

Contoh untuk deklarasi berikut menampilkan dua ImageView (pastikan file image telah disimpan di direktori [project]/res/drawable-hdpi, [project]/res/drawable-ldpi dan [project]/res/drawable-mdpi )

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout android:id="@+id/frameLayout1"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <ImageView android:id="@+id/ImageView01" android:src="@drawable/furqon"
        android:layout_width="100dp" android:layout_height="100dp" />
    <ImageView android:id="@+id/ImageView01" android:src="@drawable/furqon"
        android:layout_width="50dp" android:layout_height="50dp" />
</FrameLayout>
```

Hasilnya gambar akan bertumpuk seperti ini.





## Kombinasi Layout

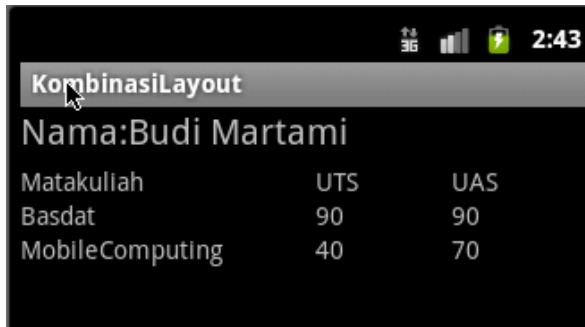
Kita bisa mengkombinasikan beberapa layout dalam satu form. Contoh berikut menggunakan linearlayout yang memiliki dua anak yaitu relativelayout dan tablelayout.

```
<?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">

    <RelativeLayout android:layout_width="match_parent"
        android:id="@+id/relativeLayout1" android:layout_height="wrap_content">
        <TextView android:id="@+id/nama" android:layout_width="wrap_content"
            android:layout_height="wrap_content" android:textSize="20sp"
            android:text="Nama: " />
        <TextView android:id="@+id/isi" android:layout_width="wrap_content"
            android:layout_height="wrap_content" android:text="Budi Martani"
            android:textSize="20sp" android:paddingBottom="5dp"
            android:layout_toRightOf="@id/nama" />
    </RelativeLayout>
    <TableLayout android:id="@+id/tableLayout1"
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:stretchColumns="*">
        <TableRow>
            <TextView android:text="Matakuliah" android:id="@+id/textView1"
                android:layout_width="wrap_content" android:layout_height="wrap_content" />
            <TextView android:text="UTS" android:id="@+id/textView1"
                android:layout_width="wrap_content" android:layout_height="wrap_content" />
            <TextView android:text="UAS" android:id="@+id/textView1"
                android:layout_width="wrap_content" android:layout_height="wrap_content" />
        </TableRow>
        <TableRow>
            <TextView android:text="Basdat" android:id="@+id/textView1"
                android:layout_width="wrap_content" android:layout_height="wrap_content" />
            <TextView android:text="90" android:id="@+id/textView1"
                android:layout_width="wrap_content" android:layout_height="wrap_content" />
            <TextView android:text="90" android:id="@+id/textView1"
                android:layout_width="wrap_content" android:layout_height="wrap_content" />
        </TableRow>
        <TableRow>
            <TextView android:text="MobileComputing" android:id="@+id/textView1"
                android:layout_width="wrap_content" android:layout_height="wrap_content" />
            <TextView android:text="40" android:id="@+id/textView1"
                android:layout_width="wrap_content" android:layout_height="wrap_content" />
            <TextView android:text="70" android:id="@+id/textView1"
                android:layout_width="wrap_content" android:layout_height="wrap_content" />
        </TableRow>
    </TableLayout>
</LinearLayout>
```

Relative  
ut  
gan dua  
View

TableLayout  
dengan tiga  
kolom dan tiga  
baris

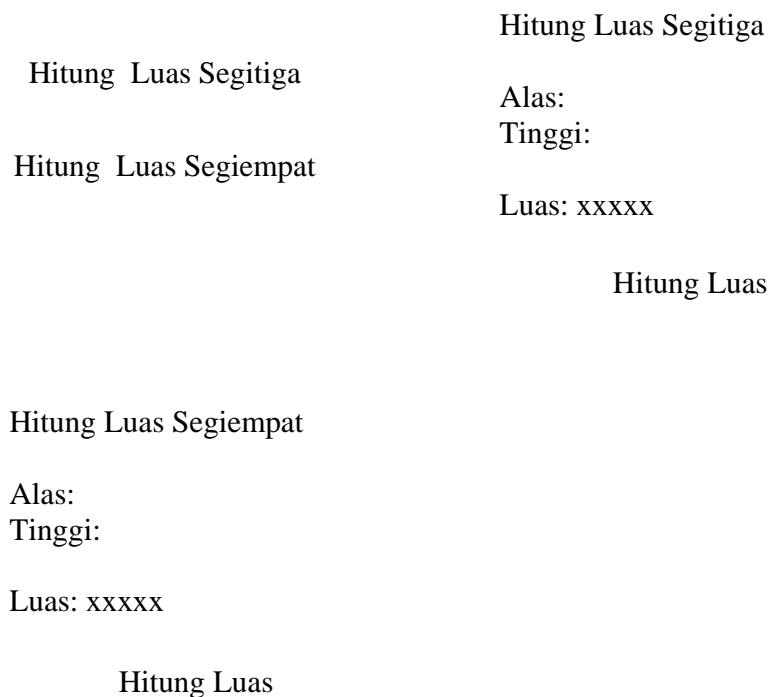


Latihan Layout

## Sub Activity

Dalam materi sebelumnya, selalu digunakan satu activity (satu form). Tentu saja sebuah aplikasi sering membutuhkan multi activity. Sebagai contoh, misalnya kita akan membuat sebuah aplikasi matematika sederhana. Aplikasi ini menyediakan menu yang berisi pilihan untuk menghitung luas segiempat dan luas segitiga.

Rancangan layarnya sebagai berikut.



Pertama kita akan membuat form utama terlebih dulu yang memiliki dua menu. Berikut deklarasi XML form main.

