

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/338819838>

# Pemrograman Aplikasi Android

Book · May 2012

---

CITATIONS

6

READS

178

2 authors:



Dodit Suprianto  
Politeknik Negeri Malang

29 PUBLICATIONS 49 CITATIONS

[SEE PROFILE](#)



Rini Agustina  
PGRI Kanjuruhan University of Malang

41 PUBLICATIONS 49 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Digital Leadership [View project](#)



Internet of Things [View project](#)

Penerjemah & Penulis: Rizki Mulyana

# PEMROGRAMAN APLIKASI android

Step by Step Membangun Aplikasi Android  
untuk Smartphone dan Tablet



Buku Kita  
Plus CD



Software Pengembang  
Grafik & Program

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/325102315>

# Pemrograman Aplikasi Android

Book · May 2012

---

CITATIONS

0

READS

4,023

2 authors:



Dodit Suprianto  
Politeknik Negeri Malang

17 PUBLICATIONS 8 CITATIONS

[SEE PROFILE](#)



R. Agustina  
Kanjuruhan University of Malang

19 PUBLICATIONS 3 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Machine learning and IOT [View project](#)

**PEMROGRAMAN APLIKASI  
SMARTPHONE & TABLET  
BERBASIS ANDROID UNTUK  
PEMULA**

# KATA PENGANTAR

Syukur Alhamdulillah kami panjatkan kehadiran AllAh SWT, atas berkat rahmat dan kemudahan yang diberikan hingga buku yang berjudul **“Pemrograman Aplikasi Smartphone & Tablet Berbasis Android Untuk Pemula”** ini dapat terselesaikan.

Buku ini ditulis atas dasar meningkatnya pasar perangkat bergerak berbasis Android smartphone maupun tablet yang semakin besar. Begitu pula bagi pengembang yang dituntut untuk turut berpartisipasi mengembangkan software aplikasi bergerak tersebut baik untuk kebutuhan pribadi maupun industri.

Isi dari buku ini tentu jauh dari sempurna dan belum lengkap, karena isinya masih berupa penjelasan-penjelasan dasar bagaimana memprogram aplikasi dalam lingkungan system operasi Android. Diharapkan pembaca terus mengembangkannya lagi menjadi aplikasi yang lebih sempurna, menyesuaikan dengan permasalahan yang sedang dihadapi dengan cara membaca literatur-literatur lain yang banyak tersebar di internet.

Akhir kata semoga buku ini bisa membantu pembaca dalam mempelajari pemrograman aplikasi Android dan bermanfaat bagi kita semua. Amin

Malang, Juli 2012

Penulis

# **DAFTAR ISI**

## **BAB 1 MENGENAL ANDROID**

- 1.1 Apa Android Itu?
- 1.2 Versi Android
- 1.3 Fitur-Fitur Android
- 1.4 Arsitektur Android
- 1.5 Contoh Perangkat Android
- 1.6 Market Android

## **BAB 2 SOFTWARE PENDUKUNG PENGEMBANGAN APLIKASI ANDROID**

- 2.1 Software Pendukung
- 2.2 Instalasi Software Pendukung
  - 2.2.1 Instalasi Java Development Kit
  - 2.2.2 Instalasi IDE Eclipse
  - 2.2.3 Instalasi Android SDK
  - 2.2.4 Instalasi Android Developers Tools Plugins
  - 2.2.4 Set Up Emulator
  - 2.2.5 Set Up Perangkat Android

## **BAB 3 PROJECT ANDROID SEDERHANA**

- 3.1 Membuat Project Sangat Sederhana
- 3.2 Struktur Direktori Project

## **BAB 4 ACTIVITY DAN INTENT**

- 4.1 Memahami Activity
- 4.2 Menerapkan Style & Themes Ke Activity
- 4.3 Menyembunyikan Judul Activity
- 4.4 Menampilkan Window Dialog
- 4.5 Menampilkan Progress Dialog
- 4.6 Menghubungkan Activity Dengan Intent
- 4.7 Menangkap Hasil Intent
- 4.8 Mengirim Data Dengan Object Intent
- 4.9 Memanggil Aplikasi Built-In Dengan Intent
- 4.10 Memahami Object Intent
- 4.11 Menampilkan Notifikasi

## **BAB 5 MEMAHAMI ANTARMUKA (UI)**

- 5.1 Memahami Komponen Layar
- 5.2 View Dan ViewGroup
  - 5.2.1 LinearLayout
  - 5.2.2 AbsoluteLayout
  - 5.2.3 TableLayout
  - 5.2.4 RelativeLayout
  - 5.2.5 FrameLayout
  - 5.2.6 ScrollView
- 5.3 Penyesuaian Orientasi Tampilan

5.3.1 Anchoring View-View

5.3.2 Resizing Dan Repositioning

5.4 Mengatur Orientasi Layar

5.4.1 Mempertahankan Kondisi Informasi Selama Perubahan Konfigurasi

5.4.2 Mendeteksi Perubahan Orientasi

5.4.3 Mengendalikan Orientasi Activity

5.4.4 Menciptakan UI Secara Program

5.5 Menangkap Notifikasi UI

5.5.1 Meng-override Method Yang Terdefinisi Dalam Activity

5.5.2 Me-Register Event Untuk View

## **BAB 6 MERANCANG UI DENGAN VIEW**

6.1 View-view Dasar

6.1.1 View TextView

6.1.2 View Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton Dan RadioGroup

6.1.3 View ProgressBar

6.1.4 View AutoCompleteTextView

6.2 View Picker

6.2.1 View TimePicker

6.2.2 Menampilkan TimePicker Dalam Window Dialog

6.2.3 View DatePicker

6.2.4 Menampilkan View DatePicker Dalam Window Dialog

### 6.3 View List

#### 6.3.1 View ListView

#### 6.3.2 Mengkustomisasi ListView

#### 6.3.3 View Spinner

## BAB 7 GAMBAR DAN MENU

### 7.1 View Image

#### 7.1.1 View Gallery Dan ImageView

#### 7.1.2 View ImageSwitcher

#### 7.1.3 View GridView

### 7.2 Membuat Menu

#### 7.2.1 Menciptakan Method Pembantu

#### 7.2.2 Menu Option

#### 7.2.3 Menu Context

### 7.3 Beberapa View Tambahan

#### 7.3.1 View Analogclock Dan DigitalClock

#### 7.3.2 View WebView

## BAB 8 PENGELOLAAN DATA

### 8.1 Menyimpan dan Me-Load User Preferences

#### 8.1.1 Penggunaan getSharedPreferences()

#### 8.1.2 Penggunaan getPreferences()

### 8.2 Pengelolaan Data Ke File

#### 8.2.1 Menyimpan Data Ke Media Penyimpanan Internal

8.2.2 Menyimpan Data Ke Media Penyimpanan Eksternal (SD Card)

8.2.3 Memilih Media Penyimpanan Yang Tepat

8.2.4 Menggunakan Sumberdaya Statis

8.3 Pembuatan Dan Penggunaan Database

8.3.1 Menciptakan Class Pembantu DBAdapter

8.3.2 Menambahkan Kontak

8.3.3 Menampilkan Seluruh Kontak

8.3.4 Menampilkan Satu Kontak

8.3.5 Meng-Update Kontak

8.3.6 Menghapus Kontak

8.3.7 Updgrade Database

8.3.8 Menciptakan Database SQLite Dengan Tool “SQLite Database Browser”

8.3.9 Menyatukan Database Ke Aplikasi

## **BAB 9 CONTENT PROVIDER**

9.1 Berbagi Data Dalam Android

9.2 Penggunaan Content Provider

9.2.1 Konstanta Predefined Query String

9.2.2 Projection

9.2.3 Penyaringan/Filter

9.2.4 Pengurutan

9.3 Membuat Content Provider Sendiri

## 9.4 Penggunaan Content

### **BAB 10 PESAN DAN JARINGAN**

#### 10.1 Pesan SMS

10.1.1 Mengirim Pesan SMS Secara Program

10.1.2 Mendapatkan Umpam Balik Pengiriman Pesan

10.1.3 Mengirim Pesan SMS Dengan Intent

10.1.4 Menerima Pesan SMS

10.1.5 Update Activity Dari BroadcastReceiver

10.1.6 Menjalankan Activity Dari BroadcastReceiver

#### 10.2 Pengiriman Email

#### 10.3 Jaringan/Networking

10.3.1 Men-Download Data Binary

10.3.2 Men-Download File Text

### **BAB 11 LAYANAN LOKASI**

#### 11.1 Peta

11.1.1 Membuat Project Peta

11.1.2 Mendapatkan Maps API Key

11.1.3 Menampilkan Peta

11.1.4 Menampilkan Zoom Control

11.1.5 Mengubah View Peta

11.1.6 Menavigasi Lokasi Tertentu

11.1.7 Menambahkan Marker

11.1.8 Mendapatkan Informasi Lokasi Yang Disentuh

## **Bab 12 PUBLIKASI APLIKASI ANDROID**

12.1 Persiapan Publikasi

12.1.1 Pemberian Versi

12.1.2 Tandatangan Digital Pada Aplikasi

12.2 Men-Deploy File APK

12.2.1 Penggunaan Tool adb.exe

12.2.2 Penggunaan Web Server

12.2.3 Publikasi Ke Android Market

# **BAB 1**

## **MENGENAL ANDROID**

Disadari atau tidak, tren aplikasi perangkat bergerak (mobile application) maju begitu pesatnya, mulai dari telepon cerdas (smartphone), tablet, sampai TV cerdas (smart TV).

Pada masa kini mobilitas seseorang yang cepat berdampak pula pada mobilitas aplikasi serta perangkatnya pendukungnya sehingga menyebabkan aplikasi bergerak terus meningkat perkembangannya seiring dengan semakin canggihnya perangkat keras yang mendukung aplikasi tersebut termasuk sistem operasi yang digunakan. Salah satu pemain yang mengembangkan sistem operasi pada perangkat bergerak adalah Android!.

### **1.1 Apa Android Itu?**

Android adalah sistem operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi Linux, namun telah dimodifikasi. Android diambil alih oleh Google pada tahun 2005 dari Android, Inc sebagai bagian strategi untuk mengisi pasar sistem operasi bergerak. Google mengambil alih seluruh hasil kerja Android termasuk tim yang mengembangkan Android.

Google menginginkan agar Android bersifat terbuka dan gratis, oleh karena itu hampir setiap kode program Android diluncurkan berdasarkan lisensi open-source Apache yang berarti bahwa semua orang yang ingin menggunakan Android dapat men-download penuh source code-nya.

Disamping itu produsen perangkat keras juga dapat menambahkan *extension*-nya sendiri ke dalam Android sesuai kebutuhan produk

mereka. Model pengembangannya yang sederhana membuat Android menarik bagi vendor-vendor perangkat keras (contoh: Samsung).

Keuntungan utama dari Android adalah adanya pendekatan aplikasi secara terpadu. Pengembang hanya berkonsentrasi pada aplikasi saja, aplikasi tersebut bisa berjalan pada beberapa perangkat yang berbeda selama masih ditenagai oleh Android (pengembang tidak perlu mempertimbangkan kebutuhan jenis perangkatnya).

## 1.2 Versi Android

Android telah mengalami update sejak diluncurkan pertama kali.

Versi Android	Diluncurkan	Nama Kode
Beta	5 November 2007	
1.0	23 September 2008	
1.1	9 Februari 2009	
1.5	30 April 2009	Cupcake
1.6	15 September 2009	Donut
2.0/2.1	26 Oktober 2009	Éclair
2.2	20 Mei 2010	Froyo
2.3	6 Desember 2010	Gingerbread
3.0	22 Februari 2011	Honeycomb
4.0.1	19 Oktober 2011	Ice Cream Sandwich
	Sekitar Pertengahan 2012	Jelly Bean
	Sekitar 2013	Key Lime Pie

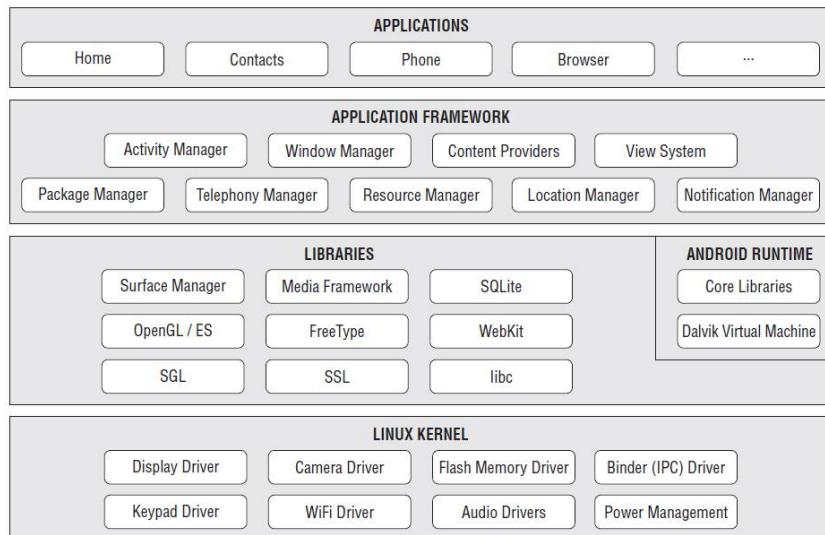
## 1.3 Fitur-Fitur Android

Android tersedia secara open source bagi manufaktur perangkat keras untuk memodifikasinya sesuai kebutuhan. Meskipun konfigurasi perangkat Android tidak sama antara satu perangkat dengan perangkat lainnya, namun Android sendiri mendukung fitur-fitur berikut ini:

- Penyimpanan (Storage) - menggunakan SQLite yang merupakan database *relational* yang ringan untuk menyimpan data.
- Koneksi (Connectivity) - mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (termasuk A2DP dan AVRCP), WiFi, LTE dan WiMAX.
- Pesan (Messaging) – mendukung SMS dan MMS.
- Web Browser – menggunakan open-source WebKit termasuk di dalamnya engine Chrome V8 JavaScript .
- Media – media yang didukung antara lain: H.263, H.264 (3GP atau MP4 container), MPEG-4 SP, AMR, AMR-WB (3GP container), AAC, HE-AAC (MP4 atau 3GP container), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF dan BMP.
- Hardware – terdapat Accelerometer Sensor, Camera, Digital Compass, Proximity Sensor dan GPS.
- Multi-touch – mendukung layar multi-touch.
- Multi-tasking – mendukung aplikasi multi-tasking.
- Dukungan Flash - Android 2.3 mendukung Flash 10.1.

## 1.4 Arsitektur Android

Agar lebih mudah memahami bagaimana Android bekerja, berikut ini bagan tingkatan-tingkatan sistem operasi Android.



**Gambar 1.1 Arsitektur Sistem Operasi Android**

Secara garis besar sistem operasi Andorid terbagi menjadi lima tingkatan:

- **Linux kernel** – adalah kernel dasar Android. Tingkat ini berisi semua *driver* perangkat tingkat rendah untuk komponen-komponen hardware perangkat Android.
- **Libraries** – berisi semua kode program yang menyediakan layanan-layanan utama sistem operasi Android. Sebagai contoh library SQLite yang menyediakan dukungan database sehingga aplikasi Android dapat menggunakannya untuk menyimpan data. Library WebKit yang menyediakan fungsi-fungsi *browsing web*, dan lain-lain.
- **Android Runtime** – kedudukannya setingkat dengan **libraries**, Android Runtime menyediakan kumpulan pustaka inti yang dapat diaktifkan oleh pengembang untuk menulis kode aplikasi Android dengan bahasa pemrograman Java. *Dalvik Virtual Machine* aktif setiap kali aplikasi Android berproses (aplikasi Android dikompilasi

menjadi Dalvik executable). Dalvik adalah mesin semu yang dirancang khusus untuk Android yang dapat mengoptimalkan daya battery perangkat bergerak dengan memori dan CPU terbatas.

- **Application framework** – adalah semacam kumpulan class built-in yang tertanam dalam sistem operasi Android sehingga pengembang dapat memanfaatkannya untuk aplikasi yang sedang dibangun.
- **Applications** – Pada tingkat ini-lah kita akan bekerja, contoh aplikasi ini banyak ditemui, seperti: Phone, Contact, Browse dan lain-lain. Seperti aplikasi Android pada umumnya yang dapat di-download dan diinstall dari Market Android. Semua aplikasi yang Anda buat terletak pada tingkat **Applications**.

## 1.5 Contoh Perangkat Android

Perangkat Android datang dengan beberapa model dan ukuran. Tipe perangkat yang ditenagai oleh Android dan tersedia dipasaran antara lain: Smartphone, Tablet, Perangkat E-Reader, Netbook, MP4 Player, Internet TV. Contoh produk Android misalnya: Samsung Galaxy S, HTC Desire HD dan LG Optimus Ome Smartphones



Gambar 1.2 Smartphone Android

Kategori perangkat lain yang popular adalah tablet. Tablet memiliki ukuran berbeda-beda, namun pada umumnya berukuran mulai tujuh inci diagonal. Contohnya antara lain: Samsung Galaxy Tab dan Dell Streak. Disamping smartphone dan tablet, ada pula e-book reader seperti Noble NOOKColor.



**Gambar 1.3 Tablet & Ebook Reader Android**

Produk lain yang sedang tren adalah TV Smart Android diawali oleh People of Lava sebuah perusahaan Swedia mengembangkan TV berbasis Android yang disebut Scandinavia TV Android. Google sendiri juga sedang mengembangkan platform TV cerdas berbasis Android dengan mengandeng beberapa perusahaan Intel, Sony dan Logitech.



**Gambar 1.4 TV Smart Android**

## **1.6 Market Android**

Salah satu faktor penentu suksesnya platform smartphone adalah karena banyaknya aplikasi yang tersedia. Oleh karenanya pada Agustus 2008 Google mengenalkan Android Market, yaitu suatu toko aplikasi online untuk perangkat Android dan mulai siap menyediakan aplikasi bagi pengguna pada oktober 2008. Dengan aplikasi Market (aplikasi *pre-installed* pada perangkat Android) pengguna dengan mudah men-download aplikasi dari pihak ketiga secara langsung melalui perangkat mereka. Android Market menyediakan aplikasi berbayar maupun gratis.

# **BAB 2**

# **SOFTWARE PENDUKUNG**

# **PENGEMBANGAN APLIKASI**

# **ANDROID**

## **2.1 Software Pendukung**

Dibutuhkan beberapa software pendukung yang harus terinstall sebelum mengembangkan aplikasi Android, antara lain:

- **Java Development Kit (JDK)**, karena bahasa pemrograman Android menggunakan bahasa Java maka dibutuhkan JDK Java Development Kit. Untuk memperoleh JDK versi terbaru, kunjungi situs <http://www.java.com/en/download/manual.jsp>.
- **IDE Eclipse**, pengembangan aplikasi Android disarankan menggunakan IDE Eclipse (*Integrated Development Environment*), Eclipse adalah IDE software yang digunakan oleh banyak bahasa pemrograman seperti Java, Ada, C, C++, COBOL, Phyton dan lain-lain. Di dalam IDE Eclipse terdapat layanan *system extensible* (semacam sistem penambahan plugins), *editor*, *debugger*, *control tools*, pengaturan direktori dan lain-lain. IDE Eclipse intinya adalah suatu software yang lingkungannya dikondisikan agar memudahkan pengembang membangun suatu aplikasi. Untuk memperoleh IDE Eclipse terbaru kunjungi situs <http://www.eclipse.org/downloads/>.

- **Android Software Development Kit (SDK)**, SDK Android berisi *debugger, library, emulator*, dokumentasi, contoh kode program dan tutorial. SDK Android adalah mesin utama untuk mengembangkan aplikasi Andorid. Untuk mendapatkan SDK Android versi terbaru silakan download di <http://developer.android.com/sdk/index.html>.
- **Android Development Tools (ADT) Plugins**, Plugins ADT berguna sebagai pengenal Android di dalam IDE Eclipse. Dengan ADT plugins kita bisa membuat project aplikasi Android baru, mengakses tools emulator dan perangkat Android, melakukan kompilasi dan mendebug aplikasi, mengekspor aplikasi ke Android Packages (APK), membuat sertifikasi digital terhadap kode program APK. Untuk memperoleh plugin ADT terbaru silakan kunjungi <http://developer.android.com/sdk/eclipse-adt.html>.

---

**Catatan:**

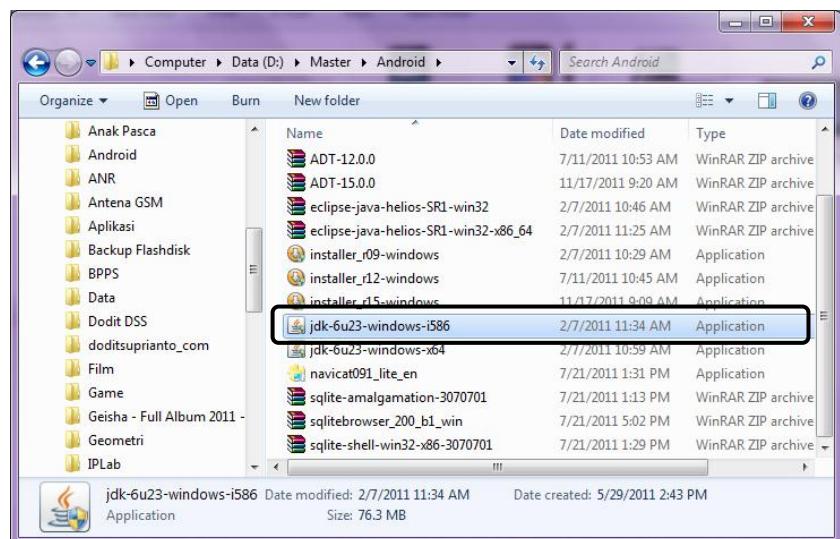
Semua software pendukung tersedia di dalam CD penyerta buku. Jika dibutuhkan software pendukung terbaru maka silakan kunjungi website yang telah tercantum di atas

---

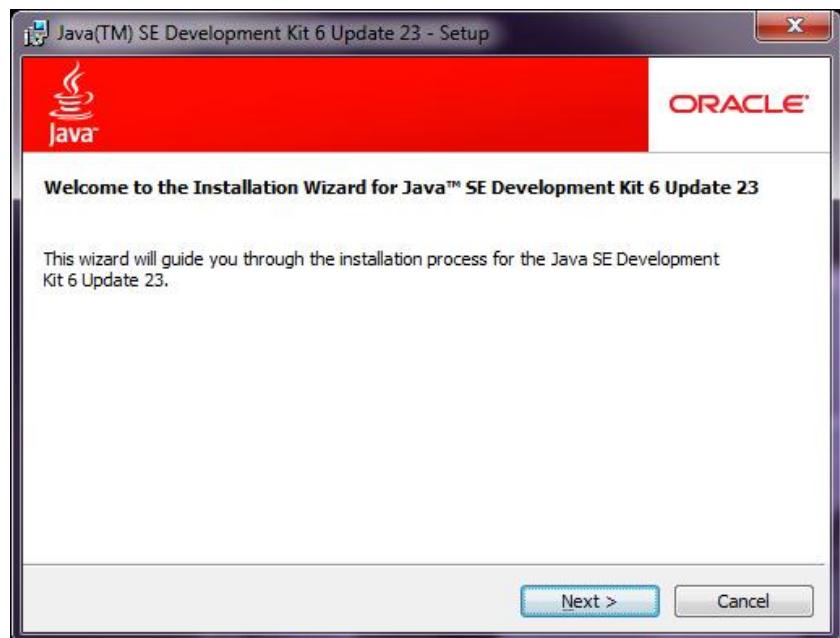
## 2.2 Instalasi Software Pendukung

### 2.2.1 Instalasi Java Development Kit

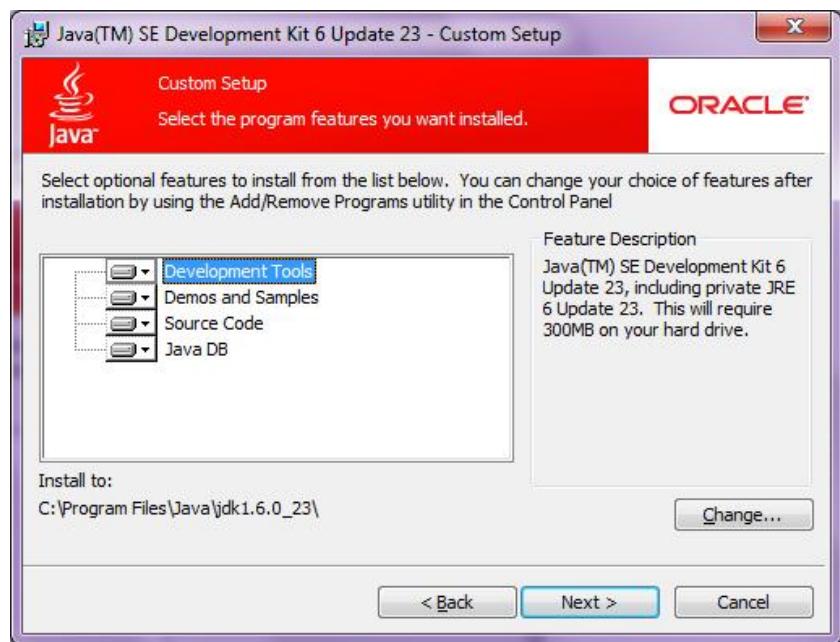
Untuk menginstal JDK sangatlah mudah, klik dua kali file **jdk-6u23-windows-i586** hingga tampak gambar seperti di bawah ini.



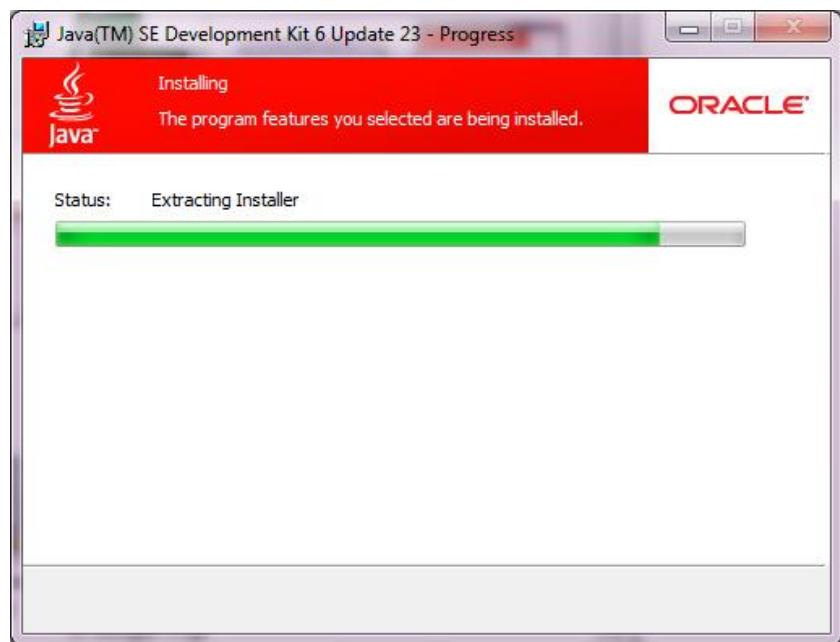
Gambar 2.1 Klik dua kali pada file jdk-6u23-windows-i586



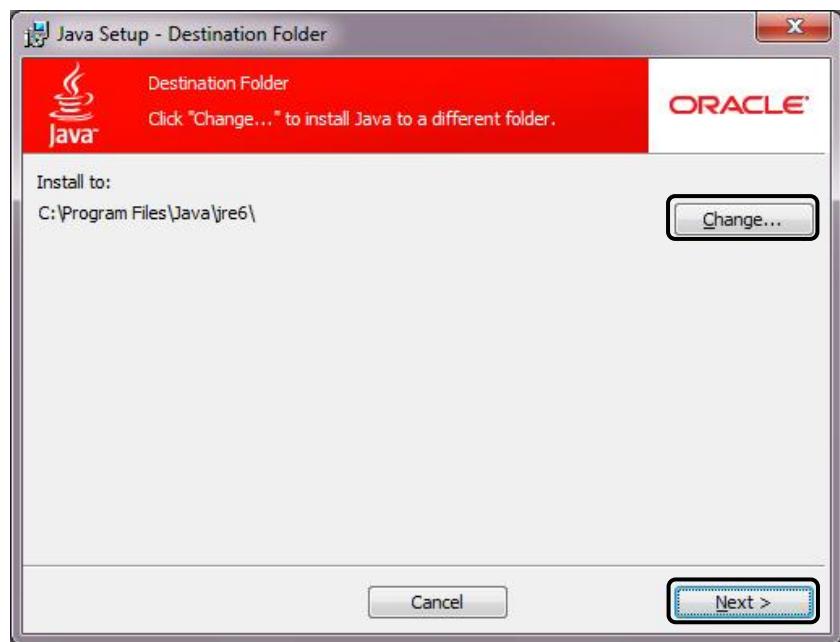
Gambar 2.2 Klik tombol Next.



Gambar 2.3 Jika tidak ada pengurangan fitur JDK maka klik tombol Next



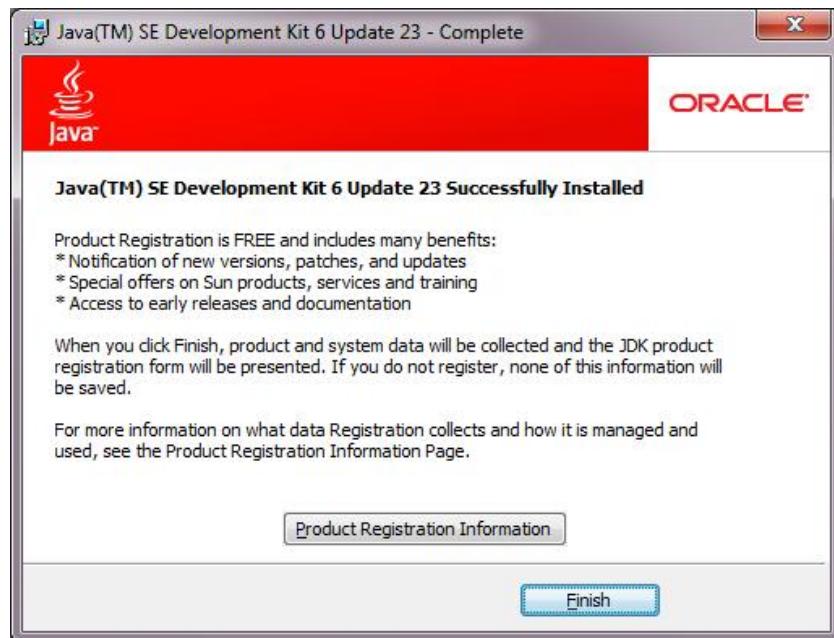
*Gambar 2.4 Tunggu sampai proses penguraian file selesai*



**Gambar 2.5 Klik tombol Change jika hendak mengubah lokasi instalasi JDK ,kemudian klik tombol Next**



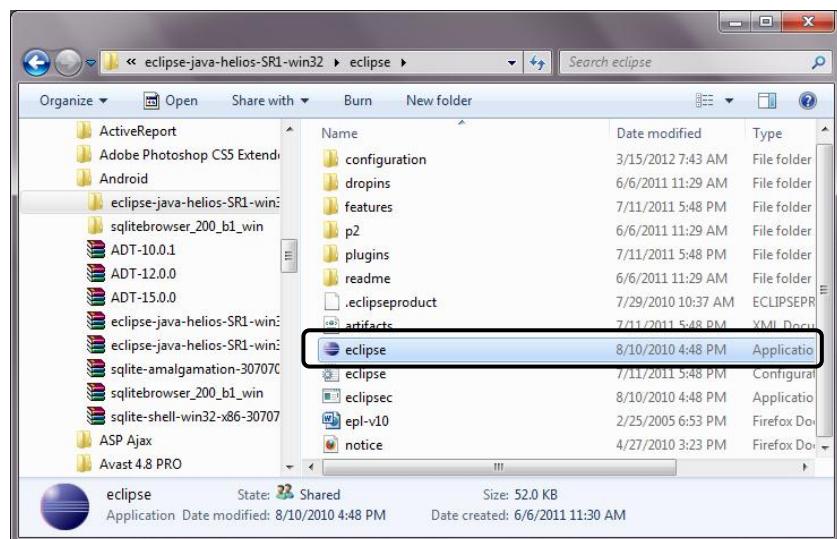
*Gambar 2.6 Proses Instalasi*



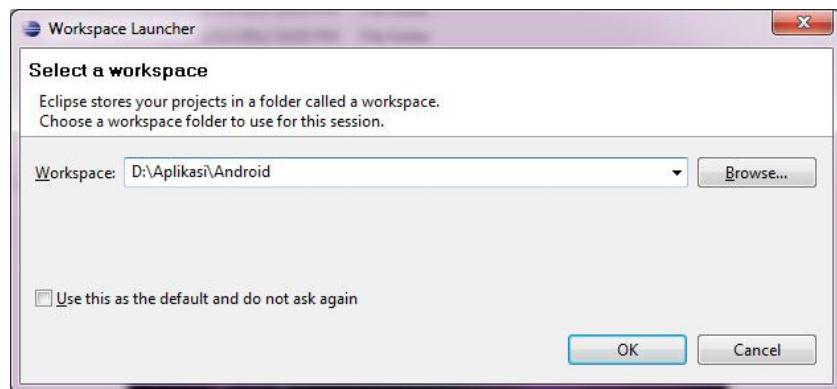
*Gambar 2.7 Klik tombol Finish, menandakan proses instalasi selesai*

### **2.2.2 Instalasi IDE Eclipse**

Tahapan instalasi IDE Eclipse adalah sebagai berikut:



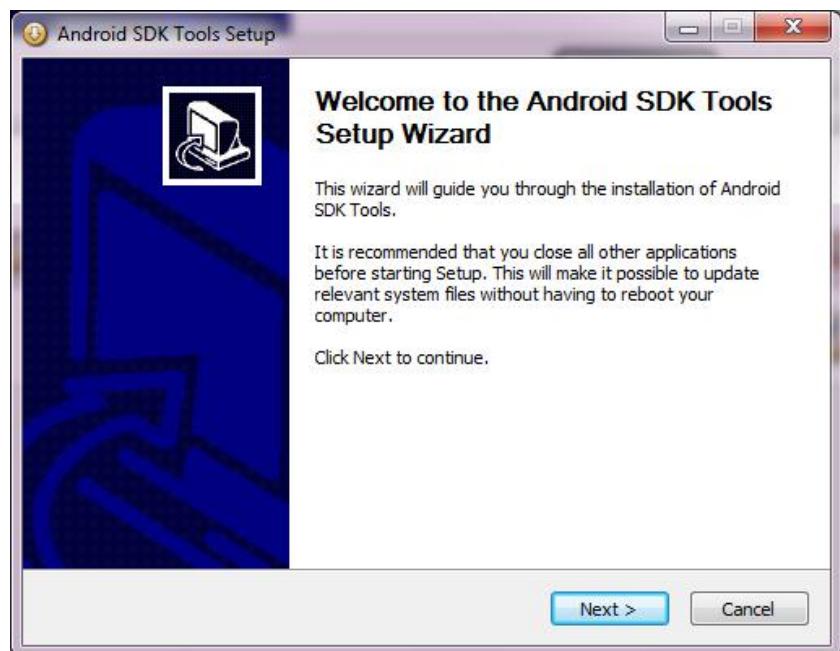
Gambar 2.8 Klik dua kali pada file **eclipse**



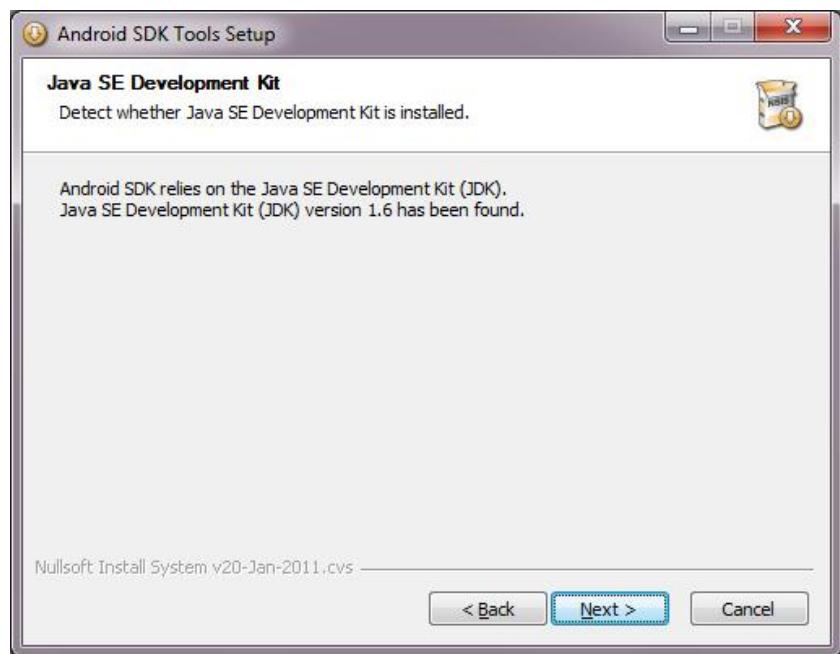
Gambar 2.9 Klik tombol **Browse** untuk menentukan direktori file project program **Eclipse**. Klik tombol **OK** untuk melanjutkan

### 2.2.3 Instalasi Android SDK

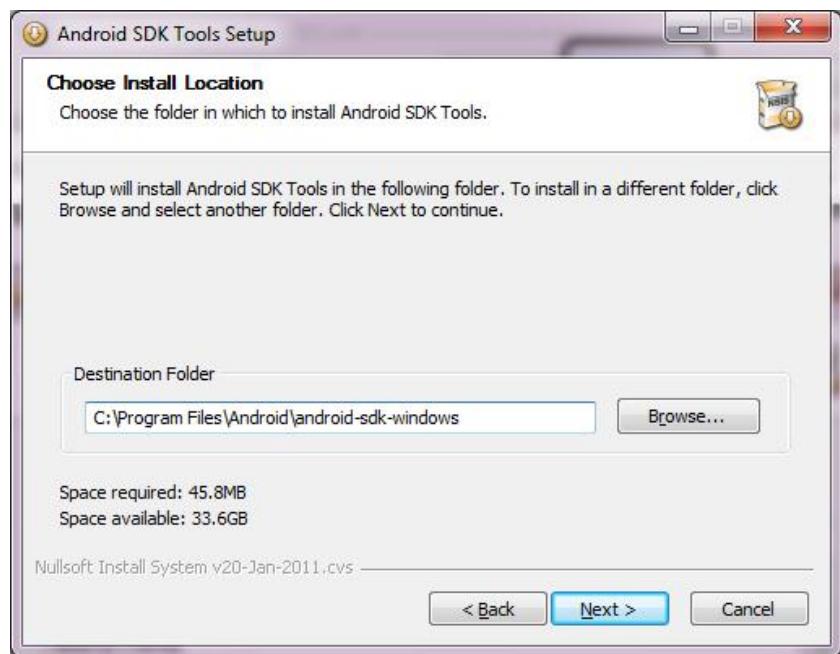
Klik dua kali file Android SDK yaitu **installer\_r09-windows**, nama file tidak harus sama, tergantung dari versi Android SDK yang di download (sebaiknya gunakan Android versi terbaru), sedangkan r\_09 menunjukkan versi Android SDK-nya.



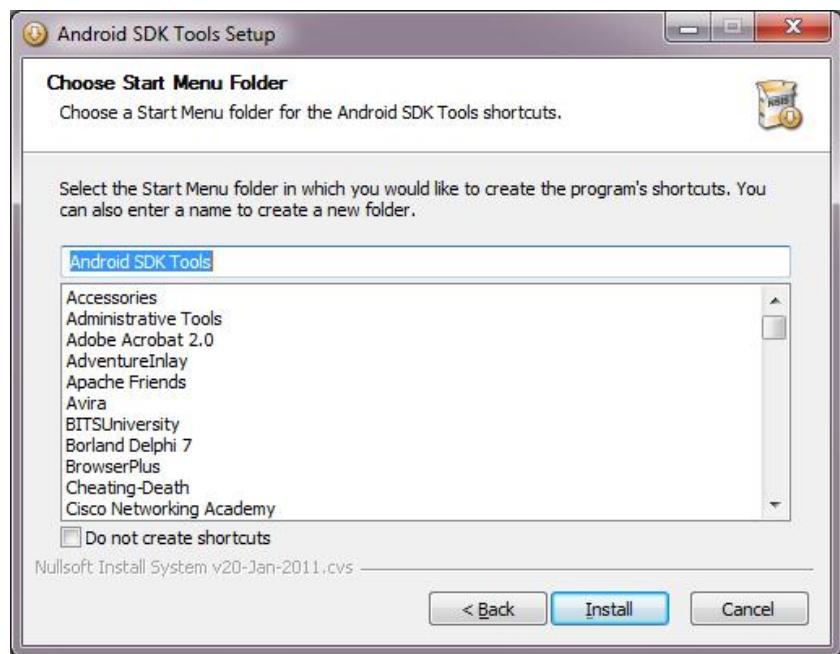
*Gambar 2.10 Klik tombol Next untuk melanjutkan*



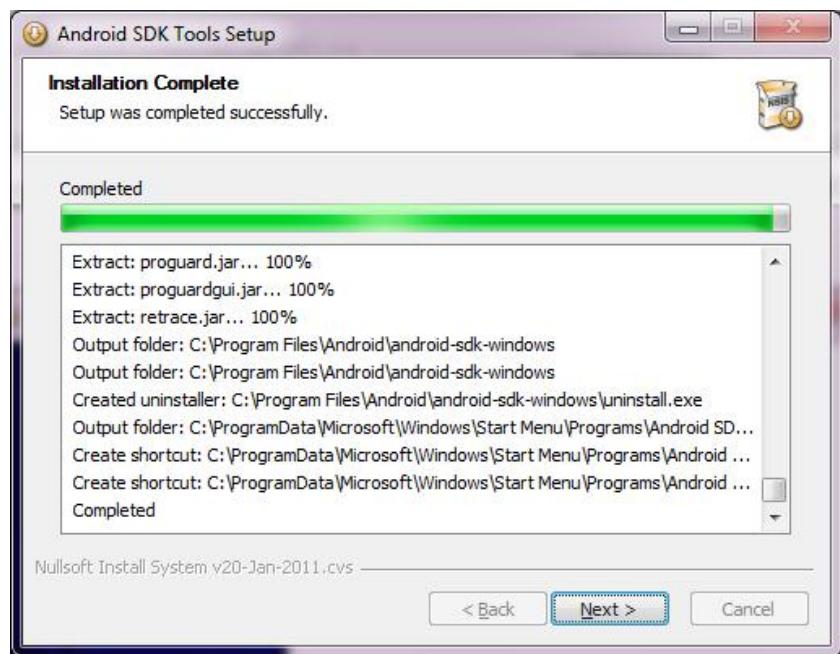
**Gambar 2.11 Klik tombol Next untuk melanjutkan**



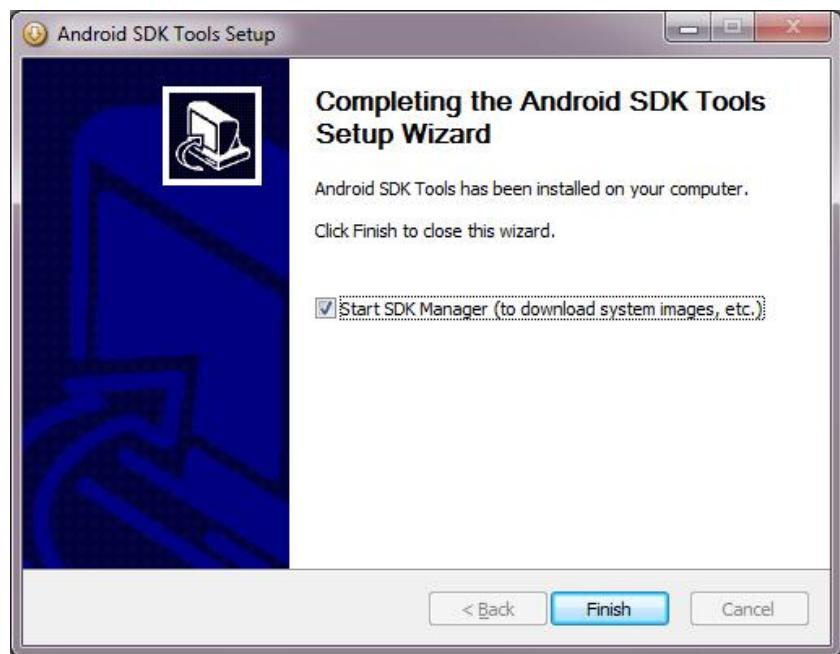
**Gambar 2.12 Klik tombol Browse untuk menentukan direktori Android SDK, kemudian klik Next untuk melanjutkan.**



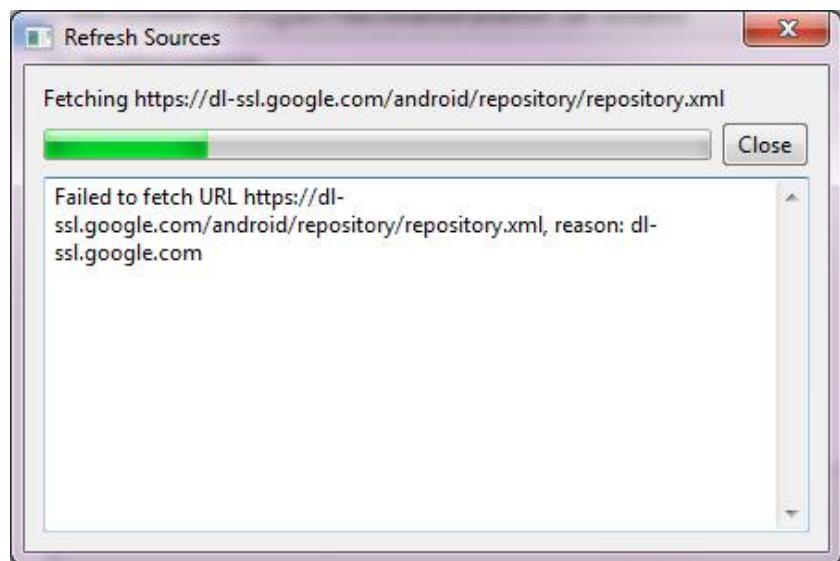
**Gambar 2.13 Klik tombol *Install* untuk melanjutkan**



**Gambar 2.14 Tunggu sampai proses selesai, klik tombol Next**



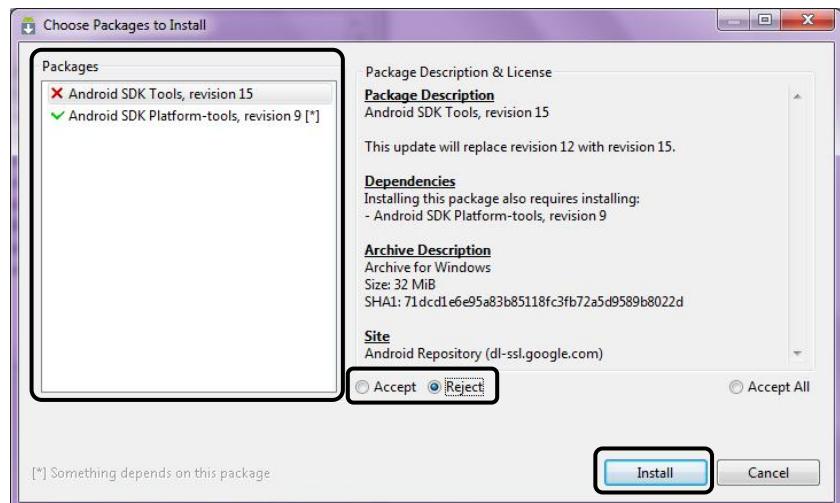
**Gambar 2.15 Klik tombol *Finish* yang menandakan bahwa proses instalasi *Android SDK* selesai.**



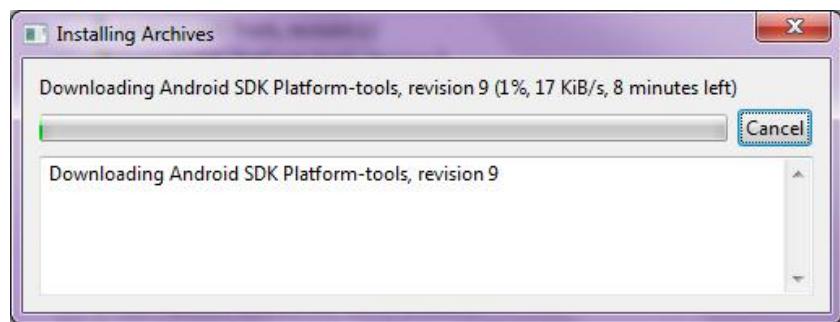
**Gambar 2.16 Terhubung ke internet untuk update file**

Selanjutnya adalah proses update, hal ini untuk memastikan bahwa SDK Android selalu versi terkini. Proses update memerlukan koneksi internet.

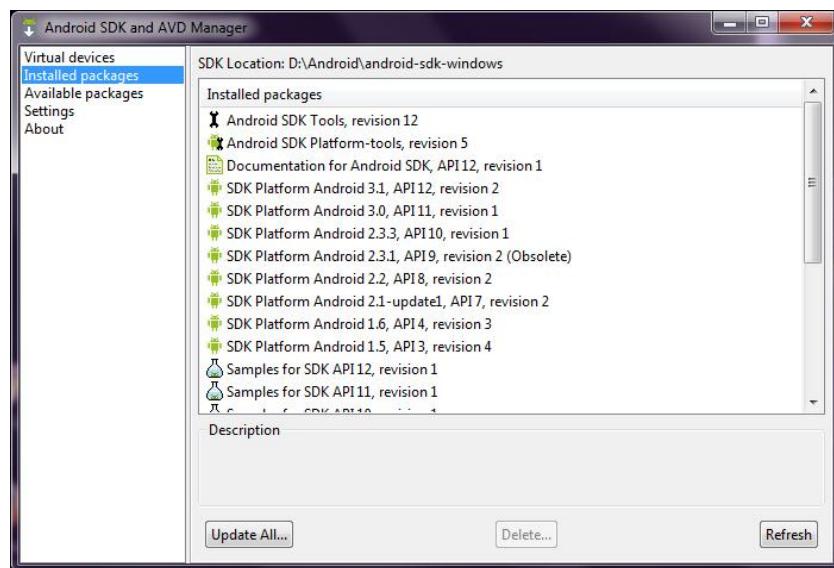
Pilih beberapa *package* yang diperlukan untuk diupdate dengan cara memilih salah satu package, kemudian pilih **Accept**. Sebaliknya jika tidak ingin mengupdate maka pilih **Reject**.



**Gambar 2.17 Accept untuk mengupdate package, Reject untuk tidak mengupdate package. Klik tombol Install untuk melanjutkan.**



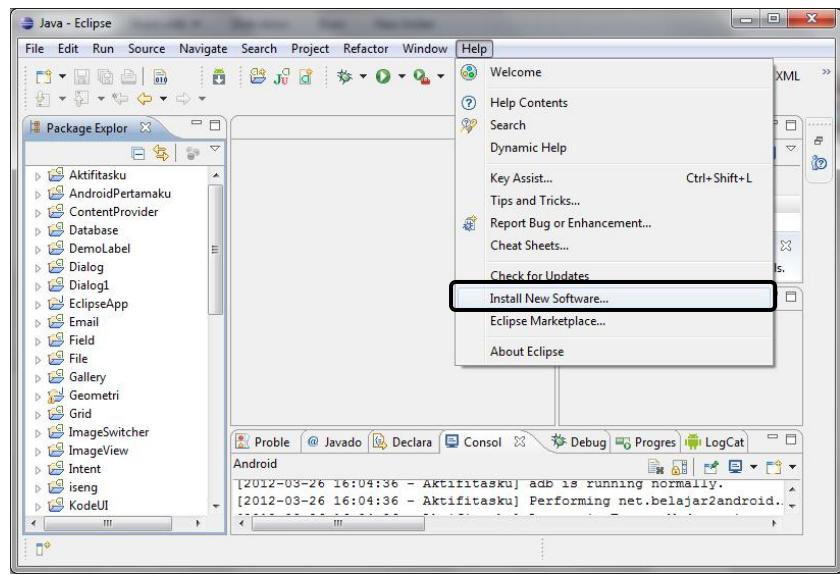
**Gambar 2.18 Proses update file, butuh proses yang lama**



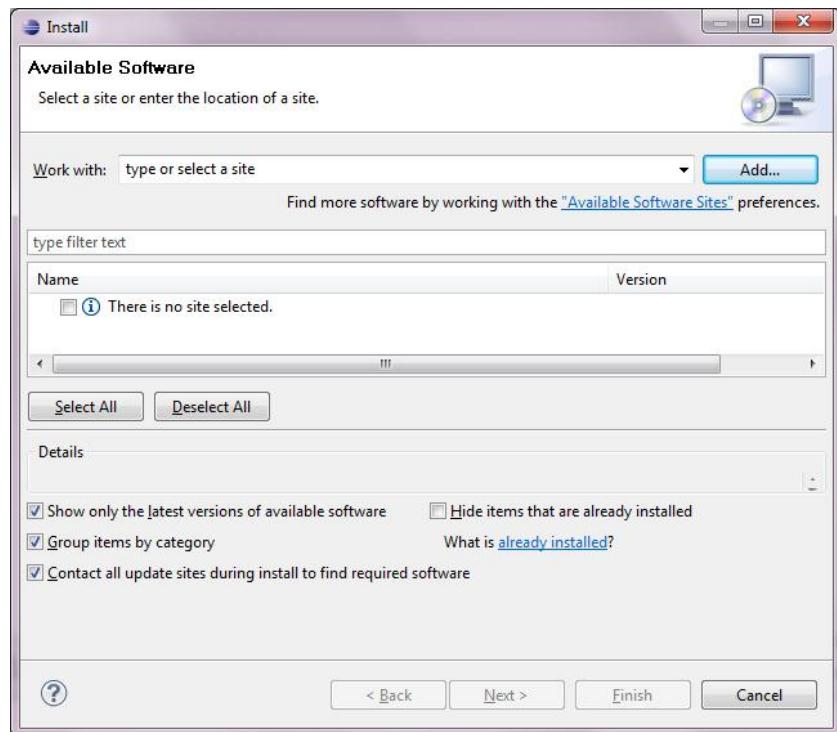
Gambar 2.19 Daftar package yang telah terinstall

#### 2.2.4 Instalasi Android Developers Tools Plugins

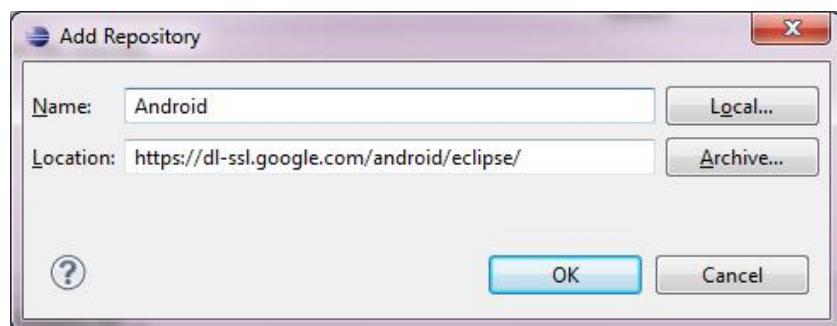
Selanjutnya adalah menginstall ADT Plugins, klik menu Eclipse **Help|Install New Software**, kemudian beri nama, misalnya: "Android", kemudian tulis pada textbox "Work with" dengan <https://dlssl.google.com/android/eclipse/> atau lengkapnya adalah "Android - <https://dl-ssl.google.com/android/eclipse/>", kemudian klik tombol **Add**, maka Eclipse akan men-download beberapa plugins yang tersedia di website tersebut.



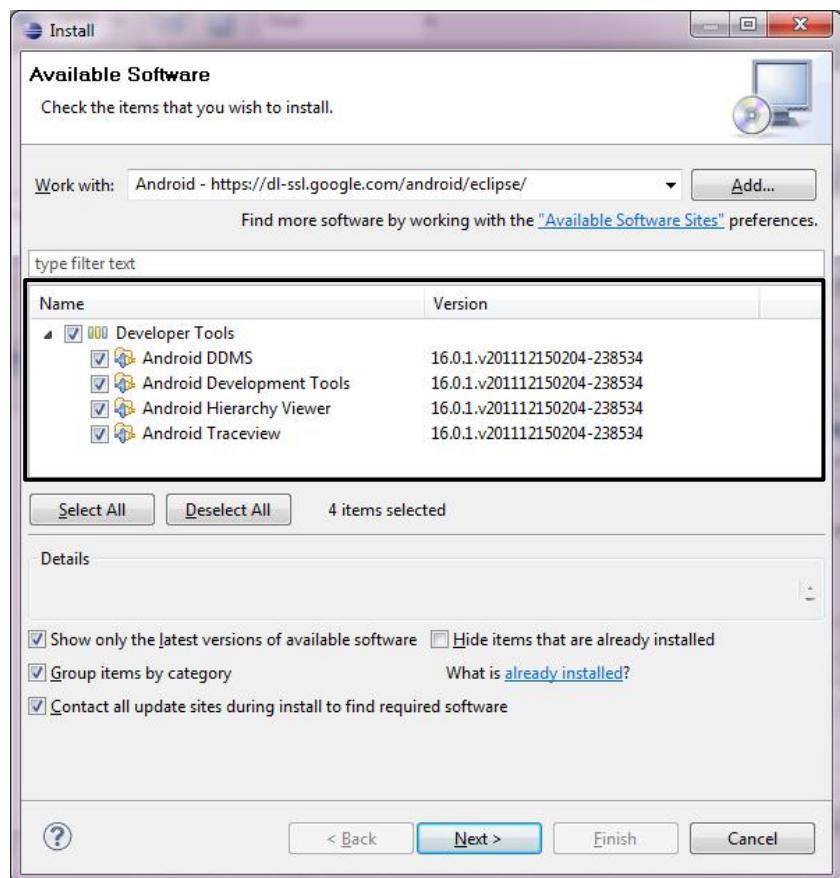
**Gambar 2.20 Klik “Install New Software ...” untuk menginstall ADT Plugins**



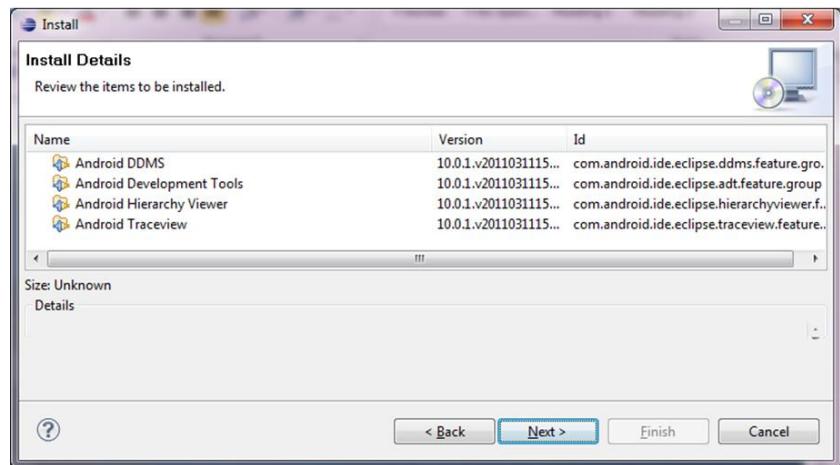
**Gambar 2.21 Klik tombol Add untuk melanjutkan**



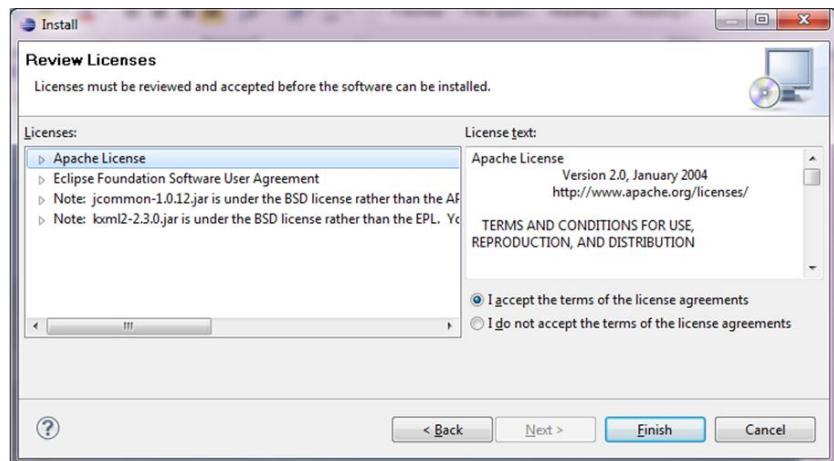
**Gambar 2.22 Tuliskan “Android” pada textbox Name, tuliskan “https://dl-ssl.google.com/android/eclipse/” pada textbox Location. Klik tombol OK untuk melanjutkan.**



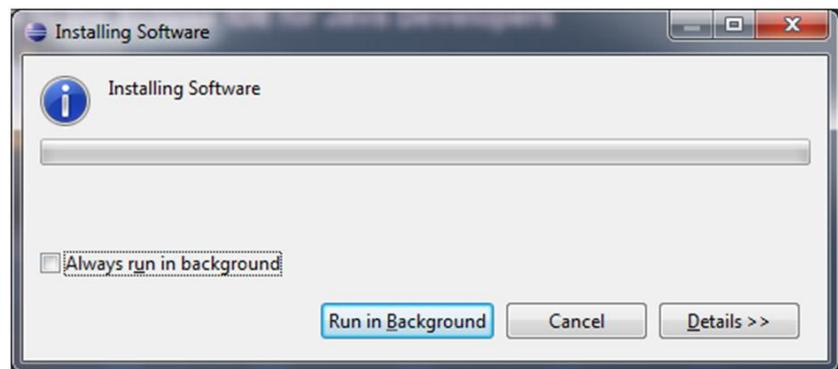
Gambar 2.23 Daftar Plugins ADT. Klik tombol Next untuk melanjutkan



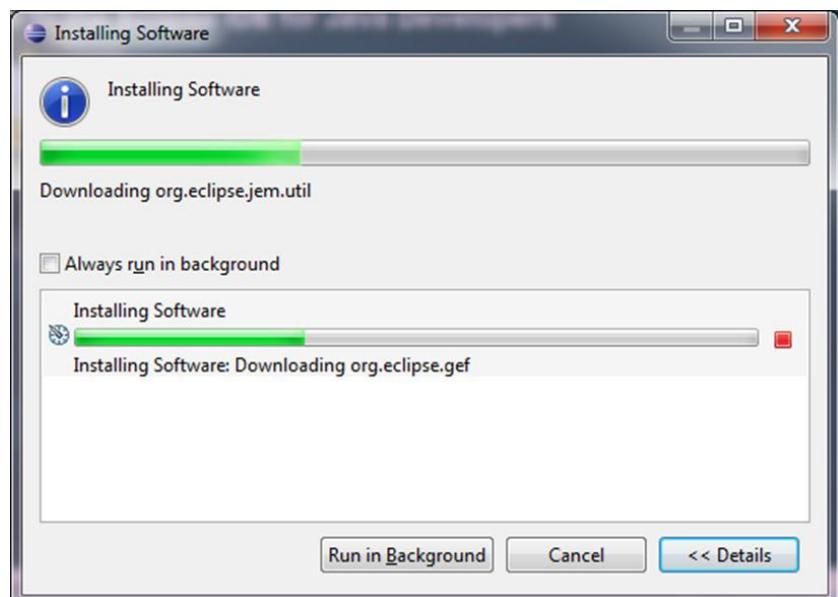
**Gambar 2.24 Plugins yang akan diinstall. Klik tombol Next untuk melanjutkan**



**Gambar 2.25 Pilih "I accept the terms of the license agreements" kemudian klik tombol Next untuk melanjutkan"**



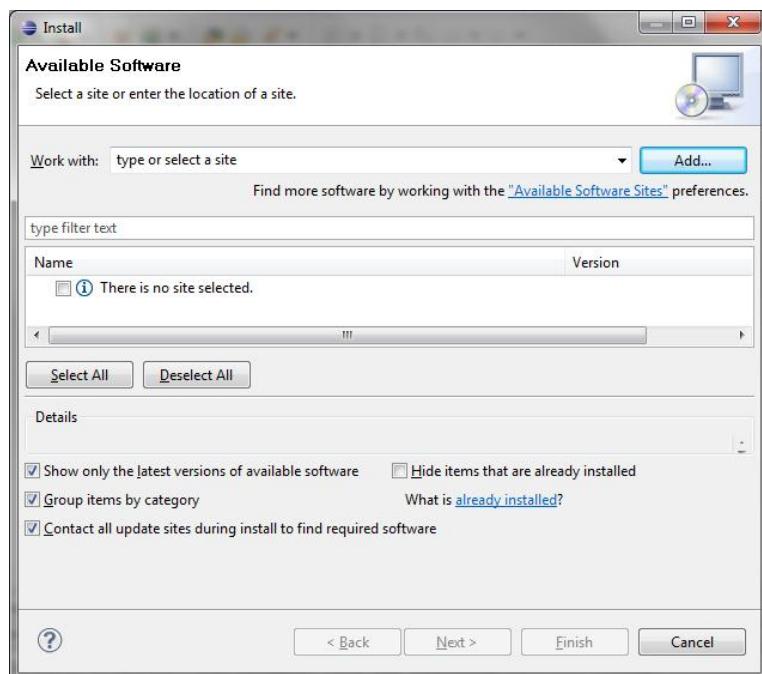
**Gambar 2.26 Proses download dan menginstall plugins. Proses ini membutuhkan koneksi dengan internet.**



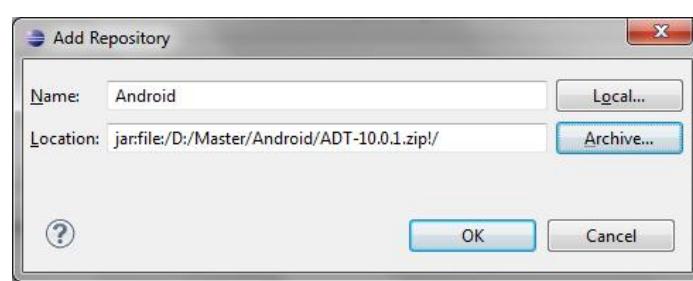
**Gambar 2.27 Sampai pada tahap ini komputer harus di-restart ulang.**

Cara lain meng-install ADT Plugins tanpa melalui koneksi internet adalah dengan men-dowload file ADT yang dapat diperoleh dari alamat URL <http://dl.google.com/android/ADT-10.0.1.zip> (atau bisa pula diperoleh

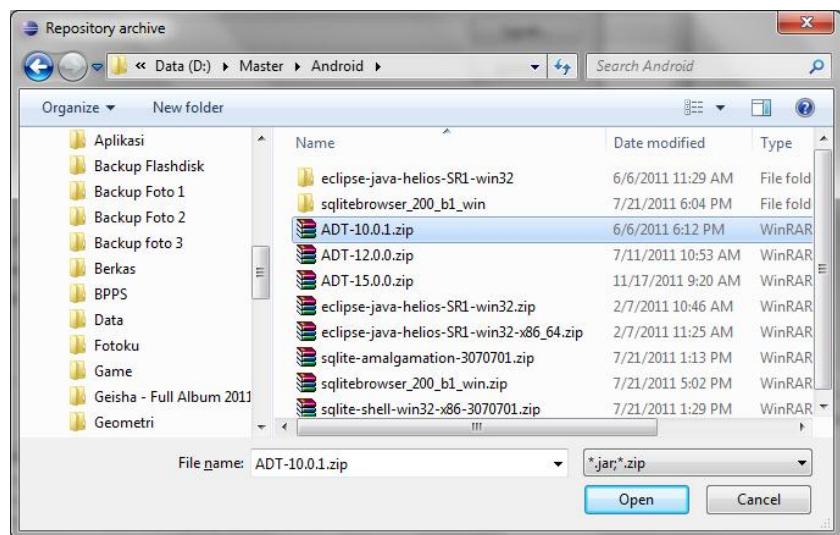
dari CD penyerta buku). Kemudian lakukan hal yang sama seperti langkah-langkah sebelumnya, tetapi tujuannya bukanlah alamat url <https://dl-ssl.google.com/android/eclipse/> tetapi mencari file zip ADT dengan mengklik tombol **Archives**, seperti berikut ini:



