

1

PENGENALAN ANDROID OS

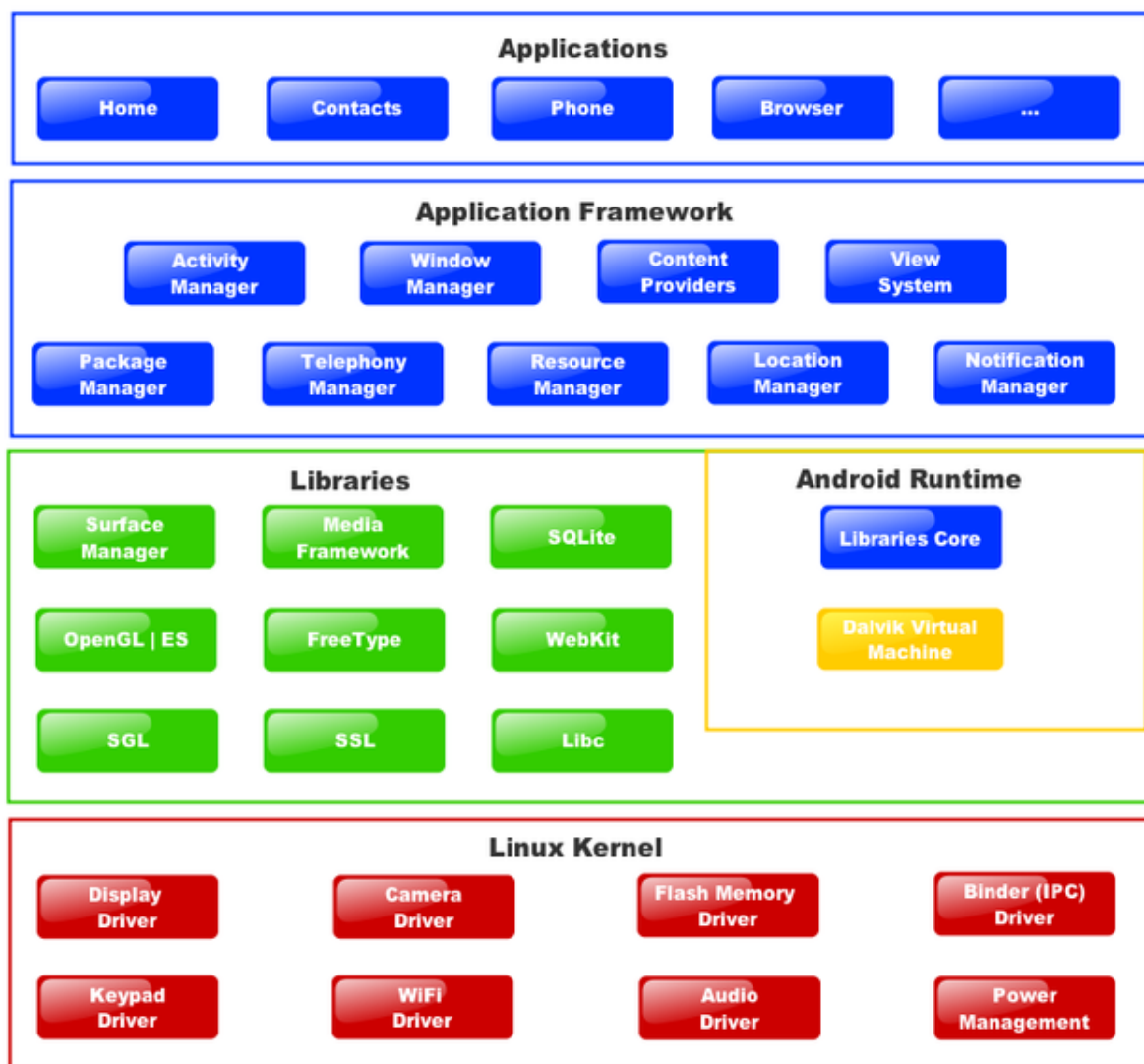
Android adalah sebuah sistem operasi untuk berbagai perangkat mobile seperti handphone, netbook, dan komputer tablet. Sistem operasi ini dikembangkan oleh Google dengan berbasis kernel Linux versi 2.6 dan berbagai perangkat lunak dari GNU yang bersifat Open Source. Pada awalnya, Android dikembangkan oleh perusahaan Android Inc. namun kemudian perusahaan tersebut diakuisisi oleh Google sehingga menjadi produk Google. Sekarang ini pengembangan Android ditentukan oleh sebuah konsorsium bernama Open Handset Alliance (OHA) yang terdiri atas berbagai vendor perangkat mobile, komputer, dan telekomunikasi seperti Intel, Nvidia, Google, Samsung, Sprint, T-Mobile, Motorola, LG, Sony Ericsson, Toshiba, Vodafone, serta masih banyak yang lain dan anggotanya terus bertambah.

Karena Android bersifat Open Source dan mempunyai lisensi Apache yang sangat terbuka dan bebas, maka Android menjadi sistem operasi yang sangat populer bagi berbagai produsen perangkat mobile. Berdasarkan laporan dari NPD Group, penjualan perangkat smartphone berbasis Android berada di peringkat kedua di seluruh Amerika pada kuartal pertama tahun 2010. Laporan dan Nielson juga menunjukkan bahwa Android berada di posisi keempat dengan pangsa pasar sebesar 9% untuk smartphone di kuartal yang sama.

Salah satu keunggulan Android adalah adanya komunitas para developer dan programmer yang luas untuk mengembangkan berbagai aplikasi yang berjalan di

perangkat berbasis Android sehingga mampu memperluas fitur dan kemampuan perangkat tersebut. Saat ini sudah terdapat lebih dari 70000 aplikasi Android sehingga berada di peringkat kedua di dunia. Banyaknya aplikasi yang tersedia di Android dimungkinkan karena pengembangan di Android relatif mudah dan cepat karena menggunakan bahasa pemrograman Java dengan dukungan library Java yang dikembangkan oleh Google.

Komponen sistem operasi Android bisa digambarkan dalam bagan berikut:



Software stack untuk Android OS terdiri atas berbagai aplikasi Java yang berjalan di atas framework Java berorientasi obyek, yang kemudian berjalan di atas core library Java dari Dalvik Virtual Machine (DVM) melalui JIT compiler. Sebagian library yang kritis ditulis dengan bahasa C demi kinerja seperti Surface Manager untuk dukungan touchpad/touchscreen, OpenCore Media Framework, SQLite untuk database, OpenGL ES untuk grafis 3D, FreeType untuk dukungan font, WebKit layout engine untuk user interface, SGL graphics engine untuk grafis 2D, SSL untuk koneksi yang aman, dan Bionic libc. Saat ini Android OS terdiri dari 12 juta baris kode termasuk 3 juta baris kode XML, 2.8 juta baris kode dengan bahasa C, 2.1 juta baris kode dengan bahasa Java, dan 1.75 juta baris kode dengan bahasa C++. Pengembangan Android masih terus berlanjut dengan menambahkan berbagai fitur yang menarik, sehingga jumlah baris kode untuk Android akan terus bertambah dengan keluarnya versi-versi baru.

Ada beberapa istilah yang perlu dipahami dalam Android sehingga kita mampu membangun aplikasi Android dengan baik dan benar.

Resource: Segala hal yang digunakan dalam aplikasi Android yang kita kembangkan seperti teks, gambar, video, audio, xml, layout, dan berbagai sumber lain yang digunakan dalam kode sumber aplikasi Android. Semua resource yang dapat diakses di suatu aplikasi secara otomatis akan didaftarkan di dalam class khusus bernama R sehingga kita mampu mengaksesnya melalui R.

Activity: Merepresentasikan satu layar di Android dalam aplikasi. Pada intinya, sebuah activity mengendalikan semua kegiatan yang terjadi dalam satu layar yang tampil di perangkat berbasis Android. Dalam paradigma MVC (Model-

View-Controller), activity mempunyai peran sebagai controller. Dalam suatu aplikasi Android yang rumit, bisa terdapat banyak activity untuk berbagai fungsi yang disediakan aplikasi tersebut.

View: Komponen user interface (UI) yang terdapat pada satu activity atau layar. Pengguna berinteraksi dengan layar melalui satu atau lebih view yang terdapat pada satu activity/layar. Suatu activity berisikan satu atau lebih view dan mengendalikan semua view dalam layar tersebut.

Intent: Kode yang bertanggung jawab untuk melakukan suatu proses dalam aplikasi Android, termasuk pergantian activity/layar atau pemanggilan aplikasi yang lain untuk interaksi antar aplikasi.

AndroidManifest.xml: File yang berisikan deskripsi dan konfigurasi umum mengenai aplikasi Android yang kita buat. File ini bisa diibaratkan sebagai deployment descriptor untuk memberitahu kepada Android bagaimana cara memasang aplikasi ini di perangkat berbasis Android.

Bisa dibilang bahwa unit paling kecil dalam suatu aplikasi Android adalah Activity. Satu Activity merepresentasikan satu layar, dan dalam satu activity bisa terdapat beberapa view yang memungkinkan pengguna untuk berinteraksi dengan activity tersebut. View bisa menggunakan berbagai Resource. Jika suatu activity ingin mengakses activity, aplikasi, atau pelayanan yang lain, maka activity dapat melakukannya dengan menggunakan Intent. Aplikasi bisa terdiri atas beberapa activity dan intent, dan semuanya terdeskripsi dalam AndroidManifest.xml.