```
<?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">
    <Button android:text="Hitung Luas Segitiga" android:id="@+id/buttonSegiTiga"
        android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:onClick="segitigaClick" />
    <Button android:text="Hitung Luas Segiempat" android:id="@+id/buttonSegiEmpat"
        android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:onClick="segiempatClick" />
</LinearLayout>
```

Dengan hasil sebagai berikut:



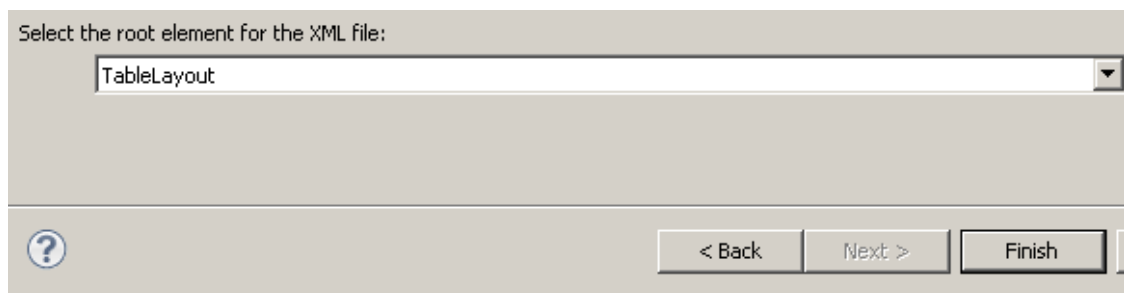
Selanjutnya kita akan membuat form kedua untuk menghitung luas segitiga. Klik layout di package Explorer lalu klik kanan → New atau File → New → Other (Ctrl+N). Pilih Android XML File.



Beri nama file segitiga



dengan root TableLayout



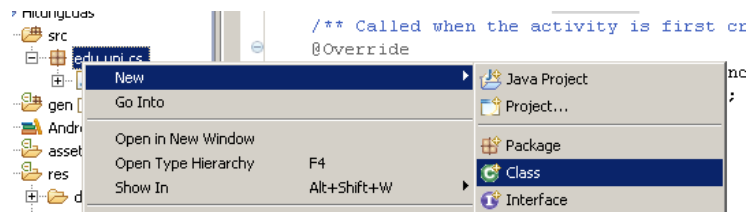
Buat form untuk menerima input dari user dan menampilkan keluaran



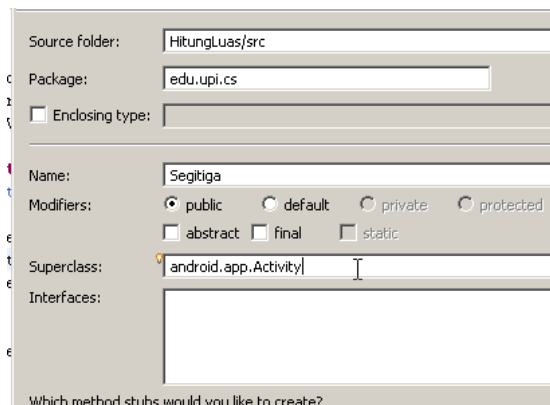
Deklarasi XML-nya adalah sebagai berikut (perhatikan penggunaan atribut `android:numeric=integer|decimal` agar teks hanya menerima nilai integer atau desimal):

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent"
    android:stretchColumns="1">
    <TableRow>
        <TextView android:text="Alas:" android:id="@+id/textView1"
            android:layout_width="wrap_content" android:layout_height="wrap_content" />
        <EditText android:text="" android:id="@+id/etAlas"
            android:layout_width="wrap_content" android:layout_height="wrap_content" />
    </TableRow>
    <TableRow>
        <TextView android:text="Tinggi:" android:id="@+id/textView1"
            android:layout_width="wrap_content" android:layout_height="wrap_content" />
        <EditText android:text="" android:id="@+id/etTinggi"
            android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:numeric="integer|decimal"/>
    </TableRow>
    <TableRow>
        <TextView android:text="Luas:" android:id="@+id/textView1"
            android:layout_width="wrap_content" android:layout_height="wrap_content" />
        <TextView android:text="Luas:" android:id="@+id/tvLuas"
            android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:numeric="integer|decimal"/>
    </TableRow>
    <Button android:text="Hitung Luas Segitiga" android:id="@+id/bHitungLuas"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:onClick="hitungLuasClick"
        />
</TableLayout>
```

