



Modul Panduan Mobile Computing System Universitas Gunadarma



Laboratorium Sistem Komputer Lanjut
Universitas Gunadarma

Daftar Isi :

Briefing :

B.1	Android	5
B.1.1	Arsitektur Android	6
B.1.2	Versi Android	9
B.1.3	Fitur-fitur Android	14
B.1.4	Android SDK (Software Development Kit)	15
B.1.5	Pengembangan Aplikasi Android	15
B.1.6	Antarmuka Pengguna Android	16
B.1.7	Pemrograman Android	16
B.1.8	File AndroidManifest.xml	18
B.2	Sejarah dan Perkembangan Java	18
B.2.1	Kelebihan Java	20
B.2.2	Karakteristik Java	21
B.3	XML	22
B.3.1	Keunggulan XML	22
B.3.2	Perintah Dasar XML Android	23
B.4	ADT (Android Development Tools) Bundle	24
B.4.1	Interface Eclipse	26
B.4.2	Emulator Android	28

BAB 1 : [Pengenalan Android](#)

1.1.	Android	38
1.1.1.	Fitur Android	38
1.2.	Android SDK (Software Development Kit)	40
1.3.	Android Development Tools (ADT)	41
1.4.	Pengembangan Aplikasi Android	41
1.4.1.	Activities	41
1.4.2.	Service	42
1.4.3.	Content Provider	42
1.4.4.	Broadcast Receiver	44
1.5.	Pemrograman Android	44
1.6.	Android Manifest	49

BAB 2 : [Android User Interface](#)

2.1.	View	54
------	------	----

2.2.	Layout	58
2.3.	Layout Manager	59
2.4.	UI Events	62
2.5.	Mendeklarasikan User Interface (UI)	63
2.6.	Widget	64
2.6.1.	Resources dan R.Java	66
2.6.2.	List Bertingkat	68
2.6.3.	Tab Menu	69

BAB 3 :Android Multimedia

3.1.	Audio Dalam Android	71
3.2.	Animasi Pada Android	72

BAB 4 :Aplikasi Game Android Tebak Kata

4.1.	Definisi Game	74
4.2.	Game Platforms	74
4.3.	Game Genres	77
4.4.	Konsep Pengembangan Game	79
4.5.	Game Tebak Kata	81
4.6.	Perbedaan Game iOS dan Game Android	82

BAB 5 :SQLite Database

5.1.	Pengertian SQLite	84
------	-------------------	----

BAB 6 :Android Client-Server

6.1.	Pengertian Client	88
6.2.	Fungsi Client Server	89
6.3.	Aplikasi Client Server	90

BAB 7 :Android Map Google API V2

7.1.	Google Maps	95
7.1.1	Satelite View	96
7.1.2	Directions	96

7.2.	Google Maps API	96
7.3.	Google Maps Untuk Perangkat Mobile	97
7.4.	Google Maps Android API V2	97
7.4.1.	Menspesifikasi Permissions Pada Android Manifest	98
7.4.2.	Geo Point	100
7.4.3.	Map Fragment	101
7.4.4.	Map View	101
7.4.5.	Tipe-tipe Google Maps Android API	101
7.4.6.	Cara Mengubah Map Type	102
7.4.7.	Marker Manager	103

BAB 8 :Android Wireless Socket

8.1	Sejarah Socket	104
8.1.1	Jenis Socket	104
8.1.2	Sekilas Tentang Socket, TCP dan UDP	105
8.2	Model Aplikasi Client Server	109
8.3	Kelas Tambahan	109



BRIEFING

PENGENALAN ANDROID, JAVA, XML DAN ADT

B.1 Android

Android merupakan sebuah sistem operasi terbuka yang diperuntukan untuk perangkat bergerak (mobile device). Dikembangkan oleh Open Handset Alliance yang terdiri dari pengembang software, hardware dan provider seperti Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan NVIDIA yang bertujuan membuat sebuah standar terbuka untuk perangkat bergerak (mobile device). Pada Juli 2005 Android telah diakuisisi oleh Google dan pada 5 November 2007 barulah secara resmi Android dirilis oleh Google.

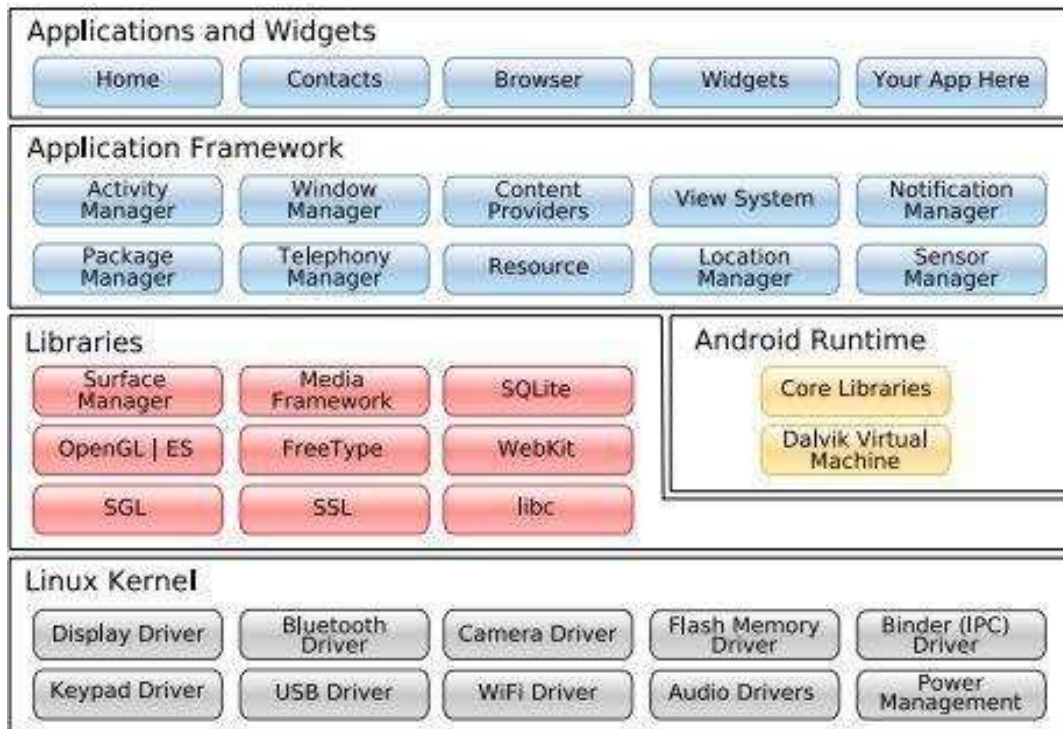


Gambar 1 Logo Android

Dalam pengembangan aplikasi Android menyediakan Android SDK yang menyediakan tools dan API untuk para pengembang aplikasi dengan platform Android. Android menggunakan Java sebagai bahasa pemrogramannya.