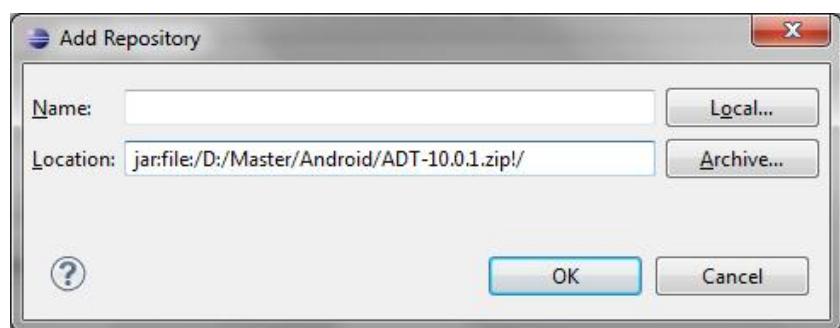
**Gambar 2.28 Klik tombol Add**



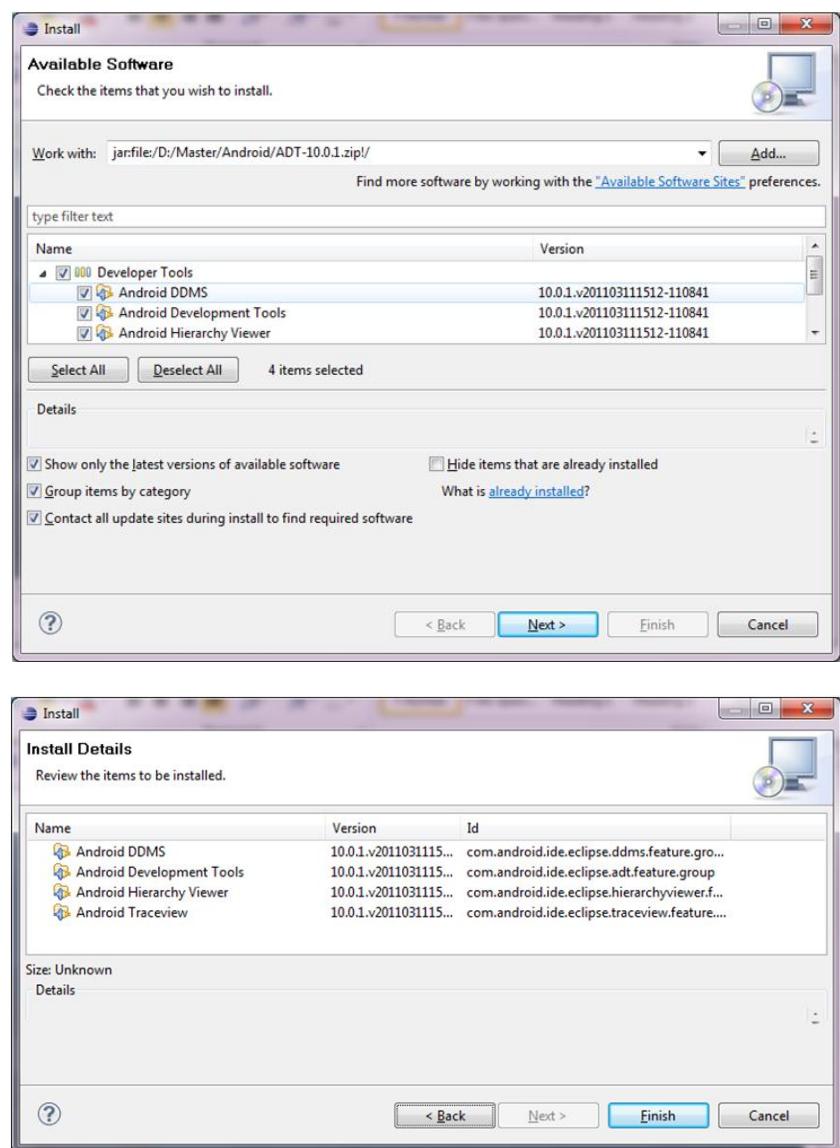
**Gambar 2.29 Beri nama bebas, misalnya "Android" pada textbox Name, kemudian klik tombol Archive**



Gambar 2.30 Tambahkan file ZIP ADT-10.0.1

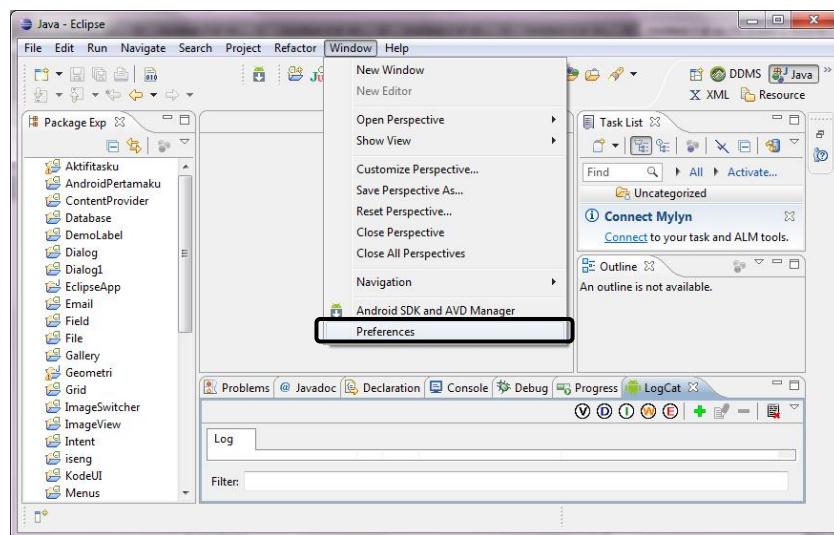


Gambar 2.31 Menempatkan alamat path ke textbox location

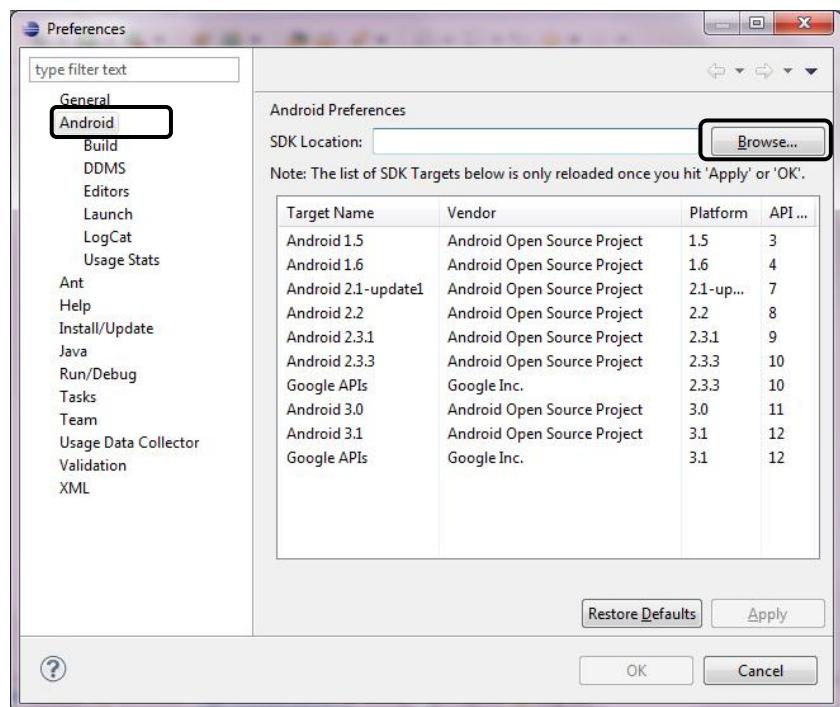


**Gambar 2.32 Sampai pada tahap ini komputer harus di-restart ulang.**

Selanjutnya adalah kita perlu memberitahukan ADT dimana instalasi SDK Android berada, caranya: buka program Eclipse klik menu **Window|Preferences**

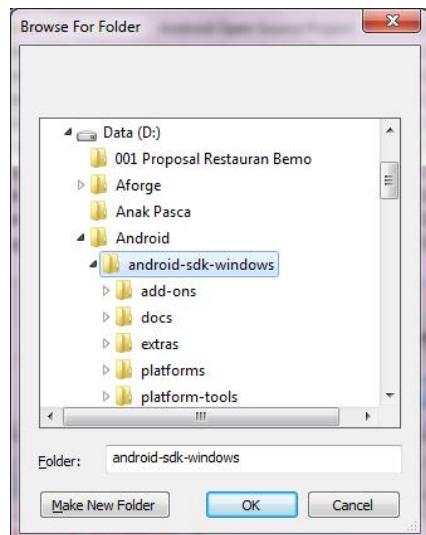


**Gambar 2.33 Klik menu Window / Preferences pada program Eclipse**

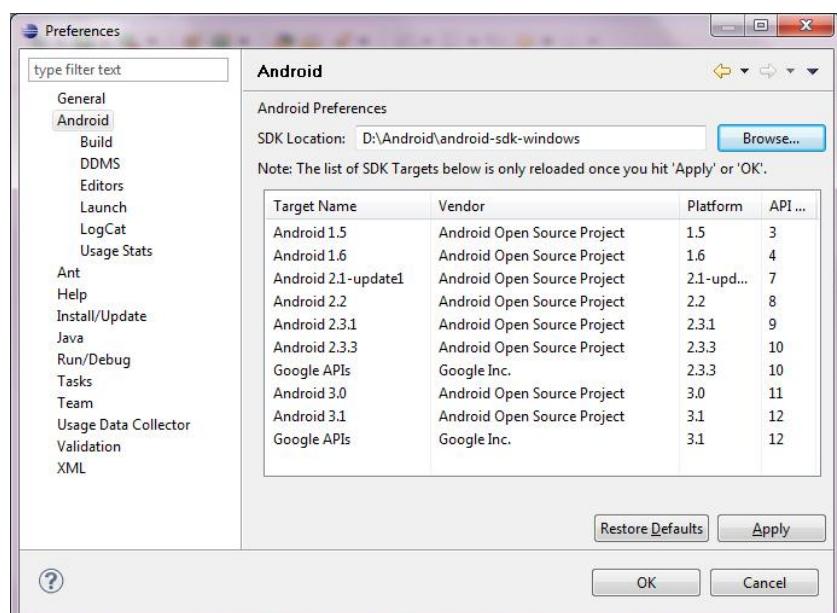


**Gambar 2.34 Klik pada tab Android, kemudian klik tombol Browse**

Tentukan di folder mana Android SDK berada, misalnya di folder  
**D:\Android\android-sdk-windows**



Gambar 2.35 Cari letak Android SDK, kemudian klik OK

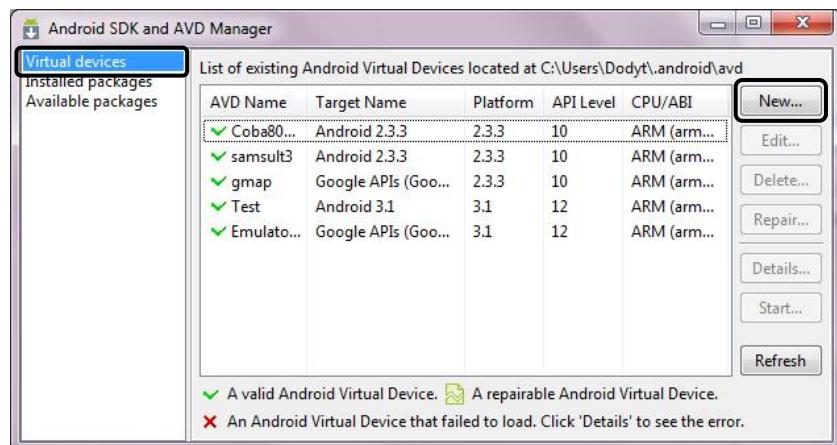


Gambar 2.36 Setelah itu klik OK dan ADT plugins siap digunakan

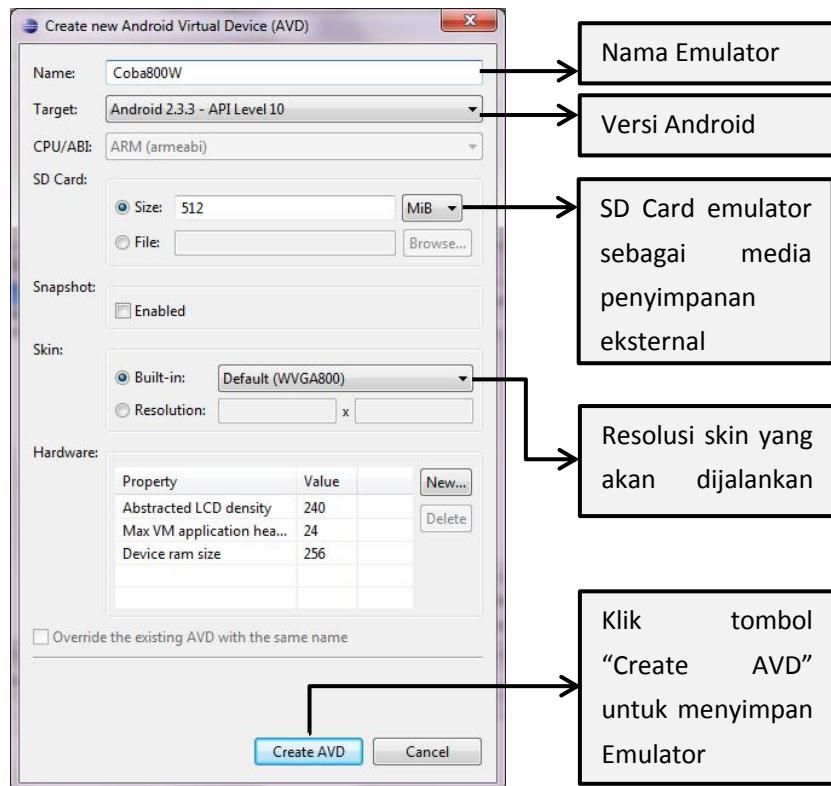
## 2.2.4 Set Up Emulator

Tools Android sudah dilengkapi dengan emulator, emulator tersebut berperilaku seperti perangkat Android (smartphone Android sesungguhnya). Ini sangat membantu ketika mengembangkan aplikasi Android, tidak hanya menjalankan aplikasi tanpa perangkat android sesungguhnya saja tetapi juga dapat membantu menguji konfigurasi perangkat android.

Emulator Android dapat meng-emulasi lebih dari satu perangkat Android. Setiap konfigurasi dapat disimpan dalam **Android Virtual Device** atau AVD. Adapun cara konfigurasinya adalah: buka Android SDK dari menu program **Start|All Programs|Android sdk Tools|SDK Manager**, pilih daftar **Virtual Devices**, kemudian klik tombol **New**. Bisa pula dibuka melalui Eclipse pada menu **Windows|Android SDK and AVD Manager**.



Gambar 2.37 Setup emulator AVD (Android Virtual Devices)

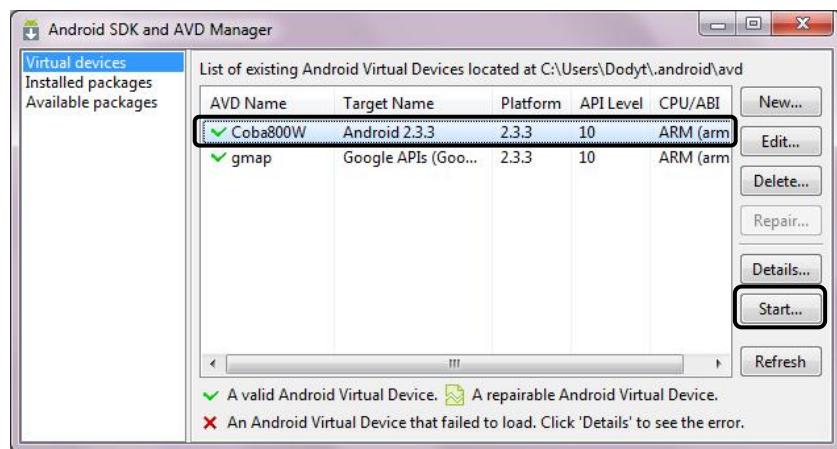


**Gambar 2.38 Konfigurasi AVD**

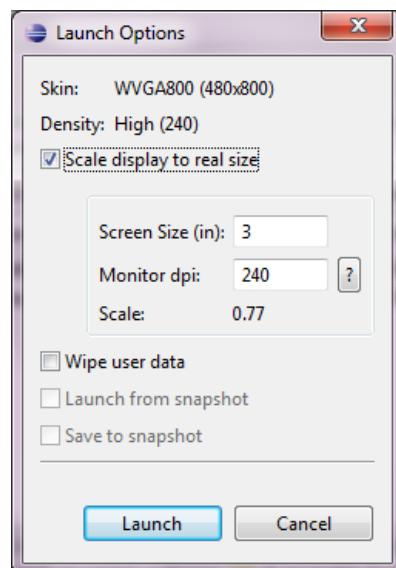
Setiap sistem operasi Android diidentifikasi berdasarkan nomor API level-nya. Contoh, Android 2.3 adalah level 9 (API 9), Android 2.2 adalah level 8 (API 8) dan seterusnya.

Jika program aplikasi membutuhkan Google Map, maka harus digunakan AVD ber-platform Google API (akan dibahas pada bab selanjutnya).

Untuk menjalankan emulator, pilih salah satu emulator kemudian klik tombol **Start**

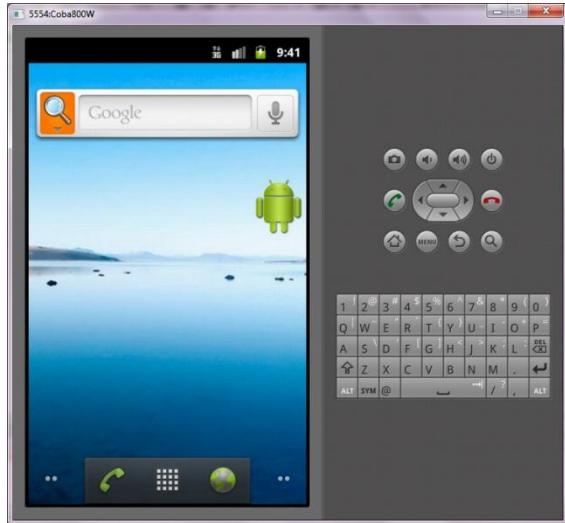


**Gambar 2.39 Pilih salah satu emulator yang akan dijalankan. Klik tombol Start untuk menjalankan emulator**



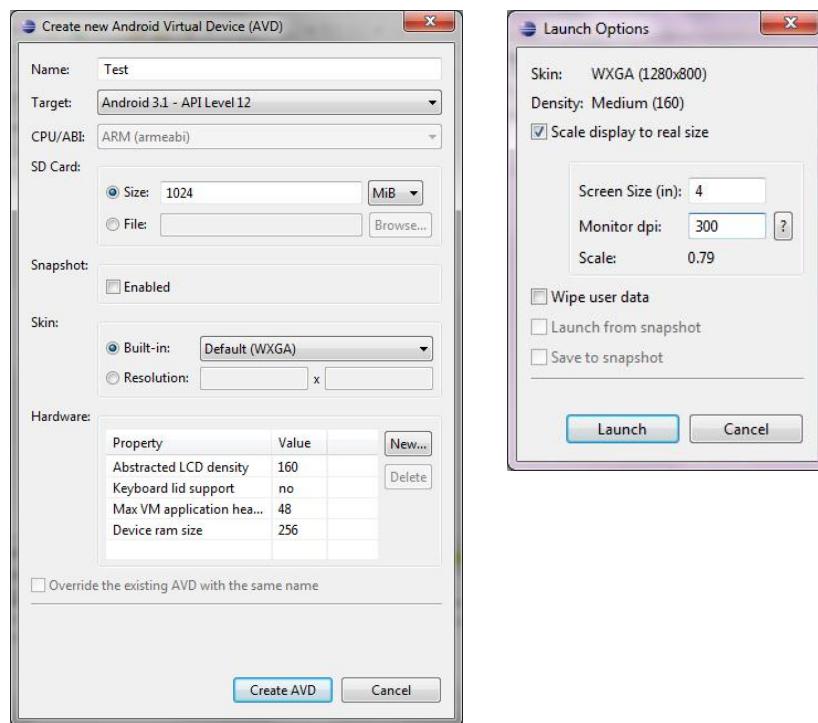
**Gambar 2.40 Centang pernyataan "Scale display to real size" untuk mengskala tampilan layar agar tidak melebihi layar monitor komputer. Selanjutnya klik tombol Launch**

Adapun tampilan emulornya adalah sebagai berikut:



**Gambar 2.41 Tampilan AVD API Level 10**

Berikut ini contoh pengaturan Emulator Android jika menggunakan Android 3.1 – API Level 12



**Gambar 2.42 Konfigurasi AVD API Level 12**

Adapun tampilannya kurang lebih seperti berikut ini:

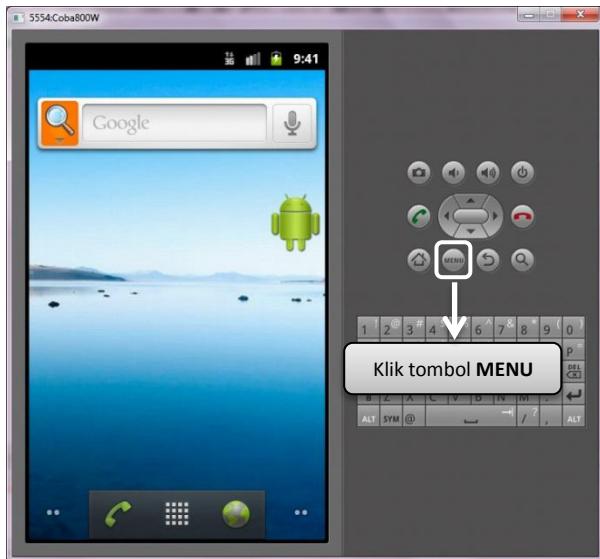


Gambar 2.43 Tampilan AVD API Level 12

### 2.2.5 Set Up Perangkat Android

Seandainya kita memiliki perangkat Android sesungguhnya, maka lebih baik bila sebelum mentransfer aplikasi ke perangkat Andorid diuji terlebih dahulu dengan emulator, hingga memastikan aplikasi tersebut telah teruji.

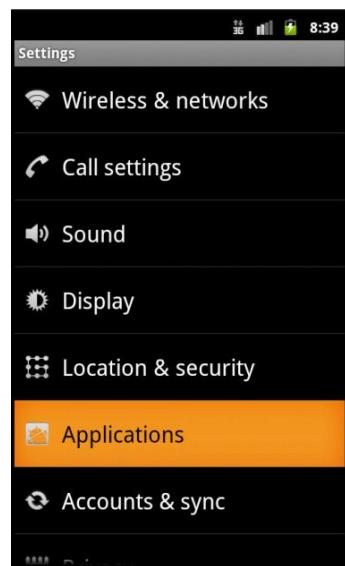
Langkah pertama untuk memastikan bahwa perangkat Android siap untuk ditransfer dengan aplikasi adalah dengan mengkonfigurasinya terlebih dahulu. Klik **Menu|Settings|Applications|Development** dan pastikan bahwa checkbox **USB debugging** dan **Stay awake** tercentang.



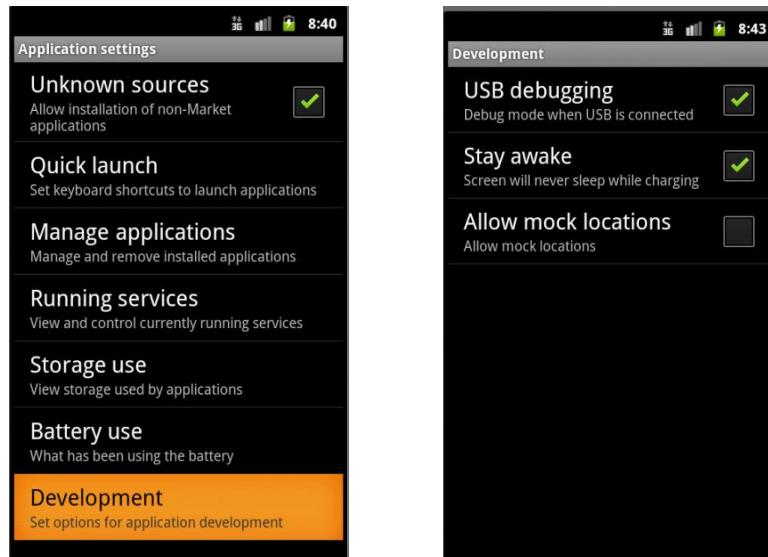
**Klik tombol MENU**



**Pilih menu Settings**



**Klik menu Applications**



**Klik menu Development**

**Pastikan checkbox USB debugging dan Stay awake tercentang**

**Gambar 2.44 Pengaturan emulator dan perangkat Android**

Dengan mengaktifkan USB debugging maka kita dapat menggunakan perangkat dengan build tools-nya Android, terutama saat proses debugging, sedangkan Stay awake bertujuan agar perangkat Android tetap menyala dalam waktu yang lama, sampai hubungan antara USB perangkat Android ke perangkat komputer terputus, sehingga tidak terjadi *Unlock* saat proses pengembangan aplikasi.

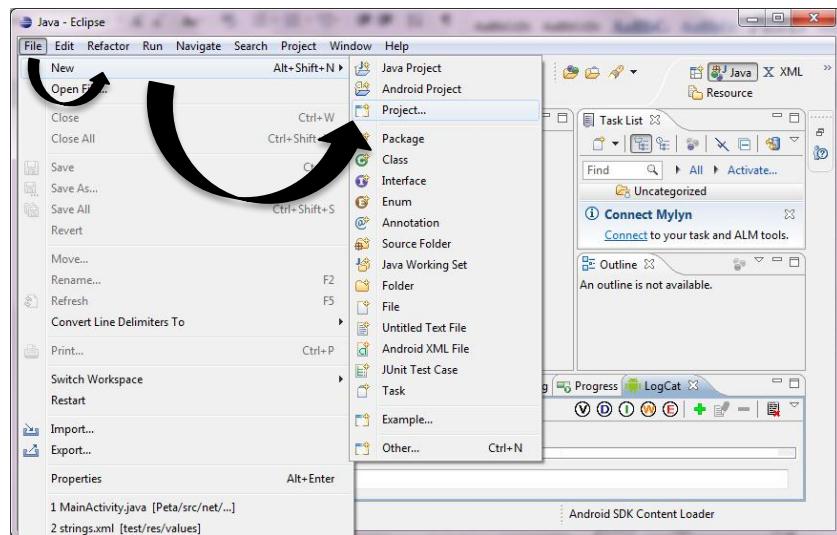
# BAB 3

# PROJECT ANDROID

# SEDERHANA

## 3.1 Membuat Project Sangat Sederhana

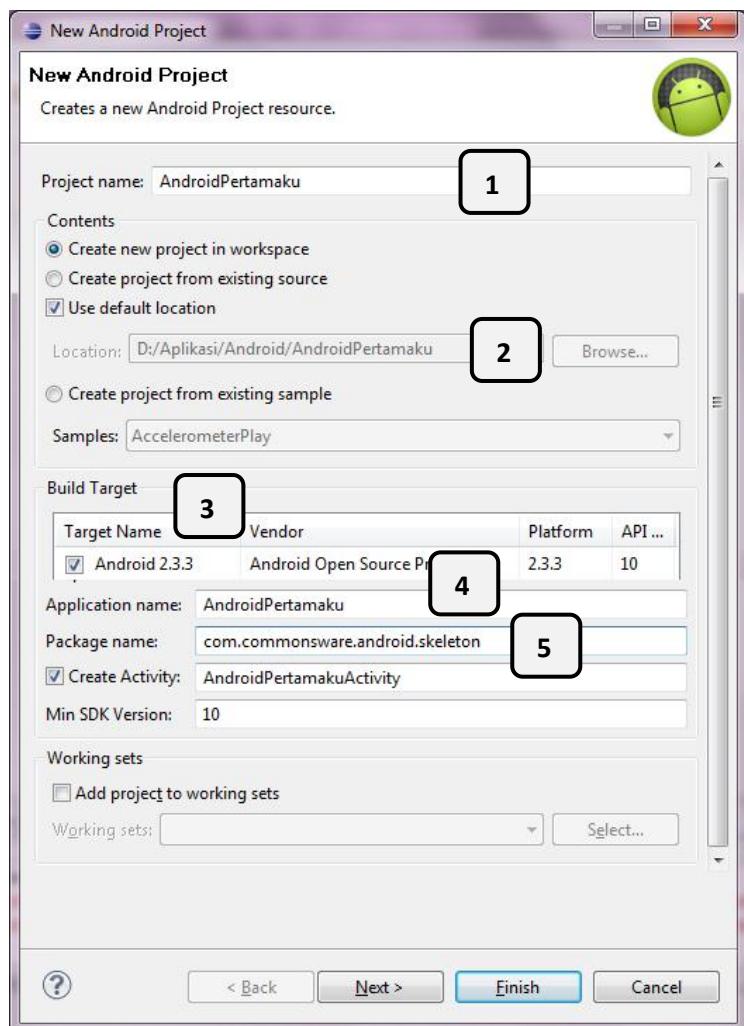
Dari menu utama program Eclipse pilih menu File | New | Project



**Gambar 3.1 Membuat Project Baru**



**Gambar 3.2 Pilih folder Android Project kemudian klik tombol Next**



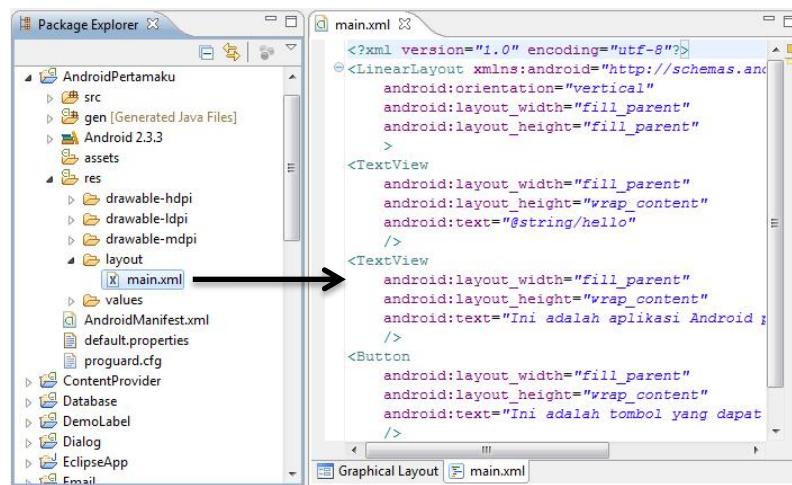
**Gambar 3.3 Klik Finish untuk mengakhiri sekaligus membuat project Android baru**

1	Project Name	Beri nama project, misalnya: <b>AndroidPertamaku</b>
2	Location	Letak file project yang sedang dibuat
3	Build Target	Target hasil kompilasi project yang sedang dibuat

4	Application name	Nama Aplikasi, misalnya: <b>AndroidPertamaku</b>
5	Package name	Nama paket Java yang dituju, misalnya: <b>com.commonsware.android.skeleton</b>

Lakukan sedikit perubahan pada file main.xml di dalam folder **res/layout/main.xml**, seperti berikut ini:

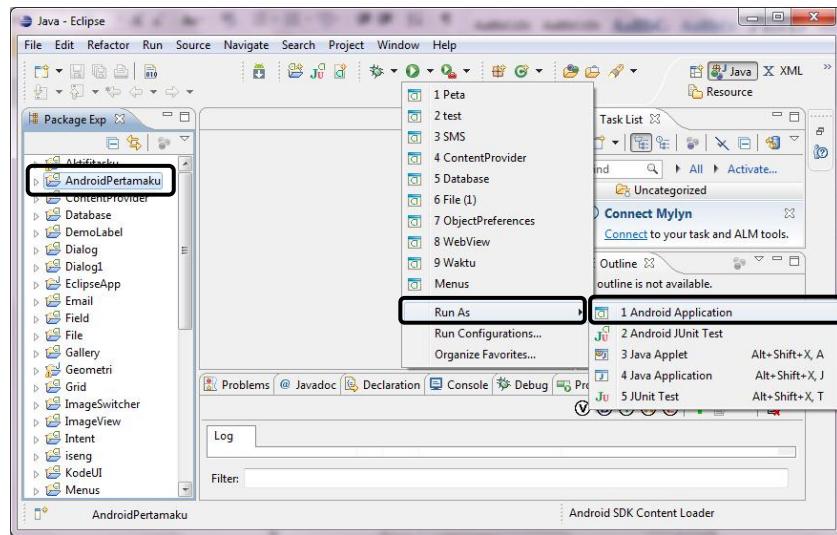
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Ini adalah aplikasi Android pertamaku!!" />
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Ini adalah tombol yang dapat diklik!" />
</LinearLayout>
```



**Gambar 3.4 kode XMLdi dalam file main.xml**

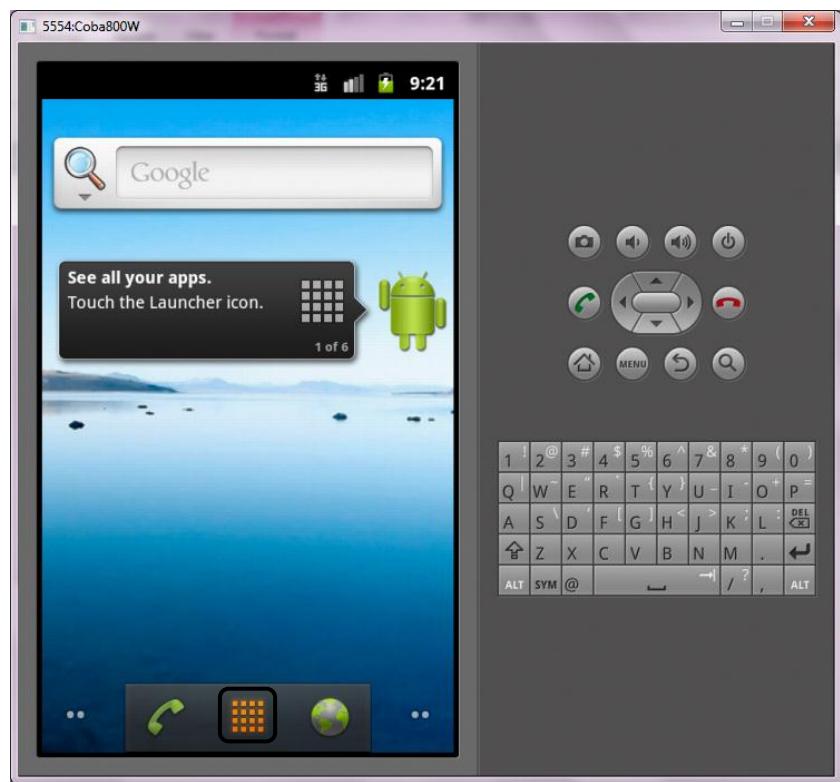
Langkah selanjutnya adalah mengkompilasi project (Built dan Run).

Pilih project bersangkutan yang terletak dalam pane Package Explorer, kemudian klik tombol **Play | Run As | Android Application** pada toolbar Eclipse.

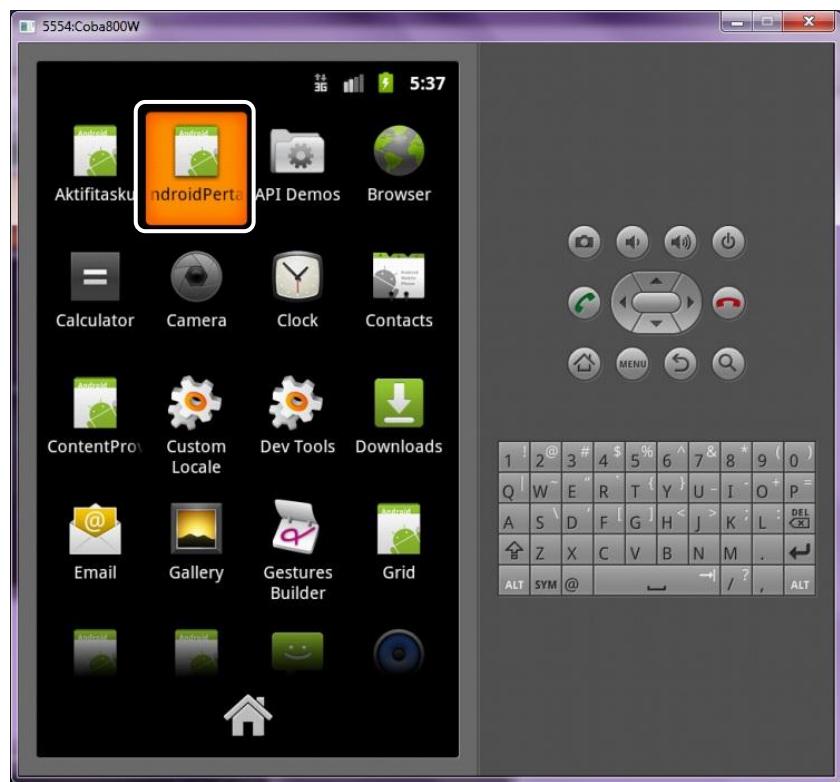


**Gambar 3.5 Langkah kompilasi project Android**

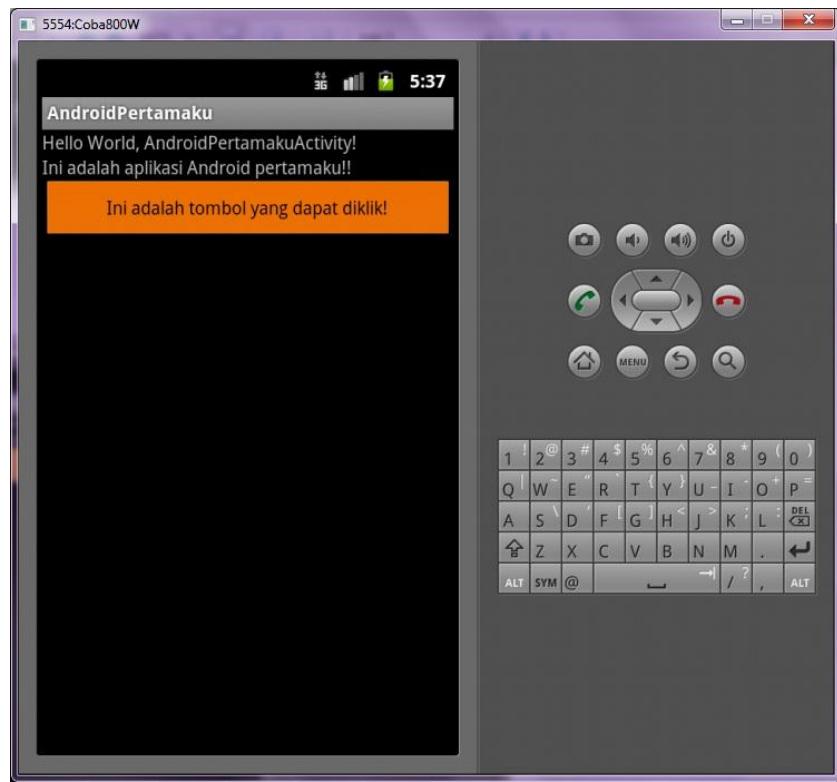
Adapun hasilnya kurang lebih akan tampak seperti:



Gambar 3.6 Klik pada icon Application Launcher



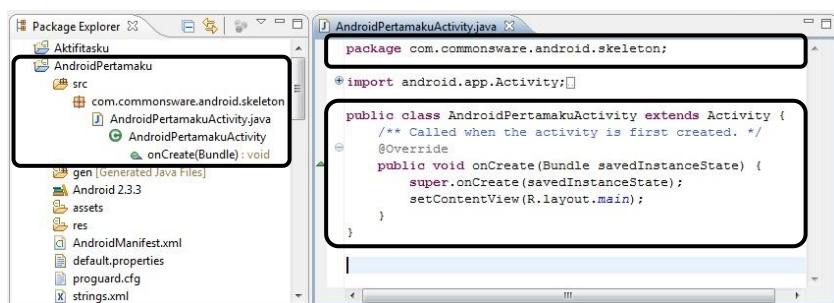
Gambar 3.7 Klik pada icon *AndroidPertamaku*



Gambar 3.8 Hasil akhir aplikasi

### 3.2 Struktur Direktori Project

Ketika project terbentuk maka secara otomatis terbentuk pula class activity utama (Main Activity) untuk sebuah aplikasi. Anda dapat menemukannya di dalam folder src/ yang telah disertai dengan namespace, class dan subclass yang mewakili activity utama.



**Gambar 3.9 Struktur directory project**

```
package com.commonware.android.skeleton;

import android.app.Activity;
import android.os.Bundle;

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

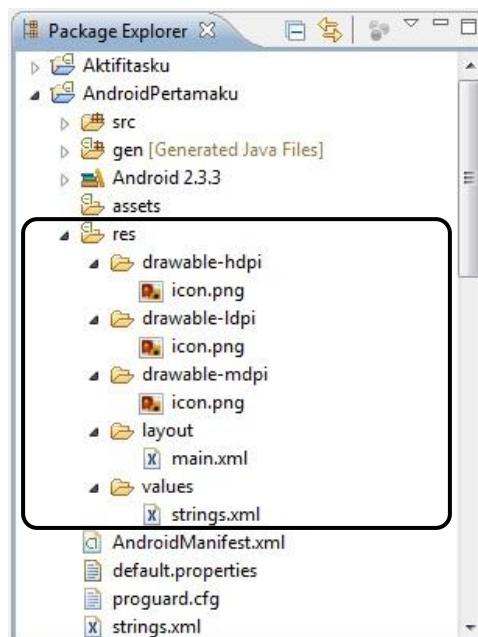
Anda bisa memodifikasi kode program yang ada di dalamnya, termasuk menambahkan file program baru ke dalam folder src/ untuk mendukung suatu aplikasi jika diperlukan.

Ketika project dikompilasi maka proses build Android akan menciptakan R.java yang berisi beberapa kode ketetapan nilai terkait dengan berbagai jenis sumberdaya dan Anda tidak perlu mengubahnya, biarkan tool Android menangani file tersebut untuk Anda. File R.java terletak di dalam folder gen/.

Sekarang perhatikan folder res/ yang berisi beberapa sumberdaya file, terdiri dari:

- **res/drawable/** untuk menempatkan file-file gambar (PNG, JPEG)

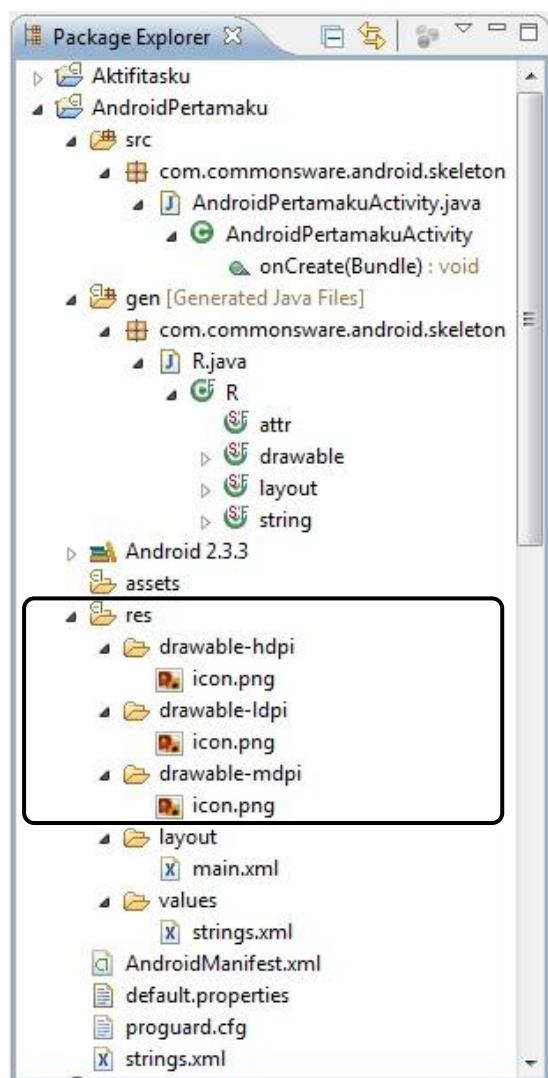
- **res/layout/** untuk menempatkan file-file XML yang berisi spesifikasi layout antarmuka.
- **res/menu/** untuk menempatkan file-file XML yang berisi spesifikasi menu.
- **res/raw/** untuk menempatkan file-file umum (misalnya: potongan file audio, file CSV untuk informasi akun)
- **res/values/** untuk menempatkan nilai-nilai string.
- **res/xml/** untuk menempatkan file-file XML bersifat umum yang akan digunakan oleh aplikasi.



**Gambar 3.10 Folder sumberdaya res/**

Sebagian folder memiliki akhiran kata berbeda, contoh: `res/drawable-hdpi/`. Seperti namanya, folder tersebut menandakan bahwa folder

berisi sumberdaya gambar yang akan diterapkan pada perangkat dengan resolusi layar kerapatan tinggi. Secara default folder drawable terdiri dari drawable-hdpi (high density), drawable-ldpi (low density), drawable-mdpi (middle density).



Gambar 3.11 Struktur lengkap project Android

Struktur sistem Android terorganisasi dalam bentuk direktori pohon (tree directory), mirip dengan struktur direktori pada project Java. Setiap direktori pada project Android memiliki maksud dan tujuan khusus. Direktori tersebut antara lain:

- **src**, berisi file-file .java project Android. Pada contoh di atas terdapat satu file yaitu **AndroidPertamakuActivity.java**, file tersebut merupakan sumberdaya activity dimana kode program akan dituliskan.
- **Android 2.3.3 Library**, di dalamnya terdapat satu file yaitu **android.jar** yang berisi semua pustaka class yang dibutuhkan oleh aplikasi Android.
- **gen**, berisi file R.java, merupakan file hasil dari *generate compiler*, merujuk pada semua sumberdaya yang terdapat dalam project. Anda tidak perlu memodifikasi file ini.
- **assets**, berisi semua modal yang digunakan oleh aplikasi seperti HTML, file teks, database dan lain-lain.
- **res**, berisi sumberdaya yang digunakan oleh aplikasi, juga berisi beberapa sub folder: drawable-<resolusi>, layout dan values.
- **AndroidManifest.xml**, berisi kode-kode untuk menentukan izin yang diperlukan oleh aplikasi.

File main.xml menentukan antarmuka user untuk activity. Perhatikan baris kode yang berhuruf tebal

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

@string mengacu ke file strings.xml yang terletak di dalam folder res/values. Oleh karena itu **@string/hello** mengacu ke string hello yang

terdapat di dalam file strings.xml, berisi "Hello World, AndroidPertamakuActivity!" :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World,
AndroidPertamakuActivity!</string>
    <string name="app_name">AndroidPertamaku</string>
</resources>
```

Disarankan untuk menyimpan ketetapan nilai (konstanta) pada aplikasi ke dalam file strings.xml dan mengacunya dengan menggunakan identifier @string.

Dasar dari aplikasi Android adalah file manifest (AndroidManifest.xml). Anda bisa mendeklarasikan activity, service, dan lain-lain dalam aplikasi melalui file manifest. Anda juga bisa menugaskan bagaimana bagian-bagian yang menyatu padanya berpengaruh terhadap keseluruhan sistem Android, sebagai contoh, Anda akan menugaskan suatu activity yang harus muncul pada menu utama (menu launcher Android).

Android Manifest terbentuk secara otomatis, namun kita masih bisa dimodifikasi sesuai kebutuhan. Untuk aplikasi sederhana yang hanya ada satu activity, manifest yang bersifat *auto-generated* sudah bisa bekerja dengan baik.

Perhatikan file AndroidManifest.xml berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.commonsware.android.skeleton"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />

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

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

File AndroidManifest.xml berisi rincian informasi mengenai aplikasi, antara lain:

- Mendefinisikan nama package aplikasi. Pada kasus di atas adalah com.commonsware.android.skeleton.
- Menginformasikan kode versi aplikasi yang dibuat. Pada kasus di atas adalah versi 1, dimana setiap kali terjadi update, kode versi harusnya akan ditambah.
- Menginformasikan nama versi. Pada kasus di atas adalah 1.0. Nama versi pada umumnya berformat <major>.<minor>.<point>.
- Menginformasikan bahwa aplikasi menggunakan gambar bernama icon.png yang terletak di dalam folder drawable.
- Menginformasikan nama aplikasi, diwakili oleh string app\_nama yang telah ditetapkan dalam file strings.xml.
- Menginformasikan activity, dimana hanya terdapat satu activity di dalam aplikasi, diwakili oleh file AndroidPertamakuActivity.java. Label activity yang tampil, sama dengan nama aplikasinya.
- Di dalam pendefinisan activity terdapat elemen <intent-filter>:
  - ❖ Action intent-filter bernama android.intent.action.MAIN, menandakan bahwa activity ini sebagai titik awal masuk bagi aplikasi.
  - ❖ Category intent-filter bernama android.intent.category.LAUNCHER, menandakan bahwa aplikasi dapat dipanggil dari icon Launcher.

- Atribut dari elemen <uses-sdk> adalah android:minSdkVersion, menentukan versi minimal sistem operasi yang dapat menjalankan aplikasi.

Jika diperhatikan kembali bahwa Eclipse secara otomatis juga akan men-generate file R.java yang kodennya tampak seperti berikut:

```
package com.commonsware.android.skeleton;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```

Kode di dalam File R.java tidak perlu diubah karena Eclipse membentuknya secara otomatis ketika dilakukan perubahan terhadap project kemudian menjalankannya (built-run).

Kode program yang menghubungkan activity ke antarmuka (main.xml) adalah method **setContentView()** yang berada di dalam file AndroidPertamakuActivity.java:

```
package com.commonsware.android.skeleton;

import android.app.Activity;
import android.os.Bundle;

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

R.layout.main mengacu ke file main.xml yang terletak di dalam folder res/layout. Anda dapat menambahkan file-file XML di folder res/layout. Method onCreate() terpicu ketika terjadi load activity.

## BAB 4

# ACTIVITY DAN INTENT

Pada bab sebelumnya telah dipelajari bahwa activity adalah window yang berisi antarmuka antara pengguna dengan aplikasi. Aplikasi bisa memiliki satu activity atau tidak sama sekali. Namun pada umumnya aplikasi memiliki satu atau lebih activity. Tujuan utama sebuah activity adalah untuk berinteraksi dengan pengguna. Ketika activity tampak pada layar maka pada saat itu pula akan melewati beberapa tingkatan langkah (meskipun tersembunyi) yang dikenal sebagai siklus hidup activity.

Pemahaman terhadap siklus hidup activity sangatlah penting untuk memastikan bahwa aplikasi yang dibuat bekerja dengan benar. Pada bab ini akan dipelajari lebih rinci bagaimana activity bekerja.

Bagian dari activity yang mempunyai konsep unik lainnya adalah intent. Intent merupakan perekat yang dapat mengaktifkan activity-activity berbeda dari aplikasi-aplikasi berbeda untuk bekerja secara bersama, dan memastikan bahwa tugas-tugas dapat dilaksanakan seolah-olah activity-activity tersebut hanya dimiliki oleh satu aplikasi.

### 4.1 Memahami Activity

Untuk menciptakan Activity adalah dengan menciptakan class Java yang meng-*extends* class dasar Activity:

```
package com.commonsware.android.skeleton;

import android.app.Activity;
import android.os.Bundle;

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

```
    }  
}
```

Class activity akan di-load pada setiap komponen UI (user interface/antarmuka user) dengan menggunakan file XML yang telah didefinisikan di dalam folder res/layout.

Dalam kasus ini, Anda akan me-load UI dari main.xml:

```
setContentView(R.layout.main);
```

Setiap activity dalam aplikasi seharusnya dideklarasikan terlebih dahulu pada file AndroidManifest.xml, seperti berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest  
    xmlns:android="http://schemas.android.com/apk/res/android"  
        package="com.commonsware.android.skeleton"  
        android:versionCode="1"  
        android:versionName="1.0">  
        <uses-sdk android:minSdkVersion="10" />  
        <application android:icon="@drawable/icon"  
            android:label="@string/app_name">  
            <activity android:name=".AndroidPertamakuActivity"  
                android:label="@string/app_name">  
                <intent-filter>  
                    <action android:name="android.intent.action.MAIN" />  
                    <category  
                        android:name="android.intent.category.LAUNCHER" />  
                </intent-filter>  
            </activity>  
        </application>  
</manifest>
```

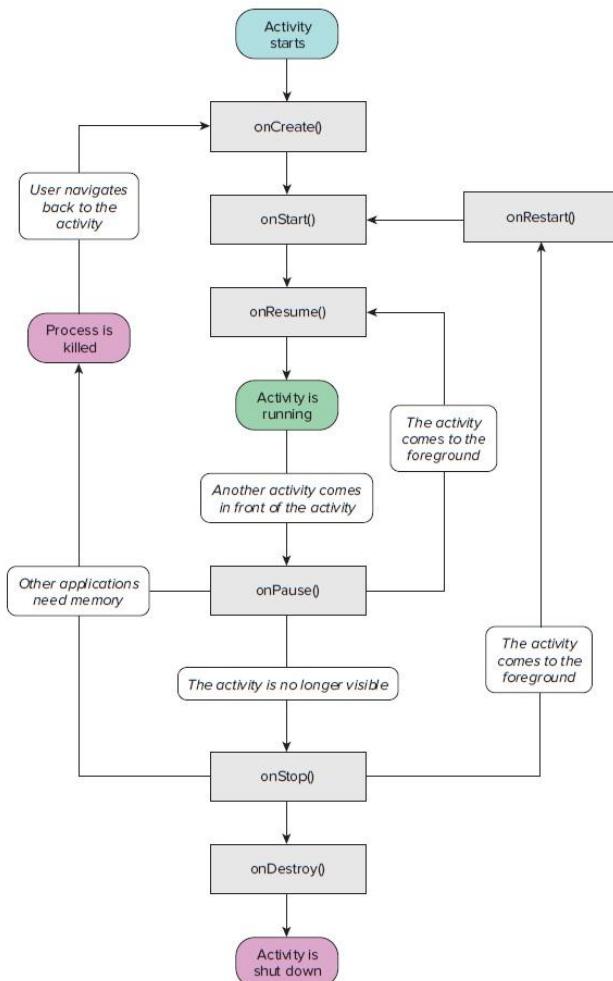
Class dasar Activity mendefinisikan serangkaian kejadian/events yang menentukan siklus hidup activity. Class activity mendefinisikan kejadian-kejadian berikut ini:

- **onCreate()**, terpanggil ketika activity dibuat untuk pertamakalinya.
- **onStart()**, terpanggil ketika activity tampak oleh pengguna.
- **onResume()**, terpanggil ketika activity mulai berinteraksi dengan pengguna.

- **onPause()**, terpanggil ketika activity yang aktif saat ini dihentikan sementara dan activity sebelumnya dilanjutkan kembali.
- **onStop()**, terpanggil ketika activity sudah tidak tampak lagi oleh pengguna.
- **onDestroy()**, terpanggil ketika activity dihapus oleh system, baik dilakukan secara manual maupun oleh system untuk menghemat memori.
- **onRestart()**, terpanggil ketika activity telah dihentikan dan dilakukan restart ulang.

Secara default activity yang terbentuk berisi event onCreate. Di dalamnya terdapat kode penanganan event yang membantu menampilkan elemen UI ke layar.

Berikut ini siklus hidup activity dan berbagai jenis tingkatan yang melewatkinya, mulai dari activity berawal sampai berakhir.



**Gambar 4.1 Siklus hidup Activity**

Sekarang cobalah buat project baru bernama “**Aktifitasku**”. Melalui Eclipse, klik menu **File | New | Android Project**. Kemudian isikan beberapa textbox dengan pengaturan berikut ini:

Project name	<b>Aktifitasku</b>
Build Target	<b>Android 2.3.3, API 10</b>
Application name	<b>Aktifitasku</b>

Package name	<b>net.belajar2android.Activity</b>
Create Activity	<b>AktifitaskuActivity</b>

Pada file **AktifitaskuActivity.java** tambahkan beberapa kode pernyataan yang berhuruf tebal berikut ini:

```
package net.belajar2android.Activity;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

public class AktifitaskuActivity extends Activity {
    String tag = "Events";
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Log.d(tag, "Dalam event onCreate()");
    }

    public void onStart()
    {
        super.onStart();
        Log.d(tag, "Dalam event onStart()");
    }

    public void onRestart()
    {
        super.onRestart();
        Log.d(tag, "Dalam event onRestart()");
    }
    public void onResume()
    {
        super.onResume();
        Log.d(tag, "Dalam event onResume()");
    }
    public void onPause()
    {
        super.onPause();
        Log.d(tag, "Dalam event onPause()");
    }
    public void onStop()
    {
        super.onStop();
        Log.d(tag, "Dalam event onStop()");
    }

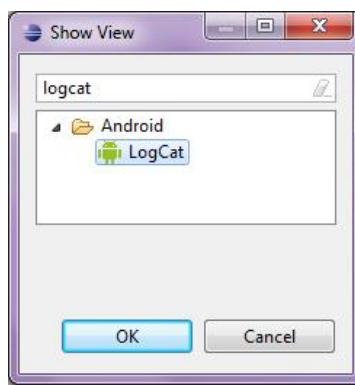
    public void onDestroy()
    {
```

```

        super.onDestroy();
        Log.d(tag, "Dalam event onDestroy()");
    }
}

```

Untuk menguji program di atas, Anda harus mengaktifkan Window **LogCat** untuk mengetahui hasil dari debugger-nya. Caranya klik menu Eclipse **Window | Show View | Other**, kemudian tuliskan logcat pada textbox seperti tampak gambar di bawah ini:



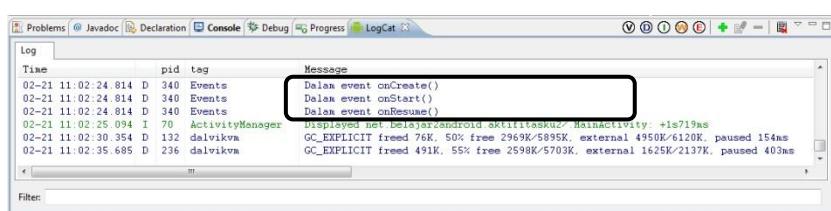
**Gambar 4.2 Show View LogCat**

Selanjutnya tekan tombol keyboard F11. Ketika activity dijalankan untuk pertamakalinya maka akan tampil debugger seperti ini:

```

02-21 11:02:24.814: DEBUG/Events(340): Dalam event onCreate()
02-21 11:02:24.814: DEBUG/Events(340): Dalam event onStart()
02-21 11:02:24.814: DEBUG/Events(340): Dalam event onResume()

```



**Gambar 4.3 Debugger Logcat**

Ketika tombol **Back** pada emulator ( ) ditekan, maka hasilnya adalah:

```
06-18 13:53:18.336: DEBUG/Events(523): Dalam event onPause()
06-18 13:53:19.196: DEBUG/Events(523): Dalam event onStop()
06-18 13:53:19.206: DEBUG/Events(523): Dalam event onDestroy()
```



Klik tombol **Home** () dan tahan dulu, kemudian klik ikon activity.

Perhatikan hasil debug-nya:

```
06-18 14:03:20.766: DEBUG/Events(523): Dalam event onCreate()
06-18 14:03:20.766: DEBUG/Events(523): Dalam event onStart()
06-18 14:03:20.776: DEBUG/Events(523): Dalam event onResume()
```

Tekan tombol **Phone** pada emulator Android () hingga tampak activity phone background. Perhatikan hasil debug-nya, maka akan tampak seperti berikut:

```
06-18 14:39:43.256: DEBUG/Events(569): Dalam event onPause()
06-18 14:39:44.316: DEBUG/Events(569): Dalam event onStop()
```

Sebagai catatan bahwa event `onDestroy()` tidak terpanggil, menandakan bahwa activity masih ada di dalam memori. Sekarang keluarlah dari phone dialer dengan cara klik tombol **Back**, maka Activity sekarang tampak lagi.

```
06-18 14:45:49.585: DEBUG/Events(569): Dalam event onRestart()
06-18 14:45:49.585: DEBUG/Events(569): Dalam event onStart()
06-18 14:45:49.595: DEBUG/Events(569): Dalam event onResume()
```

Seperti yang tampak pada contoh di atas, sebuah activity akan dihapus (`destroy`) ketika Anda menekan tombol Back. Ini penting sekali untuk dipahami, karena pada kondisi apa pun, activity yang aktif saat ini akan hilang, oleh karena itu Anda perlu menulis kode tambahan terhadap activity untuk melindunginya saat activity dihapus.

## 4.2 Menerapkan Style & Themes Ke Activity

Meskipun secara default activity menempati keseluruhan layar, namun Anda masih bisa mengaplikasikan *dialog theme* ke activity yang ditampilkan sebagai *floating dialog*. Sebagai contoh, Anda mungkin ingin

mengkustomisasi activity untuk ditampilkan sebagai pop-up untuk mengingatkan user tentang beberapa tindakan yang sedang dilaksanakan.

Untuk menerapkan dialog theme ke activity diperlukan modifikasi elemen <Activity> di dalam file AndroidManifest.xml dengan menambahkan atribut **android:theme**:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.belajar2android.Activity"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".AktifitaskuActivity"
            android:label="@string/app_name"
            android:theme="@android:style/Theme.Dialog">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```



Gambar 4.4 Dialog theme disertai judul dialog

## 4.3 Menyembunyikan Judul Activity

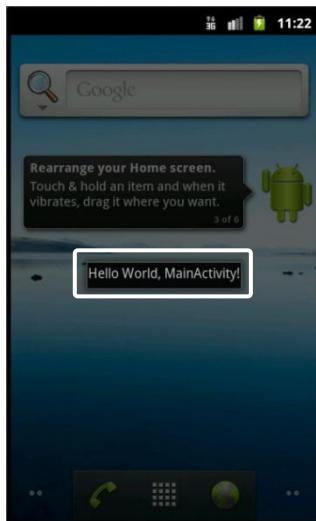
Anda juga dapat menyembunyikan judul activity jika dinginkan, misalnya hanya ingin menampilkan status update ke user.

Gunakan method `requestWindowFeature()` dan melewatkannya ke konstanta `Window.FEATURE_NO_TITLE`, seperti berikut:

```
package net.belajar2android.Activity;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.Window;

public class AktifitaskuActivity extends Activity {
    String tag = "Events";
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.main);
        Log.d(tag, "Dalam event onCreate()");
    }
}
```



Gambar 4.5 Dialog theme tanpa disertai judul dialog

## 4.4 Menampilkan Window Dialog

Berikut ini adalah cara menampilkan window dialog untuk mendapatkan konfirmasi dari user dengan meng-*override* method protected `onCreateDialog()` yang telah didefinisikan di dalam class Main Activity untuk menampilkan window.

Buat project baru bernama **Dialog**, klik menu Eclipse **File | New | Android project**. Isikan beberapa textbox pada form New Android Project seperti berikut ini:

<b>Project name</b>	Dialog
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	Dialog
<b>Package name</b>	net.belajar2android.Dialog
<b>Create Activity</b>	DialogActivity

Ubahlah pernyataan XML yang berhuruf tebal pada file main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
    <Button
        android:id="@+id/btn_dialog"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Klik untuk menampilkan dialog" />
</LinearLayout>
```

Tambahkan pernyataan yang berhuruf tebal di dalam file `DialogActivity.java`

```
package net.belajar2android.Dialog;

import android.app.Activity;
import android.os.Bundle;
import android.app.AlertDialog;
```

```
import android.app.Dialog;
import android.content.DialogInterface;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class DialogActivity extends Activity {
    CharSequence[] items = {"Android", "iPhone", "BlackBerry"};
    boolean[] itemsChecked = new boolean [items.length];

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

        Button btn = (Button) findViewById(R.id.btn_dialog);
        btn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                showDialog(0);
            }
        });
    }

    @Override
    protected Dialog onCreateDialog(int id){
        switch (id){
        case 0:
            return new AlertDialog.Builder(this)
                .setIcon(R.drawable.icon)
                .setTitle("Dialog dengan beberapa teks sederhana...")
                .setPositiveButton("OK", new DialogInterface.OnClickListener()
                {
                    public void onClick(DialogInterface dialog, int whichButton)
                    {
                        Toast.makeText(getApplicationContext(), "OK diklik",
                        Toast.LENGTH_SHORT).show();
                    }
                })
                .setNegativeButton("Batal", new
                DialogInterface.OnClickListener()
                {

                    public void onClick(DialogInterface dialog, int whichButton)
                    {
                        Toast.makeText(getApplicationContext(), "Batal diklik!",
                        Toast.LENGTH_SHORT).show();
                    }
                })
                .setMultiChoiceItems(items, itemsChecked, new
                DialogInterface.OnMultiChoiceClickListener()
                {
                    @Override
```

```

public void onClick(DialogInterface dialog, int which, boolean
isChecked)
{
    Toast.makeText(getApplicationContext(), items[which] + (isChecked ?
" dicentang! ":" tidak dicentang!"), Toast.LENGTH_SHORT).show();
}
})
.create();
}
return null;
}
}

```

Tekan tombol keyboard F11 untuk menjalankan aplikasi. Klik tombol “Klik untuk menampilkan dialog” untuk menampilkan dialog. Beri tanda centang pada CheckBox yang tersedia, maka class Toast akan menampilkan teks dari item yang dipilih.

Untuk mengakhiri dialog, klik tombol OK atau Batal.



**Gambar 4.6 Window Dialog**

Bagaimana program bekerja? Untuk menampilkan dialog, pertama Anda harus meng-override method **onCreateDialog()** class Activity:

```
@Override  
protected Dialog onCreateDialog(int id)  
{  
    //...  
}
```

Method ini dipanggil ketika memanggil method **showDialog()**:

```
Button btn = (Button) findViewById(R.id.btn_dialog);  
btn.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v){  
        showDialog(0);  
    }  
});
```

Method **onCreateDialog()** adalah *callback* untuk menciptakan dialog yang diatur oleh activity saat memanggil method **showDialog**, callback ini akan diminta. Method **showDialog()** menerima argument integer yang mengidentifikasi dialog tertentu untuk ditampilkan.

Untuk menciptakan dialog, digunakan class konstruktor **AlertDialog** Builder. Kemudian Anda mengeset property seperti icon, title, button dan checkbox:

```
@Override  
protected Dialog onCreateDialog(int id)  
{  
    switch (id){  
    case 0:  
        return new AlertDialog.Builder(this)  
            .setIcon(R.drawable.icon)  
            .setTitle("Dialog dengan beberapa teks sederhana...")  
            .setPositiveButton("OK", new  
DialogInterface.OnClickListener()  
            {  
                public void onClick(DialogInterface dialog, int  
whichButton)  
                {  
                    Toast.makeText(getApplicationContext(), "OK diklik",  
Toast.LENGTH_SHORT).show();  
                }  
            })  
            .setNegativeButton("Batal", new  
DialogInterface.OnClickListener()
```

```

    {
        public void onClick(DialogInterface dialog, int
whichButton)
        {
            Toast.makeText(getApplicationContext(), "Batal diklik!",
Toast.LENGTH_SHORT).show();
        }
    })

.setMultiChoiceItems(items, itemsChecked, new
DialogInterface.OnMultiChoiceClickListener()
{
    @Override
    public void onClick(DialogInterface dialog, int which,
boolean isChecked)
    {
        Toast.makeText(getApplicationContext(), items[which] +
(isChecked ? " dicentang!":" tidak dicentang!"),
Toast.LENGTH_SHORT).show();
    }
})
.create();
}
return null;
}

```

Pada kode di atas terdapat dua button: **OK** dan **Batal**, menggunakan method **setPositiveButton()** dan **setNegativeButton()**. Anda juga mengeset daftar checkbox agar user bisa memilihnya melalui method **setMultiChoiceltems()**. Pada method **setMultiChoiceltems()** Anda dapat melewatkna dua array: pertama untuk daftar item yang akan ditampilkan dan yang lainnya berisi nilai setiap item yang menandakan jika checkbox dicentang. Saat item diklik digunakan class **Toast** untuk menampilkan pesan.

**Catatan :**

Di dalam Android akan sering ditemui class **Context** dan instansiasinya. Instansiasi class Context sering digunakan untuk menghasilkan rujukan ke aplikasi. Seperti contoh di atas, parameter pertama dari class **Toast** mengambil object Context.

```

return new AlertDialog.Builder(this)
.setIcon(R.drawable.icon)
.setTitle("Dialog dengan beberapa teks sederhana...")

```

```
.setPositiveButton("OK", new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int whichButton)
    {
        Toast.makeText(getApplicationContext(), "OK diklik",
        Toast.LENGTH_SHORT).show();
    }
})
```

Karena class Toast() tidak bisa digunakan secara langsung di dalam activity (digunakan oleh class AlertDialog), Anda perlu mengembalikan instance class Context melalui method getBaseContext().

Context Class akan sering ditemui pula ketika menciptakan tampilan secara dinamis di dalam activity. Contoh, Mungkin Anda ingin menciptakan tampilan TextView melalui kode program. Untuk melakukannya Anda perlu membuat instansiasi class TextView, seperti berikut:

```
TextView tv = new TextView(this);
```

Konstruktor untuk class TextView mengambil sebuah Context object dan karena class Activity adalah subclass dari Context, maka Anda bisa menggunakan kata kunci **this** untuk mewakili Context Object.

## 4.5 Menampilkan Progress Dialog

Progress Dialog berguna untuk menampilkan kemajuan suatu proses, seperti status proses download. Ikuti langkah-langkah di bawah ini untuk membuat Progress Dialog.