Selanjutnya, kita akan membuat program untuk menghitung luas segitiga. Klik package, lalu klik kanan → New → Class



Beri nama Segitiga, dan merupakan turunan dari android.app.Activity



Set layout yang digunakan (R.layout.segitiga) dan implementasi saat button diklik.

```

package edu.upi.cs;

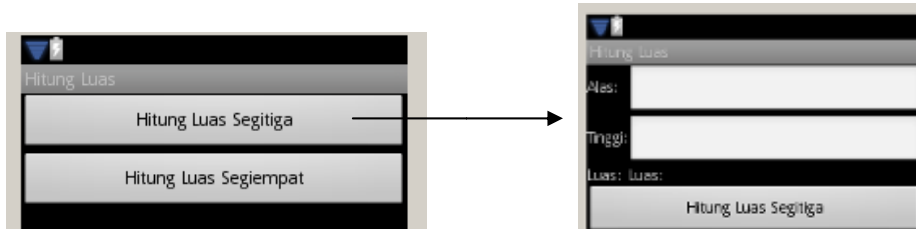
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class Segitiga extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.segitiga);
    }

    public void hitungLuasClick(View v) {
        EditText etAlas = (EditText) findViewById(R.id.etAlas);
        EditText etTinggi = (EditText) findViewById(R.id.etTinggi);
        TextView tvLuas = (TextView) findViewById(R.id.tvLuas);
        double alas = Double.valueOf(etAlas.getText().toString());
        double tinggi = Double.valueOf(etTinggi.getText().toString());
        double luas = 0.5 * alas * tinggi;
        tvLuas.setText(Double.toString(luas));
    }
}

```

Selanjutnya adalah menghubungkan menu utama dengan modul menghitung segitiga ini.



Intent dapat digunakan untuk memanggil activity lain. Intent adalah request dari satu activity ke activity atau intent listener lain. Agar dapat dipanggil, kita harus menambahkan intent ke AndroidManifest.xml.