B.1.1 Arsitektur Android

Diagram pada Gambar 2 berikut menunjukkan komponen-komponen utama dari sistem operasi Android.



Gambar 2 Arsitektur Sistem Android

Google mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Berikut ini lima susunan dari lapisan-lapisan tersebut jika di lihat dari lapisan dasar hingga lapisan teratas:

1. Linux Kernel

Tumpukan paling bawah pada arsitektur Android ini adalah kernel. Google menggunakan kernel Linux versi 2.6 untuk membangun sistem Android, yang mencakup memory management, security setting, power management, dan beberapa driver hardware. Kernel berperan sebagai abstraction layer antara hardware dan keseluruhan software. Sebagai contoh, HTC GI dilengkapi dengan kamera. Kernel Android terdapat driver kamera yang memungkinkan pengguna mengirimkan perintah kepada hardware kamera.

2. Android Runtime

Lapisan setelah Kernel Linux adalah Android Runtime. Android Runtime ini berisi Core Libraries dan Dalvik Virtual Machine. Core Libraries mencakup serangkaian inti library Java, artinya Android menyertakan satu set library-library dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada library-library dasar bahasa pemrograman Java.

Dalvik adalah Java Virtual Machine yang memberi kekuatan pada sistem Android. Dalvik VM ini di optimalkan untuk telepon seluler. Setiap aplikasi yang berjalan pada Android berjalan pada processnya sendiri, dengan instance dari Dalvik Virtual Machine. Dalvik telah dibuat sehingga sebuah piranti yang memakainya dapat menjalankan multi Virtual Machine dengan efisien. Dalvik VM dapat mengeksekusi file dengan format Dalvik Executable (.dex) yang telah dioptimasi untuk menggunakan minimal memory footprint. Virtual Machine ini register-based, dan menjalankan class-class yang dicompile menggunakan compiler Java yang kemudian ditransformasi menjadi format .dex menggunakan “dx” tool yang telah disertakan. Dalvik Virtual Machine (VM) menggunakan kernel Linux untuk menjalankan fungsi-fungsi seperti threading dan low-level memory management.

3. Libraries

Bertempat di level yang sama dengan Android Runtime adalah Libraries. Android menyertakan satu set library-library dalam bahasa C/C++ yang digunakan oleh berbagai komponen yang ada pada sistem Android. Kemampuan ini dapat diakses oleh programmer melewati Android application framework. Sebagai contoh Android mendukung pemutaran format audio, video, dan gambar. Berikut ini beberapa core library tersebut :

a. System C library

Diturunkan dari implementasi standard C system library (libc) milik BSD, dioptimasi untuk piranti embedded berbasis Linux.

b. Media Libraries

Berdasarkan PacketVideo's OpenCORE; library-library ini mendukung playback dan recording dari berbagai format audio and video populer, meliputi MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.

c. Surface Manager

Mengatur akses pada display dan lapisan composites 2D and 3D graphic dari berbagai aplikasi.

d. LibWebCore

Web browser engine modern yang mensupport Android browser maupun embeddable web view

e. SGL

The underlying 2D graphics engine

f. 3D libraries

Implementasi berdasarkan OpenGL ES 1.0 APIs; library ini menggunakan hardware 3D acceleration dan highly optimized 3D software rasterizer

g. FreeType

Bitmap dan vector font rendering

h. SQLite

Relational database engine yang powerful dan ringan tersedia untuk semua aplikasi.

4. Application Framework

Lapisan selanjutnya adalah application framework, yang mencakup program untuk mengatur fungsi-fungsi dasar smartphone. Application Framework merupakan serangkaian tool dasar seperti alokasi resource smartphone, aplikasi telepon, pergantian antar – proses atau program, dan pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki aplikasi penuh kepada tool-tool dasar tersebut, dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks.

Programmer mendapatkan akses penuh untuk memanfaatkan API-API (Android Protocol Interface) yang juga digunakan core applications. Arsitektur aplikasi didesain untuk menyederhanakan pemakaian kembali komponen-komponen, setiap aplikasi dapat menunjukkan kemampuannya dan aplikasi lain dapat memakai kemampuan tersebut. Mekanisme yang sama memungkinkan pengguna mengganti komponen-komponen yang dikehendaki. Di dalam semua aplikasi terdapat servis dan sistem yang meliputi :

- a. Satu set Views yang dapat digunakan untuk membangun aplikasi meliputi lists, grids, text boxes, buttons, dan embeddable web browser

- b. Content Providers yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain (misalnya Contacts), atau untuk membagi data yang dimilikinya.
 - c. Resource Manager, menyediakan akses ke non-code resources misalnya localized strings, graphics, dan layout files
 - d. Notification Manager yang memungkinkan semua aplikasi untuk menampilkan custom alerts pada the status bar
 - e. Activity Manager yang manage life cycle of dari aplikasi dan menyediakan common navigation backstack
5. Application

Di lapisan teratas tertera aplikasi itu sendiri. Di lapisan inilah anda menemukan fungsi-fungsi dasar smartphone seperti menelepon dan mengirim pesan singkat, menjalankan web browser, mengakses daftar kontak, dan lain-lain. Bagi rata-rata pengguna, lapisan inilah yang paling sering mereka akses. Mereka mengakses fungsi-fungsi dasar tersebut melalui user interface.

B.1.2 Versi Android

1. Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android versi 1.5 (Cupcake)

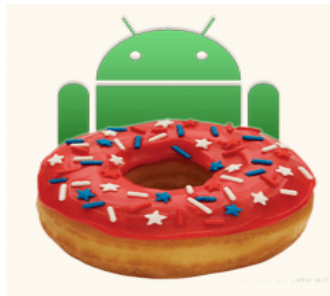
Pada pertengahan Mei 2009, Google kembali merilis telepon seluler menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa fitur yang diperbaharui dan ditambahkan dari versi sebelumnya, yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan.



Gambar 3 Android Cupcake

3. Android versi 1.6 (Donut)

Versi 1.6 (Donut) dirilis pada bulan September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder, galeri yang diintegrasikan, pengadaan resolusi VWGA, dan Text-to-speech engine seperti kemampuan dial kontak. Namun fitur tersebut tidak tersedia pada semua ponsel.



Gambar 4 Android Donut

4. Android versi 2.0/2.1 (Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.



Gambar 5 Android Eclair

5. Android versi 2.2 (Froyo: Frozen Yoghurt)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan umumnya terhadap versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portable, dan kemampuan auto update dalam aplikasi Android Market –sekarang berganti nama menjadi Google Play.



Gambar 6 Android Froyo (Frozen Yoghurt)

6. Android versi 2.3 (Gingerbread)

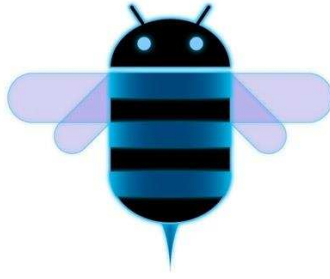
Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.



