

# Pemrograman Android

---

BAKA HAYI



---

# Daftar Isi

<b>1</b>	<b>Instalasi SDK dan AVD</b>	<b>1</b>
	Persiapan System . . . . .	1
	Instalasi Android SDK Starter Pack . . . . .	2
	Instalasi ADT plugin untuk Eclipse . . . . .	3
	Menambahkan Platform dan Komponen Lainnya . . . . .	5
	Membuat Virtual Device . . . . .	7
<b>2</b>	<b>Hello, Android</b>	<b>10</b>
	Membuat Project Baru . . . . .	10
	Pengenalan project . . . . .	12
	Menjalankan Aplikasi . . . . .	15
	Latihan . . . . .	16
<b>3</b>	<b>Kalkulator</b>	<b>20</b>
	Langkah-langkah: . . . . .	21
	Latihan . . . . .	31
<b>4</b>	<b>Notepad</b>	<b>37</b>

Langkah-langkah . . . . .	38
Storage System . . . . .	54
Menyimpan, Membaca dan Menghapus Note . . . . .	58
Latihan . . . . .	71
<b>5 Update Facebook</b>	<b>74</b>
Langkah-langkah . . . . .	75
Membuat Aplikasi . . . . .	80

---

# Instalasi SDK dan AVD

## Persiapan System

---

Sebelum memulai mengembangkan aplikasi untuk android, Anda perlu menyiapkan Komputer Anda agar siap untuk digunakan untuk instalasi Android SDK. Beberapa kebutuhannya antara lain:

### 1. Sistem Operasi

- Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)
- Mac OS X 10.5.8 or later (x86 only)
- Linux (tested on Ubuntu Linux, Lucid Lynx)
  - GNU C Library (glibc) 2.7 or later is required.
  - On Ubuntu Linux, version 8.04 or later is required.
  - 64-bit distributions must be capable of running 32-bit applications. For information about how to add support for 32-bit applications, see the Ubuntu Linux installation notes.

### 2. Hardware

Kebutuhan hardware paling tidak menyesuaikan dengan kebutuhan hardware dari Sistem Operasi yang ada di atas. Dan minimal kebutuhan space Hardisk untuk satu platform Android adalah 500MB (diluar JDK dan Eclipse).

### 3. JDK 5 atau JDK 6

Anda harus menginstall JDK terlebih dahulu sebelum bisa mengembangkan aplikasi untuk Android, JRE saja tidak cukup. Untuk mengecek apakah JDK telah terinstall coba jalankan perintah berikut:

```
prompt> javac -version  
javac 1.6.0_20
```

Jika telah muncul versi javac, maka JDK telah terinstall dengan baik.

### 4. Android SDK Starter Package dan Android SDK Component.

### 5. Eclipse IDE

Minimal Eclipse 3.5 (Galileo) atau lebih baru, dan paling tidak telah memiliki:

- Eclipse JDT plugins

Jika Anda belum menginstall Eclipse ada dapat mendownload installernya di <http://www.eclipse.org/downloads/>. Beberapa paket Eclipse yang ada telah disertakan dengan JDT namun ada juga yang belum. Maka dari itu, sebaiknya Anda menginstall salah satu dari paketan berikut:

- Eclipse IDE untuk Java Developers
- Eclipse Classic (versi 3.5.1 atau lebih besar)
- Eclipse IDE untuk Java EE Developers

- ADT Plugins untuk Eclipse

## Instalasi Android SDK Starter Pack

---

### 1. Download versi terakhir dari Android SDK sesuai dengan Sistem Operasi Anda di:

**<http://developer.android.com/sdk/index.html>**

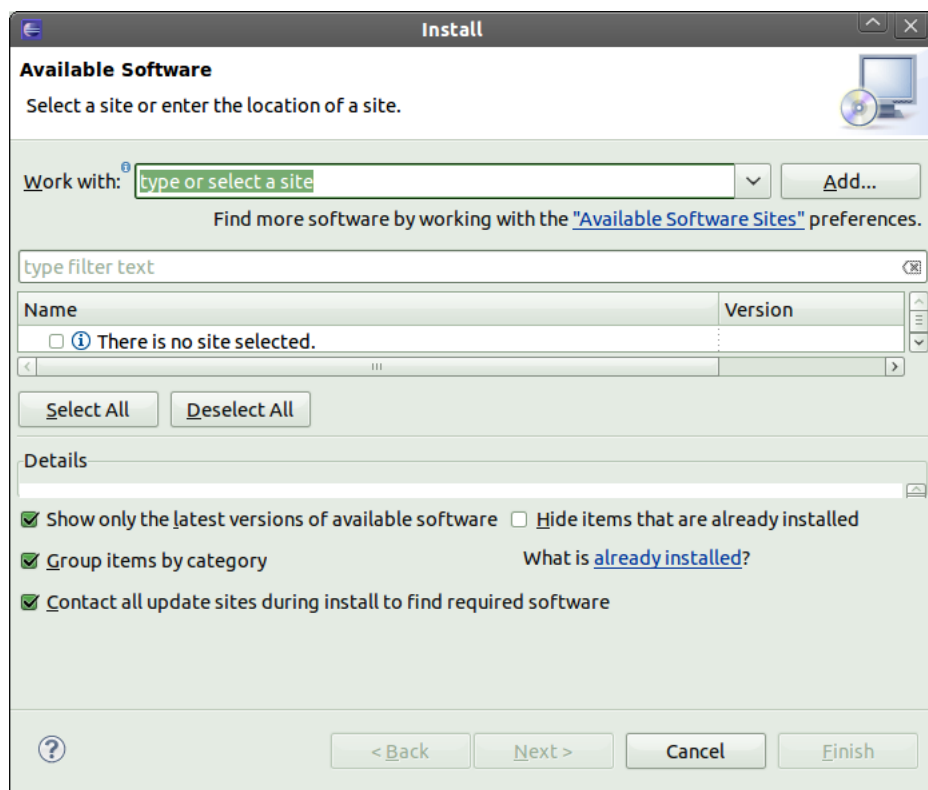
### 2. Untuk paket .zip dan .tgz, ekstrak paket yang telah anda download pada direktory yang aman. Jangan lupa dimana Anda mengekstraknya, karena akan kita butuhkan nanti untuk konfigurasi ADT Plugin di Eclipse dan ketika menggunakan tool-tool untuk menginstall komponen ADT.

3. Khusus untuk windows Anda sebaiknya mendownload yang versi .exe. Untuk menginstallnya, jalankan installer .exe yang Anda download, kemudian installer nanti akan mengecek keberadaan JDK pada system Anda, kemudian install SDK.

## Instalasi ADT plugin untuk Eclipse

---

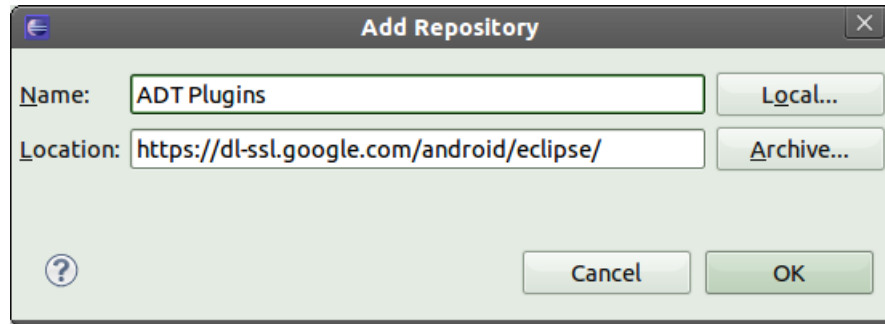
1. Jalankan Eclipse,
2. Pilih menu **Help > Install New Software...**



Gambar 1.1: Install New Software...

3. Pada jendela Install tersebut klik button **Add...** untuk menambahkan repository ADT.

4. Tambahkan alamat repository:  
**<https://dl-ssl.google.com/android/eclipse/>**  
pada location di jendela Add Repository, kemudian klik **Ok**.



Gambar 1.2: Menambah Repository

5. Jika sudah, pada dialog Available Software, centang Developer Tools kemudian klik **Next**.
6. Pada jendela berikutnya Anda akan melihat list dari paket-paket yang akan di download. Kemudian pilih **Next**.
7. Ketika instalasi selesai, restart Eclipse.

**Catatan:** Jika Anda tidak memiliki koneksi internet, sebaiknya Anda menginstallnya melalui offline repository. Filenya dapat Anda download di: **<http://developer.android.com/sdk/eclipse-adt.html>**

Selanjutnya yakni Anda perlu mengarahkan plugin ADT tersebut agar mengetahui lokasi SDK yang telah anda install atau ekstrak di atas. Caranya yakni:

1. Jalankan Eclipse,
2. Pilih menu **Window > Preferences...** untuk membuka jendela konfigurasi. (Pada Mac OS X melalui: **Eclipse > Preferences**)
3. Pada side panel, pilih Android.
4. Klik **Browse** pada *SDK Location* kemudian cari dimana anda mengekstrak atau menginstall Android SDK. Misalnya anda mengekstraknya di /opt maka pada *SDK Location* menjadi: **/opt/android-sdk-<versi-sdk>**.

5. Klik **Apply** kemudian **Ok**.

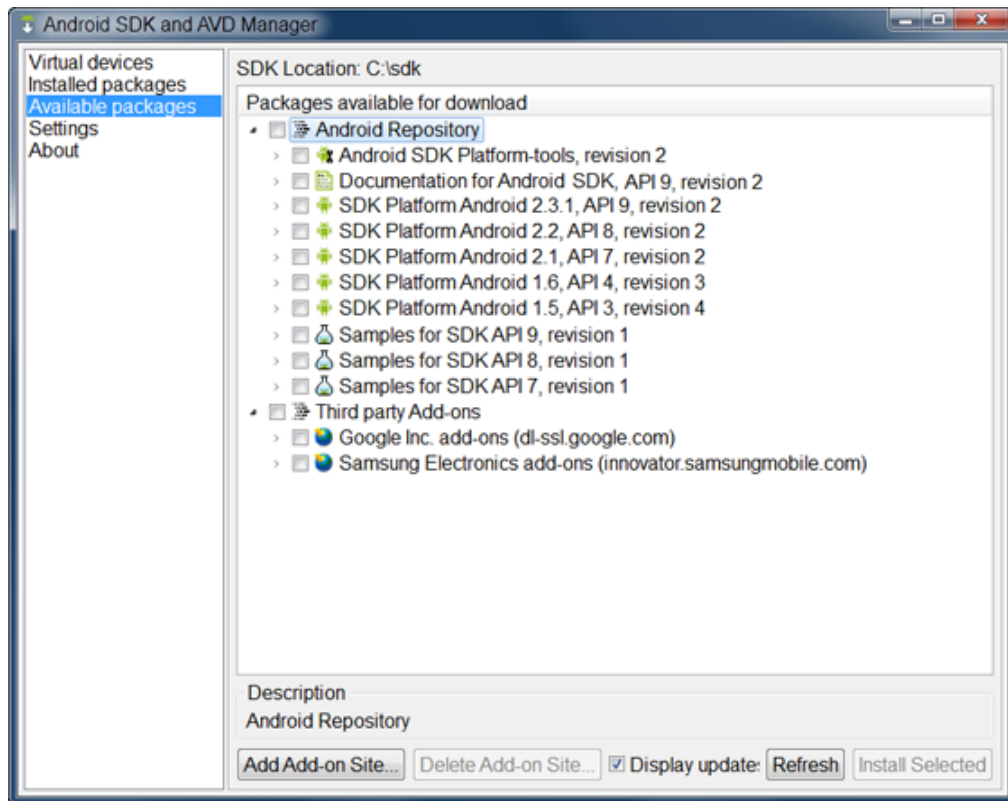
## Menambahkan Platform dan Komponen Lainnya

---

Langkah terakhir untuk dapat menggunakan SDK yakni menambahkan Platform Android ke SDK Anda. Seperti Anda ketahui, Android terdiri dari beberapa Platform yakni dari 1.5 sampai dengan yang terakhir 3.0. Anda dapat menginstall hanya satu saja atau beberapa platform sekaligus dalam SDK anda. Caranya yakni:

1. Jalankan **Android SDK and ADV Manager** dengan cara:
    - Melalui Eclipse: pilih menu **Window >Android SDK and ADV Manager**
    - Melalui Windows: double klik **SDK Manager.exe** pada direktory tempat Android SDK terinstall.
    - Melalui Linux dan Mac OS X: Buka terminal dan masuk ke direktory **tools/** yang ada di dalam direktory SDK Android Anda (misalnya **/opt/android-sdk-linux\_x86**). Kemudian jalankan perintah: **./android**
- ```
kuro@linux:/opt/android-sdk-linux_x86/tools$ ./android
```
2. Pada jendela **Android SDK and ADV Manager** navigasi ke bagian **Available package**





Gambar 1.3: Android SDK and ADV Manager

3. Pilih (centang) paket-paket yang akan di install. Rekomendasi minimal yakni:

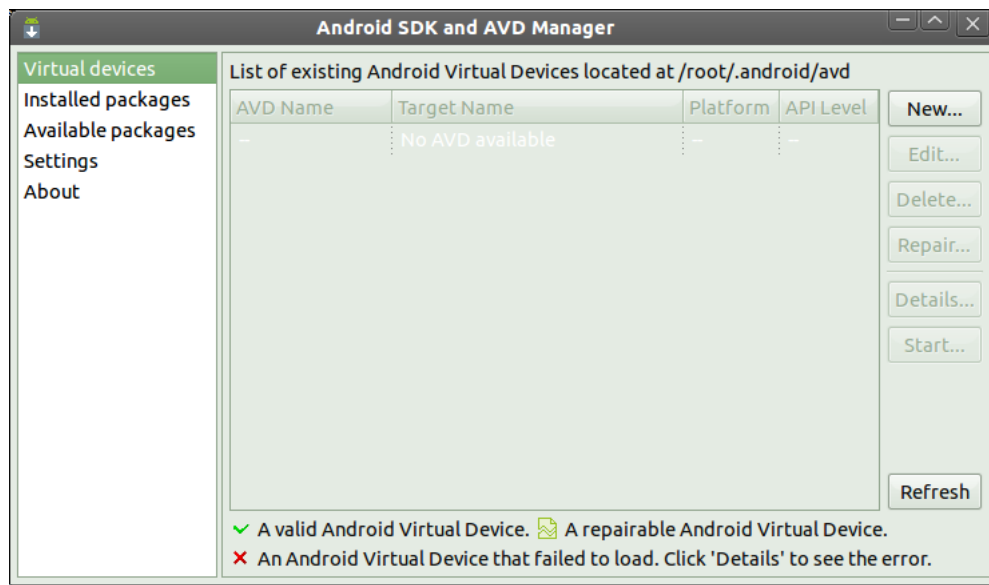
- **Android SDK Tools**
- **Android SDK Platform tools**
- **SDK Platform (sesuai kebutuhan)**
- USB Driver for Windows (Hanya untuk Windows) (Optional)
- Documentation (Optional)
- Sample For SDK API (Optional)
- Google API's (Optional)
- Additional SDK Platforms (Optional)

4. Kemudian klik **Install Selected** untuk menginstall semua paket yang dipilih.
5. Setelah installasi selesai, Anda dapat menambahkan Virtual device jika diinginkan pada SDK Anda.

## Membuat Virtual Device

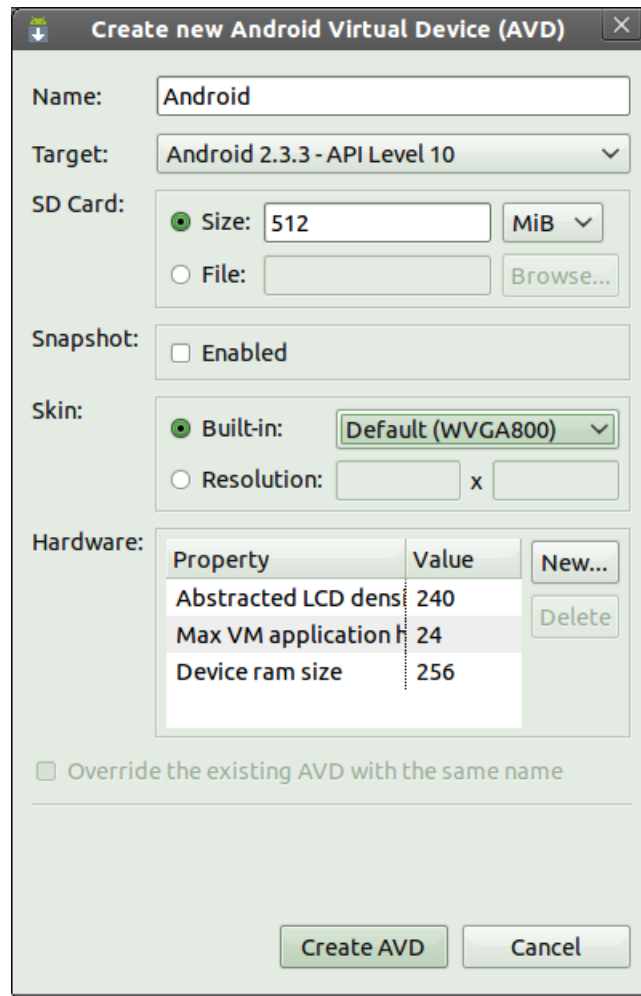
**AVD** atau *Android Virtual Devices* adalah device virtual yang bisa digunakan untuk mengemulasi dan mengujicoba program yang telah kita buat di dalam PC, tanpa harus memiliki perangkat Android itu sendiri. Cara membuatnya yakni:

1. Jalankan **Android SDK and ADV Manager**, caranya dapat Anda lihat pada bagian **Menambahkan Platform** di atas.
2. Pada jendela **Android SDK and ADV Manager** navigasi ke bagian **Virtual devices**