2

INSTALASI ANDROID SDK

Android Software Development Kit atau Android SDK adalah sekumpulan tool dan platform yang memungkinkan kita untuk mengembangkan berbagai aplikasi yang jalan di Android. Dengan Android SDK dan bahasa pemrograman Java, berbagai aplikasi canggih dapat dibangun dengan cepat dan relatif mudah yang menggunakan semua fitur yang ada di Android. Bahkan berbagai Add-On dapat ditambahkan untuk memperluas kemampuan Android seperti Google API yang memungkinkan kita untuk membangun berbagai aplikasi geografis berdasarkan Google Map.

Syarat-syarat minimal yang dibutuhkan untuk instalasi Android SDK beserta toolsnya adalah:

- Komputer dengan Microsoft Windows, Linux, atau MacOS X
- Java Development Kit (JDK) versi 6
- Android SDK (Software Development Kit) starter package
- Android SDK platform sesuai versi Android yang ingin digunakan
- Eclipse IDE bila ingin lebih mudah dan cepat dalam pembangunan aplikasi Android
- ADT (Android Development Kit) Plugin untuk Eclipse
- Documentation dan samples untuk mempelajari Android API
- USB Driver untuk Windows agar dapat melakukan debugging dengan Android device sungguhan

- Berbagai Add-On untuk Android untuk menambah fitur, seperti Google APIs

Prosedur instalasi Android SDK beserta toolnya adalah sebagai berikut.

INSTALASI JDK DAN ECLIPSE IDE:

1. Lakukan instalasi JDK untuk dukungan Java. Disarankan untuk menggunakan JDK yang terbaru yaitu versi 6 Update 26.
2. Untuk mempermudah dan mempercepat pengembangan aplikasi Android, disarankan untuk menggunakan Eclipse IDE. Eclipse dapat didownload dari <http://www.eclipse.org/downloads/>. Disarankan untuk menggunakan versi Eclipse Helios SR2 for Java Developers.
3. Untuk melakukan instalasi Eclipse, cukup melakukan ekstraksi file zip Eclipse ke suatu folder. Di dalam folder tersebut terdapat file eclipse.exe yang bisa langsung dijalankan.

INSTALASI ANDROID SDK DAN KONFIGURASI:

1. Download Android SDK dari <http://developer.android.com/sdk/index.html>, kemudian lakukan ekstraksi file zip ke suatu folder. Android SDK ini hanya berisikan tools saja dan belum siap digunakan. Yang masih dibutuhkan adalah Android Platform yang sesuai dengan versi Android yang ingin digunakan. Android Platform bisa didapatkan langsung dari server Google melalui download dengan tool Android SDK and AVD Manager yang ada di Android SDK starter package.
2. Berikutnya kita perlu menambahkan lokasi instalasi JDK dan Android SDK di bagian PATH sehingga kita dapat secara langsung mengaksesnya.

Ikuti berbagai langkah berikut untuk menambahkannya di Windows 7:

- a) Jalankan Windows Explorer
- b) Klik Kanan pada Computer dan pilih Properties
- c) Pilih Advanced System Properties
- d) Pada tab Advanced, klik pada tombol Environment Variables
- e) Pada System variables di bagian bawah dialog, klik ganda pada variable Path untuk melakukan edit terhadapnya
- f) Pada paling akhir, tambahkan titik koma (;) dan kemudian path lengkap untuk lokasi folder bin yang ada pada JDK. Sebagai contoh, tambahkan “;C:\Program Files (x86)\Java\jdk1.6.0_26\bin” di paling akhir. Hati-hati jangan sampai menghapus yang sudah ada.
- g) Setelah menambahkan lokasi JDK, tambahkan pula titik koma (;) lagi dan kemudian lokasi folder tools yang ada pada Android SDK. Sebagai contoh, tambahkan “;C:\Android\android-sdk\tools;C:\Android\android-sdk\platform-tools” di paling akhir. Hati-hati jangan sampai menghapus yang sudah ada.
- h) Klik OK terus sampai dialog System Properties berhasil ditutup semua.
- i) Untuk konfirmasi bahwa setting Path sudah benar, jalankan command prompt dengan klik pada Start, kemudian ketik cmd dan tekan Enter. Ketik perintah “java -version” dan “adb version” dan pastikan keduanya bisa jalan. Jika ada yang tidak jalan, maka terjadi kesalahan dalam menambahkan Path. Silakan dibetulkan kesalahannya bila tidak jalan.

INSTALASI ADT PLUGIN UNTUK ECLIPSE:

1. Download ADT Plugin untuk Eclipse supaya kita dapat menggunakan Eclipse untuk mengakses Android SDK. ADT Plugin dapat didownload

dari <http://developer.android.com/sdk/eclipse-adt.html>. File yang didownload adalah berupa file zip biasa. Jangan melakukan ekstraksi terhadap file zip tersebut, namun dibiarkan saja.

2. Jalankan Eclipse IDE dengan melakukan klik ganda pada file eclipse.exe yang ada pada folder eclipse hasil ekstraksi sebelumnya.
3. Tentukan lokasi folder untuk tempat Eclipse bekerja. Istilahnya adalah workspace, dan sebagai contoh kita bisa menggunakan folder “C:\Android\workspace” sebagai workspace untuk Eclipse. Workspace Eclipse digunakan untuk menyimpan berbagai project yang dibuat di Eclipse. Klik OK untuk lanjut.
4. Welcome screen di Eclipse dapat ditutup. Berikut kita perlu melakukan instalasi ADT Plugin yang telah didownload sebelumnya. Instalasi Plugin di Eclipse dapat dilakukan melalui menu Help, kemudian pilih Install New Software.
5. Pada dialog yang muncul, klik pada tombol Add untuk mendaftarkan file ADT Plugin hasil download.
6. Pada dialog yang muncul berikutnya, klik pada tombol Archive. Pilih file ADT-10.0.1.zip yang telah didownload sebelumnya. Klik Open dan kemudian OK.
7. Berikan tanda centang pada Android DDMS dan Android Development Tools untuk melakukan instalasi lengkap plugin tersebut di Eclipse. Klik pada Next dan pilih Accept waktu halaman lisensi muncul. Jika ditanyakan konfirmasi instalasi, klik pada OK untuk lanjut hingga instalasi selesai.
8. Setelah selesai melakukan instalasi ADT Plugin di Eclipse, lakukan restart pada Eclipse.

KONFIGURASI ECLIPSE UNTUK ANDROID SDK:

1. Setelah restart Eclipse selesai, klik pada menu Window dan pilih Preferences.
2. Pada dialog Preferences yang tampil, klik pada Android di daftar konfigurasi.
3. Tentukan lokasi instalasi Android SDK dengan klik pada tombol Browse dan kemudian memilihnya, misalnya "C:\Android\android-sdk". Klik pada tombol Apply.
4. Pada dialog Preferences, lakukan ekspansi pada Java, dan kemudian pilih Installed JREs.
5. Untuk mendaftarkan JDK 6 pada Eclipse, klik pada tombol Add. Pada dialog yang muncul, pilih Standard VM dan klik tombol Next.
6. Klik pada tombol Directory dan pilih folder instalasi JDK 6, misalnya "C:\Program Files (x86)\Java\jdk1.6.0_26". Klik pada tombol Finish.
7. Berikan tanda centang pada pilihan jdk1.6.0_26. Kemudian pilih jre dan klik pada tombol Remove untuk menghapusnya sehingga hanya tersisa satu saja yaitu jdk1.6.0_26. Klik pada tombol OK setelah selesai.
8. Dari menu Window, klik pada Android SDK and AVD Manager.
9. Pada dialog yang tampil, klik pada Available Packages untuk melakukan koneksi Internet ke server Google guna melakukan download Android Platform, Samples, Documentation, Google API, dan USB Driver. Pada daftar situs repository yang tampil, klik pada tanda > di sebelah kiri kotak centang untuk menampilkan daftar berbagai file yang dapat didownload dari repository Android di server Google.
10. Sesuai dengan versi Android yang ingin digunakan, berikan tanda centang pada Documentation for Android SDK, SDK Platform Android, Samples for SDK, Google APIs, dan USB Driver Package. Versi terbaru adalah Android 3.2 (API 13) namun kebanyakan Android device masih

menggunakan Android 2.1 (API 7) atau Android 2.2 (API 8).

11. Klik pada tombol Install untuk melakukan download dan instalasi file-file yang telah dipilih. Beberapa file berukuran besar (50MB) dan tergantung kecepatan koneksi Internet, akan butuh waktu yang cukup lama untuk mendapatkan semuanya.
12. Setelah semua instalasi selesai, lakukan konfirmasi bahwa semua file sudah diinstall. Konfirmasi bisa dilakukan dengan klik pada Installed Packages.