Gambar 7 Android Gingerbread

7. Android versi 3.0/3.1 (Honeycomb)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb mendukung multi prosesor dan akselerasi perangkat keras untuk grafis.



Gambar 8 Android Honeycomb

8. Android versi 4.0 (ICS :Ice Cream Sandwich)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.



Gambar 9 Android Ice Cream Sandwich

9. Android versi 4.1/4.2/4.3 (JB :Jelly Bean)

Google mengumumkan Android 4.1 (Jelly Bean) dalam konferensi Google I/O pada tanggal 27 Juni 2012. Berdasarkan kernel Linux 3.0.31, Jelly Bean adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuka pengguna (UI). Pembaruan ini diwujudkan dalam "Proyek Butter", perbaikan ini termasuk antisipasi sentuh, triple buffering, perpanjangan waktu vsync, dan peningkatan frame rate hingga 60 fps untuk menciptakan UI yang lebih halus. Android 4.1 Jelly Bean dirilis untuk Android Open Source Project pada tanggal 9 Juli 2012. Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7, yang dirilis pada 13 Juli 2012.

Google berencana merilis Jelly Bean 4.2 pada sebuah acara di New York City pada 29 Oktober 2012, tapi dibatalkan karena Badai Sandy. Jelly Bean 4.2 didasarkan pada kernel Linux 3.4.0, dan pertama kali digunakan pada Nexus 4 LG dan Nexus 10 Samsung, yang dirilis pada 13 November 2012.

Google merilis Jelly Bean 4.3 pada 24 Juli 2013 di San Francisco. Kebanyakan perangkat Nexus menerima pembaruan dengan segera. Nexus 7 generasi kedua adalah perangkat pertama yang menggunakan sistem operasi ini. Sebuah pembaruan minor dirilis pada tanggal 22 Agustus 2013.



Gambar 10 Android Jelly Bean

10. Android versi 4.4 (Kitkat)

Google mengumumkan Android 4.4 KitKat (dinamai dengan izin dari Nestlé dan Hershey) pada 3 September 2013, dengan tanggal rilis 31 Oktober 2013. Sebelumnya, rilis berikutnya setelah Jelly Bean diperkirakan akan diberi nomor 5.0 dan dinamai 'Key Lime Pie'.



Gambar 11 Android Kitkat

B.1.3 Fitur-fitur Android

Berikut ini berbagai fitur dari sistem operasi Android:

1. Rancangan handset. Platform disesuaikan dengan kebutuhan VGA (Video Graphics Adapter) yang lebih besar, library grafis 2D dan 3D yang berdasarkan pada spesifikasi OpenGL ES 1.0 serta layout smartphone yang tradisional.
2. Penyimpanan. Android menggunakan software database SQLite sebagai penyimpanan data.
3. Konektivitas. Android mendukung berbagai teknologi konektivitas, seperti GSM (Global System for Mobile Communications) /EDGE (Enhanced Data rates for GSM Evolution), CDMA (Code Division Multiple Access), EV-DO (Evolution-Data Optimized), UMTS (Universal Mobile Telecommunications System), bluetooth dan Wi-Fi (Wireless Fidelity).
4. Pesan. Android mendukung pengiriman pesan dalam bentuk SMS (Short Message Service) dan MMS (Multimedia Messaging Service).
5. Web browser. Web browser yang tersedia di Android berdasarkan pada framework aplikasi open source WebKit.
6. Dukungan Java. Software yang ditulis dalam bahasa Java dapat dikompilasi dan dieksekusi pada mesin virtual Dalvik, yang merupakan implementasi dari VM (Virtual Machine) yang dirancang khusus untuk penggunaan perangkat bergerak.
7. Dukungan media. Android mendukung beberapa format audio/video seperti: H.263, H.264 (dalam kontainer 3GP atau MP4), MPEG-4 SP, AMR, AMR-WB (dalam kontainer 3GP), AAC, HE-AAC (dalam kontainer MP4 atau 3GP), MP3, MIDI, OGG Vorbis, WAV, JPEG, PNG, GIF dan BMP.
8. Dukungan hardware lainnya. Android mendukung penggunaan kamera, layar sentuh, GPS (Global Positioning System), accelerometers, magnetometers, akselerasi 2D bit blits (dengan orientasi hardware, scaling, konversi format piksel) dan akselerasi grafis 3D.
9. Lingkungan Development. Android menyediakan perangkat emulator, debugger, memori dan profil kinerja yang tersedia pada plugin Eclipse IDE (Integrated Development Environment).
10. Market. Mirip dengan App Store pada iPhone OS, Android Market adalah sebuah katalog aplikasi yang dapat di-download dan diinstal pada telepon seluler secara online, tanpa

menggunakan PC (Personal Computer).Awalnya hanya aplikasi gratis saja yang didukung.Dan sejak tanggal 19 Februari 2009 aplikasi berbayar telah tersedia di Android Market untuk Amerika Serikat.

11. Multi-touch. Android memiliki dukungan bawaan untuk multi-touch yang tersedia pada handset terbaru, seperti HTC Hero.Pada awalnya fitur tersebut dinonaktifkan pada level kernel (mungkin untuk menghindari pelanggaran paten terhadap teknologi layar sentuh Apple). Sejak Google merilis update untuk Nexus One dan juga berencana untuk merilis update untuk Motorola Droid yang memungkinkan multi-touch.

B.1.4 Android SDK (Software Development Kit)

Android SDK mencakup perangkat tools pengembangan yang komprehensif. Android SDK terdiri dari debugger, libraries, handset emulator, dokumentasi, contoh kode program dan tutorial. Saat ini Android sudah mendukung arsitektur x86 pada Linux (distribusi Linux apapun untuk desktop modern), Mac OS X 10.4.8 atau lebih, Windows XP atau Vista.Persyaratan mencakup JDK, Apache Ant dan Python 2.2 atau lebih. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan plugin Android Development Tools (ADT), dengan ini pengembang dapat menggunakan IDE untuk mengedit dokumen Java dan XML serta menggunakan peralatan command line untuk menciptakan, membangun, melakukan debug aplikasi Android dan pengendalian perangkat Android (misalnya reboot, menginstal paket perangkat lunak).

B.1.5 Pengembangan Aplikasi Android

Ada 4 hal mendasar yang harus kita pahami dalam membangun aplikasi berbasis Android :

1. Activity, adalah tampilan grafis yang kita lihat ketika menjalankan sebuah aplikasi. Aplikasi dapat memiliki lebih dari satu Activity.
2. Intent, adalah serangkaian nilai yang menunjukkan apa yang harus dilakukan ketika terjadi perpindahan layar.
3. Service, adalah layanan yang bekerja di belakang layar (background).
4. Content provider, memungkinkan sebuah aplikasi untuk dapat menyimpan dan menerima data dari database.

