Pengembangan Aplikasi Perangkat Bergerak (Mobile)

Informatika / Ilmu Komputer PTIIK UB
Semester Ganjil 2013/2014

PENGEMBANGAN APLIKASI NATIVE BERBASIS ANDROID

Dr. Eng. Herman Tolle, ST., MT

Lab Web & Mobile App - PTIIK Brawijaya University
hermantolle.com/class
What is a mobile app?

• **mobile app** (n.) *a software program that runs on a smartphone, tablet or similar device.*
Type of Application

- **Native App**: dibuat, dikompilasi dan diinstalasi khusus untuk platform tertentu
- **Web based App**: dibangun dengan teknologi pembuatan web: HTML, CSS, Javascript. Bersifat CROSS PLATFORM
- **Hybrid**: gabungan Native + Web
Platform Native Application

- Java ME (sebelumnya J2ME)
- .NET Compact Framework (C++, C#, VB.NET) for Windows Mobile
- Qualcomm’s BREW (C or C++)
- Symbian (C++)
- BlackBerry (Java)
- **Android (Java)**
- iPhone (Objective-C)
Android History

• Android is the software platform from Google and the Open Handset Alliance.
• Android is a software environment built for mobile devices. It is not a hardware platform. While components of the underlying OS are written in C or C++, user applications are built for Android in Java.

• In July 2005, Google acquired Android, Inc., a small startup company based in Palo Alto, CA. 4 of Android's co-founders went to work at Google.
• Android has been available as open source since October 2008. Google opened the entire source code under an Apache License.
Update history

- Android has seen a number of updates since its original release. These updates to the base Operating System typically fix bugs and add new features.

- On 30 April 2009, the official 1.5 (Cupcake) update for Android was released.
- On 15 September 2009, the 1.6 (Donut) SDK was released.
- On 26 October 2009 the 2.0 (Eclair) SDK was released
- On 3 December 2009 the 2.0.1 SDK was released.
- On 12 January 2010 the 2.1 SDK was released.
- 2.2 → 20 Mei 2010 (Froyo)
- 2.3 → 6 Desember 2010 (Gingerbred)
- 3.0 → Honeycomb
- 4.0 → 19 October 2011 (Ice Cream Sandwitch)
- 4.1 → Jelly Bean
- 4.4 → KitKat
• 2011
  – SDK 3.0/3.1/3.2 (Honeycomb) for tablets only
    • New UI for tablets, support multi-core processors
  – SDK 4.0/4.0.1/4.0.2/4.0.3 (Ice Cream Sandwich)
    • Changes to the UI, Voice input, NFC

Cupcake
Android 1.5

Donut
Android 1.6

Eclair
Android 2.0/2.1

Froyo
Android 2.2

Ice cream Sandwich
Android 4.0+
Why Android?

• A truly open, free development platform based on Linux and open source
• A component-based architecture
• Tons of built-in services out of the box (GPS, SQL, browser and map views)
• Automatic management of the application life cycle
• High-quality graphics and sound
• 3D-accelerated OpenGL graphics
• various codecs for audio and video formats built-in
• Portability across a wide range of current and future hardware (apps in Java)
Perangkat Android

- Smartphone
- Tablet
- Perangkat E-Reader
- Netbooks
- Mp4 Player
- Internet TV / Smart TV
Android Features

- Reuse and replacement of components
- Dalvik virtual machine
- Integrated browser
- Optimized graphics
- SQLite
- Media support
- GSM Telephony
- Bluetooth, EDGE, 3G, and WiFi
- Camera, GPS, compass, and accelerometer
- Rich development environment
What is Google Android?

• A software stack for mobile devices that includes
  – An operating system
  – Middleware
  – Key Applications

• Uses Linux to provide core system services
  – Security
  – Memory management
  – Process management
  – Power management
  – Hardware drivers
Android Architecture

Applications
- Home
- Dialer
- SMS/MMS
- IM
- Browser
- Camera
- Alarm
- Calculator
- Contacts
- Voice Dial
- Email
- Calendar
- Media Player
- Photo Album
- Clock
- ...