PEMBUATAN ANDROID VIRTUAL DEVICE:

1. Setelah semua file sudah terkonfirmasi, maka klik pada Virtual Devices. Kita perlu membuat sebuah Android device secara virtual melalui emulator yang dapat digunakan untuk menjalankan aplikasi Android yang kita buat di Eclipse.
2. Pada dialog Virtual Devices, klik tombol New untuk menambahkan virtual device baru. Berikan nama untuk virtual device baru tersebut, misalnya "MyAndroid". Pada Target, pilih platform Android yang ingin digunakan untuk pengembangan, misalnya "Android 2.1-update 1 – API Level 7" untuk device dengan Android versi 2.1. Isi ukuran SD card untuk memory di device, misalnya 32 untuk 32MB SD card. Kita juga bisa menentukan skin dan resolusi ukuran layar. Terakhir, klik pada Create AVD.
3. Setelah MyAndroid berhasil dibuat, pilihlah Android Virtual Device tersebut dari daftar. Lakukan startup dengan mengklik pada tombol Start. Kemudian klik pada tombol Launch.

- Setelah beberapa saat, mestinya emulator Android akan tampil di layar. Proses boot Android memang cukup lama. Setelah proses boot selesai, maka akan tampil seperti di bawah ini:



- Emulator Android siap untuk digunakan.

Demikianlah proses instalasi JDK 6, Eclipse IDE, Android SDK, dan Android Platform untuk dapat mulai membangun aplikasi Android dengan cepat, mudah, dan nyaman.

Kita bisa melakukan simulasi pembuatan call telpon dan juga pengiriman SMS dengan emulator AVD. Kita bahkan dapat menjalankan lebih dari satu emulator AVD dan melakukan komunikasi call atau SMS antar AVD.

Pertama, untuk melakukan simulasi call atau pengiriman SMS ke satu emulator AVD, kita perlu melakukan koneksi dengan telnet ke localhost (komputer sendiri) melalui port yang digunakan emulator AVD. Secara default, AVD yang pertama menggunakan port nomor 5554. Nomor port tersebut bisa dilihat di sisi kiri atas pada jendela AVD yang sedang jalan.

SIMULASI CALL KE EMULATOR AVD:

1. Pastikan bahwa emulator AVD MyAndroid yang sebelumnya telah dibuat masih jalan. Jika belum, dari Eclipse pilih menu Window dan klik pada Android SDK and AVD Manager. Pilih MyAndroid dan klik tombol Start untuk melakukan proses booting AVD tersebut.
2. Untuk melakukan simulasi call, kita harus melakukan koneksi dengan telnet ke localhost di port 5554. Jalankan command prompt dengan klik pada Start, kemudian ketik “cmd” dan tekan Enter. Command prompt akan jalan dimana kita bisa mengetik berbagai perintah.
3. Di command prompt, ketik perintah berikut untuk memastikan bahwa AVD sudah jalan dan juga untuk melihat nomor portnya:

```
adb devices
```

4. Pastikan bahwa emulator kita sudah jalan dan menggunakan port 5554.
5. Berikut kita melakukan koneksi dengan telnet sebagai berikut:

```
telnet localhost 5554
```

6. Setelah berhasil melakukan koneksi dengan telnet, ketik perintah “help” untuk melihat daftar perintah yang bisa kita berikan melalui console.
7. Ketik perintah berikut untuk melakukan simulasi call ke emulator AVD MyAndroid:

```
gsm call 0123456789
```

8. Perhatikan bahwa ada telpon yang masuk di emulator AVD MyAndroid.

Silakan menerimanya atau menolaknya. Karena ini hanya emulasi, tentu saja kita tidak bisa melakukan percakapan telpon sesungguhnya.

SIMULASI PENGIRIMAN SMS KE EMULATOR AVD:

1. Pastikan bahwa emulator AVD MyAndroid yang sebelumnya telah dibuat masih jalan. Jika belum, dari Eclipse pilih menu Window dan klik pada Android SDK and AVD Manager. Pilih MyAndroid dan klik tombol Start untuk melakukan proses booting AVD tersebut.
2. Untuk melakukan simulasi call, kita harus melakukan koneksi dengan telnet ke localhost di port 5554. Jalankan command prompt dengan klik pada Start, kemudian ketik “cmd” dan tekan Enter. Command prompt akan jalan dimana kita bisa mengetik berbagai perintah.
3. Di command prompt, ketik perintah berikut untuk memastikan bahwa AVD sudah jalan dan juga untuk melihat nomor portnya:

```
adb devices
```

4. Pastikan bahwa emulator kita sudah jalan dan menggunakan port 5554.
 5. Berikut kita melakukan koneksi dengan telnet sebagai berikut:
- ```
telnet localhost 5554
```
6. Setelah berhasil melakukan koneksi dengan telnet, ketik perintah “help” untuk melihat daftar perintah yang bisa kita berikan melalui console.
  7. Ketik perintah berikut untuk melakukan simulasi call ke emulator AVD MyAndroid:

```
sms send 0123456789 Hello World, MyAndroid!
```

8. Perhatikan bahwa ada sms yang masuk di emulator AVD MyAndroid. SMS baru akan tampil di bagian atas layar secara sekilas.
9. Untuk membacanya, jalankan aplikasi Message yang merupakan bawaan Android. Jika ingin lebih bagus, kita bisa melakukan instalasi aplikasi

SMS sendiri dari pihak ketiga (third party) seperti ChompSMS.

## **SIMULASI KOMUNIKASI ANTAR EMULATOR AVD:**

1. Untuk melakukan simulasi komunikasi antar AVD, kita perlu membuat AVD baru selain MyAndroid yang sudah ada. Di Eclipse, pilih menu Window dan klik Android SDK and AVD Manager untuk membuat AVD baru.
2. Pada dialog yang tampil, pastikan kita ada di Virtual Devices. Seharusnya sudah ada satu AVD bernama MyAndroid yang telah kita buat sebelumnya. Sekarang kita klik tombol New untuk membuat satu lagi yang baru.
3. Beri nama pada AVD yang baru, misalnya “YourAndroid”. Pilih Android 2.1 untuk Target, dan tentukan ukuran SD card misalnya 16MB. Klik pada tombol Create AVD.
4. Pastikan bahwa AVD YourAndroid sudah terdaftar bersamaan dengan MyAndroid. Klik pada YourAndroid dan kemudian klik tombol Start untuk menjalankannya.
5. Setelah proses boot untuk YourAndroid selesai, maka kita memiliki dua emulator AVD yang jalan secara bersamaan. Perhatikan bahwa YourAndroid menggunakan port 5556 sedangkan MyAndroid masih di port 5554.
6. Jalankan command prompt dengan klik pada Start, kemudian ketik “cmd” dan tekan Enter. Command prompt akan jalan dimana kita bisa mengetik berbagai perintah.
7. Di command prompt, ketik perintah berikut untuk memastikan bahwa AVD sudah jalan dan juga untuk melihat nomor portnya:

```
adb devices
```

8. Pastikan bahwa memang ada dua emulator AVD yang jalan, yaitu di port 5554 dan 5556.
9. Untuk melakukan simulasi telpon di MyAndroid ke YourAndroid, jalankan aplikasi Phone di MyAndroid. Untuk menelpon YourAndroid, gunakan nomor port YourAndroid yaitu 5556 sebagai nomor telpon yang dituju. Setelah melakukan Dial, perhatikan bahwa ada telpon masuk di YourAndroid dari MyAndroid.
10. Untuk melakukan simulasi pengiriman sms di MyAndroid ke YourAndroid, jalankan aplikasi Message di MyAndroid. Klik New Message dan buat SMS ke 5556. Begitu Send dilakukan, maka YourAndroid akan menerima SMS dari MyAndroid, dan dengan aplikasi Message maka YourAndroid dapat melakukan reply terhadap SMS tersebut.

Demikianlah proses instalasi, konfigurasi, dan penggunaan Android SDK dan Eclipse IDE dengan ADT Plugin.

Berikut kita mulai membangun aplikasi Android kita yang pertama dan menggunakan berbagai fitur menarik dan berguna pada Android SDK dan Eclipse IDE.

### 3

## PEMBUATAN APLIKASI HELLO WORLD

Aplikasi Hello World sudah merupakan standar tutorial dasar untuk proses pembelajaran bahasa pemrograman atau framework baru. Meskipun aplikasi yang akan kita buat ini memang sangat sederhana, tujuan utama dari pembuatan aplikasi Hello World adalah supaya kita mengenal proses dasar pembuatan aplikasi untuk Android serta mengenal pula berbagai fitur di Eclipse IDE dan Android SDK sehingga kita bisa produktif dalam melakukan pengembangan berbagai aplikasi Android yang lebih rumit dan canggih.

### PEMBUATAN PROJECT ANDROID:

1. Di Eclipse menu, pilih File dan kemudian klik pada New, dan kemudian Project.
2. Lakukan ekspansi pada Android, dan pilih Android Project untuk membuat project Android kita yang pertama. Klik Next.
3. Pada dialog yang tampil, tentukan Project Name misalnya “HelloAndroid”. Tentukan Application Name misalnya “Hello, Android”. Isi Package Name dengan misalnya “com.inixindo.workshop”. Pastikan bahwa Create Activity telah tercentang, dan berikan nama “HelloAndroid” untuk activity yang pertama. Berikan pula tanda centang pada platform Android yang ingin digunakan, yaitu Android 2.1-update1 pada Build Target. Terakhir, klik Finish.
4. Eclipse akan menyiapkan project Android dan bahkan menyiapkan Activity kita yang pertama bernama HelloAndroid.java sesuai dengan nama yang kita berikan pada Create Activity sebelumnya.