B.1.6 Antarmuka Pengguna Android

Antarmuka Android dapat dibangun melalui dua cara, yaitu dengan menulis kode XML atau dengan menulis kode Java. Penggambaran struktur antarmuka dengan menggunakan kode XML sangat dianjurkan dan lebih baik tentunya. Karena menurut prinsip Model-Viewer-Control bahwa antarmuka pengguna sebaiknya selalu dipisahkan dari logika program. Selain itu, adaptasi sebuah program dari suatu resolusi layar ke resolusi layar lainnya menjadi lebih mudah.

B.1.7 Pemrograman Android

Pemrograman pada Android tidak jauh berbeda dengan pemrograman berbasis java.

A. Pendeklarasian Dalam Pemrograman

Pemrograman tersusun atas tiga deklarasi, yaitu :

a. Deklarasi Paket

Deklarasi paket digunakan untuk mengelompokkan class-class. Sebuah package dapat mempunyai satu atau lebih sub-package ini sehingga dapat menyusun sebuah hirarki. Bentuk umum :

```
package<namapackage>.<namasubpackage>.<namasubsubpackage>;
```

b. Deklarasi Import

Deklarasi import, bersifat opsional, digunakan untuk menunjukkan package atau class yang digunakan dalam sebuah program Java.

Untuk Java, bentuk umumnya :

```
import java.<package>.<subpackage>
```

Untuk Java Android, bentuk umumnya :

```
Import android.<package>.<subpackage>
```

c. Deklarasi Variable

Deklarasi ini dibedakan menurut dua hal yaitu:

– Type Data

- a) Primitive: menyimpan data di memori tempat mereka berada. Contoh:
byte, int, short, long, float
- b) Reference: menyimpan alamat memori dimana data tersimpan di memori tempat mereka berada.

Contoh: String, Integer

- Access Modifier
 - a) Default (no modifier): hanya bisa diakses oleh classnya dan class dalam satu package
 - b) Public : bisa diakses siapa saja
 - c) Private : hanya bisa diakses oleh classnya

B. Operator

Macam operator dalam pemrograman Android meliputi :

a. Operator Aritmatika

Simbol + merupakan simbol operator penambahan Simbol - merupakan simbol operator pengurangan

b. Operator Increment dan Decrement

Dari sisi operator dasar aritmatika, java Android juga terdiri atas operator unary increment (++) dan operator unary decrement (--). Operator increment dan decrement menambah dan mengurangi nilai yang tersimpan dalam bentuk variabel angka terhadap nilai 1. Sebagai contoh, pernyataan :

`count = count + 1; //increment nilai count dengan nilai 1` pernyataan tersebut ekuivalen dengan : `count++;`

c. Operator Relasi

`==` sama dengan

`!=` tidak sama dengan

C. Struktur Perulangan dan Percabangan

a. Perulangan

– For

Bentuk umum : `for(<inisialisasi>, <>, <iterasi>){ //perintah yang diulang }`

b. Percabangan

– if

Bentuk umum :

`If(<ekspresi>){ //perintah yang dijalankan jika benar } else{ //perintah yang dijalankan jika salah }`

– switch

Bentuk umum : `switch(variable){`

```

case<kondisi 1> : //program yang dijalankan break;
case<kondisi 2> : //program yang dijalankan break;
}

```

D. Inisialisasi layout dan widget XML

Inisialisasi ini digunakan pada saat mendeklarasikan widget atau layout yang dipakai. Secara umum, bentuk pendeklarasian layout xml yang diimplementasikan pada suatu class menggunakan method setContentView() seperti dibawah ini :

```
setContentView(<layout file>);
```

sedangkan untuk mendeklarasikan sebuah widget yang akan digunakan untuk keperluan program secara umum seperti dibawah ini :

```
<Object><name> = (<Object>) this.findViewById(<id object in xml file>);
```

E. Deklarasi Intent

Intent, adalah serangkaian nilai yang menunjukan apa yang harus dilakukan ketika terjadi perpindahan layar. Intent dapat membawa nilai dimana nilai tersebut akan digunakan pada activity yang terbuka selanjutnya. Secara umum deklarasi intent seperti dibawah ini :

```
Intent <intent_name> = new Intent(context, <activity for result>);
startActivity(<intent_name>);
```

B.1.8 FileAndroidManifest.xml

Setiap aplikasi Android pasti memiliki sebuah file xml dengan nama AndroidManifest.xml pada direktori root. File tersebut berisi daftar informasi yang dibutuhkan tentang aplikasi yang diberikan kepada sistem Android, informasi sistem ini harus dimiliki sebelum dapat menjalankan kode program.

B.2 Sejarah dan Perkembangan Java

Java adalah bahasa pemrograman Object-oriented yang dibuat dan diperkenalkan pertama kali oleh sebuah tim Sun Microsystem yang dipimpin oleh Patrick Naughton dan James Gosling pada tahun 1991 dengan code name Oak. Pada tahun 1995 Sun mengubah nama Oak tersebut menjadi Java. Ide pertama kali kenapa Java dibuat adalah karena adanya motivasi untuk

membuat sebuah bahasa pemrograman yang bersifat portable dan platform independent (tidak tergantung pada mesin atau sistem operasi) yang dapat digunakan untuk membuat peranti lunak yang dapat ditanamkan (embedded) pada berbagai macam peralatan elektronik konsumen biasa, seperti microwave, remote control, telepon seluler, card reader, dan sebagainya.



Gambar 12 Logo Java

Java berdiri di atas sebuah mesin interpreter yang diberi nama Java Virtual Machine (JVM). JVM inilah yang akan membaca bytecode dalam dokumen .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu, bahasa Java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

Platform Java terdiri dari kumpulan library, JVM, kelas-kelas loader yang dipaket dalam sebuah lingkungan rutin Java dan sebuah compiler, debugger dan tools lain yang dipaket dalam Java Development Kit (JDK). Java 2 adalah generasi yang sekarang sedang berkembang dari platform Java. Agar sebuah program Java dapat dijalankan, maka dokumen dengan ekstensi .java harus dikompilasi menjadi dokumen bytecode. Untuk menjalankan bytecode tersebut dibutuhkan JRE yang memungkinkan pengguna untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi.

Saat ini distribusi Java dan class pendukungnya dibagi dalam tiga bagian yang masing-masing memiliki konsentrasi tersendiri, yaitu:

1. Java 2 Enterprise Edition (J2EE), untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada lingkungan enterprise.
2. Java 2 Standard Edition (J2SE), untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada lingkungan desktop.

3. Java 2 Micro Edition (J2ME), untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada beberapa perangkat seperti telepon seluler, PDA (Personal Digital Assistant) dan Pocket PC.