Gunakan project sebelumnya (project **Dialog**), tambahkan kode program yang berhuruf tebal di dalam file DialogActivity.java:

```
package net.belajar2android.Dialog;

import android.app.Activity;
import android.os.Bundle;
import android.app.AlertDialog;
```

```
import android.app.Dialog;
import android.content.DialogInterface;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import android.app.ProgressDialog;
import android.os.Handler;
import android.os.Message;

public class DialogActivity extends Activity {
    CharSequence[] items = {"Android", "iPhone", "BlackBerry"};
    boolean[] itemsChecked = new boolean [items.length];

    private ProgressDialog _progressDialog;
    private int _progress = 0;
    private Handler _progressHandler;

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

        Button btn = (Button) findViewById(R.id.btn_dialog);
        btn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                showDialog(1);
                _progress = 0;
                _progressDialog.setProgress(0);
                _progressHandler.sendEmptyMessage(0);
            }
        });

        _progressHandler = new Handler() {
            public void handleMessage(Message msg) {
                super.handleMessage(msg);
                if (_progress >= 100) {
                    _progressDialog.dismiss();
                } else {
                    _progress++;
                    _progressDialog.incrementProgressBy(1);
                    _progressHandler.sendMessageDelayed(0, 100);
                }
            }
        };
    }

    @Override
    protected Dialog onCreateDialog(int id){
        switch (id){
            case 0:
                return new AlertDialog.Builder(this)
                    .setIcon(R.drawable.icon)
                    .setTitle("Dialog dengan beberapa teks sederhana... ")
        }
    }
}
```

```

        .setPositiveButton("OK", new
DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int
whichButton)
    {
        Toast.makeText(getApplicationContext(), "OK diklik",
Toast.LENGTH_SHORT).show();
    }
})

.setNegativeButton("Batal", new
DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int
whichButton)
    {
        Toast.makeText(getApplicationContext(), "Batal diklik!",
Toast.LENGTH_SHORT).show();
    }
})

.setMultiChoiceItems(items, itemsChecked, new
DialogInterface.OnMultiChoiceClickListener()
{
    @Override
    public void onClick(DialogInterface dialog, int which,
boolean isChecked)
    {
        Toast.makeText(getApplicationContext(), items[which] +
(isChecked ? " dicentang!":" tidak dicentang!"),
Toast.LENGTH_SHORT).show();
    }
})
.create();
case 1:
    progressDialog = new ProgressDialog(this);
    progressDialog.setIcon(R.drawable.icon);
    progressDialog.setTitle("Proses download file...");

    progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL
);
    progressDialog.setButton(DialogInterface.BUTTON_POSITIVE,
"Sembunyikan", new
    DialogInterface.OnClickListener(){
        public void onClick(DialogInterface dialog, int
whichButton)
        {
            Toast.makeText(getApplicationContext(), "Sembunyikan Klik",
Toast.LENGTH_SHORT).show();
        }
    });
    progressDialog.setButton(DialogInterface.BUTTON_NEGATIVE,
"Batal", new DialogInterface.OnClickListener() {

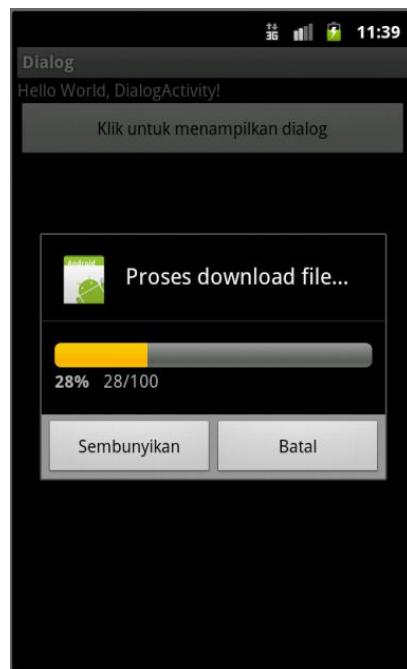
```

```

        public void onClick(DialogInterface dialog, int
whichButton)
{
    Toast.makeText(getApplicationContext(), "Batal diklik",
Toast.LENGTH_SHORT).show();
}
}
);
return _progressDialog;
}
return null;
}
}

```

Jalankan aplikasi melalui Emulator Android. Klik tombol “Klik untuk menampilkan dialog” untuk menampilkan progress dialog. Perhatikan progress bar akan menghitung nilai mulai 1 sampai 100.



*Gambar 4.7 Progress Bar*

Bagaimana program bekerja? Pertama diciptakan instance class **ProgressDialog** dan mengatur beberapa nilai property seperti icon, title dan style:

```
_progressDialog = new ProgressDialog(this);
_progressDialog.setIcon(R.drawable.icon);
_progressDialog.setTitle("Proses download file...");
_progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
);
```

Kemudian mengeset dua tombol yang akan tampil disamping progress dialog:

```
_progressDialog.setButton(DialogInterface.BUTTON_POSITIVE,
"Sembunyikan", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int whichButton)
    {
        Toast.makeText(getApplicationContext(), "Sembunyikan Klik",
Toast.LENGTH_SHORT).show();
    }
});

_progressDialog.setButton(DialogInterface.BUTTON_NEGATIVE,
"Batal", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int whichButton)
    {
        Toast.makeText(getApplicationContext(), "Batal diklik",
Toast.LENGTH_SHORT).show();
    }
});
return _progressDialog;
}
```

Untuk menampilkan status progress pada progress dialog, gunakan object Handler untuk menjalankan proses dibelakang layar (background thread):

```
_progress = 0;
_progressDialog.setProgress(0);
_progressHandler.sendEmptyMessage(0);
Background thread menghitung naik sampai mencapai 100, setiap
kenaikan diperlambat 100 mili detik:
_progressHandler = new Handler()
{
    public void handleMessage(Message msg)
    {
        super.handleMessage(msg);
        if (_progress >= 100) {
            progressDialog.dismiss();
        }
    }
};
```

```

        } else {
            _progress++;
            _progressDialog.incrementProgressBy(1);
            _progressHandler.sendMessageDelayed(0, 100);
        }
    };
}

```

Ketika perhitungan mencapai 100 maka progress dialog akan berakhir.

## 4.6 Menghubungkan Activity Dengan Intent

Aplikasi bisa berisi lebih dari satu activity atau tidak sama sekali. Ketika aplikasi memiliki lebih dari satu activity, maka perlu dikendalikan antara activity satu dengan activity lainnya. Di dalam Android, pengendalian antar beberapa activity melalui **intent**.

Ikuti langkah-langkah berikut ini yang menunjukkan bagaimana menambahkan activity lain dari activity yang sudah ada dan mengendalikan dua activity tersebut.

Gunakan project yang dibuat sebelumnya (project **Aktifitasku**), lakukan modifikasi terhadap file AndoridManifest.xml terutama yang berhuruf tebal dan hapus pernyataan yang bergaris tengah.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.belajar2android.Activity"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />

    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".AktifitaskuActivity"
            android:label="@string/app_name"
            android:theme="@android:style/Theme.Dialog" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

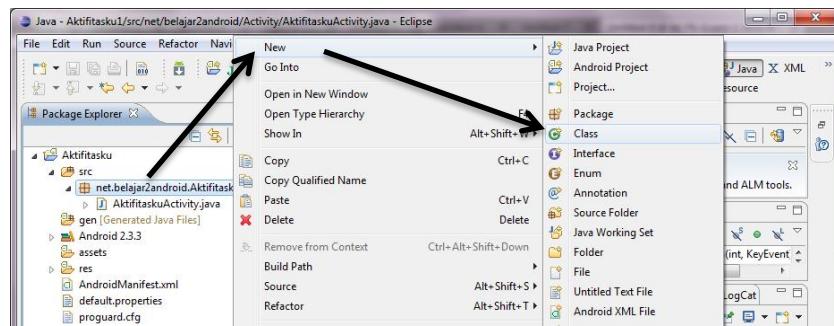
```

```

<activity android:name=".Activity2"
    android:label="Activity 2">
    <intent-filter>
        <action android:name="net.belajar2android.ACTIVITY2" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
</application>
</manifest>

```

Selanjutnya klik kanan pada file net.belajar2android.Activity yang berada di bawah folder src, kemudian pilih menu **New | Class**



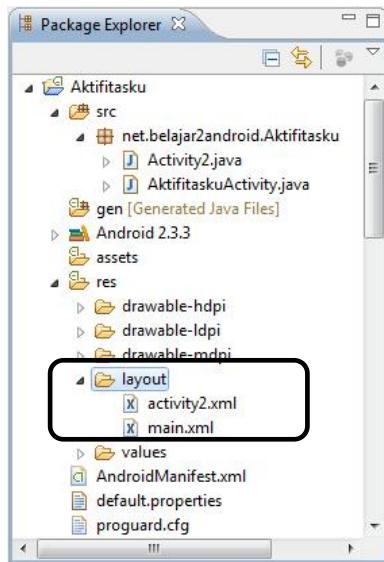
**Gambar 4.8 Membuat Class Tambahan**

Beri nama class dengan **Activity2**, kemudian klik tombol **Finish**.



**Gambar 4.9 Menciptakan Activity tambahan**

Buat salinan file main.xml dengan cara klik kanan pada file main.xml dan pilih **Copy**. Kemudian klik kanan pada direktori res/layout dan pilih **Paste**. Ubah nama filenya dengan **activity2.xml**. Sekarang direktori res/layout berisi activity2.xml.



Gambar 4.10 Menambahkan Layout Baru

Modifikasi file activity2.xml seperti berikut ini:

```
<?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="Ini adalah Activity 2!" />
</LinearLayout>
```

Selanjutnya tambahkan kode program yang ber huruf tebal di bawah ini ke dalam file **Activity2.java**:

```
package net.belajar2android.Activity;

import android.app.Activity;
import android.os.Bundle;

public class Activity2 extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity2);
    }
}
```

Ubah file AktifitaskuActivity.java menjadi seperti ini:

```
package net.belajar2android.Activity;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.Window;

import android.view.KeyEvent;
import android.content.Intent;

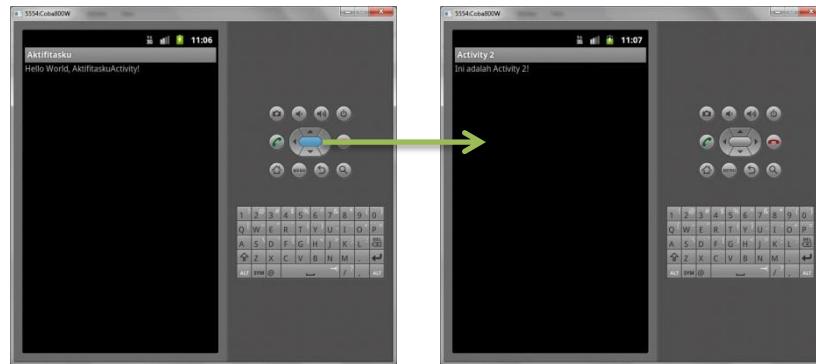
public class AktifitaskuActivity extends Activity {
    String tag = "Events";
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // --- Sembunyikan judul bar ---
        // requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.main);
        Log.d(tag, "Dalam event onCreate()");
    }

    public boolean onKeyDown(int keyCode, KeyEvent event)
    {
        if (keyCode == KeyEvent.KEYCODE_DPAD_CENTER)
        {
            startActivity(new Intent("net.belajar2android.ACTIVITY2"));
        }
        return false;
    }

    public void onStart(){ //... }
    public void onRestart(){ //... }
    public void onResume(){ //... }
    public void onPause(){ //... }
    public void onStop(){ //... }
    public void onDestroy(){ //... }
}
```

Jalankan aplikasi ke Emulator Android ( tekan tombol keyboard F11).

Saat activity pertama di-load, klik pada center pad/trackball, maka activity kedua akan di-load.



**Gambar 4.11 Pemanggilan Activity dari Activity yang lain**

Bagaimana program bekerja? Seperti yang sudah dipelajari sebelumnya, bahwa activity dibuat dari komponen UI (contoh: main.xml) dan komponen class (contoh: AktifitaskuActivity.java). Oleh karena itu jika ingin menambahkan activity lain di dalam project maka Anda harus menciptakan dua komponen tersebut.

Dalam file AndroidManifest.xml harus ditambahkan kode berikut:

```
<activity android:name=".Activity2"
    android:label="Activity 2">
    <intent-filter>
        <action android:name="net.belajar2android.ACTIVITY2" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

Di sini ditambahkan activity baru ke aplikasi.

Catatan yang perlu diperhatikan adalah sebagai berikut:

- Nama activity yang ditambahkan adalah “Activity2”.
- Label/judul activity dinamai “Activity 2”.
- Nama intent filter untuk activity adalah “net.belajar2android.ACTIVITY2”. Activity lain yang hendak

memanggil activity ini akan memintanya berdasarkan nama tersebut. Idealnya, harus dicadangkan nama domain perusahaan sebagai nama intent filter untuk mengurangi kemungkinan aplikasi lain memiliki nama intent filter yang sama.

- Kategori intent filter adalah “`android.intent.category.DEFAULT`”. Perlu ditambahkan pernyataan tersebut ke intent filter sehingga activity dapat dijalankan oleh activity lain dengan menggunakan method `startActivity()`.

Di dalam file `AktifitaskuActivity.java` diterapkan event handler `onKeyDown`. Event ini terpicu saat user menekan salah satu key pada perangkat Android, misalnya saat user menekan center key pada directional pad (diwakili sebagai konstanta `KeyEvent.KEYCODE_DPAD_CENTER`). Selanjutnya digunakan method `startActivity()` untuk menampilkan Activity2 melalui penciptaan instance class Intent dan melewatkannya ke intent filter bernama `Activity2 (net.belajar2android.ACTIVITY2)`:

```
public boolean onKeyDown(int keyCode, KeyEvent event)
{
    if (keyCode == KeyEvent.KEYCODE_DPAD_CENTER)
    {
        startActivity(new Intent("net.belajar2android.ACTIVITY2"));
        // startActivity(new Intent(this, Activity2.class));
    }
    return false;
}
```

Activity-activity di dalam Android bisa diminta oleh aplikasi mana pun yang sedang berjalan. Sebagai contoh, Anda dapat menciptakan project Android baru, kemudian menampilkan Activity2 melalui intent filter `net.belajar2andorid.ACTIVITY2`. Hal ini adalah salah satu konsep dasar pada Android yang bisa menjalankan sebuah aplikasi karena diminta oleh aplikasi lain.

Jika activity yang akan diminta telah didefinisikan dalam project yang sama, maka dapat ditulis ulang pernyataan seperti sebelumnya, yaitu:

```
startActivity(new Intent(this, Activity2.class));
```

Pendekatan ini dapat diterapkan ketika activity yang hendak ditampilkan masih dalam satu project yang sama dan salah satunya adalah activity yang sedang aktif saat ini.

## 4.7 Menangkap Hasil Intent

Method `startActivity()` adalah untuk memanggil activity lain tapi tidak mengembalikan suatu hasil ke activity yang aktif saat ini. Lalu bagaimana bila kita membutuhkan suatu hasil? contoh, mungkin Anda mempunyai activity yang meminta user untuk mengisi username dan password. Informasi dimasukkan oleh user dalam activity tersebut, hasilnya dikembalikan ke activity pemanggil untuk diproses lebih lanjut. Jika Anda perlu untuk mengambil data dari activity untuk diproses oleh activity lain, maka perlu method `startActivityForResult()`.

Gunakan project **Aktifitasku**, tambahkan beberapa pernyataan berikut ke dalam file main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Silakan masukkan nama" />
<EditText
    android:id="@+id/txt_username"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/btn_OK"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="OK" />
```

```
</LinearLayout>
```

Tambahkan pernyataan yang berhuruf tebal pada file Activity2.java:

```
package net.belajar2android.Activity;

import android.app.Activity;
import android.os.Bundle;

import android.content.Intent;
import android.net.Uri;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class Activity2 extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        //---mendapatkan tombol OK---
        Button btn = (Button) findViewById(R.id.btn_OK);

        //---event handler untuk tombol OK---
        btn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                Intent data = new Intent();

                //---mendapatkan view EditText
                EditText txt_username = (EditText)
                findViewById(R.id.txt_username);

                //--- set data untuk pass back---

                data.setData(Uri.parse(txt_username.getText().toString()));
                setResult(RESULT_OK, data);

                //---menutup activity---
                finish();
            }
        });
    }
}
```

Tambahkan pernyataan yang berhuruf tebal berikut ke dalam file AktifitaskuActivity.java:

```
package net.belajar2android.Activity;

import android.app.Activity;
```

```

import android.os.Bundle;
import android.util.Log;
import android.view.Window;

import android.view.KeyEvent;
import android.content.Intent;

import android.widget.Toast;

public class AktifitaskuActivity extends Activity {
    String tag = "Events";
    int request_Code = 1;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // --- Sembunyikan judul bar ---
        //requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity2);
        Log.d(tag, "Dalam event onCreate()");
    }

    public boolean onKeyDown(int keyCode, KeyEvent event)
    {
        if (keyCode == KeyEvent.KEYCODE_DPAD_CENTER)
        {
            startActivityForResult(new Intent(this, Activity2.class),
request_Code);
        }
        return false;
    }

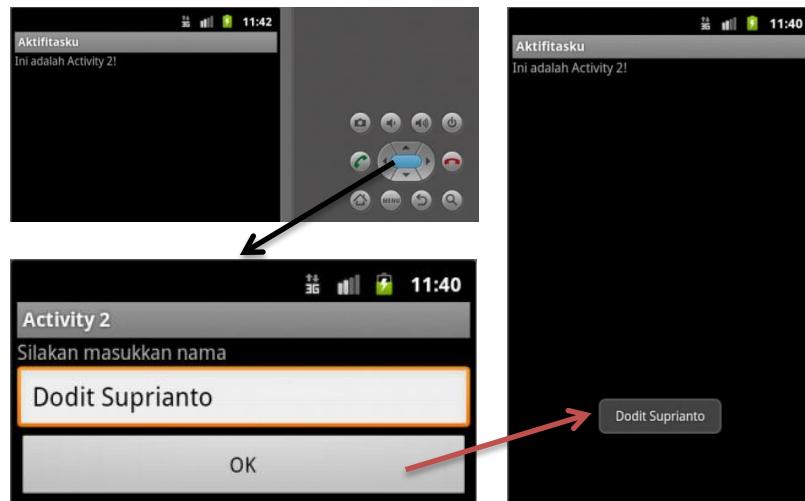
    public void onActivityResult(int requestCode, int resultCode,
Intent data)
    {
        if (requestCode == request_Code) {
            if (resultCode == RESULT_OK) {
                Toast.makeText(this, data.getData().toString(),
Toast.LENGTH_SHORT).show();
            }
        }
    }

    public void onStart() { //... }
    public void onRestart() { //... }
    public void onResume() { //... }
    public void onPause() { //... }
    public void onStop() { //... }
    public void onDestroy() { //... }
}

```

Jalankan aplikasi ke emulator Android. Ketika activity pertama di-load, klik tombol tengah dari directional pad. Activity2 sekarang akan di-load.

Tuliskan nama Anda dan klik tombol OK, maka sekarang activity menampilkan nama yang Anda tuliskan sebelumnya dengan menggunakan class Toast.



Gambar 4.12 Pemanggilan Activity disertai suatu nilai hasil

## 4.8 Mengirim Data Dengan Object Intent

Selain menangkap data dari activity, kita juga bisa mengirim data ke activity. Dari contoh sebelumnya mungkin ingin menampilkan teks default di dalam EditText sebelum activity ditampilkan (contoh: menampilkan tulisan "Nama Anda di sini" pada EditText). Oleh karena itu perlu ditambahkan object Intent untuk mengirim data ke activity yang menjadi sasaran.

Tambahkan pernyataan yang berhuruf tebal di bawah ini ke dalam file AktifitaskuActivity.java:

```
public boolean onKeyDown(int keyCode, KeyEvent event)
{
    if (keyCode == KeyEvent.KEYCODE_DPAD_CENTER)
    {
```

```

Intent i = new Intent("net.belajar2android.ACTIVITY2");
Bundle extras = new Bundle();
extras.putString("Nama", "Nama Anda di sini");
i.putExtras(extras);
startActivityForResult(i,1);
}
return false;
}

```

Tambahkan beberapa pernyataan berhuruf tebal di bawah ini ke dalam file Activity2.java:

```

public class Activity2 extends Activity {
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);

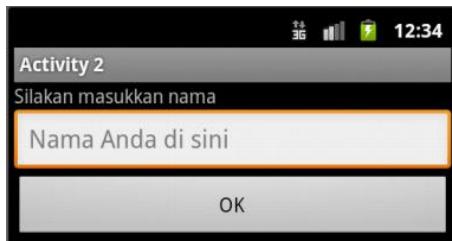
String defaultName = "";
Bundle extras = getIntent().getExtras();
if (extras!=null)
{
    defaultName = extras.getString("Nama");
}
//--- mendapatkan tampilan EditText ---
EditText txt_username = (EditText)
findViewById(R.id.txt_username);
txt_username.setHint(defaultName);

//---mendapatkan tombol OK---
Button btn = (Button) findViewById(R.id.btn_OK);

//---event handler untuk tombol OK---
btn.setOnClickListener(new View.OnClickListener()
{
    //...
});
}
}

```

Jalankan aplikasi seperti cara sebelumnya, maka hasilnya kurang lebih seperti berikut ini:



**Gambar 4.13 EditText disertai teks default**

Bagaimana program bekerja? Penggunaan object Intent untuk membawa data ke activity tujuan digunakan Bundle object:

```
Bundle extras = new Bundle();
extras.putString("Nama", "Nama Anda di sini");
i.putExtras(extras);
```

Bundle object pada dasarnya adalah object kamus yang dapat aktif dengan mengeset data dalam pasangan key/value. Disini diciptakan key bernama **Nama** yang bernilai "**Nama Anda di sini**". Bundle object kemudian menambahkan Intent object dengan menggunakan method `putExtras()`.

Dalam activity tujuan, gunakan method `getIntent()` untuk memperoleh intent yang berasal dari activity awal. Kemudian gunakan method `getExtras()` untuk memperoleh Bundle object:

```
Bundle extras = getIntent().getExtras();
if (extras!=null)
{
    defaultName = extras.getString("Nama");
}
```

Method `getString()` berguna untuk menelusuri key Nama dari Bundle object, setelah string ditemukan kemudian nilai diberikan ke EditText untuk ditampilkan dengan menggunakan method `setHint()`:

```
EditText txt_username = (EditText)
findViewById(R.id.txt_username);
txt_username.setHint(defaultName);
```

## 4.9 Memanggil Aplikasi Built-In Dengan Intent

Salah satu aspek penting pemrograman Android adalah penggunaan intent untuk memanggil activity dari aplikasi lain. Aplikasi dapat pula memanggil beberapa aplikasi built-in (aplikasi yang sudah tertanam di dalam perangkat Android). Sebagai contoh, Jika aplikasi mengharapkan user agar melakukan panggilan (call) pada orang tertentu yang sudah tersimpan di dalam aplikasi Contacts, maka Anda gunakan object Intent untuk menjalankan aplikasi Contacts, sehingga user dapat memilih daftar orang yang akan dipanggil.

Buat project baru bernama **Intent**, klik menu Eclipse **File | New | Android project**. Isikan beberapa textbox pada form seperti berikut ini:

<b>Project name</b>	Intent
<b>Build Target</b>	Google APIs, Google Inc, 2.3.3
<b>Application name</b>	Intent
<b>Package name</b>	net.belajar2android.Intent
<b>Create Activity</b>	IntentActivity

Tambahkan beberapa pernyataan berhuruf tebal dalam file main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
<Button
    android:id="@+id/btn_webBrowser"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Web Browser" />
<Button
    android:id="@+id/btn_makeCalls"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Buat Panggilan" />
<Button
    android:id="@+id/btn_showMap"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
```

```
    android:text="Tampilkan Peta" />
<Button
    android:id="@+id	btn_chooseContact"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Pilih Kontak" />
</LinearLayout>
```

Tambahkan pernyataan berhuruf tebal ke dalam file IntentActivity.java:

```
package net.belajar2android.Intent;

import android.app.Activity;
import android.os.Bundle;

import android.content.Intent;
import android.net.Uri;
import android.provider.ContactsContract;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class IntentActivity extends Activity {
    Button b1, b2, b3, b4;
    int request_Code = 1;

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

        //---Tombol web browser---
        b1 = (Button) findViewById(R.id.btn_webBrowser);
        b1.setOnClickListener(new OnClickListener() {
            public void onClick(View arg0) {
                Intent i = new
                Intent(android.content.Intent.ACTION_VIEW,
                Uri.parse("http://doditsuprianto.web.id"));
                startActivity(i);
            }
        });
        //---Membuat tombol call---
        b2 = (Button) findViewById(R.id.btn_makeCalls);
        b2.setOnClickListener(new OnClickListener() {
            public void onClick(View arg0){
                Intent i = new
                Intent(android.content.Intent.ACTION_DIAL,
                Uri.parse("tel:+6281252080603"));
                startActivity(i);
            }
        });
    }
}
```

```

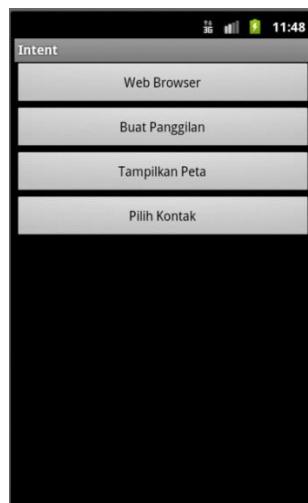
//---Tombol Tampilka Peta---
b3 = (Button) findViewById(R.id.btn_showMap);
b3.setOnClickListener(new OnClickListener() {
    public void onClick(View arg0){
        Intent i = new Intent(android.content.Intent.ACTION_VIEW,
Uri.parse("geo:-7.98444444, +112.63194444"));
        startActivity(i);
    }
});

//---Tombol memilih kontak---
b4 = (Button) findViewById(R.id.btn_chooseContact);
b4.setOnClickListener(new OnClickListener() {
    public void onClick(View arg0){
        Intent i = new Intent(android.content.Intent.ACTION_PICK);
        i.setType(ContactsContract.Contacts.CONTENT_TYPE);
        startActivityForResult(i, request_Code);
    }
});

public void onActivityResult(int requestCode, int resultCode,
Intent data) {
    if (requestCode == request_Code) {
        if (resultCode == RESULT_OK) {
            Toast.makeText(this, data.getData().toString(),
Toast.LENGTH_SHORT).show();
            Intent i = new Intent(android.content.Intent.ACTION_VIEW,
Uri.parse(data.getData().toString()));
            startActivity(i);
        }
    }
}

```

Jalankan aplikasi ke emulator Android dengan menekan tombol keyboard F11, maka tampilannya kurang lebih tampak seperti berikut.



Aplikasi Utama



Pemanggilan Browser



Pemanggilan Call



Pemanggilan Maps



**Pemanggilan Contacts**

**Gambar 4.14 Pemanggilan Activity Built-in Dengan Intent**

Aplikasi di atas adalah contoh penggunaan class Intent yang melibatkan beberapa aplikasi built-in Android (seperti Maps, Phone, Contacts dan Browser).

Intent biasanya datang berpasangan, yaitu berupa “action” dan “data”. Action menggambarkan apa yang dilakukan, seperti mengedit item, menampilkan content dari item dan lain-lain. Data menentukan data obyeknya, seperti data seseorang yang ada di dalam database Contacts, dimana Data ditetapkan sebagai object URI seperti: ACTION\_VIEW, ACTION\_DIAL, ACTION\_PICK, Contoh data adalah: <http://www.google.com>, tel:+6281252080603, geo: -7.984444444, -112.631944444, content://contacts.

Pasangan “Action” dan “Data” menggambarkan operasi yang dilakukan. Contoh, untuk *dial* nomor telpon digunakan ACTION\_DIAL/tel:+6281252080603. Untuk menampilkan daftar contacts yang tersimpan di dalam perangkat telpon, gunakan pasangan ACTION\_VIEW/content://contacts. Untuk mengambil contact dari daftar contact, gunakan pasangan ACTION\_PICK/content://contacts.

Pada tombol pertama, dibuat object Intent yang disertai oleh dua argument ke setiap constructor – Action dan Data:

```
Intent i = new Intent(android.content.Intent.ACTION_VIEW,  
Uri.parse("http://doditsuprianto.web.id"));  
startActivity(i);
```

Action diwakili oleh konstanta android.content.Intent.ACTION\_VIEW. Gunakan method parse() dari class Uri untuk mengkonversi string URL menjadi object Uri.

Konstanta android.content.Intent.ACTION\_VIEW sebenarnya merujuk ke tindakan “android.intent.action.VIEW”, sehingga pernyataan sebelumnya dapat ditulis ulang menjadi:

```
Intent i = new Intent("android.intent.action.VIEW",  
Uri.parse("http:// doditsuprianto.web.id"));  
startActivity(i);
```

Potongan kode sebelumnya dapat pula ditulis ulang menjadi :

```
Intent i = new Intent("android.intent.action.VIEW");  
i.setData(Uri.parse("http:// doditsuprianto.web.id "));  
startActivity(i);
```

Kemudian Anda mengeset data secara terpisah menggunakan method setData().

Untuk tombol kedua, Anda men-dial nomor tertentu dengan cara menuliskan nomor telpon di bagian argument data:

```
Intent i = new Intent(android.content.Intent.ACTION_DIAL,  
Uri.parse("tel:+6281252080603"));  
startActivity(i);
```

Dialer akan menampilkan nomor telpon yang akan dipanggil. User harus menekan tombol dial untuk men-dial nomor tersebut. Jika ingin memanggil/call secara langsung tanpa campur tangan user, ubah tindakan/action menjadi:

```
Intent i = new Intent(android.content.Intent.ACTION_CALL,  
Uri.parse("tel:+6281252080603"));  
startActivity(i);
```

<b>Catatan:</b>
-----------------

Jika ingin memanggil nomor tertentu secara langsung, maka Anda perlu menambahkan izin “android.permission.CALL\_PHONE” ke aplikasi.

Jika ingin menampilkan dialer tanpa menentukan nomor telpon apapun, maka caranya adalah:

```
Intent i = new Intent(android.content.Intent.ACTION_DIAL);  
startActivity(i);
```

Tombol ketiga adalah menampilkan peta menggunakan konstanta ACTION\_VIEW, seperti berikut:

```
Intent i = new Intent(android.content.Intent.ACTION_VIEW,  
Uri.parse("geo:-7.98444444, +112.63194444"));
```

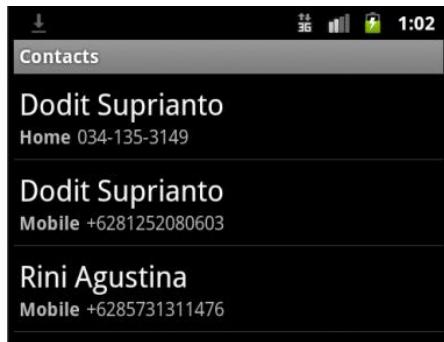
Disini digunakan skema “geo” dan bukan “http”. Aplikasi maps akan dibahas lebih detil pada bab selanjutnya.

Tombol keempat melibatkan aplikasi Contacts yang menampilkan daftar kontak kemudian memanggil salah satu kontak user. Karena Anda meminta user untuk memilih kontak, maka aplikasi Contacts perlu mengembalikan nilai; dalam kasus ini Anda harus men-set tipe data untuk memberitahukan jenis data apa yang akan dikembalikan:

```
Intent i = new Intent(android.content.Intent.ACTION_PICK);  
i.setType(ContactsContract.Contacts.CONTENT_TYPE);  
startActivityForResult(i, request_Code);
```

Jika hendak melihat nama kontak dan nomor telponnya, maka harus diatur tipenya, seperti berikut ini:

```
----Tombol memilih kontak---  
b4 = (Button) findViewById(R.id.btn_chooseContact);  
b4.setOnClickListener(new OnClickListener() {  
public void onClick(View arg0){  
    Intent i = new Intent(android.content.Intent.ACTION_PICK);  
//i.setType(ContactsContract.Contacts.CONTENT_TYPE);  
i.setType(ContactsContract.CommonDataKinds.Phone.CONTENT_TYPE);  
    startActivityForResult(i, request_Code);  
}  
});
```

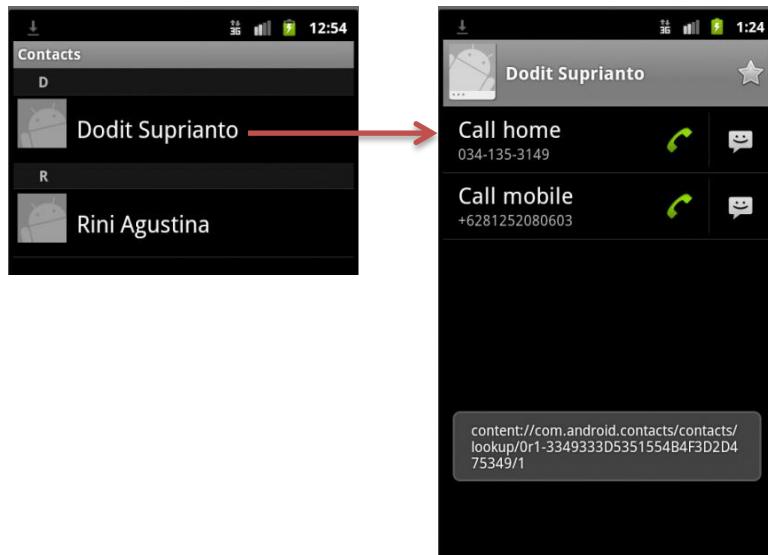


**Gambar 4.15 Contact: Dodit Suprianto, Tipe data: Phone**

Karena di sini kita mengharapkan hasil aplikasi Contacts, maka digunakan method **startActivityForResult**, kemudian mengirimnya ke object Intent dan request code. Diperlukan juga implementasi method **onActivityResult** untuk mendapatkan hasil dari activity:

```
public void onActivityResult(int requestCode, int resultCode,
Intent data)
{
    if (requestCode == request_Code)
    {
        if (resultCode == RESULT_OK)
        {
            Toast.makeText(this, data.getData().toString(),
Toast.LENGTH_SHORT).show();
            Intent i = new Intent(android.content.Intent.ACTION_VIEW,
Uri.parse(data.getData().toString()));
            startActivity(i);
        }
    }
}
```

Ketika memilih kontak tertentu maka URL akan terisi oleh kontak sesuai dengan kontak yang dipilih untuk dikembalikan hasilnya, seperti berikut ini: **content://com.android.contacts/contacts/lookup/0r1-3349333D531554B4F3D2D475349/1** (hasil tidak selalu sama).



**Gambar 4.16 Mengembalikan hasil Contact sesuai pilihan**

URL di atas tidak penting untuk diketahui, yang perlu diketahui adalah bagaimana menciptakan object Intent untuk menampilkan isi kontak:

```
Intent i = new Intent(android.content.Intent.ACTION_VIEW,  
Uri.parse(data.getData().toString()));  
startActivity(i);
```

Pernyataan di atas menunjukkan detail kontak yang dipilih.

## 4.10 Memahami Object Intent

Sampai sejauh ini kita sudah bisa menggunakan object Intent untuk memanggil activity. Kini saatnya penyegaran dan mengetahui lebih detil bagaimana pemahaman terhadap object Intent.

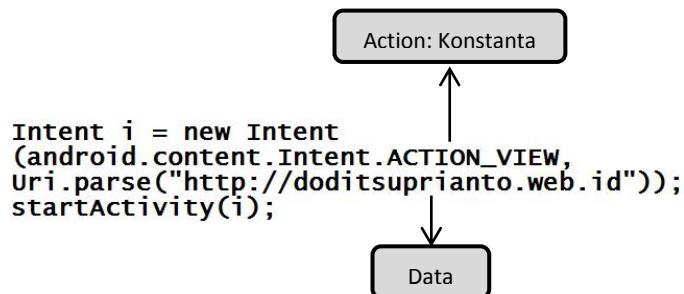
Pertama, Anda telah menggunakan object Intent untuk memanggil activity lain dengan cara mengirim parameter melalui constructor object Intent:

```
startActivity(new Intent("net.belajar2android.ACTIVITY2"));
```

Pernyataan ini `net.belajar2android.ACTIVITY2` dikenal sebagai nama komponen. Digunakan untuk mengidentifikasi sasaran activity yang hendak dilibatkan. Cara lain penulisan komponen adalah dengan menentukan nama class activity-nya jika class tersebut masih jadi satu dengan class activity lainnya (dalam satu project yang sama), seperti berikut:

```
startActivity(new Intent(this, Activity2.class));
```

Anda juga dapat menciptakan object Intent dengan cara mengirimkan action dan data, seperti berikut ini:



**Gambar 4.17 Pernyataan Action dan Data**

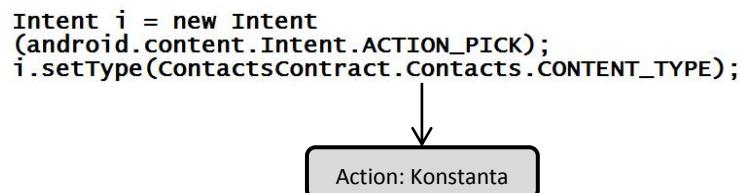
Pernyataan di atas dapat pula ditulis dengan menggunakan method `setData()` seperti berikut:

```
Intent i = new Intent(android.content.Intent.ACTION_VIEW);  
i.setData(Uri.parse("http://doditsuprianto.web.id"));
```

Contoh di atas adalah menampilkan halaman web sesuai dengan URL yang telah ditentukan. Sistem operasi Android akan mencari semua activity yang dapat melayani permintaan/request Anda. Proses ini dikenal sebagai *intent resolution*.

Untuk beberapa intent terkadang tidak perlu menentukan data tertentu. Sebagai contoh untuk memilih kontak dari aplikasi Contacts, Anda hanya

menentukan suatu action yang menandakan jenis MIME yang akan digunakan dengan method setType():



**Gambar 4.18 Penggunaan MIME pada Action**

Disamping mendefinisikan action, data, dan data type, object Intent juga mendefinisikan category. Namun pada kesempatan kali ini kita tidak membahas category.

## 4.11 Menampilkan Notifikasi

Class Toast adalah salah satu cara mudah untuk menampilkan pesan singkat kepada user secara sekilas, yaitu beberapa detik kemudian pesan menghilang dari layar.

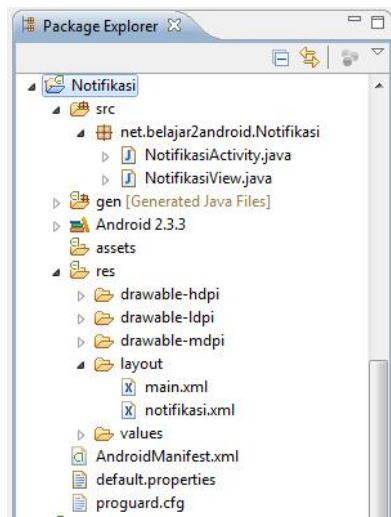
Untuk pesan-pesan yang sangat penting, sebaiknya gunakan metode yang lebih sesuai, misalnya **NotificationManager**, yaitu menampilkan pesan secara terus-menerus yang diletakkan pada posisi teratas dari layar Android (dikenal sebagai status bar atau notification bar).

Buatlah project Android baru bernama **Notifikasi**.

<b>Project name</b>	Notifikasi
<b>Build Target</b>	Android 2.3.3 API Level 10
<b>Application name</b>	Notifikasi
<b>Package name</b>	net.belajar2android. Notifikasi
<b>Create Activity</b>	Notifikasi Activity

Tambahkan file class baru bernama **NotifikasiView.java** ke dalam folder src. Klik menu Eclipse File | New | Class.

Tambahkan pula file XML bernama **notifikasi.xml** ke dalam folder res/layout. Klik menu Eclipse File | New | Android XML File.



**Gambar 4.19 Struktur project hasil penambahan file Class dan XML**

Tambahkan pernyataan berikut ke dalam file notifikasi.xml:

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

Tambahkan pernyataan berikut ke dalam file NotifikasiView.java:

```
package net.belajar2android.Notifikasi;

import android.app.Activity;
import android.app.NotificationManager;
```

```

import android.os.Bundle;

public class NotifikasiView extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.notifikasi);

        //---mencari layanan notifikasi manager---
        NotificationManager nm = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);

        //---membatalkan notifikasi yang sedang berjalan---

        nm.cancel(getIntent().getExtras().getInt("notificationID"
));
    }
}

```

Tambahkan pernyataan berikut ke dalam file AndroidManifest.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.belajar2android.Notifikasi"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />

    <application android:icon="@drawable/icon"
    android:label="@string/app_name">
        <activity android:name=".NotifikasiActivity"
        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=".NotifikasiView" android:label="Detail
Notifikasi">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <action android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.VIBRATE" />
</manifest>

```

Tambahkan pernyataan berikut ke dalam file main.xml:

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

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<Button
    android:id="@+id/btn_tampilkanNotifikasi"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Tampilkan Notifikasi" />
</LinearLayout>

```

Tambahkan pernyataan berikut ke dalam file NotifikasiActivity.java:

```

package net.belajar2android.Notifikasi;

import android.app.Activity;
import android.os.Bundle;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.view.View;
import android.widget.Button;

public class NotifikasiActivity extends Activity {
    int notificationID = 1;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button tombol = (Button)
findViewById(R.id.btn_tampilkanNotifikasi);
        tombol.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View v) {
                tampilkanNotifikasi();
            }
        });
    }

    protected void tampilkanNotifikasi() {
        //---PendingIntent mengaktifkan activity jika user memilih
        // notifikasi ini---
        Intent i = new Intent(this, NotifikasiView.class);
        i.putExtra("notificationID", notificationID);

        PendingIntent pendingIntent =
PendingIntent.getActivity(this, 0, i, 0);

        NotificationManager nm = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);

```

```

Notification notif = new Notification(R.drawable.icon,
"Pengingat: Pertemuan 10 menit lagi",
System.currentTimeMillis());

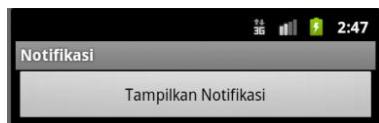
CharSequence dari = "System Alarm";
CharSequence pesan = "Pertemuan dengan pelanggan pada jam 3
sore";

notif.setLatestEventInfo(this, dari, pesan, pendingIntent);

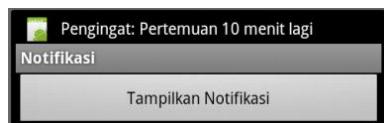
//--- jedah 100 mili detik, bergetar 250 mili detik, dan
berhenti sementara 100 mili detik dan
// kemudian bergetar 250 mili detik ---
notif.vibrate = new long[] {100, 250, 100, 500};
nm.notify(notificationID, notif);
}
}

```

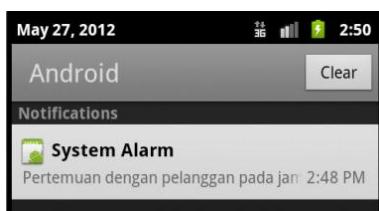
Jalankan aplikasinya dan perhatikan tampilan di Emulator. Klik tombol **Tampilan Notifikasi**, maka notifikasi akan muncul pada status bar. Klik dan geser status bar ke bawah sehingga keterangan notifikasi akan tampak.



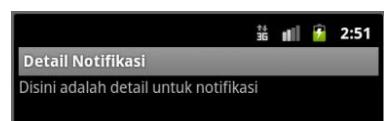
**Klik tombol “Tampilan Notifikasi”**



**Klik dan geser ke bawah  
“Pengingat: Pertemuan 10 ...”**



**Klik “System Alarm”**



**Detil notifikasi**

**Gambar 4.20 Status Bar atau Notifikasi Bar**

Bagaimana program bekerja? Pertama adalah membuat object Intent yang menunjuk ke class NotificationView:

```
//---PendingIntent mengaktifkan activity jika user memilih  
// notifikasi ini---  
Intent i = new Intent(this, NotifikasiView.class);  
i.putExtra("notificationID", notificationID);
```

Intent ini digunakan untuk menjalankan activity lain ketika user memilih notifikasi dari daftar notifikasi. Pada contoh di atas ditambahkan pasangan key/value ke object Intent sehingga diperoleh ID notifikasi, mengenalkan notifikasi ke activity target yang dimaksud. ID ini juga akan digunakan saat mengakhiri notifikasi nantinya.

Diperlukan pula object **PendingIntent** dengan Inisialisasi seperti berikut:

```
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,  
i, 0);
```

Method `getActivity()` adalah anggota dari object `PendingIntent` yang harus diberi nilai, dimana variable argumentnya adalah sebagai berikut:

- context — Context aplikasi
- request code — Request code untuk intent
- intent — Intent untuk menjalankan target activity.
- flags — flag/penanda yang mana activity akan dijalankan.

Kemudian Anda menemui instansiasi class `NotificationManager` dan menciptakan instance class `Notification`:

```
NotificationManager nm = (NotificationManager)  
getSystemService(NOTIFICATION_SERVICE);  
  
Notification notif = new Notification(R.drawable.icon,  
"Pengingat: Pertemuan 10 menit lagi",  
System.currentTimeMillis());
```

Class `Notification` mengizinkan kita untuk menentukan informasi utama apa yang akan tampak, ketika notifikasi muncul pertama kali di status

bar. Pada argument kedua, konstruktor Notification diset sebagai "tick text" pada status bar.



**Gambar 4. 21 Bar Notification**

Selanjutnya diatur detil notifikasi dengan menggunakan method `setLatestEventInfo()`:

```
CharSequence dari = "System Alarm";
CharSequence pesan = "Pertemuan dengan pelanggan pada jam 3
sore";

notif.setLatestEventInfo(this, dari, pesan, pendingIntent);

//--- jedah 100 mili detik, bergetar 250 mili detik dan berhenti
sementara 100 mili detik
// kemudian bergetar 250 mili detik ---
notif.vibrate = new long[] {100, 250, 100, 500};
```

Terakhir, untuk menampilkan notifikasi digunakan method `notify()`:

```
nm.notify(notificationID, notif);
```

Ketika user mengklik notifikasi maka activity `NotificationView` (`NotificationView.java`) dijalankan. Untuk mengakhiri notifikasi gunakan method `cancel()` dari object `NotificationManager` yang akan mengirim ID notifikasi melalui object `Intent`:

```
//---mencari layanan notifikasi manager---
NotificationManager nm = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);

//---membatalkan notifikasi yang sedang berjalan---
nm.cancel(getIntent().getExtras().getInt("notificationID"));
```

# **BAB 5**

# **MEMAHAMI ANTARMUKA (UI)**

Dalam bab ini akan dipelajari mengenai class Activity beserta siklus hidupnya. Kita sudah mempelajari activity sebelumnya, yang berarti sudah mengatahui bagaimana user berinteraksi dengan aplikasi. Namun activity sendiri tidak mempunyai kepentingan langsung terhadap urusan antarmuka, oleh karenanya activity masih memerlukan Views dan ViewGroups. Bab ini juga akan dipelajari secara detil bagaimana menciptakan antarmuka user dan bagaimana user berinteraksi terhadapnya.

## **5.1 Memahami Komponen Layar**

Activity adalah unit dasar dari aplikasi Android. Activity menampilkan antarmuka user pada aplikasi yang berisi widget seperti button, label, textbox, dan lain-lain. Secara khusus pendefinisan antarmuka adalah melalui penggunaan XML, contohnya file main.xml yang terletak di dalam folder res/layout, seperti ini:

```
<?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="@string/hello" />
</LinearLayout>
```

Saat aplikasi berjalan, antarmuka XML di-load ke dalam event handler `onCreate()` pada class Activity dengan menggunakan method `setContentView()`:

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
}
```

Ketika project dikompilasi maka setiap elemen di dalam file XML akan dikompilasi menjadi class GUI (graphic user interface) Android, dengan atribut yang diwakili oleh method. Sistem Android kemudian menciptakan UI activity ketika di-load.

Cara termudah membangun UI adalah dengan menggunakan file XML statis, namun terkadang kita memerlukan UI yang dinamis saat aplikasi berjalan, oleh karena itu dimungkinkan pula menciptakan UI melalui kode program yang akan dibahas pada bab selanjutnya.

## 5.2 View Dan ViewGroup

View adalah widget yang tampak di layar. Contoh view adalah button, label dan text box. View adalah turunan dari class dasar android android.view.View.

Satu atau lebih view dapat dikelompokkan secara bersama menjadi ViewGroup. ViewGroup (ViewGroup sendiri adalah tipe khusus dari view) menghasilkan layout dari serangkaian view. ViewGroup adalah turunan dari class dasar android.view.ViewGroup. Android mendukung ViewGroup: LinearLayout, AbsoluteLayout, TableLayout, RelativeLayout, FrameLayout, ScrollView

### 5.2.1 LinearLayout

LinearLayout menyusun view-view dalam satu kolom atau satu baris.

Turunan view-view dapat ditata baik secara vertikal maupun horizontal.

```
<?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="@string/hello" />
</LinearLayout>

```

Perhatikan elemen-elemen yang terdapat dalam file main.xml di atas. Elemen root adalah <LinearLayout> yang berisi elemen <TextView>. Elemen <LinearLayout> mengendalikan elemen-elemen lain yang ada di dalamnya (elemen-elemen yang menjadi View).

Setiap View dan ViewGroup mempunyai serangkaian atribut, antara lain:

Atribut	Keterangan
layout_width	Menentukan lebar View atau ViewGroup
layout_height	Menentukan tinggi View atau ViewGroup
layout_marginTop	Menentukan tambahan ruang pada sisi atas View atau ViewGroup
layout_marginBottom	Menentukan ruang tambahan pada sisi bawah View atau ViewGroup
layout_marginLeft	Menentukan ruang tambahan pada sisi kiri View atau ViewGroup
layout_marginRight	Menentukan ruang tambahan pada sisi kanan View atau ViewGroup
layout_gravity	Menentukan bagaimana View-View anak diposisikan
layout_weight	Menentukan berapa banyak ruang tambahan dalam layout yang seharusnya ditempatkan ke View

layout_x	Menentukan koordinat x View atau ViewGroup
layout_y	Menentukan koordinat y View atau ViewGroup

**Catatan:**

Beberapa atribut hanya dapat diaplikasikan ketika sebuah View berada dalam ViewGroup tertentu. Contoh, atribut layout\_weight dan layout\_gravity yang hanya dapat diaplikasikan jika View berada di dalam LinearLayout atau TableLayout.

Sebagai contoh, lebar elemen <TextView> mengisi keseluruhan lebar parent-nya (layar) menggunakan konstanta **fill\_parent**. Tinggi elemen <TextView> diset dengan **wrap\_content** yang berarti tinggi TextView menyesuaikan tinggi dari isinya (tinggi tulisan), contoh lain: tinggi tombol menyesuaikan tinggi teks yang tertulis dalam tombol tersebut.

Jika Anda tidak ingin view <TextView> menempati keseluruhan baris maka atur atribut layout teks menjadi wrap\_content:

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

Berikut ini menyebabkan lebar view sama dengan lebar teks yang mengisinya:

```
<?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="105dp"
    android:layout_height="wrap_content"
```

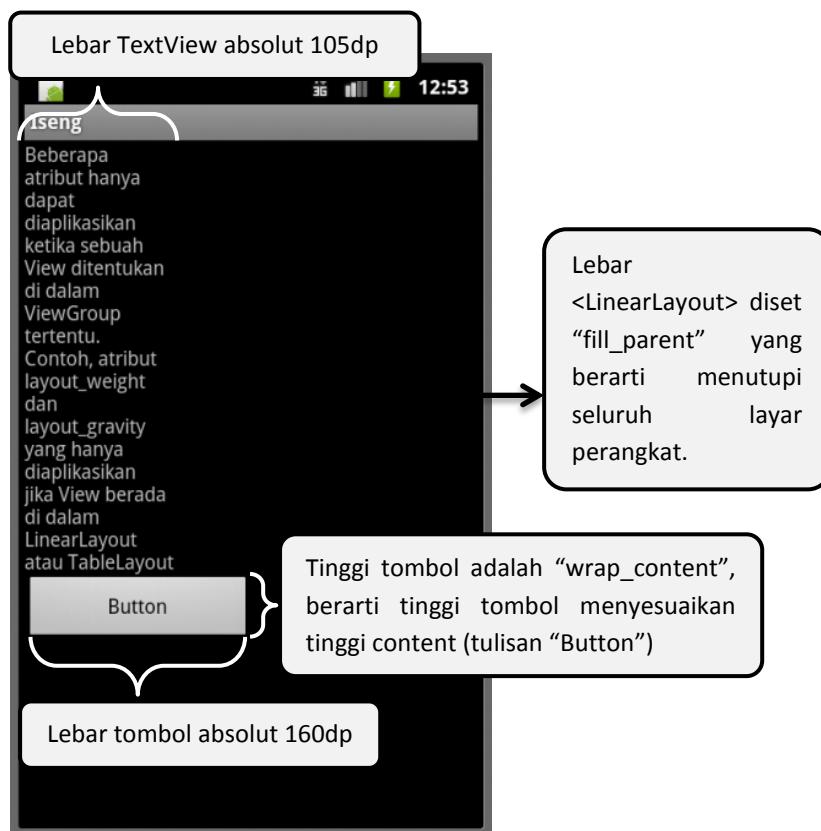
```
    android:text="@string/hello" />
<Button
    android:layout_width="160dp"
    android:layout_height="wrap_content"
    android:text="Button" />
</LinearLayout>
```

**Catatan:**

Penentuan ukuran element UI Android harus memperhatikan satuan ukurannya, antara lain sebagai berikut:

- **dp** – Density-independent pixel. 160dp sama dengan satu inch ukuran layar sebenarnya. Ini satuan yang disarankan saat menentukan dimensi view layout. Satuan “dp” bisa pula dituliskan “dip” untuk merujuk ke satuan densisty pixel.
- **sp** – Scale-independent pixel. Ini serupa dengan dp namun biasa digunakan untuk menentukan ukuran font.
- **dt** – Point. Point didefinisikan sebagai 1/72 inch didasarkan pada ukuran layar sebenarnya.
- **dx** – Pixel. Berhubungan dengan pixel sebenarnya dari layar. Penggunaan pixel tidak disarankan untuk UI karena mungkin ada perbedaan ukuran layar pada masing-masing perangkat Android.

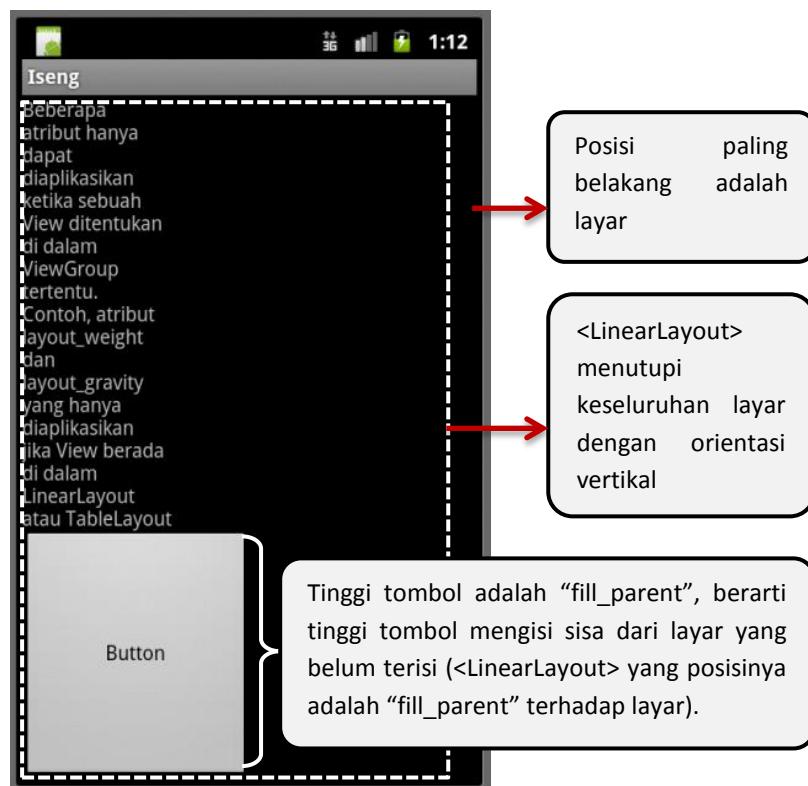
Pada contoh di atas, lebar view TextView dan Button bernilai tetap (absolute) dengan lebar TextView 105 density-independent pixel dan lebar Button 160 density-independent pixel.



**Gambar 5.1 Pengaturan dimensi view**

Modifikasi ukuran view Button dalam file main.xml, seperti ini:

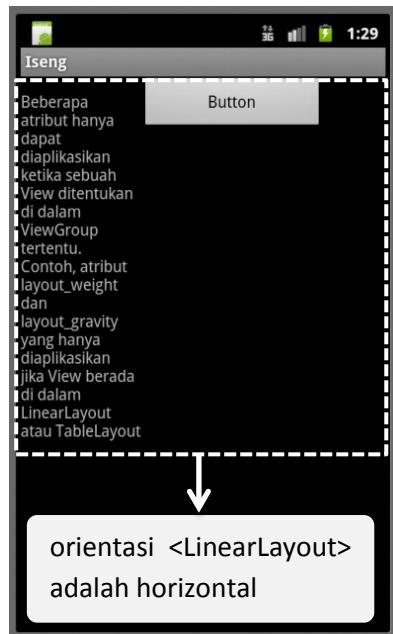
```
<Button  
    android:layout_width="160dp"  
    android:layout_height="fill_parent"  
    android:text="Button"  
/>
```



**Gambar 5.2 Pengaturan dimensi view**

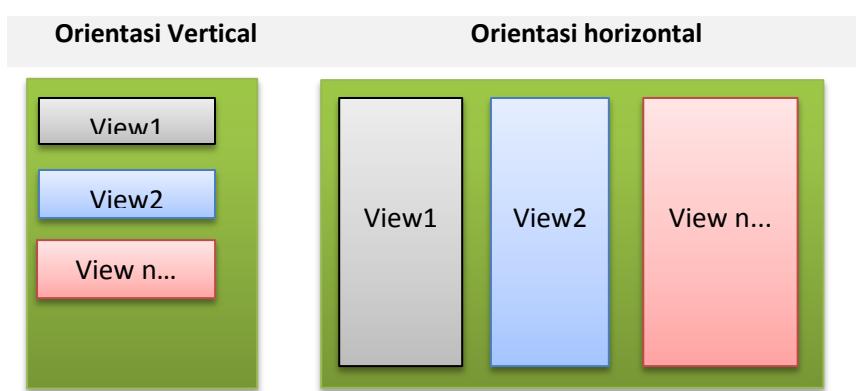
Sekarang kita coba kalau atribut orientation diset menjadi horizontal:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" >
    <TextView
        android:layout_width="105dp"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
    <Button
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:text="Button"
    />
</LinearLayout>
```



**Gambar 5.3 Pengaturan dimensi view dengan orientasi**

Berikut ini gambaran masalah orientation vertikal dan horizontal



**Gambar 5.4 Orientasi View**

Di dalam LinearLayout dapat diimplementasikan atribut layout\_weight dan layout\_gravity ke view, perhatikan file main.xml berikut ini:

```
<?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="105dp"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
    <Button
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:text="Button"
        android:layout_gravity="right"
        android:layout_weight="0.2" />
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:layout_weight="0.8" />
</LinearLayout>
```

Posisi Button adalah rata kanan dari parent-nya (dalam kasus ini adalah LinearLayout) yang menggunakan atribut layout\_gravity. Sedangkan atribut layout\_weight merupakan perbandingan antara view Button dan view EditText dalam menempati sisa ruang pada layar. Total semua nilai atribut layout\_weight harus 1 ( $0.2 + 0.8$ ).

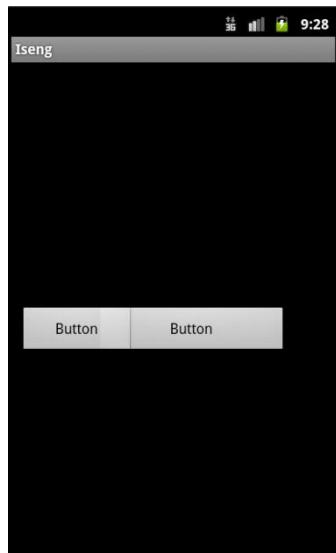


Gambar 5.5 Pengaturan dimensi view dengan layout\_gravity

### 5.2.2 AbsoluteLayout

AbsoluteLayout berguna untuk menentukan posisi pasti setiap view-view anak. Perhatikan UI yang terdefinisi dalam file main.xml berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button
        android:layout_width="188dp"
        android:layout_height="wrap_content"
        android:text="Button"
        android:layout_x="126px"
        android:layout_y="361px"/>
    <Button
        android:layout_width="113dp"
        android:layout_height="wrap_content"
        android:text="Button"
        android:layout_x="12px"
        android:layout_y="361px"/>
</AbsoluteLayout>
```



**Gambar 5.6 Pengaturan dimensi view dengan ukuran tetap**

AbsoluteLayout sering membawa masalah ketika activity ditampilkan pada layar berkerapatan tinggi. Oleh karena itu sejak Android 1.5 AbsoluteLayout mulai diabaikan meskipun masih tetap didukung sampai Android versi terbaru. Sebaiknya Anda tidak menggunakan AbsoluteLayout untuk UI, apalagi jika digunakan pada perangkat Android berbeda-beda.

### **5.2.3 TableLayout**

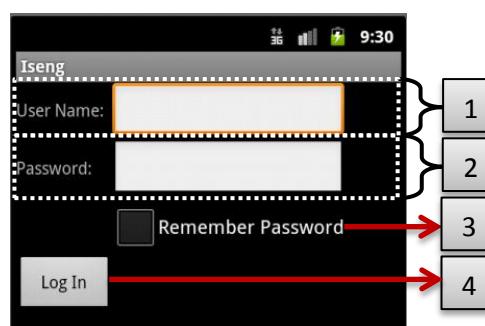
TableLayout mengelompokkan view-view menjadi baris-baris dan kolom-kolom. Gunakan elemen <TableRow> untuk mewakili baris di dalam table. Setiap baris dapat berisi satu atau lebih view-view (kolom). Setiap view yang ditempatkan di dalam baris adalah berbentuk cell, dimana lebar cell mengikuti lebar kolom view terbesar. Lebar setiap kolom ditentukan oleh lebar cell-nya.

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

```

xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_height="fill_parent"
android:layout_width="fill_parent" >
<TableRow>
    <TextView
        android:text="User Name:"
        android:width ="120px"
    />
    <EditText
        android:id="@+id/txtUserName"
        android:width="200px" />
</TableRow>
<TableRow>
    <TextView
        android:text="Password:"
    />
    <EditText
        android:id="@+id/txtPassword"
        android:password="true"
    />
</TableRow>
<TableRow>
    <CheckBox android:id="@+id/chkRememberPassword"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Remember Password"
    />
</TableRow>
<TableRow>
    <Button
        android:id="@+id/buttonSignIn"
        android:text="Log In" />
</TableRow>
</TableLayout>

```



**Gambar 5.7 Pengaturan dimensi view dengan TableLayout**

Keterangan gambar:

1. Baris pertama (<TableRow> pertama) terdiri dari 2 cell/view, yaitu 1 <TextView> dan 1 <EditText>.
2. Baris kedua (<TableRow> kedua) terdiri dari 2 cell/view, yaitu 1 <TextView> dan 1 <EditText>.
3. Baris ketiga (<TableRow> ketiga) terdiri dari 1 cell/view, yaitu 1 <CheckBox>.
4. Baris keempat (<TableRow> keempat) terdiri dari 1 cell/view, yaitu <Button>.

Khusus dibagian TableRow ketiga terdapat view <TextView />, bertujuan memberi ruang kosong agar CheckBox rata kiri sama dengan <EditText> pada TableRow kedua. Bila tidak diberi <TextView /> maka hasilnya akan tampak seperti berikut ini:



**Gambar 5.8 Pengaturan dimensi view dengan TableLayout**

#### 5.2.4 RelativeLayout

Dengan RelativeLayout Anda dapat menentukan bagaimana view-view anak diposisikan relative terhadap view lainnya.

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

```

<TextView
    android:id="@+id/lblComments"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Komentar"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true" />
<EditText
    android:id="@+id/txtComments"
    android:layout_width="fill_parent"
    android:layout_height="170px"
    android:textSize="18sp"
    android:layout_alignLeft="@+id/lblComments"
    android:layout_below="@+id/lblComments"
    android:layout_centerHorizontal="true" />
<Button
    android:id="@+id(btnSave"
    android:layout_width="125px"
    android:layout_height="wrap_content"
    android:text="Simpan"
    android:layout_below="@+id/txtComments"
    android:layout_alignRight="@+id/txtComments" />
<Button
    android:id="@+id btnCancel"
    android:layout_width="124px"
    android:layout_height="wrap_content"
    android:text="Batal"
    android:layout_below="@+id/txtComments"
    android:layout_alignLeft="@+id/txtComments" />
</RelativeLayout>

```



**Gambar 5.9 Pengaturan dimensi view dengan RelativeLayout**

Perhatikan elemen-elemen yang berhuruf tebal di atas, terlihat bahwa perataan kiri, kanan, atas dan bawah relative terhadap view lainnya.

Posisi setiap view tergantung dari posisi view lainnya. RelativeLayout mempunyai beberapa atribut, antara lain: layout\_alignParentTop, layout\_alignParentLeft, layout\_alignLeft, layout\_alignRight, layout\_below, layout\_centerHorizontal.

Nilai untuk setiap atribut di atas mengacu pada ID view yang menjadi rujukannya.

### 5.2.5 FrameLayout

FrameLayout adalah *placeholder* (semacam wadah bagi beberapa view, sehingga menjadi satu kelompok) pada layar sehingga Anda dapat menampilkannya menjadi satu view. View-view yang ditambahkan pada FrameLayout selalu berposisi di kiri atas layout.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:id="@+id/RLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
<TextView
    android:id="@+id/lblComments"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ini Fotoku dan anakku"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true" />
<FrameLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/lblComments"
    android:layout_below="@+id/lblComments"
    android:layout_centerHorizontal="true" >
<ImageView
    android:src = "@drawable/keluarga"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Print Foto" />
```

```
</FrameLayout>  
</RelativeLayout>
```

TextView lblComments masih berada di area <RelativeLayout>, sedangkan ImageView dan Button berada dalam satu kelompok <FrameLayout>, sehingga Button menimpa gambarnya.



**Gambar 5.10 Pengaturan dimensi view dengan FrameLayout**

**Catatan:** “keluarga” adalah file gambar, dengan anggapan bahwa file gambar sudah ada di dalam project. Untuk menambahkan gambar silakan copy-paste gambar dari windows explorer ke folder res/drawable-ldpi atau folder lainnya, sesuai dengan kualitas gambar yang dimiliki.

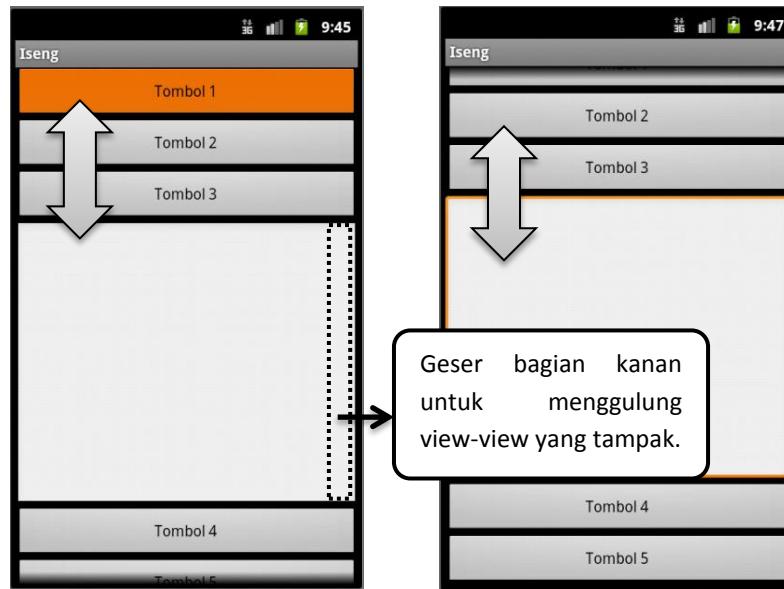
### 5.2.6 ScrollView

ScrollView adalah tipe khusus dari FrameLayout, dengan ScrollView maka user dapat meng gulung daftar view-view agar menempati ruang lebih dari tampilan fisik sebenarnya. ScrollView bisa berisi hanya satu view anak atau ViewGroup seperti normalnya LinearLayout.