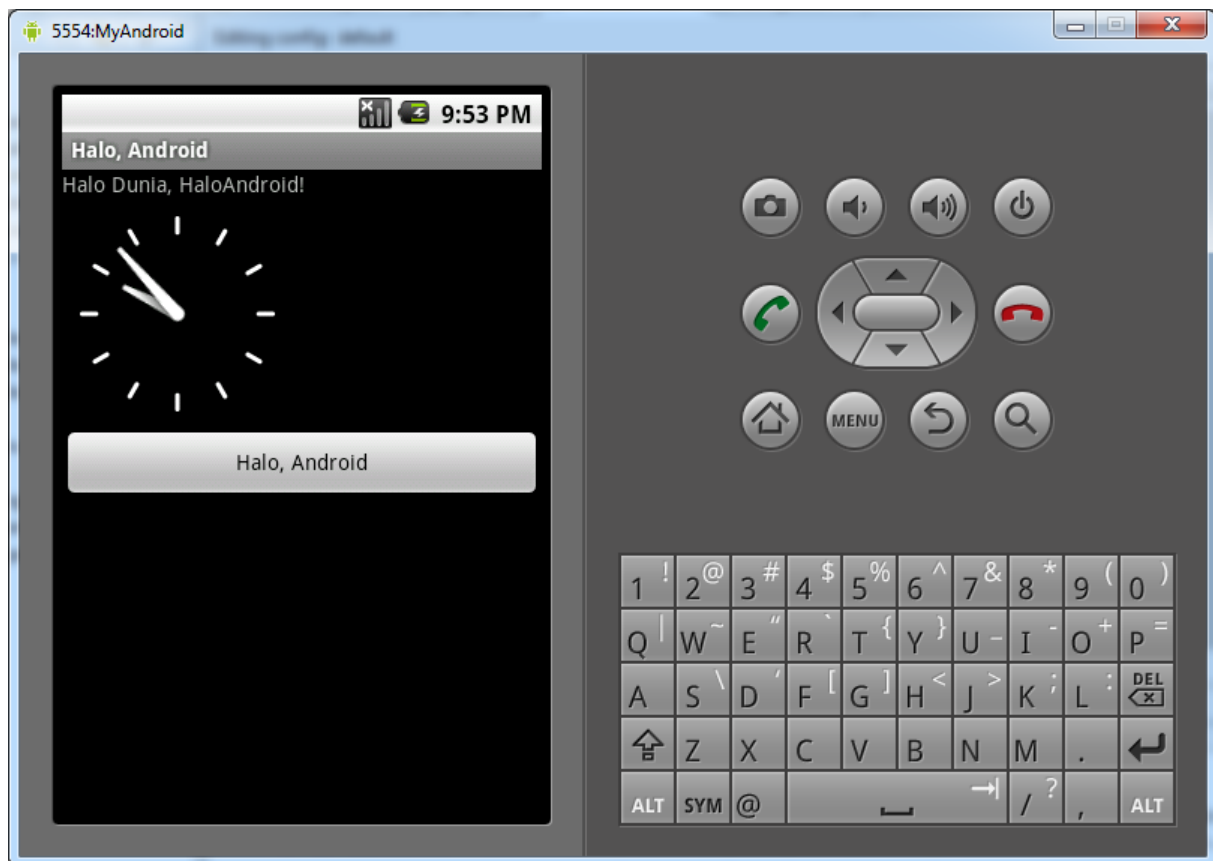


5. Buka Activity yang pertama dengan melakukan ekspansi pada src di Package Explorer yang ada di sebelah kiri, kemudian buka package com.inixindo.workshop, dan klik ganda pada file HelloAndroid.java.
6. Perhatikan kode yang tercantum. Class HelloAndroid merupakan keturunan dari Activity. Pada method onCreate terdapat kode untuk menampilkan sebuah layar bernama “R.layout.main” melalui method setContentView(). Nama “R.layout.main” merupakan sebuah resource yang berarti bahwa ada tampilan layar bernama main yang jenisnya adalah layout, dan dapat diakses melalui class R.
7. Class R dapat dibuka melalui Package Explorer pada gen (berarti Generated Java Files yaitu file yang secara otomatis dibuat), kemudian buka package com.inixindo.workshop, dan klik ganda pada file R.java. Perhatikan bahwa di class R tersebut berisikan daftar resource yang diklasifikasikan dalam berbagai kategori seperti attr, drawable, layout, dan string. Perhatikan bahwa pada kategori layout ada resource bernama main.
8. Untuk mengedit resource jenis layout bernama main tersebut, lakukan ekspansi pada res di Package Explorer. Buka layout, dan klik ganda pada main.xml. File main.xml inilah yang menentukan komponen View apa saja yang ada pada layar untuk Activity HelloAndroid.
9. Format file adalah XML, namun untuk mengeditnya di Eclipse telah disediakan Layout view. Klik pada Layout untuk melihat dalam bentuk WYSIWYG. Daftar komponen View bisa dilihat di bagian kanan melalui Outline. Perhatikan bahwa telah tersedia satu View yaitu TextView yang berisikan teks “Hello World, HelloAndroid!”
10. Pada TextView di Outline, klik kanan dan pilih Properties untuk melihat daftar properties komponen TextView tersebut. Cari property bernama “Text” dan perhatikan bahwa isinya adalah “@string/hello”. Ini berarti bahwa tulisan yang tampil di TextView tersebut diambil dari resource

jenis string yang bernama hello. Jika dilihat kembali pada R.java, maka perhatikan bahwa terdapat resource jenis string dengan nama hello yang sudah terdaftar.

11. Untuk mengubah tulisan melalui resource jenis string bernama hello tersebut, buka res di Package Explorer, kemudian buka values, dan klik ganda pada file strings.xml. Perhatikan bahwa ada 2 resource jenis string yaitu hello dan app\_name. Klik pada hello untuk melihat tulisannya, dan ubahlah menjadi “Halo Dunia, HelloAndroid!” kemudian lakukan Save dengan klik icon disket di toolbar Eclipse.
12. Kembali ke main.xml dan perhatikan bahwa tulisan di TextView sudah berubah mengikuti perubahan resource string hello tadi. Konsep ini sangat berguna karena memisahkan content dari source code. Content terdefinisi pada resource sehingga bisa diedit secara terpisah dari source code. Dengan pemisahan seperti ini, dimungkinkan bagi aplikasi yang kita kembangkan untuk mengganti tulisan dalam berbagai bahasa secara cepat dan mudah tanpa harus mengubah source codenya, namun cukup mengubah file XMLnya saja.
13. Untuk menambahkan berbagai View yang lain, tinggal melakukan drag and drop dari daftar Views yang ada di sebelah kiri. Layout yang digunakan untuk layar main adalah LinearLayout secara vertikal, sehingga View akan tersusun dari atas ke bawah. Silakan menambahkan view AnalogClock dan kemudian Button. Perhatikan bahwa Outline menunjukkan tambahan view bernama analogClock1 dan button1.
14. Di Outline, klik kanan pada button1 dan pilih Properties. Ubah property Text untuk menunjuk pada resource string bernama app\_name. Ubah pula property Layout width dari wrap\_content ke fill\_parent sehingga lebar button memenuhi layar sepenuhnya. Perhatikan perubahan pada Layout view, dan lihat pula hasil XMLnya.

15. Kita siap untuk menjalankan aplikasi HelloAndroid kita. Di Eclipse, pilih menu Run, kemudian Run As, dan pilih Android Application. Eclipse secara otomatis akan melakukan kompilasi, packaging, dan deploying ke Android Virtual Device (AVD) yang sudah jalan. Jika AVD belum jalan, maka Eclipse juga akan menjalankannya secara otomatis dan menggunakan AVD tersebut untuk menjalankan aplikasi HelloAndroid kita setelah proses boot AVD selesai. Perhatikan bahwa proses boot Android memakan waktu yang cukup lama, jadi kita harus sabar menanti.
16. Tampilan seharusnya seperti berikut ini:



## DEBUGGING PROJECT ANDROID:

1. Eclipse IDE sudah menyediakan fasilitas debugging yang lengkap dan nyaman untuk digunakan. Dengan fasilitas debugging tersebut, kita dapat menelusuri berbagai kesalahan logika dalam kode kita secara seksama

dengan memperhatikan berbagai nilai variabel dan object pada saat kode aplikasi kita sedang dieksekusi satu langkah demi satu langkah. Dengan kata lain, kita bisa melakukan code tracing step-by-step untuk mengetahui secara persis dimana letak kesalahan kode yang kita buat.