B.2.1 Kelebihan Java

Sintaks bahasa yang digunakan dalam Java merupakan pengembangan dari bahasa C/C++. Hal ini dilakukan karena bahasa C/C++ merupakan bahasa pemrograman yang telah dikenal oleh banyak pemrogram sehingga mempelajari Java seharusnya bukan merupakan sesuatu yang sulit. Selain itu, bahasa C/C++ juga telah dikenal sebagai bahasa yang handal, bahkan dapat dikatakan era pemrograman modern dimulai oleh bahasa C yang kemudian diteruskan oleh C++ dengan paradigma pemrograman berorientasi objeknya yang terkenal. Segala kelebihan bahasa C/C++ inilah yang dikembangkan lebih lanjut di dalam bahasa Java.

Berikut ini adalah rangkuman kelebihan dari bahasa Java dibandingkan dengan bahasa pemrograman lainnya

1. Bersifat portable dan platform independent. Program Java yang ditulis akan dapat dieksekusi di platform manapun tanpa memerlukan kompilasi ulang (portable) asalkan Java Virtual Machine untuk platform tersebut tersedia.
2. Memiliki garbage collection yang dapat mendealokasi memori secara otomatis. Tidak perlu secara eksplisit membebaskan suatu lokasi memori yang dipakai karena ini akan dilakukan secara otomatis oleh Java.
3. Menghilangkan pewarisan berganda yang terdapat pada C++. Walaupun kelihatannya lebih sebagai suatu kekurangan, namun banyak para ahli yang mengakui bahasa konsep pewarisan berganda lebih banyak mengakibatkan kerugian daripada keuntungan. Java telah didesain sedemikian rupa sehingga teknik ini tidak akan diperlukan dalam pembuatan program apapun.
4. Mengurangi pointer aritmatik. Pengaksesan lokasi memori secara langsung dengan menggunakan pointer memungkinkan program untuk melakukan suatu tindakan yang tidak seharusnya atau tidak boleh dilakukan, untuk mengurangi dan menghilangkan kemungkinan kesalahan seperti ini, penggunaan pointer pada Java telah dibatasi dengan menggunakan reference.

5. Mengurangi kerancuan antara pemberian nilai pada conditional statement. Contohnya, pada penggunaan tanda "=" dengan „==" pada kondisi If.

B.2.2 Karakteristik Java

Selain itu para perancang Java sendiri telah merangkum beberapa karakteristik dari bahasa Java yang telah dikembangkan, dimana karakteristik inilah yang membuat Java menjadi seperti apa yang dikenal sekarang ini sehingga dapat dikatakan sebagai ciri khas dari Java itu sendiri:

1. Berorientasi objek, Java telah menerapkan konsep pemrograman berorientasi objek yang modern dalam implementasinya. Artinya semua aspek yang terdapat di Java adalah objek. Java merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut objek. Hal ini sangat memudahkan pemrogram untuk merancang, membuat, mengembangkan, dan mengalokasi kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir.
2. Robust, Java mendorong pemrograman yang bebas dari kesalahan dengan bersifat strongly typed dan memiliki run-time checking.
3. Portable, program Java dapat dieksekusi di platform manapun selama tersedia JVM untuk platform tersebut.
4. Multithreading, Java mendukung penggunaan multithreading yang telah terintegrasi secara langsung dalam bahasa Java.
5. Dinamis, program Java dapat melakukan suatu tindakan yang ditentukan pada saat eksekusi program dan bukan pada saat kompilasi.
6. Sederhana, Java menggunakan bahasa yang sederhana dan mudah dipelajari.
7. Terdistribusi, Java dirancang untuk berjalan pada lingkungan yang terdistribusi seperti halnya internet.
8. Aman, aplikasi yang dibuat dengan bahasa Java dapat dipastikan keamanannya terutama untuk aplikasi internet.
9. Netral secara arsitektur, Java tidak terikat pada suatu mesin atau sistem operasi tertentu.
10. Interpreted, aplikasi Java dapat dieksekusi pada platform yang berbeda-beda dengan melakukan interpretasi pada byte code.

11. Berkinerja tinggi, byte code Java telah sangat teroptimasi sehingga eksekusi program dapat dilakukan secara cepat sekalipun dilakukan dengan cara interpretasi terhadap byte code.

B.3 XML

XML (eXtensible Markup Language) dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998. Teknologi yang digunakan pada XML sebenarnya bukan teknologi baru, tapi merupakan turunan dari SGML yang telah dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis berbagai proyek berskala besar. Ketika HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting pada SGML dan dengan berpedoman pada pengembangan HTML menghasilkan markup language yang tidak kalah hebatnya dengan SGML.

Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan tag pembuka (diawali dengan „<” dan diakhiri dengan „>”), tag penutup (diawali dengan „</” dan diakhiri „>”) dan atribut elemen (parameter yang dinyatakan dalam tag pembuka misal <form name=”isidata”>). Hanya bedanya, HTML mendefinisikan dari awal tag dan atribut yang dipakai didalamnya, sedangkan pada XML kita bisa menggunakan tag dan atribut sesuai kehendak kita. Untuk lebih jelasnya lihat contoh dibawah:

Tag yang ada pada xml buat sendiri sesuai keinginan kita. Sampai di sini XML tidak melakukan apapun. Yang ada hanyalah informasi yang di kemas dengan tag XML. Kita harus membuat software lagi untuk untuk mengirim, menerima atau menampilkan informasi di dalamnya.

B.3.1 Keunggulan XML

XML untuk saat ini bukan merupakan pengganti HTML. Masing-masing dikembangkan untuk tujuan yang berbeda. Kalau HTML digunakan untuk menampilkan informasi dan berfokus pada bagaimana informasi terlihat, XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri. XML terutama dibutuhkan untuk menyusun dan menyajikan informasi dengan format yang tidak mengandung format standard layaknya heading, paragraf, tabel dan lain sebagainya.

Sama dengan HTML, file XML berbentuk teks sehingga bila diperlukan kita bisa membacanya tanpa memerlukan bantuan software khusus. Hal ini memudahkan pengembang aplikasi yang menggunakan XML untuk men-debug programnya. XML lebih fleksibel dibanding HTML dalam hal kemampuannya menyimpan informasi dan data. Pada XML kita bisa menyimpan data baik dalam atribut maupun sebagai isi elemen yang diletakkan diantara tag pembuka dan tag penutup.

Kelebihan lain yang dimiliki XML adalah bahwa informasi bisa di pertukarkan dari satu sistem ke sistem lain yang berbeda platform. Misalnya dari Windows ke Unix, atau dari PC ke Machintosh bahkan dari internet ke ponsel dengan teknologi WAP.