File main.xml berikut ini menunjukkan ScrollView yang mengandung LinearLayout, berisi beberapa view Button dan EditText:

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
<Button
    android:id="@+id/button1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Tombol 1" />
<Button
    android:id="@+id/button2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Tombol 2" />
<Button
    android:id="@+id/button3"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Tombol 3" />
<EditText
    android:id="@+id/txt"
    android:layout_width="fill_parent"
    android:layout_height="400px" />
<Button
    android:id="@+id/button4"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Tombol 4" />
<Button
    android:id="@+id/button5"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Tombol 5" />
</LinearLayout>
</ScrollView>
```

Dari gambar di bawah menunjukkan bagaimana ScrollView bekerja. User dapat melakukan penggeseran ke atas atau ke bawah untuk memunculkan view-view yang tak tampak baik terletak di bawah atau di atas.



Gambar 5.11 Pengaturan dimensi view dengan ScrollView

### 5.3 Penyesuaian Orientasi Tampilan

Android mendukung dua orientasi layar, yaitu *portrait* dan *landscape*. Ketika orientasi layar berubah maka tampilan content activity secara otomatis akan ditata ulang ke dalam orientasi baru. Ini karena event `onCreate()` activity akan dijalankan apapun orientasi yang akan ditampilkan.

Terdapat dua teknik untuk menangani perubahan orientasi layar:

- **Anchoring** – Anchoring adalah cara termudah untuk menjangkar view-view ke empat titik sudut layar. Istilah anchor adalah sebagai titik acuan terhadap posisi view-view yang bersifat relative.

- **Resizing dan Repositioning** - *anchoring* dan *centralizing* adalah teknik sederhana untuk memastikan bahwa view-view dapat teratas oleh perubahan orientasi layar, namun teknik yang paling istimewa adalah *resizing* setiap view, dapat menyesuaikan orientasi layar yang sedang aktif saat ini.

### 5.3.1 Anchoring View-View

Anchoring mudah diterima oleh penggunaan RelativeLayout. Perhatikan file main.xml yang berisi lima view Button yang terikat dalam elemen <RelativeLayout>:

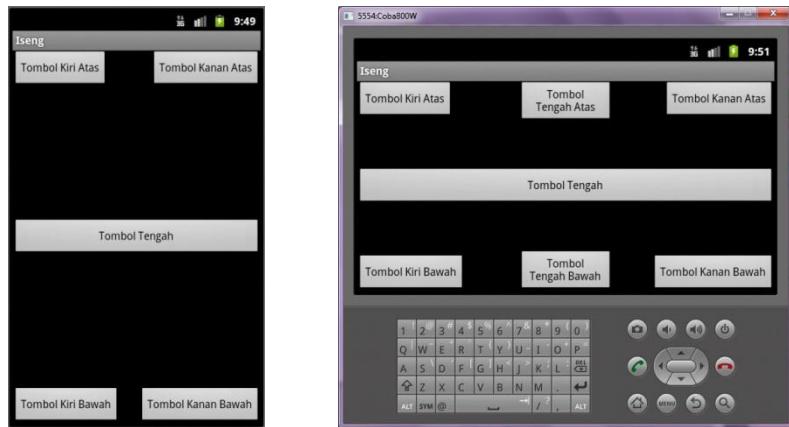
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tombol Kiri Atas"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true" />
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tombol Kanan Atas"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true" />
    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tombol Kiri Bawah"
        android:layout_alignParentLeft="true"
        android:layout_alignParentBottom="true" />
    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tombol Kanan Bawah"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true" />
    <Button
        android:id="@+id/button5"
        android:layout_width="fill_parent"
```

```
    android:layout_height="wrap_content"
    android:text="Tombol Tengah"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true" />
</RelativeLayout>
```

Perhatikan atribut berhuruf tebal pada beberapa view Button di atas:

- **layout\_alignParentLeft** — meratakan posisi view disisi kiri dari view parent.
- **layout\_alignParentRight** — meratakan posisi view disisi kanan dari view parent.
- **layout\_alignParentTop** — meratakan posisi view disisi atas dari view parent.
- **layout\_alignParentBottom** — meratakan view disisi bawah dari view parent.
- **layout\_centerVertical** — menengahkan view secara vertical diantara view parent.
- **layout\_centerHorizontal** — menengahkan view secara horizontal diantara view parent.

Gambar di bawah menunjukkan activity ketika ditampilkan dengan mode portrait. Saat orientasi layar berubah menjadi mode landscape, empat tombol diratakan ke empat titik penjuru layar dan tombol tengah berada pada posisi tengah layar dengan lebar menyesuaikan layarnya (fill\_parent).



**Gambar 5.12 Perubahan orientasi dari vertical ke horizontal**

Untuk mengubah orientasi layar, tekan tombol keyboard Ctrl+F11 saat Emulator Android aktif. Sebagai rujukan, kunjungi situs berikut ini <http://developer.android.com/guide/developing/tools/emulator.html#c ontrolling>.

Ringkasan key pada emulator Android antara lain:

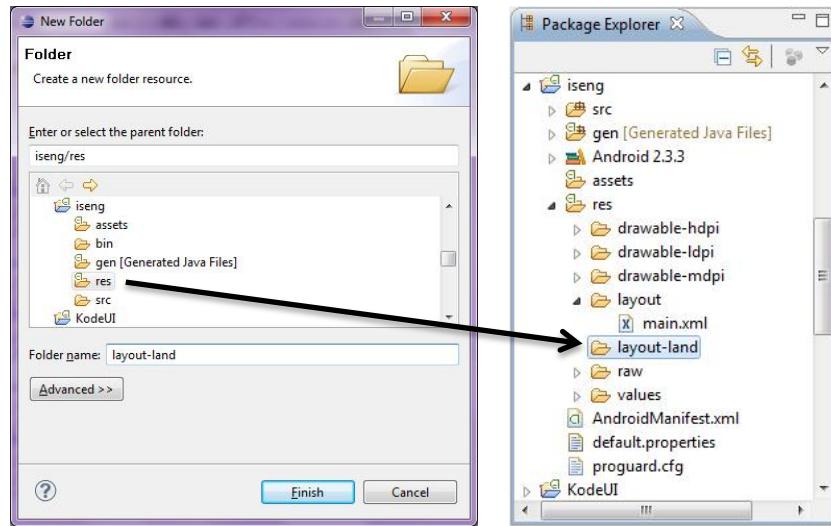
Key Perangkat Emulator	Key Keyboard
Home	HOME
Menu (softkey kiri)	F2 atau Page-up button
Star (softkey kanan)	Shift-F2 atau Page Down
Back	ESC
Tombol Call/dial	F3
Tombol Hangup/end call	F4
Search	F5
Tombol Power	F7
Tombol menaikkan volume Audio	KEYPAD_PLUS, Ctrl-5

Tombol menurunkan volume Audio	KEYPAD_MINUS, Ctrl-F6
Tombol Camera	Ctrl-KEYPAD_5, Ctrl-F3
Mengubah orientasi layout sebelumnya (contoh, portrait, landscape)	KEYPAD_7, Ctrl-F11
Mengubah orientasi layout setelahnya (contoh, portrait, landscape)	KEYPAD_9, Ctrl-F12
Toggle cell networking on/off	F8
Toggle code profiling	F9
Toggle mode fullscreen	Alt-Enter
Toggle mode trackball	F6
Enter trackball mode sementara (selama key ditekan)	Delete
DPad kiri/atas/kanan/bawah	KEYPAD_4/8/6/2
DPad klik tengah	KEYPAD_5
Onion alpha menaikkan/menurunkan	KEYPAD_MULTIPLY(*) / KEYPAD_DIVIDE(/)

### 5.3.2 Resizing Dan Repositioning

Karena penjangkaran view-view terhadap empat titik sudut layar sangatlah penting, maka cara termudah untuk mengkustomisasi UI yang didasari oleh orientasi layar adalah dengan membuat file-file XML secara terpisah di folder res/layout untuk UI disetiap orientasi.

Untuk mendukung mode landscape, dibuat folder baru di dalam folder res dan beri nama **layout-land** (contoh folder yang mewakili landscape). Caranya adalah dengan klik kanan pada folder res kemudian klik menu New | Folder seperti tampak berikut ini:



**Gambar 5.13 Penambahan folder baru dalam res**

Pada dasarnya file main.xml yang berada di dalam folder layout mendefinisikan UI untuk activity dalam mode portrait, sedangkan file main.xml yang berada di dalam folder layout-land mendefinisikan UI dalam mode landscape.

Berikut ini isi file main.xml yang berada di dalam folder layout:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    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="Tombol Kiri Atas"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true" />
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tombol Kanan Atas"
        android:layout_alignParentTop="true"
```

```

        android:layout_alignParentRight="true" />
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tombol Kiri Bawah"
    android:layout_alignParentLeft="true"
    android:layout_alignParentBottom="true" />
<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tombol Kanan Bawah"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true" />
<Button
    android:id="@+id/button5"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Tombol Tengah"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true" />
</RelativeLayout>

```

Selanjutnya lakukan copy file main.xml yang berada dalam folder layout dan paste-kan ke folder **layout-land**. Pernyataan yang berhuruf tebal adalah tambahan view yang akan ditampilkan dengan mode landscape:

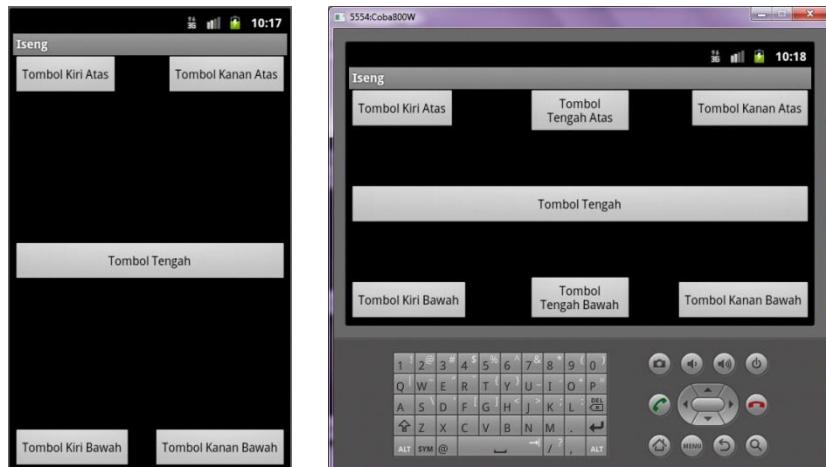
```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    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="Tombol Kiri Atas"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
/>
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tombol Kanan Atas"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true" />
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tombol Kiri Bawah"

```

```
    android:layout_alignParentLeft="true"
    android:layout_alignParentBottom="true" />
<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tombol Kanan Bawah"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true" />
<Button
    android:id="@+id/button5"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Tombol Tengah"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true" />
<Button
    android:id="@+id/button6"
    android:layout_width="180px"
    android:layout_height="wrap_content"
    android:text="Tombol Tengah Atas"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:layout_alignParentTop="true" />
<Button
    android:id="@+id/button7"
    android:layout_width="180px"
    android:layout_height="wrap_content"
    android:text="Tombol Tengah Bawah"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:layout_alignParentBottom="true" />
</RelativeLayout>
```

Ketika activity di-load dalam mode portrait maka akan ditampilkan lima tombol. Ketika activity di-load dalam mode landscape maka akan ditampilkan tujuh tombol, ini membuktikan bahwa file-file XML yang berbeda akan di-load semua saat perangkat memiliki orientasi berbeda.



**Gambar 5.14 Menampilkan orientasi berbeda dari file XML berbeda**

Gunakan cara di atas untuk mengubah orientasi layar. Android secara otomatis akan me-load file XML dengan tepat untuk activity Anda, tergantung dari orientasi yang aktif saat ini.

## 5.4 Mengatur Orientasi Layar

Ikuti pernyataan-pernyataan di bawah ini untuk mendemonstrasikan perilaku activity ketika perangkat mengubah orientasi. Buatlah project baru bernama **Orientasi**. Klik menu **File | New | Android Project**.

<b>Project name</b>	Orientasi
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	Orientasi
<b>Package name</b>	net.belajar2android.Orientasi
<b>Create Activity</b>	OrientasiActivity

Tambahkan pernyataan berhuruf tebal di bawah ini ke dalam file main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
    >
<EditText
    android:id="@+id/txtField1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
<EditText
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
</LinearLayout>

```

Tambahkan pernyataan berhuruf tebal ini ke dalam file

### **OrientasiActivity.java:**

```

package net.belajar2android.Orientasi;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

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

    @Override
    public void onStart() {
        Log.d("StateInfo", "onStart");
        super.onStart();
    }

    @Override
    public void onResume() {
        Log.d("StateInfo", "onResume");
        super.onResume();
    }

    @Override
    public void onPause() {
        Log.d("StateInfo", "onPause");
        super.onPause();
    }

    @Override
    public void onStop() {

```

```

        Log.d("StateInfo", "onStop");
        super.onStop();
    }

    @Override
    public void onDestroy() {
        Log.d("StateInfo", "onDestroy");
        super.onDestroy();
    }

    @Override
    public void onRestart() {
        Log.d("StateInfo", "onRestart");
        super.onRestart();
    }
}

```

Jalankan aplikasi dengan emulator Android. Tuliskan kedua EditText yang ada, kemudian ganti orientasinya menjadi landscape dengan menekan tombol keyboard Ctrl+F11, maka tulisan yang terdapat pada EditText pertama akan bertahan, sedangkan tulisan pada EditText kedua akan hilang.

Perhatikan window perspective LogCat, hasilnya adalah sebagai berikut:

```

07-03 22:49:38.070: DEBUG/StateInfo(1495): onCreate
07-03 22:49:38.070: DEBUG/StateInfo(1495): onStart
07-03 22:49:38.130: DEBUG/StateInfo(1495): onResume

07-03 22:53:41.220: DEBUG/StateInfo(1495): onPause
07-03 22:53:41.240: DEBUG/StateInfo(1495): onStop
07-03 22:53:41.240: DEBUG/StateInfo(1495): onDestroy
07-03 22:53:42.040: DEBUG/StateInfo(1495): onCreate
07-03 22:53:42.040: DEBUG/StateInfo(1495): onStart
07-03 22:53:42.090: DEBUG/StateInfo(1495): onResume

```

Bagaimana program bekerja? Dari keluaran window LogCat tampak bahwa ketika perangkat mengubah orientasi, activity akan dihapus:

```

07-03 22:53:41.220: DEBUG/StateInfo(1495): onPause
07-03 22:53:41.240: DEBUG/StateInfo(1495): onStop
07-03 22:53:41.240: DEBUG/StateInfo(1495): onDestroy

```

Kemudian dibuat ulang:

```

07-03 22:53:42.040: DEBUG/StateInfo(1495): onCreate
07-03 22:53:42.040: DEBUG/StateInfo(1495): onStart
07-03 22:53:42.090: DEBUG/StateInfo(1495): onResume

```

Anda harus memahami perilaku ini, karena terkadang perlu diambil tindakan untuk mencadangkan kondisi tertentu terhadap activity sebelum melakukan perubahan orientasi. Untuk semua activity, sebaiknya simpan segala kondisi yang memang perlu untuk disimpan di dalam event onPause() sebelum nilai variable hilang ketika terjadi perubahan orientasi layar.



**Gambar 5.15 Perilaku variable saat pergantian orientasi**

Perilaku penting lain yang perlu dipahami adalah hanya view-view yang memiliki nama (penggunaan atribut android:id) dalam activity yang akan memiliki kondisi bertahan ketika activity yang didalamnya terdapat view-view terhapus. Sebagai contoh, user bisa jadi mengubah orientasi saat menuliskan suatu teks ke view EditText. Ketika hal ini terjadi, semua teks di dalam view EditText akan bertahan dan disimpan secara otomatis saat activity diciptakan ulang. Sebaliknya, jika Anda tidak memberi nama terhadap view EditText, activity tidak akan bisa mempertahankan teks yang terdapat pada view saat ini.

### **5.4.1 Mempertahankan Kondisi Informasi Selama Perubahan Konfigurasi**

Perubahan orientasi layar akan menghapus activity dan akan menciptakan ulang activity. Ketika activity diciptakan ulang, kondisi activity saat ini bisa hilang.

Activity dihapus bisa disebabkan oleh dua kejadian:

- **onPause()** — event ini selalu terpicu kapanpun ketika activity dimatikan atau ditempatkan di *background* (kondisi tak tampak).
- **onSaveInstanceState()** — event ini juga terpicu ketika activity dimatikan atau ditempatkan ke *background* seperti event onPause(), namun event onSaveInstanceState() tidak terpicu ketika activity sedang tidak di-load dari antrian (contoh: ketika user menekan tombol Back), karena tidak perlu mengembalikan kondisi sebelumnya.

Untuk mencadangkan kondisi activity, selalu terapkan event onPause() dan kemudian gunakan cara Anda sendiri untuk mencadangkan kondisi activity tersebut, misalnya dengan menggunakan database, media penyimpanan internal/eksternal, dan lain-lain.

Jika ingin mencadangkan kondisi activity secara lebih sederhana dan hendak menampilkan kembali kondisi sebelumnya saat activity diciptakan ulang (misalnya saat terjadi perubahan orientasi layar) maka lebih mudah bila menerapkan method onSaveInstanceState(). Method tersebut menyediakan object Bundle berupa argument sehingga Anda dapat menggunakanya untuk menyimpan kondisi activity

Kode berikut menunjukkan bagaimana menyimpan string ID ke object Bundle saat terjadi onSaveInstanceState:

```
@Override  
public void onSaveInstanceState(Bundle outState) {
```

```
//---simpan apapun yang perlu untuk dipertahankan ---
    outState.putString("ID", "1234567890");
    super.onSaveInstanceState(outState);
}
```

Ketika activity diciptakan ulang, event `onCreate` dijalankan terlebih dahulu, mengikuti event `onRestoreInstanceState()` yang akan mendapatkan kembali kondisi yang telah disimpan sebelumnya di dalam event `onSaveInstanceState` melalui argument object `Bundle`:

```
@Override
public void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    //---mendapatkan data/informasi yang dipertahankan sebelumnya-
    --
    String ID = savedInstanceState.getString("ID");
}
```

Meskipun Anda dapat menggunakan event `onSaveInstanceState()` untuk menyimpan kondisi informasi, tapi masih ada keterbatasan karena cara ini bukan penyelesaian memadai jika menyimpan struktur data yang kompleks.

Cara lain penanganan event adalah dengan menggunakan event `onRetainNonConfigurationInstance()`. Event ini terjadi ketika activity terhapus saat terjadi perubahan konfigurasi. Anda dapat menyimpan data saat ini dengan mengembalikannya ke dalam event `onRetainNonConfigurationInstance()`, seperti berikut ini:

```
@Override
public Object onRetainNonConfigurationInstance() {
    //---simpan apapun disini; ini mengambil tipe object---
    return("Sembarang teks yang dicadangkan");
}
```

Event ini akan mengembalikan tipe `Object` sehingga Anda bisa mengembalikan segala jenis tipe data. Untuk menguraikan data yang tersimpan dalam event `onCreate()` dengan menggunakan method `getLastNonConfigurationInstance()`, seperti berikut ini:

```
@Override
public void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
Log.d("StateInfo", "onCreate");
String str = (String) getLastNonConfigurationInstance();
}
```

#### 5.4.2 Mendeteksi Perubahan Orientasi

Terkadang kita perlu tahu informasi orientasi perangkat yang aktif saat ini, yaitu dengan menggunakan class WindowManager.

Contoh program berikut mendemonstrasikan bagaimana mendeteksi orientasi yang aktif saat ini:

```
import android.util.Log;
import android.view.Display;
import android.view.WindowManager;

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

    //---memperoleh informasi tampilan yang aktif saat ini---
    WindowManager wm = getWindowManager();
    Display d = wm.getDefaultDisplay();

    if (d.getWidth() > d.getHeight())
    {
        //---mode landscape---
        Log.d("Orientasi", "mode Landscape");
    }
    else
    {
        //--- mode portrait---
        Log.d("Orientasi", "mode Portrait");
    }
}
```

Method `getDefaultDisplay()` mengembalikan object **Display** mewakili layar perangkat, dari situ didapatkan lebar dan tinggi layar, sehingga dapat diambil kesimpulan orientasi apa yang aktif saat ini.

#### 5.4.3 Mengendalikan Orientasi Activity

Terkadang pula kita ingin memastikan bahwa aplikasi hanya ditampilkan pada orientasi tertentu, misalnya saat menuliskan kode program untuk

game yang seharusnya ditampilkan dalam mode landscape. Dalam kasus ini dapat dipaksakan perubahan orientasi-nya secara program dengan menggunakan method setRequestOrientation() pada class Activity:

```
import android.content.pm.ActivityInfo;

public class MainActivity extends Activity {
    /** terpanggil saat activity diciptakan pertamakali.*/
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        //---ubah ke mode landscape ---

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
    }
}
```

Untuk mengubah ke mode portrait, gunakan konstanta ActivityInfo.SCREEN\_ORIENTATION\_PORTRAIT :

```
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
```

Disamping penggunaan method setRequestOrientation() bisa pula digunakan atribut android:screenOrientation pada element <activity> di dalam file AndroidManifest.xml untuk memaksa activity agar berorientasi tertentu:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.learn2develop.Orientations"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".MainActivity"
            android:label="@string/app_name"
            android:screenOrientation="landscape" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```

```
<uses-sdk android:minSdkVersion="9" />
</manifest>
```

contoh di atas adalah pemaksaan activity agar ke orientasi tertentu dan mencegah activity dari usaha penghapusan, oleh karenanya activity tidak akan dihapus, dan event `onCreate()` tidak akan bekerja lagi ketika orientasi perangkat berubah.

#### 5.4.4 Menciptakan UI Secara Program

Pendekatan lain dalam membuat UI selain XML yang bersifat statis adalah membuat UI secara dinamis yang terbentuk saat program berjalan (*run-time*).

Buatlah project baru bernama **KodeUI**.

<b>Project name</b>	KodeUI
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	KodeUI
<b>Package name</b>	net.belajar2android.KodeUI
<b>Create Activity</b>	KodeUIActivity

Tambahkan pernyataan berhuruf tebal di bawah ini ke dalam file **KodeUIActivity.java**:

```
package net.belajar2android.KodeUI;

import android.app.Activity;
import android.os.Bundle;

import android.view.ViewGroup.LayoutParams;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;

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

        //---parameter untuk view-view---
```

```

LayoutParams params = new
LinearLayout.LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.WRAP_CONTENT);

//---menciptakan layout---
LinearLayout layout = new LinearLayout(this);
layout.setOrientation(LinearLayout.VERTICAL);

//---menciptakan textView---
TextView tv = new TextView(this);
tv.setText("Ini adalah TextView");
tv.setLayoutParams(params);

//---menciptakan button---
Button btn = new Button(this);
btn.setText("Ini adalah Button");
btn.setLayoutParams(params);

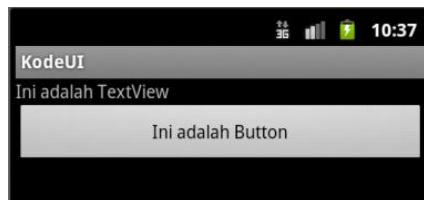
//---menambahkan textView---
layout.addView(tv);

//---menambahkan button---
layout.addView(btn);

//---menciptakan parameter layout untuk layout---
LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.WRAP_CONTENT);
    this.addContentView(layout, layoutParams);
}
}

```

Jalankan aplikasi ke emulator android. Hasilnya kurang lebih seperti berikut ini



**Gambar 5.16 Membentuk UI secara run-time**

Bagaimana program bekerja? Pertama memberi tanda komentar (//) untuk mematikan pernyataan setContentView() sehingga UI yang berasal dari file main.xml tidak di-load. Selanjutnya menciptakan object

**LayoutParams** untuk menentukan parameter layout yang dapat digunakan oleh view-view lainnya:

```
//---parameter untuk view-view---
LayoutParams params = new
LinearLayout.LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.WRAP_CONTENT);
```

Ciptakan object LinearLayout untuk diisi semua view-view dalam Activity:

```
//---menciptakan layout---
LinearLayout layout = new LinearLayout(this);
layout.setOrientation(LinearLayout.VERTICAL);
```

Buat view TextView dan Button:

```
//---menciptakan textview---
TextView tv = new TextView(this);
tv.setText("Ini adalah TextView");
tv.setLayoutParams(params);

//---menciptakan button---
Button btn = new Button(this);
btn.setText("Ini adalah Button");
btn.setLayoutParams(params);
```

Tambahkan view-view ke object LinearLayout:

```
//---menambahkan textview---
layout.addView(tv);

//---menambahkan button---
layout.addView(btn);
```

Ciptakan object LayoutPArms untuk dipakai object LinearLayout:

```
//---menciptakan parameter layout untuk layout---
LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.WRAP_CONTENT);
```

Terakhir, tambahkan object LinearLayout ke activity:

```
this.addContentView(layout, layoutParams);
```

Kode program untuk UI dinamis lumayan sulit sehingga gunakan bila diperlukan saja.

## 5.5 Menangkap Notifikasi UI

Ada dua tingkatan bagaimana user berinteraksi dengan UI, yaitu tingkat activity dan tingkat view. Pada tingkat activity, class Activity menyediakan method-method yang dapat meng-override-nya.

Beberapa method yang dapat meng-override di dalam activity antara lain:

- **onKeyDown** — terpanggil ketika key ditekan.
- **onKeyUp** — terpanggil ketika key dilepas.
- **onMenuItemSelected** — terpanggil jika menu panel dipilih.
- **onMenuOpened** — terpanggil jika menu panel dibuka.

### 5.5.1 Meng-override Method Yang Terdefinisi Di Dalam Activity

Untuk memahami bagaimana activity berinteraksi dengan user, mulailah dengan meng-override beberapa method yang terdefinisi dalam class dasar activity.

Buat android project bernama **UIActivity**.

<b>Project name</b>	UIActivity
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	UIActivity
<b>Package name</b>	net.belajar2android.UIActivity
<b>Create Activity</b>	UIActivity

Tambahkan pernyataan berhuruf tebal ini ke dalam file main.xml

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

```

<TextView
    android:layout_width="214dp"
    android:layout_height="wrap_content"
    android:text="Nama Anda" />
<EditText
    android:id="@+id/txt1"
    android:layout_width="214dp"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id	btn1"
    android:layout_width="106dp"
    android:layout_height="wrap_content"
    android:text="OK" />
<Button
    android:id="@+id	btn2"
    android:layout_width="106dp"
    android:layout_height="wrap_content"
    android:text="Batal" />
</LinearLayout>

```

Tambahkan pernyataan berikut ke dalam file UIActivityActivity.java:

```

package net.belajar2android.UIActivity;

import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.widget.Toast;

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

    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        switch (keyCode) {
            case KeyEvent.KEYCODE_DPAD_CENTER:
                Toast.makeText(getApplicationContext(), "Tombol tengah telah
diklik", Toast.LENGTH_LONG).show();
                break;
            case KeyEvent.KEYCODE_DPAD_LEFT:
                Toast.makeText(getApplicationContext(), "Panah kiri telah
diklik", Toast.LENGTH_LONG).show();
                break;
            case KeyEvent.KEYCODE_DPAD_RIGHT:
                Toast.makeText(getApplicationContext(), "Panah kanan telah
diklik", Toast.LENGTH_LONG).show();
                break;
            case KeyEvent.KEYCODE_DPAD_UP:

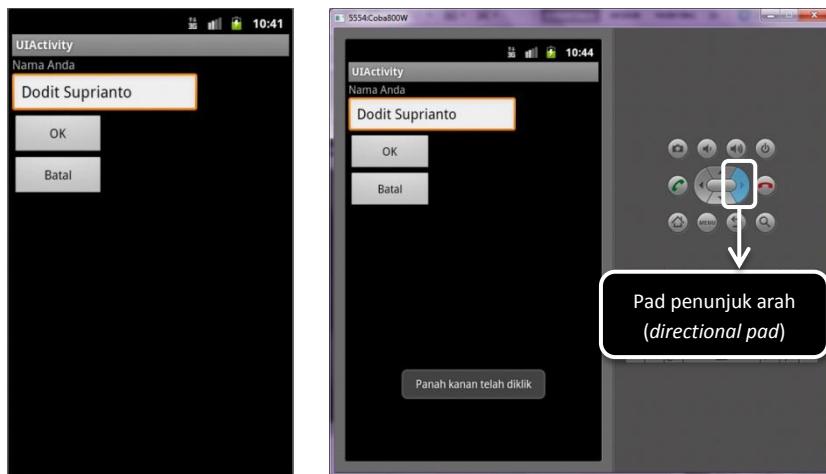
```

```

        Toast.makeText(getApplicationContext(), "Panah atas telah
diklik", Toast.LENGTH_LONG).show();
        break;
    case KeyEvent.KEYCODE_DPAD_DOWN:
        Toast.makeText(getApplicationContext(), "Panah bawah telah
diklik", Toast.LENGTH_LONG).show();
        break;
    }
    return false;
}
}

```

Jalankan aplikasi ke Emulator, jika Emulator Android sudah me-load activity, selanjutnya tuliskan sembarang teks pada EditText, klik panah pada pad penunjuk arah (*directional pad*) dan amati pesan yang muncul.



**Gambar 5.17 Interaksi UI dan User**

Bagaimana program bekerja? Ketika activity di-load, cursor akan berkedip pada view EditText yang menandakan bahwa EditText menjadi pusatnya. Di dalam class UIActivityActivity.java Anda meng-override method onKeyDown() dari class dasar Activity, seperti berikut:

```

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    switch (keyCode) {
        case KeyEvent.KEYCODE_DPAD_CENTER:

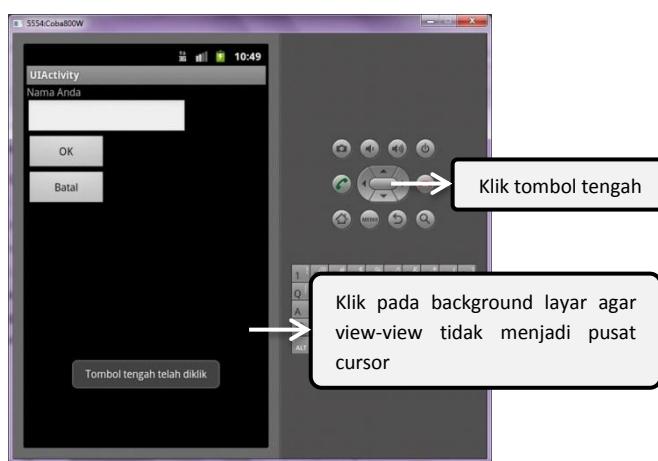
```

```
//...
break;
case KeyEvent.KEYCODE_DPAD_LEFT:
//...
break;
case KeyEvent.KEYCODE_DPAD_RIGHT:
//...
break;
case KeyEvent.KEYCODE_DPAD_UP:
//...
break;
case KeyEvent.KEYCODE_DPAD_DOWN:
//...
break;
}
return false;
}
```

Apapun key yang ditekan maka view yang aktif saat ini (yang menjadi focus/pusat) akan menangani event secara otomatis. Dalam kasus ini, ketika EditText menjadi focus dan Anda menekan key, maka view EditText akan menangani event dan menampilkan karakter yang baru saja dituliskan. Bila Anda menekan tombol panah atas atau bawah, maka view EditText tidak menanganinya namun melewatkannya ke activity. Dalam kasus ini method onKeyDown() dipanggil. Di sini Anda mendandai key bahwa telah ditekan dan ditampilkan pesan yang menunjukkan bahwa key ditekan. Perhatikan bahwa pusatnya sekarang juga dikirim ke view berikutnya, yaitu button OK.

Menariknya, jika view EditText telah mempunyai beberapa teks di dalamnya dan cursor berada di posisi akhir teks (paling kanan teks), kemudian panah kiri diklik maka key tidak menjalankan event onKeyDown(), tapi malah menggerakan cursor satu karakter ke kiri, ini terjadi karena view EditText telah siap menangani event. Jika Anda menekan panah kanan maka method onKeyDown() akan dipanggil (karena sekarang view EditText tidak akan menangani event). Tindakan yang sama berlaku ketika cursor berada diawal view EditText. Klik panah kiri maka event onKeyDown() akan bekerja, dan jika panah kanan diklik maka cursor akan bergerak ke kanan.

Saat tombol **OK** menjadi pusatnya, tekan tombol *center* pad penunjuk arah. Amati bahwa pesan “**Tombol tengah telah diklik**” tidak ditampilkan. Ini terjadi karena view Button sedang menangani event klik terhadap dirinya sendiri. Oleh sebab itu event tidak tertangkap oleh method `onKeyDown()`. Namun jika tidak satu pun view lain menjadi pusat cursor (dengan mengklik background kosong dari layar), kemudian klik key tengah (center pad) maka akan muncul pesan “**Tombol tengah telah diklik**”, seperti tampak gambar di bawah ini:



**Gambar 5.18 Interaksi UI dan user**

Method `onKeyDown()` akan mengembalikan nilai Boolean. Sebaiknya kembalikan nilai menjadi true ketika hendak memberitahukan system bahwa telah selesai berurusan dengan event agar system tidak memprosesnya lebih jauh.

Perhatikan pernyataan **case** ketika Anda mengembalikan nilai true, setelah key menemui kesesuaian:

```
@Override  
public boolean onKeyDown(int keyCode, KeyEvent event) {  
    switch (keyCode) {  
        case KeyEvent.KEYCODE_DPAD_CENTER:  
            return true;  
    }  
    return super.onKeyDown(keyCode, event);  
}
```

```

        Toast.makeText(getApplicationContext(), "Tombol tengah telah
diklik", Toast.LENGTH_LONG).show();
    return true;
    case KeyEvent.KEYCODE_DPAD_LEFT:
        Toast.makeText(getApplicationContext(), "Panah kiri telah diklik",
Toast.LENGTH_LONG).show();
    return true;
    case KeyEvent.KEYCODE_DPAD_RIGHT:
        Toast.makeText(getApplicationContext(), "Panah kanan telah diklik",
Toast.LENGTH_LONG).show();
    return true;
    case KeyEvent.KEYCODE_DPAD_UP:
        Toast.makeText(getApplicationContext(), "Panah atas telah diklik",
Toast.LENGTH_LONG).show();
    return true;
    case KeyEvent.KEYCODE_DPAD_DOWN:
        Toast.makeText(getApplicationContext(), "Panah bawah telah diklik",
Toast.LENGTH_LONG).show();
    return true;
}
return false;
}

```

Sekarang Anda tidak dapat berpindah antar view dengan menggunakan key panah, karena setiap kali key menemui kesesuaian maka nilai true dikembalikan dan pernyataan di bawahnya tidak dilanjutkan.

### 5.5.2 Me-Register Event Untuk View

View dapat menjalankan event ketika user berinteraksi antar keduanya. Contoh, saat user menyentuh view Button, maka event segera dilayani untuk melakukan tindakan yang sesuai. Untuk melakukannya diperlukan *registry event* untuk view.

Gunakan project sebelumnya (UIActivity), dimana activity mempunyai dua view Button, kemudian Anda me-registry event tombol klik dengan menggunakan class *anonymous* seperti tampak berikut ini:

```

package net.belajar2android.UIActivity;

import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.widget.Toast;
import android.view.View;
import android.view.View.OnClickListener;

```

```

import android.widget.Button;

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

        //---dua tombol terkait ke penanganan event yang sama---
        Button btn1 = (Button) findViewById(R.id.btn1);
        btn1.setOnClickListener(btnListener);

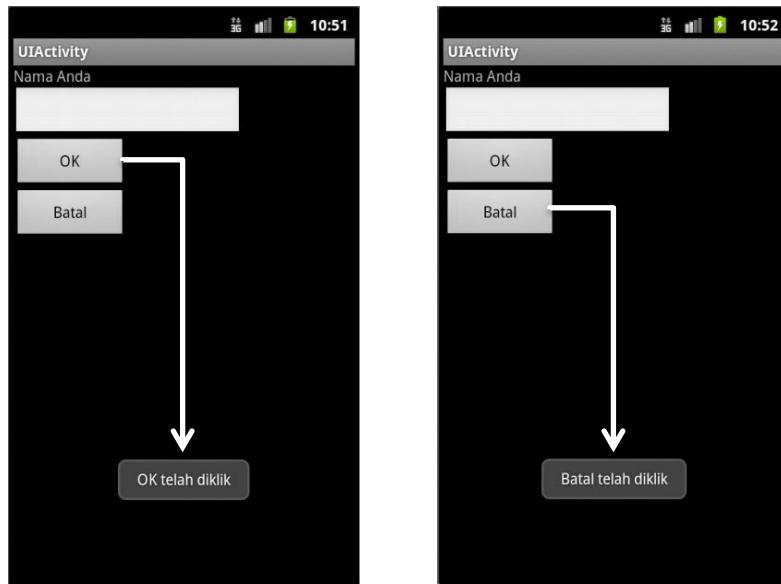
        Button btn2 = (Button) findViewById(R.id.btn2);
        btn2.setOnClickListener(btnListener);
    }

    //---menciptakan class anonymous agar bertindak sebagai button
    //click listener---
    private OnClickListener btnListener = new OnClickListener() {
        public void onClick(View v) {
            Toast.makeText(getApplicationContext(), ((Button) v).getText() +
" telah diklik", Toast.LENGTH_LONG).show();
        }
    };

    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        switch (keyCode) {
            //...
            //...
        }
        return false;
    }
}

```

Jika tombol OK atau Batal ditekan maka akan muncul pesan yang sesuai dengan tombolnya. Ini membuktikan bahwa event telah terkait dengan benar.



**Gambar 5.19 Meregister event ke view**

Disamping pendefinisian anonymous class, ada pula pendefinisian anonymous inner class untuk penanganan event. Contoh berikut menunjukkan bagaimana penggunaan event onFocusChange pada view EditText:

```
package net.belajar2android.UIActivity;

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

import android.widget.EditText;

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

```

//---dua tombol terkait ke penanganan event yang sama---
Button btn1 = (Button) findViewById(R.id.btn1);
btn1.setOnClickListener(btnListener);

Button btn2 = (Button) findViewById(R.id.btn2);
btn2.setOnClickListener(btnListener);

EditText txt1 = (EditText) findViewById(R.id.txt1);

//---menciptakan anonymous inner class yang bertindak
//sebagai onfocus listener---
txt1.setOnFocusChangeListener(new
View.OnFocusChangeListener() {
    @Override
    public void onFocusChange(View v, boolean hasFocus) {

        Toast.makeText(getApplicationContext(), ((EditText) v).getId() +
" menjadi pusat - " + hasFocus, Toast.LENGTH_LONG).show();
    }
});

//---menciptakan class anonymous agar bertindak sebagai button
//click listener---
private OnClickListener btnListener = new OnClickListener() {
    public void onClick(View v) {
        Toast.makeText(getApplicationContext(), ((Button) v).getText() +
" telah diklik", Toast.LENGTH_LONG).show();
    }
};

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    switch (keyCode) {
        //..
        //..
    }
    return false;
}
}

```

Ketika EditText menjadi pusat/focus maka pesan akan ditampilkan:



**Gambar 5.20 anonymous inner class**

Dari sini dapat disimpulkan bagaimana UI diciptakan di dalam Android, memahami perbedaan layout dan bagaimana memosisikan view di dalam UI Android, juga bagaimana UI bisa beradaptasi terhadap perubahan orientasi layar. Berikut ini ringkasannya:

Topik	Konsep Penting
LinearLayout	Menyusun view-view dalam satu kolom atau satu baris.
AbsoluteLayout	Menentukan lokasi pasti view-view anak.
TableLayout	Mengelompokkan view-view menjadi baris dan kolom.
RelativeLayout	Menentukan bagaimana view-view anak diposisikan terhadap view yang lain.
FrameLayout	Satu wadah pada layar bagi view-view lain,

	dimana Anda dapat menampilkannya menjadi satu view.
ScrollView	Tipe khusus dari FrameLayout, dimana user dapat meng gulung melalui list view agar menempati ruang kosong, menyesuaikan ukuran layar sesungguhnya.
Satuan pengukuran	Penentuan satuan ukuran.
Dua cara mengadaptasi perubahan orientasi	Anchoring, resizing dan repositioning
Penggunaan file XML berbeda untuk orientasi yang berbeda	Menggunakan folder layout untuk UI portrait dan folder layout-land UI lanscpae.
Tiga cara mempertahankan kondisi activity	Menggunakan event onPause(). Menggunakan event onSaveInstanceState(). Menggunakan event onRetainNonConfigurationInstance().
Mendapatkan ukuran layar saat ini pada perangkat	Menggunakan method dari class WindowManager yaitu method getDefaultDisplay().
Pembatasan orientasi activity	Gunakan method setRequestedOrientation() atau atribut android:screenOrientation yang ada dalam file AndroidManifest.xml.

# BAB 6

# MERANCANG UI DENGAN VIEW

Bab ini akan membahas berbagai macam view yang dapat digunakan untuk merancang antarmuka aplikasi., khususnya view group, antara lain **View Dasar** (misalnya TextView, EditText dan Button), **View Picker**, (TimePicker dan DatePicker), dan **View List** (ListView, SpinnerView).

## 6.1 View-view Dasar

Terdapat view-view dasar yang dapat digunakan untuk merancang UI bagi aplikasi Android: TextView, EditText, Button, ImageButton, CheckBox, ToggleButton, RadioButton, RadioGroup.

### 6.1.1 View TextView

Ketika project Android tercipta maka Eclipse selalu membentuk file main.xml yang berisi elemen <TextView>:

```
<?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="@string/hello" />
</LinearLayout>
```

TextView digunakan untuk menampilkan teks ke user. TextView adalah view dasar yang paling sering ditemui dan digunakan saat pengembangan aplikasi. Jika Anda mengizinkan user untuk mengubah isi teks yang tampil, maka sebaiknya gunakan subclass dari TextView yaitu **EditText**.

**Catatan:**

Pada beberapa platform, TextView secara umum dikenal sebagai Label

### **6.1.2 View Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton Dan RadioGroup**

Disamping view TextView terdapat view Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton dan RadioGroup:

- **Button** — mewakili tombol yang dapat ditekan.
- **ImageButton** — serupa dengan view Button tetapi dapat menampilkan gambar.
- **EditText** — subclass dari view TextView yang mengizinkan user untuk mengubah teks yang terkandung di dalamnya.
- **CheckBox** — tombol yang memiliki tipe khusus yang terdiri dari dua kondisi: **checked** atau **unchecked** (tercentang atau tidak tercentang).
- **RadioGroup** dan **RadioButton** — RadioButton mempunyai dua status, yaitu checked ataupun unchecked. RadioGroup digunakan untuk mengelompokkan secara bersama satu atau lebih view RadioButton, namun hanya ada satu RadioButton yang dipilih diantara RadioButton lain, tapi masih berada dalam satu kelompok RadioGroup.
- **ToggleButton** — Menampilkan kondisi checked/unchecked dengan menggunakan penanda tertentu.

Buatlah project baru bernama **ViewDasar1**

<b>Project name</b>	ViewDasar1
<b>Build Target</b>	Android 2.3.3, API 10

<b>Application name</b>	ViewDasar1
<b>Package name</b>	net.belajar2android.ViewDasar1
<b>Create Activity</b>	MainActivity

Tambahkan elemen yang berhuruf tebal ke dalam file main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<Button android:id="@+id/btnSimpan"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Simpan" />

<Button android:id="@+id/btnBuka"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Buka" />

<ImageButton android:id="@+id/btnImg1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:src="@drawable/icon" />

<EditText android:id="@+id/txtNama"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />

<CheckBox android:id="@+id/chkSimpanOtomatis"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Simpan Otomatis" />

<CheckBox android:id="@+id/bintang"
    style="?android:attr/starStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<RadioGroup android:id="@+id/rdGp1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
    <RadioButton android:id="@+id/rdb1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Pilihan 1" />
    <RadioButton android:id="@+id/rdb2"
```

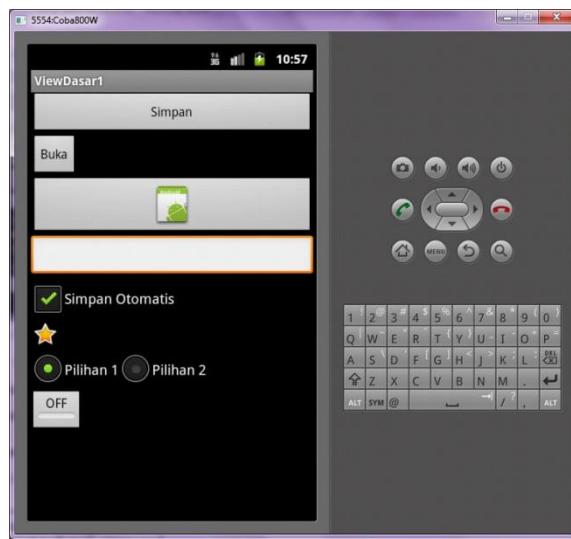
```

        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Pilihan 2" />
    </RadioGroup>

<ToggleButton android:id="@+id/toggle1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>

```

Jalankan aplikasi Android ke emulator dengan menekan tombol F11, klik pada beberapa view, maka hasilnya kurang lebih tampak seperti berikut:



**Gambar 6.1 Menampilkan View-view dasar**

Bagaimana program bekerja? Sejauh ini semua view relatif jelas, view-view tersusun dalam elemen <LinearLayout> sehingga view-view menumpuki satu sama lain mulai dari atas (dalam satu kolom) ditampilkan dalam activity.

Button pertama mempunyai atribut layout\_width diisi fill\_parent sehingga lebarnya memenuhi seluruh lebar layar.

```
<Button android:id="@+id/btnSimpan"
    android:layout_width="fill_parent"
```

```
    android:layout_height="wrap_content"
    android:text="Simpan" />
```

Button kedua, atribut layout\_width diisi wrap\_content sehingga lebar tombol mengikuti lebar isi tombol (tulisan “Buka”).

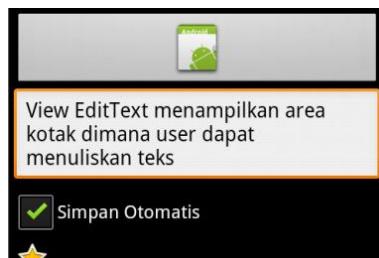
```
<Button android:id="@+id/btnBuka"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Buka" />
```

ImageButton menampilkan tombol beserta gambar. Gambar diisi melalui atribut src, gambar berasal dari icon aplikasi:

```
<ImageButton android:id="@+id/btnImg1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:src="@drawable/icon" />
```

View EditText menampilkan area kotak dimana user dapat menuliskan teks. layout\_height diisi wrap\_content sehingga ketika user menuliskan teks yang panjang secara otomatis tingginya akan membesar menyesuaikan tinggi teks keseluruhan.

```
<EditText android:id="@+id/txtNama"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
```



**Gambar 6.2 Tinggi EditText Otomatis Meninggi**

View CheckBox menampilkan sebuah checkbox dimana user dapat memberi tanda centang menjadi **check** atau **uncheck**:

```
<CheckBox android:id="@+id/chkSimpanOtomatis"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
```

```
        android:text="Simpan Otomatis" />
```

Anda dapat menerapkan atribut **style** untuk menampilkan CheckBox berbentuk *image*, contohnya adalah gambar bintang:

```
<CheckBox android:id="@+id/bintang"
    style="?android:attr/starStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

Adapun format nilai atribut **style** adalah `?[package:] [type:]name`.

RadioGroup melingkupi dua RadioButton. Ini penting dipahami karena radiobutton biasa digunakan untuk mewakili pilihan ganda agar user bisa memilih. Ketika RadioButton dalam RadioGroup dipilih, maka semua RadioButton lainnya secara otomatis tidak terpilih:

```
<RadioGroup android:id="@+id/rdGp1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
    <RadioButton android:id="@+id/rdb1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Pilihan 1" />
    <RadioButton android:id="@+id/rdb2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Pilihan 2" />
</RadioGroup>
```

RadioButton tersusun secara vertical, satu menutupi yang lain. Jika ingin menyusun RadioButton secara horizontal, maka perlu mengubah atribut orientasi menjadi **horizontal** dan pastikan bahwa atribut **layout\_width** RadioButton diisi **wrap\_content**:

```
<RadioGroup android:id="@+id/rdGp1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <RadioButton android:id="@+id/rdb1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pilihan 1" />
    <RadioButton android:id="@+id/rdb2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:text="Pilihan 2" />
    </RadioGroup>
```



**Gambar 6.3 RadioButton disusun secara horisontal**

ToggleButton menampilkan tombol kotak dimana user dapat beralih dari **on** menjadi **off** atau sebaliknya:

```
<ToggleButton android:id="@+id/toggle1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

Satu hal yang selalu ada disetiap view pada contoh di atas adalah atribut **id** yang diisi dengan nilai tertentu, contoh:

```
<Button android:id="@+id/btnSimpan"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Simpan" />
```

Atribut id adalah sebuah indentifikasi view yang akan dikenali oleh method melalui `View.findViewById()` atau `Activity.findViewById()`.

Berikut cara mengendalikan view-view di atas melalui program.

Gunakan project yang sama (**ViewDasar1**), modifikasi pernyataan yang ada di dalam **MainActivity.java**, seperti berikut ini:

```
package net.belajar2android.ViewDasar1;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;
```

```
import android.widget.ToggleButton;
import android.widget.RadioGroup.OnCheckedChangeListener;

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

        //---view Button---
        Button btnBuka = (Button) findViewById(R.id.btnBuka);
        btnBuka.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {
                DisplayToast("Anda telah mengklik tombol Buka");
            }
        });

        //---view Button---
        Button btnSimpan = (Button) findViewById(R.id.btnSimpan);
        btnSimpan.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {
                DisplayToast("Anda telah mengklik tombol Simpan");
            }
        });

        //---view CheckBox---
        CheckBox checkBox = (CheckBox)
        findViewById(R.id.chkSimpanOtomatis);
        checkBox.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                if (((CheckBox) v).isChecked())
                    DisplayToast("CheckBox dicentang");
                else
                    DisplayToast("CheckBox tidak dicentang");
            }
        });

        //---view RadioButton---
        RadioGroup radioGroup = (RadioGroup)
        findViewById(R.id.rdGp1);
        radioGroup.setOnCheckedChangeListener(new
        OnCheckedChangeListener() {
            public void onCheckedChanged(RadioGroup group, int
            checkedId) {
                RadioButton rb1 = (RadioButton) findViewById(R.id.rdb1);
                if (rb1.isChecked()) {
                    DisplayToast("Pilihan 1 dipilih!");
                }
                else {
                    DisplayToast("Pilihan 2 dipilih!");
                }
            }
        });
    }
}
```

```

        });

//---view ToggleButton---
ToggleButton toggleButton = (ToggleButton)
findViewById(R.id.toggle1);
toggleButton.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if (((ToggleButton) v).isChecked())
            DisplayToast("Tombol Toggle On");
        else
            DisplayToast("Tombol Toggle Off");
    }
});

private void DisplayToast(String pesan) {
    Toast.makeText(getApplicationContext(), pesan,
    Toast.LENGTH_LONG).show();
}
}

```

Jalankan aplikasi, kemudian klik beberapa view, maka pesan Toast akan ditampilkan.



*Gambar 6.4 Mengendalikan View*

Bagaimana program bekerja? Pertama, harus memprogram view bersangkutan yang dibuat saat event `onCreate()`. Kemudian gunakan method `Activity.findViewById()` dan memberinya ID:

```
//---view Button---
Button btnBuka = (Button) findViewById(R.id.btnBuka);
Method setOnClickListener() me-register callback untuk digunakan
nantinya saat view diklik:
btnBuka.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        DisplayToast("Anda telah mengklik tombol Buka");
    }
});
```

Method `onClick()` dipanggil ketika view diklik.

Pada CheckBox terdapat pernyataan kondisi untuk mengetahui status CheckBox yang diletakkan dalam method `onClick()`. Status setiap CheckBox saling berkebalikan, karena hanya terdapat dua CheckBox maka cukup digunakan pernyataan If...else...

```
//---view CheckBox---
CheckBox checkBox = (CheckBox)
findViewById(R.id.chkSimpanOtomatis);
checkBox.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if (((CheckBox) v).isChecked())
            DisplayToast("CheckBox dicentang");
        else
            DisplayToast("CheckBox tidak dicentang");
    }
});
```

Untuk RadioButton digunakan method `setOnCheckedChangeListener()` pada RadioGroup untuk me-register sebuah *callback* untuk dilibatkan ketika status centang RadioButton berubah di dalam group:

```
//---view RadioButton---
RadioGroup radioGroup = (RadioGroup) findViewById(R.id.rdgP1);
radioGroup.setOnCheckedChangeListener(new
OnCheckedChangeListener() {
    public void onCheckedChanged(RadioGroup group, int checkedId)
    {
        RadioButton rbl = (RadioButton) findViewById(R.id.rdb1);
        if (rbl.isChecked()) {
            DisplayToast("Pilihan 1 dipilih!");
```

```

        }
        else {
            DisplayToast("Pilihan 2 dipilih!");
        }
    });
}
);

```

Ketika RadioButton dipilih, maka method onCheckedChanged() dijalankan, dimana di dalamnya terdapat RadioButton. Kemudian method isChecked() dipanggil untuk menentukan RadioButton mana yang dipilih.

Pada argument kedua method onCheckedChanged (**int checkedId**) merupakan identifikasi unik dari RadioButton yang dipilih.

### 6.1.3 View ProgressBar

View ProgressBar menghasilkan umpan balik secara visual dari suatu pekerjaan yang sedang berproses.

Buatlah project Android bernama **ViewDasar2**.

<b>Project name</b>	ViewDasar2
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	ViewDasar2
<b>Package name</b>	net.belajar2android.ViewDasar2
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```

<?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" >
    <ProgressBar android:id="@+id/progressbar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>

```

Modifikasi file MainActivity.java menjadi seperti berikut ini:

```
package net.belajar2android.ProgressBar;

import android.app.Activity;
import android.os.Bundle;
import android.os.Handler;
import android.app.Activity;
import android.widget.ProgressBar;

public class MainActivity extends Activity {
    private static int progress;
    private ProgressBar progressBar;
    private int progressStatus = 0;
    private Handler handler = new Handler();

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

        progress = 0;
        progressBar = (ProgressBar) findViewById(R.id.progressbar);

        //---lakukan beberapa pekerjaan di background thread---
        new Thread(new Runnable() {
            public void run() {
                //---lakukan beberapa pekerjaan disini---
                while (progressStatus < 10) {
                    progressStatus = doSomeWork();
                }

                //---sembunyikan progress bar---
                handler.post(new Runnable() {
                    public void run() {
                        //---0 - TAMPAK; 4 - TAK TAMPAK; 8 - HILANG---
                        progressBar.setVisibility(0);
                    }
                });
            }
        });
    }

    //---lakukan perkerjaan yang lama disini---
    private int doSomeWork() {
        try {
            //---simulasi sedang melakukan pekerjaan---
            Thread.sleep(500);
        } catch (InterruptedException e) {
        {
            e.printStackTrace();
        }
        return ++progress;
    }
    }).start();
}
}
```



**Gambar 6.5 Progress Bar Lingkaran**

Bagaimana program bekerja? Ketika proses telah selesai lengkap, maka ProgressBar disembunyikan dengan memberi nilai property Visibility “HILANG” (nilai 8). Perbedaan antara konstanta “TAK TAMPAK” dan “HILANG” adalah bahwa konstanta “TAK TAMPAK” hanya menghilangkan ProgressBar (area yang ditempati ProgressBar masih mengambil ruang kosong di dalam activity). Konstanta “HILANG” membuang view ProgressBar dari activity dan tidak mengambil ruang kosong manapun di dalam activity.

Cobalah program berikut ini, yang menunjukkan bagaimana Anda dapat mengubah tampilan ProgressBar hasil kustomisasi.

Gunakan project yang sama dengan sebelumnya (ViewDasar2).

Modifikasi file main.xml seperti berikut ini:

```
<?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" >
    <ProgressBar android:id="@+id/progressbar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        style="?android:attr/progressBarStyleHorizontal" />
</LinearLayout>
```

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.ProgressBar;

import android.app.Activity;
```

```
import android.os.Bundle;
import android.os.Handler;
import android.app.Activity;
import android.widget.ProgressBar;

public class MainActivity extends Activity {
    private static int progress;
    private ProgressBar progressBar;
    private int progressStatus = 0;
    private Handler handler = new Handler();

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

        progress = 0;
        progressBar = (ProgressBar) findViewById(R.id.progressbar);
        progressBar.setMax(200);

        //---lakukan beberapa pekerjaan di background thread---
        new Thread(new Runnable() {
            public void run() {
                //---lakukan beberapa pekerjaan disini---
                while (progressStatus < 100) {
                    progressStatus = doSomeWork();

                    //---update progress bar---
                    handler.post(new Runnable() {
                        public void run() {
                            progressBar.setProgress(progressStatus);
                        }
                    });
                }
            }
        });

        //---sembunyikan progress bar---
        handler.post(new Runnable() {
            public void run() {
                //---0 - TAMPAK; 4 - TAK TAMPAK; 8 - Hilang---
                progressBar.setVisibility(8);
            }
        });
    }

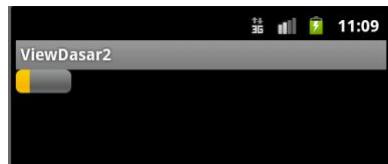
    //---lakukan perkerjaan yang luarbiasa.... disini---
    private int doSomeWork() {
        try {
            //---simulasi sedang melakukan pekerjaan---
            Thread.sleep(50);
        } catch (InterruptedException e)
        {
            e.printStackTrace();
        }
        return ++progress;
    }
}
```

```

        }
    }).start();
}
}

```

Jalankan aplikasi ke emulator dengan menekan tombol keyboard F11.



**Gambar 6.6 ProgressBar akan menghilang ketika proses mencapai 50%**

Bagaimana program bekerja? Untuk membuat ProgressBar tampil secara horizontal, maka atur atributnya menjadi **?android:attr/progressBarStyleHorizontal**, seperti berikut ini:

```
<ProgressBar android:id="@+id/progressbar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="?android:attr/progressBarStyleHorizontal" />
```

Untuk menampilkan kemajuan proses, panggil method `setProgress()` dengan melewatkannya sebuah nilai integer yang menandakan suatu kemajuan proses:

```
//---update progress bar---
handler.post(new Runnable() {
    public void run() {
        progressBar.setProgress(progressStatus);
    }
});
```

Pada contoh di atas, rentang nilai ProgressBar dimulai dari 0 sampai 200 (melalui method `setMax()`). Oleh karena itu Progress Bar akan berhenti, dan kemudian menghilang ketika ProgressBar setengah jalan dari seharusnya (pemanggilan method `doSomeWork()` selama `progressStatus` kurang dari 100). Untuk memastikan bahwa ProgressBar menghilang ketika kemajuan proese mencapai 100%, maka atur maksimumnya

menjadi 100, atau modifikasi perulangan while untuk menghentikan progressStatus mencapai 200:

```
//---lakukan beberapa pekerjaan disini---
while (progressStatus < 200)
```



Gambar 6.7 ProgressBar berporses 100%

#### 6.1.4 View AutoCompleteTextView

AutoCompleteTextView adalah view yang serupa dengan EditText tetapi ia bisa menampilkan daftar saran-saran untuk melengkapi tulisan yang sedang ditulis oleh user secara otomatis.

Buatlah project Android bernama **ViewDasar3**.

<b>Project name</b>	ViewDasar3
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	ViewDasar3
<b>Package name</b>	net.belajar2android.ViewDasar3
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:padding="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Country" />
    <AutoCompleteTextView android:id="@+id/autocomplete_country"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"/>
```

```
</LinearLayout>
```

Buat copy file main.xml dan ubah namanya menjadi **list\_item.xml**, kemudian tambahkan pernyataan XML seperti berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:padding="10dp"
        android:textSize="16sp"
        android:textColor="#000">
</TextView>
```

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.ViewDasar3;

import android.app.Activity;
import android.os.Bundle;

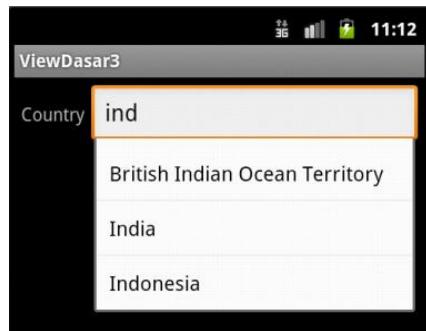
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;

public class MainActivity extends Activity {

    static final String[] COUNTRIES = new String[] { "Afghanistan",
    "Albania", "Algeria", "American Samoa", "Andorra", "Angola",
    "Anguilla", "Antarctica", "Daftar lengkap negara ada pada contoh
    program di CD penyerta buku"};

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

        AutoCompleteTextView textView = (AutoCompleteTextView)
        findViewById(R.id.autocomplete_country);
        ArrayAdapter<String> adapter = new
        ArrayAdapter<String>(this, R.layout.list_item, COUNTRIES);
        //textView.setThreshold(3);
        textView.setAdapter(adapter);
    }
}
```



Gambar 6.8 Autocomplete TextView

## 6.2 View Picker

Memilih tanggal dan waktu adalah salah satu kegiatan yang sering dilakukan dalam aplikasi bergerak. Android mendukung fungsi ini melalui view TimePicker dan DatePicker.

### 6.2.1 View TimePicker

Dengan TimePicker user dapat memilih waktu dalam suatu hari, dimana mode bisa diatur menjadi format jam 24 maupun AM/PM.

Buat project Android bernama **ViewDasar4**.

<b>Project name</b>	ViewDasar4
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	ViewDasar4
<b>Package name</b>	net.belajar2android.ViewDasar4
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```
<?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" >
```

```

<TimePicker android:id="@+id/timePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<Button android:id="@+id	btnSet"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Saya telah diatur!"/>
</LinearLayout>

```

Jalankan aplikasi, maka hasilnya tampak seperti berikut ini:



**Gambar 6.9 TimePicker format AM/PM**

Kembali ke Eclipse, modifikasi file MainActivity.java seperti berikut ini:

```

package net.belajar2android.ViewDasar4;

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

public class MainActivity extends Activity {
    TimePicker timePicker;

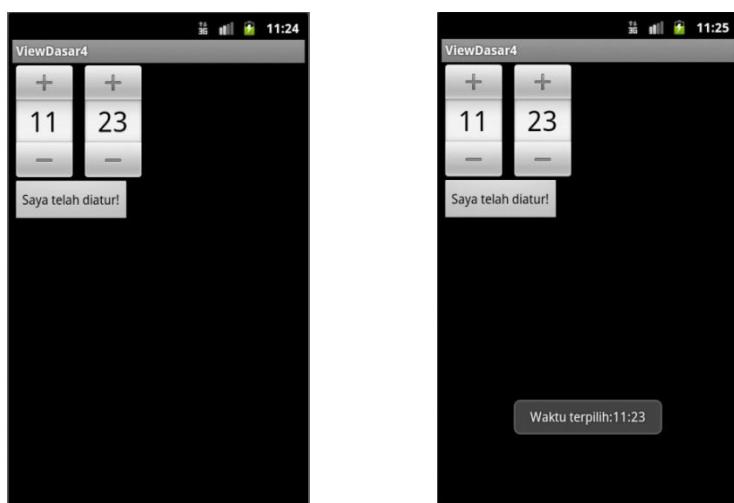
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        timePicker = (TimePicker) findViewById(R.id.timePicker);
        timePicker.setIs24HourView(true);

        //---view Button---
        Button btnOpen = (Button) findViewById(R.id.btnSet);
        btnOpen.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {

```

```
        Toast.makeText(getApplicationContext(), "Waktu terpilih:" +  
timePicker.getCurrentHour() + ":" +  
timePicker.getCurrentMinute(), Toast.LENGTH_SHORT).show();  
    }  
});  
}  
}
```

Jalankan aplikasi di Emulator Android. TimePicker akan menampilkan waktu dalam format 24 jam. Klik pada tombolnya maka akan ditampilkan waktu yang telah diatur dalam TimePicker.



**Gambar 6.10 TimePicker format 24 jam**

### 6.2.2 Menampilkan TimePicker Dalam Window Dialog

Kita juga bisa menampilkan TimePicker ke window dialog, sehingga sekali waktu diatur maka TimePicker akan menghilang dan tidak dapat mengambil ruang manapun di dalam activity.

Tetap menggunakan project sebelumnya (ViewDasar4), modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.ViewDasar4;
```

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TimePicker;
import android.widget.Toast;
import android.app.Dialog;
import android.app.TimePickerDialog;

public class MainActivity extends Activity {
    TimePicker timePicker;
    int hour, minute;
    static final int TIME_DIALOG_ID = 0;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        showDialog(TIME_DIALOG_ID);
        timePicker = (TimePicker) findViewById(R.id.timePicker);
        timePicker.setIs24HourView(true);

        //---view Button---
        Button btnOpen = (Button) findViewById(R.id.btnSet);
        btnOpen.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(), "Waktu terpilih: " +
                timePicker.getCurrentHour() + ":" +
                timePicker.getCurrentMinute(), Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Override
    protected Dialog onCreateDialog(int id) {
        switch (id) {
            case TIME_DIALOG_ID:
                return new TimePickerDialog(this, mTimeSetListener, hour,
                minute, false);
            }
            return null;
        }

    private TimePickerDialog.OnTimeSetListener mTimeSetListener =
    new TimePickerDialog.OnTimeSetListener(){
    public void onTimeSet(
        TimePicker view, int hourOfDay, int minuteOfHour){
        hour = hourOfDay;
        minute = minuteOfHour;
        Toast.makeText(getApplicationContext(), "Anda telah memilih : " +
        hour + ":" + minute, Toast.LENGTH_SHORT).show();
    }
};
```

```
}
```

Ketika activity di-load maka akan terlihat TimePicker dalam window dialog. Atur waktunya, kemudian klik tombol Set, maka akan muncul window Toast yang menampilkan waktu yang baru saja diatur.



**Gambar 6.11 TimePicker dengan Window Dialog**

### 6.2.3 View Datepicker

View yang serupa dengan TimePicker adalah DatePicker. Dengan DatePicker user dapat memilih tanggal tertentu di dalam activity.

Gunakan project sebelumnya project sebelumnya (ViewDasar4), modifikasi file main.xml seperti berikut ini:

```
<?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" >
<DatePicker android:id="@+id/datePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<TimePicker android:id="@+id/timePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<Button android:id="@+id/btnSet"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Saya telah diatur!"/>
</LinearLayout>
```

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.ViewDasar4;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import android.app.Dialog;
import android.app.TimePickerDialog;
import android.widget.TimePicker;
import android.widget.DatePicker;

public class MainActivity extends Activity {
    TimePicker timePicker;
    DatePicker datePicker;
    int hour, minute;
    static final int TIME_DIALOG_ID = 0;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        //showDialog(TIME_DIALOG_ID);
        timePicker = (TimePicker) findViewById(R.id.timePicker);
        timePicker.setIs24HourView(true);
        datePicker = (DatePicker) findViewById(R.id.datePicker);

        //---Button view---
        Button btnOpen = (Button) findViewById(R.id.btnSet);
        btnOpen.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(), "Date selected:" +
                        datePicker.getMonth() + 1 + "/" + datePicker.getDayOfMonth() +
                        "/" + datePicker.getYear() + "\n" + "Time selected:" +
                        timePicker.getCurrentHour() + ":" +
                        timePicker.getCurrentMinute(), Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Override
    protected Dialog onCreateDialog(int id)
    {
        switch (id) {
        case TIME_DIALOG_ID:
```

```

        return new TimePickerDialog(this, mTimeSetListener, hour,
minute, false);
    }
    return null;
}

private TimePickerDialog.OnTimeSetListener mTimeSetListener =
new TimePickerDialog.OnTimeSetListener()
{
    public void onTimeSet(
    TimePicker view, int hourOfDay, int minuteOfHour)
    {
        hour = hourOfDay;
        minute = minuteOfHour;
        Toast.makeText(getApplicationContext(), "You have selected : " +
hour + ":" + minute,
        Toast.LENGTH_SHORT).show();
    }
};
}

```

Tekan tombol F11 untuk menjalankan aplikasi ke Emulator Android, gambar berikut ini menunjukan view TimePicker dan DatePicker.



**Gambar 6.12 DatePicker**

#### **6.2.4 Menampilkan View DatePicker Dalam Window Dialog**

Berikut ini cara menampilkan DatePicker ke window dialog.

Tetap menggunakan project sebelumnya (ViewDasar4), modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.ViewDasar4;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TimePicker;
import android.widget.Toast;
import android.app.Dialog;
import android.app.TimePickerDialog;
import android.widget.DatePicker;

import android.app.DatePickerDialog;
import java.util.Calendar;

public class MainActivity extends Activity {
    TimePicker timePicker;
    DatePicker datePicker;

    int hour, minute;
    int yr, month, day;

    static final int TIME_DIALOG_ID = 0;
    static final int DATE_DIALOG_ID = 1;

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

        //showDialog(TIME_DIALOG_ID);

        //---mendapatkan tanggal sekarang---
        Calendar today = Calendar.getInstance();
        yr = today.get(Calendar.YEAR);
        month = today.get(Calendar.MONTH);
        day = today.get(Calendar.DAY_OF_MONTH);
        showDialog(DATE_DIALOG_ID);

        timePicker = (TimePicker) findViewById(R.id.timePicker);
        timePicker.setIs24HourView(true);
        datePicker = (DatePicker) findViewById(R.id.datePicker);

        //---view Button---
        Button btnOpen = (Button) findViewById(R.id.btnSet);
        btnOpen.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(), "Tanggal terpilih:" +
                        datePicker.getMonth() + 1 + "/" + datePicker.getDayOfMonth() +
                        "/" + datePicker.getYear() + "\n" + "Waktu terpilih:" +
```

```

        timePicker.getCurrentHour() + ":" +
        timePicker.getCurrentMinute(), Toast.LENGTH_SHORT).show();
    }
}
}

@Override
protected Dialog onCreateDialog(int id)
{
    switch (id) {
    case TIME_DIALOG_ID:
        return new TimePickerDialog(this, mTimeSetListener, hour,
minute, false);
    case DATE_DIALOG_ID:
        return new DatePickerDialog(this, mDateSetListener, yr,
month, day);
    }
    return null;
}

private DatePickerDialog.OnDateSetListener mDateSetListener =
new DatePickerDialog.OnDateSetListener()
{
    public void onDateSet(DatePicker view, int year, int
monthOfYear, int dayOfMonth)
    {
        yr = year;
        month = monthOfYear;
        day = dayOfMonth;
        Toast.makeText(getApplicationContext(), "Anda telah memilih : " +
(month + 1) + "/" + day + "/" + year,
Toast.LENGTH_SHORT).show();
    }
};

private TimePickerDialog.OnTimeSetListener mTimeSetListener =
new TimePickerDialog.OnTimeSetListener()
{
    public void onTimeSet(TimePicker view, int hourOfDay, int
minuteOfHour)
    {
        hour = hourOfDay;
        minute = minuteOfHour;
        Toast.makeText(getApplicationContext(), "Anda telah memilih : " +
hour + ":" + minute, Toast.LENGTH_SHORT).show();
    }
};
}

```

Jalankan aplikasi ke Emulator Android. Ketika activity di-load, DatePicker ditampilkan dalam window dialog. Atur tanggalnya, kemudian klik

tombol Set. Window Toast akan menampilkan tanggal yang baru saja diatur.