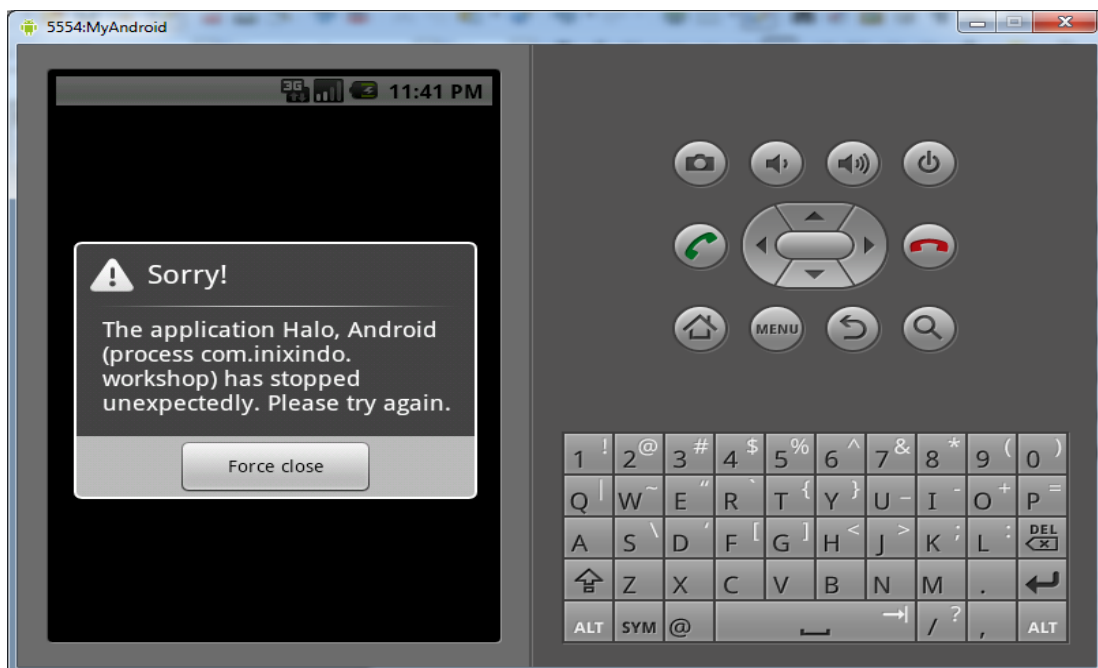
2. Sebagai contoh, lakukan modifikasi kode pada activity HelloAndroid.java sehingga menjadi yang seperti di bawah ini:

```
package com.inixindo.workshop;

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

public class HelloAndroid extends Activity {
 /** Called when the activity is first created. */
 @Override
 public void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 String tulisan = "Klik Saya!";
 Button button = (Button)
 this.findViewById(R.id.button1);
 button.setText(tulisan);
 setContentView(R.layout.main);
 }
}
```

3. Jalankan aplikasi tersebut pada emulator AVD dan perhatikan apa yang terjadi. Seharusnya akan tampil error seperti berikut:



4. Sekarang kita ingin mengetahui di mana letak error tersebut untuk bisa diperbaiki. Pertama, kita harus menentukan baris dimana eksekusi dihentikan sehingga bisa melakukan eksekusi kode step-by-step. Kita menentukan baris tempat pemberhentian ini dengan Breakpoint. Untuk menambah breakpoint, kita tinggal klik ganda di sebelah kiri baris yang dimaksud sampai muncul titik berwarna biru. Alternatifnya, kita bisa klik kanan dan pilih Toggle Breakpoint. Berikan breakpoint pada baris **String tulisan = “Klik Saya!”** sehingga eksekusi kode akan berhenti di baris tersebut.
5. Jalankan debug dengan klik pada icon gambar serangga di toolbar Eclipse. Alternatifnya bisa dengan memilih menu Run, kemudian klik Debug.
6. Setelah menjalankan Debug, maka Eclipse akan menanyakan apakah boleh mengganti Perspective ke Debug. Klik pada Yes untuk mengijinkannya. Tampilan layout Eclipse akan berubah untuk mendukung proses debugging.
7. Perhatikan bahwa baris yang telah diberi breakpoint berwarna hijau untuk menunjukkan bahwa eksekusi berhenti di baris tersebut. Di bagian kanan atas terdapat daftar Variables yang dapat kita amati.
8. Tekan F5 atau klik pada icon Step Into untuk mengeksekusi satu baris kode. Perhatikan bahwa variable tulisan akan muncul di daftar Variables dan berisikan “Klik Saya!” sesuai kode yang dieksekusi.
9. Tekan F6 atau klik pada icon Step Over untuk mengeksekusi baris berikut namun tanpa loncat masuk ke dalam kode sumber untuk baris tersebut. Baris ini adalah untuk mendapatkan button dengan nama R.id.button1 yang ada di layout main.xml. Perhatikan bahwa variable button muncul di daftar Variables namun nilainya adalah null.
10. Tekan F5 untuk Step Into untuk mengeksekusi baris berikut yaitu

pemanggilan method `setText` dari object button. Perhatikan bahwa telah terjadi `NullPointerException` yang bisa terlihat di Variables dan Outline. `NullPointerException` inilah yang menyebabkan aplikasi kita crash di Android sehingga tampil pesan error di Android.

11. Dengan menelusurinya, kita bisa mengetahui sebab dari masalah ini, yaitu bahwa button berupa null yang seharusnya menunjuk pada `button1`. Ini disebabkan karena kita mencoba mendapatkan `button1` padahal layar belum menampilkan layout main, sehingga `button1` memang belum ada.
12. Untuk membetulkan masalah tersebut, kita harus memindahkan kode untuk mengakses `button1` di bawah setelah pemanggilan method `setContentView()`. Hentikan debugging dengan klik tombol Disconnect, dan kemudian kembalikan Perspective ke Java untuk edit source code. Kode yang benar adalah sebagai berikut:

```
package com.inixindo.workshop;

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

public class HaloAndroid extends Activity {
 /** Called when the activity is first created. */
 @Override
 public void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 String tulisan = "Klik Saya!";
 setContentView(R.layout.main);
 Button button = (Button)
 this.findViewById(R.id.button1);
 button.setText(tulisan);
 }
}
```

13. Ulangi proses debugging dan pastikan sekarang bahwa object button memang ada dan tidak null lagi.
14. Selesailah sudah proses debugging. Jalankan lagi aplikasinya dan pastikan telah jalan di Android dengan baik.

## **DEBUGGING DENGAN ANDROID DEVICE:**

1. Selain menggunakan emulator AVD, kita juga bisa melakukan eksekusi dan debugging dengan perangkat Android yang sungguhan. Dalam pengembangan aplikasi Android memang sangat disarankan untuk menjalankan aplikasi yang kita buat di perangkat Android yang sebenarnya, tidak hanya di emulator saja. Ini dikarenakan tidak ada jaminan bahwa emulator AVD memang sama persis seperti perangkat yang sesungguhnya. Debugging dengan perangkat Android yang sesungguhnya juga bisa lebih cepat dan handal ketimbang di emulator AVD.
2. Untuk melakukan eksekusi atau debugging di perangkat Android yang sesungguhnya, maka perangkat Android tersebut harus terhubung ke PC melalui kabel USB. Driver USB perangkat Android juga sudah harus terinstall di PC. Pastikan semua sudah terhubung dan perangkat Android sudah menyala.
3. Perangkat Android yang sesungguhnya harus disetting supaya bisa menerima Thurd Party application, dan juga bisa melakukan debugging melalui USB. Pada Android yang sesungguhnya, pilih Menu dan kemudian pilih Settings.
4. Pada layar Settings, pilih Applications, dan berikan tanda centang pada Unknown sources untuk mengizinkan instalasi aplikasi Android dari pihak ketiga (Third Party apps).
5. Masih di layar Application settings, pilih Development. Berikan tanda centang pada USB debugging untuk mengizinkan proses debugging pada perangkat Android tersebut.
6. Di Eclipse, kita juga harus mengizinkan aplikasi kita agar bisa melakukan proses debug. Penambahan ijin ini harus dilakukan di file AndroidManifest.xml. Bukalah file tersebut di Project Explorer di Eclipse

IDE.

7. Di file AndroidManifest.xml, klik pada tab Application, dan cari setting untuk Debuggable dan ubahlah menjadi true. Lakukan save.
8. Supaya aplikasi kita jalan di perangkat Android sesungguhnya dan bukan di emulator AVD, di Eclipse IDE di bagian Project Explorer, klik kanan pada project kita dan pilih Run As, kemudian Run Configurations.
9. Pada dialog yang tampil, klik pada tab Target. Pada Deployment Target Selection Mode, pilih Manual. Klik Run.
10. Setelah klik Run, maka akan muncul dialog untuk memilih perangkat Android yang ingin digunakan. Jika perangkat Android sesungguhnya sudah terhubung ke PC melalui kabel USB dan driver USB sudah terinstall dengan benar, maka perangkat Android tersebut seharusnya tampil pada daftar device. Pilih perangkat Android tersebut, dan klik OK.
11. Aplikasi kita akan jalan pada perangkat Android yang sesungguhnya secara langsung.
12. Jika ingin melakukan proses debugging, maka tinggal klik saja icon untuk Debug di Toolbar, kemudian pilih perangkat Android yang sesungguhnya, dan klik OK untuk melakukan debugging seperti biasa.



## VIEW LAYOUT & PREFERENCE ACTIVITY

Aplikasi Android yang rumit menggunakan berbagai View dalam Activity, dan biasanya juga mempunyai layar Settings atau Preferences sehingga pengguna dapat melakukan kustomisasi terhadap aplikasi tersebut. Untuk menunjukkan pengaturan tata letak berbagai View dan juga cara untuk menambahkan layar Settings/Preferences dalam Menu di Android, contoh berikutnya yang akan kita buat adalah sebuah aplikasi kalkulator untuk konversi Rupiah ke Dolar atau Dolar ke Rupiah. Layar Settings akan digunakan supaya pengguna dapat memasukkan sendiri nilai kurs yang sedang berlaku saat itu.