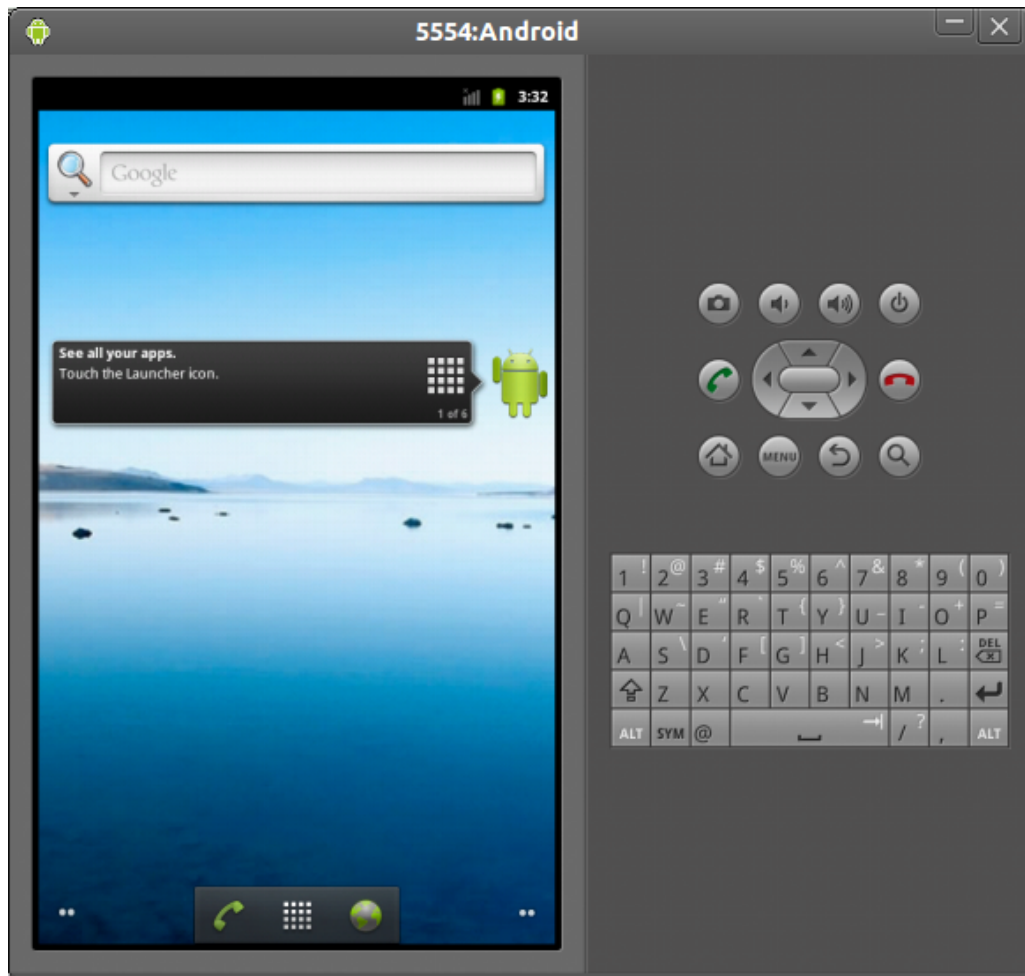
Gambar 1.4: Android SDK and ADV Manager

3. Klik **New...** Untuk membuat AVD baru.



Gambar 1.5: Create New Android Virtual Device (ADV)

4. Beri nama, kemudian pilih Target Platformnya. Untuk ukuran SD Card dapat anda tentukan bebas misalnya 512MiB. Anda juga dapat menambahkan beberapa fitur lainnya untuk AVD tersebut dengan mengklik **New...** pada bagian Hardware.
5. Jika sudah selesai, klik **Create AVD**.
6. Untuk mencoba, pada Jendela **Android SDK and ADV Manager** pilih AVD yang telah Anda buat kemudian klik **Start...** kemudian pilih **Launch** pada jendela Launch Option.



Gambar 1.6: AVD yang telah berhasil dibuat

---

# Hello, Android

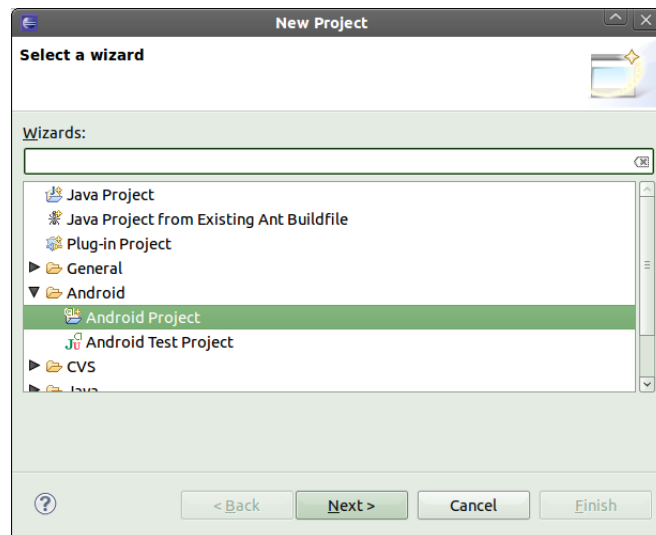
---

## Membuat Project Baru

---

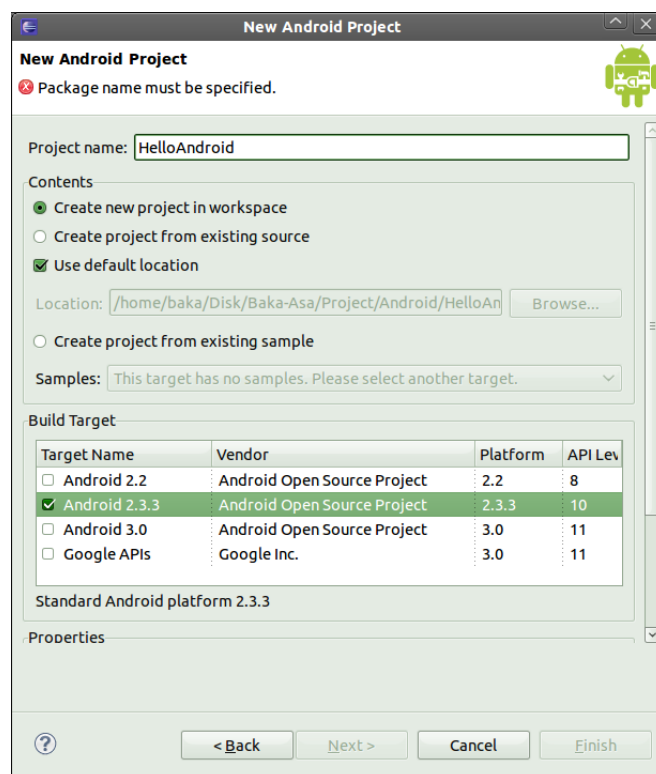
Untuk membuat project Android baru, langkah-langkahnya yakni:

1. Jalankan Eclipse,
2. Pilih menu **File >New >Project...** untuk membuka dialog New Project.



Gambar 2.1: New Project

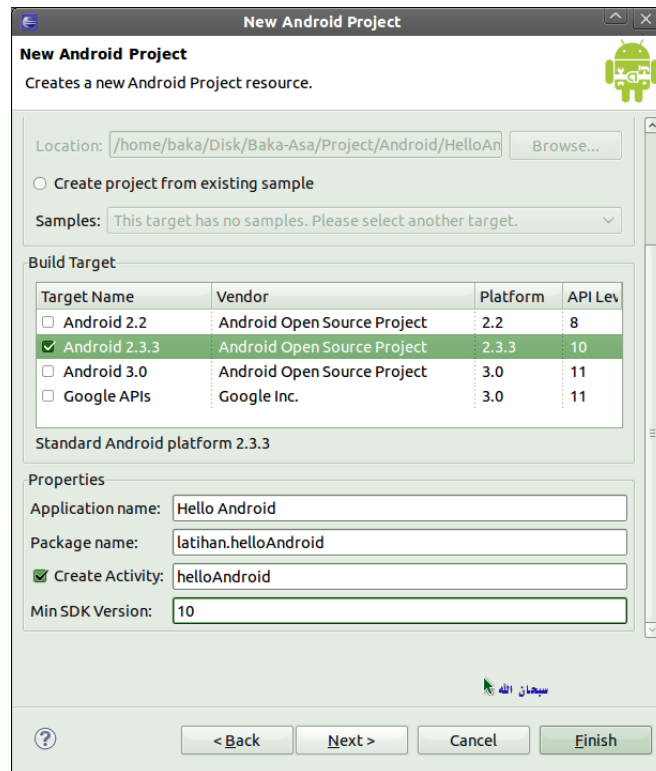
3. Pada jendela **New Project** tersebut, pilih project **Android > Android Project**. Kemudian Klik **Next**>.
4. Selanjutnya pada dialog **New Android Project** isi dengan:
  - **Project name:** HelloAndroid<sup>1</sup>
  - **Contents:** pilih ” *Create new project in workspace*” dan centang ” *Use default location*”
  - **Build target:** Android 2.3.3<sup>2</sup>



Gambar 2.2: New Android Project #1

<sup>1</sup>dapat anda ganti dengan nama yang lain.

<sup>2</sup>Sesuaikan dengan AVD atau target versi Android dimana aplikasi Anda akan berjalan nantinya.



Gambar 2.3: New Android Project #2

- **Application name:** Hello Android
- **Package name:** latihan.helloAndroid<sup>3</sup>
- **Create Activity:** helloAndroid<sup>4</sup>
- **Min SDK Version:** 10 (Atau dikosongkan)

Seperti terlihat pada kedua gambar di atas. Kemudian klik **Finish**.

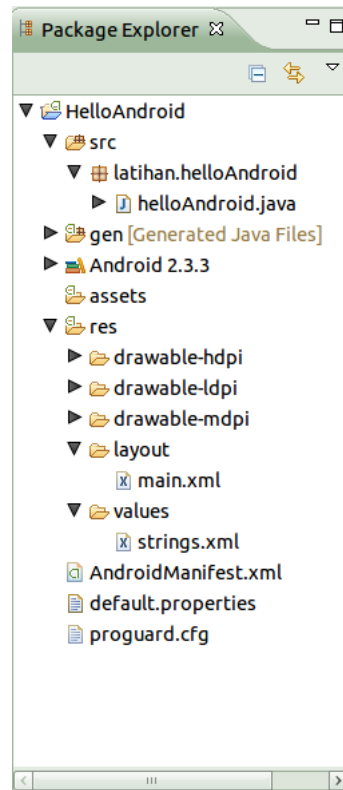
## Pengenalan project

Pada jendela Eclipse, anda akan menemukan tiga sub window lainnya yakni:

<sup>3</sup>harus minimal dua identifier dan dipisan tanda titik.

<sup>4</sup>Hanya boleh satu kata.

## 1. Package Explorer



Pada Package Explorer ini Anda akan menemukan tree project yang telah anda buat. Beberapa hal yang perlu Anda perhatikan dalam project explorer ini yakni:

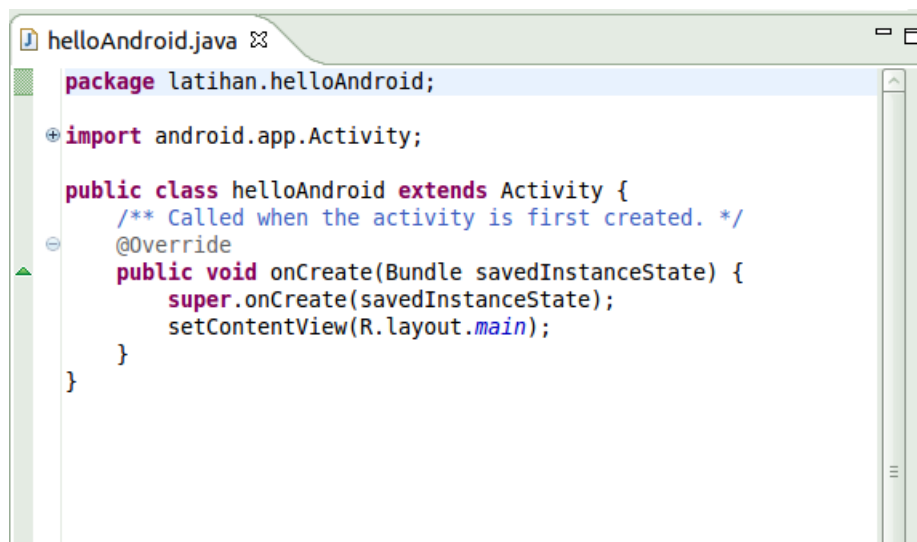
- Source program (dibawah **src/** dan **gen/**)  
File-file yang berada di bawah **src/** adalah source code program yang dapat Anda ubah. Sedangkan yang berada di bawah **gen/** adalah file-file source code yang digenerate otomatis oleh SDK.
- Android 2.3.3 (atau terkadang Android Library)  
Berisi library target Android.
- assets  
Berisi file-file yang akan dipaketkan bersama dengan aplikasi Anda. Pada latihan pertama ini kita masih belum menggunakannya.



- resource (dibawah **res/**)  
Berisi:
  - drawable-\*, yakni resource yang berisi gambar-gambar untuk aplikasi Anda. Untuk aplikasi Hello Android ini hanya berisi gambar Icon aplikasi.
  - Layout (file **res/layout/main.xml**)  
Berisi konfigurasi Layout atau susunan tampilan widget-widget pada layar Android.
  - values  
Berisi konstanta, string dan lain sebagainya yang digunakan pada Aplikasi.
- Manifest (file **AndroidManifest.xml**)  
File XML ini berisi konfigurasi project Android Anda.

## 2. Jendela Editor

Pada jendela editor ini Anda dapat mengedit source code, layout, values dan lain sebagainya. Untuk mencoba, silahkan buka file **helloAndroid.java** yang ada di bawah **src/latihan.helloAndroid/**. Selanjutnya coba buka file **main.xml** yang ada pada **res/layout/** atau file **AndroidManifest.xml**. Jendela editor akan langsung berubah sesuai dengan file apa yang dibuka.