B.3.2 Perintah Dasar XML Android

XML Android pada dasarnya berbeda dengan XML secara umum. Pada XML Android, nama-nama tag telah didefinisikan oleh Android SDK, tidak seperti XML pada umumnya yang dapat menuliskan nama-nama tag sesuai keinginan. Beberapa tag XML Android yang sering digunakan dapat dilihat pada tabel 1 dibawah.

Tabel 1. Tag XML Android

No	Perintah	Fungsi
1	<AbsolutLayout	digunakan jika hendak menggunakan sistem koordinat x dan y untuk tata letak widget (berdasar piksel)
2	<LinearLayout	jika jenis ini yang digunakan, widget akan terurut secara vertikal berdasarkan urutan penulisan widget
3	<RelativeLayout	tata letak widget relatif, bergantung widget lainnya
4	<Button	membuat widget tombol
5	<ImageButton	membuat widget tombol gambar
6	<EditText	membuat widget kolom penulisan teks
7	<TextView	membuat widget teks
8	<ImageView	membuat widget untuk menampilkan gambar

9	<Spinner	membuat widget spinner (combo box)
---	----------	------------------------------------

Selain nama-nama tag, pada XML Android juga menyediakan perintah untuk mengatur widget-widget tersebut yang dituliskan di dalam tag-tag tersebut. Perintah perintah yang sering digunakan antara lain ada pada Tabel 2.

Tabel 2. Perintah XML untuk mengatur widget

No	Perintah	Fungsi
1	android:id	memberikan nama id
2	android:layout_height	mengatur tinggi widget
3	android:layout_weight	mengatur lebar widget
4	android:layout_x	mengatur posisi widget terhadap sumbu x
5	android:layout_y	mengatur posisi widget terhadap sumbu y
6	android:text	memberikan teks
7	android:textSize	mengatur ukuran teks
8	android:textStyle	mengatur gaya teks
9	android:textColor	mengatur warna teks
10	android:gravity	mengatur posisi widget secara vertikal
11	android:layout_gravity	mengatur posisi widget terhadap layar
12	android:layout_width	mengatur lebar layar
13	android:layout_height	mengatur tinggi layar
14	android:background	memberi background pada layar

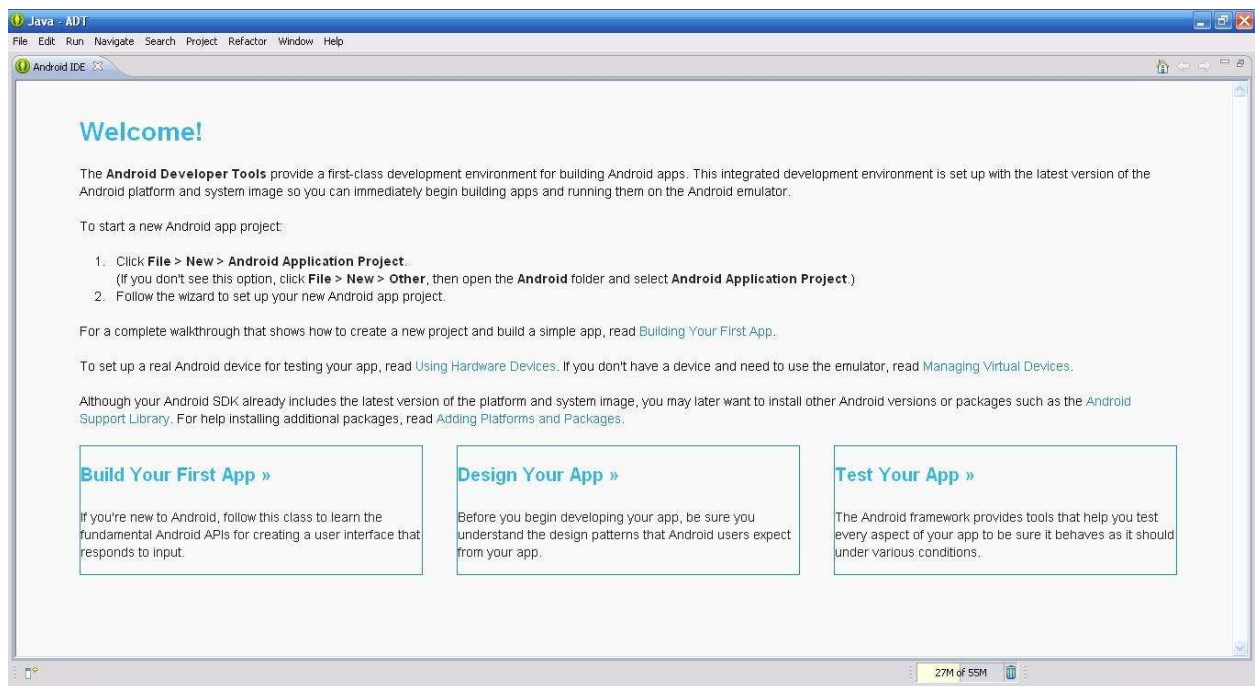
B.4 ADT (Android Development Tools) Bundle

ADT adalah sebuah plugin untuk mengintegrasikan IDE Eclipse dengan SDK Android, sehingga IDE Eclipse dapat digunakan untuk membuat aplikasi Android. Tetapi sekarang google telah menyatukan ADT, IDE Eclipse, dan SDK Android dalam 1 paket yang disebut ADT Bundle. Dengan menggunakan ADT Bundle membuat aplikasi menjadi semakin mudah, karena kita hanya perlu mengekstrak ADT Bundle yang telah kita punya. Sebelum dapat menggunakan ADT Bundle, kita membutuhkan JDK yang terinstall dalam PC kita.