Gambar 6.13 DatePicker Window Dialog

## 6.3 View List

View List adalah untuk menampilkan daftar item yang panjang. Ada dua jenis view list: **ListView** dan **SpinnerView**.

### 6.3.1 View ListView

ListView menampilkan item dalam bentuk daftar vertikal yang bisa menggulung.

Buat project baru dan beri nama **ViewDasar5**.

<b>Project name</b>	ViewDasar5
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	ViewDasar5
<b>Package name</b>	net.belajar2android.ViewDasar5
<b>Create Activity</b>	MainActivity

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.ViewDasar5;

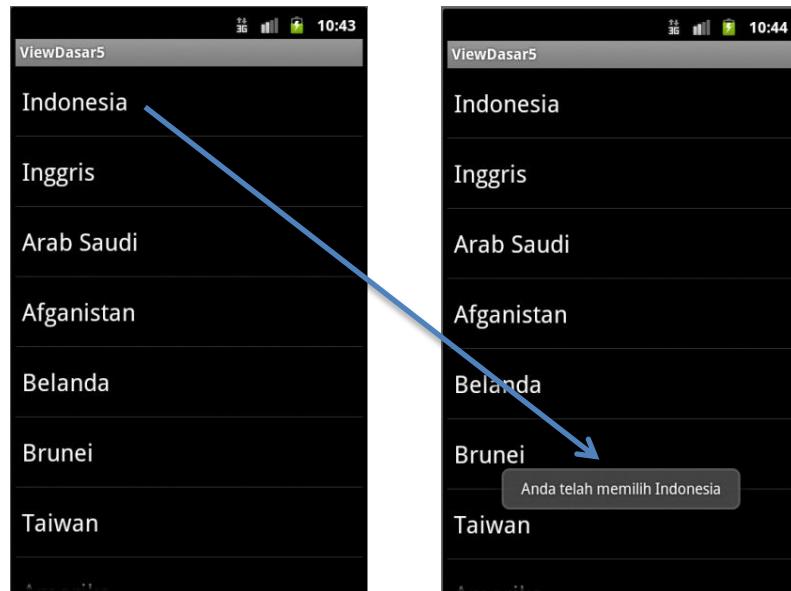
import android.app.Activity;
import android.os.Bundle;
import android.app.ListActivity;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

public class MainActivity extends ListActivity {
    String[] negara = {
        "Indonesia", "Inggris", "Arab Saudi", "Afganistan",
        "Belanda", "Brunei", "Taiwan", "Amerika", "Turki", "Singapore"
    };

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //setContentView(R.layout.main);
        setListAdapter(new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, negara));
    }

    public void onListItemClick(ListView parent, View v, int
    position, long id) {
        Toast.makeText(this, "Anda telah memilih " +
        negara[position], Toast.LENGTH_SHORT).show();
    }
}
```

Jalankan aplikasi ke Emulator Android. Activity akan menampilkan daftar item nama negara. Pilih salah satu Negara, maka akan muncul pesan nama Negara yang telah dipilih.



**Gambar 6.14 ListView**

Bagaimana program bekerja? Pada contoh di atas terdapat class `MainActivity` yang me-extends(mewarisi) class `ListActivity`. Class `ListActivity` me-extend class `Activity` dan menampilkan daftar item melalui data source. Disini tidak perlu memodifikasi file `main.xml` untuk memasukkan `ListView`; class `ListActivity` sendiri berisi sebuah `ListView`. Oleh karena itu, di dalam method `onCreate()` tidak memerlukan pemanggilan method `view setContentView()` untuk me-load UI dari file `main.xml`:

```
//setContentView(R.layout.main);
```

Dalam method `onCreate()` digunakan method `setListAdapter()` untuk mengisi keseluruhan layar activity dengan `ListView` secara program. Object  `ArrayAdapter` mengatur array string yang akan ditampilkan oleh

`ListView`. Pada contoh sebelumnya, `ListView` ditampilkan dalam mode `simple_list_item_1`:

```
setListAdapter(new ArrayAdapter<String>(this,  
        android.R.layout.simple_list_item_1, negara));
```

Method `onListItemClick()` dijalankan ketika item di dalam `ListView` diklik:

```
public void onListItemClick(ListView parent, View v, int  
    position, long id) {  
    Toast.makeText(this, "Anda telah memilih " + negara[position],  
    Toast.LENGTH_SHORT).show();  
}
```

### 6.3.2 Mengkustomisasi `ListView`

`ListView` dapat digunakan untuk memilih item lebih dari satu. Anda juga dapat mem-filter item-item yang ada.

Masih tetap menggunakan project sebelumnya (`ViewDasar5`), modifikasi file `MainActivity.java` seperti berikut ini:

```
package net.belajar2android.ViewDasar5;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.app.ListActivity;  
import android.view.View;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
import android.widget.Toast;  
  
public class MainActivity extends ListActivity {  
    String[] negara = {  
        "Indonesia", "Inggris", "Arab Saudi", "Afghanistan",  
        "Belanda", "Brunei", "Taiwan", "Amerika", "Turki", "Singapore"  
    };  
  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        //setContentView(R.layout.main);  
  
        ListView lstView = getListView();  
        //lstView.setChoiceMode(0);  
        //lstView.setChoiceMode(1);  
        lstView.setChoiceMode(2);  
        lstView.setTextFilterEnabled(true);  
    }  
}
```

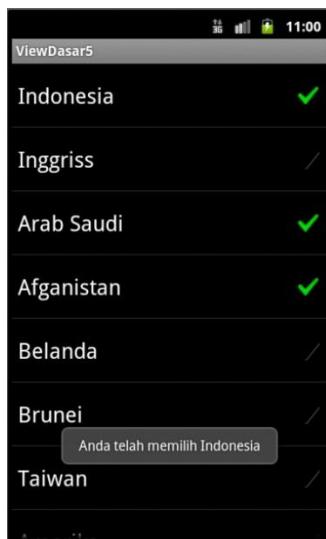
```

        setListAdapter(new ArrayAdapter<String>(this,
                android.R.layout.simple_list_item_checked, negara));
    }

    public void onListItemClick(ListView parent, View v, int
            position, long id) {
        parent.setItemChecked(position,
                parent.isItemChecked(position));
        Toast.makeText(this, "Anda telah memilih " +
                negara[position], Toast.LENGTH_SHORT).show();
    }
}

```

Jalankan aplikasi dan beri tanda centang pada beberapa item, maka hasilnya akan tampak seperti ini:



*Gambar 6.15 Kustomisasi ListView*

### 6.3.3 View Spinner

View Spinner pada pemrograman lain serupa dengan ComboBox, jadi Spinner bisa menampung item dalam jumlah banyak tapi hanya bisa menampilkan satu item.

Buat project baru bernama **ViewDasar6**.

<b>Project name</b>	ViewDasar6
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	ViewDasar6
<b>Package name</b>	net.belajar2android.ViewDasar6
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```
<?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">

    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:drawSelectorOnTop="true" />
</LinearLayout>
```

Modifikasi file string.xml yang terletak di dalam folder res/values:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, MainActivity!</string>
    <string name="app_name">ViewDasar6</string>
    <string-array name="negara_array">
        <item>Indonesia</item>
        <item>Inggris</item>
        <item>Arab Saudi</item>
        <item>Afganistan</item>
        <item>Belanda</item>
        <item>Brunei</item>
        <item>Taiwan</item>
        <item>Amerika</item>
        <item>Turki</item>
        <item>Singapore</item>
    </string-array>
</resources>
```

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.ViewDasar6;

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

```

import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends Activity {
    String[] negara;

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

        negara=getResources().getStringArray(R.array.negara_array);
        Spinner s1 = (Spinner) findViewById(R.id.spinner1);

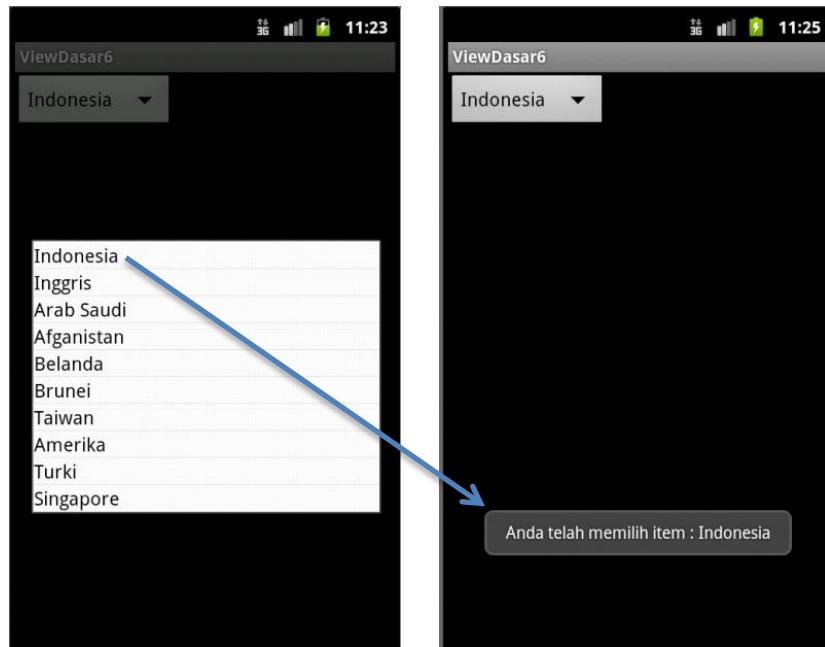
        ArrayAdapter<String> adapter = new
        ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,
        negara);

        s1.setAdapter(adapter);
        s1.setOnItemSelectedListener(new OnItemSelectedListener()
        {
            @Override
            public void onItemSelected(AdapterView<?> arg0, View arg1,
                int arg2, long arg3)
            {
                int index = arg0.getSelectedItemPosition();
                Toast.makeText(getApplicationContext(), "Anda telah memilih
                item : " + negara[index], Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onNothingSelected(AdapterView<?> arg0) {}
        });
    }
}

```

Jalankan aplikasi ke emulator. Klik pada SpinnerView, maka akan tampak tampilan pop-up yang menampilkan daftar nama Negara kemudian klik pada salah satu item maka akan muncul pesan item yang dipilih.

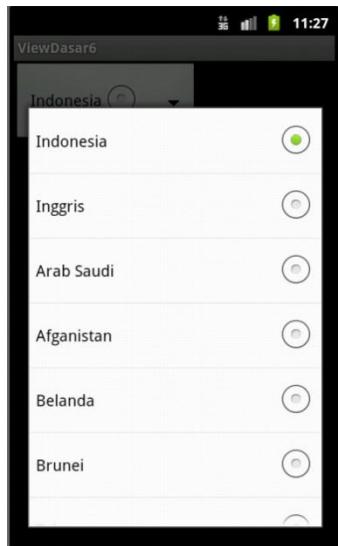


**Gambar 6.16 SpinnerView**

Bagaimana program bekerja? program bekerja seperti ListView, tetapi perlu tambahan method onNothingSelected(). Method ini bekerja ketika user menekan tombol Back, yang akan mengakhiri tampilan daftar item-item. Disamping menampilkan item-item dalam ArrayAdapter sebagai daftar/list, Anda juga dapat menampilkannya dengan menggunakan radio button. Untuk melakukannya, modifikasi parameter kedua constructor class ArrayAdapter:

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,  
        android.R.layout.simple_spinner_dropdown_item, negara);
```

Jika dijalankan aplikasinya maka list akan tampil disertai radio button seperti berikut ini:



**Gambar 6.17 Kustomisasi SpinnerView**

# BAB 7

## GAMBAR DAN MENU

Pembahasan bab ini masih tetap melibatkan view, bagaimana menampilkan gambar, menciptakan option dan membuat menu context dalam aplikasi Android.

### 7.1 View Image

Untuk menampilkan gambar, digunakan ImageView, Gallery, ImageSwitcher dan GridView.

#### 7.1.1 View Gallery Dan ImageView

Gallery adalah view yang menampilkan item-item foto dalam suatu bingkai yang bisa digulung secara horisontal.

Buat project baru bernama **Gallery**.

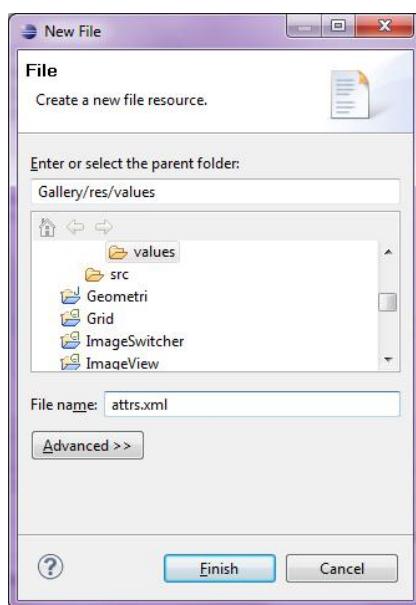
<b>Project name</b>	Gallery
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	Gallery
<b>Package name</b>	net.belajar2android.Gallery
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```
<?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="Koleksi Foto" />
    <Gallery android:id="@+id/gallery1" />
```

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
<ImageView android:id="@+id/image1"
    android:layout_width="320px"
    android:layout_height="250px"
    android:scaleType="fitXY" />
</LinearLayout>
```

Klik kanan pada folder res/values, pilih menu File|New|File, kemudian beri nama dengan **attrs.xml**



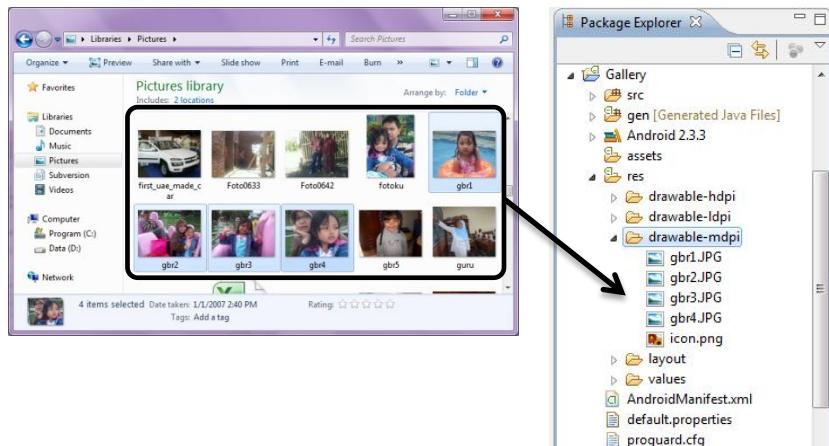
**Gambar 7.1 Menciptakan file XML**

Tuliskan pernyataan berikut ini ke dalam file attrs.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="Gallery1">
        <attr name="android:galleryItemBackground" />
    </declare-styleable>
</resources>
```

Siapkan gambar dengan nama yang sudah terurut (lakukan perubahan nama file gambar, misalnya gbr1.jpg, gbr2.jpg, gbr3.jpg dan seterusnya).

Kemudian copy-paste gambar dari windows explorer menuju folder res/drawable-mdpi di project Android.



**Gambar 7.2 Copy file gambar terurut ke drawable**

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.Gallery;

import android.app.Activity;
import android.os.Bundle;
import android.content.Context;
import android.content.res.TypedArray;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends Activity {
    //---gambar yang ditampilkan---
    Integer[] imageID = {
        R.drawable.gbr1,
        R.drawable.gbr2,
        R.drawable.gbr3,
        R.drawable.gbr4
    };

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

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    Gallery gallery = (Gallery) findViewById(R.id.gallery1);
    gallery.setAdapter(new ImageAdapter(this));
    gallery.setOnItemClickListener(new OnItemClickListener()
    {
        public void onItemClick(AdapterView<?> parent, View v, int
position, long id)
        {
            Toast.makeText(getApplicationContext(), "Foto" + (position + 1)
+ " dipilih", Toast.LENGTH_SHORT).show();
        }
    });
}

public class ImageAdapter extends BaseAdapter {
    private Context context;
    private int itemBackground;

    public ImageAdapter(Context c) {
        context = c;
        //---pengaturan style---
        TypedArray a =
obtainStyledAttributes(R.styleable.Gallery1);
        itemBackground =
a.getResourceId(R.styleable.Gallery1_android_galleryItemBackgroun
d, 0);
        a.recycle();
    }

    //--- mengembalikan jumlah gambar---
    public int getCount() {
        return imageID.length;
    }

    //---mengembalikan ID item---
    public Object getItem(int position) {
        return position;
    }

    //---mengembalikan ID item---
    public long getItemId(int position) {
        return position;
    }

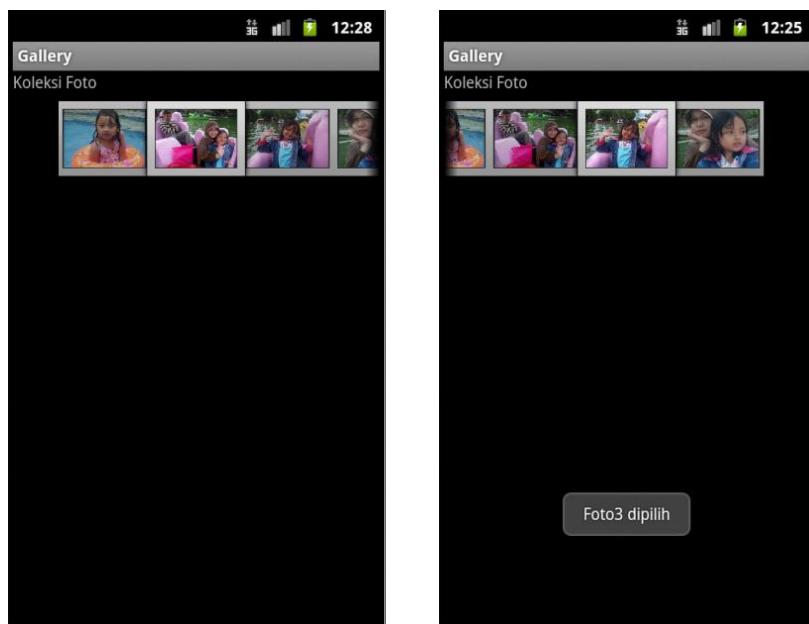
    //---mengembalikan view ImageView---
    public View getView(int position, View convertView,
ViewGroup parent) {
        ImageView imageView = new ImageView(context);
        imageView.setImageResource(imageID[position]);
        imageView.setScaleType(ImageView.ScaleType.FIT_XY);
        imageView.setLayoutParams(new Gallery.LayoutParams(150,
120));
    }
}
```

```

        imageView.setBackgroundResource(itemBackground);
        return imageView;
    }
}
}

```

Jalankan aplikasi ke Emulator. Gallery akan menampilkan serangkaian foto yang dapat digulung dengan cara menggeser (klik kiri kemudian foto digeser ke kiri atau ke kanan). Perhatikan pula saat salah satu foto diklik, maka akan muncul pesan foto dipilih.



**Gambar 7.3 View Gallery**

Untuk menampilkan gambar yang dipilih ke ImageView, tambahkan pernyataan yang berhuruf tebal di bawah ini ke file MainActivity.java:

```

Gallery gallery = (Gallery) findViewById(R.id.gallery1);
gallery.setAdapter(new ImageAdapter(this));
gallery.setOnItemClickListener(new OnItemClickListener()
{
    public void onItemClick(AdapterView<?> parent, View v, int
position, long id)

```

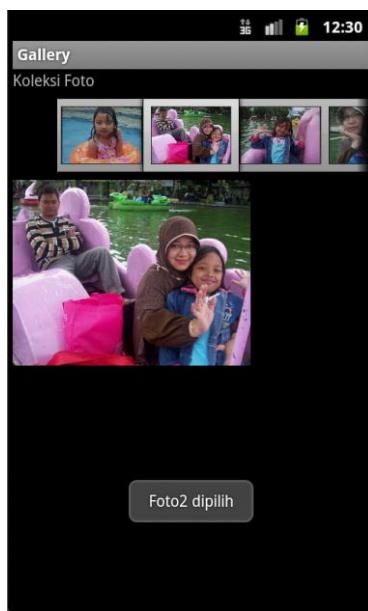
```

    {
        Toast.makeText(getApplicationContext(), "Foto" + (position + 1) + " dipilih", Toast.LENGTH_SHORT).show();

        //---menampilkan gambar yang dipilih---
        ImageView imageView = (ImageView) findViewById(R.id.image1);
        imageView.setImageResource(imageID[position]);
    }
});

```

Kali ini akan terlihat gambar yang dipilih dalam ImageView.



**Gambar 7.4 Menampilkan gambar ke ImageView**

Bagaimana program bekerja? Pertama tambahkan view Gallery dan ImageView ke main.xml:

```

<Gallery android:id="@+id/gallery1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
<ImageView android:id="@+id/image1"
    android:layout_width="320px"
    android:layout_height="250px"
    android:scaleType="fitXY" />

```

View Gallery digunakan untuk menampilkan gambar-gambar dalam bentuk daftar gulungan secara horizontal. ImageView digunakan untuk menampilkan gambar yang telah dipilih oleh user. Daftar gambar yang akan ditampilkan disimpan dalam array imageID:

```
----gambar yang ditampilkan---
Integer[] imageID = {
    R.drawable.gbr1,
    R.drawable.gbr2,
    R.drawable.gbr3,
    R.drawable.gbr4
};
```

Kemudian diciptakan class ImageAdapter yang meng-extend/mewarisi class BaseAdapter sehingga class ImageAdapter bisa mengikat view Gallery dengan serangkaian view ImageView:

```
Gallery gallery = (Gallery) findViewById(R.id.gallery1);
gallery.setAdapter(new ImageAdapter(this));
gallery.setOnItemClickListener(new OnItemClickListener()
{
    public void onItemClick(AdapterView<?> parent, View v, int
position, long id)
    {
        Toast.makeText(getApplicationContext(), "Foto" + (position + 1) + "dipilih", Toast.LENGTH_SHORT).show();

        ----menampilkan gambar yang dipilih---
        ImageView imageView = (ImageView) findViewById(R.id.image1);
        imageView.setImageResource(imageID[position]);
    }
});
```

### 7.1.2 View ImageSwitcher

Terkadang kita tidak ingin sebuah gambar tampil secara tiba-tiba ketika user memilihnya dari view Gallery. Nah pada bagian ini akan diaplikasikan animasi transisi dari satu gambar ke gambar yang lain. Untuk itu digunakan ImageSwitcher bersama-sama dengan view Gallery.

Buat project baru bernama **ImageSwitcher**.

Project name	ImageSwitcher
--------------	---------------

<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	ImageSwitcher
<b>Package name</b>	net.belajar2android.ImageSwitcher
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```
<?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"
        android:background="#ff0000" >
<Gallery
    android:id="@+id/gallery1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
<ImageSwitcher
    android:id="@+id/switcher1"
    android:layout_width="fill_parent"
    android:layout_height="fill_height"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true" />
</RelativeLayout>
```

Klik kanan pada folder **res/values** dan pilih menu **New | File**. Beri nama file dengan **attrs.xml**. Kemudian tuliskan pernyataan berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="Gallery1">
        <attr name="android:galleryItemBackground" />
    </declare-styleable>
</resources>
```

copy-paste atau drag-drop beberapa gambar (gambar yang memiliki nama file berurutan, contoh pic1, pic2, pic3 dan seterusnya) dari windows explorer ke folder Android project res/drawable-mdpi.

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.ImageSwitcher;

import android.app.Activity;
import android.os.Bundle;
```

```
import android.content.Context;
import android.content.res.TypedArray;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewGroup.LayoutParams;
import android.view.animation.AnimationUtils;
import android.widget.BaseAdapter;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Gallery;
import android.widget.ViewSwitcher.ViewFactory;
import android.widget.ImageSwitcher;
import android.widget.ImageView;

public class MainActivity extends Activity implements
ViewFactory {
    //---gambar yang ditampilkan---
    Integer[] imageID = {
        R.drawable.pic1, R.drawable.pic2, R.drawable.pic3,
        R.drawable.pic4, R.drawable.pic5, R.drawable.pic6,
        R.drawable.pic7, R.drawable.pic8, R.drawable.pic9
    };

    private ImageSwitcher imageSwitcher;

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

        imageSwitcher = (ImageSwitcher)
        findViewById(R.id.switcher1);
        imageSwitcher.setFactory(this);

        imageSwitcher.setInAnimation(AnimationUtils.loadAnimation(this,
        android.R.anim.fade_in));

        imageSwitcher.setOutAnimation(AnimationUtils.loadAnimation(this,
        android.R.anim.fade_out));
        Gallery gallery = (Gallery) findViewById(R.id.gallery1);
        gallery.setAdapter(new ImageAdapter(this));
        gallery.setOnItemClickListener(new OnItemClickListener()
        {
            public void onItemClick(AdapterView<?> parent, View v, int
            position, long id)
            {
                imageSwitcher.setImageResource(imageID[position]);
            }
        });
    }

    public View makeView()
    {
        ImageView imageView = new ImageView(this);
```

```
        imageView.setBackgroundColor(0xFF000000);
        imageView.setScaleType(ImageView.ScaleType.FIT_CENTER);
        imageView.setLayoutParams(new
ImageSwitcher.LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.FILL_PARENT));
        return imageView;
    }

    public class ImageAdapter extends BaseAdapter
    {
        private Context context;
        private int itemBackground;
        public ImageAdapter(Context c)
        {
            context = c;
            //---pengaturan style---
            TypedArray a =
obtainStyledAttributes(R.styleable.Gallery1);
            itemBackground =
a.getResourceId(R.styleable.Gallery1_android_galleryItemBackgrou
nd, 0);
            a.recycle();
        }

        //---mengembalikan jumlah gambar---
        public int getCount()
        {
            return imageID.length;
        }

        //---mengembalikan item ID---
        public Object getItem(int position)
        {
            return position;
        }

        public long getItemId(int position)
        {
            return position;
        }

        //---mengembalikan view ImageView---
        public View getView(int position, View convertView,
ViewGroup parent)
        {
            ImageView imageView = new ImageView(context);
            imageView.setImageResource(imageID[position]);
            imageView.setScaleType(ImageView.ScaleType.FIT_XY);
            imageView.setLayoutParams(new Gallery.LayoutParams(150,
120));
            imageView.setBackgroundResource(itemBackground);
            return imageView;
        }
    }
}
```

### 7.1.3 View GridView

GridView menampilkan item-item dalam dua dimensi petak-petak (grid) yang dapat digulung. GridView digunakan bersama-sama dengan ImageView untuk menampilkan serangkaian gambar.

Buat project baru dengan Eclipse bernama **Grid**.

<b>Project name</b>	Grid
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	Grid
<b>Package name</b>	net.belajar2android.Grid
<b>Create Activity</b>	MainActivity

Copy-paste beberapa gambar dengan nama berurutan dari windows explorer ke folder res/drawable-mdpi.

Modifikasi file main.xml seperti berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<GridView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gridview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:numColumns="auto_fit"
    android:verticalSpacing="10dp"
    android:horizontalSpacing="10dp"
    android:columnWidth="90dp"
    android:stretchMode="columnWidth"
    android:gravity="center" />
```

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.Grid;

import android.app.Activity;
import android.os.Bundle;

import android.content.Context;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
```

```
import android.widget.AdapterView.OnItemClickListener;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends Activity {
    Integer[] imageIDs = {
        R.drawable.pic1, R.drawable.pic2, R.drawable.pic3,
        R.drawable.pic4, R.drawable.pic5, R.drawable.pic6,
        R.drawable.pic7 };

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

        GridView gridView = (GridView) findViewById(R.id.gridview);
        gridView.setAdapter(new ImageAdapter(this));
        gridView.setOnItemClickListener(new OnItemClickListener() {
            public void onItemClick(AdapterView<?> parent, View v, int position, long id){
                Toast.makeText(getApplicationContext(), "Foto" + (position + 1)
+ " dipilih", Toast.LENGTH_SHORT).show();
            }
        });
    }

    public class ImageAdapter extends BaseAdapter {
        private Context context;
        public ImageAdapter(Context c) {
            context = c;
        }

        public int getCount() {
            return imageIDs.length;
        }

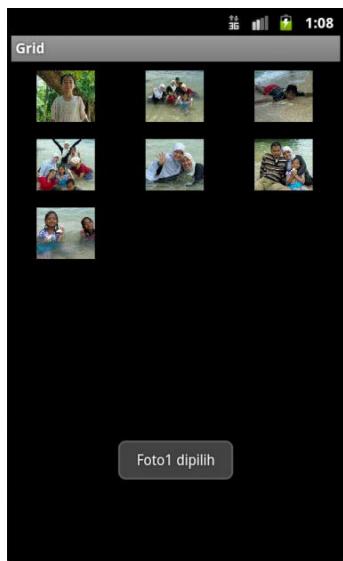
        public Object getItem(int position) {
            return position;
        }

        public long getItemId(int position) {
            return position;
        }

        public View getView(int position, View convertView,
ViewGroup parent) {
            ImageView imageView;
            if (convertView == null) {
                imageView = new ImageView(context);
                imageView.setLayoutParams(new GridView.LayoutParams(85,
85));
                imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
            }
            imageView.setImageResource(imageIDs[position]);
            return imageView;
        }
    }
}
```

```
        imageView.setPadding(5, 5, 5, 5);
    } else
    {
        imageView = (ImageView) convertView;
    }
    imageView.setImageResource(imageIDs[position]);
    return imageView;
}
}
```

Jalankan aplikasi ke Emulator Android. GridView akan menampilkan seluruh gambar, seperti berikut ini:



**Gambar 7.5 Menampilkan gambar dengan Grid**

Bagaimana program bekerja? Seperti pada contoh Gallery dan ImageSwitcher, diimplementasikan class ImageAdapter kemudian mengikatnya ke GridView

```
GridView gridView = (GridView) findViewById(R.id.gridview);
gridView.setAdapter(new ImageAdapter(this));
gridView.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View v, int
position, long id) {
        Toast.makeText(getApplicationContext(), "Foto" + (position + 1) + "
dipilih", Toast.LENGTH_SHORT).show();
```

```
    }
});
```

Ketika gambar dipilih maka akan tampil pesan Toast yang menandakan bahwa gambar dipilih. Pada GridView dapat ditentukan pula ukuran gambar dan bagaimana gambar-gambar diletakkan pada GridView melalui pengaturan padding setiap gambar:

```
public View getView(int position, View convertView, ViewGroup
parent) {
    ImageView imageView;
    if (convertView == null) {
        imageView = new ImageView(context);
        imageView.setLayoutParams(new GridView.LayoutParams(85,
85));
        imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
        imageView.setPadding(5, 5, 5, 5);
    } else
    {
        imageView = (ImageView) convertView;
    }
    imageView.setImageResource(imageIDs[position]);
    return imageView;
}
```

## 7.2 Membuat Menu

Menu berguna untuk menampilkan pilihan tambahan, namun tidak tampak secara langsung pada UI utama aplikasi.

Ada dua jenis menu di dalam Android:

- **Menu Option** – menampilkan informasi yang terhubung ke activity saat ini. Pengaktifan menu option melalui penekanan key **MENU**.
- **Menu Context** - menampilkan informasi yang terhubung ke view tertentu pada activity. Pengaktifan menu context dengan menekan dan menahan link/menu bersangkutan.

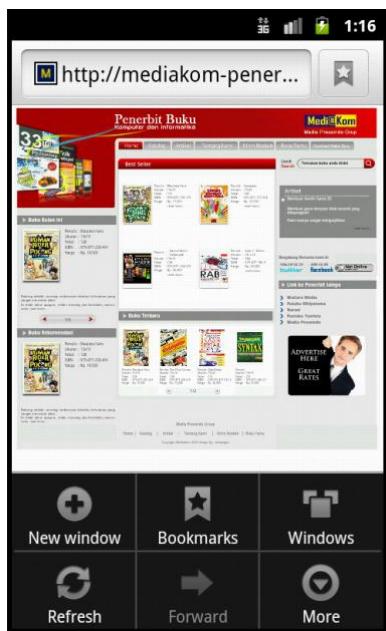
Gambar 7.8.1 menunjukkan contoh menu option pada aplikasi browser.

Menu option ditampilkan kapan pun ketika user menekan tombol MENU

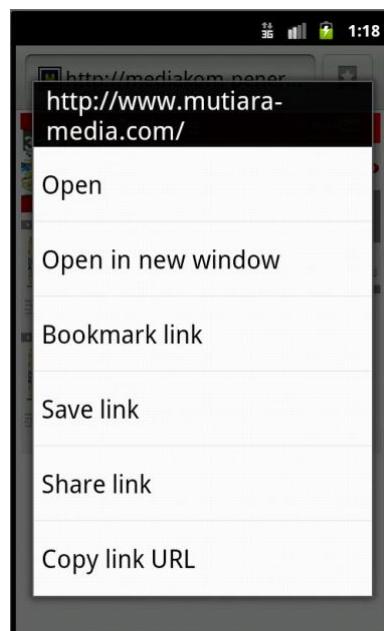


. Item menu ditampilkan berubah-ubah menyesuaikan activity yang

sedang aktif saat ini. Gambar 7.8.2 menunjukkan contoh context menu yang tampil ketika user menekan dan menahan pada salah satu hyperlink halaman website. Item menu tampil berubah-ubah menyesuaikan komponen atau view yang dipilih pada saat ini.



Gambar 7.8.1 Menu Option



Gambar 7.8.2 Menu Context

### 7.2.1 Menciptakan Method Pembantu

Sebelum menciptakan menu option dan menu context perlu dibuat dua method pembantu. Method pertama adalah membuat daftar item untuk ditampilkan pada menu, method lainnya menangani event yang dijalankan ketika user memilih item pada menu.

Buatlah project baru bernama **Menus**.

Project name	Menus
--------------	-------

<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	Menus
<b>Package name</b>	net.belajar2android.Menus
<b>Create Activity</b>	MainActivity

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.Menus;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.Toast;

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

    private void CreateMenu(Menu menu) {
        MenuItem mnu1 = menu.add(0, 0, 0, "Item 1");
        {
            mnu1.setAlphabeticShortcut('a');
            mnu1.setIcon(R.drawable.icon);
        }
        MenuItem mnu2 = menu.add(0, 1, 1, "Item 2");
        {
            mnu2.setAlphabeticShortcut('b');
            mnu2.setIcon(R.drawable.icon);
        }
        MenuItem mnu3 = menu.add(0, 2, 2, "Item 3");
        {
            mnu3.setAlphabeticShortcut('c');
            mnu3.setIcon(R.drawable.icon);
        }
        MenuItem mnu4 = menu.add(0, 3, 3, "Item 4");
        {
            mnu4.setAlphabeticShortcut('d');
        }
        menu.add(0, 3, 3, "Item 5");
        menu.add(0, 3, 3, "Item 6");
        menu.add(0, 3, 3, "Item 7");
    }

    private boolean MenuChoice(MenuItem item)
    {
```

```

        switch (item.getItemId()) {
        case 0:
            Toast.makeText(this, "Anda telah mengklik Item 1",
Toast.LENGTH_LONG).show();
            return true;
        case 1:
            Toast.makeText(this, "Anda telah mengklik Item 2",
Toast.LENGTH_LONG).show();
            return true;
        case 2:
            Toast.makeText(this, "Anda telah mengklik Item 3",
Toast.LENGTH_LONG).show();
            return true;
        case 3:
            Toast.makeText(this, "Anda telah mengklik Item 4",
Toast.LENGTH_LONG).show();
            return true;
        case 4:
            Toast.makeText(this, "Anda telah mengklik Item 5",
Toast.LENGTH_LONG).show();
            return true;
        case 5:
            Toast.makeText(this, "Anda telah mengklik Item 6",
Toast.LENGTH_LONG).show();
            return true;
        case 6:
            Toast.makeText(this, "Anda telah mengklik Item 7",
Toast.LENGTH_LONG).show();
            return true;
        }
        return false;
    }
}

```

Bagaimana program bekerja? Contoh di atas adalah menciptakan dua method: **CreateMenu()** dan **MenuChoice()**. Method **CreateMenu()** mengambil argument menu dan menambahkan serangkaian item menu kepadanya.

Untuk menambahkan sebuah item menu ke Menu, perlu diciptakan instance class MenuItem dan menggunakan method add() dari object menu.

```

MenuItem mnu1 = menu.add(0, 0, 0, "Item 1");
{
    mnu1.setAlphabeticShortcut('a');
    mnu1.setIcon(R.drawable.icon);
}

```

Empat argument dari method add() secara berurutan dari kiri ke kanan adalah sebagai berikut:

- **groupId** – Penanda kelompok, dimana item menu tersebut merupakan bagian darinya. Gunakan 0 jika item bukan bagian dari kelompok.
- **itemId** – ID item yang bersifat unik.
- **order** - Urutan item menu yang akan ditampilkan.
- **title** – teks yang ditampilkan ke menu.

Method setAlphabeticShortcut() digunakan untuk menugaskan key shortcut ke item menu sehingga user dapat memilih item melalui tombol keyboard.

Method setIcon() mengatur gambar yang ditampilkan pada item menu.

Method MenuChoice() mengambil argument MenuItem dan mengecek setiap ID untuk menentukan item menu yang diklik.

Method MenuChoice() menampilkan pesan Toast agar user mengetahui item menu mana yang telah diklik.

### 7.2.2 Menu Option

Sekarang kita siap untuk memodifikasi aplikasi agar dapat menampilkan menu option ketika user menekan tombol **MENU**.

Gunakan project sebelumnya (Menus) dan modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.Menus;

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

```
import android.widget.Toast;

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

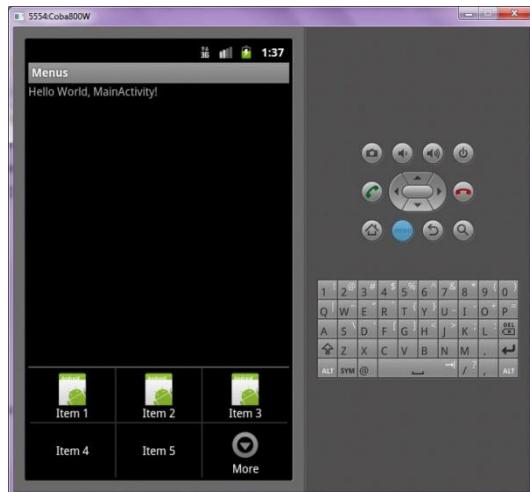
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);
        CreateMenu(menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        return MenuChoice(item);
    }

    private void CreateMenu(Menu menu) {
        //...
    }

    private boolean MenuChoice(MenuItem item)
    {
        //...
    }
}
```

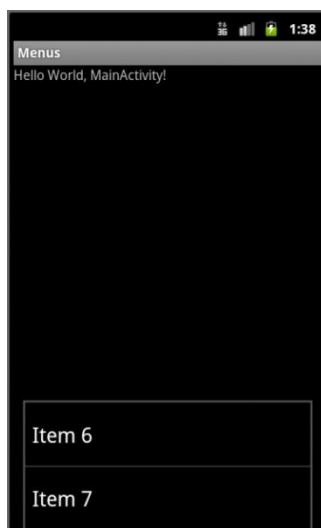
Sekarang jalankan aplikasi ke Emulator Android. Gambar di bawah ini menunjukan menu option ketika tombol MENU ditekan. Untuk memilih item menu, klik pada salah satu item atau gunakan shortcut key (A sampai D; hanya diaplikasikan pada empat item pertama).



**Gambar 7.9 Menu Option**

Bagaimana program bekerja? Untuk menampilkan menu option pada activity perlu meng-override dua method dalam activity: **onCreateOptionsMenu()** dan **onOptionsItemSelected()**. Method **onCreateOptionsMenu()** dipanggil ketika tombol MENU ditekan. Dalam event ini, method pembantu **CreateMenu()** dipanggil untuk menampilkan menu option. Ketika item menu dipilih, method **onOptionsItemSelected()** dipanggil. Dalam kasus ini , Anda memanggil method **MenuChoice()** untuk menampilkan item menu yang dipilih.

Perhatikan icon yang tampak pada item menu 1, 2 dan 3. Perhatikan juga, jika menu option memiliki lebih dari enam item, maka menu "**More**" akan ditampilkan yang menandakan bahwa ada pilihan tambahan. Gambar di bawah ini menunjukkan tambahan item menu yang tampil berupa daftar item menu setelah user menekan item menu "**More**".



*Gambar 7.10 Item Menu Tambahan*

### 7.2.3 Menu Context

Menu context biasanya berhubungan dengan view pada sebuah activity dan menu context ditampilkan ketika user meng-klik item dalam waktu tertentu. Contoh, bila user menekan view Button dan menahannya untuk beberapa detik maka menu context akan ditampilkan.

Jika hendak menghubungkan menu context dengan view pada activity, maka perlu dipanggil method setOnCreateOptionsMenu() dari view bersangkutan.

Masih tetap memanfaatkan project sebelumnya (Menus), modifikasi file main.xml seperti berikut ini:

```
<?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:id="@+id/btn1"
        android:layout_width="fill_parent"
```

```
        android:layout_height="wrap_content"
        android:text="Klik dan tahan tombol ini untuk beberapa saat"
    />
</LinearLayout>
```

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.Menus;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.Toast;
import android.view.View;
import android.view.ContextMenu;
import android.view.ContextMenu.ContextMenuItemInfo;

public class MenusActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button btn = (Button) findViewById(R.id.btn1);
        btn.setOnCreateContextMenuListener(this);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);
        CreateMenu(menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        return MenuChoice(item);
    }

    @Override
    public void onCreateContextMenu(ContextMenu menu, View view,
        ContextMenuItemInfo menuInfo) {
        super.onCreateContextMenu(menu, view, menuInfo);
        CreateMenu(menu);
    }

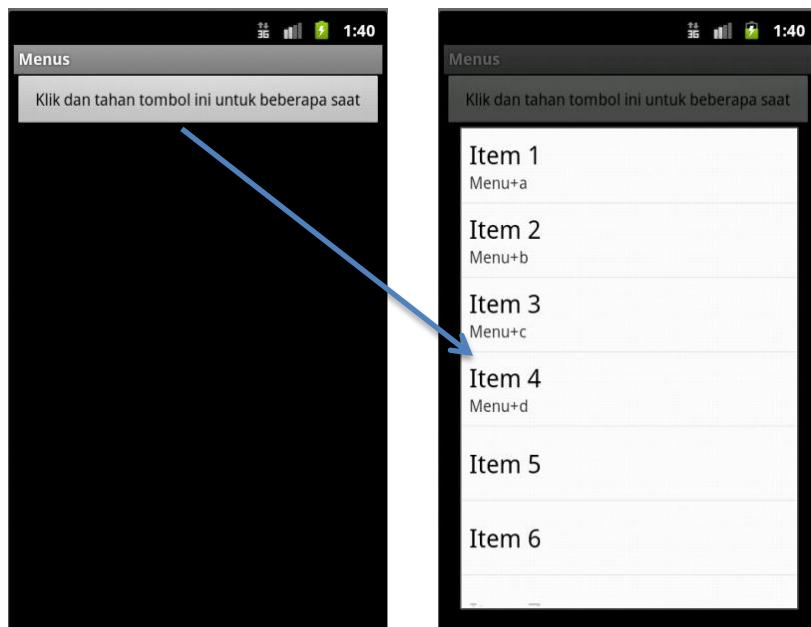
    @Override
    public boolean onContextItemSelected(MenuItem item) {
        return MenuChoice(item);
    }

    private void CreateMenu(Menu menu) {
```

```
//...
}

private boolean MenuChoice(MenuItem item)
{
    //...
}
```

Jalankan aplikasi ke Emulator Android, tekan tombolnya beberapa saat maka menu context akan tampil.



**Gambar 7.11 Menu Context**

Bagaimana program bekerja? Anda memanggil method setOnCreateContextMenuListener() dari view Button untuk menghubungkannya dengan menu context.

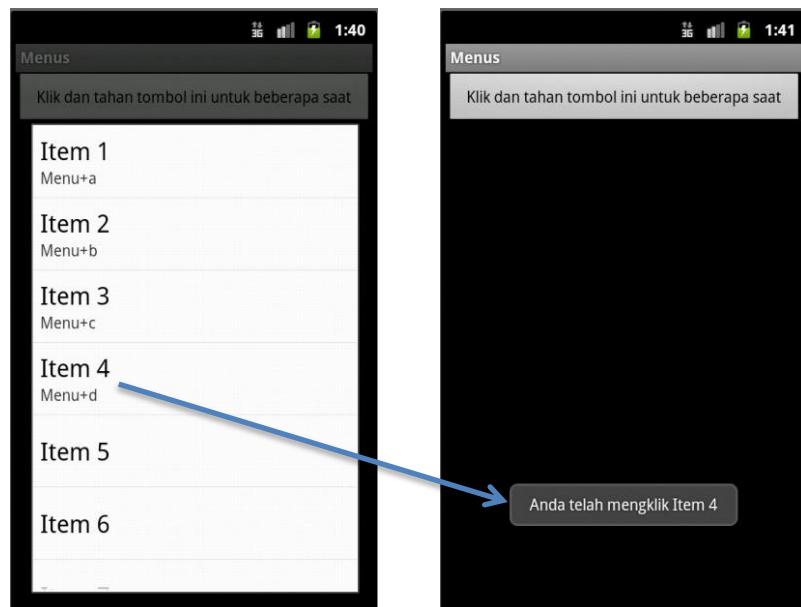
Ketika user mengklik view Button dalam waktu lama, method onCreateContextMenu() akan dipanggil. Dalam method tersebut,

method `CreateMenu()` dipanggil untuk menampilkan menu context. Serupa dengan hal tersebut, ketika sebuah item dalam menu context dipilih, method `onContextItemSelected()` akan dipanggil, dimana method `MenuChoice()` dipanggil untuk menampilkan pesan ke user.

**Catatan:**

Shortcut key untuk item menu tidak bisa bekerja. Agar shortcut key dapat bekerja maka method `setQwertyMode()` dipanggil dari object `Menu`, seperti berikut ini:

```
private void CreateMenu(Menu menu) {  
    menu.setQwertyMode(true);  
    MenuItem mn1 = menu.add(0, 0, 0, "Item 1");  
    //...  
}
```



**Gambar 7.12 Pemanggilan Menu Context**

## 7.3 Beberapa View Tambahan

Disamping view-view standar yang sudah dibahas, SDK Android juga menyediakan beberapa view-view tambahan yang membuat aplikasi jadi lebih menarik. Pada bagian ini akan dipelajari tentang view AnalogClock, DigitalClock dan WebView.

### 7.3.1 View Analogclock Dan DigitalClock

View AnalogClock menampilkan jam analog menjadi dua sisi, yaitu menit dan jam. Sedang view DigitalClock menampilkan jam secara digital. Keduanya hanya sekedar menampilkan waktu sistem perangkat Android.

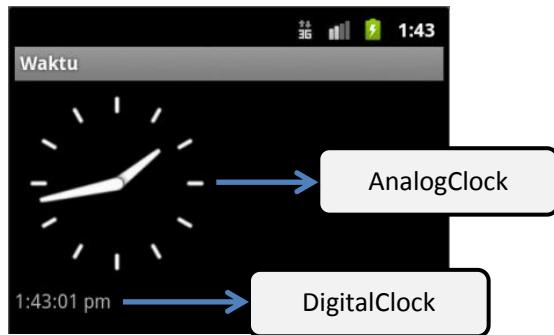
Buat project baru bernama **Waktu** dengan Eclipse.

<b>Project name</b>	Waktu
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	Waktu
<b>Package name</b>	net.belajar2android.Waktu
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```
<?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" >
<AnalogClock
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<DigitalClock
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
```

Jalankan aplikasi ke Emulator Android, hasilnya akan tampak seperti ini



Gambar 7.13 View AnalogClock

### 7.3.2 View WebView

WebView mengizinkan Anda untuk menempelkan browser ke dalam activity. Ini sangat berguna jika aplikasi perlu menyertakan beberapa konten web, seperti peta dari beberapa penyedia layanan dan lain-lain.

Buat project baru dan beri nama **WebView**.

<b>Project name</b>	WebView
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	WebView
<b>Package name</b>	net.belajar2android.WebView
<b>Create Activity</b>	MainActivity

Karena aplikasi membutuhkan akses internet maka tambahkan pernyataan **uses-permission** ke dalam file AndroidManifest.xml seperti berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.belajar2android.WebView"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />
    <uses-permission android:name="android.permission.INTERNET" />
```

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

Modifikasi file main.xml seperti berikut ini:

```
<?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" >
<WebView android:id="@+id/webview1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
```

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.WebView;

import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;

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

        WebView wv = (WebView) findViewById(R.id.webview1);
        WebSettings webSettings = wv.getSettings();
        webSettings.setBuiltInZoomControls(true);
        wv.loadUrl("http://mediakita.com/components/com_virtuemart/shop_
image/product/Membuat_Aplikasi_4c4954362a6ab.jpg");
    }
}
```

Jalankan aplikasi ke Emulator Android. Gambar di bawah ini menampilkan isi webview, yaitu berupa gambar.



**Gambar 7.14 WebView**

Bagaimana program bekerja? agar WebView dapat me-load halaman web maka digunakan method loadUrl():

```
wv.loadUrl ("http://mediakita.com/components/com_virtuemart/shop_image/product/Membuat_Aplikasi_4c4954362a6ab.jpg");
```

Untuk menampilkan kontrol zoom dibutuhkan property WebSettings dari WebView, kemudian memanggil method setBuiltInZoomControls() :

```
WebSettings webSettings = wv.getSettings();  
webSettings.setBuiltInZoomControls(true);
```

Terkadang ketika me-load halaman web yang di-rediect (link dialihkan) seperti www.amazon.com, penggunaan WebView akan menyebabkan aplikasi dijalankan dari aplikasi browser perangkat. Misal, Anda menugaskan WebView me-load www.amazon.com maka secara otomatis link akan menuju ke http://www.amazon.com/aw/h.html/191-6984621-9339869. Dalam kasus ini, aplikasi secara otomatis menjalankan aplikasi browser (browser built-in android) untuk me-load halaman tersebut.



**Gambar 7.15 WebView yang dialihkan ke browser Android**

Untuk mencegah hal ini, perlu diimplementasikan class `WebViewClient` dan meng-override method `shouldOverrideUrlLoading()`:

```

package net.belajar2android.WebView;

import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;

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

        WebView wv = (WebView) findViewById(R.id.webview1);
        wv.setWebViewClient(new Callback());
        WebSettings webSettings = wv.getSettings();
        webSettings.setBuiltInZoomControls(true);
        wv.loadUrl("http://amazon.com/");
    }

    private class Callback extends WebViewClient {
        @Override
        public boolean shouldOverrideUrlLoading(WebView view, String
url) {
            return (false);
    }
}

```

```
        }  
    }  
}
```

Jalankan aplikasi ke Emulator Android. Sekarang halaman web akan tetap di-load dari WebView dan tidak lagi dialihkan ke aplikasi browser perangkat.



**Gambar 6.17 amazon.com tetap di-load melalui WebView**

Anda dapat pula menyisipkan string HTML secara dinamis dan me-load HTML ke WebView, untuk itu gunakan method loadData():

```
package net.belajar2android.WebView;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.webkit.WebView;  
  
public class MainActivity extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```

```
WebView wv = (WebView) findViewById(R.id.webview1);
final String mimeType = "text/html";
final String encoding = "UTF-8";
String html=<H1>Halaman HTML
sederhana</H1><body><p>Ternyata Android gak susah-susah
amat!</p>";
wv.loadData(html, mimeType, encoding);
}
}
```

Gambar di bawah menunjukkan script HTML yang dibuat secara dinamis melalui program dan dijalankan oleh WebView.



**Gambar 7.17 Halaman web dinamis**

# **BAB 8**

# **PENGELOLAAN DATA**

Pengelolaan data adalah topik penting dalam pengembangan aplikasi. Pada intinya yang akan dibahas pada bab ini antara lain:

- Bagaimana menyimpan data sederhana dengan menggunakan object SharedPreferences.
- Bagaimana menulis dan membaca file dari penyimpanan internal dan penyimpanan eksternal.
- Bagaimana menciptakan dan menggunakan database SQLite.

## **8.1 Menyimpan dan Me-Load User Preferences**

Android menyediakan object SharedPreferences untuk menyimpan data sederhana suatu aplikasi. Contoh, Aplikasi yang membolehkan user mengubah ukuran font teks yang tampil pada aplikasi. Dalam kasus ini, aplikasi perlu mengingat berapa nilai ukuran font yang telah ditentukan oleh user, sehingga untuk waktu selanjutnya user dapat menggunakan aplikasi dengan ukuran font seperti yang diharapkan karena aplikasi Anda dapat menge-set ukuran yang tepat. Untuk melakukan tugas tersebut ada beberapa pilihan. Pertama, menyimpan data ke sebuah file, tapi Anda harus membuat routine (potongan program kecil) pengaturan file, seperti menulis data ke file, mengindikasikan berapa banyak karakter yang dibaca atau beberapa bagian informasi lainnya, seperti ukuran teks, nama font, warna background dan lain-lain, kemudian menuliskannya ke sebuah file akan menjadi lebih rumit.

Alternatif kedua selain menulis data ke file teks adalah dengan menggunakan database, tetapi penyimpanan data sederhana ke database adalah suatu yang berlebihan, kedua cara di atas bisa dipilih oleh pengembang aplikasi, tergantung sudut pandang mana yang paling sesuai dengan kebutuhan dan tentunya perlu mempertimbangkan kinerja aplikasi.

Sedangkan Object SharedPreferences adalah menyimpan data berupa pasangan key/value. Key dan value disimpan secara otomatis ke sebuah file XML untuk Anda.

### 8.1.1 Penggunaan getSharedPreferences()

Untuk melihat bagaimana object SharedPreferences bekerja, ikuti tahapan berikut ini yang mendemonstrasikan bagaimana menyimpan data user ke file XML, kemudian mencari data tersebut melalui object yang sama.

Buat project baru bernama **ObjectPreferences**.

<b>Project name</b>	ObjectPreferences
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	ObjectPreferences
<b>Package name</b>	net.belajar2android.ObjectPreferences
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```
<?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" >
<SeekBar
    android:id="@+id/SeekBar01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
<TextView
```

```

    android:id="@+id/TextView01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello" />
<EditText
    android:id="@+id/EditText01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/btnSave"
    android:text="Save"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>

```

Modifikasi file MainActivity.java seperti berikut ini:

```

package net.belajar2android.ObjectPreferences;

import android.app.Activity;
import android.os.Bundle;
import android.content.SharedPreferences;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.SeekBar;
import android.widget.SeekBar.OnSeekBarChangeListener;
import android.widget.Toast;

public class MainActivity extends Activity {
    private SharedPreferences prefs;
    private String prefName = "MyPref";
    private EditText editText;
    private SeekBar seekBar;
    private Button btn;

    private static final String FONT_SIZE_KEY = "fontsize";
    private static final String TEXT_VALUE_KEY = "textvalue";

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

        editText = (EditText) findViewById(R.id.EditText01);
        seekBar = (SeekBar) findViewById(R.id.SeekBar01);
        btn = (Button) findViewById(R.id.btnSave);

        btn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                //---mendapatkan object SharedPreferences---
                prefs = getSharedPreferences(prefName, MODE_PRIVATE);
                SharedPreferences.Editor editor = prefs.edit();

```

```

    //---menyimpan nilai dalam view EditText ke
preferences---
    editor.putFloat(FONT_SIZE_KEY, editText.getTextSize());
    editor.putString(TEXT_VALUE_KEY,
editText.getText().toString());

    //---menyimpan nilai---
    editor.commit();

    //---menampilkan file yang menyimpan pesan---
    Toast.makeText(getApplicationContext(), "Ukuran font sukses
disimpan!", Toast.LENGTH_SHORT).show();
}
};

//--load object SharedPreferences---
SharedPreferences prefs = getSharedPreferences(prefName,
MODE_PRIVATE);

//--set ukuran font TextView ke nilai yang terseimpan
sebelumnya---
float fontSize = prefs.getFloat(FONT_SIZE_KEY, 12);

//--inisialisasi SeekBar dan EditText---
seekBar.setProgress((int) fontSize);
editText.setText(prefs.getString(TEXT_VALUE_KEY, ""));
editText.setTextSize(seekBar.getProgress());

seekBar.setOnSeekBarChangeListener(new
OnSeekBarChangeListener() {
    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
    }

    @Override
    public void onProgressChanged(SeekBar seekBar, int
progress, boolean fromUser) {
        //---mengubah ukuran font EditText---
        editText.setTextSize(progress);
    }
});
}
}

```

Jalankan aplikasi ke Emulator Android. Tuliskan beberapa teks ke view EditText, ubah ukuran font melalui view SeekBar. Kemudian klik Save.



**Gambar 8.1 ObjectPreferences**

Kembali ke Eclipse dan tekan lagi tombol F11 untuk menjalankan aplikasi ke Emulator (atau jalankan aplikasi ObjectPreferences melalui menu Launcher). Aplikasi sekarang menampilkan teks yang sama seperti yang dituliskan sebelumnya namun memiliki ukuran font yang lebih besar, sesuai dengan ukuran yang disimpan sebelumnya.



**Gambar 8.2 Hasil setelah perubahan dengan ObjectPreferences**

Bagaimana program bekerja? Di dalam object SharedPreferences, gunakan method getSharedPreferences(), melewatkna nama file shared preferences (sehingga semua yang ada di dalamnya akan disimpan) pada argument pertama dan melewatkna modenya (menentukan bagaimana file seharusnya dibuka).

```
private SharedPreferences prefs;  
...  
//---mendapatkan object SharedPreferences---  
prefs = getSharedPreferences(prefName, MODE_PRIVATE);  
SharedPreferences.Editor editor = prefs.edit();
```

Konstanta MODE\_PRIVATE menandakan bahwa file shared preferences hanya bisa dibuka melalui aplikasi yang telah dibuat. Dengan Class Editor dapat disimpan pasangan key/value ke file preference melalui layanan-layanan method seperti: putString(), putBoolean(), putLong(),.putInt(), putFloat().

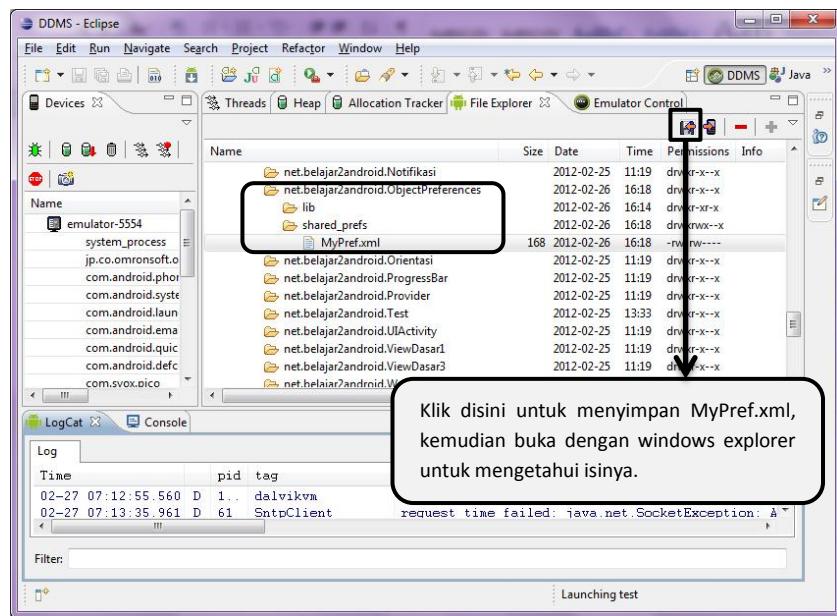
Saat nilai-nilai telah tersimpan, selanjutnya memanggil method commit() untuk menyimpan perubahan:

```
//---menyimpan nilai dalam view EditText ke preferences---  
editor.putFloat(FONT_SIZE_KEY, editText.getTextSize());  
editor.putString(TEXT_VALUE_KEY, editText.getText().toString());  
  
//---menyimpan nilai---  
editor.commit();
```

Ketika activity di-load, pertama ditemui object SharedPreferences untuk mencari semua nilai-nilai yang pernah disimpan sebelumnya:

```
//---load object SharedPreferences---  
SharedPreferences prefs = getSharedPreferences(prefName,  
MODE_PRIVATE);  
  
//---set ukuran font TextView ke nilai yang terseimpan  
sebelumnya---  
float fontSize = prefs.getFloat(FONT_SIZE_KEY, 12);  
  
//---inisialisasi SeekBar dan EditText---  
seekBar.setProgress((int) fontSize);  
editText.setText(prefs.getString(TEXT_VALUE_KEY, ""));  
editText.setTextSize(seekBar.getProgress());
```

File shared preferences disimpan dalam format file XML, diletakkan di dalam folder /data/data/<package\_name>/shared\_prefs. Untuk melihatnya klik menu Eclipse: **Window | Open Perspective | DDMS**, kemudian pilih tab File Explorer. Seperti tampak berikut ini:



**Gambar 8.3 Menyimpan Informasi ke MyPref.XML**

Adapun isi yang ada di dalam file MyPref.xml adalah sebagai berikut:

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<string name="textvalue">Tulisan ini akan disimpan</string>
<float name="fontsize" value="57.0" />
</map>
```

### 8.1.2 Penggunaan getPreferences()

Dibagian sebelumnya telah digunakan object Shared Preferences melalui penyediaan nama variabel ke object tersebut, seperti berikut ini:

```
//---mendapatkan object Shared Preferences---
prefs = getSharedPreferences(prefName, MODE_PRIVATE);
```

Dalam kasus ini informasi yang tersimpan di dalam object SharedPreferences dapat dilihat oleh semua activity-activity yang terdapat dalam satu aplikasi. Jika tidak ingin berbagi data diantara activity-activity, maka dapat digunakan method **getPreferences()**, seperti berikut ini:

```
//---get the SharedPreferences object---  
prefs = getPreferences(MODE_PRIVATE);
```

method **getPreferences()** tidak membutuhkan sebuah nama dan data disimpan secara terbatas di activity yang menciptakannya.

## 8.2 Pengelolaan Data Ke File

Object SharePreferences cocok digunakan untuk menyimpan data sederhana seperti user ID, tanggal lahir, jenis kelamin, no ktp, dan lain-lain. Tapi terkadang perlu juga menggunakan file system sederhana untuk menyimpan data. Misalnya untuk menyimpan puisi berupa teks panjang yang akan ditampilkan pada aplikasi. Dalam Android, Anda dapat menggunakan class java.io untuk membuatnya.

### 8.2.1 Menyimpan Data Ke Media Penyimpanan Internal

Cara pertama untuk menyimpan file dalam aplikasi Android adalah menulis ke media penyimpanan internal perangkat.

Buat project baru bernama **File**.

<b>Project name</b>	File
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	File
<b>Package name</b>	net.belajar2android.File
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```
<?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="Tuliskan beberapa kalimat" />
<EditText android:id="@+id/txtText1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>
<Button android:id="@+id btnSave"
    android:text="Save"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>
<Button android:id="@+id btnLoad"
    android:text="Load"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>
</LinearLayout>
```

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.File;

import android.app.Activity;
import android.os.Bundle;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {
    private EditText textBox;
    private static final int READ_BLOCK_SIZE = 100;

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

        textBox = (EditText) findViewById(R.id.txtText1);
        Button saveBtn = (Button) findViewById(R.id.btnSave);
```

```

Button loadBtn = (Button) findViewById(R.id.btnLoad);
saveBtn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        String str = textBox.getText().toString();
        try {
            FileOutputStream fOut =
openFileOutput("file_teks.txt", MODE_WORLD_READABLE);
            OutputStreamWriter osw = new
OutputStreamWriter(fOut);

            //---menulis string ke file---
osw.write(str);
osw.flush();
osw.close();

            //---menampilkan file yang telah menyimpan pesan---
Toast.makeText(getApplicationContext(), "File sukses
tersimpan!", Toast.LENGTH_SHORT).show();

            //---membersihkan EditText---
textBox.setText("");
        }
        catch (IOException e){
            e.printStackTrace();
        }
    }
});
loadBtn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        try {
            FileInputStream fIn = openFileInput("file_teks.txt");
InputStreamReader isr = new InputStreamReader(fIn);

char[] inputBuffer = new char[READ_BLOCK_SIZE];
String s = "";

int charRead;
while ((charRead = isr.read(inputBuffer)) > 0)
{
    //---konversi karakter ke string---
    String readString =
String.valueOf(inputBuffer, 0, charRead);
    s += readString;

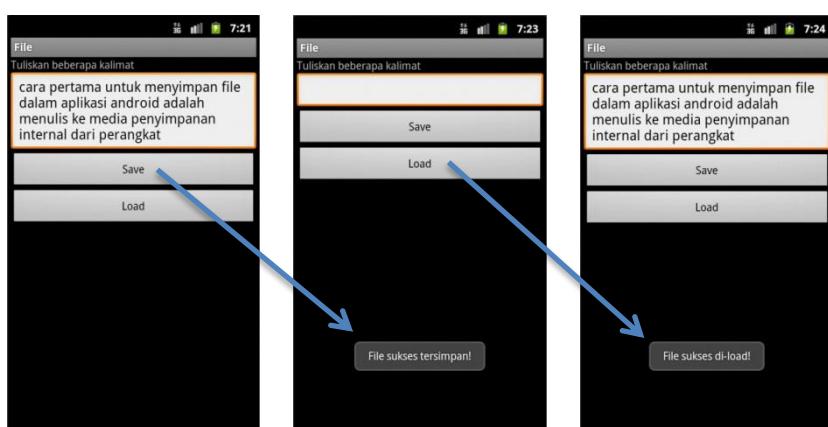
    inputBuffer = new char[READ_BLOCK_SIZE];
}
//---beri nilai EditText ke teks yang telah dibaca---
textBox.setText(s);

Toast.makeText(getApplicationContext(), "File sukses di-
load!", Toast.LENGTH_SHORT).show();
    }
}

```

```
        catch(IOException e) {
            e.printStackTrace();
        }
    });
}
}
```

Jalankan aplikasi ke emulator Android. Tuliskan beberapa kalimat ke TextBox. Klik tombol Save, maka akan muncul pesan “**File sukses tersimpan**”, ini berarti teks telah tersimpan ke dalam file teks. Klik tombol Load, maka akan muncul pesan “**File sukses di load**”, ini berarti teks berhasil di-load dari tempat penyimpanan untuk ditempatkan di TextBox.



**Gambar 8.4 Penyimpanan ke file**

Bagaimana program bekerja? Untuk menyimpan teks ke file, gunakan class FileOutputStream. Method openFileOutput() membuka nama file untuk ditulisi dengan mode yang telah ditentukan. Disini digunakan konstanta MODE\_WORLD\_READABLE yang berarti bahwa file dapat dibaca oleh semua aplikasi lainnya:

```
FileOutputStream fOut = openFileOutput("file_teks.txt",  
MODE_WORLD_READABLE);
```

Selain konstanta MODE\_WORLD\_READABLE dapat pula diganti dengan: MODE\_PRIVATE (file hanya dapat diakses oleh aplikasi yang menciptakannya), MODE\_APPEND (untuk menambahkan ke file yang sudah ada) dan MODE\_WORLD\_WRITABLE (seluruh aplikasi mempunyai akses untuk menulis ke file).

Untuk mengkonversi sebuah character stream ke byte stream digunakan instance class OutputStreamWriter dengan cara melewatkkan instance object FileOutputStream:

```
OutputStreamWriter osw = new OutputStreamWriter(fOut);
```

Selanjutnya, gunakan method write() untuk menulis string ke file. Untuk memastikan bahwa semua byte tertulis ke dalam file digunakan method flush(). Terakhir, gunakan method close() untuk menutup file:

```
//---menulis string ke file---
osw.write(str);
osw.flush();
osw.close();
```

Untuk membaca isi file, digunakan class FileInputStream bersama-sama dengan class InputStreamReader:

```
FileInputStream fIn = openFileInput("file_teks.txt");
InputStreamReader isr = new InputStreamReader(fIn);
```

Isi file yang terbaca berbentuk blok, per bloknya terdiri dari 100 karakter yang diletakkan dalam buffer (array character), kemudian character di-copy ke object String:

```
char[] inputBuffer = new char[READ_BLOCK_SIZE];
String s = "";

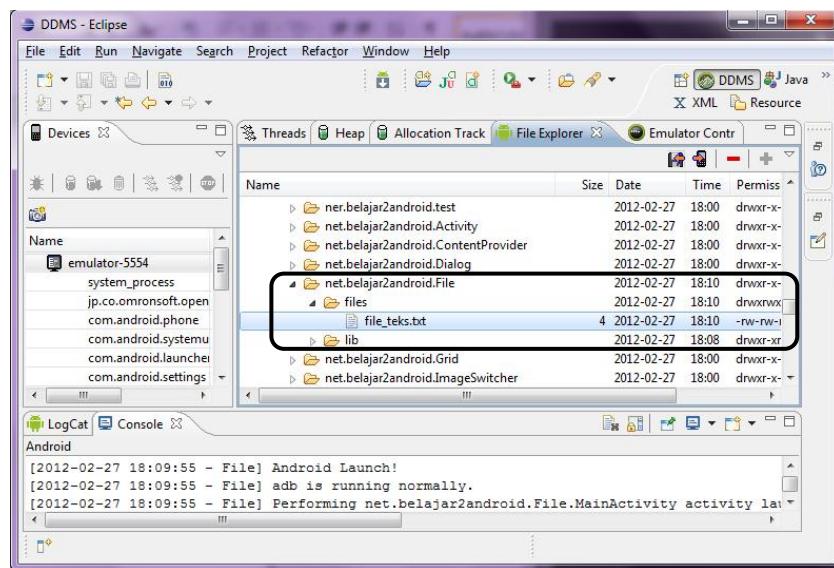
int charRead;
while ((charRead = isr.read(inputBuffer)) > 0)
{
    //---konversi karakter ke string---
    String readString = String.valueOf(inputBuffer, 0,
charRead);
    s += readString;

    inputBuffer = new char[READ_BLOCK_SIZE];
```

```
}
```

Method read() dari object InputStreamReader membaca banyaknya character yang terbaca dan mengembalikan -1 jika sudah mencapai akhir file.

Untuk membuktikan apakah aplikasi benar-benar menyimpan file ke direktori atau tidak dapat digunakan DDMS. Letak direktori berada di **/data/data/net.belajar2android.File/files**



**Gambar 8.5 Menyimpan file ke DDMS**

Adapun isi file file\_teks.txt adalah sebagai berikut: “cara pertama untuk menyimpan file dalam aplikasi android adalah menulis ke media penyimpanan internal dari perangkat”.

### **8.2.2 Menyimpan Data Ke Media Penyimpanan Eksternal (SD Card)**

Terkadang penyimpanan perlu juga dilakukan pada media penyimpanan Eksternal (seperti SD Card) karena ukuran kapasitasnya yang besar, sehingga memiliki kemampuan yang baik untuk berbagi file-file dengan user lainnya (dengan cara melepas SD card dari perangkat, kemudian memasukkannya ke perangkat Android lainnya).

Untuk mencobanya gunakan project sebelumnya (project File), modifikasi method onClick pada button Save:

```
saveBtn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        String str = textBox.getText().toString();
        try {
            //---SD Card Storage
            File sdCard = Environment.getExternalStorageDirectory();
            File directory = new File(sdCard.getAbsolutePath() +
"MyFile");
            directory.mkdirs();
            File file = new File(directory, "file_teks.txt");
            FileOutputStream fOut = new FileOutputStream(file);

            //FileOutputStream fOut = openFileOutput("file_teks.txt",
            MODE_WORLD_READABLE);
            OutputStreamWriter osw = new OutputStreamWriter(fOut);

            //---menulis string ke file---
            osw.write(str);
            osw.flush();
            osw.close();

            //---menampilkan file yang telah menyimpan pesan---
            Toast.makeText(getApplicationContext(), "File sukses
tersimpan!", Toast.LENGTH_SHORT).show();

            //---membersihkan EditText---
            textBox.setText("");
        }
        catch (IOException e){
            e.printStackTrace();
        }
    }
});
```

Method `getExternalStorageDirectory()` mengembalikan alamat path ke media penyimpanan eksternal. Path “**/sdcard**” untuk perangkat Android sesungguhnya dan “**mnt/sdcard**” untuk Emulator Android. Sebaiknya

jangan menuliskan direktori secara *hardcode* (menulis alamat path secara langsung dengan kode program) ke SD card, karena beberapa manfaktur bisa jadi menempatkan path berbeda untuk setiap SD card. Oleh karena itu gunakan method `getExternalStorageDirectory()` untuk mengembalikan alamat path secara lengkap ke SD card.

Selanjutnya ciptakan direktori **MyFile** ke dalam SD card dan simpan file ke direktori tersebut.

Untuk me-load file dari media penyimpanan eksternal, modifikasi method `onClick()` dari button Load:

```
loadBtn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        try {
            //---SD Storage---
            File sdCard = Environment.getExternalStorageDirectory();
            File directory = new File(sdCard.getAbsolutePath() +
"/MyFile");
            File file = new File(directory, "file_teks.txt");
            FileInputStream fIn = new FileInputStream(file);
            InputStreamReader isr = new InputStreamReader(fIn);

            //FileInputStream fIn = openFileInput("file_teks.txt");
            //InputStreamReader isr = new InputStreamReader(fIn)

            char[] inputBuffer = new char[READ_BLOCK_SIZE];
            String s = "";

            int charRead;
            while ((charRead = isr.read(inputBuffer)) > 0)
            {
                //---konversi karakter ke string---
                String readString = String.valueOf(inputBuffer, 0,
charRead);
                s += readString;

                inputBuffer = new char[READ_BLOCK_SIZE];
            }
            //---beri nilai EditText ke teks yang telah dibaca---
            textBox.setText(s);
            Toast.makeText(getApplicationContext(), "File sukses di-load!",
Toast.LENGTH_SHORT).show();
        }
        catch(IOException e) {
            e.printStackTrace();
        }
    }
})
```

```
});
```

Untuk menyimpan data ke media penyimpanan eksternal perlu ditambahkan izin WRITE\_EXTERNAL\_STORAGE di dalam file AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.belajar2android.File"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE">
        </uses-permission>
        <application android:icon="@drawable/icon"
            android:label="@string/app_name">
            <activity android:name=".MainActivity"
                android:label="@string/app_name">
                <intent-filter>
                    <action android:name="android.intent.action.MAIN" />
                    <category
                        android:name="android.intent.category.LAUNCHER" />
                </intent-filter>
            </activity>
        </application>
    </manifest>
```

### 8.2.3 Memilih Media Penyimpanan Yang Tepat

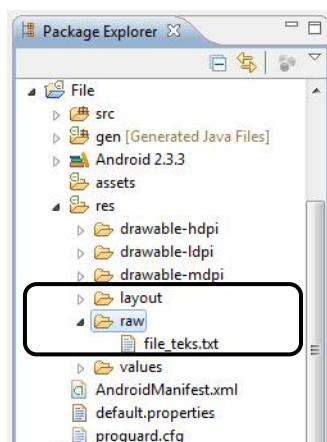
Dari beberapa pilihan cara menyimpan, mana yang paling tepat untuk aplikasi Anda? Berikut ini beberapa saran:

- Jika data hanya mewakili pasangan key/value (kunci/nilai) maka sebaiknya gunakan object SharedPreferences. Contoh, ketika menyimpan data user preference seperti nama user, warna background, tanggal lahir, tanggal login terakhir dan lain-lain.
- Jika memerlukan data yang sifatnya mendadak maka media penyimpanan internal merupakan pilihan yang baik. Contoh, ketika aplikasi perlu men-download gambar dari web, kemudian menampilkannya.