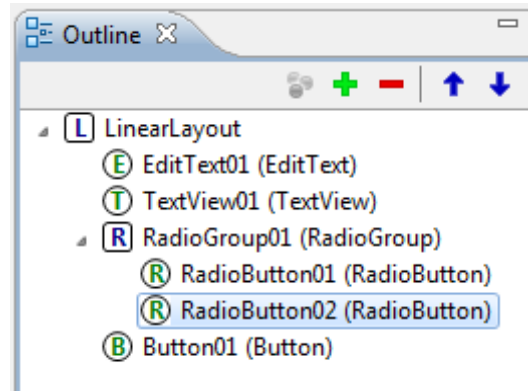
### PEMBUATAN APLIKASI KALKULATOR KURS:

1. Di Eclipse IDE, buatlah project Android baru untuk Android 2.1 dengan nama Kalkulator Kurs. Gunakan package name `com.inixindo.workshop.kurs`, dan buat activity dengan nama `KalkulatorKursActivity`.
2. Tambahkan resources di file `strings.xml` sebagai berikut dan jangan lupa untuk melakukan save setelah selesai mengeditnya:
  - Resource color bernama `myColor` dengan nilai `"#3399CC"`
  - Resource color bernama `textColor` dengan nilai `"#000"`
  - Resource string bernama `rupiah` dengan nilai `"Dolar ke Rupiah"`
  - Resource string bernama `dolar` dengan nilai `"Rupiah ke Dolar"`
  - Resource string bernama `buttonHandler` dengan nilai `"buttonClicked"`

- Resource string bernama hitung dengan nilai “Lakukan Kalkulasi”
3. Klik pada tab XML untuk melihat nilai dalam bentuk XML mentahnya.
  4. Buka file main.xml yang ada di resource layout. File ini menentukan komponen view untuk user interface UI dalam activity. Tampilkan dalam bentuk Layout mode, bukan XML mode supaya bisa melakukan drag and drop komponen view ke dalam layout.
  5. Hapus komponen TextView yang ada di LinearLayout dengan klik kanan dan pilih Remove.
  6. Klik pada LinearLayout, buka Properties, dan ubah property background supaya menggunakan resource color myColor.
  7. Lakukan drag and drop komponen view EditText, TextView, RadioGroup, dan 1 Button. Hapus RadioButton yang terakhir sehingga hanya tersisa 2 saja. Tampilan seharusnya seperti berikut ini:



8. Perhatikan Outline untuk memastikan bahwa kedua RadioButton ada di dalam radioGroup1. Bila tidak, pilih RadioButton dan gunakan anak panah untuk menggesernya sehingga masuk ke dalam radioGroup1. Tampilan Outline seharusnya demikian:



9. Ubah property pada editText1 sehingga Layout width menjadi fill\_parent. Kosongkan pula property Text supaya tidak ada tulisannya. Pada property Input type, beri tanda centang pada number supaya hanya angka saja yang boleh dimasukkan ke dalamnya.
10. Ubah property pada textView1 sehingga Layout width menjadi fill\_parent. Kosongkan pula property Text supaya tidak ada tulisannya. Ubah property Text color menjadi resource color textColor, property Text style menjadi bold, dan property Text size menjadi 10pt.
11. Ubah property pada radio0 sehingga Text menggunakan resource string rupiah.
12. Ubah property pada radio1 sehingga Text menggunakan resource string dolar.
13. Ubah property pada button1 sehingga Text menggunakan resource string hitung. Ubah pula property On click sehingga menggunakan resource string buttonHandler.
14. Lakukan save pada main.xml. Berikut kita perlu mengisi kode untuk class KalkulatorKursActivity. Buka file KalkulatorKursActivity.java di Eclipse IDE.
15. Tambahkan kode di bawah ini di dalam class KalkulatorKursActivity setelah method onCreate(). Pastikan bahwa kode tambahan di bawah ini

masih di dalam kurung kurawal untuk class KalkulatorKursActivity:

```
private long kursDolar = 9000;

private long rupiahToDolar(long rupiah) {
 return rupiah / kursDolar;
}

private long dolarToRupiah(long dolar) {
 return dolar * kursDolar;
}

public void buttonClicked(View view) {
 switch (view.getId()) {
 case R.id.button1:
 EditText input = (EditText) findViewById(R.id.editText1);
 TextView output = (TextView) findViewById(R.id.textView1);
 RadioButton rupiah = (RadioButton) findViewById(R.id.radio0);
 RadioButton dolar = (RadioButton) findViewById(R.id.radio1);
 long inputValue = 0;
 try {
 inputValue = Long.parseLong(input.getText().toString());
 } catch (Exception e) {
 Toast.makeText(this, "Masukkan input berupa angka",
 Toast.LENGTH_LONG).show();
 return;
 }
 if (rupiah.isChecked()) {
 output.setText("$ " + inputValue + " = Rp "
 + dolarToRupiah(inputValue));
 } else if (dolar.isChecked()) {
 output.setText("Rp " + inputValue + " = $ "
 + rupiahToDolar(inputValue));
 } else {
 Toast.makeText(this, "Pilih jenis konversi kurs",
 Toast.LENGTH_LONG).show();
 }
 break;
 }
}
```

16. Pastikan tidak ada kesalahan dalam source code di class KalkulatorKursActivity.java. Jangan lupa untuk melakukan Ctrl-Shift-O untuk auto import. Untuk merapikan source code, tekan Ctrl-Shift-F. Lakukan save pada file.
17. Buka file AndroidManifest.xml, klik pada tab Application, kemudian klik pada KalkulatorKursActivity di bagian Application Nodes. Untuk memastikan bahwa hanya ada satu instance yang jalan pada saat kita menjalankan aplikasi Kalkulator Kurs sehingga tidak berjalan berkali-kali dan menghabiskan memory, pada bagian Attributes pilihlah

“singleInstance” untuk Launch mode. Lakukan save.

18. Jalankan project sebagai Android Application dan periksa hasilnya.

## **PEMBUATAN LAYAR SETTINGS DI MENU:**

1. Kita ingin menambahkan sebuah layar settings yang dapat diakses dengan menekan tombol Menu di perangkat Android. Dalam settings menu ini, kita bisa menentukan kurs dolar saat ini sehingga kalkulasi mencerminkan kurs yang terbaru. Kurs ini juga akan terus tersimpan meskipun kita mematikan aplikasi Kurs Kalkulator, sehingga kita tidak harus selalu menentukan kurs tiap kali menjalankan aplikasi, namun cukup mengubahnya jika memang ada perubahan saja.
2. Untuk menambahkan menu, kita perlu membuat file XML baru untuk mendefinisikan menu tersebut. Dari Eclipse IDE, klik kanan pada project dan pilih New, kemudian pilih Other, kemudian pilih Android, dan pilih Android XML File.
3. Beri nama file “settings.xml” dan pilih Menu sebagai tipe resource. Pada intinya kita menambahkan resource tipe menu untuk aplikasi kita. Klik Finish.
4. Untuk menambahkan menu element baru, klik pada tombol Add. Pilih Item dan klik OK untuk membuat item baru di menu. Ganti Id menjadi “@+id/settings” dan beri Title “Settings”. Lakukan save untuk file.
5. Supaya menu yang kita daftarkan di file settings.xml bisa tampil di aplikasi kita, maka kita harus menambahkan method untuk membuka menu item Settings tersebut bila user melakukan klik pada tombol Menu di perangkat Android. Kembali ke file KalkulatorKursActivity.java, klik pada menu Source di Eclipse IDE, pilih Override/Implement Methods.
6. Berikan centang pada method onCreateOptionsMenu(Menu) dan klik tombol OK. Ini akan menyiapkan method untuk membuka menu settings.

Tambahkan kode di dalam method `onCreateOptionsMenu()` tersebut sehingga menjadi seperti di bawah ini:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
 MenuInflater inflater = getMenuInflater();
 inflater.inflate(R.menu.settings, menu);
 return true;
}
```

7. Lakukan Ctrl-Shift-O untuk melakukan auto-import terhadap class `MenuInflater`. Class tersebut digunakan untuk menampilkan (inflate) menu yang terdapat pada resource file `settings.xml` dengan nama `R.menu.settings`.
8. Jalankan aplikasi sekarang, dan coba klik tombol Menu pada perangkat Android. Perhatikan bahwa menu Settings muncul, namun belum ada kode untuk melakukan apa pun bila Settings dipilih.
9. Supaya setting kurs bisa tersimpan di Android, kita harus membuat resource dengan tipe Preference. Klik kanan pada project di Eclipse IDE Project Explorer, pilih New, kemudian Other, kemudian Android, dan pilih Android XML File.
10. Berikan nama file “kurs.xml” dan pilih Preference sebagai tipe resource. Klik Finish.
11. Klik tombol Add untuk menambahkan sebuah preference setting yang digunakan untuk menyimpan kurs dolar. Pada dialog yang muncul, pilih “Create a new element at the top level, in PreferenceScreen”. Pilih `EditTextPreference` dan klik OK. Element baru ini merupakan setting dalam bentuk `EditText` dimana user bisa mengisi kurs dolar saat itu.
12. Lakukan ekspansi pada “Attributes from Preference”. Isi Key dengan “kursDolar”, Title dengan “Kurs Dolar”, dan Summary dengan “Isi kurs Dolar saat ini”. Lakukan save pada file.
13. Berikut kita perlu membuat Java class baru dengan tipe `PreferenceActivity` sebagai layar Settings yang menggunakan resource file

kurs.xml tersebut.

14. Di Eclipse IDE pada Project Explorer, klik kanan pada project, pilih New, kemudian pilih Class untuk membuat Java class biasa. Pilih com.inixindo.workshop.kurs sebagai package, dan beri nama Settings untuk Name. Untuk Superclass, gunakan PreferenceActivity, klik Browse untuk memilih “PreferenceActivity – android.preference”. Klik Finish.
15. Di menu Source, pilih Override/Implement Methods. Beri centang pada method onCreate(Bundle) dan klik OK. Method onCreate() melakukan inisialisasi activity untuk layar Settings.
16. Agar activity Settings ini menampilkan setting kursDolar yang telah kita definisikan di file kurs.xml sebelumnya, tambahkan kode pada method onCreate() sehingga menjadi seperti di bawah ini:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
 // TODO Auto-generated method stub
 super.onCreate(savedInstanceState);
 this.addPreferencesFromResource(R.xml.kurs);
}
```