Intent dapat digunakan untuk memanggil activity lain. Intent adalah request dari satu activity ke activity atau intent listener lain. Intent juga digunakan untuk notifikasi event misalnya SD card dimasukkan, SMS masuk, activity dijalankan dst

**Penting:** agar activity hitung luas segitiga dapat dipanggil, kita harus menambahkan activity ini ke AndroidManifest.xml.

```
<application android:icon="@drawable/icon" android:label="@string/app_name">
  <activity android:name=".Main"
            android:label="@string/app_name">
    <intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
  </activity>
  <activity android:name=".Segitiga">
</activity>
```

Sekarang kembali ke Main.Java, tambahkan pada method segitigaClick intent berisi permintaan untuk menjalankan activity segitiga dan jalankan dengan startActivity.

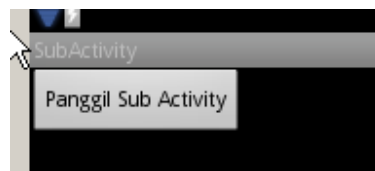
```
public class Main extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
  
    public void segitigaClick(View v) {  
        Intent iSegitiga = new Intent (this,Segitiga.class);  
        startActivity(iSegitiga);  
    }  
}
```

Latihan:

Lengkap program diatas dengan hitung segiempat. Buat form agar pengguna dapat memasukan panjang dan lebar, tambahkan tombol yang jika diklik menampilkan luas segiempat. Hubungkan menu utama dengan form tersebut dengan intent.

Pada contoh diatas digunakan startActivity untuk memanggil sub activity. Tidak ada notifikasi apakah subactivity sudah selesai atau belum dan terkadang kita juga membutuhkan data dari subactivity. Untuk hal tersebut, dapat digunakan startActivityForResult.

Perlu penjelasan yang agak panjang untuk mencoba startActivityForResult. Buatlah sebuah project baru. Pada form Main, tambah satu button yang jika diklik akan memanggil sub activity dan mengirimkan data.





Kemudian kita buat form sub activity-nya. Klik kanan di layout → New → Other → Android XML File, beri nama sub.xml

Form-nya seperti berikut, TextView akan menampilkan kiriman data dari form utama,



Potongan XML-nya adalah sebagai berikut:

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

Sekarang kita membuat Sub.java. Klik kanan dibagian package (misal edu.upi.cs) → New → Class. Beri nama Sub, dan isi superclass dengan android.app.Activity.

Jangan lupa ganti parameter setContentView dengan .sub.

onStart adalah method yang dipanggil setelah onCreate dan activity telah ditampilkan. Isi method ini dengan menampilkan isi data yang dikirimkan program utama.

```
public class Sub extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sub);
    }

    public void onStart() {
        super.onStart();
        Intent i= getIntent();//ambil intent yg dikirimkan
        String msg = i.getStringExtra("data1"); //ambil nilai yg dikirim
        TextView tv = (TextView) findViewById(R.id.tvKirimanActivityUtama);
        tv.setText(msg);
    }
}
```

Sekarang kita **kembali ke main.java**, gunakan method `putExtra` untuk menambahkan data ke intent. Pada `startActivityForResult` terdapat parameter angka 99 (silahkan ganti dengan angka atau konstanta apapun). Ini karena satu activity bisa mengirimkan banyak intent dan diperlukan cara untuk membedakan antara satu intent dengan intent yang lain.

Method `onActivityResult` akan dipanggil saat activity yang dipanggil telah selesai.

```
import android.app.Activity;

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

    public void PanggilSubActivity (View v) {
        Intent i = new Intent(this,Sub.class);
        i.putExtra("data1", "1234567989,BUDI MARTAMI"); //data yg dikirim
        startActivityForResult(i,99);
    }

    protected void onActivityResult(int requestCode, int resultCode,Intent data) {
        if ((requestCode==99) && (resultCode == RESULT_OK)) {
            //subactivity sukses dipanggil
        }
    }
}
```

### Jangan lupa mendaftarkan sub activity di `AndroidManifest.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.upi.cs"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />

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

        <activity android:name=".Sub" > </activity>

    </application>
</manifest>
```

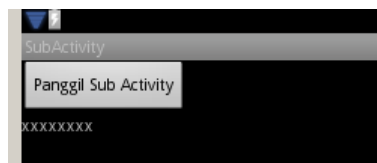
Test program, maka subactivity akan menampilkan data yang dikirim dari form utama.



Jika terjadi force close, ada kemungkinan activity belum ditambahkan di AndroidManifest.xml

Pada program diatas, anda telah berhasil mengirimkan data ke sub activity. Sekarang kita akan membuat sebaliknya, data dari sub activity akan dikirim ke form utama. Kita akan memodifikasi program diatas sbb:

Pada main.xml, tambahkan TextView untuk menampilkan hasil dari sub activity, beri id tvHasilSub.



Lalu pada main.java, lengkapi method onActivityResult. Pada method ini, diterima dan diproses hasil dari sub activity.

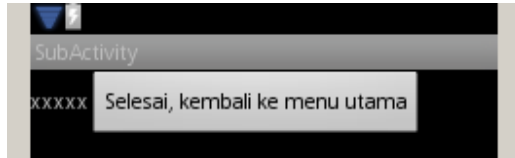
```
import android.app.Activity;

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

    public void PanggilSubActivity (View v) {
        Intent i = new Intent(this,Sub.class);
        i.putExtra("data1", "1234567989,BUDI MARTAMI"); //data yg dikirim
        startActivityForResult(i,99);
    }

    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        if ((requestCode==99) && (resultCode == RESULT_OK)) {
            TextView tvHasilSubForm = (TextView) findViewById(R.id.tvHasilSub);
            String msg = data.getStringExtra("dataSub");
            tvHasilSubForm.setText(msg);
        }
    }
}
```

Kemudian untuk subactivitynya sendiri, tambahkan Button yang diklik saat pengguna telah selesai dengan subactivity. Edit sub.xml sebagai berikut

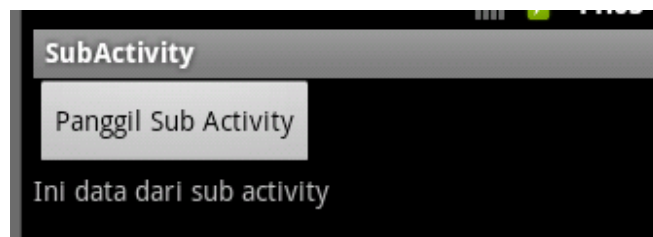


```
<Button android:text="Selesai, kembali ke menu utama"
        android:onClick="selesaiClick"
        android:id="@+id/bSelesai" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
```

Implementasikan method saat Button selesai diklik di sub.java

```
public void selesaiClick(View v) {
    Intent i= getIntent();
    i.putExtra("dataSub", "Ini data dari sub activity");
    setResult(RESULT_OK, i);
    finish();
}
```

Jalankan program, setelah subactivity ditutup, maka di program utama akan muncul kiriman data dari sub activity



## Menyimpan Data

Android menyediakan beberapa cara untuk menyimpan data: file, relational database dengan SQLite dan pasangan key/value yang disebut sistem preferences. Untuk sharing antar aplikasi disediakan mekanisme yang disebut ContentProvider.

### Shared Preferences

Shared Preferences (SP) adalah mekanisme untuk menyimpan pasangan key-value untuk tipe data primitif (integer, double, string, boolean). SP cocok untuk penggunaan ringan seperti menyimpan setting aplikasi dan kondisi user interface (misalnya saat activity masuk ke state sleep karena device menerima telepon).

Data dalam shared preferences disimpan dalam device android dalam bentuk XML.

Shared Preference objek untuk activity diperoleh melalui `getPreferences()`.

Ada tiga mode untuk shared preferences (SP)

MODE\_PRIVATE: hanya aplikasi yang membuat SP yang dapat mengakses data

MODE\_WORLD\_READABLE: aplikasi yang lain boleh membaca

MODE\_WORLD\_WRITEABLE: aplikasi yang lain boleh membaca sekaligus menulis.

Code berikut memperlihatkan cara penggunaan SP. Jangan lupa memanggil commit!

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

        SharedPreferences sp = getSharedPreferences("dataku", MODE_PRIVATE);

        //isi data
        Editor ed = sp.edit();
        ed.putString("nama", "Budi Martami");
        ed.putInt("umur", 20);
        ed.commit(); //jangan lupa!!

        //ambil data, bisa diclass yg berbeda
        String nama = sp.getString("nama", "");
        int umur = sp.getInt("umur", 0);

        //buat dialog untuk menampilkan data
        AlertDialog dialog = new AlertDialog.Builder(this).create();
        dialog.setMessage("Yg punya progam ini "+nama+ " berumur "+umur);
        dialog.show();
    }
}
```