JDK dapat didownload di : <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

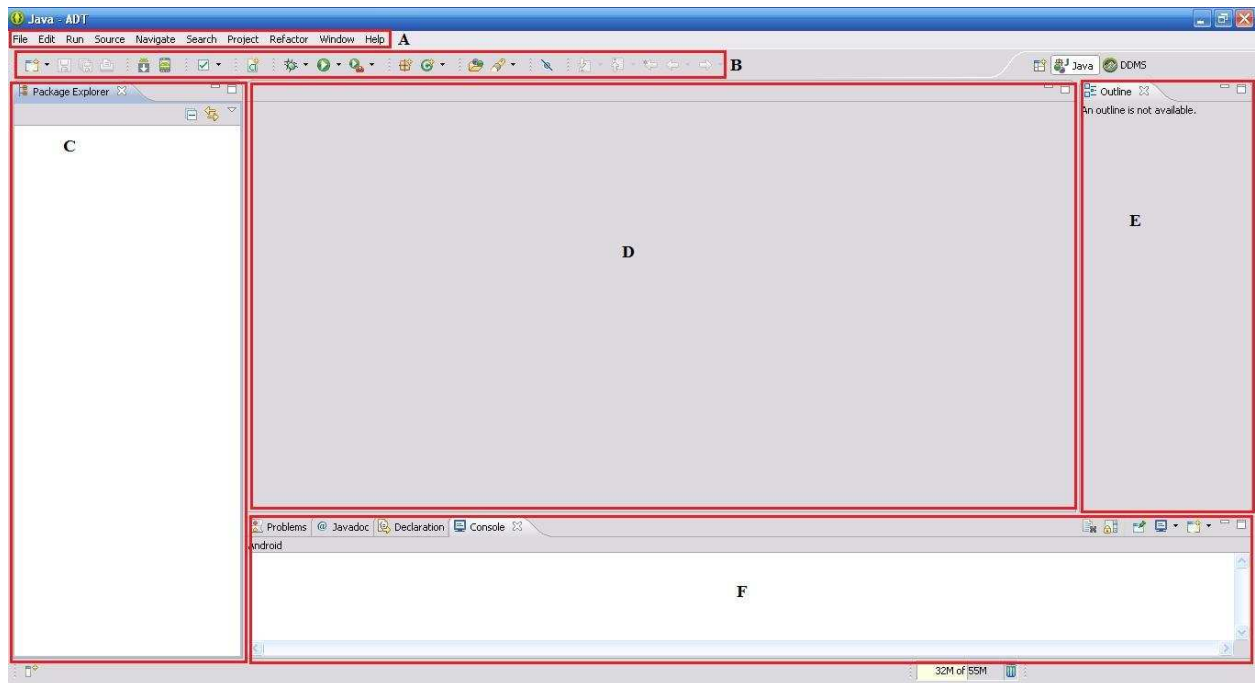
ADT Bundle dapat didownload di : <http://developer.android.com/sdk/index.html>

Jika sudah mendownload ADT Bundle maka kita hanya perlu mengekstraknya saja, kemudian kita jalankan eclipse.exe. Ketika menjalankan eclipse.exe akan muncul konfirmasi path project klik saja OK sehingga muncul gambar seperti gambar 13



Gambar 13 Welcome Screen eclipse yang ada pada ADT bundle

B.4.1 Interface Eclipse



Gambar 14 Main Screen eclipse yang ada pada ADT bundle

A. Main Menu

Main Menu ini terdiri dari dua komponen yaitu title bar dan menu bar. Title bar (baris judul atau balok judul) digunakan untuk menampilkan nama project Eclipse yang sedang dikerjakan, sedangkan menubar (baris menu atau balok menu) digunakan untuk menampilkan semua menu utama Eclipse dan semua perintah pengoperasian untuk menjalankan aksi tertentu.

Terdapat sembilan menu pada Eclipse yaitu File, Edit, Source, Navigate, Search, Project, Refactor, Window, dan Help.

B. Toolbar

Toolbar ini digunakan untuk menampilkan serangkaian icon sebagai alternatif yang lebih cepat dalam pengaksesan perintah-perintah Eclipse. Terdapat dua puluh satu buah Toolbar pada Eclipse yaitu:

- New (dengan berbagai macam pilihan diantaranya untuk membuat File, Package, Class, Project Eclipse baru dan lainnya).

- Save (untuk menyimpan Project, File mau pun hal lainnya yang telah dibuat pada Eclipse).
- Print (untuk mencetak Project mau pun File yang telah dibuat pada Eclipse).
- Opens the Android SDK and AVD Manager (untuk mengatur Android SDK maupun AVD manager yang digunakan untuk Project Android).
- Opens a wizard to help create a new Android project (untuk membuat Project baru Android).
- Opens a wizard to help create a new Android Test project (untuk membuat Project Test baru Android).
- Opens a wizard to help create a new Android XML File (untuk membuat File XML baru Android).
- Debug (untuk mencari kesalahan Project yang telah dibuat pada Eclipse).
- Run (untuk menjalankan Project utama Eclipse).
- Run Last Tool (berkaitan dengan konfigurasi tools pada Eclipse).
- New Java Project (untuk membuat Project baru Java).
- New Java Package (untuk membuat Package baru Java).
- New Java Class (untuk membuat Class baru Java).
- Open Type (untuk membuka Project pada Eclipse dengan terlebih dahulu mencari project berdasarkan masukan user).
- Open Task (untuk membuka Task pada Eclipse dengan terlebih dahulu mencari project berdasarkan masukan user).
- Search (untuk mencari File, Task mau pun Project Java dengan parameter tertentu).
- Next Annotation (lanjut ke anotasi berikutnya).
- Previous Annotation (kembali ke anotasi sebelumnya).
- Last Edit Location
- Back (untuk kembali ke Project sebelumnya yang aktif pada Eclipse).
- Forward (untuk lanjut ke Project berikutnya yang aktif pada Eclipse).

C. Package Explorer dan Hierarchy

Package Explorer berisi list Project mau pun File yang diaktifkan pada Eclipse sedangkan Hierarchy menampilkan hierarki atau tingkatan dari File mau pun Project pada Eclipse berdasarkan tipe tertentu.

D. Area Kerja

Area kerja menampilkan koding dari Project yang dibuat.

E. Outline

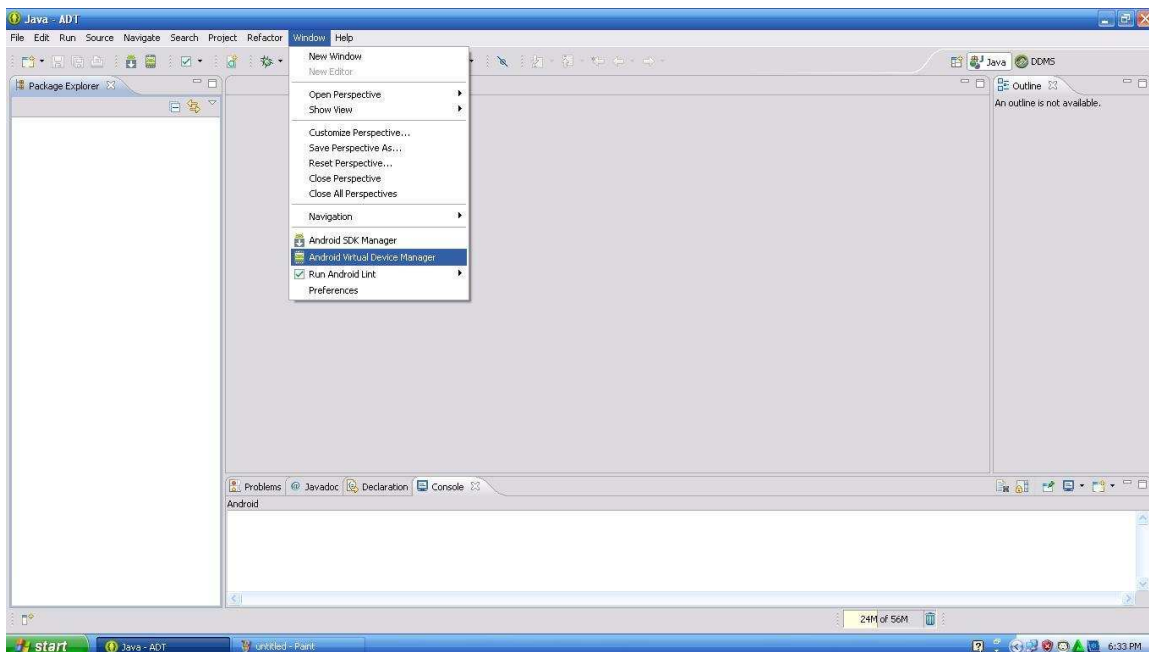
Outline menampilkan komponen-komponen yang ada di Project aktif pada Eclipse.