- Jika hendak berbagi data aplikasi kepada user lain. Contoh, ketika aplikasi Android yang mencatat koordinat suatu lokasi dan Anda ingin berbagi data ke user lain, maka sebaiknya simpan file-file di dalam SD Card perangkat untuk ditransfer ke perangkat lain (komputer, perangkat Android).

#### **8.2.4 Menggunakan Sumberdaya Statis**

Disamping pembentukan file secara dinamis saat program berjalan (run time), dimungkinkan pula membentuk file melalui *package project* saat proses desain (design time). Contoh, penyatuan beberapa file pertolongan (*help files*) ke package sehingga dapat ditampilkan beberapa pesan pertolongan ketika user memerlukannya. File-file tersebut harus ditambahkan ke dalam folder res/raw (folder yang dibuat sendiri)



**Gambar 8.6 Penambahan file ke package**

Agar file bisa digunakan dalam kode, gunakan method getResources untuk mengembalikan sebuah object Resources, kemudian gunakan method getResources untuk membuka file yang terdapat di dalam folder res/raw:

```

import java.io.InputStream;
import java.io.BufferedReader;

public class MainActivity extends Activity {
    private EditText textBox;
    private static final int READ_BLOCK_SIZE = 100;

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

        InputStream is =
this.getResources().openRawResource(R.raw.file_teks);
        BufferedReader br = new BufferedReader(new
InputStreamReader(is));
        String str = null;
        try {
            while ((str = br.readLine()) != null) {
                Toast.makeText(getApplicationContext(), str,
Toast.LENGTH_SHORT).show();
            }
            is.close();
            br.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

        textBox = (EditText) findViewById(R.id.txtText1);
        Button saveBtn = (Button) findViewById(R.id.btnSave);
        Button loadBtn = (Button) findViewById(R.id.btnLoad);

        saveBtn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
            });
        });

        loadBtn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
            });
        });
    }
}

```

ID sumberdaya adalah sumberdaya yang tersimpan di dalam folder res/raw yang diambil namanya namun tanpa disertai extensi file (tanpa .txt). Misal, jika file teksnya adalah file\_teks.txt maka sumberdaya ID nya adalah R.raw.file\_teks.

## 8.3 Penciptaan Dan Penggunaan Database

Untuk menyimpan data-data yang saling berhubungan (relational database), maka sebaiknya gunakan database karena jauh lebih efisien, dimana Android memanfaatkan SQLite sebagai database-nya. Database yang dibangun untuk aplikasi hanya bisa diakses oleh aplikasi itu sendiri; sedangkan aplikasi lain tidak dapat mengaksesnya.

Database SQLite yang dibentuk secara program selalu tersimpan di dalam folder `/data/data/<nama_package>/databases`.

### 8.3.1 Menciptakan Class Pembantu DBAdapter

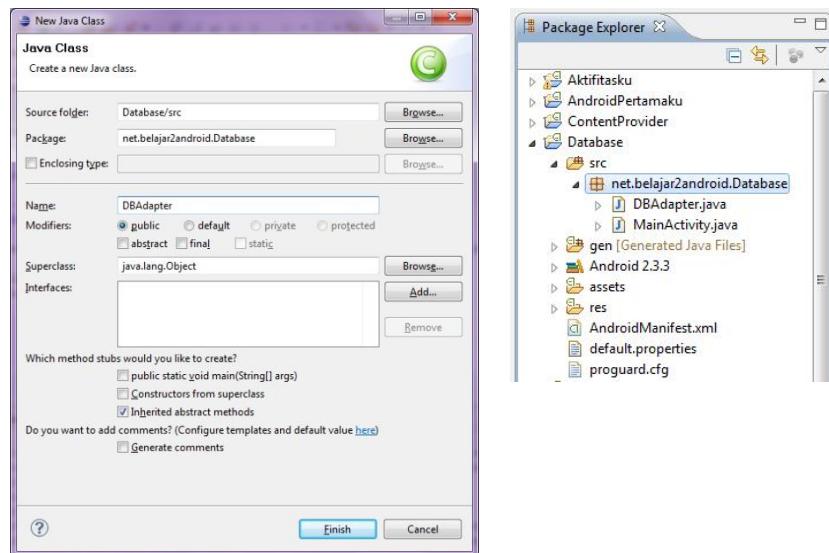
Cara terbaik bila berurusan dengan database adalah membuat class pembantu (helper class) terlebih dahulu. Class tersebut membungkus segala kerumitan pengaksesan data sehingga menjadi lebih jelas saat pemanggilan kode. Class pembantu disebut DBAdapter yang bertugas untuk menciptakan, membuka, menutup dan menggunakan database SQLite.

Dalam contoh ini akan diciptakan database bernama **MyDB** yang berisi satu table bernama **kontak**. Table ini memiliki tiga kolom, antara lain **\_id**, **nama** dan **email**.

Buat project baru bernama **Database**.

<b>Project name</b>	Database
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	Database
<b>Package name</b>	net.belajar2android.Database
<b>Create Activity</b>	MainActivity

Tambahkan class baru bernama **DBAdapter.java** ke dalam project Database. Klik menu File|New|Class pada Eclipse. File diletakkan dalam folder **Database/src**, kemudian klik tombol Finish.



**Gambar 8.7 Class DBAdapter**

Modifikasi kode program yang terdapat dalam DBAdapter.java:

```
package net.belajar2android.Database;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class DBAdapter {
    public static final String KEY_ROWID = "_id";
    public static final String KEY_NAME = "nama";
    public static final String KEY_EMAIL = "email";
    public static final String TAG = "DBAdapter";

    private static final String DATABASE_NAME = "MyDB";
    private static final String DATABASE_TABLE = "kontak";
```

```
private static final int DATABASE_VERSION = 1;

private static final String DATABASE_CREATE =
    "create table kontak (_id integer primary key autoincrement,
" + "nama text not null, email text not null);";

private final Context context;

private DatabaseHelper DBHelper;
private SQLiteDatabase db;

public DBAdapter(Context ctx)
{
    this.context = ctx;
    DBHelper = new DatabaseHelper(context);
}

private static class DatabaseHelper extends SQLiteOpenHelper {
    DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        try {
            db.execSQL(DATABASE_CREATE);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion)
    {
        Log.w(TAG, "Upgrade database dari versi " + oldVersion + " ke " +
newVersion + ", yang akan menghapus semua data lama");
        db.execSQL("DROP TABLE IF EXISTS kontak");
        onCreate(db);
    }
}

//---membuka database---
public DBAdapter open() throws SQLException {
    db = DBHelper.getWritableDatabase();
    return this;
}

//---menutup database---
public void close() {
    DBHelper.close();
}

//---memasukkan sebuah kontak ke database---
public long insertContact(String name, String email) {
```

```

ContentValues initialValues = new ContentValues();
initialValues.put(KEY_NAME, name);
initialValues.put(KEY_EMAIL, email);
return db.insert(DATABASE_TABLE, null, initialValues);
}

//---menghapus kontak tertentu---
public boolean deleteContact(long rowId) {
    return db.delete(DATABASE_TABLE, KEY_ROWID + "=" + rowId,
null) > 0;
}

//---menampilkan semua kontak---
public Cursor getAllContacts() {
    return db.query(DATABASE_TABLE, new String[] {KEY_ROWID,
KEY_NAME, KEY_EMAIL}, null, null, null, null, null);
}

//---menampilkan kontak tertentu---
public Cursor getContact(long rowId) throws SQLException {
    Cursor mCursor = db.query(true, DATABASE_TABLE, new String[]
{KEY_ROWID, KEY_NAME, KEY_EMAIL}, KEY_ROWID + "=" + rowId, null,
null, null, null);
    if (mCursor != null) {
        mCursor.moveToFirst();
    }
    return mCursor;
}

//---update kontak---
public boolean updateContact(long rowId, String name, String
email) {
    ContentValues args = new ContentValues();
    args.put(KEY_NAME, name);
    args.put(KEY_EMAIL, email);
    return db.update(DATABASE_TABLE, args, KEY_ROWID + "=" +
rowId, null) > 0;
}
}

```

Bagaimana program bekerja? Pertama dideklarasikan beberapa konstanta untuk mengisi beberapa field table yang dibuat untuk database:

```

public static final String KEY_ROWID = "_id";
public static final String KEY_NAME = "nama";
public static final String KEY_EMAIL = "email";
public static final String TAG = "DBAdapter";

private static final String DATABASE_NAME = "MyDB";
private static final String DATABASE_TABLE = "kontak";
private static final int DATABASE_VERSION = 1;

```

```
private static final String DATABASE_CREATE =
    "create table kontak (_id integer primary key autoincrement,
    + \"nama text not null, email text not null);";
```

Konstanta DATABASE\_CREATE berisi pernyataan SQL untuk menciptakan table **kontak** ke dalam database MyDB.

Dalam class DBAdapter terdapat extend class SQLiteOpenHelper yang merupakan class helper untuk mengelola penciptaan database dan pengaturan versi, terutama penggunaan method **onCreate()** dan **onUpgrade()**:

```
public class DBAdapter {
    public static final String KEY_ROWID = "_id";
    public static final String KEY_NAME = "nama";
    public static final String KEY_EMAIL = "email";
    public static final String TAG = "DBAdapter";
    private static final String DATABASE_NAME = "MyDB";
    private static final String DATABASE_TABLE = "kontak";
    private static final int DATABASE_VERSION = 1;

    private static final String DATABASE_CREATE =
        "create table kontak (_id integer primary key autoincrement,
        + \"nama text not null, email text not null);";

    private final Context context;
    private DatabaseHelper DBHelper;
    private SQLiteDatabase db;

    public DBAdapter(Context ctx)
    {
        this.context = ctx;
        DBHelper = new DatabaseHelper(context);
    }

    private static class DatabaseHelper extends SQLiteOpenHelper {
        DatabaseHelper(Context context) {
            super(context, DATABASE_NAME, null, DATABASE_VERSION);
        }

        @Override
        public void onCreate(SQLiteDatabase db) {
            try {
                db.execSQL(DATABASE_CREATE);
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        @Override
```

```

    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion)
    {
        Log.w(TAG, "Upgrade database dari versi " + oldVersion + "
ke "
        + newVersion + ", yang akan menghapus semua data lama");
        db.execSQL("DROP TABLE IF EXISTS kontak");
        onCreate(db);
    }
}

```

Method `onCreate()` akan menciptakan database baru jika saat database dibutuhkan tidak tersedia. Method `onUpgrade` dipanggil ketika database perlu untuk di-upgrade. Ini tercapai melalui pengecekan nilai yang telah ditetapkan pada konstanta `DATABASE_VERSION`. Penerapan method `onUpgrade` sederhana saja, yaitu saat terjadi penghapusan table maka akan diciptakan table ulang.

Selanjutnya pendefinisian beberapa method untuk membuka database, menutup database dan method untuk menambah, mengubah, dan menghapus baris-baris table:

```

public class DBAdapter {
    //...
    //...

    //---membuka database---
    public DBAdapter open() throws SQLException {
        db = DBHelper.getWritableDatabase();
        return this;
    }

    //---menutup database---
    public void close() {
        DBHelper.close();
    }

    //---memasukkan sebuah kontak ke database---
    public long insertContact(String name, String email) {
        ContentValues initialValues = new ContentValues();
        initialValues.put(KEY_NAME, name);
        initialValues.put(KEY_EMAIL, email);
        return db.insert(DATABASE_TABLE, null, initialValues);
    }

    //---menghapus kontak tertentu---
    public boolean deleteContact(long rowId) {

```

```

        return db.delete(DATABASE_TABLE, KEY_ROWID + "=" + rowId,
null) > 0;
    }

    //---menampilkan semua kontak---
    public Cursor getAllContacts() {
        return db.query(DATABASE_TABLE, new String[] {KEY_ROWID,
KEY_NAME, KEY_EMAIL}, null, null, null, null, null);
    }

    //---menampilkan kontak tertentu---
    public Cursor getContact(long rowId) throws SQLException {
        Cursor mCursor = db.query(true, DATABASE_TABLE, new String[]
{KEY_ROWID, KEY_NAME, KEY_EMAIL}, KEY_ROWID + "=" + rowId, null,
null, null, null);
        if (mCursor != null) {
            mCursor.moveToFirst();
        }
        return mCursor;
    }

    //---update kontak---
    public boolean updateContact(long rowId, String name, String
email) {
        ContentValues args = new ContentValues();
        args.put(KEY_NAME, name);
        args.put(KEY_EMAIL, email);
        return db.update(DATABASE_TABLE, args, KEY_ROWID + "=" +
rowId, null) > 0;
    }
}

```

**Catatan:** Android menggunakan class Cursor untuk mengembalikan nilai query (*result set*). Bisa dibayangkan bahwa Cursor adalah penunjuk hasil query database. Dengan Cursor maka pengelolaan baris dan kolom table menjadi lebih efisien.

Gunakan object **ContentValues** untuk menyimpan pasangan key/value. Dalam object ContentValues terdapat method **put()** sehingga dapat ditambahkan key dengan nilai yang berbeda tipe data.

### 8.3.2 Menambahkan Kontak

Sekarang kita telah siap menggunakan database dengan bantuan class pembantu yang telah dibahas sebelumnya.

Gunakan project sebelumnya (project Database), modifikasi file `MainActivity.java` seperti berikut ini:

```
package net.belajar2android.Database;

import android.app.Activity;
import android.os.Bundle;

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

        //---menambah kontak---
        db.open();
        long id = db.insertContact("Dodit Suprianto",
        "doditsuprianto@gmail.com");
        id = db.insertContact("Rini Agustina", "ryf4ny@yahoo.com");
        db.close();
    }
}
```

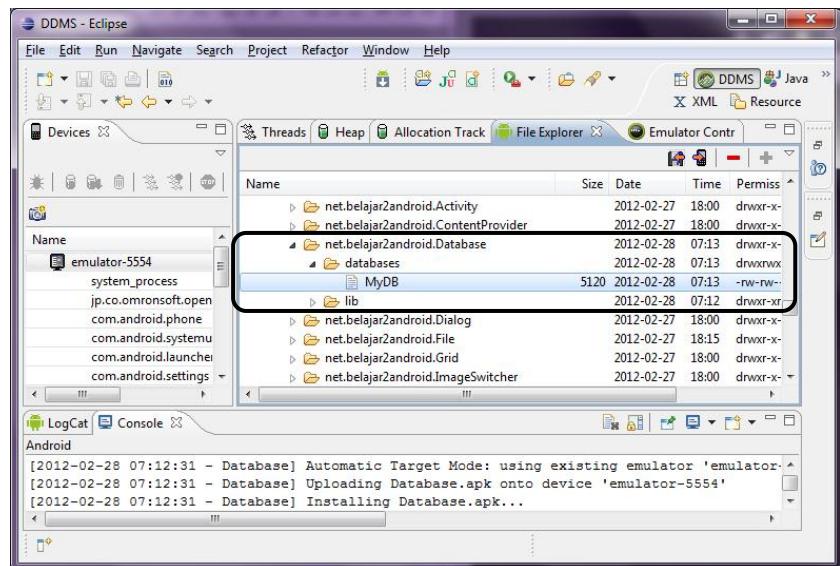
Jalankan aplikasi ke Emulator Android.

Bagaimana program bekerja? Pertama Anda menciptakan sebuah instance class `DBAdapter`:

```
DBAdapter db = new DBAdapter(this);
```

Method `insertContact()` mengembalikan ID pada setiap penambahan baris. Jika terjadi kesalahan selama pengoperasian maka ia akan mengembalikan nilai -1.

Bila diamati dalam DDMS, terlihat bahwa database **MyDB** telah terbentuk di bawah folder **databases**, seperti berikut ini:



**Gambar 8.8 File database dalam DDMS**

### 8.3.3 Menampilkan Seluruh Kontak

Untuk menampilkan seluruh kontak yang terdapat dalam table kontak, maka gunakan method **getAllContacts** dari class DBAdapter.

Masih tetap menggunakan project sebelumnya (project Database), modifikasi file MainActivity.java seperti berikut ini:

```

package net.belajar2android.Database;

import android.app.Activity;
import android.os.Bundle;
import android.widget.Toast;
import android.database.Cursor;

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

        DBAdapter db = new DBAdapter(this);
        /*
    
```

```

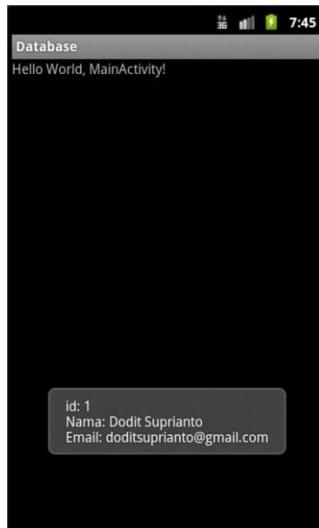
//---menambah kontak---
db.open();
long id = db.insertContact("Dodit Suprianto",
"doditsuprianto@gmail.com");
id = db.insertContact("Rini Agustina", "ryf4ny@yahoo.com");
db.close();
*/

//--- mendapatkan kontak ---
db.open();
Cursor c = db.getAllContacts();
if (c.moveToFirst()) {
    do {
        TampilkanKontak(c);
    } while (c.moveToNext());
}
db.close();
}

public void TampilkanKontak(Cursor c) {
    Toast.makeText(this, "id: " + c.getString(0) + "\n" + "Nama:
" + c.getString(1) + "\n" + "Email: " + c.getString(2),
Toast.LENGTH_LONG).show();
}
}

```

Jalankan aplikasi ke Emulator Android. class Toast akan menampilkan seluruh kontak yang terdapat dalam table kontak.



**Gambar 8.9 Menampilkan seluruh kontak dari table**

Bagaimana program bekerja? Method getAllContacts dari class DBAdapter menampilkan seluruh kontak yang tersimpan dalam database. Hasil yang dikembalikan berupa object Cursor. Untuk menampilkan seluruh kontak, pertama dipanggil method moveFirst dari object Cursor. Jika sukses (paling tidak terdapat satu baris/record), maka tampilkan detail kontaknya dengan menggunakan method TampilkanKontak(). Untuk menggeser cursor ke kontak selanjutnya, dipanggil method moveNext() dari object Cursor.

### 8.3.4 Menampilkan Satu Kontak

Untuk menampilkan satu kontak berdasarkan ID-nya maka panggil method getContact() dari class DBAdapter.

Gunakan project sebelumnya (Database), modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.Database;

import android.app.Activity;
import android.os.Bundle;
import android.widget.Toast;
import android.database.Cursor;

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

        DBAdapter db = new DBAdapter(this);

        /*
        //---menambah kontak---
        //...
        */

        /*
        //---mendapatkan seluruh kontak---
        //...
        */

        //---mendapatkan sebuah kontak---
        db.open();
    }
}
```

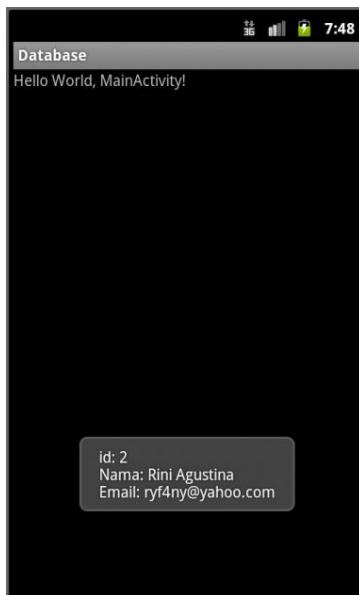
```

Cursor c = db.getContact(2);
if (c.moveToFirst())
    TampilkanKontak(c);
else
    Toast.makeText(this, "Kontak tidak ditemukan",
Toast.LENGTH_LONG).show();
db.close();
}

public void TampilkanKontak(Cursor c)
{
    Toast.makeText(this, "id: " + c.getString(0) + "\n" + "Nama:
" + c.getString(1) + "\n" + "Email: " + c.getString(2),
Toast.LENGTH_LONG).show();
}

```

Jalankan aplikasi ke Emulator. Class Toast akan menampilkan baris/record kontak kedua (id = 2), seperti tampak berikut ini:



**Gambar 8.10 Menampilkan satu kontak dari table**

Bagaimana program bekerja? Method `getContact()` dari class `DBAdapter` menampilkan satu kontak berdasarkan ID-nya, yaitu melewatkkan ID ke-2 yang berarti menampilkan kontak kedua:

```
Cursor c = db.getContact(2);
```

Hasil yang dikembalikan berupa object Cursor. Jika sebuah baris dikembalikan maka akan ditampilkan detil kontak dengan menggunakan method TampilkanKontak().

### 8.3.5 Meng-Update Kontak

Untuk mengubah/update kontak tertentu, gunakan method updateContact() dari class DBAdapter dengan cara melewatkkan ID kontak yang hendak di-update.

Tetap menggunakan project sebelumnya (project Database), modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.Database;

import android.app.Activity;
import android.os.Bundle;
import android.widget.Toast;
import android.database.Cursor;

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

        DBAdapter db = new DBAdapter(this);

        /*
        //---menambah kontak---
        //...
        */

        /*
        //---mendapatkan seluruh kontak---
        //...
        */

        /*
        //---mendapatkan sebuah kontak---
        //...
        */

        //---update kontak---
        db.open();
    }
}
```

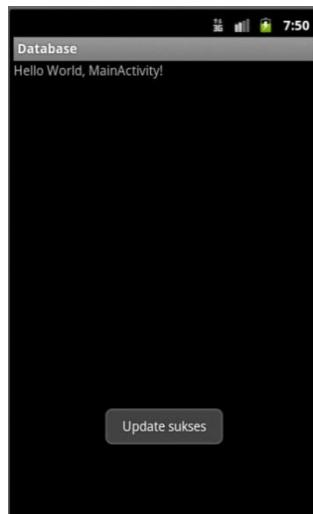
```

        if (db.updateContact(1, "Dodit Suprianto",
"d0dit@yahoo.com"))
            Toast.makeText(this, "Update sukses",
Toast.LENGTH_LONG).show();
        else
            Toast.makeText(this, "Update gagal!",
Toast.LENGTH_LONG).show();
        db.close();
    }

    public void TampilkanKontak(Cursor c)
{
    Toast.makeText(this, "id: " + c.getString(0) + "\n" + "Nama:
" + c.getString(1) + "\n" + "Email: " + c.getString(2),
Toast.LENGTH_LONG).show();
}
}

```

Jalankan aplikasi ke Emulator Android. Sebuah pesan akan ditampilkan jika update sukses.



**Gambar 8.11 Update kontak**

Bagaimana program bekerja? Method `updateContact()` dari class `DBAdapter` mengubah detil kontak melalui ID kontak yang hendak diubah. Hal ini akan mengembalikan nilai Boolean yang berarti bahwa update sukses.

### 8.3.6 Menghapus Kontak

Untuk menghapus sebuah kontak, gunakan method deleteContact() dari class DBAdapter dengan cara melewatkna ID kontak yang hendak dihapus.

Masih tetap menggunakan project sebelumnya (project Database), modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.Database;

import android.app.Activity;
import android.os.Bundle;
import android.widget.Toast;
import android.database.Cursor;

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

        DBAdapter db = new DBAdapter(this);

        /*
        //---menambah kontak---
        //...
        */

        /*
        //---mendapatkan seluruh kontak---
        //...
        */

        /*
        //---mendapatkan sebuah kontak---
        //...
        */

        /*
        //---update kontak---
        //...
        */

        //---menghapus kontak---
        db.open();
        if (db.deleteContact(1))
            Toast.makeText(this, "Penghapusan sukses",
            Toast.LENGTH_LONG).show();
        else
    }
}
```

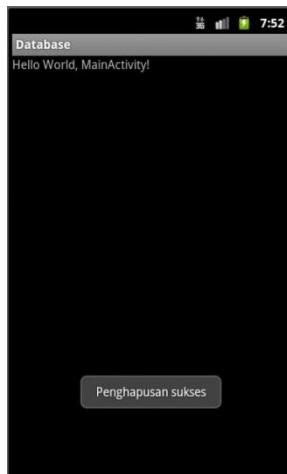
```

        Toast.makeText(this, "Penghapusan gagal",
        Toast.LENGTH_LONG).show();
        db.close();
    }

    public void TampilKontak(Cursor c)
    {
        Toast.makeText(this, "id: " + c.getString(0) + "\n" + "Nama:
" + c.getString(1) + "\n" + "Email: " + c.getString(2),
        Toast.LENGTH_LONG).show();
    }
}

```

Jalankan aplikasi ke Emulator. Sebuah pesan akan ditampilkan jika penghapusan sukses dilakukan.



**Gambar 8.12 Penghapusan kontak**

Bagaimana program bekerja? Untuk menghapus kontak berdasarkan ID-nya, gunakan Method deleteContact() dari class DBAdapter. Hal ini akan mengembalikan nilai Boolean yang menandakan bahwa penghapusan sukses dilakukan.

### 8.3.7 Updageade Database

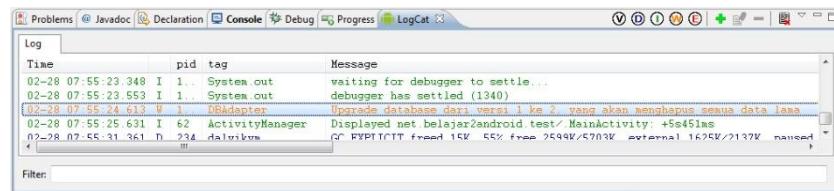
Terkadang setelah menciptakan dan menggunakan database, diperlukan penambahan beberapa table, mengubah skema database atau

menambahkan kolom/field table. Oleh karenanya diperlukan migrasi data dari database lama ke database lebih baru.

Untuk meng-upgrade database, ubahlah konstanta DATABASE\_VERSION ke nilai yang lebih tinggi dari sebelumnya. Contoh, jika nilai sebelumnya adalah 1 maka ubah menjadi 2:

```
public class DBAdapter {  
    public static final String KEY_ROWID = "_id";  
    public static final String KEY_NAME = "nama";  
    public static final String KEY_EMAIL = "email";  
    public static final String TAG = "DBAdapter";  
  
    private static final String DATABASE_NAME = "MyDB";  
    private static final String DATABASE_TABLE = "kontak";  
    private static final int DATABASE_VERSION = 2;  
}
```

Ketika Anda menjalankan aplikasi sekali lagi maka akan muncul pesan dalam Eclipse window perspective LogCat seperti berikut ini:



**Gambar 8.13 Window perspective Logcat**

```
07-20 07:55:23.348 I 1.. System.out waiting for debugger to settle...  
07-20 07:55:23.553 I 1.. System.out debugger has settled (1340)  
07-20 07:55:24.613 W 1 DBAdapter Upgrade database dari versi 1 ke 2, yang akan menghapus semua data lama  
07-20 07:55:25.631 I 62 ActivityManager Displayed net.belajar2android.test/.MainActivity +5s45ms  
07-20 07:55:31.361 D 234 dalvikvm  
" "
```

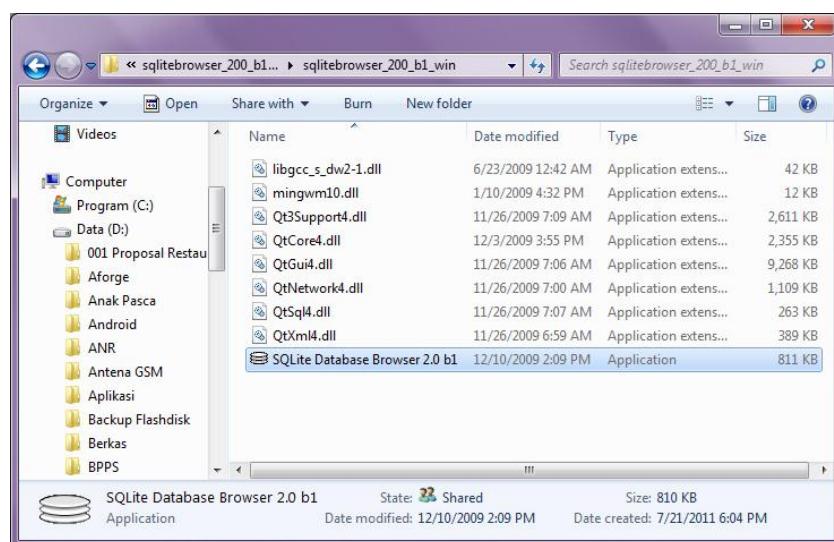
Pada contoh di atas, sederhananya adalah melakukan penghapusan table yang sudah ada (drop table) dan kemudian menciptakan table baru.

### **8.3.8 Menciptakan Database SQLite Dengan Tool “SQLite Database Browser”**

Dalam aplikasi sesungguhnya akan lebih efisien bila menciptakan database diawal, yaitu saat *design time* daripada saat *run time* (desain

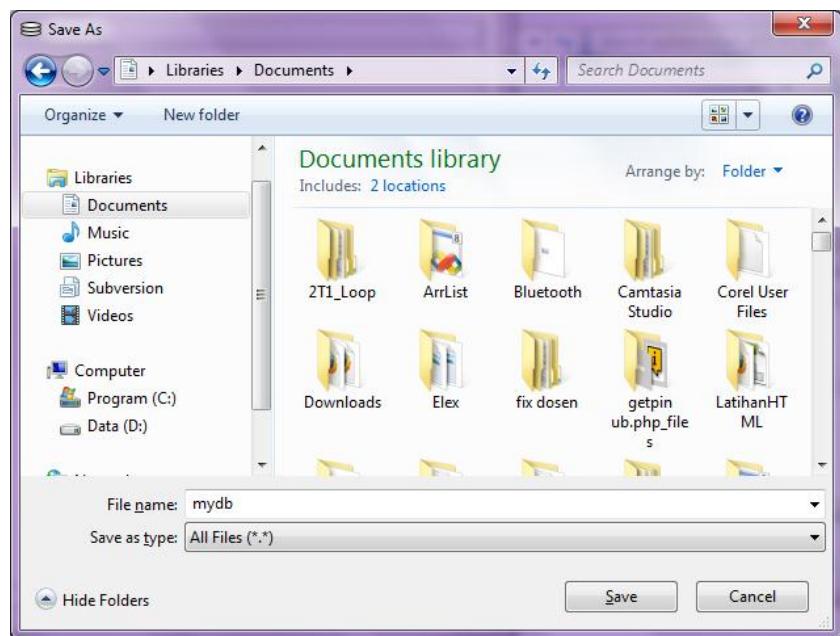
database lebih efisien jika dibentuk dari luar Android tanpa menggunakan program). Untuk mendesain database SQLite, sebaiknya gunakan software bantu, contohnya **SQLite Database Browser** yang bisa didownload di <http://sourceforge.net/projects/sqlitebrowser/>.

Berikut tahapan membuat database SQLite melalui SQLite Database Browser:



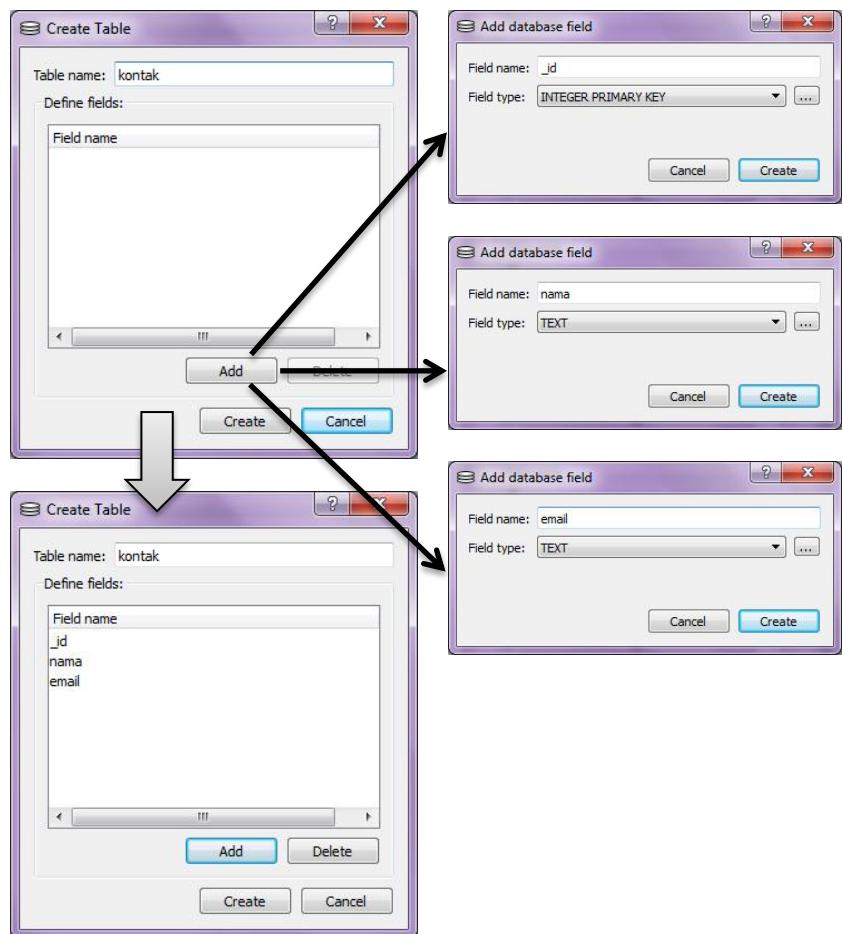
**Gambar 8.14 SQLite Database Browser**

Klik menu **File|New Database** pada SQLite Database Browser. Beri nama database-nya, contoh **mydb**, kemudian klik tombol **Save**.

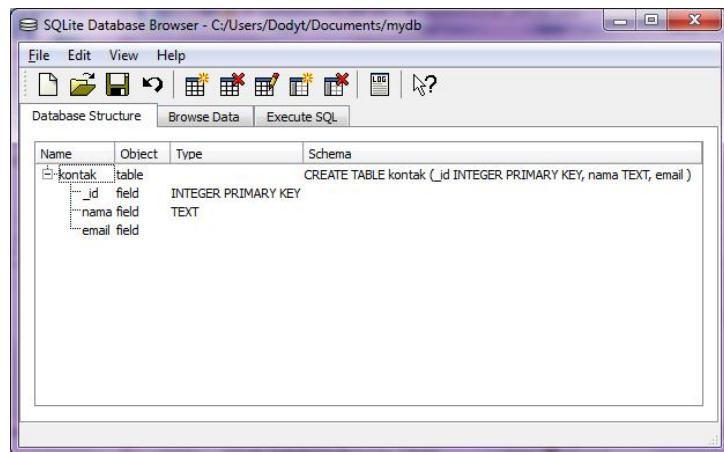


**Gambar 8.15 Membentuk file database dengan SQLite**

Bentuk table baru dan beri nama table-nya, contoh **kontak**. Klik tombol **Add** untuk menambahkan field-field beserta tipe datanya. Dalam kasus ini terdapat tiga field yaitu id, nama dan email.



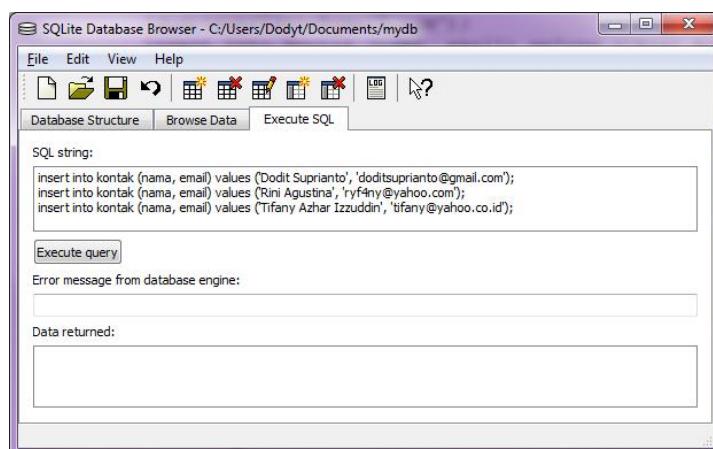
**Gambar 8.16 Menciptakan table beserta field**



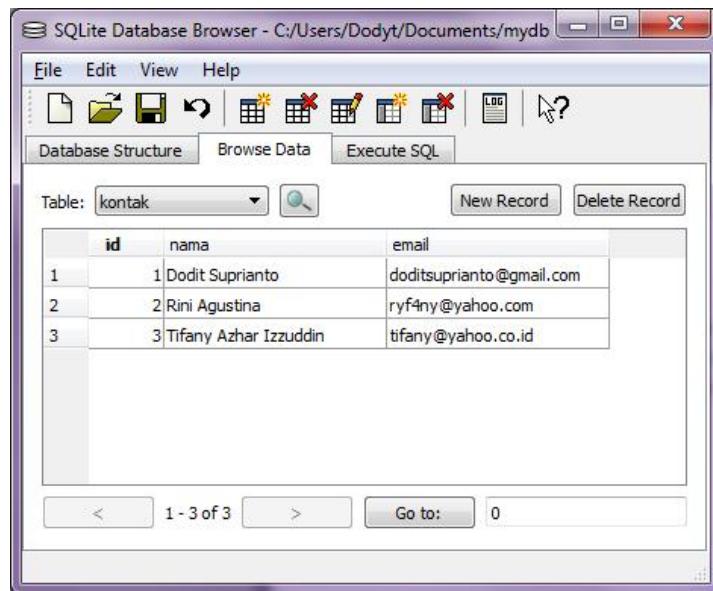
**Gambar 8.17 Struktur table yang telah terbentuk.**

Tambahkan perintah SQL INSERT untuk menambahkan data ke dalam table **kontak** pada tab **Execute SQL**, seperti berikut ini:

```
insert into kontak (nama, email) values ('Dodit Suprianto',  
'doditsuprianto@gmail.com');  
insert into kontak (nama, email) values ('Rini Agustina',  
'ryf4ny@yahoo.com');  
insert into kontak (nama, email) values ('Tifany Azhar  
Izzuddin', 'tiffany@yahoo.co.id');
```



**Gambar 8.18 Perintah SQL Insert**

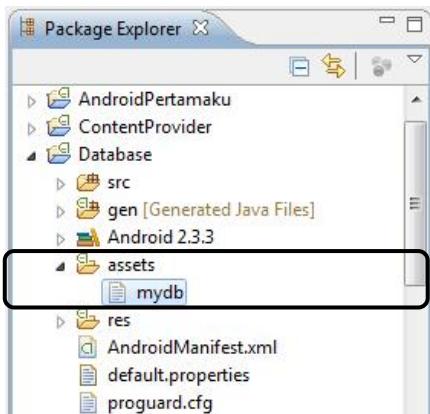


Gambar 8.19 Hasil penambahan baris ke table Kontak.

### 8.3.9 Menyatukan Database Ke Aplikasi

Dengan database yang diciptakan saat *design time*, maka langkah selanjutnya adalah menyatukan database tersebut bersama-sama dengan aplikasi.

Masih tetap menggunakan project yang sama dengan sebelumnya (project Database), *copy-paste* file database SQLite yang telah dibuat sebelumnya (file **mydb**) ke dalam folder **assets** di project Android.



**Gambar 8.20 Menggabungkan database ke project Android**

**Catatan:** Nama file yang ditambahkan ke folder assets harus berhuruf kecil. Nama file seperti **MyDB** adalah salah.

Selanjutnya modifikasi file `MainActivity.java` seperti berikut ini:

```
package net.belajar2android.Database;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import android.app.Activity;
import android.os.Bundle;
import android.widget.Toast;
import android.database.Cursor;

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

        try {
            String pathTujuan = "/data/data/" + getPackageName() +
"/databases/MyDB";
            File f = new File(pathTujuan);
```

```

        if (!f.exists()) {
            CopyDB(getContext().getAssets().open("mydb") , new
OutputStream(pathTujuan));
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    DBAdapter db = new DBAdapter(this);

    /**-mendapatkan seluruh kontak---
db.open();
Cursor c = db.getAllContacts();
if (c.moveToFirst()) {
    do {
        TampilkanKontak(c);
    } while (c.moveToNext());
}
db.close();

/*
//---menambah kontak---
//...
*/
    /**
//---mendapatkan seluruh kontak---
//...
*/
    /**
//---mendapatkan sebuah kontak---
//...
*/
    /**
//---update kontak---
//...
*/
    /**
//---menghapus kontak---
//...
*/
}

public void CopyDB(InputStream inputStream, OutputStream
outputStream)
throws IOException {
    //--copy 1K byte pada saat ini---
    byte[] buffer = new byte[1024];
    int length;
    while ((length = inputStream.read(buffer))>0) {

```

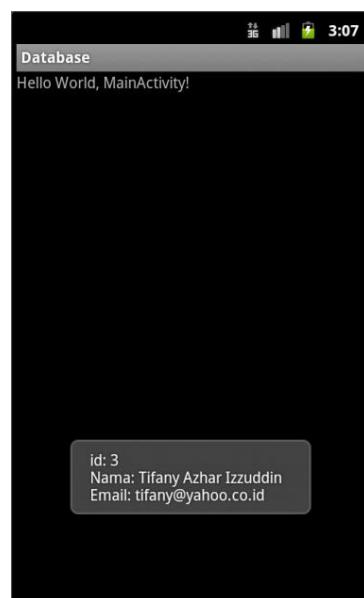
```

        outputStream.write(buffer, 0, length);
    }
    inputStream.close();
    outputStream.close();
}

public void TampilkanKontak(Cursor c)
{
    Toast.makeText(this, "id: " + c.getString(0) + "\n" + "Nama:
" + c.getString(1) + "\n" + "Email: " + c.getString(2),
Toast.LENGTH_LONG).show();
}
}

```

Jalankan aplikasi ke Emulator Android. Ketika aplikasi berjalan, aplikasi akan meng-copy file database **mydb** ke folder **/data/data/net.belajar2android.Database/databases/** dengan nama **MyDB**. Anda bisa melihat folder tersebut melalui DDMS.



*Gambar 8.21 Menampilkan isi table hasil peng-copy-an*

Bagaimana program bekerja? Pertama adalah pendefinisian method CopyDB() untuk meng-copy file database dari satu lokasi ke lokasi lainnya:

```
public void CopyDB(InputStream inputStream, OutputStream
outputStream)
throws IOException {
    //---copy 1K byte pada saat ini---
    byte[] buffer = new byte[1024];
    int length;
    while ((length = inputStream.read(buffer))>0) {
        outputStream.write(buffer, 0, length);
    }
}
```

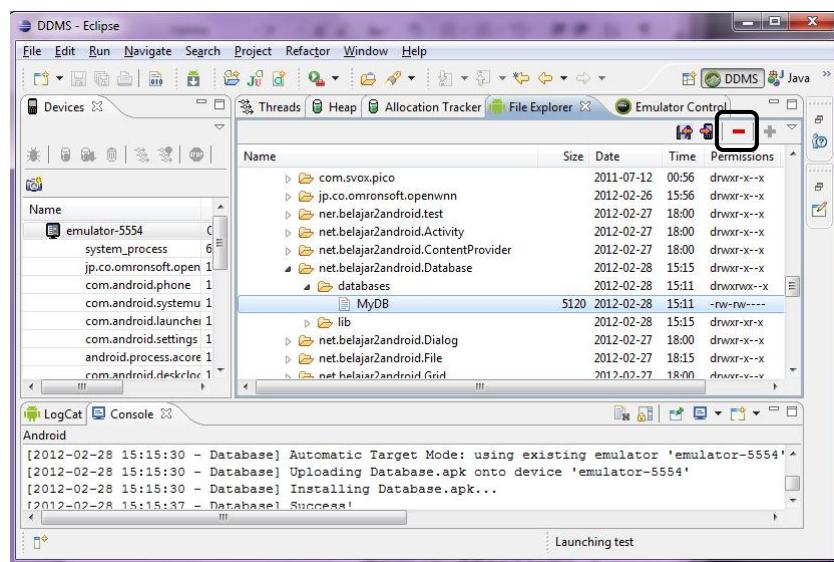
Disini digunakan object InputStream untuk membaca file sumber, kemudian menuliskannya ke file tujuan dengan menggunakan object OutputStream.

Ketika activity terbentuk, Anda meng-copy file database yang terletak di dalam folder assets ke folder /data/data/net.belajar2android.Database/databases/ pada perangkat Android/emulator:

```
try {
    String pathTujuan = "/data/data/" + getPackageName() +
"/databases/MyDB";
    File f = new File(pathTujuan);
    if (!f.exists()) {
        CopyDB(getApplicationContext().getAssets().open("mydb"), new
FileOutputStream(pathTujuan));
    }
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
```

Copy file database jika file belum tersedia di dalam folder tujuan. Jika tidak dilakukan pengecekan maka setiap kali activity terbentuk maka file database akan ditimpa dengan database yang berasal dari folder assets.

Untuk memastikan bahwa file database benar-benar ter-copy maka file database harus dihapus dari emulator (jika memang sudah terbentuk sebelumnya), hal ini hanya bersifat pengujian terhadap aplikasi saja. Anda juga dapat menghapusnya melalui DDMS.



**Gambar 8.22 Klik MyDB kemudian hapus (tombol bersimbol minus)  
setelah itu jalankan aplikasi ulang**

# BAB 9

# CONTENT PROVIDER

Bab ini akan dipelajari penggunaan content provider (built-in), juga penerapan content provider buatan sendiri untuk berbagi data secara lintas package.

## 9.1 Berbagi Data Dalam Android

Penggunaan content provider adalah cara yang disarankan untuk berbagi data lintas package. Bayangkan bahwa content provider adalah sebagai media penyimpan data. Bagaimana content provider menyimpan data tidaklah penting bagi aplikasi yang menggunakannya; yang terpenting adalah bagaimana package-package dapat mengakses data yang tersimpan di dalamnya dengan menggunakan program UI secara konsisten. Sebuah content provider berperilaku mirip dengan database - Anda dapat melakukan query terhadapnya, mengubah content, dan menghapus content. Perbedaannya, content provider menggunakan cara berbeda untuk menyimpan data. Data dapat disimpan di dalam database, file, bahkan melewati sebuah jaringan.

Android memiliki beberapa content provider, antara lain:

- **Browser** — Menyimpan data seperti bookmark browser, history browser, dan lain-lain.
- **CallLog** — Menyimpan data seperti missed calls, detail panggilan, dan lain-lain.
- **Contacts** — Menyimpan detail contact.
- **MediaStore** — Menyimpan file media seperti audio, video dan image.

- **Settings** — Menyimpan pengaturan perangkat (device setting) dan preference.

Disamping beberapa content provider built-in, Anda juga dapat menciptakan content provider sendiri.

Untuk melakukan query terhadap content provider, harus ditentukan query string pada form URI dengan beberapa pilihan segement.

Adapun formatnya adalah:

```
<standard_prefix>://<authority>/<data_path>/<id>
```

Bagian URI antara lain:

- **standard\_prefix**, untuk content provider selalu **content://**.
- **authority**, menentukan nama content provider. Contohnya adalah content provider **Contacts** (built-in content provider). Untuk content provider yang berasal dari pihak ketiga maka Authority bisa jadi **net.belajar2android.provider** atau **com.mediakom.provider**.
- **data\_path**, menentukan jenis data yang diminta. Contoh, jika hendak mendapatkan seluruh kontak dari content provider **Contacts**, kemudian data path yang diharapkan adalah **people** maka URI seharusnya seperti berikut: **content://contacts/people**.
- **id**, menentukan record tertentu yang diminta. Contoh, jika mencari kontak bernomor 2 di dalam content provider **Contacts** maka URI seharusnya seperti berikut: **content://contacts/people/2**.

Berikut ini contoh-contoh query string:

Query String	Penjelasan
content://media/internal/images	Mengembalikan daftar semua image yang tersimpan di dalam media

	penyimpanan internal
content://media/external/images	Mengembalikan daftar semua image yang tersimpan dalam media penyimpanan eksternal (SD card)
content://call_log/calls	Mengembalikan daftar semua panggilan/call yang terdaftar dalam Call Log
content://browser/bookmarks	Mengembalikan daftar bookmark yang tersimpan dalam browser

## 9.2 Penggunaan Content Provider

Ikuti instruksi di bawah ini untuk mengetahui bagaimana penggunaan content provider dari aplikasi Android Anda.

Buat project baru bernama **Provider**.

<b>Project name</b>	Provider
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	Provider
<b>Package name</b>	net.belajar2android.Provider
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```
<?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" >
<ListView
    android:id="@+id/android:list"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
```

```

        android:layout_weight="1"
        android:stackFromBottom="false"
        android:transcriptMode="normal" />
<TextView
        android:id="@+id/contactName"
        android:textStyle="bold"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
<Textview
        android:id="@+id/contactID"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>

```

Modifikasi file MainActivity.java seperti berikut ini:

```

package net.belajar2android.Provider;

import android.app.Activity;
import android.os.Bundle;
import android.app.ListActivity;
import android.database.Cursor;
import android.net.Uri;
import android.provider.ContactsContract;
import android.widget.SimpleCursorAdapter;

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

        Uri semuaKontak = Uri.parse("content://contacts/people");
        Cursor c = managedQuery(semuaKontak, null, null, null,
null);
        String[] kolom = new String[] {
            ContactsContract.Contacts.DISPLAY_NAME,
            ContactsContract.Contacts._ID
        };
        int[] views = new int[] {R.id.contactName, R.id.contactID};
        SimpleCursorAdapter adapter = new SimpleCursorAdapter(this,
R.layout.main, c, kolom, views);
        this.setListAdapter(adapter);
    }
}

```

Modifikasi file AndroidManifest.xml seperti berikut ini:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.belajar2android.Provider"
    android:versionCode="1"

```

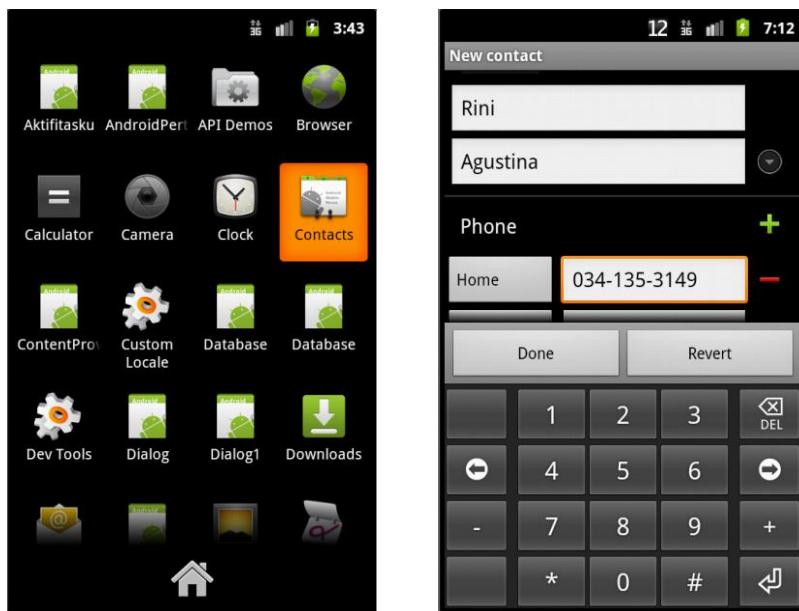
```

    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />
    <uses-permission
        android:name="android.permission.READ_CONTACTS">
    </uses-permission>

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

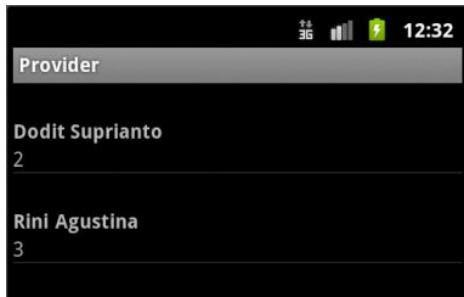
```

Jalankan aplikasi ke Emulator Android. Klik aplikasi **Contacts** dan tambahkan beberapa kontak pada aplikasi tersebut.



**Gambar 9.1 Menambah kontak baru pada aplikasi Contacts**

Jalankan aplikasi **Provider** ke Emulator Android. Activity akan menampilkan kontak yang baru saja dibuat, seperti berikut ini:



**Gambar 9.2 Daftar kontak dengan project Provider**

Bagaimana program bekerja? Pada contoh di atas, kita mencari semua kontak yang tersimpan di dalam aplikasi **Contacts** dan menampilkannya ke ListView.

Method `managedQuery()` dari class Activity mengambil sebuah *managed cursor*. *Managed cursor* menangani semua pekerjaan yang tidak bisa *loading* dengan sendirinya saat aplikasi berhenti sesaat dan melakukan *query* ulang ketika aplikasi berjalan kembali.

Pernyataan:

```
Cursor c = managedQuery(semuaKontak, null, null, null, null);
```

Sama dengan pernyataan berikut:

```
Cursor c = getContentResolver().query(semuaKontak, null, null,
null, null);
startManagingCursor(c);
//---membolehkan activity untuk mengatur siklus hidup cursor
berdasarkan
//siklus hidup activity ---
```

Method `getContentResolver()` mengembalikan sebuah object `ContentResolver` yang membantu menguraikan URI dengan content provider yang sesuai.

Object SimpleCursorAdapter memetakan sebuah cursor ke TextView (atau ImageView) yang telah ditentukan dalam file XML (main.xml). Object SimpleCursorAdapter memetakan data (diwakili oleh kolom) ke view-view (diwakili oleh **views**):

```
String[] kolom = new String[] {
    ContactsContract.Contacts.DISPLAY_NAME,
    ContactsContract.Contacts._ID
};
int[] views = new int[] {R.id.contactName, R.id.contactID};

SimpleCursorAdapter adapter = new SimpleCursorAdapter(this,
    R.layout.main, c, kolom, views);
this.setAdapter(adapter);
```

**Catatan:**

Agar aplikasi dapat mengakses aplikasi Contacts, maka perlu izin READ\_CONTACTS dalam file AndroidManifest.xml.

### 9.2.1 Konstanta Predefined Query String

Selaian menggunakan query URI, Anda dapat menggunakan daftar konstanta *Predefined Query String* untuk menentukan URI dengan tipe data yang berbeda. Contoh, query `content://contacts/people` bisa pula ditulis ulang menjadi:

```
Uri semuaKontak = Uri.parse("content://contacts/people");
```

Jika menggunakan konstanta Predefined akan menjadi:

```
Uri semuaKontak = ContactsContract.Contacts.CONTENT_URI;
```

**Catatan:**

Untuk Android 2.0 ke atas sudah menggunakan URI `ContactsContract.Contacts.CONTENT_URI` untuk query yang membutuhkan record Contacts.

Beberapa contoh konstanta predefined query string antara lain:

- Browser.BOOKMARKS\_URI
- Browser.SEARCHES\_URI
- CallLog.CONTENT\_URI
- MediaStore.Images.Media.INTERNAL\_CONTENT\_URI
- MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI
- Settings.CONTENT\_URI

Jika hendak menampilkan kontak pertama maka tentukan ID kontak tersebut, seperti berikut ini:

```
Uri satuKontak = Uri.parse("content://contacts/people/1");
```

Alternatif lain gunakan konstanta predefines bersama-sama dengan method **withAppendedId()** dari class **ContentUris**:

```
import android.content.ContentUris;
//...
Uri satuKontak = ContentUris.withAppendedId(
    ContactsContract.Contacts.CONTENT_URI, 1);
```

Disamping menampilkan ke ListView, Anda juga dapat mencetak hasilnya dengan menggunakan object Cursor, seperti berikut ini:

```
package net.belajar2android.Provider;

import android.app.Activity;
import android.os.Bundle;
import android.app.ListActivity;
import android.database.Cursor;
import android.net.Uri;
import android.provider.ContactsContract;
import android.widget.SimpleCursorAdapter;
import android.util.Log;

public class MainActivity extends ListActivity {
    /** Called when the activity is first created. */
    @Override
```

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    Uri semuaKontak = Uri.parse("content://contacts/people");
    Cursor c = managedQuery(semuaKontak, null, null, null,
    null);
    String[] kolom = new String[] {
        ContactsContract.Contacts.DISPLAY_NAME,
        ContactsContract.Contacts._ID
    };
    int[] views = new int[] {R.id.contactName, R.id.contactID};

    SimpleCursorAdapter adapter = new SimpleCursorAdapter(this,
    R.layout.main, c, kolom, views);
    this.setListAdapter(adapter);
    PrintContacts(c);
}

private void PrintContacts(Cursor c)
{
    if (c.moveToFirst()) {
        do {
            String contactID =
c.getString(c.getColumnIndex(ContactsContract.Contacts._ID));
            String contactDisplayName =
c.getString(c.getColumnIndex(ContactsContract.Contacts.DISPLAY_N
AME));
            Log.v("Content Providers", contactID + ", " +
contactDisplayName);
        } while (c.moveToNext());
    }
}

```

Method PrintContacts() akan mencetak hasilnya ke windows LogCat:

```

07-23 15:07:26.570: VERBOSE/Content Providers(336): 2, Dodit
Suprianto
07-23 15:07:26.580: VERBOSE/Content Providers(336): 3, Rini
Agustina

```

Ia akan mencetak ID dan nama setiap kontak yang tersimpan dalam aplikasi **Contacts** dengan cara mengakses field `ContactsContract.Contacts._ID` untuk memperoleh ID kontak dan `ContactsContract.Contacts.DISPLAY_NAME` untuk mendapatkan nama kontak. Jika ingin menampilkan nomor telpon dari kontak maka dibuat query content provider lagi.

```
package net.belajar2android.Provider;
```



```
        Log.v("Content Providers",
phoneCursor.getString(phoneCursor.getColumnIndex(ContactContract.CommonDataKinds.Phone.NUMBER)));
    }
    phoneCursor.close();
}
} while (c.moveToNext());
}
}
```

Untuk mengakses nomor telpon sebuah kontak perlu dibuat query pada URI yang tersimpan dalam `ContactsContract.CommonDataKinds.Phone.CONTENT_URI`.

Pada potongan kode sebelumnya dilakukan pengecekan kontak nomor telpon menggunakan field `ContactsContract.Contacts.HAS_PHONE_NUMBER`. Jika kontak memiliki lebih dari satu nomor telpon maka dibuat query content provider lagi berdasarkan ID kontak. Sekali nomor telpon ditemukan (bisa jadi berisi lebih dari satu nomor telpon) maka dilakukan perulangan terhadapnya dan mencetak beberapa nomor.

## 9.2.2 Projection

Parameter kedua method managedQuery() mengatur berapa jumlah kolom yang dikembalikan oleh query, parameter ini dikenal sebagai *projection*. Disini diset null:

```
Cursor c = managedQuery(semuaKontak, null, null, null, null);
```

Kita dapat menentukan secara pasti kolom-kolom yang dikembalikan melalui sebuah array yang berisi nama kolom yang dikembalikan.

```
String[] projection = new String[] {
    ContactsContract.Contacts._ID,
    ContactsContract.Contacts.DISPLAY_NAME,
    ContactsContract.Contacts.HAS_PHONE_NUMBER
};
Cursor c = managedQuery(semuaKontak, projection, null, null,
null);
```

Pada kasus di atas adalah field \_ID, DISPLAY\_NAME dan HAS\_PHONE\_NUMBER.

### 9.2.3 Penyaringan

Parameter ketiga dan keempat method managedQuery() mengizinkan kita untuk menetapkan klausa SQL WHERE untuk menyaring hasil query. Contoh, perintah untuk menampilkan orang dengan nama berawalan “Dod”:

```
Cursor c = managedQuery(semuaKontak, projection,
    ContactsContract.Contacts.DISPLAY_NAME + " LIKE 'Dod%'", null,
    null);
```

Parameter ketiga berisi pernyataan SQL untuk mencari nama berawalan “Dod”. Kita juga dapat meletakkan string pencarian di parameter keempat dari method, seperti ini:

```
Cursor c = managedQuery(allContacts, projection,
    ContactsContract.Contacts.DISPLAY_NAME + " LIKE ?", new String[]
    {"Dod%"}, null);
```

### 9.2.4 Pengurutan

Parameter kelima method managedQuery() membolehkan kita untuk menetapkan sebuah klausa SQL ORDER BY untuk mengurutkan hasil query. Contoh, pernyataan yang mengurutkan nama kontak dari kecil ke besar (ascending order):

```
Cursor c = managedQuery(allContacts, projection,
    ContactsContract.Contacts.DISPLAY_NAME + " LIKE ?", new String[]
    {"%"}, ContactsContract.Contacts.DISPLAY_NAME + " ASC");
```

untuk mengurutkan secara descending order (dari besar ke kecil), maka gunakan kata kunci DESC:

```
Cursor c = managedQuery(allContacts, projection,
    ContactsContract.Contacts.DISPLAY_NAME + " LIKE ?", new String[]
    {"%"}, ContactsContract.Contacts.DISPLAY_NAME + " DESC");
```

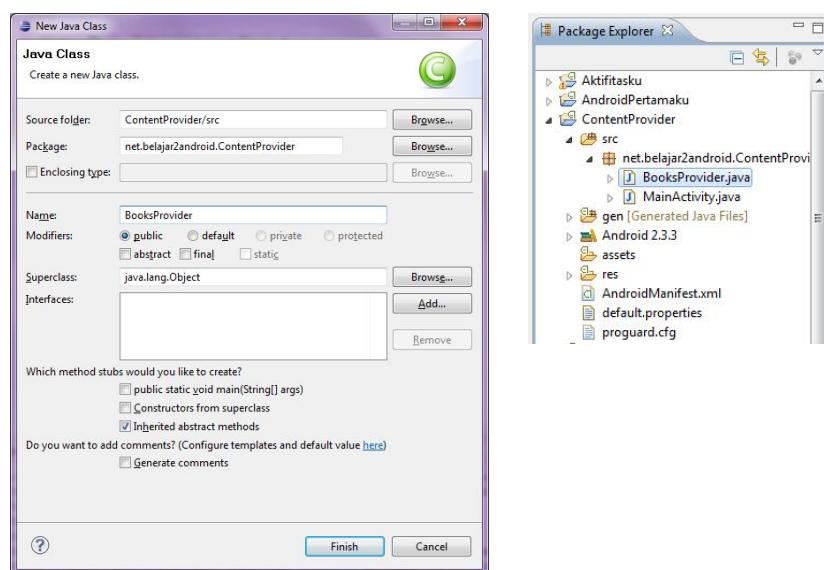
## 9.3 Membuat Content Provider Sendiri

Untuk menciptakan content provider Anda sendiri adalah dengan meng-extends class abstract `ContentProvider` dan meng-override beberapa method yang ada di dalamnya. Sebagai gambaran mudah adalah content provider untuk menyimpan buku dalam table database, dimana table berisi tiga field **yaitu \_id, title dan isbn**.