Jalankan. Kemudian coba komentari bagian isi data, dan jalankan kembali aplikasi.

```
//isi data
/* Editor ed = sp.edit();
ed.putString("nama", "Budi Martami");
ed.putInt("umur",20);
ed.commit(); //jangan lupa!!
*/

//ambil data, bisa diclass yg berbeda
String nama = sp.getString("nama", "");
int umur = sp.getInt("umur",0);
```

Dapat dilihat data nama dan umur tersimpan walaupun aplikasi sudah ditutup,

Latihan:

Buat aplikasi yang menyimpan berapa kali aplikasi telah dijalankan. Tampilkan jumlah tersebut.

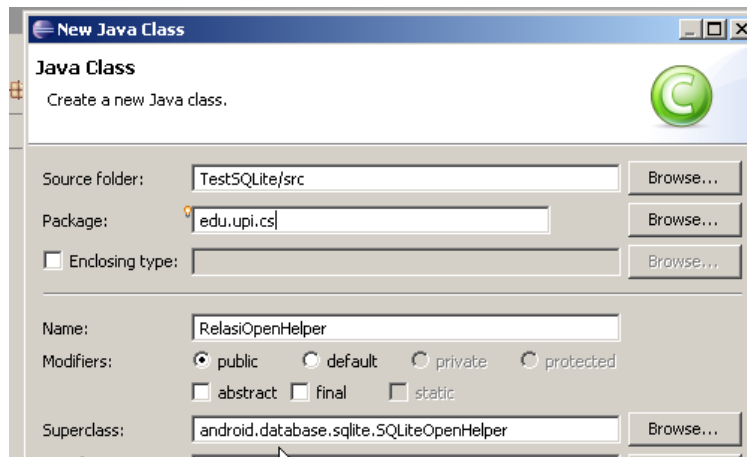
## **SQLite**

SQLite menyediakan fasilitas relational DBMS dengan perintah SQL. Referensi lengkap SQL yang digunakan dapat dilihat di <http://www.sqlite.org/lang.html>

Setiap aplikasi memiliki instance database tersendiri, dan berbeda dengan shared preference, tidak disediakan fasilitas agar aplikasi dapat membaca database aplikasi lain.

Untuk menggunakan SQLite, sangat dianjurkan menggunakan class helper untuk membuka dan menutup database, menginisiasi tabel dst. Helper juga digunakan saat melakukan upgrade struktur database jika suatu saat aplikasi diupdate. Class helper ini diturunkan dari class SQLiteOpenHelper.

Untuk latihan kita akan membuat aplikasi sederhana yang menyimpan data relasi berupa nama dan nomor telpon. Pertama kita buat class helper terlebih dulu. Buatlah project Android, tambahkan ke dalam project tersebut satu class (file → New → class). Beri class ini RelasiOpenHelper.