A screenshot of the Android Studio IDE showing the 'helloAndroid.java' file in the editor. The code is as follows:

```
package latihan.helloAndroid;

import android.app.Activity;

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

The editor has a tab at the top labeled 'helloAndroid.java' with a close icon. On the left, there is a vertical toolbar with icons for package explorer, search, and run. On the right, there is a vertical toolbar with icons for layout, values, and other resources.

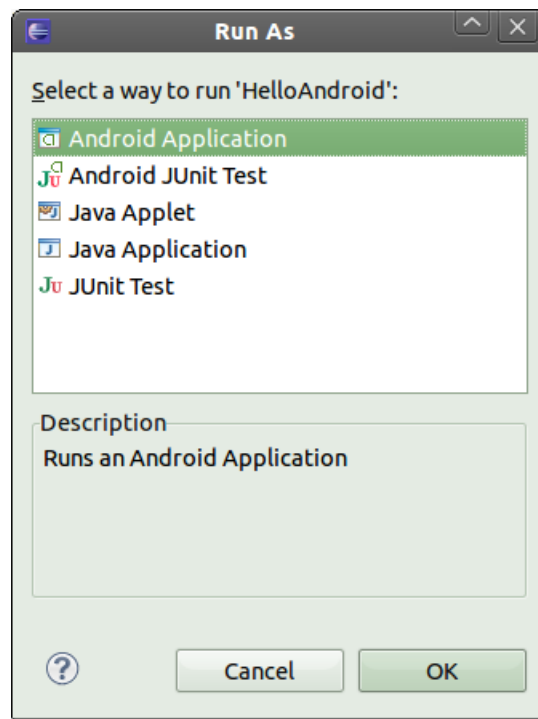
## Menjalankan Aplikasi

---

Untuk menjalankan aplikasi pada AVD, caranya yakni:

- pilih menu **Run >Run** atau
- tekan tombol kombinasi **Ctrl+F11** atau
- tekan tombol Run yang ada pada toolbars.

Pilih **Android Application** pada dialog **Run As**. Kemudian klik **Ok**.



Gambar 2.4: Run As

Tunggu sampai AVD berhasil di load sempurna dan menghasilkan Run program seperti:

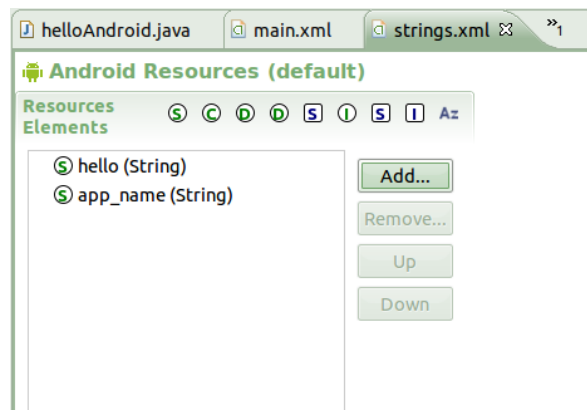


Gambar 2.5: Hello, Android

## Latihan

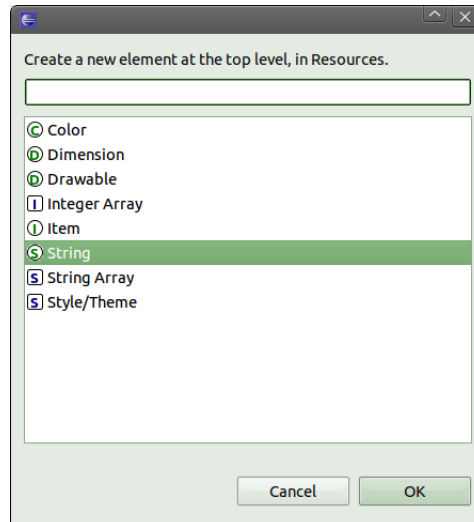
---

1. Buka file **strings.xml** yang ada pada **res/values/strings.xml**.

Gambar 2.6: **string.xml** editor.

Kemudian klik **Add...** untuk menambahkan String.

2. Pada dialog pilih **String** untuk menambahkan sebuah string. Kemudian klik **OK**.

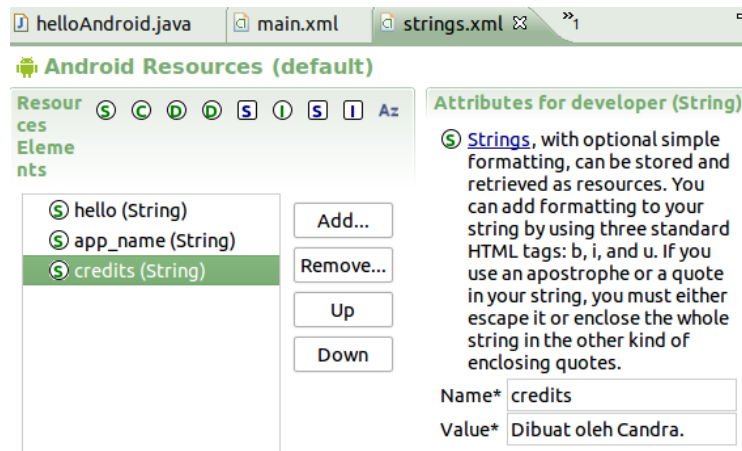


Gambar 2.7: Add new value.

3. Isi string baru dengan:

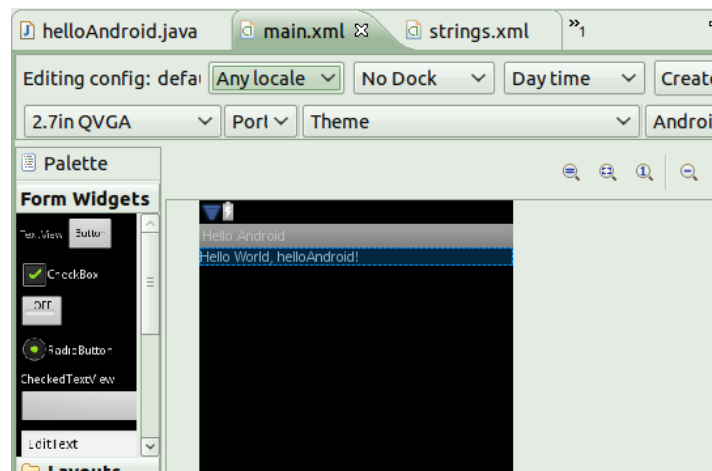
- **Name\***: credits
- **Value\***: Dibuat oleh Candra (atau isi dengan nama Anda).

Kemudian simpan file.



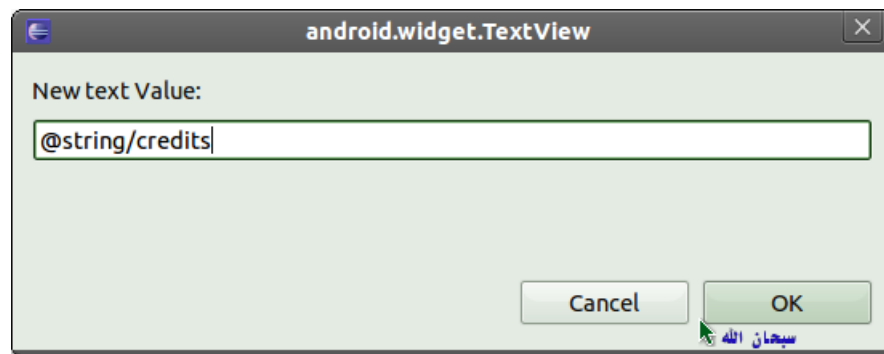
Gambar 2.8: Mengubah atribut String

4. Buka file `res/layout/main.xml`

Gambar 2.9: layout `main.xml`.

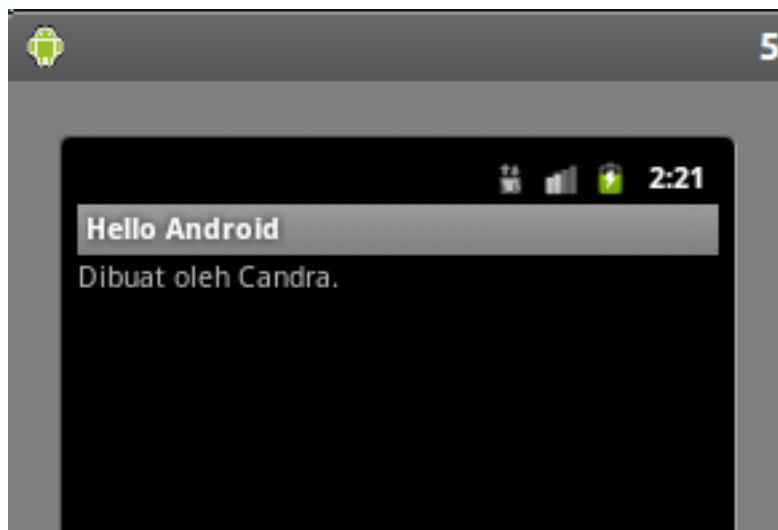
Pada layout, klik kanan pada text "**Hello World, helloAndroid**" kemudian pilih **Properties >Text...**

5. Ubah **New Text Value** yang tadinya berisi `@string/hello` menjadi `@string/credits`. Kemudian klik **OK**.



Gambar 2.10: Mengubah property **Text** dari label.

6. Simpan file, dan Run ulang program. Anda akan mendapatkan tampilan program yang tadinya bertuliskan "*Hello World, helloAndroid*" menjadi "**Dibuat oleh Candra**"



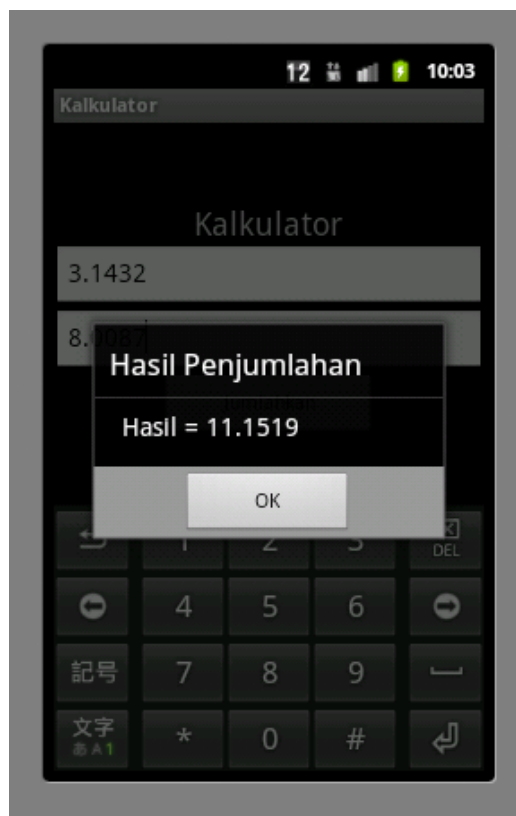
Gambar 2.11: Run Hello Android.

MODUL

3

# Kalkulator

Target:



Gambar 3.1: Kalkulator

## Langkah-langkah:

---

### Membuat Project

1. Buat sebuah Android Project baru, beri nama project **Kalkulator**, dengan konfigurasi:  
Project name: **Kalkulator**  
Build target: **Android 2.3.3** (sesuaikan dengan AVD yang ada)  
Application name: **Kalkulator**  
Package name: **latihan.Kalkulator**  
Create Activity: **Kalkulator**  
Min SDK Version: **10** (atau dikosongkan)

### Menyiapkan Resource

2. Buka file **string.xml** yang berada pada **res/values/string.xml**. Kemudian Tambahkan beberapa item:
  - String
    - **Name\***: labelJumlahkan
    - **Value\***: Jumlahkan
  - String
    - **Name\***: buttonHandler
    - **Value\***: buttonHandler

Kemudian simpan file.

### Membuat Layout UI

3. Buka file **main.xml** yang ada pada **res/layout/**. Kemudian tambahkan beberapa komponen sehingga menjadi (pastikan anda berada pada mode *Graphical layout*):



Gambar 3.2: Layout `main.xml`

## Mengubah property komponen

- Setelah layout dibuat, kemudian ubah property masing-masing komponen. Caranya dengan meng-klik kanan komponen yang akan di ubah propertynya. kemudian pilih menu **Properties** >[property yang akan diubah]. Ubah propert masing-masing komponen menjadi:

- **textView1**

- **Text...** : @string/app\_name
- **Text Size...** : 24sp
- **Layout gravity** : Center
- **Gravity** : Bottom
- **Layout Height** : 90dip

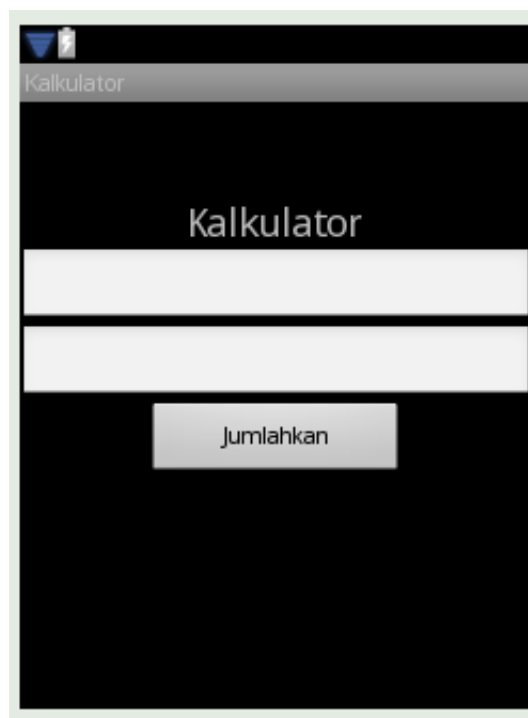
- **editText1**

- **Edit ID...** : editPertama
- **Text...** : (kosongkan)
- **Numeric** : Decimal

- **editText2**

- **Edit ID...** : editKedua
- **Text...** : (kosongkan)

- **Numeric** : Decimal
  - **button1**
    - **Text...** : @string/labelJumlahkan
    - **OnClick...** : @string/buttonHandler
    - **Layout gravity** : Center
    - **Gravity** : Center
    - **Layout Width** : 160dip
5. Untuk mengubah property ID caranya yakni: klik kanan pada komponen, kemudian pilih **Edit ID...**. Sedangkan untuk mengatur property Layout Height dan Layout Width caranya yakni: Klik kanan komponen kemudian pilih **Layout Width** atau **Layout Height** kemudian pilih **Other...**, setelah itu set ukurannya. Sehingga layout akan terlihat seperti:

Gambar 3.3: Layout **main.xml**

6. Kemudian simpan semua perubahan yang telah dilakukan.

## Menambahkan Program

7. Untuk melakukan penjumlahan kita perlu menambahkan logika penjumlahan ke dalam program, caranya yakni:

- (a) Buka file **kalkulator.java** yang berada pada **src/latihan.Kalkulator/**.
- (b) Tambahkan beberapa baris kode program berikut setelah deklarasi **package**:

```
1 package latihan.kalkulator;
2
3 import android.app.Activity;
4 import android.app.AlertDialog;
5 import android.content.DialogInterface;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.EditText;
9 import android.widget.Toast;
```

- (c) Tambahkan kode program berikut (yang diberi warna biru) pada kelas Kalkulator:

```
10 ...
11 public class Kalkulator extends Activity {
12     /** Called when the activity is first created. */
13     private EditText angka1;
14     private EditText angka2;
15
16     @Override
17     ...
```

- (d) Pada method **onCreate** tambahkan bebarap baris berikut:

```
15 ...
16     @Override
17     public void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.main);
20         angka1 = (EditText) findViewById(R.id.editPertama);
21         angka2 = (EditText) findViewById(R.id.editKedua);
22     }
23     ...
```

Untuk mensinkronisasikan antara Edit Text **angka1** dengan Edit Text **editPertama** yang ada pada UI (pada layout yang telah dibuat di atas). Begitu juga dengan Edit Text **angka2**.

- (e) Tambahkan methode **buttonHandler** setelah methode **onCreate**. Kemudian isi dengan kode berikut:

```
23 public void buttonHandler(View view) {
24     double a, b, hasil;
25     if(angka1.getText().length() == 0) {
26         Toast.makeText(this,
27             "Angka pertama masih kosong!",
28             Toast.LENGTH_LONG).show();
29         return;
30     }
31     if(angka2.getText().length() == 0) {
32         Toast.makeText(this,
33             "Angka kedua masih kosong!",
34             Toast.LENGTH_LONG).show();
35         return;
36     }
37     a = Double.parseDouble(angka1.getText().toString());
38     b = Double.parseDouble(angka2.getText().toString());
39     hasil = a + b;
40
41     AlertDialog.Builder builder =
42         new AlertDialog.Builder(this);
43     builder.setTitle("Hasil Penjumlahan");
44     builder.setMessage("Hasil = "+String.valueOf(hasil))
45     .setPositiveButton("OK",
46         new DialogInterface.OnClickListener() {
47
48             @Override
49             public void onClick(DialogInterface dialog,
50                 int which) {
51                 // TODO Auto-generated method stub
52                 angka1.setText("");
53                 angka2.setText("");
54                 dialog.dismiss();
```

```
55     }  
56   });  
57   AlertDialog dialogHasil = builder.create();  
58   dialogHasil.show();  
59 }
```

Penjelasan:

**Baris ke-23 :**

```
23 | public void buttonHandler(View view) {
```

Methode **buttonHandler**, methode ini digunakan untuk meng-handle event **On Click** untuk **button1** (property **On Click** pada **button1**)

**Baris ke-24 :**

```
24 | double a, b, hasil;
```

Membuat variabel **a**, **b** dan **hasil** untuk menampung nilai-nilai double yang dimasukkan ke Edit Text **editPertama** dan **editKedua**.

**Baris ke-25 sampai 29 :**

```
25 |         if(angka1.getText().length() == 0) {  
26 |             Toast.makeText(this,  
27 |                 "Angka pertama masih kosong!",  
28 |                 Toast.LENGTH_LONG).show();  
29 |             return;  
30 |         }
```

Membuat logika untuk handle jika Edit Text **editPertama** masih kosong. Objek Toast disini digunakan untuk menampilkan tooltip atau baloon yang berisi pesan "**Angka pertama masih kosong!**".

Gambar 3.4: Tampilan **Toast** untuk **editPertama**

Baris ke-31 sampai 35 :

```
31         if(angka2.getText().length() == 0) {
32             Toast.makeText(this,
33                 "Angka kedua masih kosong!",
34                 Toast.LENGTH_LONG).show();
35             return;
36         }
```

Sama dengan baris ke-25 sampai 29, akan tetapi ini untuk Edit Text **editKedua**.

Gambar 3.5: Tampilan **Toast** untuk **editKedua**

Baris ke-37 sampai 38 :

```

37 |         a = Double.parseDouble(angka1.getText().toString());
38 |         b = Double.parseDouble(angka2.getText().toString());

```

Baris ke-37: menginisialisasi variabel **a** dengan nilai double dari hasil **parseDouble()** dari string yang ada pada Text Edit **angka1** (ingat, Edit Text **angka1** telah disinkronisasikan dengan Edit Text **editPertama** yang ada di layout). Demikian juga dengan baris ke-38, yakni untuk mengisi variabel **b** dengan hasil **parseDouble** dari string yang ada pada Edit Text **angka2**.

Baris ke-39 :

```

39 |         hasil = a + b;

```

Menjumlahkan nilai dalam variabel **a** dengan **b**. Kemudian hasil penjumlahan disimpan ke dalam variabel **hasil**.

Baris ke-41 sampai 42 :

```
41 |         AlertDialog.Builder builder =  
42 |             new AlertDialog.Builder(this);
```

Membuat objek **builder** dari kelas **AlertDialog.Builder**. Objek ini akan kita gunakan untuk membuat Alert dialog, tempat menampilkan hasil penjumlahan.

Baris ke-43 :

```
43 |         builder.setTitle("Hasil Penjumlahan");
```

Mengatur title dari alert dialog dengan string "**Hasil Penjumlahan**".

Baris ke-44 sampai 56 :

```
44 |         builder.setMessage("Hasil = "+String.valueOf(hasil))  
45 |         .setPositiveButton("OK",  
46 |             new DialogInterface.OnClickListener() {  
47 |  
48 |             @Override  
49 |             public void onClick(DialogInterface dialog,  
50 |                 int which) {  
51 |                 // TODO Auto-generated method stub  
52 |                 angka1.setText("");  
53 |                 angka2.setText("");  
54 |                 dialog.dismiss();  
55 |             }  
56 |         });
```

Menampilkan isi pesan dalam dialog dengan string "**Hasil =** " daigabungkan dengan string dari variabel **hasil**. Kemudian membuat sebuah button **OK** pada dialog. Dengan event ketika button **OK** tersebut di-klik, maka string yang ada pada Edit Text **angka1** dan **angka2** dikosongkan atau di set dengan "" (baris ke-52 dan 53). Kemudian di akhirnya dialog tersebut di **dismiss()** untuk menutup dialog.



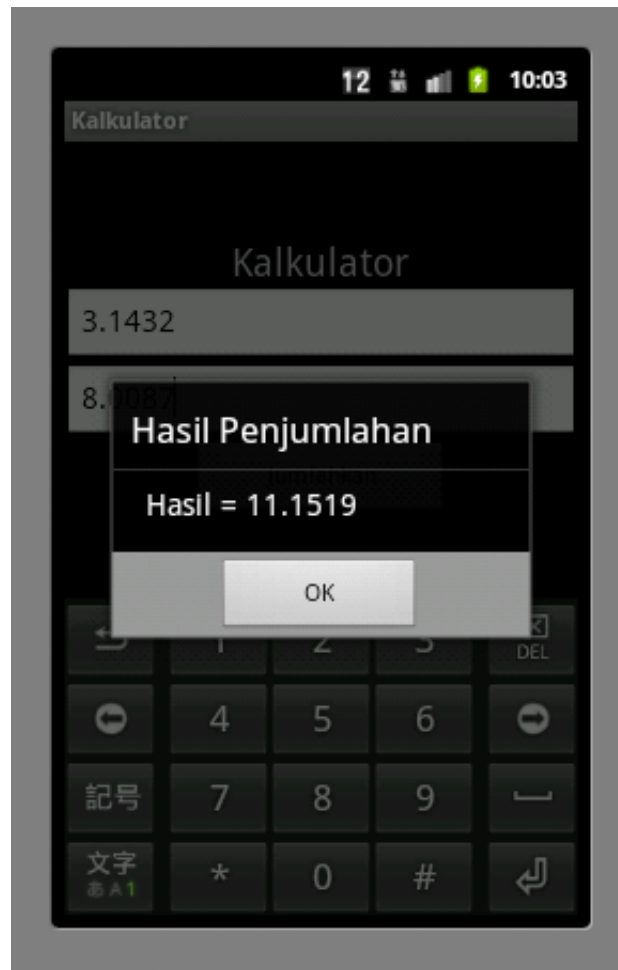
Baris ke-57 sampai 58 :