Buatlah project baru bernama **ContentProvider**.

<b>Project name</b>	ContentProvider
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	ContentProvider
<b>Package name</b>	net.belajar2android.ContentProvider
<b>Create Activity</b>	MainActivity

Dalam folder src, tambahkan file class Java dan beri nama **BooksProvider.java** (klik menu Eclipse File|New|Class):



**Gambar 9.3 Class BooksProvider.java**

Tambahkan pernyataan berikut ke dalam file **BooksProvider.java**:

```
package net.belajar2android.ContentProvider;

import android.content.ContentProvider;
import android.content.ContentUris;
import android.content.ContentValues;
import android.content.Context;
import android.content.UriMatcher;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteQueryBuilder;
import android.net.Uri;
import android.text.TextUtils;
import android.util.Log;

public class BooksProvider extends ContentProvider
{
    public static final String PROVIDER_NAME =
"net.belajar2android.provider.Books";
    public static final Uri CONTENT_URI = Uri.parse("content://" +
PROVIDER_NAME + "/books");
    public static final String _ID = "_id";
    public static final String TITLE = "title";
    public static final String ISBN = "isbn";
    private static final int BOOKS = 1;
    private static final int BOOK_ID = 2;
    private static final UriMatcher uriMatcher;

    static {
        uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
        uriMatcher.addURI(PROVIDER_NAME, "books", BOOKS);
        uriMatcher.addURI(PROVIDER_NAME, "books/#", BOOK_ID);
    }

    //---untuk penggunaan database---
    private SQLiteDatabase booksDB;
    private static final String DATABASE_NAME = "Books";
    private static final String DATABASE_TABLE = "titles";
    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_CREATE = "create table " +
DATABASE_TABLE + " (_id integer primary key autoincrement, " +
"title text not null, isbn text not null);";

    private static class DatabaseHelper extends SQLiteOpenHelper
    {
        DatabaseHelper(Context context) {
            super(context, DATABASE_NAME, null, DATABASE_VERSION);
        }
    }

    @Override
    public void onCreate(SQLiteDatabase db)
```

```

    {
        db.execSQL(DATABASE_CREATE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    Log.w("Content provider database", "Upgrade database dari
versi " + oldVersion + " ke " + newVersion + ", yang mana akan
menghapus seluruh data lama");
    db.execSQL("DROP TABLE IF EXISTS titles");
    onCreate(db);
}
}

@Override
public int delete(Uri arg0, String arg1, String[] arg2) {
    // arg0 = uri
    // arg1 = selection
    // arg2 = selectionArgs
    int count=0;
    switch (uriMatcher.match(arg0)){
    case BOOKS:
        count = booksDB.delete(DATABASE_TABLE, arg1, arg2);
        break;
    case BOOK_ID:
        String id = arg0.getPathSegments().get(1);
        count = booksDB.delete(DATABASE_TABLE, _ID + " = " + id +
(!TextUtils.isEmpty(arg1) ? " AND (" + arg1 + ')' : ""), arg2);
        break;
    default: throw new IllegalArgumentException("URI tidak
dikenal " + arg0);
    }
    getContext().getContentResolver().notifyChange(arg0, null);
    return count;
}

@Override
public String getType(Uri uri) {
    switch (uriMatcher.match(uri)){
    //---dapatkan semua buku---
    case BOOKS:
        return "vnd.android.cursor.dir/vnd.belajar2android.books";
    //---dapatkan buku tertentu---
    case BOOK_ID:
        return
    "vnd.android.cursor.item/vnd.belajar2android.books";
    default:
        throw new IllegalArgumentException("URI tidak didukung:
" + uri);
    }
}

@Override
public Uri insert(Uri uri, ContentValues values) {

```

```

    //---tambahkan buku baru---
    long rowID = booksDB.insert(DATABASE_TABLE, "", values);
    //---jika penambahan berhasil---
    if (rowID>0)
    {
        Uri _uri = ContentUris.withAppendedId(CONTENT_URI, rowID);
        getContext().getContentResolver().notifyChange(_uri,
null);
        return _uri;
    }
    throw new SQLException("Penyisipan baris gagal ke " + uri);
}

@Override
public boolean onCreate() {
    Context context = getContext();
    DatabaseHelper dbHelper = new DatabaseHelper(context);
    booksDB = dbHelper.getWritableDatabase();
    return (booksDB == null)? false:true;
}

@Override
public Cursor query(Uri uri, String[] projection, String
selection, String[] selectionArgs, String sortOrder) {
    SQLiteQueryBuilder sqlBuilder = new SQLiteQueryBuilder();
    sqlBuilder.setTables(DATABASE_TABLE);
    if (uriMatcher.match(uri) == BOOK_ID)
        //---jika mendapatkan buku tertentu---
        sqlBuilder.appendWhere("_ID + " = " +
uri.getPathSegments().get(1));
    if (sortOrder==null || sortOrder=="") sortOrder = TITLE;
    Cursor c = sqlBuilder.query(booksDB, projection, selection,
selectionArgs, null, null, sortOrder);

    //---register to watch a content URI for changes---
    c.setNotificationUri(getContext().getContentResolver(),
uri);
    return c;
}

@Override
public int update(Uri uri, ContentValues values, String
selection, String[] selectionArgs) {
    int count = 0;
    switch (uriMatcher.match(uri)){
    case BOOKS:
        count = booksDB.update(DATABASE_TABLE, values, selection,
selectionArgs);
        break;
    case BOOK_ID:
        count = booksDB.update(DATABASE_TABLE, values, "_ID + " =
+ uri.getPathSegments().get(1) + (!TextUtils.isEmpty(selection)
? " AND (" + selection + ')') : ""), selectionArgs);
        break;
    }
}

```

```

        default: throw new IllegalArgumentException("Unknown URI " +
uri);
    }

    getContext().getContentResolver().notifyChange(uri, null);
    return count;
}
}

```

Modifikasi file AndroidManifest.xml seperti berikut ini:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.belajar2android.ContentProvider"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />

    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".MainActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <provider android:name="BooksProvider"
            android:authorities="net.belajar2android.provider.Books"/>
    </application>
</manifest>

```

Bagaimana program bekerja? Pertama diciptakan sebuah class **BooksProvider** yang meng-extends class dasar **ContentProvider**.

Beberapa method yang meng-override class ini antara lain:

- **getType()** — Mengembalikan tipe MIME dari data yang diberikan oleh URI.
- **onCreate()** — Dipanggil ketika provider dimulai.
- **query()** — Menerima request dari client. Hasil dikembalikan sebagai object **Cursor**.
- **insert()** — Menambahkan record baru ke dalam content provider.

- **delete()** — Menghapus record yang ada di content provider.
- **update()** — Meng-update record yang ada di content provider.

Di dalam content provider kita bebas memilih bagaimana hendak menyimpan data (file sistem tradisional, XML, database atau bahkan web service). Pada contoh ini digunakan pendekatan database SQLite yang telah dibahas pada bab sebelumnya.

Kemudian didefinisikan konstanta dalam class BooksProvider:

```
public static final String PROVIDER_NAME =
"net.learn2develop.provider.Books";
public static final Uri CONTENT_URI =
Uri.parse("content://" + PROVIDER_NAME + "/books");

public static final String _ID = "_id";
public static final String TITLE = "title";
public static final String ISBN = "isbn";
private static final int BOOKS = 1;
private static final int BOOK_ID = 2;
private static final UriMatcher uriMatcher;

static {
    uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
    uriMatcher.addURI(PROVIDER_NAME, "books", BOOKS);
    uriMatcher.addURI(PROVIDER_NAME, "books/#", BOOK_ID);
}
```

Perhatikan kode di atas, kita menggunakan object UriMatcher untuk menguraikan isi URI yang dilewatkan ke content provider melalui ContentResolver. Contoh, berikut content URI yang mewakili sebuah request semua buku di dalam content provider:

content://net.belajar2android.Books/books

Berikut ini mewakili request buku tertentu dengan `_id = 5`:

content://net.belajar2android.provider.MailingList/books/5

Content provider menggunakan database SQLite untuk menyimpan buku-buku (diperlukan class helper SQLiteOpenHelper untuk membantu pengelolaan database):

```
private static class DatabaseHelper extends SQLiteOpenHelper
```

```

{
    DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db)
    {
        db.execSQL(DATABASE_CREATE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        Log.w("Content provider database", "Upgrade database dari
versi " +
                oldVersion + " ke " + newVersion + ", yang mana akan
menghapus seluruh data lama ");
        db.execSQL("DROP TABLE IF EXISTS titles");
        onCreate(db);
    }
}

```

Selanjutnya, meng-override method `getType()` yang secara unik menggambarkan tipe data content provider. Gunakan object UriMatcher yang mengembalikan “`vnd.android.cursor.item/vnd.learn2develop.books`” untuk satu buku dan “`vnd.android.cursor.dir/vnd.learn2develop.books`” untuk beberapa buku:

```

@Override
public String getType(Uri uri) {
    switch (uriMatcher.match(uri)){
        //---dapatkan semua buku---
        case BOOKS:
            return "vnd.android.cursor.dir/vnd.belajar2android.books ";
        //---dapatkan buku tertentu---
        case BOOK_ID:
            return "vnd.android.cursor.item/vnd. belajar2android.books
";
        default:
            throw new IllegalArgumentException("URI tidak didukung: " +
uri);
    }
}

```

Kemudian meng-override method `onCreate()` untuk membuka koneksi ke database ketika content provider dimulai:

```

@Override
public boolean onCreate() {
    Context context = getContext();
    DatabaseHelper dbHelper = new DatabaseHelper(context);
    booksDB = dbHelper.getWritableDatabase();
    return (booksDB == null)? false:true;
}

```

Selanjutnya meng-override method query() untuk mengizinkan client meng-query buku-buku:

```

@Override
public Cursor query(Uri uri, String[] projection, String
selection, String[] selectionArgs, String sortOrder) {
    SQLiteQueryBuilder sqlBuilder = new SQLiteQueryBuilder();
    sqlBuilder.setTables(DATABASE_TABLE);
    if (uriMatcher.match(uri) == BOOK_ID)
        //---if getting a particular book---
        sqlBuilder.appendWhere("_ID + " = " +
uri.getPathSegments().get(1));
    if (sortOrder==null || sortOrder=="") sortOrder = TITLE;
    Cursor c = sqlBuilder.query(booksDB, projection, selection,
selectionArgs, null, null, sortOrder);

    c.setNotificationUri(getContext().getContentResolver(), uri);
    return c;
}

```

Secara default hasil query diurutkan berdasarkan field title. Hasil query dikembalikan sebagai object Cursor.

Untuk mengizinkan sebuah buku baru disisipkan ke content provider, maka lakukan override terhadap method insert():

```

@Override
public Uri insert(Uri uri, ContentValues values) {
    //---tambahkan buku baru---
    long rowID = booksDB.insert(DATABASE_TABLE, "", values);
    //---jika penambahan berhasil---
    if (rowID>0)
    {
        Uri _uri = ContentUris.withAppendedId(CONTENT_URI, rowID);
        getContext().getContentResolver().notifyChange(_uri, null);
        return _uri;
    }
    throw new SQLException("Penyisipan baris gagal ke " + uri);
}

```

Sekali record sukses ditambahkan, panggil method `notifyChange()` dari `ContentResolver`, hal ini akan memberi catatan pengamatan bahwa baris/record telah di-update.

Untuk menghapus buku, lakukan override method `delete()`:

```
@Override  
public int delete(Uri arg0, String arg1, String[] arg2) {  
    // arg0 = uri  
    // arg1 = selection  
    // arg2 = selectionArgs  
    int count=0;  
    switch (uriMatcher.match(arg0)) {  
        case BOOKS:  
            count = booksDB.delete(DATABASE_TABLE, arg1, arg2);  
            break;  
        case BOOK_ID:  
            String id = arg0.getPathSegments().get(1);  
            count = booksDB.delete(DATABASE_TABLE, _ID + " = " + id +  
                (!TextUtils.isEmpty(arg1) ? " AND (" + arg1 + ')' : ""), arg2);  
            break;  
        default: throw new IllegalArgumentException("URI tidak dikenal  
" + arg0);  
    }  
    getContext().getContentResolver().notifyChange(arg0, null);  
    return count;  
}
```

Demikian pula, panggil method `notifyChange()` dari `ContentResolver` setelah penghapusan, hal ini akan memberi catatan pengamatan bahwa baris/record yang terhapus.

Terakhir, untuk meng-update buku, override method `update()`:

```
@Override  
public int update(Uri uri, ContentValues values, String  
selection, String[] selectionArgs) {  
    int count = 0;  
    switch (uriMatcher.match(uri)) {  
        case BOOKS:  
            count = booksDB.update(DATABASE_TABLE, values, selection,  
                selectionArgs);  
            break;  
        case BOOK_ID:  
            count = booksDB.update(DATABASE_TABLE, values, _ID + " =  
                " + uri.getPathSegments().get(1) +  
                (!TextUtils.isEmpty(selection) ? " AND (" + selection + ')' :  
                    ""), selectionArgs);  
            break;
```

```

        default: throw new IllegalArgumentException("Unknown URI " +
uri);
    }

    getContext().getContentResolver().notifyChange(uri, null);
    return count;
}

```

Seperti pada method insert() dan delete(), kita memanggil method notifyChange() dari ContentResolver setelah update.

Terakhir, untuk me-register content provider dengan Android, modifikasi file AndroidManifest.xml melalui penambahan elemen <provider>.

```
<provider android:name=".BooksProvider"
    android:authorities="net.belajar2android.provider.Books" />
```

## 9.4 Penggunaan Content

Sekarang Anda telah bisa membangun content provider Anda sendiri, Anda bisa pula mengujinya dari aplikasi Android. Ikuti langkah-langkah di bawah ini.

Masih tetap menggunakan project sebelumnya (ContentProvider) modifikasi pernyataan yang berada dalam main.xml seperti berikut ini:

```

<?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="ISBN" />
<EditText android:id="@+id/txtISBN"
    android:layout_height="wrap_content"
    android:layout_width="fill_parent" />
<TextView android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Judul" />
<EditText android:id="@+id/txtTitle"
    android:layout_height="wrap_content"
    android:layout_width="fill_parent" />
<Button android:text="Tambahkan Judul"
    android:id="@+id/btnAdd"
    android:layout_width="fill_parent"
```

```
    android:layout_height="wrap_content" />
<Button
    android:text="Mencari Judul"
    android:id="@+id	btnRetrieve"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
</LinearLayout>
```

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.ContentProvider;

import android.app.Activity;
import android.os.Bundle;

import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.content.ContentValues;
import android.database.Cursor;
import android.net.Uri;

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

        Button btnAdd = (Button) findViewById(R.id.btnAdd);
        btnAdd.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                //---menambahkan buku---
                ContentValues values = new ContentValues();
                values.put(BooksProvider.TITLE,
                        ((EditText)findViewById(R.id.txtTitle)).getText().toString());
                values.put(BooksProvider.ISBN,
                        ((EditText)findViewById(R.id.txtISBN)).getText().toString());
                Uri uri =
                getContentResolver().insert(BooksProvider.CONTENT_URI, values);
                Toast.makeText(getApplicationContext(),uri.toString(),
                Toast.LENGTH_LONG).show();
            }
        });

        Button btnRetrieve = (Button)
        findViewById(R.id.btnRetrieve);
        btnRetrieve.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                //---mencari judul-judul---
                Uri allTitles =
                Uri.parse("content://net.belajar2android.provider.Books/books");
```

```
        Cursor c = managedQuery(allTitles, null, null, null,
"title desc");
        if (c.moveToFirst()) {
            do{
                Log.v("ContentProviders",
c.getString(c.getColumnIndex(BooksProvider._ID)) + ", " +
c.getString(c.getColumnIndex(BooksProvider.TITLE)) + ", " +
c.getString(c.getColumnIndex(BooksProvider.ISBN)));
                } while (c.moveToNext());
            }
        });
    }
}
```

Jalankan aplikasi ke Emulator Android. Tuliskan ISBN dan Judul buku kemudian klik tombol “**Tambahkan Judul**”. Class Toast akan menampilkan URI buku yang ditambah ke content provider. Untuk menampilkan semua judul yang tersimpan dalam content provider, klik tombol “**Mencari Judul**” dan perhatikan nilai-nilai yang tercetak dalam window LogCat



#### **Gambar 9.4 ContentProvider**

Bagaimana program bekerja? Pertama, modifikasi activity sehingga user dapat menuliskan ISBN dan judul buku untuk ditambahkan ke content provider yang baru saja dibuat.

Untuk menambahkan buku ke content provider, diciptakan object ContentValues baru, kemudian mengisinya dengan beberapa informasi tentang buku bersangkutan:

```
//---menambahkan buku---
ContentValues values = new ContentValues();
values.put(BooksProvider.TITLE,
((EditText) findViewById(R.id.txtTitle)).getText().toString());

values.put(BooksProvider.ISBN,
((EditText) findViewById(R.id.txtISBN)).getText().toString());

Uri uri = getContentResolver().insert(BooksProvider.CONTENT_URI,
values);
```

Karena content provider masih dalam package yang sama, maka gunakan konstanta BooksProvider.TITLE dan BooksProvider.ISBN untuk merujuk ke field “title” dan “isbn”.

Jika mengakses content provider ini dari package lainnya, Anda tidak bisa menggunakan konstanta tersebut, perlu ditentukan nama field secara langsung, seperti berikut ini:

```
ContentValues values = new ContentValues();
values.put("title",
((EditText) findViewById(R.id.txtTitle)).getText().toString());

values.put("isbn",
((EditText) findViewById(R.id.txtISBN)).getText().toString());

Uri uri =
getContentResolver().insert(Uri.parse("content://net.belajar2and
roid.provider.Books/books"),values);

Toast.makeText(getApplicationContext(),uri.toString(),
Toast.LENGTH_LONG).show();
```

Perlu diperhatikan bahwa untuk package eksternal perlu merujuk ke content URI dengan menggunakan content URI yang memenuhi syarat penuh, seperti berikut ini:

```
Uri.parse("content://net.learn2develop.provider.Books/books");
```

Untuk menampilkan seluruh judul yang terdapat dalam content provider, gunakan potongan kode berikut ini:

```
Uri allTitles =  
Uri.parse("content://net.belajar2android.provider.Books/books");  
Cursor c = managedQuery(allTitles, null, null, null, "title  
desc");  
if (c.moveToFirst()) {  
    do{  
        Log.v("ContentProviders",  
c.getString(c.getColumnIndex(BooksProvider._ID)) + ", " +  
c.getString(c.getColumnIndex(BooksProvider.TITLE)) + ", " +  
c.getString(c.getColumnIndex(BooksProvider.ISBN)));  
    } while (c.moveToNext());  
}
```

Pada query sebelumnya akan mengembalikan hasil yang terurut *descending* berdasarkan field title. Jika ingin meng-update detil buku, panggil method update() dengan content URI yang menandakan ID buku:

```
ContentValues editedValues = new ContentValues();  
editedValues.put(BooksProvider.TITLE, "Tip & Trik Pemrograman  
C#");  
  
getContentResolver().update(Uri.parse("content://net.belajar2and  
roid.provider.Books/books/2"), editedValues, null, null);
```

Untuk menghapus sebuah buku, gunakan method delete() dengan content URI yang menandakan ID buku:

```
getContentResolver().delete(  
Uri.parse("content://net.belajar2android.provider.Books/books/2"  
, null, null);
```

Untuk menghapus keseluruhan buku, kemudian hilangkan ID buku dari content URI:

```
getContentResolver().delete(Uri.parse("content://net.  
belajar2android.provider.Books/books"), null, null);
```

# **BAB 10**

# **PESAN DAN JARINGAN**

Bab ini akan dipelajari bagaimana mengirim pesan SMS, menerima pesan SMS secara program dan bagaimana menggunakan protocol HTTP untuk berkomunikasi dengan web server sehingga dapat mendownload teks dan data biner.

## **10.1 Pesan SMS**

Pesan SMS adalah salah satu aplikasi utama pada telpon bergerak saat ini, dimana Android datang dilengkapi dengan aplikasi SMS secara built-in untuk mengirim dan menerima pesan SMS. Namun, pada beberapa kasus mungkin ingin memadukan kemampuan SMS ke dalam aplikasi Android Anda sendiri. Contoh, bila hendak melacak lokasi anak Anda – sederhana saja beri dia perangkat Android yang bisa mengirim keluar pesan SMS yang berisi lokasi geografis setiap 30 menit secara otomatis tanpa disadari oleh si anak. Sekarang Anda bisa mengetahui jika mereka benar-benar pergi ke perpustakaan setelah sekolah atau tidak (tentu saja, ini berarti Anda harus membayar biaya lebih untuk pengiriman pesan SMS yang dikirim)

### **10.1.1 Mengirim Pesan SMS Secara Program**

Berikut ini contoh aplikasi yang mengirim pesan SMS secara otomatis tanpa campur tangan user.

Buatlah project baru bernama **SMS**.

<b>Project name</b>	SMS
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	SMS

<b>Package name</b>	net.belajar2android.SMS
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```
<?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:id="@+id/btnSendSMS"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Send SMS" />
</LinearLayout>
```

Modifikasi file AndroidManifest.xml seperti berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
        package="net.belajar2android.SMS"
        android:versionCode="1"
        android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />
    <uses-permission
        android:name="android.permission.SEND_SMS"></uses-permission>

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

Modifikasi file MainActivity.java:

```
package net.belajar2android.SMS;

import android.app.Activity;
import android.os.Bundle;
import android.app.PendingIntent;
import android.content.Intent;
```

```

import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.IntentFilter;

public class MainActivity extends Activity {
    Button btnSendSMS;

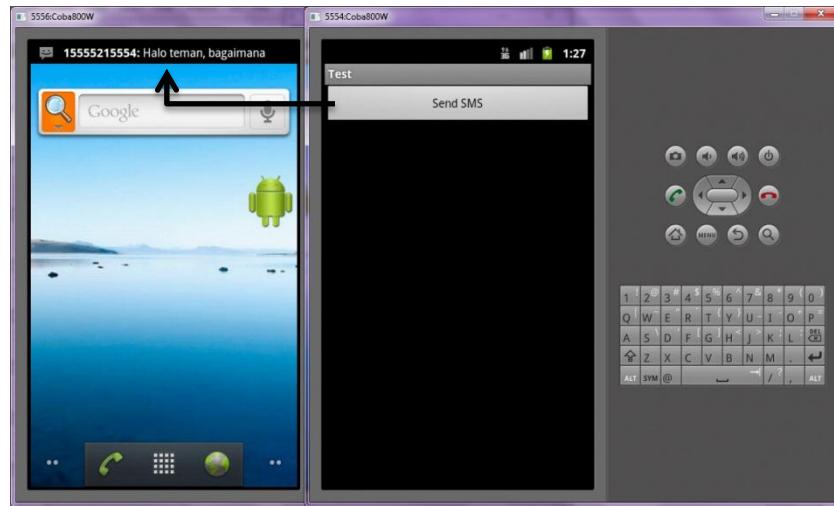
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        btnSendSMS = (Button) findViewById(R.id.btnSendSMS);
        btnSendSMS.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v)
            {
                sendSMS("5556", "Halo teman, bagaimana kabarnya?");
            }
        });
    }

    //---mengirim sebuah pesan SMS ke perangkat lain---
    private void sendSMS(String phoneNumber, String message)
    {
        SmsManager sms = SmsManager.getDefault();
        sms.sendTextMessage(phoneNumber, null, message, null, null);
    }
}

```

Sekarang buka lagi Emulator Android (AVD) baru sehingga terdapat dua emulator (contoh: beridentitas 5554 dan 5556), kemudian jalankan aplikasi ke Emulator. Pada Emulator Android pertama klik tombol “**Send SMS**” untuk mengirim sebuah pesan SMS ke emulator kedua.



Gambar 10.1 Notifikasi pesan SMS muncul pada Emulator lainnya

Bagaimana program bekerja? Untuk mengirim pesan SMS secara program digunakan class **SmsManager**. Tidak seperti class lain, kita tidak dapat secara langsung meng-instansiasi class ini; oleh karena itu perlu dipanggil method static `getDefault()` untuk memperoleh object **SmsManager**. Kemudian gunakan method `sendTextMessage()` untuk mengirim pesan SMS:

```
private void sendSMS(String phoneNumber, String message)
{
    SmsManager sms = SmsManager.getDefault();
    sms.sendTextMessage(phoneNumber, null, message, null, null);
}
```

Terdapat lima argument pada method `sendTextMessage()`, antara lain:

- **destinationAddress** – Nomor telpon penerima.
- **scAddress** – Alamat Service Center; gunakan null untuk default SMSC.
- **text** – Isi pesan SMS.
- **sentIntent** – Pending intent ketika pesan terkirim.

- **deliveryIntent** – Pending intent ketika pesan diterima.

### 10.1.2 Mendapatkan Umpan Balik Pengiriman Pesan

Bagaimana cara mengetahui bahwa pesan sms yang terkirim sudah benar? Untuk mengetahuinya, Anda dapat menciptakan dua object PendingIntent untuk memantau status proses pengiriman pesan SMS. Dua object PendingIntent dilewatkan ke dua argument terakhir dari method sendTextMessage().

```
private void sendSMS(String phoneNumber, String message)
{
    String SENT = "SMS_SENT";
    String DELIVERED = "SMS_DELIVERED";
    PendingIntent sentPI = PendingIntent.getBroadcast(this, 0, new Intent(SENT), 0);
    PendingIntent deliveredPI = PendingIntent.getBroadcast(this,
0, new Intent(DELIVERED), 0);

    //---ketika SMS terkirim---
    registerReceiver(new BroadcastReceiver() {
        @Override
        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode())
            {
                case Activity.RESULT_OK:
                    Toast.makeText(getApplicationContext(), "SMS sent",
Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                    Toast.makeText(getApplicationContext(), "Generic failure",
Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_NO_SERVICE:
                    Toast.makeText(getApplicationContext(), "No service",
Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_NULL_PDU:
                    Toast.makeText(getApplicationContext(), "Null PDU",
Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_RADIO_OFF:
                    Toast.makeText(getApplicationContext(), "Radio off",
Toast.LENGTH_SHORT).show();
                    break;
            }
        }
    }, new IntentFilter(SENT));
}
```

```

//---ketika SMS sampai tujuan---
registerReceiver(new BroadcastReceiver() {
    @Override
    public void onReceive(Context arg0, Intent arg1) {
        switch (getResultCode())
        {
            case Activity.RESULT_OK:
                Toast.makeText(getApplicationContext(), "SMS diterima",
Toast.LENGTH_SHORT).show();
                break;
            case Activity.RESULT_CANCELED:
                Toast.makeText(getApplicationContext(), "SMS tidak
diterima", Toast.LENGTH_SHORT).show();
                break;
        }
    }
}, new IntentFilter(DELIVERED));

SmsManager sms = SmsManager.getDefault();
sms.sendTextMessage(phoneNumber, null, message, sentPI,
deliveredPI);
}

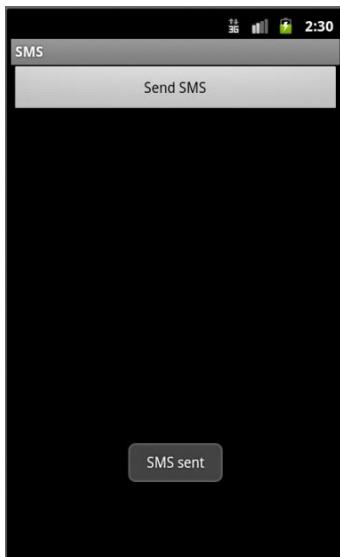
```

Disini diciptakan dua object PendingIntent. Kemudian me-register ke dua BroadcastReceiver. Dua BroadcastReceivers mendengar intent yang sesuai “SMS\_SENT” dan “SMS\_DELIVERED” (yang dijalankan oleh system operasi ketika pesan telah terkirim dan diterima secara berturut-turut). Dalam setiap BroadcastReceiver Anda meng-override method onReceive() dan mendapatkan hasil kode saat ini.

Dua object PendingIntent dilewatkan ke dua argument terakhir dari method sendTextMessage():

```
sms.sendTextMessage(phoneNumber, null, message, sentPI,
deliveredPI);
```

Dalam kasus ini, apapun pesan yang terkirim baik benar atau gagal diterima, akan diberi catatan status melalui dua object PendingIntent.



*Gambar 10.2 Pesan SMS terkirim benar*

### **10.1.3 Mengirim Pesan SMS Dengan Intent**

Dengan class SmsManager maka dapat dikirim pesan SMS dari aplikasi tanpa perlu melibatkan aplikasi built-in Messaging. Namun akan lebih mudah bila tetap melibatkan aplikasi built-in Messaging dan membiarkannya melakukan semua pekerjaan mengirim pesan untuk Anda.

Untuk mengaktifkan aplikasi built-in Messaging dari aplikasi dapat digunakan object Intent bersama-sama dengan MIME type “vnd.android-dir/mms-sms”.

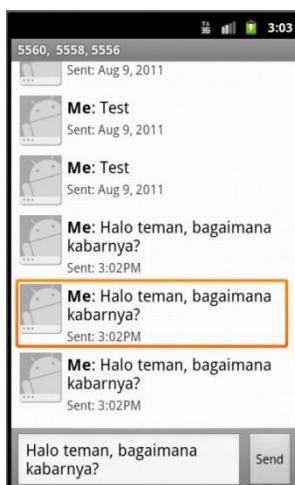
```
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    btnSendSMS = (Button) findViewById(R.id.btnSendSMS);
    btnSendSMS.setOnClickListener(new View.OnClickListener()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

```

        //sendSMS("5556", "Hello my friends!");
        Intent i = new Intent(android.content.Intent.ACTION_VIEW);
        i.putExtra("address", "5556; 5558; 5560");
        i.putExtra("sms_body", "Halo teman, bagaimana kabarnya?");
        i.setType("vnd.android-dir/mms-sms");
        startActivity(i);
    }
}
);
}
}

```

Kita bisa mengirim SMS lebih dari satu penerima dengan mudah, yaitu memisahkan setiap nomor telpon dengan tanda titik koma (di dalam method putExtra()).



*Gambar 10.3 Mengirim pesan melalui Messaging*

#### 10.1.4 Menerima Pesan SMS

Disamping mengirim pesan SMS dari aplikasi Android, kita juga bisa menerima pesan SMS yang masuk ke aplikasi melalui penggunaan object BroadcastReceiver. Ini berguna ketika kita ingin agar aplikasi melakukan suatu tindakan tertentu ketika pesan SMS khusus diterima. Contoh, untuk melacak lokasi perangkat telpon yang dicuri. Dalam kasus ini kita dapat membuat aplikasi yang secara otomatis mendengarkan(listen) pesan SMS, dimana pesan SMS yang masuk berisi beberapa kode

rahasia. Sekali pesan diterima, maka kita dapat mengirim balik sebuah pesan SMS yang berisi koordinat lokasi ke pengirimnya

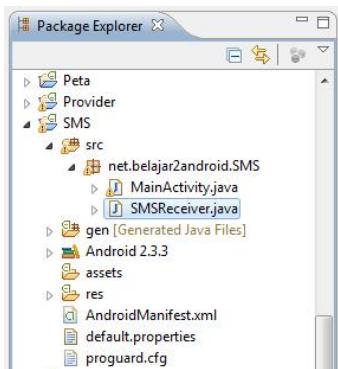
Ikuti langkah berikut ini yang menunjukkan bagaimana mendengarkan pesan SMS yang datang.

Masih tetap menggunakan project sebelumnya (project SMS), modifikasi file AndroidManifest.xml seperti berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
        package="net.belajar2android.SMS"
        android:versionCode="1"
        android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />
    <uses-permission
        android:name="android.permission.SEND_SMS"></uses-permission>
    <uses-permission
        android:name="android.permission.RECEIVE_SMS"></uses-permission>
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".MainActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <receiver android:name=".SMSReceiver">
            <intent-filter>
                <action
                    android:name="android.provider.Telephony.SMS_RECEIVED" />
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

Di dalam folder src project, tambahkan file Class baru ke package net.belajar2android.SMS dan beri nama class tersebut dengan **SMSReceiver.java**. Klik kanan folder project **SMS | src | net.belajar2android.SMS** kemudian pilih menu **New | Class**.



Gambar 10.4 Menambahkan class SMSReceiver.java

Modifikasi file class SMSReceiver.java menjadi seperti ini:

```
package net.belajar2android.SMS;

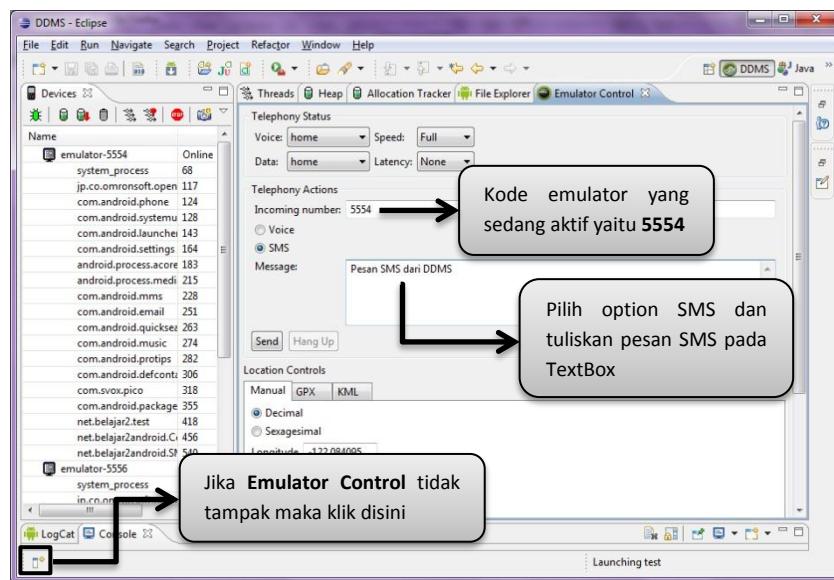
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.widget.Toast;

public class SMSReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        //---mendapatkan pesan SMS yang dilewatkan (pass in)---
        Bundle bundle = intent.getExtras();
        SmsMessage[] msgs = null;

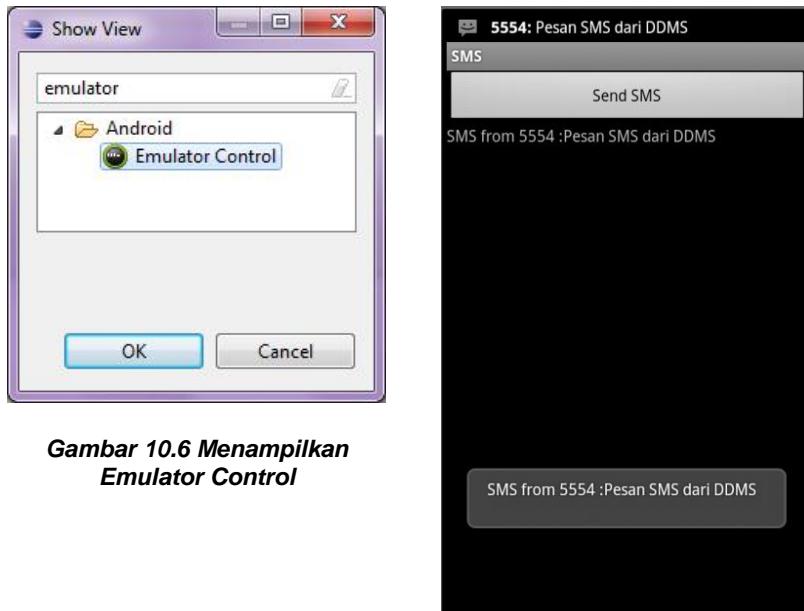
        String str = "";
        if (bundle != null)
        {
            //---Menampilkan pesan SMS yang diterima---
            Object[] pdus = (Object[]) bundle.get("pdus");
            msgs = new SmsMessage[pdus.length];
            for (int i=0; i<msgs.length; i++) {
                msgs[i] = SmsMessage.createFromPdu((byte[])pdus[i]);
                str += "SMS from " + msgs[i].getOriginatingAddress();
                str += " :";
                str += msgs[i].getMessageBody().toString();
                str += "\n";
            }
            //---menampilkan pesan SMS baru---
            Toast.makeText(context, str, Toast.LENGTH_SHORT).show();
        }
    }
}
```

```
}
```

Jalankan aplikasi ke Emulator Android. Gunakan DDMS, kirim sebuah pesan ke emulator. Sekarang aplikasi seharusnya bisa menerima pesan dan menampikannya ke class Toast.



**Gambar 10.5 Mengirim pesan melalui DDMS**



**Gambar 10.6 Menampilkan Emulator Control**

**Gambar 10.7 Menerima pesan dari DDMS**

Bagaimana program bekerja? Untuk mendengar pesan SMS yang datang, diciptakan class BroadcastReceiver. Class BroadcastReceiver mengizinkan aplikasi menerima intent-intent yang dikirim oleh aplikasi lain dengan menggunakan method sendBroadcast(). Pada dasarnya, dengan BroadcastReceiver aplikasi dapat menangani event-event yang muncul dari aplikasi lain. Ketika sebuah intent diterima, method onReceive() dipanggil, selanjutnya meng-override method onReceive().

Ketika sebuah pesan SMS diterima, method onReceive() dijalankan. Pesan SMS dilanjutkan ke dalam object Intent (intent, parameter kedua dalam method onReceive()) melalui object Bundle. Pesan-pesan disimpan dalam array Object dalam format PDU. Untuk menguraikan setiap pesan, gunakan method static createFromPdu() dari class SmsMessage. Pesan SMS kemudian ditampilkan melalui class Toast.

Nomor telpon pengirim diperoleh melalui method `getOriginatingAddress()`. Jika perlu mengirim sebuah autoreply ke pengirim, maka gunakan method ini untuk mengetahui nomor telpon pengirim.

Salah satu karakteristik yang menarik dari `BroadcastReceiver` adalah secara terus menerus `BroadcastReceiver` mendengarkan pesan SMS yang datang, bahkan jika aplikasi tidak berjalan sekalipun. Selama aplikasi ter-install pada perangkat maka apapun pesan SMS yang datang akan diterima oleh aplikasi.

### 10.1.5 Update Activity Dari `BroadcastReceiver`

Pada bagian sebelumnya menggambarkan bagaimana class `BroadcastReceiver` mendengarkan pesan SMS yang datang, kemudian class `Toast` menampilkan pesan SMS yang diterima. Namun mungkin Anda ingin mengirim pesan SMS balik ke main activity dari aplikasi. Contoh, menampilkan pesan ke dalam `TextView`.

Masih tetap menggunakan project sebelumnya (project SMS), modifikasi file `main.xml` seperti berikut ini:

```
<?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:id="@+id/btnSendSMS"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Send SMS" />
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

Modifikasi file `SMSReceiver.java` seperti berikut ini:

```

package net.belajar2android.SMS;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.widget.Toast;

public class SMSReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        //---mendapatkan pesan SMS yang dilewatkan (pass in)---
        Bundle bundle = intent.getExtras();
        SmsMessage[] msgs = null;

        String str = "";
        if (bundle != null)
        {
            //---Menampilkan pesan SMS yang diterima---
            Object[] pdus = (Object[]) bundle.get("pdus");
            msgs = new SmsMessage[pdus.length];
            for (int i=0; i<msgs.length; i++) {
                msgs[i] = SmsMessage.createFromPdu((byte[])pdus[i]);
                str += "SMS from " + msgs[i].getOriginatingAddress();
                str += " :";
                str += msgs[i].getMessageBody().toString();
                str += "\n";
            }
            //---menampilkan pesan SMS baru---
            Toast.makeText(context, str, Toast.LENGTH_SHORT).show();

            //---mengirim broadcast intent untuk meng-update SMS yang
            //diterima activity---
            Intent broadcastIntent = new Intent();
            broadcastIntent.setAction("SMS_RECEIVED_ACTION");
            broadcastIntent.putExtra("sms", str);
            context.sendBroadcast(broadcastIntent);
        }
    }
}

```

Modifikasi file MainActivity.java seperti berikut ini:

```

package net.belajar2android.SMS;

import android.app.Activity;
import android.os.Bundle;
import android.app.PendingIntent;
import android.content.Intent;
import android.telephony.SmsManager;
import android.view.View;

```

```
import android.widget.Button;
import android.widget.Toast;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.IntentFilter;
import android.widget.TextView;

public class MainActivity extends Activity {
    Button btnSendsSMS;
    IntentFilter intentFilter;

    private BroadcastReceiver intentReceiver = new
BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            //---menampilkan SMS yang diterima ole TextView---
            TextView SMSes = (TextView) findViewById(R.id.textView1);
            SMSes.setText(intent.getExtras().getString("sms"));
        }
    };

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

        //---intent untuk menyarings pesan SMS yang diterima---
        intentFilter = new IntentFilter();
        intentFilter.addAction("SMS_RECEIVED_ACTION");

        btnSendSMS = (Button) findViewById(R.id.btnSendSMS);
        btnSendSMS.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v)
            {
                //sendSMS("5556", "Halo teman, bagaimana kabarnya?");
                Intent i = new
Intent(android.content.Intent.ACTION_VIEW);
                i.putExtra("address", "5556; 5558; 5560");
                i.putExtra("sms_body", "Halo teman, bagaimana
kabarnya?");
                i.setType("vnd.android-dir/mms-sms");
                startActivity(i);
            }
        });
    }

    @Override
    protected void onResume() {
        //---register penerima---
        registerReceiver(intentReceiver, intentFilter);
        super.onResume();
    }
}
```

```

@Override
protected void onPause() {
    //---unregister penerima---
    unregisterReceiver(intentReceiver);
    super.onPause();
}

//---mengirim sebuah pesan SMS ke perangkat lain---
private void sendSMS(String phoneNumber, String message)
{
    String SENT = "SMS_SENT";
    String DELIVERED = "SMS_DELIVERED";
    PendingIntent sentPI = PendingIntent.getBroadcast(this, 0,
new Intent(SENT), 0);
    PendingIntent deliveredPI = PendingIntent.getBroadcast(this,
0, new Intent(DELIVERED), 0);

    //---ketika SMS terkirim---
    registerReceiver(new BroadcastReceiver(){
        @Override
        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode())
            {
                case Activity.RESULT_OK:
                    Toast.makeText(getApplicationContext(), "SMS sent",
Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                    Toast.makeText(getApplicationContext(), "Generic failure",
Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_NO_SERVICE:
                    Toast.makeText(getApplicationContext(), "No service",
Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_NULL_PDU:
                    Toast.makeText(getApplicationContext(), "Null PDU",
Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_RADIO_OFF:
                    Toast.makeText(getApplicationContext(), "Radio off",
Toast.LENGTH_SHORT).show();
                    break;
            }
        }
    }, new IntentFilter(SENT));

    //---ketika SMS sampai tujuan---
    registerReceiver(new BroadcastReceiver(){
        @Override
        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode())
            {
                case Activity.RESULT_OK:

```

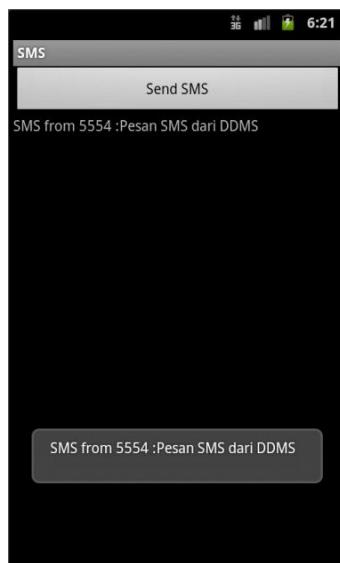
```

        Toast.makeText(getApplicationContext(), "SMS diterima",
        Toast.LENGTH_SHORT).show();
        break;
    case Activity.RESULT_CANCELED:
        Toast.makeText(getApplicationContext(), "SMS tidak
diterima", Toast.LENGTH_SHORT).show();
        break;
    }
}
}, new IntentFilter(DELIVERED));

SmsManager sms = SmsManager.getDefault();
sms.sendTextMessage(phoneNumber, null, message, sentPI,
deliveredPI);
}
}

```

Jalankan aplikasi ke Emulator Android. Maka hasilnya kurang lebih tampak seperti di bawah ini



**Gambar 10.8 Update activity**

Bagaimana program bekerja? Pertama ditambahkan TextView pada activity sehingga dapat digunakan untuk menampilkan pesan SMS yang diterima.

Kemudian, memodifikasi class SMSReceiver sehingga ketika class SMSReceiver menerima pesan SMS, pesan SMS akan menyebar ke object Intent lainnya, aplikasi lain yang mendengar intent tersebut dapat dinotifikasi. SMS yang diterima juga mengirim keluar melalui intent ini:

```
//---mengirim broadcast intent untuk meng-update SMS yang  
diterima activity---  
Intent broadcastIntent = new Intent();  
broadcastIntent.setAction("SMS_RECEIVED_ACTION");  
broadcastIntent.putExtra("sms", str);  
context.sendBroadcast(broadcastIntent);
```

Selanjutnya, di dalam activity diciptakan object BroadcastReceiver untuk mendengar intent-intent broadcast:

```
private BroadcastReceiver intentReceiver = new  
BroadcastReceiver() {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        //---menampilkan SMS yang diterima oleh TextView---  
        TextView SMSes = (TextView) findViewById(R.id.textView1);  
        SMSes.setText(intent.getExtras().getString("sms"));  
    }  
};
```

Ketika intent broadcast diterima, Anda meng-update pesan SMS di dalam TextView, Anda perlu membuat sebuah object IntentFilter sehingga dapat mendengarkan intent tertentu. Dalam kasus ini adalah intent "SMS\_RECEIVED\_ACTION":

```
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
  
    //---intent untuk menyaring pesan SMS yang diterima---  
    intentFilter = new IntentFilter();  
    intentFilter.addAction("SMS_RECEIVED_ACTION");  
    //..  
}
```

Terakhir, Anda me-register BroadcastReceiver dalam event onResume() dari activity dan me-unregister-nya dalam event onPause():

```
@Override  
protected void onResume() {
```

```

    //---register penerima---
    registerReceiver(intentReceiver, intentFilter);
    super.onResume();
}

@Override
protected void onPause() {
    //---unregister penerima---
    unregisterReceiver(intentReceiver);
    super.onPause();
}

```

Ini berarti bahwa TextView akan menampilkan pesan SMS hanya ketika pesan diterima saat activity tampak dilayar. Jika pesan SMS diterima ketika activity tidak tampak dimuka maka TextView tidak akan diperbarui.

### 10.1.6 Menjalankan Activity Dari BroadcastReceiver

Di bagian sebelumnya ditunjukkan bagaimana melewatkkan pesan SMS yang diterima untuk ditampilkan pada activity. Pada beberapa situasi mungkin activity masih berada di *background* ketika pesan SMS diterima. Dalam kasus ini, akan sangat berguna bila membawa activity ke *foreground* ketika pesan diterima.

Gunakan project sebelumnya (project SMS), modifikasi file MainActivity.java seperti berikut ini:

```

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

    //---intent untuk menyaring pesan SMS yang diterima---
    intentFilter = new IntentFilter();
    intentFilter.addAction("SMS_RECEIVED_ACTION");

    //---register penerima---
    registerReceiver(intentReceiver, intentFilter);

    btnSendSMS = (Button) findViewById(R.id.btnSendSMS);
    btnSendSMS.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            //sendSMS("5556", "Halo teman, bagaimana kabarnya?");
        }
    });
}

```

```

        Intent i = new Intent(android.content.Intent.ACTION_VIEW);
        i.putExtra("address", "5556; 5558; 5560");
        i.putExtra("sms_body", "Halo teman, bagaimana kabarnya?");
        i.setType("vnd.android-dir/mms-sms");
        startActivity(i);
    }
});
}
}

@Override
protected void onResume() {
    //---register penerima---
    //registerReceiver(intentReceiver, intentFilter);
    super.onResume();
}

@Override
protected void onPause() {
    //---unregister penerima---
    //unregisterReceiver(intentReceiver);
    super.onPause();
}

@Override
protected void onDestroy() {
    //---unregister penerima---
    unregisterReceiver(intentReceiver);
    super.onDestroy();
}

```

Modifikasi file SMSReceiver.java seperti berikut ini:

```

@Override
public void onReceive(Context context, Intent intent)
{
    //---mendapatkan pesan SMS yang dilewatkan (pass in)---
    Bundle bundle = intent.getExtras();
    SmsMessage[] msgs = null;

    String str = "";
    if (bundle != null)
    {
        //---Menampilkan pesan SMS yang diterima---
        Object[] pdus = (Object[]) bundle.get("pdus");
        msgs = new SmsMessage[pdus.length];
        for (int i=0; i<msgs.length; i++) {
            msgs[i] = SmsMessage.createFromPdu((byte[])pdus[i]);
            str += "SMS from " + msgs[i].getOriginatingAddress();
            str += " :";
            str += msgs[i].getMessageBody().toString();
            str += "\n";
        }
        //---menampilkan pesan SMS baru---
        Toast.makeText(context, str, Toast.LENGTH_SHORT).show();
    }
}

```

```

    //---jalankan MainActivity---
    Intent mainActivityIntent = new Intent(context,
    MainActivity.class);
    mainActivityIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    context.startActivity(mainActivityIntent);

    //---mengirim broadcast intent untuk meng-update SMS yang
    diterima activity---
    Intent broadcastIntent = new Intent();
    broadcastIntent.setAction("SMS_RECEIVED_ACTION");
    broadcastIntent.putExtra("sms", str);
    context.sendBroadcast(broadcastIntent);
}
}

```

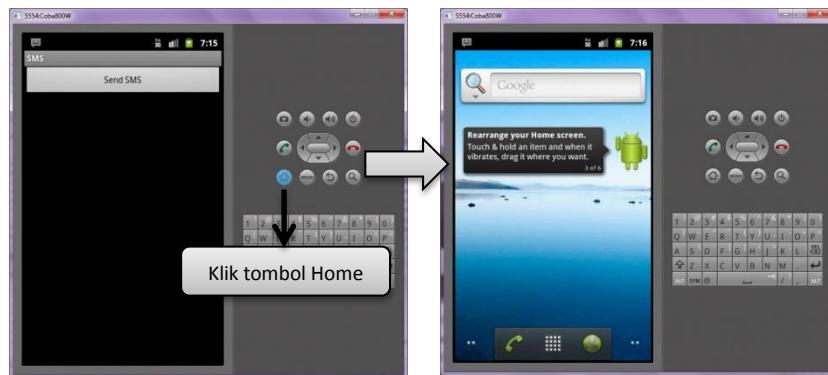
Modifikasi file AndroidManifest.xml seperti berikut ini:

```

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

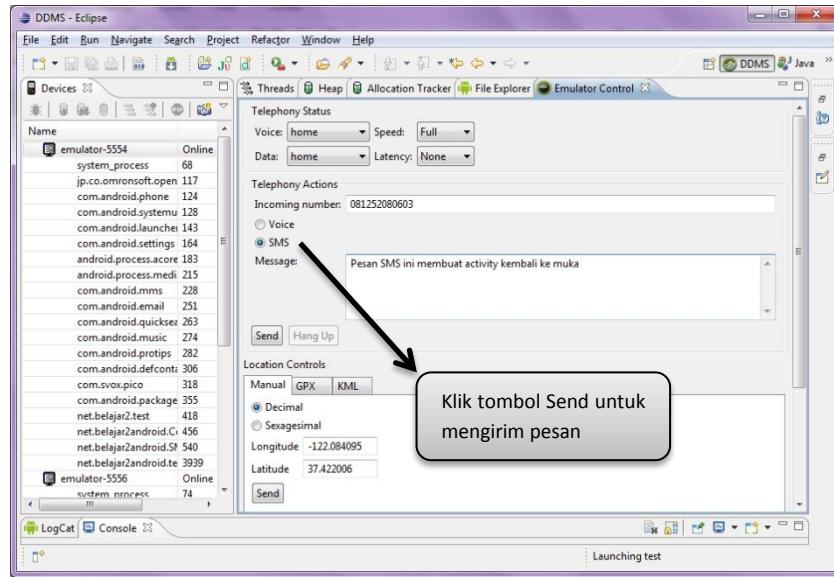
```

Jalankan aplikasi ke Emulator Android. Saat MainActivity ditampilkan, klik tombol **Home** untuk mengirim activity ke belakang layar (background).

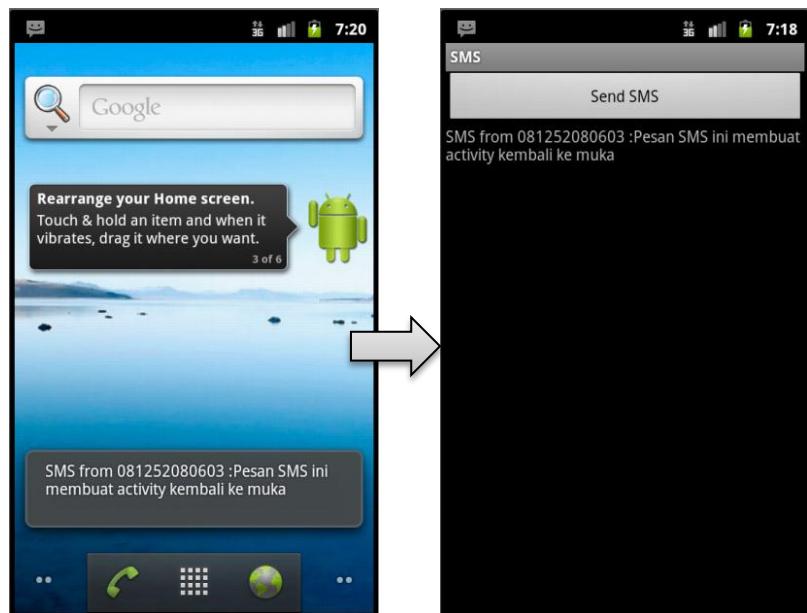


**Gambar 10.9 Klik home untuk mengirim main activity ke background**

Gunakan DDMS (telephony actions) untuk mengirim pesan SMS ke emulator lagi. Pada saat ini, activity akan dibawa ke muka lagi (foreground), menampilkan pesan SMS yang diterima.



Gambar 10.10 Mengirim pesan SMS melalui DDMS



**Gambar 10.11 Menempatkan main activity ke foreground setelah menerima pesan SMS**

Bagaimana program bekerja? Pertama me-register BroadcastReceiver dalam activity, khususnya event onCreate(), event onResume(), unregister event onPause() dan yang baru adalah penambahan event onDestroy(), hal ini untuk memastikan bila activity benar-benar berada di-background. Walaupun begitu activity ini masih mampu mendengar intent broadcast.

Selanjutnya, memodifikasi event onReceive() dalam class SMSReceiver melalui penggunaan intent untuk membawa activity ke foreground sebelum mem-broadcast intent lainnya:

```
----jalankan MainActivity---
Intent mainActivityIntent = new Intent(context,
MainActivity.class);
mainActivityIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
context.startActivity(mainActivityIntent);
```

```
//---mengirim broadcast intent untuk meng-update SMS yang  
diterima activity---  
Intent broadcastIntent = new Intent();  
broadcastIntent.setAction("SMS_RECEIVED_ACTION");  
broadcastIntent.putExtra("sms", str);  
context.sendBroadcast(broadcastIntent);
```

Method `startActivity()` menjalankan activity dan membawanya ke depan/foreground.

Kemudian mengeset atribut `launchMode`, khususnya elemen `<activity>` di dalam file `AndroidManifest.xml` menjadi `singleTask`:

```
<activity android:name=".MainActivity"  
        android:label="@string/app_name"  
        android:launchMode="singleTask">
```

Jika tidak mengesetnya maka akan banyak instance activity yang dijalankan, seperti aplikasi penerimaan pesan SMS. Perlu diperhatikan pula untuk mengeset flag `Intent.FLAG_ACTIVITY_NEW_TASK` karena pemanggilan `startActivity()` dari luar konteks activity membutuhkan flag `FLAG_ACTIVITY_NEW_TASK`.

Pada contoh di atas, ketika activity berada di background (seperti ketika mengklik tombol Home untuk menampilkan layar home), activity akan dibawa ke depan dan `TextView` di-update dengan SMS yang diterima. Namun jika activity dimatikan (seperti ketika mengklik tombol Back untuk menghapusnya), activity akan dijalankan lagi, tetapi `TextView` tidak di-update.

## 10.2 Pengiriman Email

Android juga mendukung pengiriman e-mail. Selain Gmail atau aplikasi email Android lainnya, Android juga mengizinkan Anda untuk mengkonfigurasi akun e-mail dengan protokol POP3 atau IMAP.

Disamping pengiriman e-mail menggunakan Gmail/aplikasi email. Anda juga dapat mengirim pesan e-mail secara program dari aplikasi Android.

Buatlah project baru bernama bernama **Email**.

<b>Project name</b>	Email
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	Email
<b>Package name</b>	net.belajar2android.Email
<b>Create Activity</b>	MainActivity

Modifikasi file main.xml seperti berikut ini:

```
<?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:id="@+id/btnSendEmail"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Send Email" />
</LinearLayout>
```

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.Email;

import android.app.Activity;
import android.os.Bundle;
import android.content.Intent;
import android.net.Uri;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {
    Button btnSendEmail;

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

        btnSendEmail = (Button) findViewById(R.id.btnSendEmail);
        btnSendEmail.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v)
            {
```

```

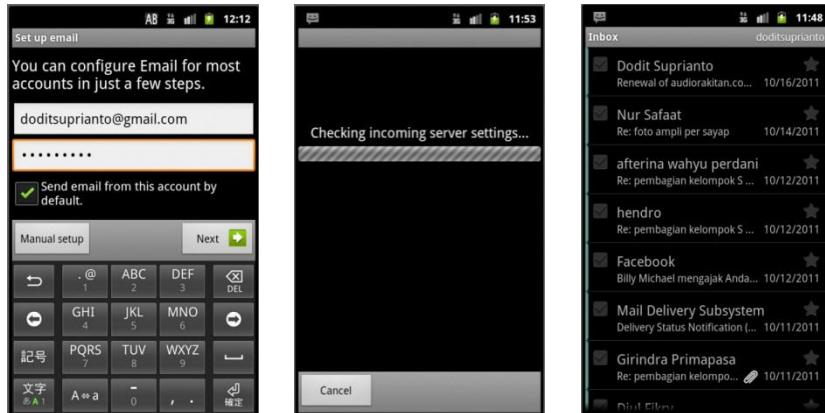
        String[] to = {"doditsuprianto@gmail.com",
"ryf4ny@yahoo.com"};
        String[] cc = {"d0dit@yahoo.com"};
        sendEmail(to, cc, "Halo", "Halooo teman!");
    }
});
}

//---mengirim pesan email ke perangkat lain ---
private void sendEmail(String[] emailAddresses, String[]
carbonCopies, String subject, String message) {
    Intent emailIntent = new Intent(Intent.ACTION_SEND);
    emailIntent.setData(Uri.parse("mailto:"));
    String[] to = emailAddresses;
    String[] cc = carbonCopies;
    emailIntent.putExtra(Intent.EXTRA_EMAIL, to);
    emailIntent.putExtra(Intent.EXTRA_CC, cc);
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, subject);
    emailIntent.putExtra(Intent.EXTRA_TEXT, message);
    emailIntent.setType("message/rfc822");
    startActivity(Intent.createChooser(emailIntent, "Email"));
}
}

```

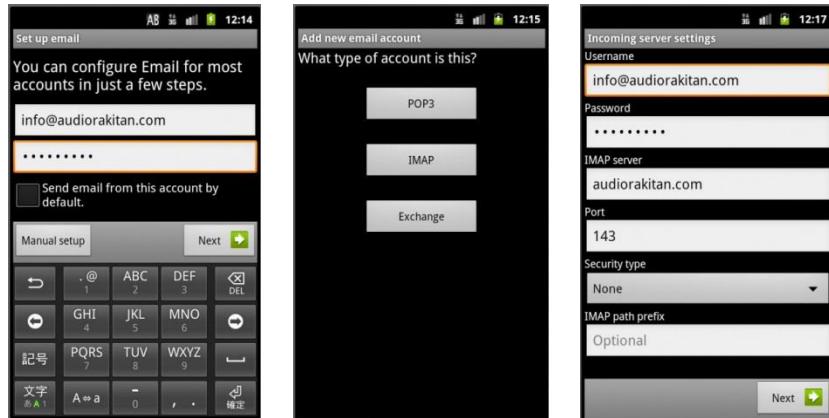
Jika belum melakukan konfigurasi terhadap aplikasi email, silakan pilih

menu Launcher|Email () , gambar di bawah ini adalah konfigurasi untuk Gmail.



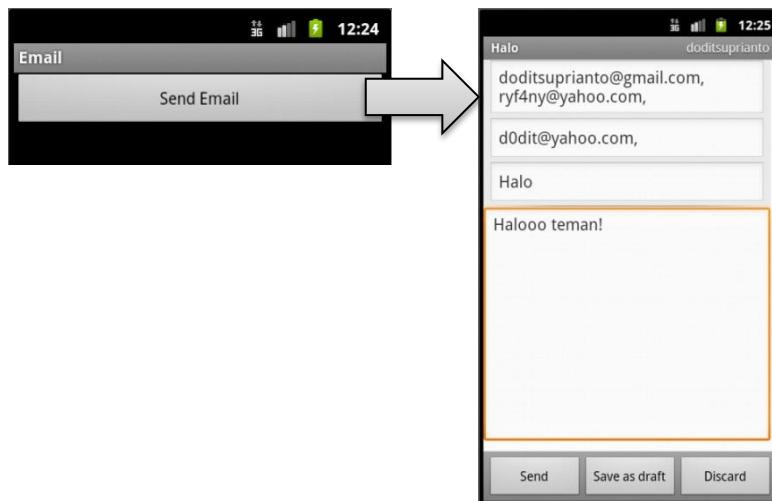
**Gambar 10.12 Konfigurasi Email GMail**

Berikut ini pengaturan server email yang lain:



**Gambar 10.13 Konfigurasi Email non GMail**

Jalankan aplikasi email yang baru saja dibuat. Klik tombol **Email** dan kemudian klik tombol **Send**, maka hasilnya akan seperti berikut ini:



**Gambar 10.14 Kirim Email dari program**

Kalau aplikasi terkoneksi dengan internet maka akan didapatkan sebuah email dari emulator android ini.

Bagaimana program bekerja? Aplikasi menjalankan Email secara built-in untuk mengirim pesan email. Oleh karenanya digunakan object Intent dan mengeset beberapa parameter menggunakan method setData(), putExtra() dan setType().

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);
emailIntent.setData(Uri.parse("mailto:"));
String[] to = emailAddresses;
String[] cc = carbonCopies;

emailIntent.putExtra(Intent.EXTRA_EMAIL, to);
emailIntent.putExtra(Intent.EXTRA_CC, cc);
emailIntent.putExtra(Intent.EXTRA_SUBJECT, subject);
emailIntent.putExtra(Intent.EXTRA_TEXT, message);
emailIntent.setType("message/rfc822");

startActivity(Intent.createChooser(emailIntent, "Email"));
```

### 10.3 Jaringan/Networking

Cara lain untuk mendapatkan koneksi ke dunia luar adalah melalui penggunaan protocol HTTP. Dengan protocol HTTP dapat dilakukan berbagai macam tugas, seperti men-download halaman website dari web server, men-download data biner dan lain-lain.

Buatlah project baru bernama **Networking**.

<b>Project name</b>	Networking
<b>Build Target</b>	Android 2.3.3, API 10
<b>Application name</b>	Networking
<b>Package name</b>	net.belajar2android.Networking
<b>Create Activity</b>	MainActivity

Modifikasi file AndroidManifest.xml seperti berikut ini:

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

```
package="net.belajar2android.Networking"
    android:versionCode="1"
    android:versionName="1.0">
<uses-sdk android:minSdkVersion="10" />
<uses-permission
    android:name="android.permission.INTERNET"></uses-permission>

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

Impor beberapa namespaces ke dalam file MainActivity.java dan definisikan pula method OpenHttpConnection():

```
package net.belajar2android.Networking;

import android.app.Activity;
import android.os.Bundle;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.widget.ImageView;
import android.widget.Toast;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class MainActivity extends Activity {
    private InputStream OpenHttpConnection(String urlString)
throws IOException
    {
        InputStream in = null;
        int response = -1;
        URL url = new URL(urlString);
        URLConnection conn = url.openConnection();
```

```

        if (!(conn instanceof HttpURLConnection)) throw new
IOException("Not an HTTP connection");
try {
    HttpURLConnection httpConn = (HttpURLConnection) conn;
    httpConn.setAllowUserInteraction(false);
    httpConn.setInstanceFollowRedirects(true);
    httpConn.setRequestMethod("GET");
    httpConn.connect();
    response = httpConn.getResponseCode();
    if (response == HttpURLConnection.HTTP_OK) {
        in = httpConn.getInputStream();
    }
}
catch (Exception ex)
{
    throw new IOException("Error connecting");
}
return in;
}

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

```

Bagaimana program bekerja? Karena menggunakan protocol HTTP untuk berhubungan dengan Web, aplikasi memerlukan izin INTERNET; oleh karena itu, pertama harus ditambahkan izin dalam file AndroidManifest.xml.

Kemudian mendefinisikan method OpenHttpConnection() yang mengambil string URL dan mengembalikan object InputStream. Dengan object InputStream dapat mendownload data melalui pembacaan bytes dari object stream. Dalam method ini, digunakan object HttpURLConnection untuk membuka koneksi HTTP dengan remote URL. Kemudian mengeset property koneksi, seperti method request dan lain-lain:

```

HttpURLConnection httpConn = (HttpURLConnection) conn;
httpConn.setAllowUserInteraction(false);
httpConn.setInstanceFollowRedirects(true);
httpConn.setRequestMethod("GET");

```

Setelah membentuk koneksi dengan server, akan diperoleh tanggapan dari HTTP berupa kode. Jika koneksi terbentuk (melalui kode response `HTTP_OK`), kemudian lakukan proses mendapatkan object `InputStream` dari koneksi tersebut:

```
httpConn.connect();
response = httpConn.getResponseCode();
if (response == HttpURLConnection.HTTP_OK) {
    in = httpConn.getInputStream();
}
```

### 10.3.1 Men-Download Data Binary

Berikut ini cara mendownload file biner dari web, misalnya file *image* dari server kemudian menampilkannya ke aplikasi.

Masih tetap menggunakan sebelumnya (project Networking), modifikasi file `main.xml` seperti berikut ini:

```
<?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" >
<ImageView
    android:id="@+id/img"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center" />
</LinearLayout>
```

Modifikasi file `MainActivity.java` seperti berikut ini:

```
package net.belajar2android.Networking;

import android.app.Activity;
import android.os.Bundle;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.widget.ImageView;
import android.widget.Toast;
```

```

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class MainActivity extends Activity {
    ImageView img;

    private InputStream OpenHttpConnection(String urlString)
    throws IOException
    {
        //...
    }

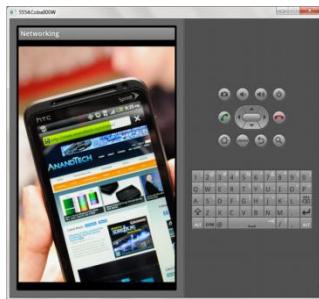
    private Bitmap DownloadImage(String URL)
    {
        Bitmap bitmap = null;
        InputStream in = null;
        try {
            in = OpenHttpConnection(URL);
            bitmap = BitmapFactory.decodeStream(in);
            in.close();
        } catch (IOException e1) {
            Toast.makeText(this, e1.getLocalizedMessage(),
            Toast.LENGTH_LONG).show();
            e1.printStackTrace();
        }
        return bitmap;
    }

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

        //----download image---
        Bitmap bitmap =
        DownloadImage("http://images.anandtech.com/doc1/4416/EVO3D-
9423.jpg");
        img = (ImageView) findViewById(R.id.img);
        img.setImageBitmap(bitmap);
    }
}

```

Jalankan aplikasi ke Emulator Android. Gambar di bawah ini menunjukkan image yang di-download dari Web, kemudian ditampilkan ke ImageView.