Buatlah code berikut

```

package edu.upi.cs;

import android.content.Context;

public class RelasiOpenHelper extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "dbRelasi";
    private static final String TABLE_CREATE =
        "CREATE TABLE RELASI (ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
        "NAMA STRING, TELEPON STRING);";

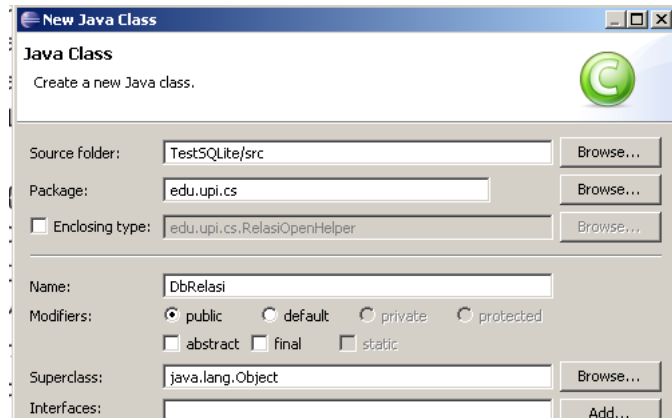
    public RelasiOpenHelper(Context context, String name,
        CursorFactory factory, int version) {
        super(context, DATABASE_NAME, factory, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // dipanggil saat pertamakali DB di create
        db.execSQL(TABLE_CREATE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVer, int newVer) {
        // dipanggil jika versi app berubah
        db.execSQL("DROP TABLE IF EXISTS RELASI");
        onCreate(db);
    }
}

```

Selanjutnya kita akan membuat class yang melakukan manipulasi terhadap data (insert, ambil data), beri nama DbRelasi.



Source code DbRelasi adalah sebagai berikut:

```
package edu.upi.cs;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

public class DbRelasi {

    public static class Relasi {
        public String nama;
        public String telepon;
    }

    private SQLiteDatabase db;
    private final Context con;
    private final RelasiOpenHelper dbHelper;

    public DbRelasi(Context c) {
        //constructor
        con = c;
        dbHelper = new RelasiOpenHelper(con, "", null, 0);
    }

    public void open() {
        db = dbHelper.getWritableDatabase();
    }

    public void close()
    {
        db.close();
    }
}
```



```

//insert data
public long insertRelasi (String nama, String noTelp) {
    ContentValues newValues = new ContentValues();
    newValues.put("NAMA", nama);
    newValues.put("TELEPON", noTelp);
    //insert data ke tabel
    return db.insert("RELASI",null,newValues);
}

//ambil data berdasarkan nama relasi
public Relasi getRelasi(String nama) {
    Cursor cur = null;
    Relasi R = new Relasi();

    //kolom yg akan diambil
    String[] COLS = new String[] { "ID", "NAMA", "TELEPON"};
    //query
    cur = db.query("RELASI", COLS, "NAMA='"+nama+"'", null, null, null, null);
    if (cur.getCount()>0) { //ada data
        cur.moveToFirst();
        R.nama = cur.getString(1); //sesuai urutan COLS, 0 untuk ID, 1 untuk nama
        R.telepon = cur.getString(2); //2 untuk notelp
    }
    return R;
}

//pelajari sendiri untuk delete (db.delete) dan update (db.update)
}

```

Setelah kita membuat class helper dan class dbRelasi, sekarang kita akan memanfaatkan dua kelas tersebut. Kembali ke activity utama yaitu class Main. Tambahkan code sebagai berikut:

```

package edu.upi.cs;

import edu.upi.cs.DbRelasi.Relasi;

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

        DbRelasi db = new DbRelasi(this);
        db.open();

        //test simpan data
        db.insertRelasi("ani", "113");

        //test ambil data
        Relasi r = db.getRelasi("ani");

        AlertDialog dialog = new AlertDialog.Builder(this).create();
        dialog.setMessage("Nama= "+r.nama+ " Telpon= "+r.telepon);
        dialog.show();

        db.close();
    }
}

```

**Latihan:**

Buat aplikasi untuk menyimpan datamahasiswa (NIM, NAMA, ALAMAT). Tersedia menu untuk menambahkan data dan mencari berdasarkan NIM (NIM diinput oleh user).

## ***File***

File dapat disimpan baik di dalam device maupun di media penyimpanan (SD card). Secara default, aplikasi tidak dapat mengakses file milik aplikasi lain.

TBD

## **TBD:**

### Canvas

- Cocok untuk objek bergerak (sprite) maks 10
- Cocok untuk game yang tidak membutuhkan draw untuk setiap frame.
- Sederhana (dibandingkan dengan OpenGL)

### onDraw

Menggunakan view

Menggunakan surfaceview