```
57 |      AlertDialog dialogHasil = builder.create();  
58 |      dialogHasil.show();
```

Membuat objek **dialogHasil** dari kelas **AlertDialog** dan diinisialisasi dengan object **builder.create()**. Kemudian **dialogHasil** tersebut ditampilkan pada layar dengan memanggil metode **show()**.

**Keterangan:** Objek **dialogHasil** dari kelas **AlertDialog** digunakan untuk membuat dialog untuk menampilkan hasil.

8. Simpan semua perubahan dan **Run** program.



Gambar 3.6: Hasil Run program.

## Latihan

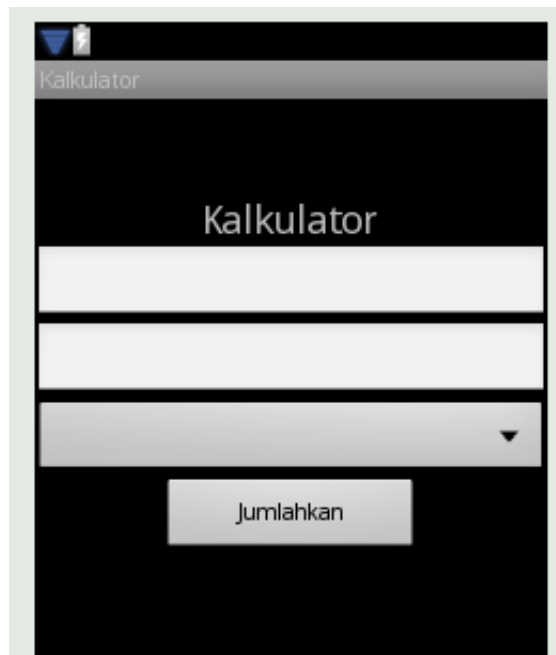
---

1. Tambahkan value baru pada **string.xml** yakni:
  - String
    - Name\* : **labelProses**
    - Value\* : **Proses**

- String Array ; Name\* : **prosesList**
  - Item
    - \* Value\* : **Penjumlahan**
  - Item
    - \* Value\* : **Pengurangan**
  - Item
    - \* Value\* : **Perkalian**
  - Item
    - \* Value\* : **Pembagian**

Kemudian simpan perubahan.

2. Tambahkan sebuah widget/komponen baru yakni **Spinner** pada Layout. Sehingga menjadi:



Gambar 3.7: Menambahkan Spinner

3. Atur property Spinner menjadi:

- ID : **spinnerOperasi**

- Entries... : **prosesList** (ada pada Array/prosesList)

4. Ubah property text pada button1 menjadi:

- Text... : **@string/labelProses**

Sehingga layout akan terlihat seperti:



Gambar 3.8: Menambahkan Spinner

5. Ubah program menjadi (Tambahkan bagian-bagian yang berwarna biru):

```
11 ...  
12 public class kalkulator extends Activity {  
13     /** Called when the activity is first created. */  
14     private EditText angka1;  
15     private EditText angka2;  
16     private Spinner spinOperasi;  
17 }
```

```
18     @Override
19     public void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.main);
22         angka1 = (EditText) findViewById(R.id.editPertama);
23         angka2 = (EditText) findViewById(R.id.editKedua);
24         spinOperasi = (Spinner) findViewById(
25             R.id.spinnerOperasi);
26     }
27
28     public void buttonHandler(View view) {
29         double a, b, hasil;
30         if(angka1.getText().length() == 0) {
31             Toast.makeText(this,
32                 "Angka pertama masih kosong!",
33                 Toast.LENGTH_LONG).show();
34             return;
35         }
36         if(angka2.getText().length() == 0) {
37             Toast.makeText(this,
38                 "Angka kedua masih kosong!",
39                 Toast.LENGTH_LONG).show();
40             return;
41         }
42         a = Double.parseDouble(angka1.getText().toString());
43         b = Double.parseDouble(angka2.getText().toString());
44         //hasil = a + b;
45
46         switch(spinOperasi.getSelectedItemPosition()) {
47             case 0:
48                 hasil = a + b;
49                 break;
50             case 1:
51                 hasil = a - b;
52                 break;
53             case 2:
54                 hasil = a * b;
55                 break;
```

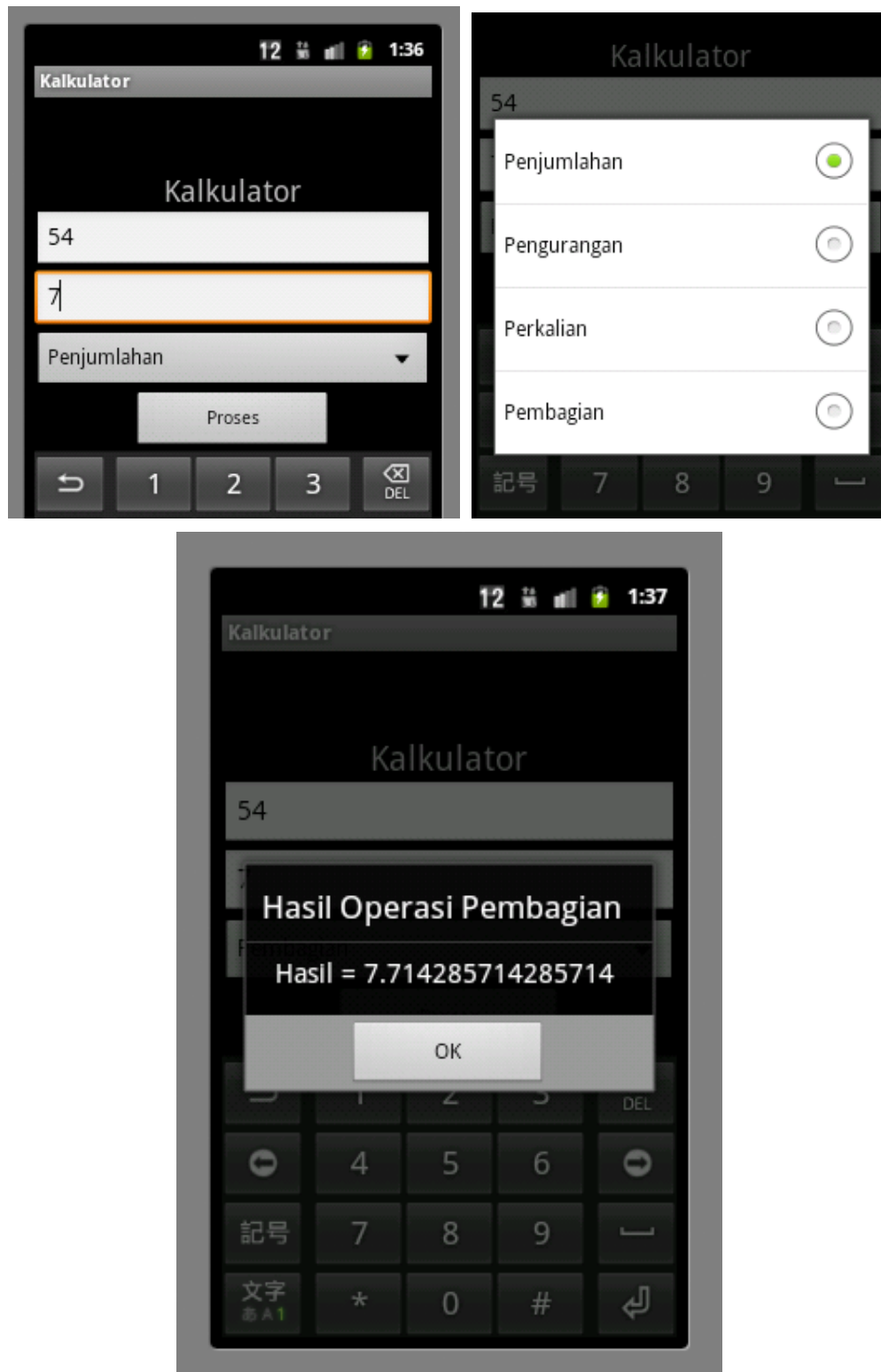
```

56         case 3:
57             hasil = a / b;
58             break;
59         default:
60             hasil = a + b;
61             break;
62     }
63
64     AlertDialog.Builder builder =
65         new AlertDialog.Builder(this);
66     builder.setTitle("Hasil Operasi "
67         + spinOperasi.getSelectedItem().toString());
68     builder.setMessage("Hasil = "+String.valueOf(hasil))
69     .setPositiveButton("OK",
70         new DialogInterface.OnClickListener() {
71
72         @Override
73         public void onClick(DialogInterface dialog,
74             int which) {
75             // TODO Auto-generated method stub
76             angka1.setText("");
77             angka2.setText("");
78             dialog.dismiss();
79         }
80     });
81     AlertDialog dialogHasil = builder.create();
82     dialogHasil.show();
83 }
84 }

```

Penambahan bagian kode program ini bertujuan untuk handle perubahan pilihan pada **spinnerOperasi**. Jika pilihan pada spinner adalah **Penjumlahan** maka yang akan dioperasikan adalah **case 0** ketika button **Proses** di-klik. Dan jika pilihan pada spinner adalah **Perkalian** maka yang akan dikerjakan adalah **case 2**, begitu juga untuk pilihan-pilihan lainnya.

6. Simpan dan Run program yang telah diubah tadi.



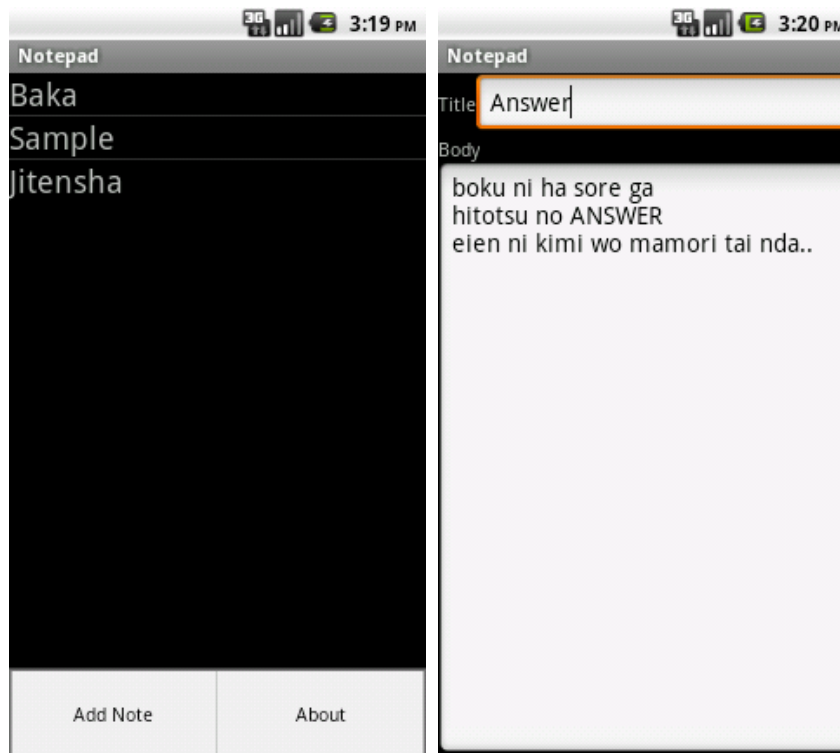
Gambar 3.9: Hasil Run

MODUL

4

# Notepad

Target:



(a) Note List

(b) Edit Note

Gambar 4.1: Target Akhir



## Langkah-langkah

---

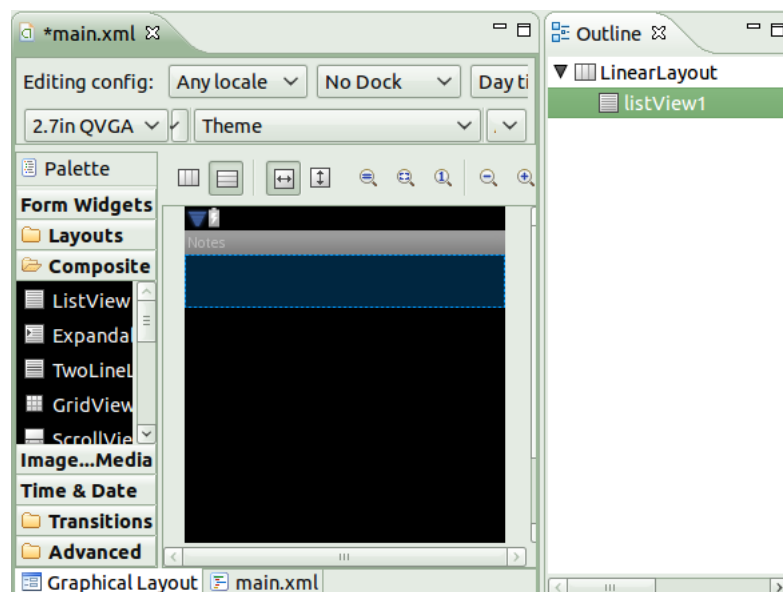
### Membuat Project

1. Buat sebuah project baru dengan konfigurasi:  
Project name: **Notepad**  
Build target: **Android 2.3.3**  
Application name: **Catatan**  
Package name: **latihan.Notepad**  
Create Activity: **Notepad**  
Min SDK Version: **10** (atau dikosongkan)

### Membuat Layout

#### Layout Utama

2. Pada layout **main.xml**, tambahkan sebuah List View.



Gambar 4.2: Menambahkan List View pada Layout **main.xml**

**Catatan:** List View berada pada Palette **Composite**.

3. Atur property List View tersebut menjadi:

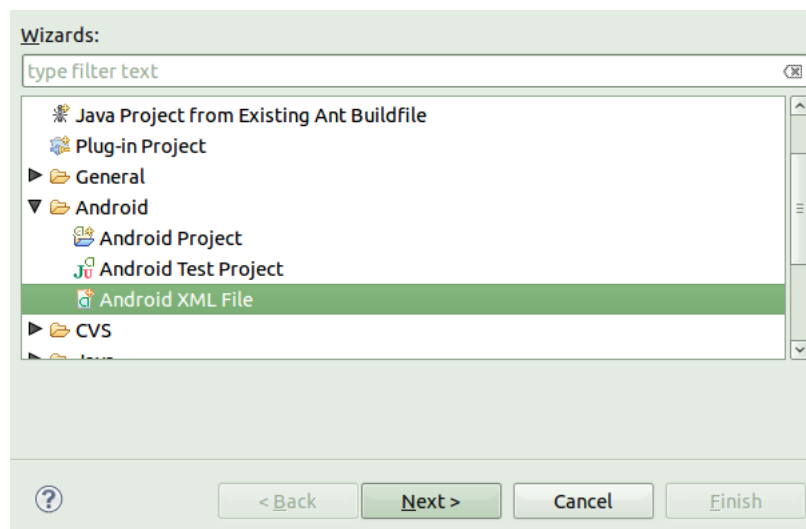
- ID : **listNote**
- Layout Height : **Match Parent**
- Layout Width : **Match Parent**

#### Layout edit\_note

4. Buka **string.xml** kemudian tambahkan beberapa String baru:

- String
  - Name\* : **note\_title**
  - Value\* : **Title**
- String
  - Name\* : **note\_body**
  - Value\* : **Body**

5. Untuk membuat sebuah layout baru caranya yakni: melalui menu **File > New > Other...** kemudian Pilih **Android > Android XML File**. Kemudian tekan **Next >**.

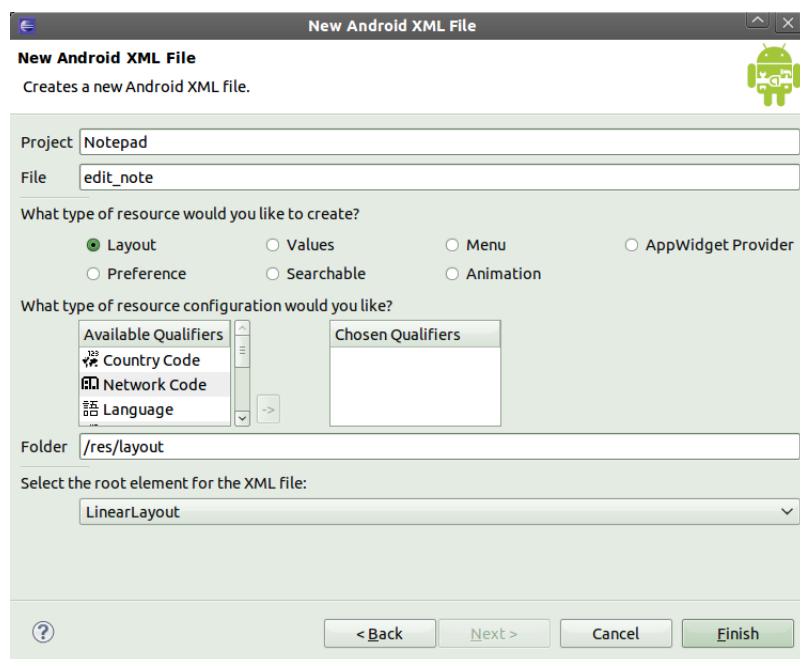


Gambar 4.3: New Android XML File

6. Pada jendela **New Android XML File** buat File XML dengan konfigurasi:

- Project : **Notepad**
- File : **edit\_note**
- Type : **Layout**
- Folder : **/res/layout**

Kemudian klik **Finish** untuk membuat file.

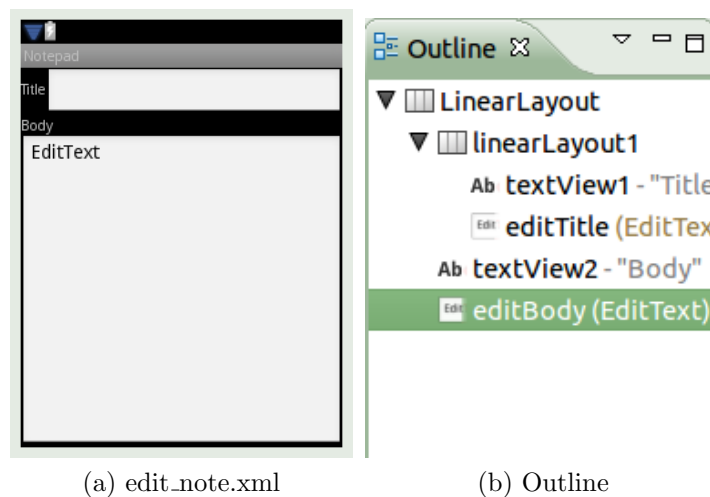


Gambar 4.4: New Android XML File

7. Ubah Orientasi Linier Layout yang telah ada menjadi **Vertikal**. Caranya Klik kanan pada layout (Bisa melalui Outline Dock yang berada di sebelah kanan jendela Eclipse), kemudian pilih **Orientation>Vertical**.
8. Tambahkan beberapa komponen baru ke layout yakni:
- Linier Layout (pada palette **Layouts**)
    - Text View

- \* Text... : **@string/note\_title**
- Edit Text
  - \* ID : **editTitle**
  - \* Text... : (Kosongkan)
  - \* Single Line : True
- Text View
  - Text... : **@string/note\_body**
- Edit Text
  - ID : **editBody**
  - Text... : (Kosongkan)
  - Gravity : Top

Sehingga layoutnya menjadi:

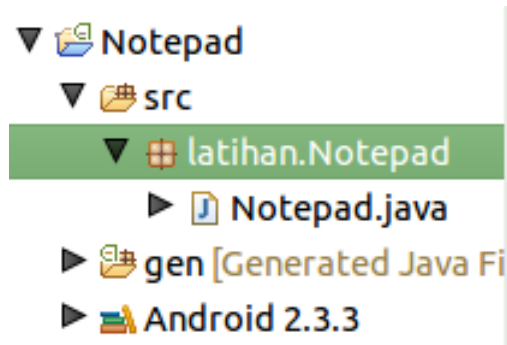


Gambar 4.5: Layout **edit\_note.xml** dan Outline-nya

### Source code untuk layout **edit\_note.xml**

9. Untuk dapat memanggil layout **edit\_note.xml** yang telah di buat di atas. Kita perlu membuat sebuah Kelas yang bertugas untuk meng-handle layout tersebut. Cara membuatnya yakni:

- (a) Pada **Package Explorer**, klik kanan package **latihan.Notepad** yang ada di bawah **/Notepad/src/**. Pada popup menu, pilih **New >Class**

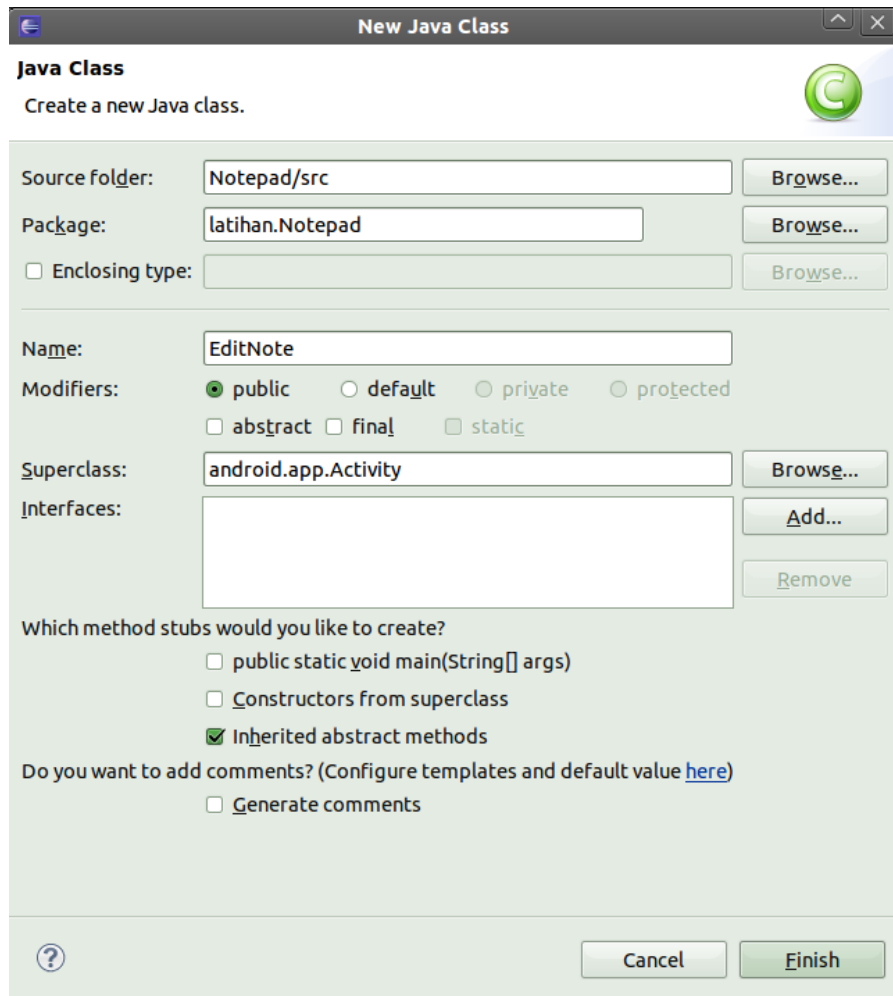


Gambar 4.6: Membuat Class Baru

- (b) Pada jendela **New Class**, Buat kelas dengan konfigurasi sebagai berikut:

- Source Folder : **Notepad/src**
- Package : **latihan.Notepad**
- Name : **EditNote**
- Modifier : **Public**
- Superclass : **android.app.Activity**

Kemudian Klik **Finish**.



Gambar 4.7: Membuat Class Baru

- (c) Kemudian buka file **EditNote.java** yang telah dibuat dari langkah di atas. Lalu tambahkan beberapa baris kode berikut:

```

1 package latihan.Notepad;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5
6 public class EditNote extends Activity {
7

```

```
8      @Override
9      public void onCreate(Bundle savedInstanceState) {
10          super.onCreate(savedInstanceState);
11          setContentView(R.layout.edit_note);
12      }
13  }
14
```

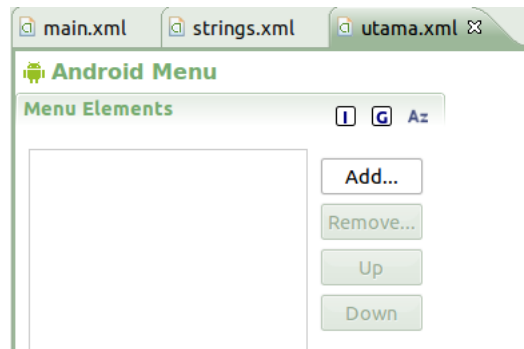
(d) Kemudian simpan file.

## Menambahkan Menu

10. Buka **string.xml** kemudian tambahkan beberapa String baru:
  - String
    - Name\* : **menu\_add\_new**
    - Value\* : **Add New**
  - String
    - Name\* : **menu\_about**
    - Value\* : **About**
11. Untuk membuat menu caranya sama dengan yang ada pada point 5 di atas, yakni melalui menu **File >New >Other...** kemudian pilih **Android >Android XML File**.
12. Pada jendela **New Android XML File** buat File XML dengan konfigurasi:
  - Project : **Notepad**
  - File : **utama**
  - Type : **Menu**
  - Folder : **/res/menu**

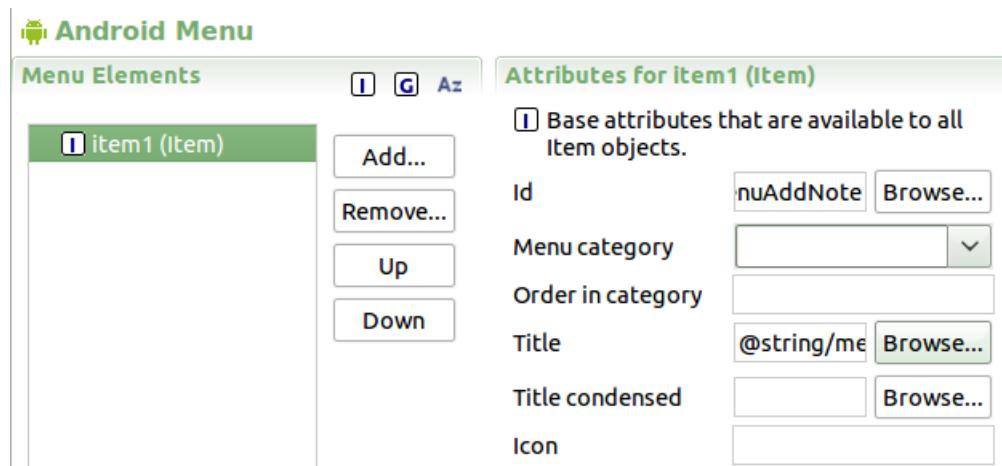
Kemudian klik **Finish** untuk membuat file.

13. Buka file **utama.xml** yang berada di bawah **res/menu/**.



Gambar 4.8: Menu Editor

14. Tambahkan dua buah Menu Item dengan mengklik **Add...** Pilih "Create a new element at the top level, in Menu" kemudian pilih **Item**.



Gambar 4.9: Menu Item

Atur atribut masing-masing menu item menjadi:

- Item
  - Id : `@+id/menuAddNote`
  - Title : `@string/menu_add_note`
- Item
  - Id : `@+id/menuAbout`
  - Title : `@string/menu_about`

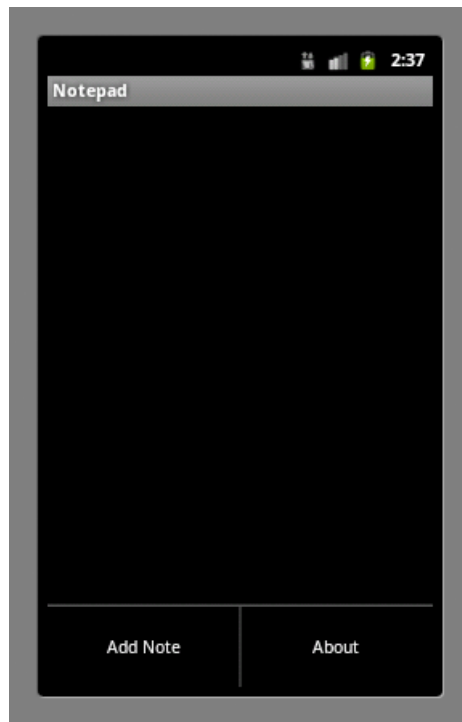


## Menambahkan Menu *utama* ke Activity Notepad

15. Sekarang, buka file **Notepad.java** yang ada pada **src/latihan.Notepad/**. Kemudian tambahkan beberapa baris kode program berikut:

```
1 package latihan.Notepad;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.Menu;
6
7 public class Notepad extends Activity {
8     /** Called when the activity is first created. */
9
10    @Override
11    public void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.main);
14    }
15
16    @Override
17    public boolean onCreateOptionsMenu(Menu menu) {
18        getMenuInflater().inflate(R.menu.utama, menu);
19        return super.onCreateOptionsMenu(menu);
20    }
21 }
```

16. Sekarang coba Run program. Pada AVD, tekan button menu untuk melihat apakah menu sudah ada pada program.



Gambar 4.10: Tampilan menu

Akan tetapi menu-menu itu masih belum memiliki handler di dalam program.

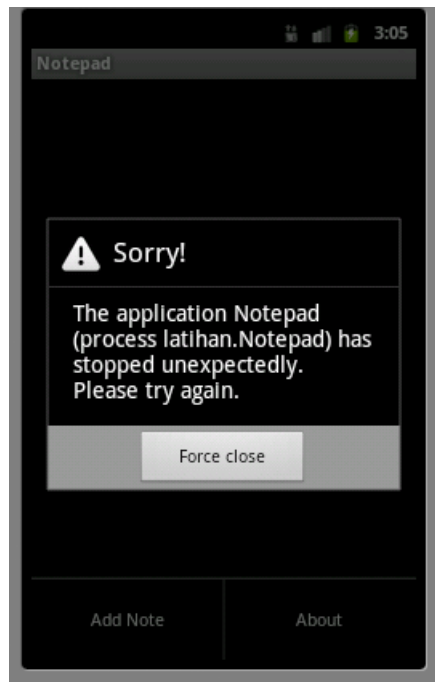
### Memanggil Kelas EditNote dari Menu.

17. Untuk memanggil kelas EditNote kita perlu membuat handler untuk masing-masing Menu Item terlebih dahulu. Untuk membuat handler, tambahkan metode **onMenuItemSelected** berikut pada program:

```
1 package latihan.Notepad;
2 ...
3 public class Notepad extends Activity {
4     /** Called when the activity is first created. */
5     private static final int ACTIVITY_CREATE=0;
6     private static final int ACTIVITY_EDIT=1;
7     ...
```

```
8      @Override
9      public boolean onCreateOptionsMenu(Menu menu) {
10          getMenuInflater().inflate(R.menu.utama, menu);
11          return super.onCreateOptionsMenu(menu);
12      }
13
14      @Override
15      public boolean onOptionsItemSelected
16      (int featureId, MenuItem item) {
17          switch(item.getItemId()) {
18              case R.id.menuAddNote: //handler untuk menu Add Note
19              createNote();
20              return true;
21              case R.id.menuAbout: // handler untuk menu About
22              showAbout();
23              return true;
24          }
25          return super.onOptionsItemSelected(featureId, item);
26      }
27
28      private void showAbout() {
29          // Dialog About. Belum diimplementasikan.
30      }
31
32      private void createNote() {
33          // Memanggil Kelas EditNote
34          Intent i = new Intent(this,EditNote.class);
35          startActivityForResult(i, ACTIVITY_CREATE);
36      }
37  }
```

18. Simpan dan Run program. Kemudian coba pilih menu Add Note pada program. Apa yang terjadi?



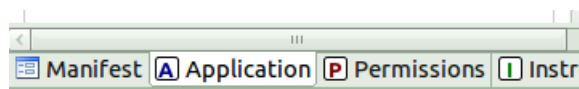
Gambar 4.11: Crash

Program akan Crash. Ini bukan terjadi karena kesalahan dalam program, akan tetapi ada satu langkah penting lagi yang harus dilakukan untuk dapat memanggil Activity lainnya.

## Menambahkan Activity

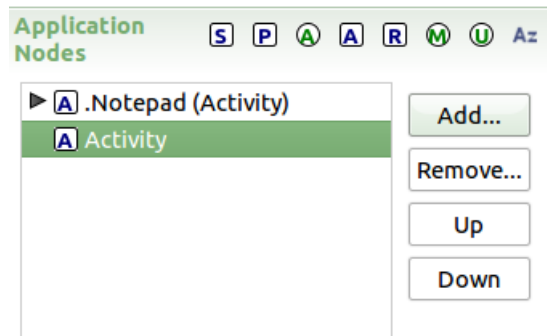
Untuk dapat menanggulangi masalah di atas, kita perlu menambahkan/mengenalkan Activity yang telah kita buat (EditNote) ke **AndroidManifest.xml**. Caranya:

19. Buka file **AndroidManifest.xml**.
20. Navigasi ke tab **Application** (tab berada di bagian bawah).



Gambar 4.12: Tab Application

21. Scroll layar ke bawah. Kemudian pada bagian **Application Nodes** Klik **Add...** kemudian pilih **Activity** untuk menambahkan Activity baru.



Gambar 4.13: Application Nodes

22. Atur atribut Activity baru tersebut menjadi:
- Name\* : **.EditNote**
23. Simpan dan Run kembali program. Kemudian coba pilih kembali menu Add Note.

Gambar 4.14: Activity **.EditNote**

## Menambahkan Menu untuk Activity EditNote

24. Buka **string.xml** kemudian tambahkan beberapa String baru:

- String
  - Name\* : **menu\_save**
  - Value\* : **Save**
- String
  - Name\* : **menu\_revert**
  - Value\* : **Revert**
- String
  - Name\* : **menu\_delete**
  - Value\* : **Delete**

25. Buat menu baru (Caranya, baca pada point 11 dan 12) dengan konfigurasi:

- Project : **Notepad**

- File : **edit**
- Type : **Menu**
- Folder : **/res/menu**

26. Buka file **edit.xml** yang berada di bawah **res/menu/**.

27. Tambahkan tiga buah Menu Item dengan mengklik **Add....** Pilih ” *Create a new element at the top level, in Menu*” kemudian pilih **Item**. Atur atribut masing-masing menu item menjadi:

- Item
  - Id : **@+id/menuSave**
  - Title : **@string/menu\_save**
- Item
  - Id : **@+id/menuRevert**
  - Title : **@string/menu\_revert**
- Item
  - Id : **@+id/menuDelete**
  - Title : **@string/menu\_delete**

28. Kemudian buat handlernya pada Kelas **EditNote**:

```

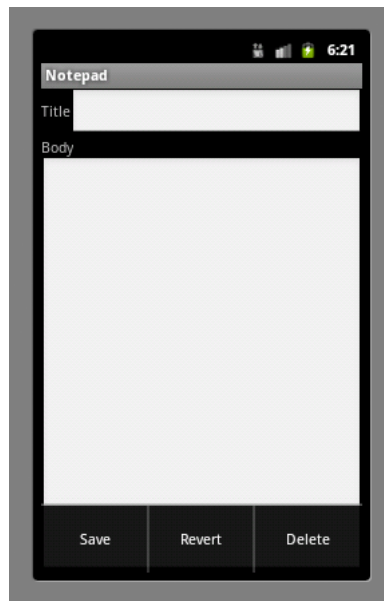
1  ...
2  import android.os.Bundle;
3  import android.view.Menu;
4  import android.view.MenuItem;
5  import android.widget.EditText;
6
7  public class EditNote extends Activity {
8      private EditText noteTitle;
9      private EditText noteBody;
10
11     @Override
12     public void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.edit_note);

```

```
15
16         noteTitle = (EditText) findViewById(R.id.editTitle);
17         noteBody = (EditText) findViewById(R.id.editBody);
18     }
19
20     @Override
21     public boolean onCreateOptionsMenu(Menu menu) {
22         getMenuInflater().inflate(R.menu.edit, menu);
23         return super.onCreateOptionsMenu(menu);
24     }
25
26     @Override
27     public boolean onOptionsItemSelected(int featureId,
28         MenuItem item) {
29         switch(item.getItemId()) {
30             case R.id.menuSave:
31                 // save note
32                 return true;
33             case R.id.menuRevert:
34                 // revert note
35                 return true;
36             case R.id.menuDelete:
37                 // delete note
38                 return true;
39         }
40         return super.onOptionsItemSelected(featureId, item);
41     }
42 }
```

29. Simpan dan Run program.



Gambar 4.15: Menu pada **EditNote**

## Storage System

---

Ada lima model penyimpanan data yang bisa kita gunakan dalam aplikasi android yakni:

- Shared preferences  
Untuk menyimpan data primitif ke dalam pasangan key-value.
- Internal Storage  
Menyimpan data dalam device memory.
- External Storage  
Menyimpan data ke External memory.
- SQLite Database  
Menyimpan data yang terstruktur ke dalam Database.
- Network Connection  
Menyimpan data ke dalam jaringan.

Untuk Aplikasi Notepad ini, kita akan mencoba menyimpan datanya ke dalam database. Untuk model penyimpanan lainnya akan kita bahas di lain waktu.

## Menyiapkan Database Adapter

30. Buat sebuah kelas baru di bawah package **latihan.Notepad** (Caranya baca pada point 9). Dengan Konfigurasi:

- Source Folder : **Notepad/src**
- Package : **latihan.Notepad**
- Name : **NoteDbAdapter**
- Modifier : **Public**
- Superclass : (kosongkan)

31. Kemudian Edit kelas **NoteDbAdapter** tersebut, dan tambahkan beberapa baris source code berikut:

```
1 package latihan.Notepad;
2
3 import android.content.ContentValues;
4 import android.content.Context;
5 import android.database.Cursor;
6 import android.database.SQLException;
7 import android.database.sqlite.SQLiteDatabase;
8 import android.database.sqlite.SQLiteOpenHelper;
9 import android.util.Log;
10
11 public class NoteDbAdapter {
12     public static final String KEY_TITLE = "title";
13     public static final String KEY_BODY = "body";
14     public static final String KEY_ROWID = "_id";
15
16     private static final String TAG = "NotesDbAdapter";
17     private DatabaseHelper mDbHelper;
18     private SQLiteDatabase mDb;
19 }
```