Application Framework
- Activity Manager
- Window Manager
- Content Providers
- View System
- Notification Manager
- Package Manager
- Telephony Manager
- Resource Manager
- Location Manager
- ...

Libraries
- Surface Manager
- Media Framework
- SQLite
- WebKit
- Libc
- OpenGL
- Audio Manager
- FreeType
- SSL
- ...

Android Runtime
- Core Libraries
- Dalvik Virtual Machine

Hardware Abstraction Layer
- Graphics
- Audio
- Camera
- Bluetooth
- GPS
- Radio (RIL)
- WiFi
- ...

Linux Kernel
- Display Driver
- Camera Driver
- Bluetooth Driver
- Shared Memory Driver
- Binder (IPC) Driver
- USB Driver
- Keypad Driver
- WiFi Driver
- Audio Drivers
- Power Management
Native Libraries

- **Surface Manager:** drawing commands go into off-screen bitmaps that are then combined with other bitmaps to form the display the user sees.
- **2D and 3D graphics:** Two- and three-dimensional elements can be combined in a single user interface with Android. The library will use 3D hardware if the device has it or a fast software renderer if it doesn’t.
- **Media codecs:** Android can play video and record and play back audio in a variety of formats including AAC, AVC (H.264), H.263, MP3, and MPEG-4.
- **SQL database:** Android includes the lightweight SQLite database engine, the same database used in Firefox and the Apple iPhone.
- **Browser engine:** For the fast display of HTML content, Android uses the WebKit library. Same engine used in the Google Chrome browser, Apple’s Safari browser, the Apple iPhone, and Nokia’s S60 platform.
Android Runtime

- Core Java Libraries (different than Java SE but there is a substantial amount of overlap)
- Dalvik: Java virtual machine optimized for low memory requirements
  - runs .dex files, which are converted at compile time from standard .class and .jar files.
  - .dex files are more compact and efficient than class files, an important consideration for the limited memory and battery-powered devices that Android targets.
Application Framework

• Provides the high-level building blocks you will use to create your applications.

• **Activity Manager**
  – Controls the life cycle of applications
  – Maintains a common backstack for user navigation

• **Content providers**: Objects encapsulate data that needs to be shared between applications (e.g. contacts)

• **Resource manager**: Resources are anything that goes with your program that is not code

• **Location manager**: An Android phone always knows where it is

• **Notification manager**: Events such as arriving messages, appointments, proximity alerts, alien invasions, and more can be presented to the user
Applications and Widgets

• End users will see only these programs.
  – **Applications**: Programs that can take over the whole screen and interact with the user
  – **Widgets**: Operate only in a small rectangle of the Home screen application

• Android phones prepackaged with a number of standard system applications, including:
  – Phone dialer
  – Email
  – Contacts
  – Web browser
  – Android Market
Android App Life Cycle

State:
- New Activity
- Running
- Pause
- Stopped
- Destroyed

Activity Lifecycle

1. `onCreate()`
2. `onStart()`
3. `onResume()`

- `onPause()`
- `onResume()`
- `onStop()`
- `onDestroy()`
- `<<kill>>`
LifeCycle

- **onCreate(Bundle):** Dipanggil saat pertama kali aplikasi dijalankan. Biasanya digunakan untuk deklarasi variabel atau membuat UI.
- **onStart():** bisa digunakan untuk menampilkan indikator aktivitas yang ditampilkan ke pengguna.
- **onResume():** Dipanggil saat aplikasi kita mulai berinteraksi dengan pengguna. Biasanya digunakan untuk memulai tampilan animasi atau musik.
- **onPause():** Dipanggil saat aplikasi kita harus menjadi background karena ada aplikasi lain yang diaktifkan. Bisa digunakan untuk menyimpan informasi saat ini, persistent state, seperti misalnya database yang sedang diedit.
- **onStop():** Dipanggil saat aplikasi kita berjalan dibelakang layar dalam waktu yang lama. If memory is tight, onStop() may never be called (the system may simply terminate your process).
• **onRestart()**: Kondisi dimana aplikasi kita diaktifkan kembali dari kondisi Stopped atau Paused

• **onDestroy()**: Dikerjakan sebelum aplikasi kita benar-benar dihentikan dari sistem.