### 10.3.2 Men-Download File Text

Disamping mendownload data binary, kita juga dapat mendownload file plain-text, misalnya menuliskan aplikasi RSS Reader dengan mendownload file RSS XML feeds untuk memprosesnya.

Masih tetap menggunakan project **Networking**, modifikasi file `MainActivity.java` seperti berikut ini:

```
package net.belajar2android.Networking;

import android.app.Activity;
import android.os.Bundle;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.widget.ImageView;
import android.widget.Toast;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class MainActivity extends Activity {
    ImageView img;
    private InputStream OpenHttpConnection(String urlString)
        throws IOException
```

```

{
    //...
}

private Bitmap DownloadImage(String URL)
{
    //...
}

private String DownloadText(String URL)
{
    int BUFFER_SIZE = 2000;
    InputStream in = null;
    try {
        in = OpenHttpConnection(URL);
    } catch (IOException e1) {
        Toast.makeText(this, e1.getLocalizedMessage(),
        Toast.LENGTH_LONG).show();
        e1.printStackTrace();
        return "";
    }

    InputStreamReader isr = new InputStreamReader(in);
    int charRead;
    String str = "";
    char[] inputBuffer = new char[BUFFER_SIZE];
    try {
        while ((charRead = isr.read(inputBuffer))>0)
        {
            //---konversi char ke string---
            String readString = String.valueOf(inputBuffer, 0,
charRead);
            str += readString;
            inputBuffer = new char[BUFFER_SIZE];
        }
        in.close();
    } catch (IOException e) {
        Toast.makeText(this, e.getLocalizedMessage(),
        Toast.LENGTH_LONG).show();
        e.printStackTrace();
        return "";
    }
    return str;
}

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

    //---download image---
    Bitmap bitmap =
DownloadImage("http://images.anandtech.com/doci/4416/EVO3D-
9423.jpg");
}

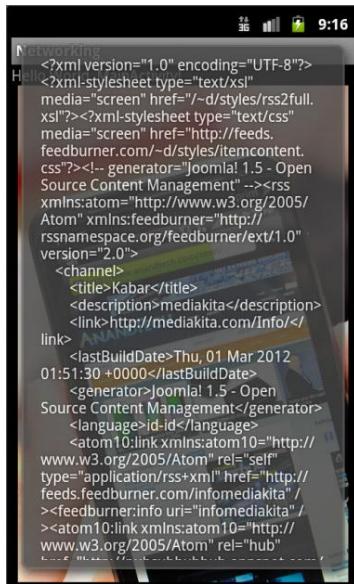
```

```

        img = (ImageView) findViewById(R.id.img);
        img.setImageBitmap(bitmap);

        //---download RSS feed---
        String str =
        DownloadText("http://feeds.feedburner.com/infomediakita.rss");
        Toast.makeText(getApplicationContext(), str,
        Toast.LENGTH_SHORT).show();
    }
}

```



**Gambar 10.16 Download RSS Feed XML**

Bagaimana program bekerja? Method `DownloadText()` mengambil alamat URL file text untuk di-download, kemudian mengembalikan string file text yang didownload. Pada dasarnya adalah membuka koneksi HTTP server, kemudian menggunakan object `InputStreamReader` untuk membaca setiap karakter dari stream dan menyimpannya ke dalam object `String`.

# BAB 11

# LAYANAN LOKASI

Salah satu aplikasi perangkat bergerak yang popular adalah layanan lokasi/peta yang umum dikenal sebagai LBS (*location-based services*). Aplikasi LBS menelusuri informasi lokasi tertentu, seperti lokasi-lokasi hiburan terdekat, saran-saran jalur bepergian dan lain-lain. Kunci utama LBS adalah peta yang akan menampilkan sebagian dari lokasi secara visual.

Pada bagian ini akan dipelajari bagaimana penggunaan Google Maps dalam aplikasi, bagaimana memanipulasinya secara program dan bagaimana memperoleh informasi lokasi geografis dengan menggunakan class LocationManager yang terdapat dalam SDK Android.

## 11.1 Peta

Adapun yang bisa dipelajari dari Google Maps pada aplikasi Android adalah sebagai berikut: Mengubah view Google Maps, Mendapatkan informasi bujur dan lintang suatu lokasi dalam Google Maps, Melakukan geocoding dan membalik geocoding (yaitu menerjemahkan alamat ke bentuk bujur dan lintang), serta Menambahkan penanda (*marker*) di Google Maps.

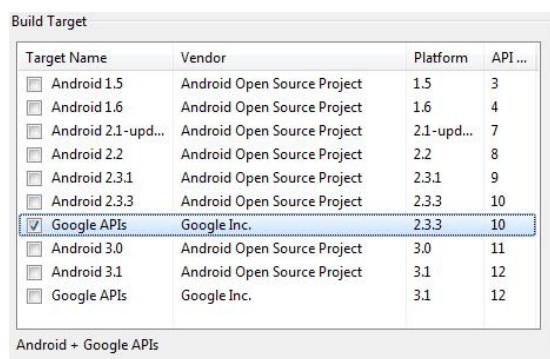
### 11.1.1 Membuat Project Peta

Buat project Android baru bernama **Peta**. Untuk penggunaan Google Maps di dalam aplikasi Android, pastikan untuk memberi tanda centang pada Google APIs di dalam build target.

<b>Project name</b>	Peta
<b>Build Target</b>	Google APIs, Google Inc, 2.3.3

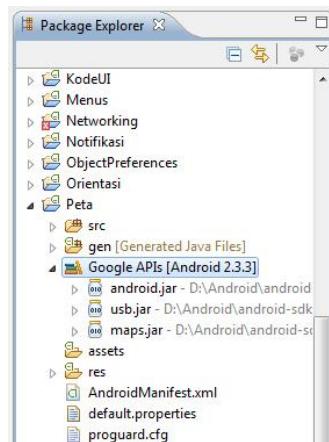
<b>Application name</b>	Peta
<b>Package name</b>	net.belajar2android.Peta
<b>Create Activity</b>	MainActivity

Google Maps bukanlah bagian standar dari SDK Android, sehingga layanannya perlu didownload melalui Google APIs add-on.



**Gambar 11.1 Build Target Google APIs**

Setelah project terbentuk maka ada tambahan file JAR (maps.jar) yang terletak di bawah folder Google API.



**Gambar 11.2 File maps.jar**

### **11.1.2 Mendapatkan Maps API Key**

Sebelum menggabungkan Google Maps ke dalam aplikasi Android, diperlukan Google Maps API key. Cara mendapatkan Maps API Key adalah sebagai berikut.

Jika pengujian aplikasi dilakukan di Emulator Android atau perangkat Android yang terhubung langsung ke mesin pengembangan, cari lokasi SDK debug certificate di dalam folder defaultnya (untuk pengguna Windows 7 adalah C:\Users\<username>\.android). Anda dapat memverifikasi debug certificate melalui Eclipse dengan memilih menu **Windows|Preferences**. Perluas item Android dan pilih Build. Pada sisi kanan windows akan terlihat lokasi debug certificate.

**Catatan:**

- Google menyediakan dokumentasi lengkap bagaimana mengaplikasikan Maps API key di <http://code.google.com/android/add-ons/google-apis/mapkey.html>.
- Untuk pengguna Windows XP, folder default Android adalah C:\Documents and Settings\<username>\Local Settings\Application Data\Android.

Nama file debug keystore adalah **debug.keystore**, ini adalah certificate yang Eclipse gunakan untuk menandai aplikasi, oleh karena itu ia dapat berjalan pada emulator atau perangkat Android.

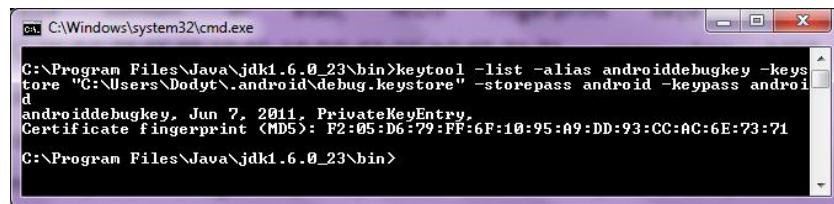
Penggunaan debug keystore dengan cara menguraikan fingerprint MD5 melalui aplikasi **Keytool.exe** yang terdapat pada instalasi JDK. Fingerprint ini diperlukan untuk mengaplikasikan Google Maps key secara gratis. Pada umumnya Keytool.exe terletak di dalam folder **C:\Program Files\Java\<JDK\_version\_number>\bin**.

Untuk menguraikan MD5 fingerprint melalui perintah command seperti berikut ini:

```
keytool.exe -list -alias androiddebugkey -keystore  
"C:\Users\<username>\.android\debug.keystore" -storepass android  
-keypass android
```

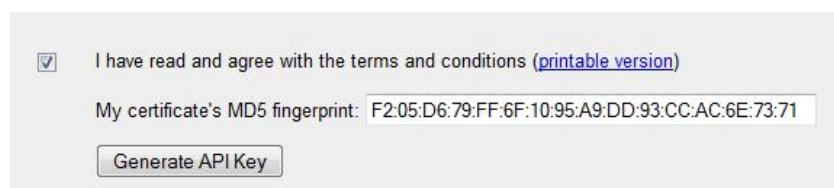
contoh (storepass: "android" dan keypass: "android")

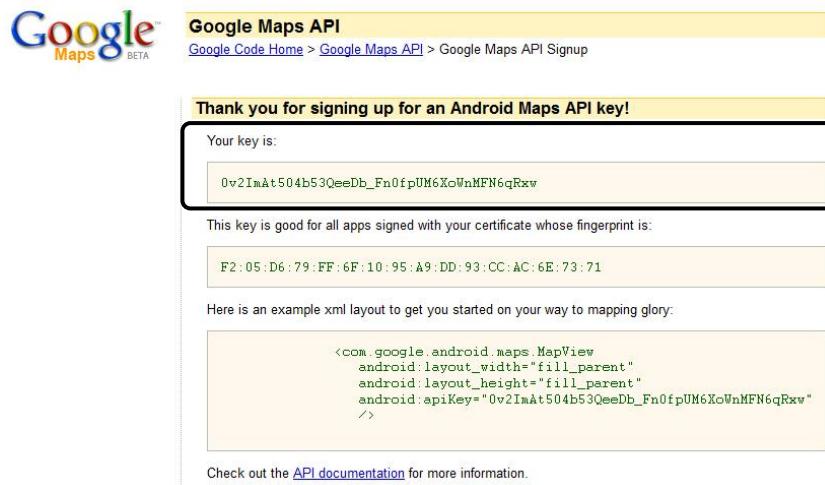
```
keytool.exe -list -alias androiddebugkey -keystore  
C:\Users\Dodyt\.android\debug.keystore -storepass android -  
keypass android
```



**Gambar 11.3 Membuat debug keystore dengan keytool**

Pada contoh di atas, MD5 fingerprint saya adalah F2:05:D6:79:FF:6F:10:95:A9:DD:93:CC:AC:6E:73:71, copy-kan MD5 certificate tersebut ke <http://code.google.com/android/maps-api-signup.html>. Ikuti instruksi yang terdapat pada website tersebut untuk mendapatkan Google Maps key. Ketika telah selesai, maka tampilannya kurang lebih seperti berikut:





**Gambar 11.4 Mendaftarkan MD5 ke Google Maps API**

**Catatan:** Meskipun dapat menggunakan MD5 fingerprint dari debug store untuk mendapatkan Maps API key untuk menjalankan aplikasi pada Emulator atau perangkat Android, key tersebut tidak akan valid jika di-deploy menjadi file APK (akan di bahas pada bab deploy).

### 11.1.3 Menampilkan Peta

Sekarang kita siap menampilkan Google Maps ke dalam aplikasi Android. Hal ini melibatkan dua tugas utama:

- Memodifikasi file `AndroidManifest.xml` dengan menambahkan element `<uses-library>` dan `INTERNET` permission.
- Menambahkan element `MapView` ke UI.

Gunakan project **Peta** pada bagian sebelumnya.

Modifikasi file main.xml seperti berikut ini:

```
<?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" >
    <com.google.android.maps.MapView
        android:id="@+id/mapView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:enabled="true"
        android:clickable="true"
        android:apiKey="0v2ImAt504b53QeeDb_Fn0fpUM6XoWnMFN6qRxw" />
</LinearLayout>
```

Modifikasi file AndroidManifest.xml seperti berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
        package="net.belajar2android.Peta"
        android:versionCode="1"
        android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <uses-library android:name="com.google.android.maps" />
        <activity android:name=".MainActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="10" />
    <uses-permission
        android:name="android.permission.INTERNET"></uses-permission>
</manifest>
```

Modifikasi file MainActivity.java seperti berikut ini:

```
package net.belajar2android.Peta;

import android.os.Bundle;
import com.google.android.maps.MapActivity;

public class MainActivity extends MapActivity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

        setContentView(R.layout.main);
    }

    @Override
    protected boolean isRouteDisplayed() {
        // TODO Auto-generated method stub
        return false;
    }
}

```



**Gambar 11.5 Google Maps pada Android**

Bagaimana program bekerja? pertama diperlukan INTERNET permission pada file manifest. Tambahkan elemen <com.google.android.maps.MapView> ke dalam file UI untuk melekatkan peta dalam anctivity. Kemudian Activity meng-extend class MapActivity, dimana activity itu sendiri adalah extension dari class Activity. Untuk class MapActivity, perlu diimplementasikan satu method **isRouteDisplayed()**. Method ini digunakan oleh Google untuk tujuan perhitungan, kemudian mengembalikan nilai true untuk method ini jika menampilkan informasi routing pada peta.

#### 11.1.4 Menampilkan Zoom Control

Kita dapat menggali lebih dalam sebuah peta dari lokasi yang dinginkan melalui pemanfaatan kontrol *zoom-in* atau *zoom-out* pada lokasi tertentu.

Masih tetap menggunakan project **Peta**, modifikasi file `MainActivity.java` seperti berikut ini:

```
package net.belajar2android.Peta;

//import android.app.Activity;
import android.os.Bundle;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapView;

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

        mapView = (MapView) findViewById(R.id.mapView);
        mapView.setBuiltInZoomControls(true);
    }

    @Override
    protected boolean isRouteDisplayed() {
        // TODO Auto-generated method stub
        return false;
    }
}
```

Jalankan aplikasi ke Emulator. Perhatikan kontrol zoom yang tampak di bawah, klik ikon minus (-) untuk memperkecil peta dan ikon plus (+) untuk memperbesar peta.



**Gambar 11.6 Tampilan default peta**

Bagaimana program bekerja? Untuk menampilkan kontrol zoom, pertama harus didapatkan rujukan ke peta dan memanggil method `setBuiltInZoomControls()`:

```
mapView = (MapView) findViewById(R.id.mapView);
mapView.setBuiltInZoomControls(true);
```

Kita dapat memrogram kontrol zoom dengan menggunakan method `zoomIn()` atau `zoomOut()` dari class `MapController`. Masih tetap menggunakan project **Peta**, modifikasi file `MainActivity.java` seperti berikut ini:

```
package net.belajar2android.Peta;

//import android.app.Activity;
import android.os.Bundle;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapView;
import android.view.KeyEvent;
import com.google.android.maps.MapController;

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

        mapView = (MapView) findViewById(R.id.mapView);
        mapView.setBuiltInZoomControls(true);
    }

    public boolean onKeyDown(int keyCode, KeyEvent event)
    {
        MapController mc = mapView.getController();

        switch (keyCode)
        {
            case KeyEvent.KEYCODE_3:
                mc.zoomIn();
                break;
            case KeyEvent.KEYCODE_1:
                mc.zoomOut();
                break;
        }
        return super.onKeyDown(keyCode, event);
    }

    @Override
```

```
protected boolean isRouteDisplayed() {
    // TODO Auto-generated method stub
    return false;
}
```

Jalankan aplikasi ke Emulator Android. Sekarang peta dapat diperbesar dengan menekan key angka 3 dari emulator atau memperkecil peta dengan menekan key angka 1 dari emulator.

Bagaimana program bekerja? Untuk menangani penekanan key pada activity digunakan event onKeyDown:

```
public boolean onKeyDown(int keyCode, KeyEvent event)
{
    //...
}
```

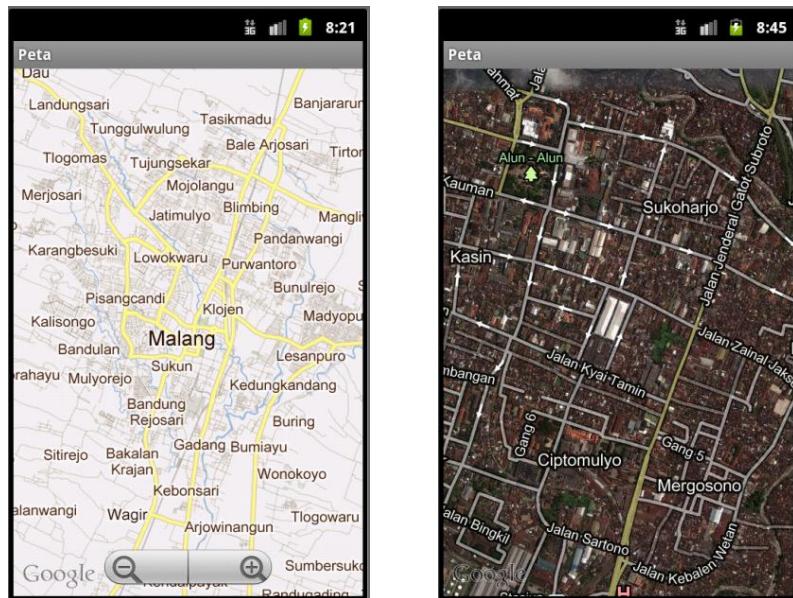
Untuk memperbesar peta diperlukan instansiasi class MapController dari object MapView. Class MapController berisi method zoomIn() dan zoomOut() (termasuk beberapa method lainnya yang bertujuan mengendalikan peta) untuk memperbesar dan memperkecil skala peta.

### 11.1.5 Mengubah View Peta

Secara default, Google Maps ditampilkan dalam view map, namun Google Maps juga bisa ditampilkan dalam bentuk view street (sekarang sudah tidak dukung oleh Google Maps) dan view satellite. Untuk view Satellite digunakan method setSatellite() dari class MapView, seperti berikut ini:

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

    mapView = (MapView) findViewById(R.id.mapView);
    mapView.setBuiltInZoomControls(true);
    mapView.setSatellite(true);
}
```



**Gambar 11.7 Peta ditampilkan dalam view map dan view street**

### 11.1.6 Menavigasi Lokasi Tertentu

Secara default, Google Map menampilkan peta Amerika Serikat ketika pertama kali di-load. Namun, Google Map bisa diatur untuk menampilkan lokasi tertentu dengan menggunakan method `animateTo()` dari class `MapController`.

Masih tetap menggunakan project **Peta**, modifikasi file `MainActivity.java` seperti berikut ini:

```
package net.belajar2android.Peta;

//import android.app.Activity;
import android.os.Bundle;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapView;
import android.view.KeyEvent;
import com.google.android.maps.MapController;
import com.google.android.maps.GeoPoint;
```

```

public class MainActivity extends MapActivity {
    MapView mapView;
    MapController mc;
    GeoPoint p;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mapView = (MapView) findViewById(R.id.mapView);
        mapView.setBuiltInZoomControls(true);
        //mapView.setSatellite(true);
        //mapView.setStreetView(true);

        mc = mapView.getController();
        String coordinates[] = {"-7.98444444", "112.631944444"};
        double lat = Double.parseDouble(coordinates[0]);
        double lng = Double.parseDouble(coordinates[1]);
        p = new GeoPoint((int) (lat * 1E6), (int) (lng * 1E6));
        mc.animateTo(p);
        mc.setZoom(15);
        mapView.invalidate();
    }

    public boolean onKeyDown(int keyCode, KeyEvent event)
    {
        MapController mc = mapView.getController();
        switch (keyCode)
        {
            case KeyEvent.KEYCODE_3:
                mc.zoomIn();
                break;
            case KeyEvent.KEYCODE_1:
                mc.zoomOut();
                break;
        }
        return super.onKeyDown(keyCode, event);
    }

    @Override
    protected boolean isRouteDisplayed() {
        // TODO Auto-generated method stub
        return false;
    }
}

```

Bagaimana program bekerja? Pertama didapatkan map controller dari instance MapView dan menugaskannya ke object MapController (mc). Kemudian gunakan object GeoPoint untuk mewakili lokasi geografis. Perlu diingat bahwa pada class tersebut lintang dan bujur suatu lokasi

diwakili dalam derajat mikro, sehingga harus disimpan dan dikonversikan menjadi derajat desimal terlebih dahulu.

Rumus DerajatMikro (format E6) = DerajatDesimal \* 1E6.

Contoh, diberikan koordinat kota Malang dengan DMS (Degrees, Minutes, Seconds) adalah **07° 59' 4" LS** dan **112° 37' 55" BT**. Koordinat DMS tersebut dikonversikan menjadi desimal, sehingga didapatkan koordinat baru **-7.984444444** dan **+112.631944444**. Adapun cara perhitungannya adalah sebagai berikut:

Lintang Selatan (LS)	Bujur Timur (BT)
<p>Rumus = <math>(7+(59*60 + 4*1))/3600</math></p> <ul style="list-style-type: none"><li>• Total derajat = 7</li><li>• Total detik = <math>59' 4'' = (59*60 + 4*1) = 3544</math> detik</li><li>• Pembagian total jumlah detik dengan 3600 adalah <math>3580 / 3600 = 0.984444444</math></li><li>• Jumlahkan total derajat dengan total hasil pembagian detik, <math>7+0.984444444 = 7.984444444</math></li><li>• Sehingga koordinat desimalnya adalah <b>-7.984444444 LS</b></li></ul>	<p>Rumus= <math>(112+(37*60+55*1))/3600</math></p> <ul style="list-style-type: none"><li>• Total derajat = 112</li><li>• Total detik = <math>36' 37'' = (37*60 + 55*1) = 2275</math> detik</li><li>• Pembagian total jumlah detik dengan 3600 adalah <math>2275 / 3600 = 0.631944444</math></li><li>• Jumlahkan total derajat dengan total hasil pembagian detik, <math>112+0.631944444= 112.631944444</math></li><li>• Sehingga Koordinat desimalnya adalah <b>+112.631944444 BT</b></li></ul>

Berikut adalah referensi aturan dasarnya untuk menentukan kapan bujur dan lintang bernilai positif atau negatif:

- Lintang Utara (Latitude North) = +
- Lintang Selatan (Latitude Selatan) = -
- Bujur Timur (Longitude East) = +

- Bujur Barat (Longitude West) = -

Dari koordinat format desimal kemudian diubah lagi menjadi koordinat GeoPoint format E6 yang diacu oleh Google Map, seperti berikut ini:

```
p = new GeoPoint((int)(lat * 1E6), (int)(lng * 1E6));
```

Untuk menavigasi peta ke lokasi tertentu dapat digunakan method `animateTo()` dari class `MapController`. Method `setZoom()` untuk menentukan tingkat perbesaran peta yang akan ditampilkan (semakin besar angkanya, semakin rinci peta yang akan tampak). Method `invalidate()` memaksa `MapView` untuk menggambar ulang.

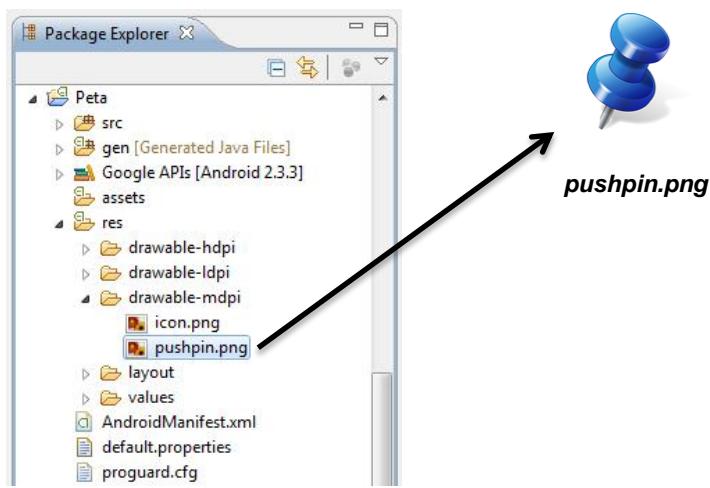


*Gambar 11.8 Peta dengan posisi lintang dan bujur tertentu*

### 11.1.7 Menambahkan Marker

Marker pada peta bertujuan untuk memberi perhatian lebih terhadap suatu lokasi tertentu agar user mudah mencari tempat-tempat yang dianggap menarik untuk dilihat.

Berikut ini cara membuatnya. Copy file gambar *pushpin* berformat GIF/PNG ke folder res/drawable-mdpi dari project Peta. Untuk efek terbaik, usahakan latar belakang gambarnya transparan sehingga tidak tampak bagian blok kotaknya saat gambar *pushpin* ditambahkan ke peta.



**Gambar 11.9 Menambahkan *pushpin.png* ke project**

Mais tetap menggunakan project sebelumnya (project Peta), modifikasi file *MainActivity.java* seperti berikut ini:

```
package net.belajar2android.Peta;

//import android.app.Activity;
import android.os.Bundle;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapView;
import android.view.KeyEvent;
import com.google.android.maps.MapController;
import com.google.android.maps.GeoPoint;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Point;
import com.google.android.maps.Overlay;
import java.util.List;

public class MainActivity extends MapActivity {
    MapView mapView;
```

```

MapController mc;
GeoPoint p;

class MapOverlay extends com.google.android.maps.Overlay {
    @Override
    public boolean draw(Canvas canvas, MapView mapView, boolean
shadow, long when) {
        super.draw(canvas, mapView, shadow);
        //---menterjemahkan GeoPoint ke pixel layar---
        Point screenPts = new Point();
        mapView.getProjection().toPixels(p, screenPts);
        //---menambahkan marker---
        Bitmap bmp = BitmapFactory.decodeResource(getResources(),
R.drawable.pushpin);
        canvas.drawBitmap(bmp, screenPts.x, screenPts.y-40, null);
        return true;
    }
}

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

    mapView = (MapView) findViewById(R.id.mapView);
    mapView.setBuiltInZoomControls(true);
    //mapView.setSatellite(true);
    //mapView.setStreetView(true);

    mc = mapView.getController();
    String coordinates[] = {"-7.98444444", "112.631944444"};
    double lat = Double.parseDouble(coordinates[0]);
    double lng = Double.parseDouble(coordinates[1]);
    p = new GeoPoint((int) (lat * 1E6), (int) (lng * 1E6));
    mc.animateTo(p);
    mc.setZoom(15);
    //---menambahkan lokasi marker---
    MapOverlay mapOverlay = new MapOverlay();
    List<Overlay> listOverlays = mapView.getOverlays();
    listOverlays.clear();
    listOverlays.add(mapOverlay);
    mapView.invalidate();
}

public boolean onKeyDown(int keyCode, KeyEvent event) {
    MapController mc = mapView.getController();
    switch (keyCode)
    {
        case KeyEvent.KEYCODE_3:
            mc.zoomIn();
            break;
        case KeyEvent.KEYCODE_1:
            mc.zoomOut();
            break;
    }
}

```

```

        }
        return super.onKeyDown(keyCode, event);
    }

@Override
protected boolean isRouteDisplayed() {
    // TODO Auto-generated method stub
    return false;
}
}

```

Sekarang jalankan aplikasi ke Emulator Android. Maka tampilannya kurang lebih seperti ini:



**Gambar 11.10 Peta disertai marker pushpin**

Bagaimana program bekerja? Untuk menambahkan marker ke peta, pertama dibutuhkan pendefinisian class yang meng-extend class Overlay:

```

class MapOverlay extends com.google.android.maps.Overlay {
    @Override
    public boolean draw(Canvas canvas, MapView mapView, boolean
shadow, long when) {
    //...
}

```

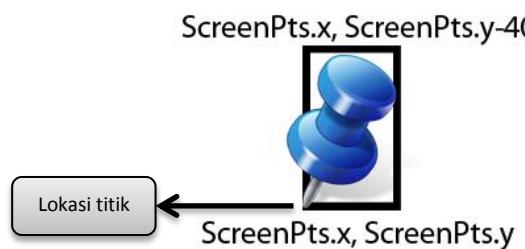
```
    }  
}
```

Sebuah overlay mewakili item secara individual yang nantinya akan gambarkan ke peta. Beberapa overlay bisa ditambahkan sebanyak yang diinginkan. Dalam class MapOverlay, method draw() di-override sehingga image pushpin dapat digambar pada peta. Kita juga perlu menerjemahkan lokasi geografis (diwakili oleh object GeoPoint) ke dalam bentuk koordinat layar.

```
//---menterjemahkan GeoPoint ke pixel layar---  
Point screenPts = new Point();  
mapView.getProjection().toPixels(p, screenPts);
```

Karena ujung penunjuk pushpin menjadi penanda posisi lokasi, maka kita perlu mengurangi tinggi image pushpin (pushpin berukuran 40x40 pixel) dari titik koordinat y dan menggambarkan image pada lokasi, seperti berikut ini:

```
//---menambahkan marker---  
Bitmap bmp = BitmapFactory.decodeResource(getResources(),  
R.drawable.pushpin);  
canvas.drawBitmap(bmp, screenPts.x, screenPts.y-40, null);
```



**Gambar 11.11 Penentuan koordinat ujung pushpin**

Untuk menambahkan marker, ciptakan instance class MapOverlay dan tambahkan ke daftar overlay yang tersedia pada object mapView:

```
//---menambahkan lokasi marker---  
MapOverlay mapOverlay = new MapOverlay();  
List<Overlay> listOfOverlays = mapView.getOverlays();  
listOfOverlays.clear();
```

```
listOfOverlays.add(mapOverlay);
```

### 11.1.8 Mendapatkan Informasi Lokasi Yang Disentuh

Terkadang user perlu tahu lintang dan bujur suatu lokasi, dimana posisinya ditentukan saat layar disentuh.

Kita dapat meng-override method `onTouchEvent()` pada class `MapOverlay`. Method ini dipicu setiap saat ketika user menyentuh peta. Method ini memiliki dua parameter, yaitu: `MotionEvent` dan `MapView`. Parameter `MotionEvent` untuk menentukan apakah user mengangkat jari dari layar melalui penggunaan method `getAction()`.

Potongan kode berikut adalah bila user menyentuh layar kemudian mengangkat jari, yaitu menampilkan lintang dan bujur lokasi yang disentuh:

```
import android.view.MotionEvent;
import android.widget.Toast;

//...

class MapOverlay extends com.google.android.maps.Overlay {
    @Override
    public boolean draw(Canvas canvas, MapView mapView, boolean
shadow, long when) {
        super.draw(canvas, mapView, shadow);
        //---menterjemahkan GeoPoint ke pixel layar---
        Point screenPts = new Point();
        mapView.getProjection().toPixels(p, screenPts);
        //---menambahkan marker---
        Bitmap bmp = BitmapFactory.decodeResource(getResources(),
R.drawable.pushpin);
        canvas.drawBitmap(bmp, screenPts.x, screenPts.y-40, null);
        return true;
    }

    @Override
    public boolean onTouchEvent(MotionEvent event, MapView
mapView) {
        //---ketika user mengangkat jari---
        if (event.getAction() == 1) {
            GeoPoint p = mapView.getProjection().fromPixels((int)
event.getX(), (int) event.getY());
        }
    }
}
```

```

        Toast.makeText(getApplicationContext(), "Lokasi: " +
p.getLatitudeE6() / 1E6 + "," + p.getLongitudeE6() /1E6,
Toast.LENGTH_SHORT).show();
    }
    return false;
}
}

```

Method `getProjection()` mendapatkan proyeksi yang mengkonversi antara koordinat pixel layar dan koordinat lintang/bujur. Kemudian method `fromPixels()` mengkonversi koordinat layar menjadi object `GeoPoint`.

Gambar di bawah ini menunjukkan pasangan koordinat bujur/lintang ketika user mengklik lokasi peta.



**Gambar 11.12 Koordinat lokasi ketika peta diklik**

# BAB 12

## PUBLIKASI APLIKASI ANDROID

Bab ini akan mempelajari bagaimana men-deploy (menjadikan file biner yang sudah terkompilasi) dan mendistribusikannya dengan beberapa cara.

### 12.1 Persiapan Publikasi

Langkah-langkah publikasi aplikasi Android secara umum meliputi:

- Mengekspor aplikasi Android menjadi file APK (Android Package).
- Membangkitkan *self-signed certificate* Anda sendiri dan membuat *digital signed* pada aplikasi.
- Men-deploy aplikasi yang sudah di-signed.
- Menempatkan aplikasi ke hosting Android Market.

#### 12.1.1 Pemberian Versi

Awalilah Android SDK dengan versi 1.0 yang diletakkan dalam file `AndroidManifest.xml` khususnya pada atribut **android:versionCode** dan **android:versionName**:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.learn2develop.LBS"
        android:versionCode="1"
        android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <uses-library android:name="com.google.android.maps" />
        <activity android:name=".MainActivity"
            android:label="@string/app_name">
            <intent-filter>
```

```
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
<uses-sdk android:minSdkVersion="7" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
    android:name="android.permission.ACCESS_COARSE_LOCATION" />
</manifest>
```

Atribut android:versionCode mewakili nomor versi aplikasi. Setiap revisi aplikasi yang dibuat sebaiknya nilainya ditambah dengan 1, sehingga secara program memiliki perbedaan versi antara yang terbaru dengan versi sebelumnya. Nilai ini tidak digunakan oleh sistem Android, tapi digunakan pengembang untuk mendapatkan versi aplikasi dan digunakan pula oleh Android Market untuk mengetahui bila terdapat versi aplikasi terbaru yang tersedia.

Atribut android:versionCode bisa dicari secara program dengan method `getPackageInfo()` dari class `PackageManager`:

```
PackageManager pm = getPackageManager();
try {
    //---mendapatkan info package---
    PackageInfo pi = pm.getPackageInfo("net.belajar2android.Peta",
0);
    //---menampilkan versioncode---
    Toast.makeText(getApplicationContext(), "VersionCode: "
+Integer.toString(pi.versionCode), Toast.LENGTH_SHORT).show();
} catch (NameNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

Atribut android:versionName berisi informasi versi yang disediakan bagi user. Nilai dari Atribut android:versionName seharusnya berformat `<major>.<minor>.<point>`. Jika aplikasi mengalami perubahan besar, maka `<major>` ditingkatkan menjadi 1. Untuk perubahan kecil, `<minor>` atau `<point>` dapat ditingkatkan menjadi 1.

Jika Anda berencana mempublikasikan aplikasi ke Android Market ([www.android.com/market/](http://www.android.com/market/)), maka file AndroidManifest.xml harus mempunyai atribut:

- android:versionCode (berada di element <manifest>)
- android:versionName (berada di element <manifest>)
- android:icon (berada di element <application>)
- android:label (berada di element <application>)

Atribut **android:label** menentukan nama aplikasi. Nama ini akan ditampilkan dibagian menu **Launcher**. Pada contoh di bawah, nama aplikasi adalah “**Dimana Saya**”.

Sebagai tambahan, jika aplikasi memerlukan versi SDK minimum, Anda dapat menentukannya di dalam file AndroidManifest.xml melalui element <uses-sdk>:

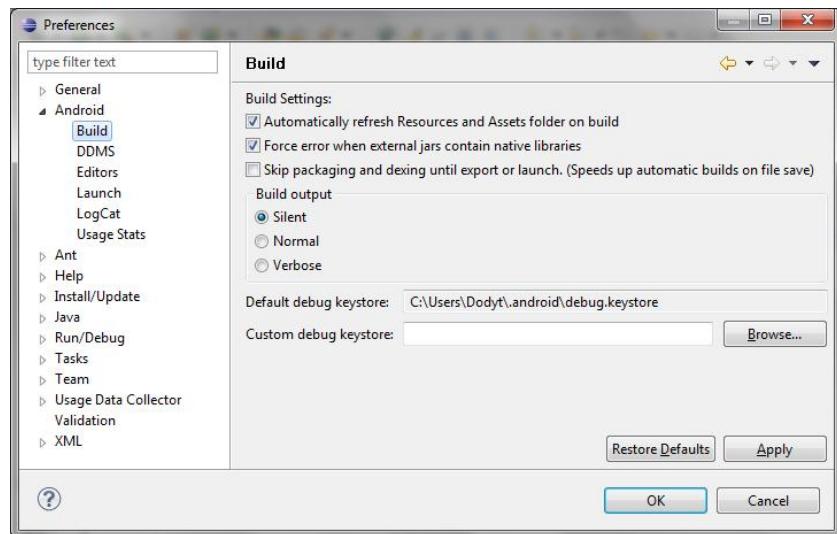
```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.belajar2android.Peta"
        android:versionCode="1"
        android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="Dimana Saya">
        <uses-library android:name="com.google.android.maps" />
        <activity android:name=".MainActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="7" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
        android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION" />
</manifest>
```

### 12.1.2 Tandatangan Digital Pada Aplikasi

Semua aplikasi Android harus diberi tandatangan digital (*digitally signed*) sebelum di-deploy ke perangkat (atau emulator). Tidak seperti perangkat bergerak lainnya yang perlu membeli *digital certificates* dari *certificate authority (CA)* untuk aplikasi. Pada Android kita dapat membangkitkan *signed certificate* sendiri dan menggunakannya pada aplikasi Android.

Saat menggunakan Eclipse untuk mengembangkan aplikasi Android, dan menjalankan aplikasi untuk men-deploynya ke emulator, Eclipse secara otomatis memberi sign pada aplikasi.

Anda dapat mem-verifikasi hal ini melalui menu **Windows|Preferences |Android|Build**.



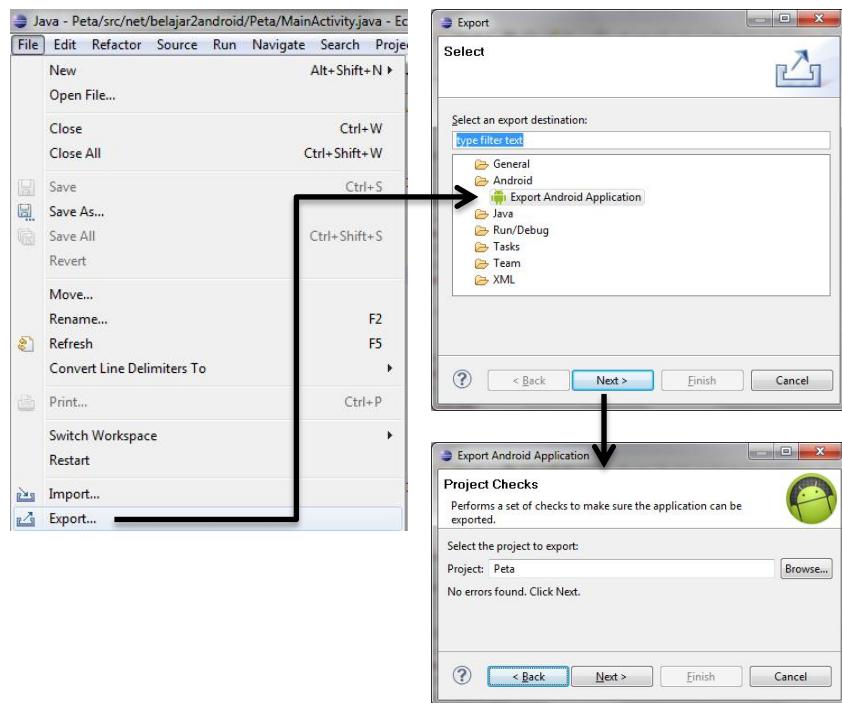
**Gambar 12.1 Memverifikasi project Android**

Secara default Eclipse menggunakan debug keystore (biasanya diberi nama “debug.keystore”) untuk men-sign aplikasi. Sebuah keystore biasanya dikenal sebagai *digital certificate*.

Aplikasi yang akan dipublikasi harus di-signed dengan certificate Anda sendiri. Anda dapat membangkitkan secara manual certificate Anda sendiri dengan menggunakan utility **keytool.exe** yang disediakan oleh Java SDK.

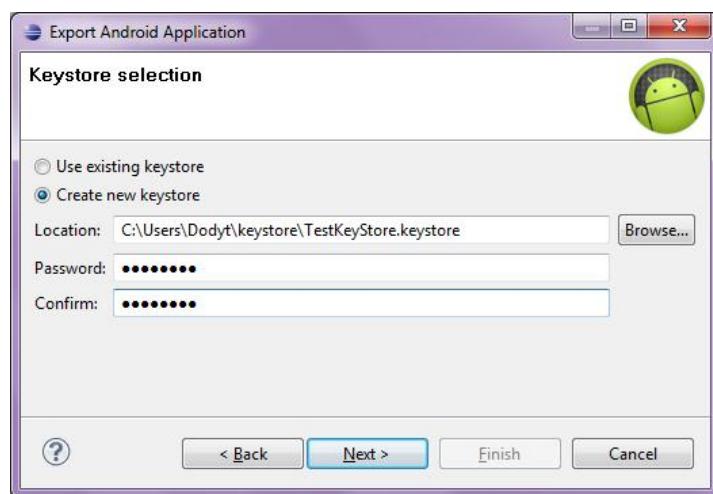
Berikut ini penggunaan Eclipse untuk mengekspor aplikasi Android dan men-sign aplikasi dengan pembangkitan certificate yang lebih baru.

Buka project **Peta** pada bab sebelumnya. Pilih project **Peta** pada Eclipse, kemudian pilih menu **File | Export...**, dalam Export dialog, perluas item **Android** dan pilih **Export Android Application**. Klik Next.



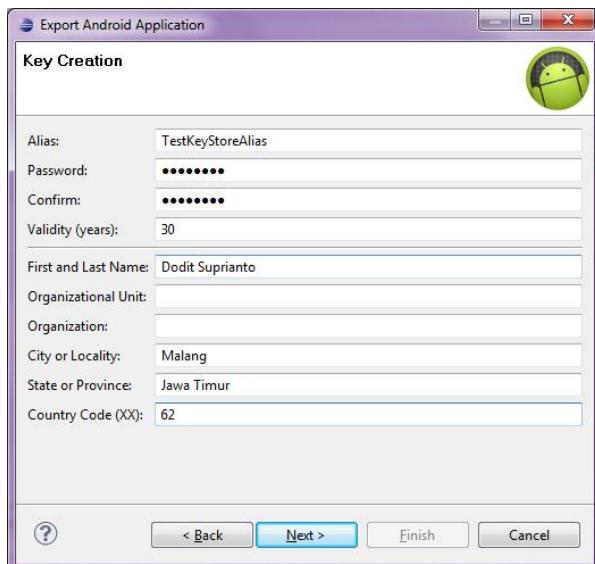
**Gambar 12.2 Eksport project Android**

Pilih option “**Create new keystore**” untuk menciptakan certificate (keystore) baru dan memberi sign pada aplikasi Anda. Masukkan nama file (**TestKeyStore.keystore**) beserta path-nya sebagai tempat penyimpanan keystore baru (atau klik browse), kemudian tuliskan password (contoh “password”) untuk melindungi keystore. Selanjutnya klik Next.



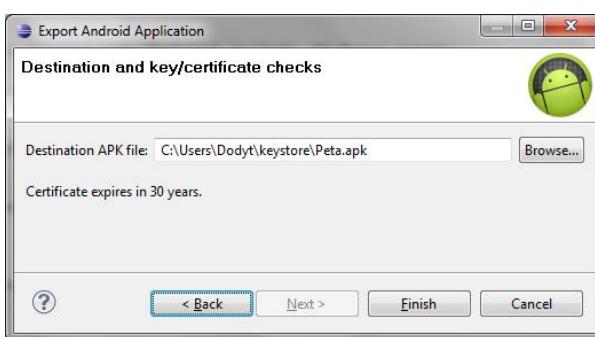
**Gambar 12.3 Membentuk folder dan file keystore**

Disini digunakan nama alias (nama file **TestKeyStoreAlias**) dan masukkan password untuk melindungi private key (contoh: “password”). Anda juga perlu menuliskan masa berlaku key-nya. Aplikasi harus di-sign dengan *cryptographic private key* yang masa berlakunya berakhir setelah 22 oktober 2033. Oleh karena itu masukkan angka lebih dari 2033 dikurangi tahun saat ini. Klik Next.



**Gambar 12.4 Membuat key alias**

Masukkan path tujuan dari file APK, kemudian klik Finish. File APK sekarang akan dibangkitkan.



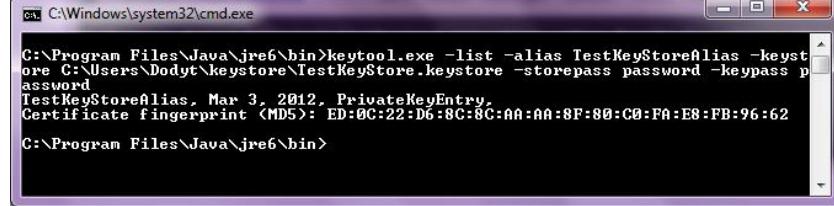
**Gambar 12.5 key/certificate untuk Peta.apk**

API key sebelumnya pada dasarnya mengikat debug.keystore yang digunakan untuk men-sign aplikasi Anda. Karena Anda sekarang membangkitkan keystore baru untuk men-sign aplikasi yang akan di-

*deployment*, maka Anda perlu mengaplikasikannya ke Google Maps API key lagi dengan menggunakan MD5 fingerprint keystore baru.

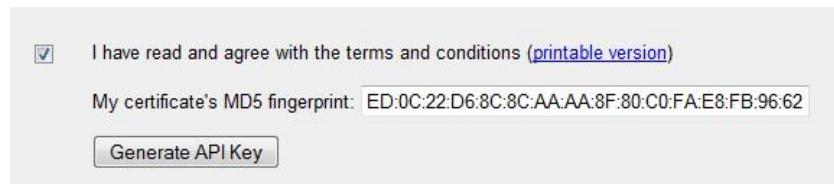
Untuk melakukannya, masuk ke command prompt dan tuliskan perintahnya dengan memanfaatkan keytool.exe.

```
C:\Program Files\Java\jre6\bin>keytool.exe -list -alias  
TestKeyStoreAlias -keystore  
C:\Users\Dodyt\keystore\TestKeyStore.keystore -storepass  
password -keypass password  
  
TestKeyStoreAlias, Mar 3, 2012, PrivateKeyEntry, Certificate  
fingerprint (MD5):  
ED:0C:22:D6:8C:8C:AA:AA:8F:80:C0:FA:E8:FB:96:62
```



**Gambar 12.6 Membangkitkan MD5 fingerprint dengan keytool.exe**

Masukkan kode fingerprint MD5 ke Google API (<http://code.google.com/android/maps-api-signup.html>):



**Gambar 12.7 Menuliskan MD5 Fingerprint ke Google API**

Sehingga diperoleh API key: 0v2ImAt504b6aPHg0uRsP-woRq0aMn3mdIbw2LA

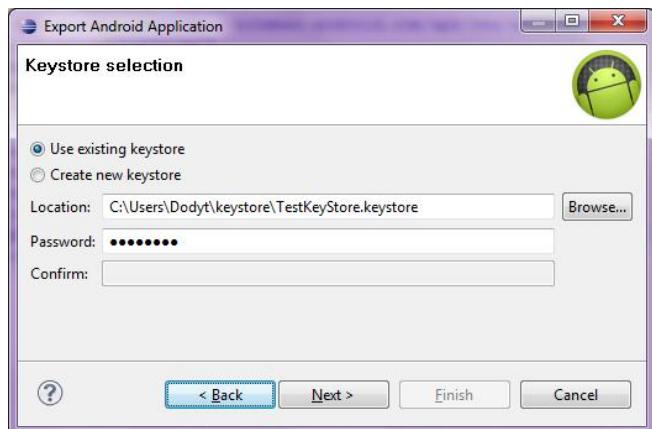


**Gambar 12.8 Android Maps API Key**

Tambahkan API key baru di atas ke dalam file main.xml

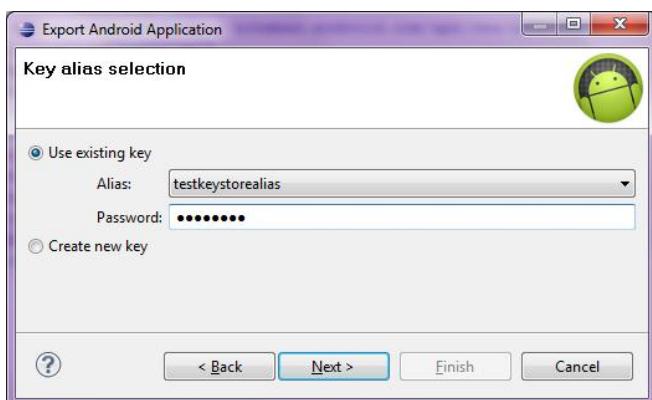
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <com.google.android.maps.MapView
        android:id="@+id/mapView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:enabled="true"
        android:clickable="true"
        android:apiKey="0v2ImAt504b6aPHg0uRsP-woRq0aMn3mdIbw2LA" />
</LinearLayout>
```

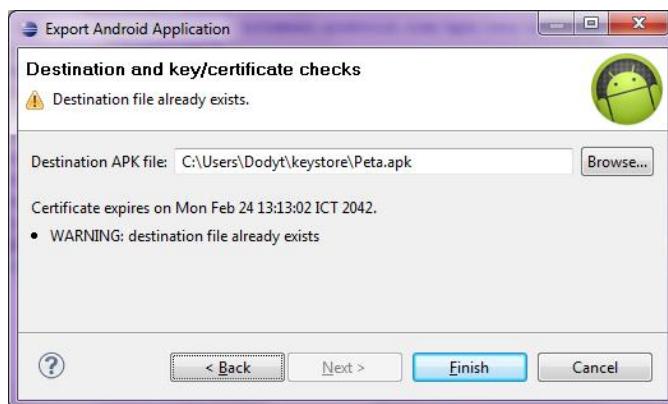
Selanjutnya Anda harus mengekspor aplikasi sekali lagi dan me-sign ulang seperti cara sebelumnya. Ketika diminta untuk memilih keystore, maka pilih "**Use existing keystore**" dan tuliskan password yang sama dengan sebelumnya (dalam kasus ini adalah "password") dan klik Next.



**Gambar 12.9 Mengekspor aplikasi sekali lagi**

Pilih “**Use existing key**” dan tuliskan password yang sama dengan sebelumnya (dalam kasus ini adalah “password”) dan klik Next.





**Gambar 12.10 Membuat keystore sekali lagi**

Klik Finish untuk membangkitkan file APK lagi.

Sekarang APK telah dibangkitkan dan telah mengandung Map API key yang terikat ke keystore baru.

## 12.2 Men-Deploy File APK

Setelah memiliki file APK yang telah ter-sign, selanjutnya adalah mendistribusikannya ke perangkat user. Ada tiga cara untuk men-deploy file APK, antara lain:

- Men-deploy secara manual dengan menggunakan tool adb.exe
- Menempatkan file apk web server (hosting)
- Mempublikasikan file apk melalui Android Market.

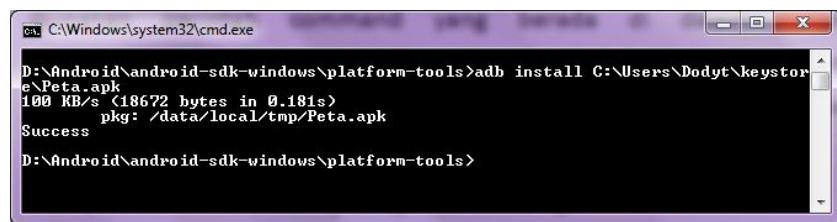
Disamping ketiga cara di atas, kita dapat menginstall aplikasi pada perangkat user melalui email, SD card dan lain-lain, selama masih bisa mentransfer file APK ke perangkat user, Anda dapat menginstall aplikasi.

### 12.2.1 Penggunaan Tool adb.exe

Anda dapat men-deploy aplikasi yang ter-signed ke emulator atau perangkat dengan menggunakan tool adb.exe (Android Debug Bridge) yang terletak dalam folder **platform-tools** dari SDK Android.

Gunakan perintah *command* yang berada di dalam folder "**<Android\_SDK>\platform-tools**". Untuk menginstall aplikasi ke emulator/device (dengan syarat emulator sedang aktif atau perangkat sedang terhubung):

```
adb install C:\Users\Dodyt\keystore\Peta.apk
```



**Gambar 12.11 Men-deploy aplikasi dengan tool ADB**

adb.exe adalah alat serbaguna yang bisa mengendalikan perangkat Android (emulator) yang terhubung dengan komputer.

Secara default, ketika menggunakan perintah adb, diasumsikan bahwa saat ini hanya terhubung dengan satu perangkat/emulator. Jika Anda memiliki hubungan lebih dari satu emulator/perangkat maka perintah adb akan mengembalikan pesan kesalahan seperti ini:

```
error: more than one device and emulator
```

Anda dapat melihat perangkat yang terhubung saat ini ke komputer melalui adb, seperti berikut:

```
D:\Android\android-sdk-windows\platform-tools>adb devices
List of devices attached
emulator-5554    device
```

Untuk menentukan perangkat/emulator mana yang akan dituju maka gunakan opsi perintah seperti berikut ini:

```
adb -s emulator-5554 install C:\Users\Dodyt\keystore\Peta.apk
```

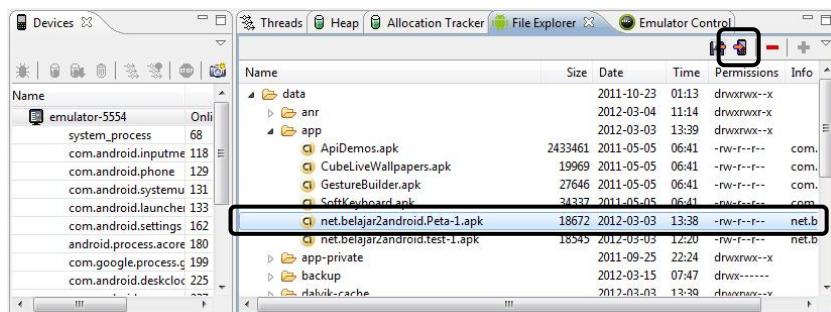
Jika Anda berusaha menginstall file APK ke perangkat yang sudah memiliki file APK, maka perangkat akan menampilkan pesan kesalahan:

```
Failure [INSTALL_FAILED_ALREADY_EXISTS]
```

Oleh karena itu Anda harus meng-uninstall aplikasi Peta terlebih dahulu, baru kemudian diinstall ulang. Untuk menghapus aplikasi maka pilih menu **Home|Settings|Applications|Manage applications** dan hapus aplikasinya.

**Catatan:** jika DDMS pada Eclipse suatu saat mengalami kerusakan dengan pesan kesalahan “Failed to parse the output of ‘adb version’”, maka buat perintah command di adb sebagai berikut : **adb kill-server**.

Cara lain untuk men-deploy aplikasi adalah dengan menggunakan DDMS. Melalui emulator (perangkat) yang terpilih, gunakan File Explorer dalam DDMS (folder /data/app) dan klik tombol “**Push a file onto the device**” untuk meng-copy file APK ke perangkat.



Gambar 12.12 Men-deploy file APK melalui DDMS

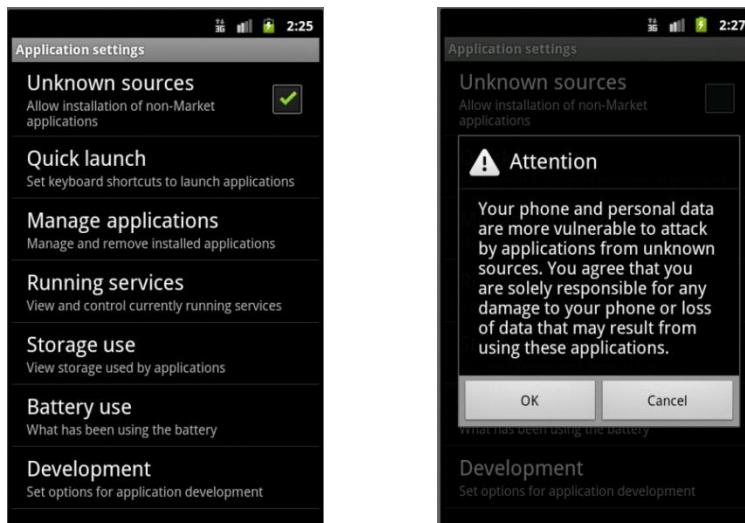
### 12.2.2 Penggunaan Web Server

Kita bisa menempatkan aplikasi Android ke server (hosting). Disini saya gunakan web server Apache dari instalasi XAMPP sebagai localhost. Caranya, Copy file Peta.apk yang sudah di-signed ke \xampp\htdocs\android\. Sebagai tambahan, ciptakan file HTML bernama Install.html:

```
<html>
<title>Aplikasi Peta</title>
<body>
Download Aplikasi Peta <a href="Peta.apk">disini</a>
</body>
</html>
```

Oleh karena aplikasi diinstall melalui web server maka kita perlu mengkonfigurasi Emulator android/perangkat agar bisa menerima aplikasi yang berasal dari selain Android Market.

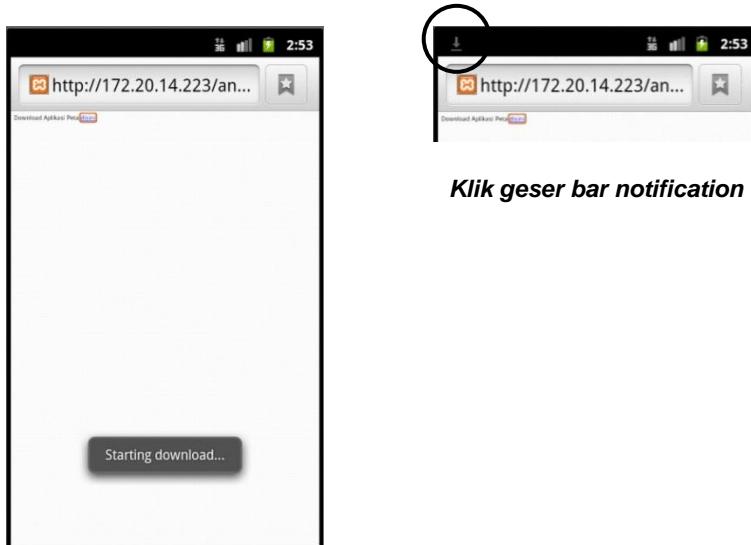
Dari menu pengaturan aplikasi, centang item “**Unknown sources**” sehingga muncul peringatan pesan kesalahan, kemudian klik OK.



**Gambar 12.13 Pengaturan agar bisa menerima file APK selain dari Android Market**

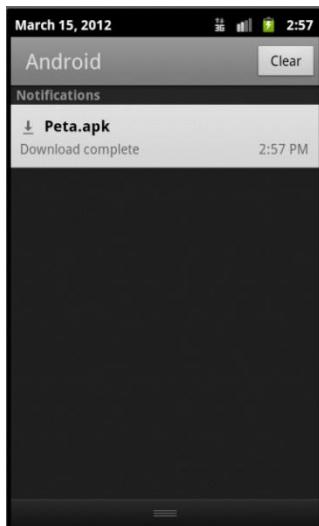
Untuk menginstall aplikasi Peta.apk dari web server yang berjalan pada komputer Anda, jalankan aplikasi **Browser** pada Emulator/perangkat Android dan navigasi URL yang menunjuk ke file APK tersebut (contoh: <http://172.20.14.232/android/install.html>).

Klik link “disini” yang akan men-download file APK ke perangkat. Drag (klik geser) bar notification ke bawah untuk membuka status download.



*Proses download*

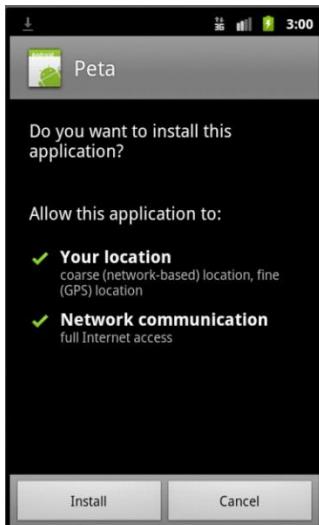
*Klik geser bar notification*



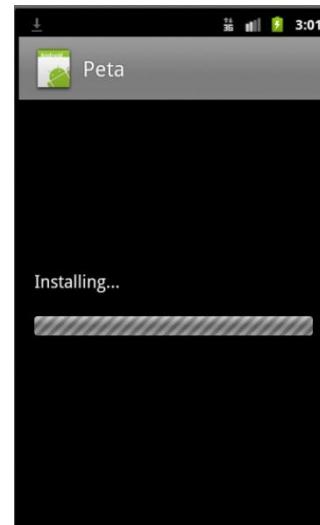
*Klik file Peta.apk*



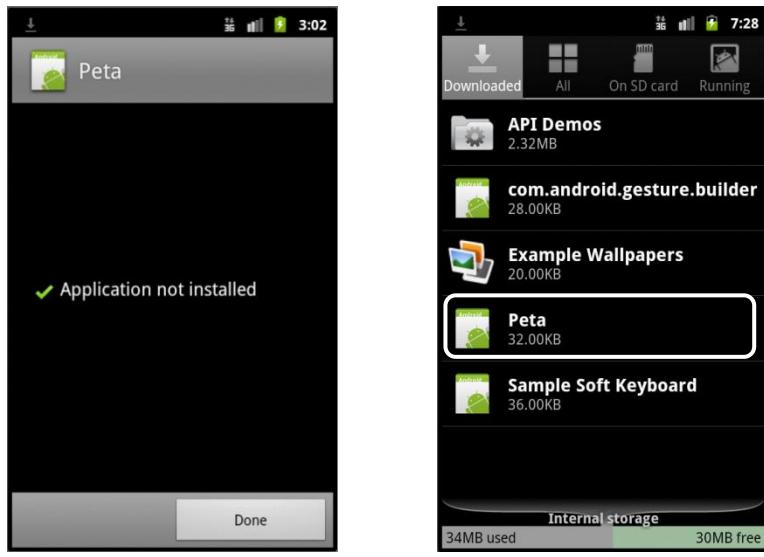
*File Peta.apk akan ditimpas jika sudah ada, klik OK*



*Klik Install*



*Proses instalasi*



*Proses install selesai*

*Aplikasi Peta sudah terinstall*

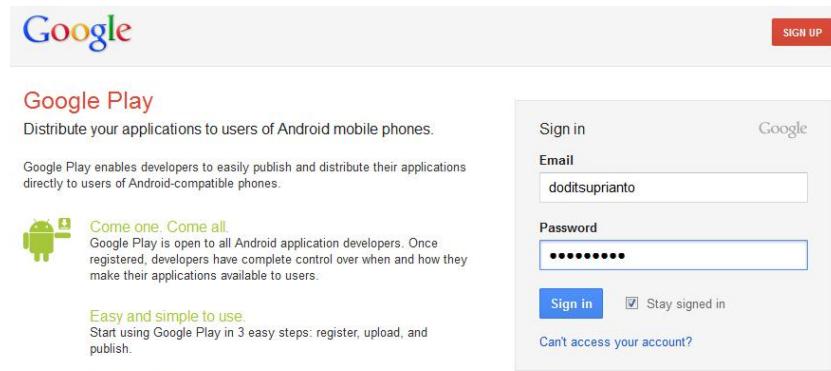
*Gambar 12.14 Proses instalasi file APK dari web server*

### 12.2.3 Publikasi Ke Android Market

Cara terbaik untuk mempublikasikan aplikasi Android adalah dengan menempatkannya ke Android Market. Layanan hosting Google membuat user lebih mudah menemukan dan mendownload aplikasi-aplikasi pada perangkat Android mereka.

Android Market cocok bagi pengembang yang menekuni aplikasi Android secara serius dan ada keinginan untuk mendistribusikan aplikasi Android secara komersial.

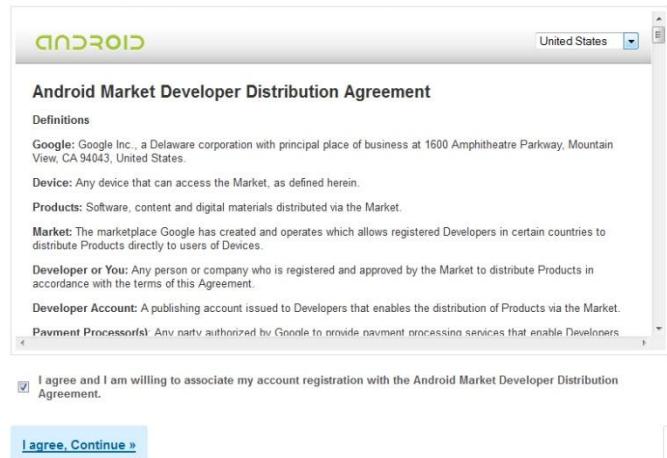
Sebelum mempublikasikan aplikasi ke Android Market, terlebih dahulu perlu diciptakan profile pengembang pada <https://play.google.com/apps/publish/signup>.



**Gambar 12.15 Login ke Google**

Hal ini memerlukan akun Google (seperti akun Gmail) agar dapat masuk ke Android Market dan menciptakan profile pengembang. Klik **I agree** **Continue** setelah menuliskan informasi yang dibutuhkan.

The terms and conditions vary according to the country where your billing address is located.  
Please choose your billing country below to view the appropriate terms and conditions.



**Gambar 12.16 Pernyataan persetujuan mengenai Android Market**

Tapi sayangnya untuk mempublikasikan aplikasi ke Android Market, pengembang dikenakan biaya pendaftaran oleh Google sebesar U.S.\$25 (dibayar sekali saja), sehingga saya tidak bisa melanjutkan proses publikasi ke Android Market, tapi saya yakin hal tersebut tidak sulit untuk dilakukan. Pembaca dapat melaksanakan sendiri sesuai tahapan-tahapan yang diberikan oleh website Android Market.

The screenshot shows the 'Getting Started' section of the Google Play Developer Console. At the top, there's a navigation bar with the email 'meozit@gmail.com' and links for 'Home', 'Help', 'Android.com', and 'Sign out'. Below the navigation is the 'Google play | ANDROID DEVELOPER CONSOLE' logo. The main content area has a green header 'Getting Started' and a sub-header 'Before you can publish software on Google Play, you must do three things:'. A bulleted list follows:

- Create a developer profile
- Agree to the [Android Market Developer Distribution Agreement](#)
- Pay a registration fee (\$25.00) with your credit card (using Google Checkout)

Below this is a 'Listing Details' section with a sub-header 'Your developer profile will determine how you appear to customers in Google Play'. It contains fields for 'Developer Name' (set to 'Dodit Suprianto'), 'Email Address' ('doditsuprianto@gmail.com'), 'Website URL' ('http://doditsuprianto.web'), 'Phone Number' ('+62-81252080603'), and 'Email Updates' (checkbox checked). The 'Developer Name' field includes a note: 'Will appear to users under the name of your application'.

At the bottom of this section is a blue 'Continue >' button.

The screenshot shows the 'Register as a developer' section of the Google Play Developer Console. At the top, there's a navigation bar with the email 'doditsuprianto@gmail.com' and links for 'Home', 'Help', 'Android.com', and 'Sign out'. Below the navigation is the 'Google play | ANDROID DEVELOPER CONSOLE' logo. The main content area has a green header 'Register as a developer' and a sub-header 'Registration fee: \$25.00'. It states: 'Your registration fee enables you to publish software in the market. The name and billing address used to register will bind you to the [Android Market Developer Distribution Agreement](#). So make sure you double check!'. Below this is a 'Pay your registration fee with' section featuring the 'Google Checkout' logo and the text 'Fast checkout through Google'. At the bottom are two buttons: a blue 'Continue >' button on the left and a white 'Back to change profile' button on the right.

**Gambar 12.17 Mengisi profile pengembang aplikasi Android**

# TENTANG PENULIS



**Dodit Suprianto,**

Email: [doditsuprianto@gmail.com](mailto:doditsuprianto@gmail.com)

Merupakan alumni Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) Malang dan saat ini menempuh pendidikan S2 di Universitas Brawijaya, Teknik Elektro (Sistem Teknologi Komunikasi & Informasi). Penulis pernah bekerja di beberapa perusahaan swasta nasional sebagai pengembang sistem informasi. Namun saat ini lebih aktif mengajar dibeberapa institusi pendidikan di kota Malang, sebagai penulis dan sebagai wiraswasta.



**Rini Agustina, S.Kom, M.Pd,**

Email: [ryf4ny@yahoo.com](mailto:ryf4ny@yahoo.com)

Meraih gelar S.Kom di STIKI Malang pada tahun 2001, setelah 3 tahun bekerja di Industri beralih ke dunia pendidikan pada tahun 2005 dengan mengabdi di SMK Negeri 8 Malang, pada 2007 mendapatkan beasiswa BPKN untuk melanjutkan ke S2 di Universitas Negeri Malang Jurusan Magister Pendidikan Kejuruan konsentrasi Teknik Elektro, lulus pada tahun 2009, kemudian mendapat kesempatan untuk mengajar di Universitas Kanjuruhan Malang dan menjadi Dosen tetap yayasan dan sekarang menjabat sebagai Kepala Manajement Information System (MIS) di Universitas Kanjuruhan Malang (UNIKAMA).