```
20     private static final String DATABASE_CREATE =
21         "create table notes (_id integer primary key autoincrement, "
22         + "title text not null, body text not null);";
23
24     private static final String DATABASE_NAME = "data";
25     private static final String DATABASE_TABLE = "notes";
26     private static final int DATABASE_VERSION = 2;
27
28     private final Context mContext;
29
30     private static class DatabaseHelper
31         extends SQLiteOpenHelper {
32
33         DatabaseHelper(Context context) {
34             super(context, DATABASE_NAME,
35                 null, DATABASE_VERSION);
36         }
37
38         @Override
39         public void onCreate(SQLiteDatabase db) {
40             db.execSQL(DATABASE_CREATE);
41         }
42
43         @Override
44         public void onUpgrade(SQLiteDatabase db,
45                               int oldVersion, int newVersion) {
46             Log.w(TAG, "Upgrading database from version "
47                 + oldVersion + " to " + newVersion
48                 + ", which will destroy all old data");
49             db.execSQL("DROP TABLE IF EXISTS notes");
50             onCreate(db);
51         }
52     }
53
54     public NoteDbAdapter(Context ctx) {
55         this.mContext = ctx;
56     }
57
```

```
58     public NoteDbAdapter open() throws SQLException {
59         mDbHelper = new DatabaseHelper(mCtx);
60         mDb = mDbHelper.getWritableDatabase();
61         return this;
62     }
63
64     public void close() {
65         mDbHelper.close();
66     }
67
68     public long createNote(String title, String body) {
69         ContentValues initialValues = new ContentValues();
70         initialValues.put(KEY_TITLE, title);
71         initialValues.put(KEY_BODY, body);
72
73         return mDb.insert(DATABASE_TABLE, null,
74             initialValues);
75     }
76
77     public boolean deleteNote(long rowId) {
78         return mDb.delete(DATABASE_TABLE, KEY_ROWID
79             + "=" + rowId, null) > 0;
80     }
81
82     public Cursor fetchAllNotes() {
83         return mDb.query(DATABASE_TABLE, new String[]
84             {KEY_ROWID, KEY_TITLE, KEY_BODY},
85             null, null, null, null, null);
86     }
87
88     public Cursor fetchNote(long rowId) throws SQLException {
89         Cursor mCursor =
90             mDb.query(true, DATABASE_TABLE, new String[]
91                 {KEY_ROWID, KEY_TITLE, KEY_BODY},
92                 KEY_ROWID + "=" + rowId, null,
93                 null, null, null, null);
94         if (mCursor != null) {
95             mCursor.moveToFirst();
```

```

96         }
97         return mCursor;
98     }
99 }
100
101 public boolean updateNote(long rowId, String title,
102                           String body) {
103     ContentValues args = new ContentValues();
104     args.put(KEY_TITLE, title);
105     args.put(KEY_BODY, body);
106
107     return mDb.update(DATABASE_TABLE, args, KEY_ROWID
108                      + "=" + rowId, null) > 0;
109 }
110 }

```

32. Dalam kelas tersebut telah ada methodode untuk menyimpan note ke dalam database yakni **createNote**, menghapus note yakni **deleteNote**, membaca note **fetchNote** dan meng-*update* yakni **updateNote**.

## Menyimpan, Membaca dan Menghapus Note

---

Nah pada langkah di atas, kita sudah membuat sebuah kelas untuk membantu proses CRUD (Create, Read, Update dan Delete) data note yang ada dalam database. Selanjutnya, kita perlu source code ketika user memilih menu **Save** pada Activity **EditNote**.

### Menyimpan Note

Untuk menyimpan note, kita perlu menambahkan proses penyimpanan pada handler untuk menu save. Caranya:

33. Buka file EditNote.java, kemudian tambahkan sebuah methodode **saveNote()** pada kelas tersebut.

```

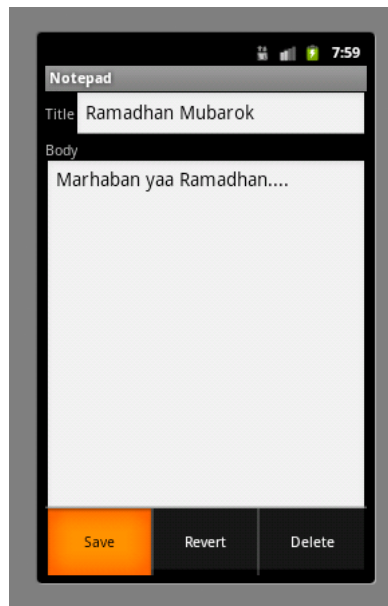
1  ...
2  public class EditNote extends Activity {

```

```
3     private EditText noteTitle;
4     private EditText noteBody;
5     private Long mRowId;
6     private NoteDbAdapter mDbHelper;
7
8     @Override
9     public void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11
12         setContentView(R.layout.edit_note);
13         noteTitle = (EditText) findViewById(R.id.editTitle);
14         noteBody = (EditText) findViewById(R.id.editBody);
15
16         mDbHelper = new NoteDbAdapter(this);
17         mDbHelper.open();
18
19         mRowId = (savedInstanceState == null) ? null :
20             (Long) savedInstanceState.getSerializable(
21                 NoteDbAdapter.KEY_ROWID);
22         if (mRowId == null) {
23             Bundle extras = getIntent().getExtras();
24             mRowId = extras != null ?
25                 extras.getLong(NoteDbAdapter.KEY_ROWID)
26                 : null;
27         }
28     }
29
30     ...
31
32     @Override
33     public boolean onOptionsItemSelected(int featureId,
34         MenuItem item) {
35         switch(item.getItemId()) {
36             case R.id.menuSave:
37                 // save note
38                 saveNote();
39                 return true;
40             case R.id.menuRevert:
```

```
41         // revert note
42         return true;
43     case R.id.menuDelete:
44         // delete note
45         return true;
46     }
47     return super.onMenuItemSelected(featureId, item);
48 }
49
50 public void saveNote() {
51     String title = noteTitle.getText().toString();
52     String body = noteBody.getText().toString();
53
54     if (mRowId == null) {
55         long id = mDbHelper.createNote(title, body);
56         if (id > 0) {
57             mRowId = id;
58         }
59     } else {
60         mDbHelper.updateNote(mRowId, title, body);
61     }
62 }
63
64 }
```

34. Simpan dan coba Run program.

Gambar 4.16: Menu pada **EditNote**

Catatan: Ketika menu **Save** dipilih, masih belum terlihat efeknya karena kita masih belum membuat metode untuk menampilkannya pada tampilan utama aplikasi Notepad (pada layout **main.xml**).

## Membaca Note

Untuk mbaca note, kita perlu menambahkan beberapa proses untuk membaca note. Yakni pada kelas Notepad untuk menampilkan list Note yang telah ada, dan pada kelas EditNote untuk mengedit note yang telah ada.

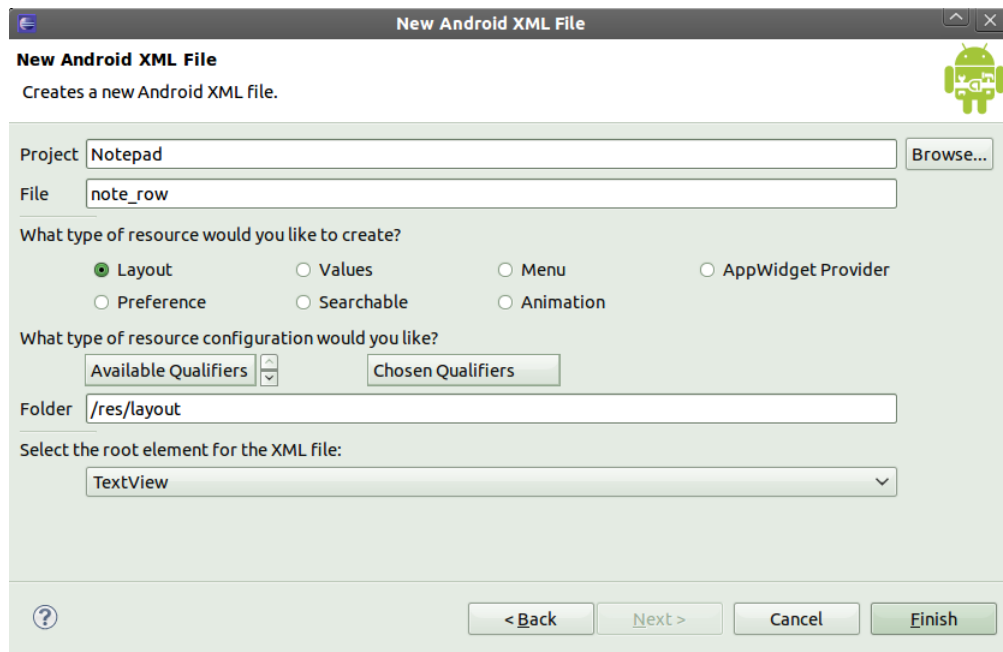
## Menampilkan List Note

35. Buat sebuah layout baru, caranya lihat pada point **5**.
36. Pada jendela **New Android XML File**, buat file XML dengan konfigurasi:
  - Project : **Notead**
  - File : **note\_row**



- Type : **Layout**
- Folder : **/res/layout**
- root element : **TextView**

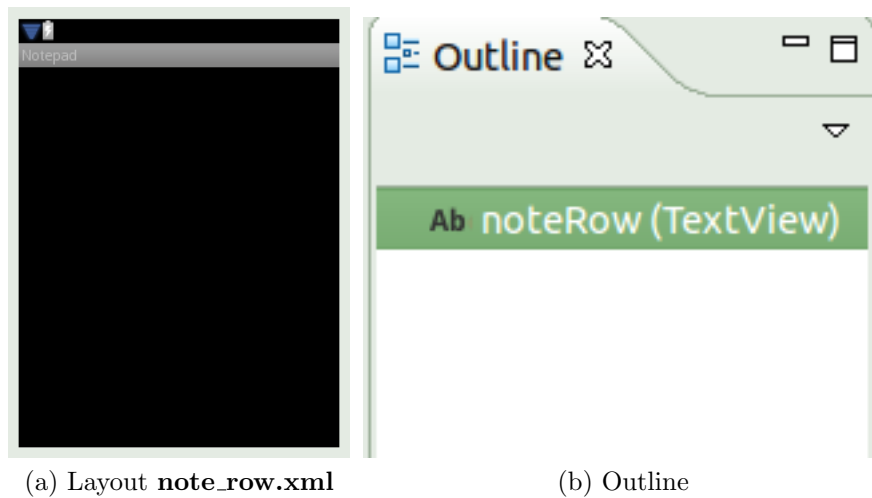
Pastikan Root element-nya menggunakan TextView. Kemudian klik **Finish**



Gambar 4.17: Layout **note\_row** dengan TextView sebagai Root element-nya.

37. Buka file **note\_row.xml** yang ada di bawah **res/layout**.
38. Ubah ID **TextView** yang ada pada layout, dengan cara mengklik kanan **TextView** pada Outline kemudian pilih **Edit ID**. Kemudian ganti menjadi:
  - ID : **noteRow**

Kemudian simpan perubahan.

Gambar 4.18: Layout **note\_row.xml** dan Outline-nya

### Menambahkan Program

39. Buka file **Notepad.java**. Kemudian tambahkan import class:

```
package latihan.Notepad;

//import android.app.Activity;
import android.app.ListActivity;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.ContextMenu;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ContextMenu.ContextMenuInfo;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.AdapterView.AdapterContextMenuInfo;

public class Notepad extends Activity {
```

```

    /** Called when the activity is first created. */
    ...

```

40. Ganti superclass dari Notepad yang tadinya dari kelas **Activity** menjadi **ListActivity**

```

package latihan.Notepad;
...
import android.widget.AdapterView.AdapterContextMenuInfo;

public class Notepad extends ListActivity {
    /** Called when the activity is first created. */
    private static final int ACTIVITY_CREATE=0;
    private static final int ACTIVITY_EDIT=1;
    ...

```

41. Tambahkan objek **mDbHelper** untuk koneksi ke databse dan **DELETE\_ID** untuk membuat Contxt menu:

```

package latihan.Notepad;
...
public class Notepad extends ListActivity {
    /** Called when the activity is first created. */
    private static final int ACTIVITY_CREATE=0;
    private static final int ACTIVITY_EDIT=1;
    private NoteDbAdapter mDbHelper;
    private static final int DELETE_ID = Menu.FIRST + 1;
    ...

```

42. Buka koneksi ke database dengan memanggil methode open() dari objek mDbHelper pada methode **onCreate()**:

```

package latihan.Notepad;
...
public class Notepad extends ListActivity {
    ...
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

        setContentView(R.layout.main);
        mDbHelper = new NoteDbAdapter(this);
        mDbHelper.open();

        fillData();
        registerForContextMenu(getListView());
    }
    ...

```

43. Buat method **fillData()** untuk mengisi listView yang ada pada layout **main.xml**:

```

package latihan.Notepad;
...
public class Notepad extends ListActivity {
...
    private void fillData() {
        Cursor notesCursor = mDbHelper.fetchAllNotes();
        startManagingCursor(notesCursor);

        String[] from = new String[] {NoteDbAdapter.KEY_TITLE};

        int[] to = new int[] {R.id.noteRow};

        SimpleCursorAdapter notes =
            new SimpleCursorAdapter(this,
                R.layout.note_row, notesCursor, from, to);
        setListAdapter(notes);
    }
    ...

```

**Catatan:** Di sini kita memanggil layout **note\_row.xml**, dimana TextView **noteRow** akan diisi dengan string berupa title dari note yang ada di dalam database.

44. Tambahkan method **onListItemClick()** untuk handle event ketika list di klik:

```

package latihan.Notepad;
...

```

```

public class Notepad extends ListActivity {
    ...
    @Override
    protected void onItemClick(ListView l, View v,
                               int position, long id) {
        super.onItemClick(l, v, position, id);
        Intent i = new Intent(this, EditNote.class);
        i.putExtra(NoteDbAdapter.KEY_ROWID, id);
        startActivityForResult(i, ACTIVITY_EDIT);
    }
    ...

```

Methode ini digunakan untuk menghandle event klik pada listView. Dimana ketika salah satu item dalam list di klik, maka ia akan menjalankan activity **EditNote** dalam mode Edit (**ACTIVITY\_EDIT**).

45. Tambahkan methode untuk membuat context menu pada List, serta handler menunya:

```

1 package latihan.Notepad;
2 ...
3
4 public class Notepad extends ListActivity {
5     ...
6     @Override
7     public void onCreateContextMenu(ContextMenu menu, View v,
8                                     ContextMenuInfo menuInfo) {
9         super.onCreateContextMenu(menu, v, menuInfo);
10        menu.add(0, DELETE_ID, 0, R.string.menu_delete);
11    }
12
13    @Override
14    public boolean onContextItemSelected(MenuItem item) {
15        switch(item.getItemId()) {
16            case DELETE_ID:
17                AdapterContextMenuInfo info =
18                    (AdapterContextMenuInfo) item.getMenuInfo();
19                mDbHelper.deleteNote(info.id);
20                fillData();

```

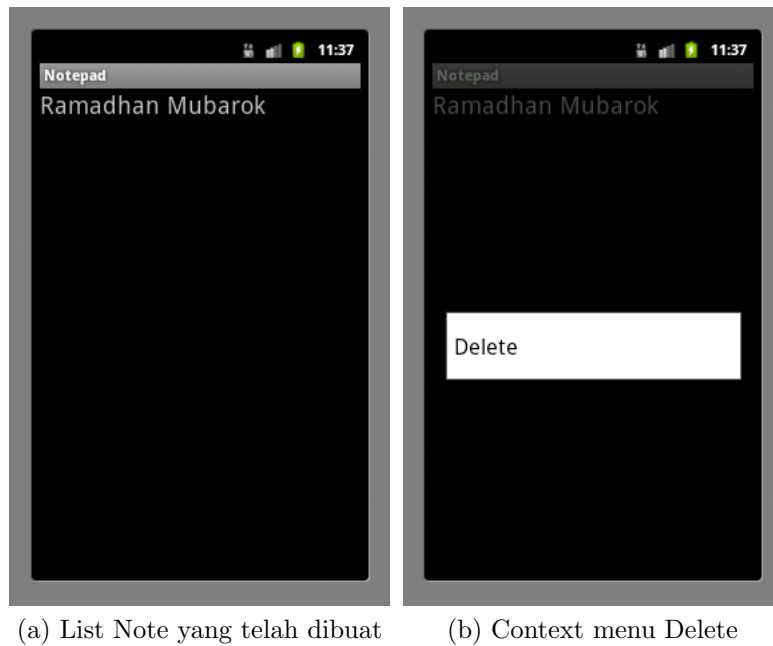
```
21         return true;
22     }
23     return super.onContextItemSelected(item);
24 }
25 ...
```

46. Dan tambahkan sebuah method untuk handle ketika sebuah activity selesai, maka List note yang ada di update kembali dengan memanggil method `fillData()`:

```
1 package latihan.Notepad;
2
3 ...
4 public class Notepad extends ListActivity {
5     ...
6     @Override
7     protected void onActivityResult
8         (int requestCode, int resultCode, Intent intent) {
9         super.onActivityResult(requestCode, resultCode, intent);
10        fillData();
11    }
12    ...
```

Listing full program dapat dilihat pada halaman terakhir.

47. Simpan dan Run program. Jika tidak ada kesalahan maka program akan berjalan seperti:



Gambar 4.19: Hasil Run Program

Akan tetapi jika anda mengklik note, ketika masuk ke activity **Edit-Note** note masih belum terbaca. Karena itu kita perlu menambahkan juga beberapa baris program pada kelas EditNote untuk membaca note dari database.

### Membaca Note untuk mode Editing

48. Buka file **EditNote.java**. Kemudian tambahkan import class:

```
1 | package latihan.Notepad;  
2 |  
3 | import android.app.Activity;  
4 | import android.database.Cursor;  
5 | import android.os.Bundle;  
6 | ...
```

49. Selanjutnya kita perlu menambahkan methode untuk me-*retrive* data (Title dan Body) Note dari database:

```
1 package latihan.Notepad;
2 ...
3
4 public class EditNote extends Activity {
5     ...
6
7     @Override
8     public void onCreate(Bundle savedInstanceState) {
9         ...
10    }
11
12    private void populateFields() {
13        if (mRowId != null) {
14            Cursor note = mDbHelper.fetchNote(mRowId);
15            startManagingCursor(note);
16            noteTitle.setText(note.getString(
17                note.getColumnIndexOrThrow(NoteDbAdapter.KEY_TITLE)));
18            noteBody.setText(note.getString(
19                note.getColumnIndexOrThrow(NoteDbAdapter.KEY_BODY)));
20        }
21    }
22    ...
```

50. Selanjutnya kita perlu memanggil metode **populateFields()** tersebut ketika activity **EditNote** dijalankan, yakni pada metode **onCreate()**:

```
1 package latihan.Notepad;
2 ...
3
4 public class EditNote extends Activity {
5     ...
6
7     @Override
8     public void onCreate(Bundle savedInstanceState) {
9         ...
10        mDbHelper = new NoteDbAdapter(this);
11        mDbHelper.open();
```

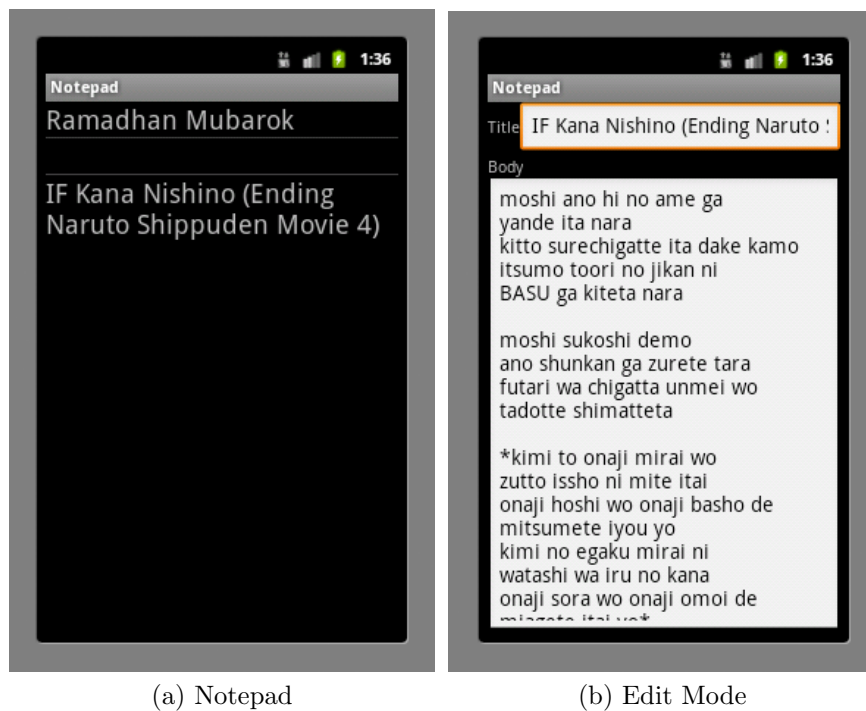


```

12
13         mRowId = (savedInstanceState == null) ? null :
14             (Long) savedInstanceState.getSerializable(
15                 NoteDbAdapter.KEY_ROWID);
16         if (mRowId == null) {
17             Bundle extras = getIntent().getExtras();
18             mRowId = extras != null ?
19                 extras.getLong(NoteDbAdapter.KEY_ROWID)
20                 : null;
21         }
22         populateFields();
23     }
24
25     private void populateFields() {
26     ...

```

51. Simpan kemudian Run program.



Gambar 4.20: Hasil Run Program

## Latihan

---

Pada program di atas, Kita masih belum membuat handler untuk menu **Revert** dan **Delete** pada menu **EditNote**. Sebagai latihan mari kita coba tambahkan handler-handler ini:

1. Buka dan edit file **EditNote.java**.
2. tambahkan dua buah method baru yakni method **revertNote()** dan **deleteNote()**:

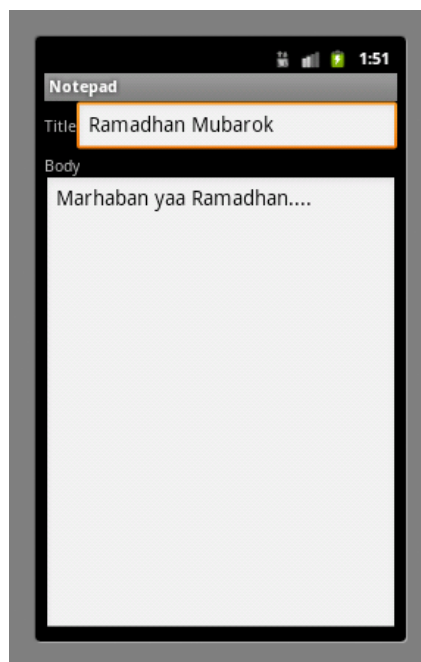
```
1 private void revertNote() {
2     AlertDialog.Builder builder =
3         new AlertDialog.Builder(this);
4     builder.setTitle("Revert");
5     builder.setMessage("Are you sure?")
6     .setPositiveButton("OK",
7         new DialogInterface.OnClickListener() {
8             @Override
9             public void onClick(DialogInterface dialog,
10                 int which) {
11                 // TODO Auto-generated method stub
12                 populateFields();
13                 dialog.dismiss();
14             }
15         });
16     AlertDialog dialogHasil = builder.create();
17     dialogHasil.show();
18 }
```

Dan method **deleteNote()**:

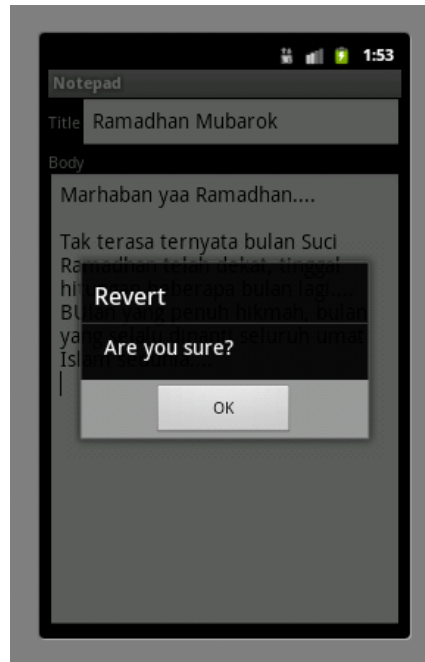
```
1 private void deleteNote() {
2     AlertDialog.Builder builder =
3         new AlertDialog.Builder(this);
4     builder.setTitle("Delete Note");
5     builder.setMessage("Are you sure?")
6     .setPositiveButton("OK",
7         new DialogInterface.OnClickListener() {
```

```
8      @Override
9      public void onClick(DialogInterface dialog,
10         int which) {
11         // TODO Auto-generated method stub
12         mDbHelper.deleteNote(mRowId);
13         dialog.dismiss();
14         finish();
15     }
16     });
17     AlertDialog dialogHasil = builder.create();
18     dialogHasil.show();
19 }
```

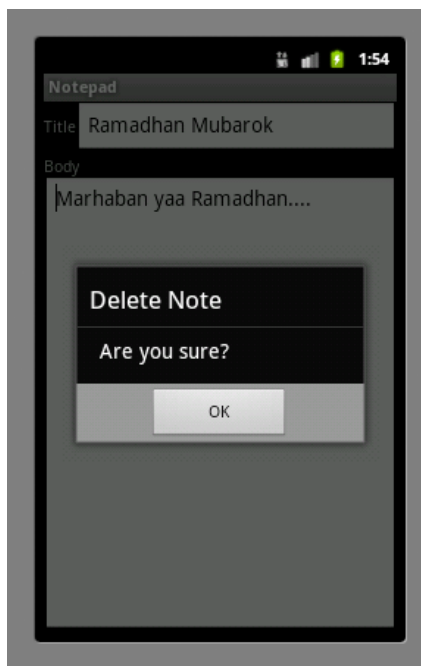
3. Kemudian panggil metode-metode tersebut melalui handler menu yang telah dibuat pada metode **onMenuItemSelected()**.
4. Simpan dan coba Run kembali program.



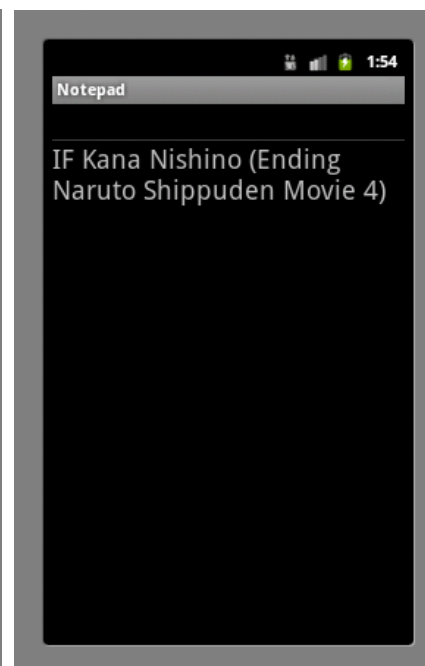
(a) Note awal



(b) Revert hasil perubahan



(c) Delete Note



(d) List Note setelah dihapus

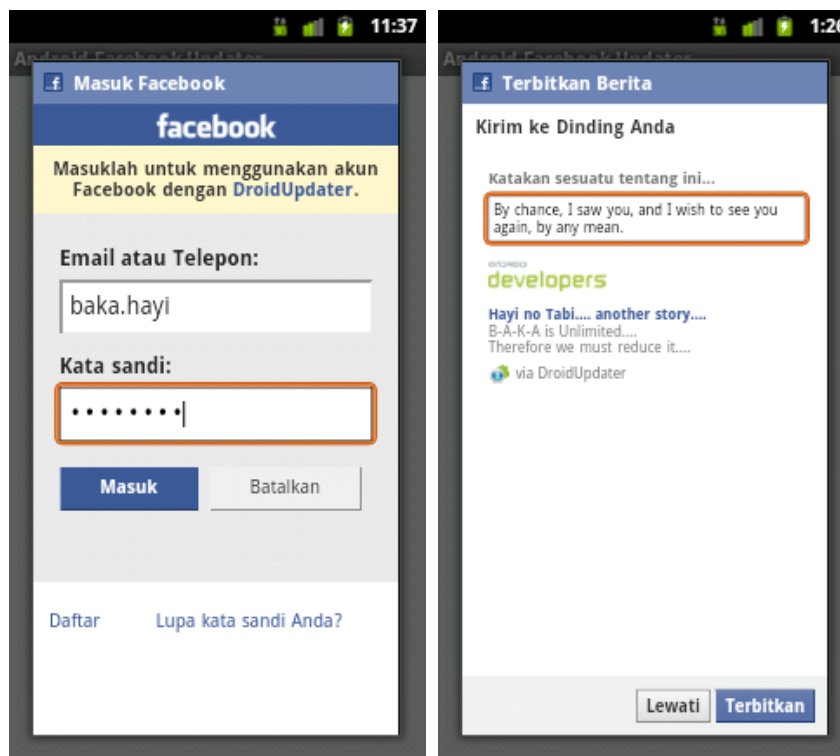
Gambar 4.21: Hasil Akhir Run Program

MODUL

5

## Update Facebook

Target:



(a) Halaman login

(b) Update Status

Gambar 5.1: Target Akhir

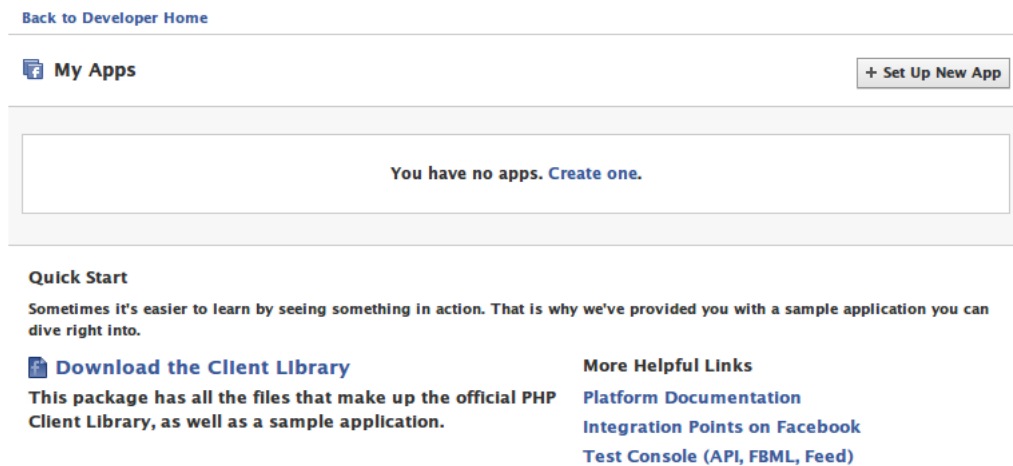
## Langkah-langkah

---

### Membuat Facebook Apps

Sebelum bisa membuat aplikasi untuk mengakses Facebook, pertama-tama kita perlu mendaftarkan aplikasi kita terlebih dahulu di Facebook untuk mendapatkan **Apps ID**, **API key** dan **Apps Secret**. Untuk melakukan registrasi sebuah aplikasi di Facebook langkah-langkahnya yakni:

1. Buka alamat: <http://www.facebook.com/developers/apps.php>



Gambar 5.2: Creating Facebook Apps.

2. Kemudian pilih **+Set Up New Application**

facebook [Search](#) [Home](#) [Profile](#)

[Create Application](#) [Back to My Apps](#)

**Essential Information**

App Name  Cannot contain Facebook trademarks or have a name that can be confused with an app built by Facebook.

Terms Do you agree to the Facebook Terms?



☐ Agree ☒ Disagree

[Create App](#)

Facebook © 2011 · [English \(US\)](#) [About](#) · [Advertising](#) · [Create a Page](#) · [Developers](#) · [Careers](#) · [Privacy](#)



Gambar 5.3: Creating Facebook Apps.

3. Isi **Apps Name** dengan nama aplikasi yang akan Anda buat. Misalnya **DoroidUpdater** (Facebook tidak memperkenalkan nama aplikasi yang mengandung trademark Facebook). Kemudian Pilih **Agree** dan klik **Create App**.
4. Selanjutnya pada halaman berikutnya Anda akan dihadapkan dengan captcha.
5. Setelah itu Anda akan masuk ke halaman edit Apps. Pada halaman ini anda dapat mengganti nama aplikasi, menambahkan Icon dan Logo (untuk ditampilkan di halaman Facebook) serta pengaturan lainnya.

| Basic Information    |                                                                                                                     |                                                                                 |
|----------------------|---------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| Application Name     | <input type="text" value="DroidUpdater"/>                                                                           | Cannot contain Facebook trademarks                                              |
| Description          | <input type="text" value="Update Status facebook lewat Android"/>                                                   | The plaintext description of your app                                           |
| Icon                 |  <a href="#">Change your icon</a>  | Appears next to your app name throughout Facebook (16x16)                       |
| Logo                 |  <a href="#">Change your logo</a>  | Appears in authorization dialogs, search results, and the app directory (75x75) |
| Language             | <input type="text" value="English (US)"/>                                                                           | The native language of your app.                                                |
| User Support Address | <input checked="" type="radio"/> Email <input type="radio"/> URL<br><input type="text" value="hayi.nkm@gmail.com"/> | The email address or URL where users can contact you about your app             |

Gambar 5.4: Edit Apps.

6. Kemudian simpan perubahan yang telah anda lakukan.
7. Pada halaman berikutnya Anda akan mendapatkan informasi dari aplikasi Anda, berupa **App ID**, **API Key**, **App Secret** dan Lainnya.

## DroidUpdater

**Directory Status: Not Submitted**

Once you have completed your app, you may [submit it](#) to the App Directory.

---

| Monthly Active Users | People Who Like This | Total Users |
|----------------------|----------------------|-------------|
| 0                    | 0                    | 0           |

---

|                                  |                                          |
|----------------------------------|------------------------------------------|
| App ID                           | <a href="#">Edit Settings</a>            |
| 147478081988384                  | <a href="#">Application Profile Page</a> |
| API Key                          | <a href="#">Insights</a>                 |
| android:apiKey="147478081988384" | <a href="#">Translations</a>             |
| App Secret                       | <a href="#">Advertise</a>                |
| 867ee8f9b550a529b5153ce45f195da  | <a href="#">Reset App Secret</a>         |

---

Contact Email

**hayi.nkm@gmail.com**

Gambar 5.5: Apps Information

8. Simpan **App ID**, **API Key** dan **App Secret** Anda. Karena nanti



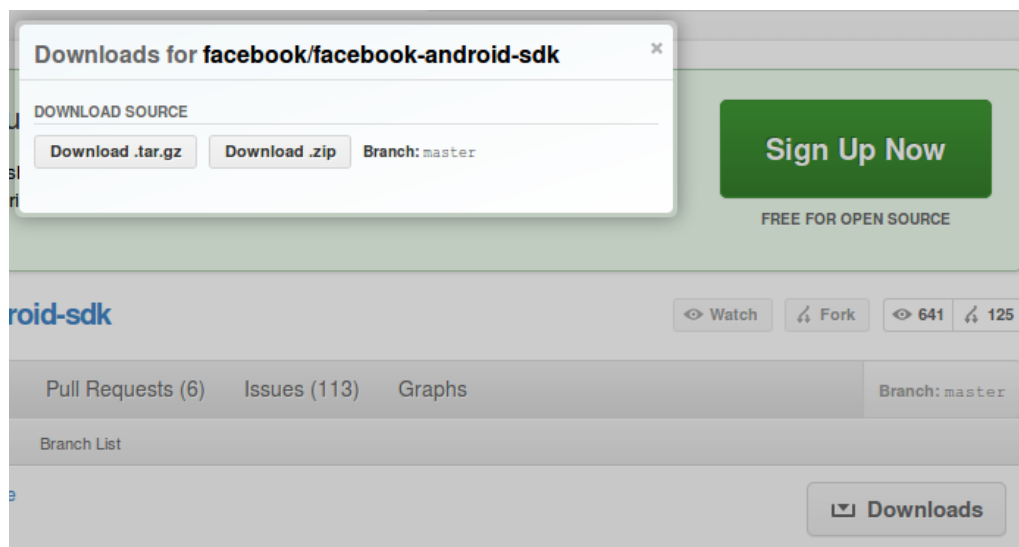
akan digunakan saat melakukan request untuk koneksi dengan Facebook.

## Facebook Android SDK

Selanjutnya kita membutuhkan library Facebook untuk dapat menggunakan Kelas-kelas untuk melakukan koneksi ke Facebook. Untuk mendapatkannya buka halaman:

<https://github.com/facebook/facebook-android-sdk>

Kemudian download SDK-nya.



Gambar 5.6: Facebook Android SDK

Selanjutnya ekstrak SDK yang telah di download di tempat yang aman.

### Mengimport SDK sebagai Library

Setelah berhasil di download dan di ekstrak. Kita perlu mengimport SDK tersebut ke Eclipse untuk dijadikan sebuah Project library. Caranya:

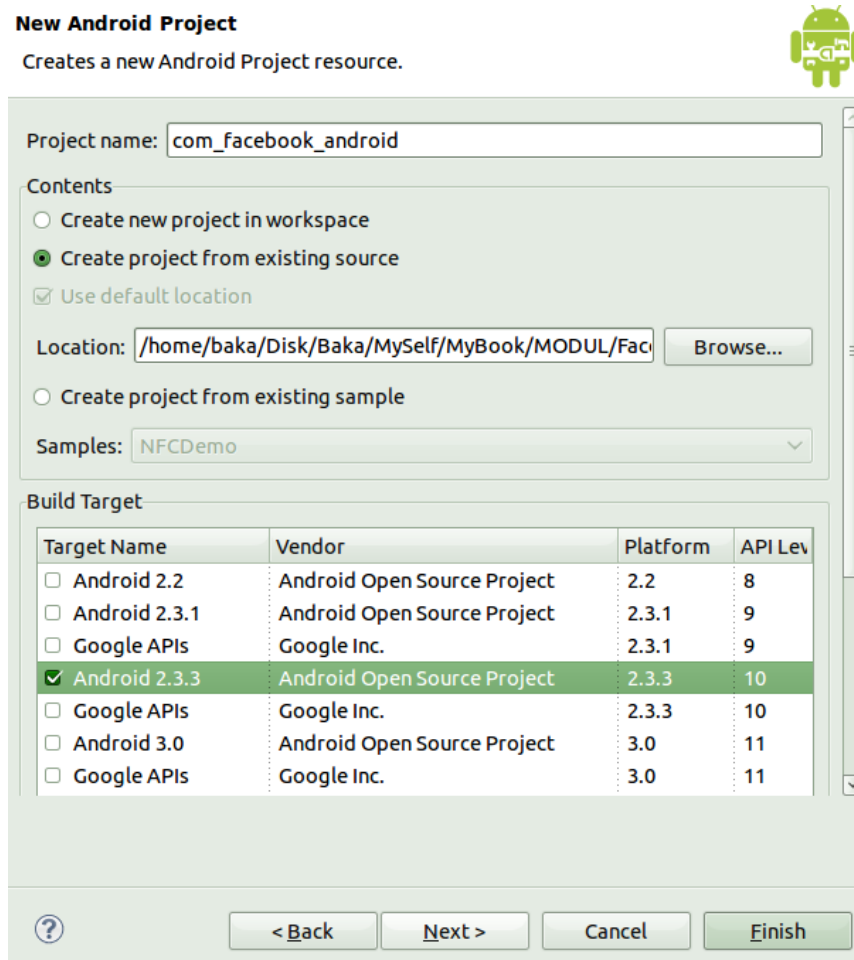
1. Dari menu **File >New >Project...** pilih Android Project.
2. Pada jendela **New Android Project** pilih **Create project from existing sources**.

3. Pada **Location**, klik **Browse...** dan cari direktory dimana anda meng-extrak facebook SDK yang telah anda download, kemudian pilih sub direktory **facebook**.

**Catatan :**

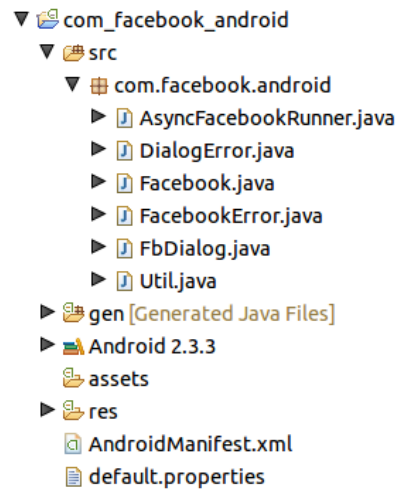
Pada direktory SDK facebook tersebut terdapat tiga buah direktory, yakni **example**, **facebook** dan **test**. Direktory example berisi contoh-contoh penggunaan SDK. Direktory facebook berisi library SDK. Dan direktory test berisi test project.

4. Pada **Build target** pilih Android 2.3.3 (atau sesuaikan).



Gambar 5.7: Import Facebook Android SDK ke Eclipse

5. Jika berhasil, maka Anda akan mendapatkan sebuah project baru bernama **com\_facebook\_android** pada Package Explorer Anda.



Gambar 5.8: Project **com\_facebook\_android**

## Membuat Aplikasi

---

Setelah semua persiapan untuk pembuatan aplikasi sudah selesai, kita sudah siap untuk membuat Aplikasi Android yang terintegrasi dengan Facebook.

### Membuat Project

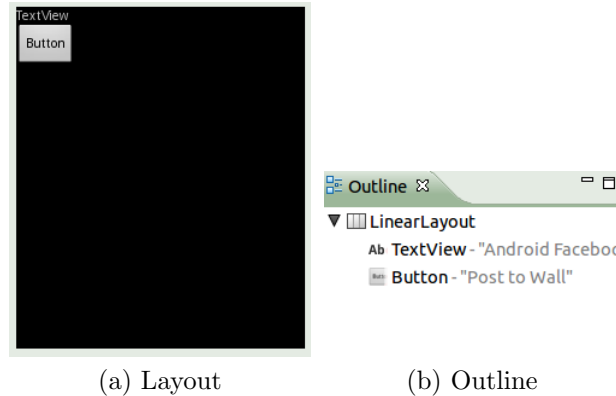
1. Buat sebuah project baru, dengan property:
  - Project name : **DroidUpdater**
  - Build Target : **Android 2.3.3** (sesuaikan)
  - Application Name : **Facebook Updater**
  - Package name : **latihan.droidupdater**
  - Create Activity : **DroidUpdater**
  - Min SDK : **10** (sesuaikan dengan Build target)

## Membuat UI

2. Tambahkan beberapa value pada **string.xml**:

- String
  - Name : **post\_wall**
  - Value : Post to Wall
- String
  - Name : **postToWall**
  - Value : postToWall
- Color
  - Name : **biru**
  - Value : #FF0000FF
- String
  - Name : **abu**
  - Value : #FFEEEEEE

3. Ubah layout **main.xml** menjadi:



Gambar 5.9: Layout **main.xml**

4. Kemudian ubah property masing-masing objek menjadi:

- LinearLayout
  - Orientation : **Vertical**

- Layout width : **Fill parent**
- Layout height : **Fill parent**
- Background : **@color/abu**
- Text View
  - Layout Width : **Fill parent**
  - Layout Height : **90dip**
  - Text Color : **@color/biru**
  - Text Size : **20sp**
  - Text Style : **bold**
  - Gravity : **center**
  - Layout Gravity : **center**
  - Text : **@string/app\_name**
- Button
  - Layout Width : **Wrap Content**
  - Layout Height : **Wrap Content**
  - Text Style : **bold**
  - Gravity : **center**
  - Layout Gravity : **center**
  - On Click : **@string/postToWall**
  - Text : **@string/post\_wall**

Sehingga layout akan terlihat seperti:



Gambar 5.10: Project **com\_facebook\_android**

## Membuat Koneksi

5. Buka file **AndroidManifest.xml** kemudian masuk ke Tab Permission. Pilih Add untuk menambahkan Permission tambahkan:

- **Uses Permission**

- Name : **android.permission.INTERNET**

Hint: Cukup pilih permission yang ada pada Combo Box.

6. Buka file **DroidUpdater.java** untuk mengubah activity **DroidUpdater**.
7. Tambahkan beberapa import library:

```
1 import com.facebook.android.*;
2 import com.facebook.android.Facebook.*;
3 import com.facebook.android.FacebookError;
4 import android.app.Activity;
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.view.View;
8 ...
```

8. Kemudian tambahkan property pada kelas **DroidUpdater**:

```
1 ...
2 public class DroidUpdater extends Activity {
3     /** Called when the activity is first created. */
4     private static final Facebook fb =
5         new Facebook("147478081988384");
6
7     @Override
8     ...
```

**Penting:** Ganti "147478081988384" dengan **App ID** Anda yang telah di buat di atas.

9. Pada methode **onCreate()** tambahkan beberapa baris berikut:

```

1  ...
2  public class DroidUpdater extends Activity {
3      /** Called when the activity is first created. */
4      private static final Facebook fb =
5          new Facebook("147478081988384");
6
7      @Override
8      public void onCreate(Bundle savedInstanceState) {
9          super.onCreate(savedInstanceState);
10         setContentView(R.layout.main);
11
12         fb.authorize(this ,new DialogListener() {
13             @Override
14             public void onFacebookError(FacebookError e) {
15             }
16
17             @Override
18             public void onError(DialogError e) {
19             }
20
21             @Override
22             public void onComplete(Bundle values) {
23             }
24
25             @Override
26             public void onCancel() {
27             }
28         });
29     }
30     ...

```

#### 10. Tambahkan methode **onActivityResult**:

```

1  ...
2  public class DroidUpdater extends Activity {
3      /** Called when the activity is first created. */
4      private static final Facebook fb =
5          new Facebook("147478081988384");

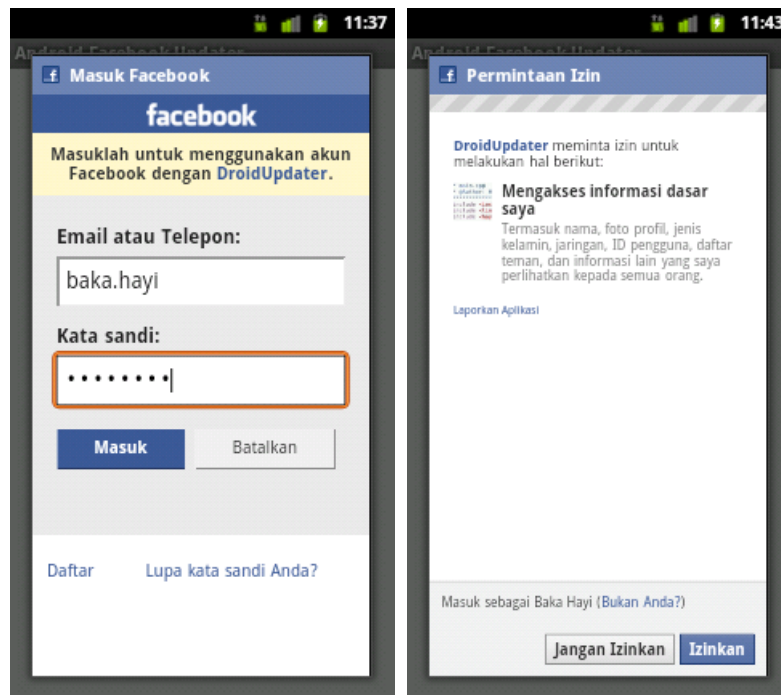
```

```

6
7     @Override
8     public void onCreate(Bundle savedInstanceState) {
9         ...
10    }
11
12    @Override
13    public void onActivityResult(
14        int requestCode, int resultCode, Intent data) {
15        super.onActivityResult(requestCode, resultCode, data);
16
17        fb.authorizeCallback(requestCode, resultCode, data);
18    }
19 }

```

11. Simpan dan coba Run program.



(a) Halaman login

(b) Authentikasi

Gambar 5.11: Hasil Run



Akan tetapi, program masih belum bisa digunakan untuk mengupdate status karena kita belum membuat metode untuk handle event **onClick** pada Button.

## Post to Wall

Untuk bisa melakukan posting ke Wall, Aplikasi kita harus memiliki akses untuk melakukan posting ke Wall. Pada hasil Run di atas, hak akses yang dimiliki hanya hak untuk mengakses informasi dasar saja. Kita perlu menambahkan akses untuk bisa melakukan post ke Wall. Untuk menambahkannya, kita perlu mengubah mode autentikasi ketika memanggil metode **authorize()**. Caranya:

12. Tambahkan sebuah property pada kelas **DroidUpdater**:

```
1  ...
2  public class DroidUpdater extends Activity {
3      /** Called when the activity is first created. */
4      private static final Facebook fb =
5          new Facebook("147478081988384");
6      private static final String[] POST_AUTH = new String[] {
7          "publish_stream", "friends_status"
8      };
9
10     @Override
11     public void onCreate(Bundle savedInstanceState) {
12         ...
```

13. Ubah mode autentikasi saat pemanggilan metode **authorize()** dari objek **fb**:

```
1  ...
2  public class DroidUpdater extends Activity {
3      /** Called when the activity is first created. */
4      private static final Facebook fb =
5          new Facebook("147478081988384");
6      private static final String[] POST_AUTH = new String[] {
7          "publish_stream", "friends_status"
8      };
```

```

9
10     @Override
11     public void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.main);
14
15         fb.authorize(this, POST_AUTH ,new DialogListener() {
16             @Override
17             public void onFacebookError(FacebookError e) {
18             }
19
20             @Override
21             public void onError(DialogError e) {
22             }
23
24             @Override
25             public void onComplete(Bundle values) {
26             }
27
28             @Override
29             public void onCancel() {
30             }
31         });
32     }
33     ...

```

14. Tambahkan methode **postToWall()** untuk mempost ke Wall:

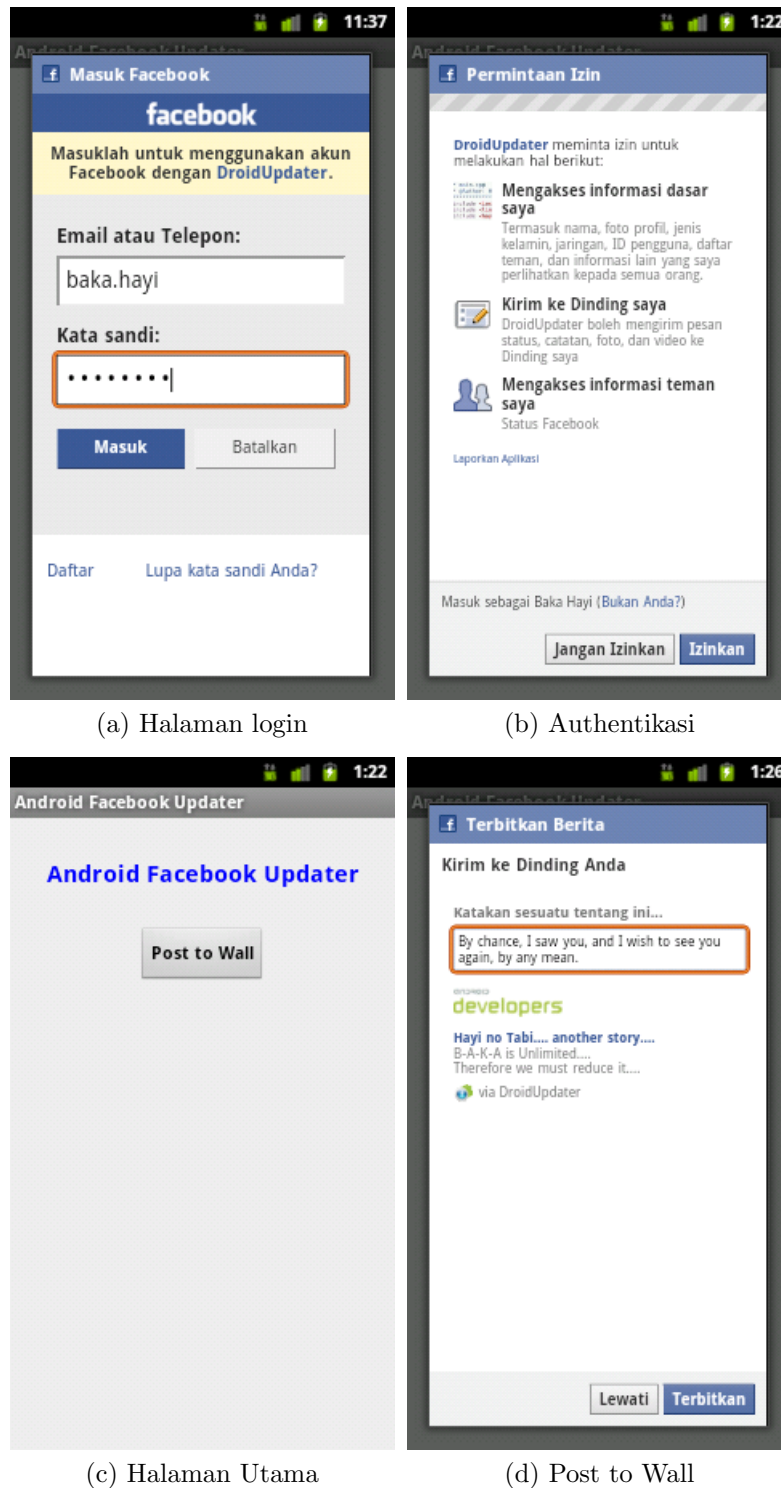
```

1     ...
2     public class DroidUpdater extends Activity {
3         ...
4         @Override
5         public void onCreate(Bundle savedInstanceState) {
6             ...
7         }
8
9         public void postToWall(View view) {
10             Bundle param = new Bundle();

```

```
11         param.putString("link", "http://kumachii.wordpress.com");
12         param.putString("picture",
13             "http://developer.android.com/assets/images/bg_logo.png");
14         param.putString("caption", "B-A-K-A is Unlimited....");
15         param.putString("description",
16             "Therefore we must reduce it....");
17
18         fb.dialog(this, "feed", param, new DialogListener() {
19
20             @Override
21             public void onFacebookError(FacebookError e) {
22             }
23
24             @Override
25             public void onError(DialogError e) {
26             }
27
28             @Override
29             public void onComplete(Bundle values) {
30             }
31
32             @Override
33             public void onCancel() {
34             }
35         });
36     }
37     ...
```

15. Kemudian simpan dan Run Program.



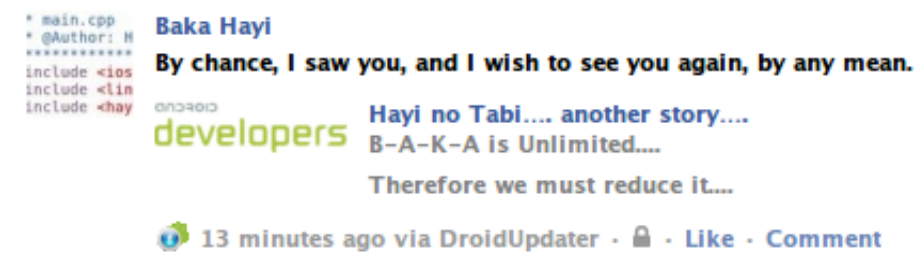
(a) Halaman login

(b) Authentikasi

(c) Halaman Utama

(d) Post to Wall

Gambar 5.12: Hasil Run



Gambar 5.13: Status yang terupdate menggunakan **DroidUpdate**.



---

## Daftar Referensi

- [1] Google Inc., *<http://developer.android.com/guide/index.html>*, Android Developers Guide, 2011.
- [2] Facebook., *<http://developers.facebook.com/docs/guides/mobile/>*, Mobile Apps - Facebook Developers, 2011.