• **onSaveInstanceState(Bundle)**: Android will call this method to allow the activity to save per-instance state, such as a cursor position within a text field. Usually you wont need to override it because the default implementation saves the state for all your user interface controls automatically.

• **onRestoreInstanceState(Bundle)**: This is called when the activity is being reinitialized from a state previously saved by the onSaveInstanceState( ) method. The default implementation restores the state of your user interface.
Komponen Aplikasi

1. **Activity**: bagian dari aplikasi yang berinteraksi langsung dengan pengguna. Misalnya: User Interface (UI)

2. **Service**: tidak memiliki UI tapi dpt berjalan secara background, misalnya: music player

3. **Broadcast Receiver**: menerima isyarat dari sistem android misalnya saat *baterai low* atau *screen off*

4. **Content Providers**: data aplikasi lain yang dapat diakses bersama, misalnya: contact, galeri foto
Development Tools

- Java SE Development Kit (JDK)
- Android SDK
  - http://developer.android.com/sdk
- Android Development Tools (ADT)
- Program Editor: Eclipse
Setup

• Menginstal JDK
• Mengistal Eclipse
• Menginstal Android SDK
• Menginstal Plugin ADT untuk Eclipse
• Konfigurasi SDK
• Membuat Android Virtual Device (AVD)
Create an Android Virtual Device (AVD)

- Defines the system image and device settings used by the Emulator

- To create an AVD in Eclipse:
  1. Select Window > Android SDK and AVD Manager. The Android SDK and AVD Manager displays.
  2. Make sure the entry for Virtual Devices is selected and click New. The Create new AVD window displays.
  3. Enter a Name for the AVD.
  4. Click Create AVD.
  5. Close the Android SDK and AVD Manager.
Eclipse and Android Emulator
Hello World App

1. Jalankan Eclipse, File → New → Android Project
2. Isikan kotak dialog new dengan
   Nama Project: Halo
   Build Target: Android 2.2,
   Nama Aplikasi: Halo, World,
   Nama Package: contoh.halo,
   Create Activity: halo,
   SDK Version minimal: 8
3. Pada Package Explorer, buka main.xml, modifikasi untuk menambahkan teks “Hello Android!”
5. Pada package src/contoh.halo, buka halo.java dan pastikan kodenya

```java
public class halo extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```
• Jalankan program dengan shortcut CTRL-F11 atau Package→Run As→Android Project
Deadline: 6 Oktober

TUGAS MINGGU KE-4
Dokumen Laporan

• Membuat Dokumen tentang Framework atau Emulator/ Simulator
• Judul dipilih dari pilihan yang ada
• Dikerjakan 2 orang, membuat 1 laporan dokumen
• Panjang minimal 4 lembar A4 dengan format sama seperti sebelumnya
• Gambar maksimal ¼ dari keseluruhan isi
Isi Dokumen (Framework)

- Tagline
- Sekilas (About, Pencipta, kapan, language, link, Free?)
- Features (Fitur, Kapabilitas, keunggulan, dll)
- Showcase / Aplikasi (yang dibuat dengan Framework tsb)
- Download (link, ukuran file, instalasi)
- Contoh / Prosedur Cara penggunaan (dilengkapi Screenshot)
Isi Dokumen (Simulator/Emulator)

- Untuk semua emulator/simulator yg ada
- Sekilas (Link, About, Pencipta, link, Free?)
- Features (Fitur, Kapabilitas, keunggulan, dll)
- Download (link, ukuran file)
- Contoh / Prosedur Cara penggunaan (dilengkapi Screenshot)
Pengumpulan Tugas

• Upload sebagai konten WIKI di Web E-Lerning Class
• File dalam format doc/docx/odt dikirimkan ke email dosen dengan format subjek:
  • Kelas-NIM-NoTugas
  • NIM→11_123