17. Lakukan save pada Settings.java. Karena kita menambahkan sebuah activity baru di aplikasi kita, maka kita harus mendaftarkannya di AndroidManifest.xml supaya bisa dikenal dan diakses. Buka file AndroidManifest.xml dari Project Explorer di Eclipse IDE, dan klik pada tab Application.
18. Di Application Nodes, tambahkan activity baru dengan klik tombol Add, pilih Activity, dan klik OK.
19. Pada Name, klik tombol Browse dan pilih Settings. Klik OK. Pada Label, beri nama “Settings”.
20. Supaya activity Settings hanya berjalan sekali saja saat aplikasi dijalankan, maka pada Launch mode pilihlah “singleInstance”. Lakukan save pada file AndroidManifest.xml.

21. Untuk menampilkan activity Settings pada saat user melakukan klik tombol Menu di perangkat Android dan memilih Settings, maka kita harus menambahkan method di KalkulatorKursActivity yang membuat sebuah Intent untuk loncat ke activity Settings. Buka file KalkulatorKursActivity.java, kemudian klik menu Source dan pilih Override/Implement Methods.
22. Berikan tanda centang pada method onOptionsItemSelected(MenuItem) dan klik OK. Isi method tersebut dengan kode berikut sehingga menjadi seperti di bawah ini:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
 switch (item.getItemId()) {
 case R.id.settings:
 Intent intent = new Intent(this, Settings.class);
 startActivity(intent);
 break;
 }
 return true;
}
```

23. Gunakan Ctrl-Shift-O untuk auto-import class Intent. Kode di atas hanya memastikan bahwa menu item yang dipilih adalah settings, dan bila memang demikian maka sebuah Intent baru dibuat untuk pindah ke activity Settings.
24. Terakhir, kita perlu melakukan sedikit perubahan dan penambahan kode pada method buttonClicked sehingga nilai kurs dolar diambil dari settings yang telah ditentukan oleh user di Settings preference activity. Kode untuk method buttonClicked adalah seperti berikut:

```
public void buttonClicked(View view) {
 switch (view.getId()) {
 case R.id.Button01:
 EditText input = (EditText) findViewById(R.id.editText1);
 TextView output = (TextView) findViewById(R.id.textView1);
 RadioButton rupiah = (RadioButton) findViewById(R.id.radio0);
 RadioButton dolar = (RadioButton) findViewById(R.id.radio1);
 long inputValue = 0;
 try {
 inputValue = Long.parseLong(input.getText().toString());
 } catch (Exception e) {
 Toast.makeText(this, "Masukkan input berupa angka",

```



```

 Toast.LENGTH_LONG).show();
 return;
 }
 SharedPreferences settings =
 PreferenceManager.getDefaultSharedPreferences(this);
 String inputKurs = settings.getString("kursDolar", "9000");
 try {
 kursDolar = Long.parseLong(inputKurs);
 } catch (NumberFormatException e) {
 Toast.makeText(this, "Kurs dolar tidak valid",
 Toast.LENGTH_LONG).show();
 return;
 }
 if (rupiah.isChecked()) {
 output.setText(" $ " + inputValue + " = Rp "
 + dolarToRupiah(inputValue));
 } else if (dolar.isChecked()) {
 output.setText(" Rp " + inputValue + " = $ "
 + rupiahToDolar(inputValue));
 } else {
 Toast.makeText(this, "Pilih jenis konversi kurs",
 Toast.LENGTH_LONG).show();
 }
 break;
}
}
}

```

25. Jalankan aplikasi tersebut dan lakukan testing untuk memastikan bahwa layar Settings tampil waktu kita klik tombol Menu pada perangkat Android. Kita dapat memasukkan kurs dolar di layar Settings, dan kalkulasi menggunakan kurs yang kita simpan di settings tersebut.

## 5

### KONEKSI ANDROID WEB SERVICE

Aplikasi Android tidak mempunyai kemampuan untuk melakukan koneksi secara langsung ke database server. Jika aplikasi Android perlu mengakses data eksternal, maka diperlukan bantuan sebuah server yang menyediakan data tersebut kepada aplikasi Android melalui jaringan komputer termasuk Internet. Salah satu cara bagi aplikasi Android untuk mengakses data eksternal adalah dengan melakukan koneksi ke suatu web service. Dengan kata lain, aplikasi Android dapat bertindak sebagai web service client.

Standar web service yang ada sekarang ini sebenarnya ada 2, yaitu SOAP web service dan RESTful web service. Untuk SOAP web service, para developer aplikasi Android dapat menggunakan library yang sudah tersedia yaitu ksoap2-android. Library ini dapat di-download dan digunakan secara gratis dari <http://code.google.com/p/ksoap2-android/> yang merupakan situs resmi ksoap2-android. File yang dibutuhkan adalah ksoap2-android-assembly-2.5.7-jar-with-dependencies.jar.

Sebagai contoh penggunaan ksoap2-android, kita akan menyempurnakan aplikasi Kalkulator Kurs yang telah dibuat sebelumnya sehingga nilai kurs dolar yang terbaru didapatkan melalui web service yang ada di <http://www.webservice.net/ws/wsdetails.aspx?wsid=10> untuk aplikasi kita. Tiap kali aplikasi Kalkulator Kurs dijalankan, maka aplikasi itu akan melakukan koneksi ke web service untuk meng-update nilai kurs dolar dan kemudian menyimpannya sebagai settings preference sehingga kalkulasi selalu

menggunakan nilai yang terbaru.

## **PENAMBAHAN KSOAP2-ANDROID DI ECLIPSE:**

1. Lakukan download file ksoap2-android-assembly-2.5.7-jar-with-dependencies.jar dan taruhlah di suatu folder.
2. Di Eclipse, klik kanan pada project KalkulatorKurs dan pilih Properties.
3. Klik pada Java Build Path, kemudian pada tab Libraries. Klik pada Add External JARs dan pilih file ksoap2-android yang telah didownload sebelumnya. Klik Open untuk kembali ke Java Build Path.
4. Klik pada tab Order and Export yang ada di sebelah kanan tab Libraries. Berikan tanda centang pada ksoap2-android library, dan kemudian pilih library tersebut dan klik tombol Up untuk menggeser urutan library tersebut sehingga berada di atas library Android 2.2. Hal ini dilakukan untuk memastikan bahwa library ksoap2-android akan diikutsertakan dalam aplikasi Kalkulator Kurs waktu di-export menjadi file APK setelah semuanya selesai. Klik OK untuk menutup Properties.
5. Pastikan di Package Explorer bahwa library ksoap2-android sudah terdaftar di bagian Referenced Libraries.

## **PEMBUATAN CLASS UNTUK KONEKSI WEB SERVICE:**

1. Kita ingin membuat class baru untuk menangani koneksi ke web service Currency Convertor sebagai web service client dengan menggunakan library ksoap2-android.
2. Untuk memastikan bahwa web service Currency Convertor memang jalan, kita bisa menggunakan aplikasi StrikeIron Web Services Analyzer untuk melakukan koneksi ke web service tersebut dan melihat hasilnya.
3. Perhatikan bahwa lokasi WSDL untuk web service Currency Convertor

ada di <http://www.webservices.net/CurrencyConvertor.asmx?WSDL> yang merupakan input untuk StrikeIron Web Services Analyzer. Setelah memasukkan lokasi WSDL di bagian WSDL Location dan menekan tombol Analyze, maka akan tampil rincian web service operation. Perhatikan bahwa nama operation adalah ConversionRate yang menerima 2 input yaitu FromCurrency dan ToCurrency.

4. Untuk melihat kurs dolar yang terakhir dari web service tersebut, pilih USD untuk FromCurrency dan IDR untuk ToCurrency, kemudian klik kanan dan pilih Invoke untuk menjalankan web service. Setelah menunggu beberapa saat, hasil akan tampil di bagian kanan.
5. Jika ingin membaca file WSDL secara langsung, jalankan web browser dan buka <http://www.webservices.net/CurrencyConvertor.asmx?WSDL> untuk melihat rinciannya. Cara membaca WSDL yang efektif adalah membacanya dari bawah ke atas. Beberapa informasi yang ada di file WSDL ini sangat dibutuhkan waktu membuat class untuk koneksi web service dengan library ksoap2-android. Informasi yang dibutuhkan dari file WSDL adalah targetNamespace, operation name, dan address location. Kita juga perlu mengetahui platform yang digunakan oleh web service apakah menggunakan teknologi Microsoft .NET atau bukan. Ekstensi .asmx pada address location meyakinkan kita bahwa implementasi web service Currency Converter menggunakan teknologi Microsoft .NET.
6. Di Eclipse, klik kanan pada project KalkulatorKurs, kemudian New, kemudian Class untuk membuat class baru. Berikan package com.inixindo.workshop.kurs dan nama CurrencyConvertorSoap untuk class tersebut. Klik Finish.
7. Tambahkan empat attribute berikut di dalam class CurrencyConvertorSoap untuk menentukan koneksi web service.

Perhatikan bahwa informasi di attribute tersebut didapatkan dari informasi WSDL sebelumnya:

```
private static final String NAMESPACE =
 "http://www.webserviceX.NET/";
private static final String METHOD_NAME = "ConversionRate";
private static final String ADDRESS =
 "http://www.webserviceX.net/CurrencyConvertor.asmx";
private static final boolean IS_DOT_NET = true;
```

8. Tambahkan method berikut di dalam class CurrencyConvertorSoap di bawah keempat attribute di atas:

```
public double conversionRate(String fromCurrency, String toCurrency)
 throws Exception {
 SoapSerializationEnvelope envelope =
 new SoapSerializationEnvelope(SoapEnvelope.VER11);
 envelope.dotNet = IS_DOT_NET;
 SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
 request.addProperty("FromCurrency", fromCurrency);
 request.addProperty("ToCurrency", toCurrency);
 envelope.setOutputSoapObject(request);
 HttpTransportSE androidHttpTransport =
 new HttpTransportSE(ADDRESS);
 androidHttpTransport.call(NAMESPACE + METHOD_NAME, envelope);
 Object response = envelope.getResponse();
 double resp = Double.parseDouble(response.toString());
 return resp;
}
```

9. Pada source code di atas, kita menggunakan berbagai class yang tersedia di library ksoap2-android untuk melakukan koneksi ke web service Currency Convertor. Perhatikan bahwa versi SOAP yang digunakan adalah versi 1.1 karena versi itulah yang masih banyak digunakan. Namun jika web service menggunakan versi 1.2 yang lebih baru, maka ubahlah SoapEnvelope.VER11 menjadi SoapEnvelope.VER12.
10. Perhatikan pula bahwa envelope.dotNet dijadikan true karena Currency Convertor menggunakan teknologi Microsoft .NET. Bila web service tidak menggunakan Microsoft .NET, maka IS\_DOT\_NET perlu diubah menjadi false.
11. Perhatikan penambahan property FromCurrency dan ToCurrency ke dalam object request sebagai input untuk web service. Nama kedua

parameter input ini didapatkan dari file WSDL dan berupa case-sensitive sehingga kita harus memastikan bahwa huruf besar dan kecil sama persis!

12. Sisa source code adalah untuk memasukkan object request ke dalam SOAP envelope, membuat object HttpTransportSE untuk mengirim SOAP envelope, dan memanggil method call untuk melakukan pengirimannya. Perhatikan bahwa pemanggilan call menerima parameter gabungan dari NAMESPACE dan METHOD\_NAME untuk menentukan SOAP action. Jika dilihat di file WSDL, maka soapAction yang ada di file WSDL tersebut pasti sama dengan gabungan NAMESPACE dan METHOD\_NAME. Beberapa baris terakhir hanya mengambil hasil response dari pemanggilan web service dan melakukan konversi ke double.
13. Tekan Ctrl-Shift-O dan Ctrl-Shift-F untuk menambahkan import dan merapikan source code. Lakukan save terhadap class tersebut.
14. Selesailah sudah pembuatan class untuk menangani koneksi ke web service Currency Convertor. Berikutnya, kita perlu melakukan perubahan terhadap class KalkulatorKursActivity sehingga menggunakan class CurrencyConvertorSoap tersebut untuk mendapatkan nilai kurs terbaru dan melakukan update settings preference kursDolar untuk digunakan dalam kalkulasi.

## **PENGGUNAAN CURRENCYCONVERTORSOAP:**

1. Koneksi ke web service Currency Convertor untuk mengupdate kursDolar ingin kita lakukan pada saat awal waktu user menjalankan aplikasi Kalkulator Kurs. Oleh karena itu, kita bisa menambahkan kodenya di dalam method onCreate() yang pasti jalan di awal. Namun, kita perlu menjalankannya sebagai Thread tersendiri agar user interface tidak hang karena koneksi ke web service bisa memakan waktu yang agak lama.

Dengan menjalankan koneksi web service sebagai Thread tersendiri yang terpisah, maka koneksi web service dapat berjalan di background tanpa mengganggu Thread utama untuk user interface. Bila update berhasil ataupun koneksi ke web service gagal, maka user akan mendapatkan pesan singkat untuk memberitahunya.

2. Di Eclipse, buka class KalkulatorKursActivity dan gantilah isi dari method onCreate() yang sudah ada menjadi berikut ini:

```
super.onCreate(savedInstanceState);
setContentView(R.layout.main);

Runnable webServiceClientTask = new Runnable() {
 @Override
 public void run() {
 try {
 CurrencyConvertorSoap ccs = new
 CurrencyConvertorSoap();
 kursDolar = (long)
 ccs.conversionRate("USD", "IDR");
 SharedPreferences settings = PreferenceManager
 .getDefaultSharedPreferences(
 KalkulatorKursActivity.this);
 Editor editor = settings.edit();
 editor.putString("kursDolar",
 String.valueOf(kursDolar));
 editor.commit();
 KalkulatorKursActivity.this.runOnUiThread(
 new Runnable() {
 @Override
 public void run() {
 Toast.makeText(KalkulatorKursActivity.this,
 "Kurs berhasil diupdate",
 Toast.LENGTH_LONG).show();
 }
 });
 } catch (Exception e) {
 e.printStackTrace();
 KalkulatorKursActivity.this.runOnUiThread(
 new Runnable() {
 @Override
 public void run() {
 Toast.makeText(KalkulatorKursActivity.this,
 "Koneksi web service gagal",
 Toast.LENGTH_LONG).show();
 }
 });
 }
 }
};

Thread webServiceClientThread = new Thread(webServiceClientTask);
webServiceClientThread.start();
```

3. Tekan Ctrl-Shift-O dan Ctrl-Shift-F untuk menambahkan import dan merapikan source code. Lakukan save terhadap class tersebut.
4. Source code di atas membuat object Runnable yang berisikan kode untuk menggunakan class CurrencyConvertorSoap dan memanggil method `conversionRate()` dengan parameter “USD” dan “IDR” sehingga web service Currency Convertor memberikan nilai kurs terbaru untuk US Dollar ke Rupiah. Hasilnya di-cast sebagai long untuk disimpan ke attribute `kursDolar` yang ada di class `KalkulatorKursActivity`. Nilai ini kemudian disimpan ke settings preference “kursDolar” sehingga tersimpan secara permanen meskipun aplikasi Kalkulator Kurs dimatikan. Bila penyimpanan ke settings preference berhasil dilakukan, maka Toast digunakan untuk menampilkan pesan ke user. Perhatikan bahwa penggunaan Toast harus dilakukan melalui method `runOnUiThread()` dari activity karena Thread untuk koneksi web service tidak dapat memanipulasi komponen user interface secara langsung. Hanya Thread utama untuk user interface yang dapat memanipulasi komponen user interface secara langsung.
5. Bila terjadi masalah waktu koneksi ke web service, maka catch akan menangkap semua Exception dan menampilkan pesan ke user. Pesan Exception juga ditampilkan melalui method `printStackTrace()` untuk diselidiki lebih lanjut dan dapat dilihat di LogCat. Untuk menampilkan LogCat, di Eclipse pilih menu Window, kemudian Show View, kemudian Other, kemudian lakukan ekspansi Android dan pilih LogCat.
6. Step terakhir adalah untuk mengijinkan aplikasi Kalkulator Kurs agar boleh melakukan koneksi ke Internet sehingga dapat mengakses web servicenya. Buka file `AndroidManifest.xml` dan pilih tab Permissions. Klik pada tombol Add dan pilih Uses Permission. Pada Name, pilihlah



“android.permission.INTERNET”. Lakukan save.

7. Jalankan aplikasi Kalkulator Kurs dan pastikan bahwa sudah berjalan dengan baik. Waktu menjalankannya, tunggu sesaat sampai muncul pesan bahwa kurs dolar berhasil diupdate, dan periksa hasilnya baik melalui menu Settings maupun melakukan kalkulasi.
8. Jika aplikasi ingin langsung dicoba di phone Android Anda, klik kanan pada project dan pilih Export. Lakukan ekspansi Android dan pilih Export Android Application. Ikuti petunjuk dari wizard hingga selesai menjadi sebuah file APK. File APK ini dapat langsung di-install ke phone Android sungguhan dan kemudian dijalankan. Ada beberapa cara untuk melakukan instalasi, termasuk melalui web server. Silakan eksplorasi sendiri dan selamat mencoba!