F. Problem, Javadoc, Declaration Console

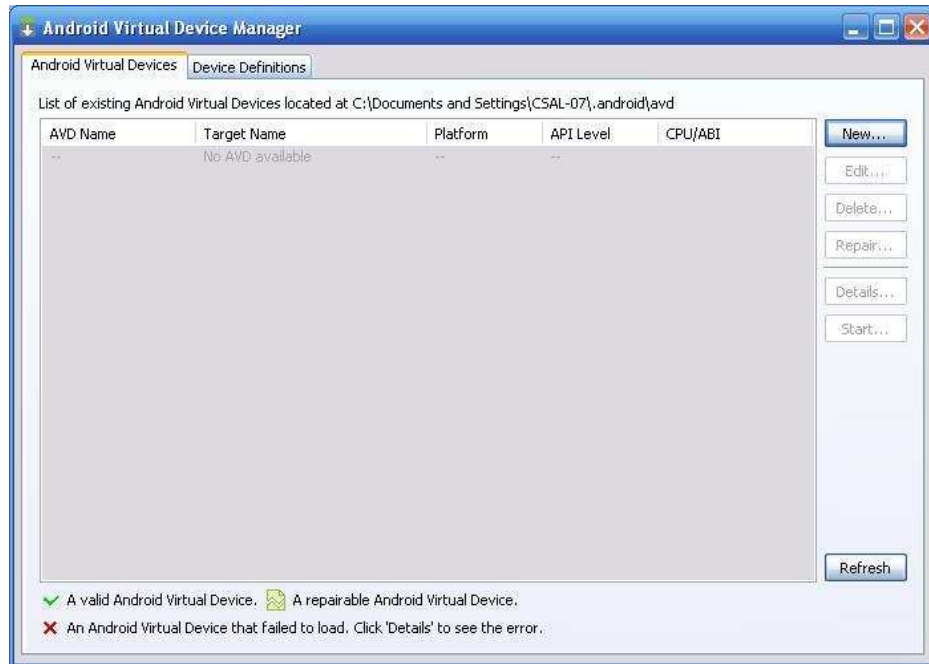
Problem menampilkan kesalahan yang terdapat pada Project saat Project di jalankan, Javadoc menampilkan dokumen Java yang terdapat pada Project aktif, dan Declaration menampilkan deklarasi yang terdapat pada Project aktif.

B.4.2 Emulator Android

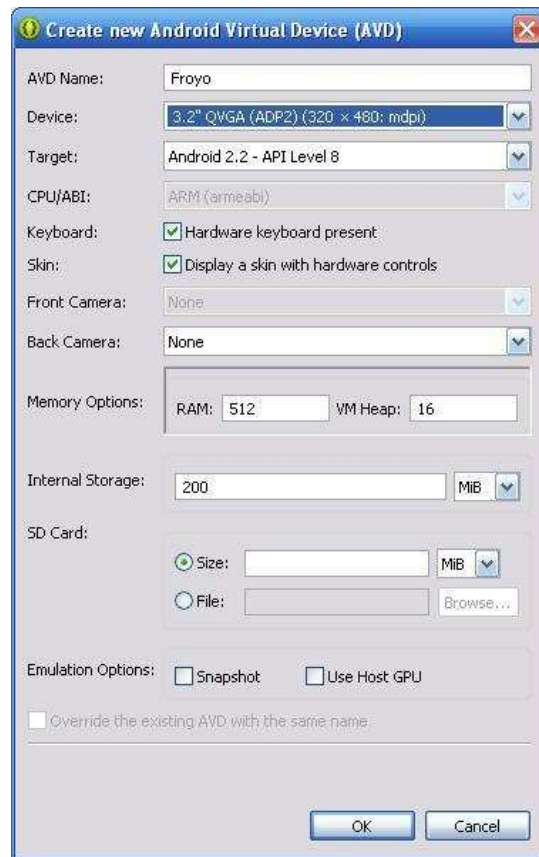
Pilih Windows > Android Virtual Device Manager. Liat gambar 15 dan gambar 16



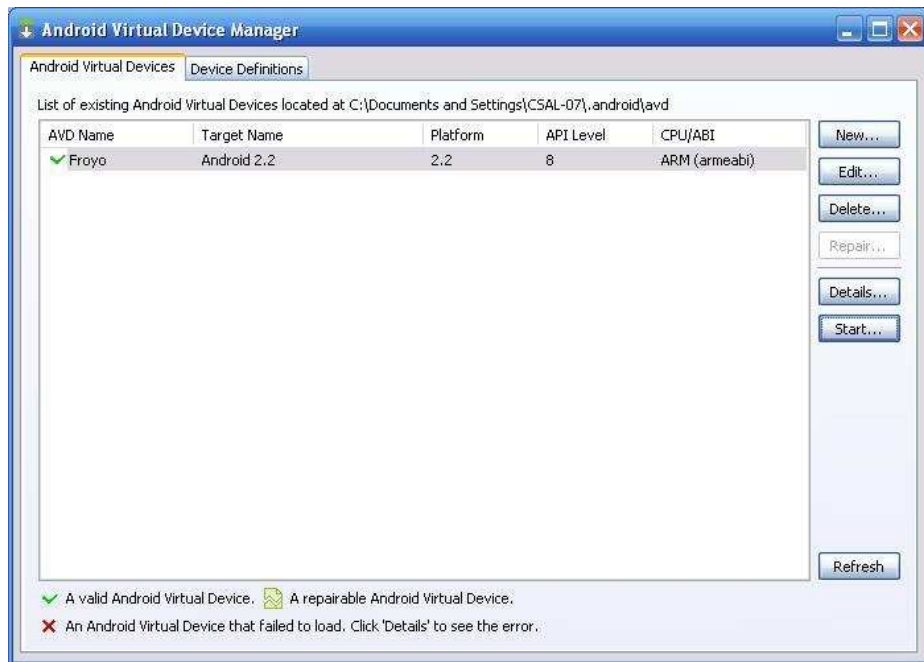
Gambar 15 Pilih AVD(Android Virtual Device) Manager



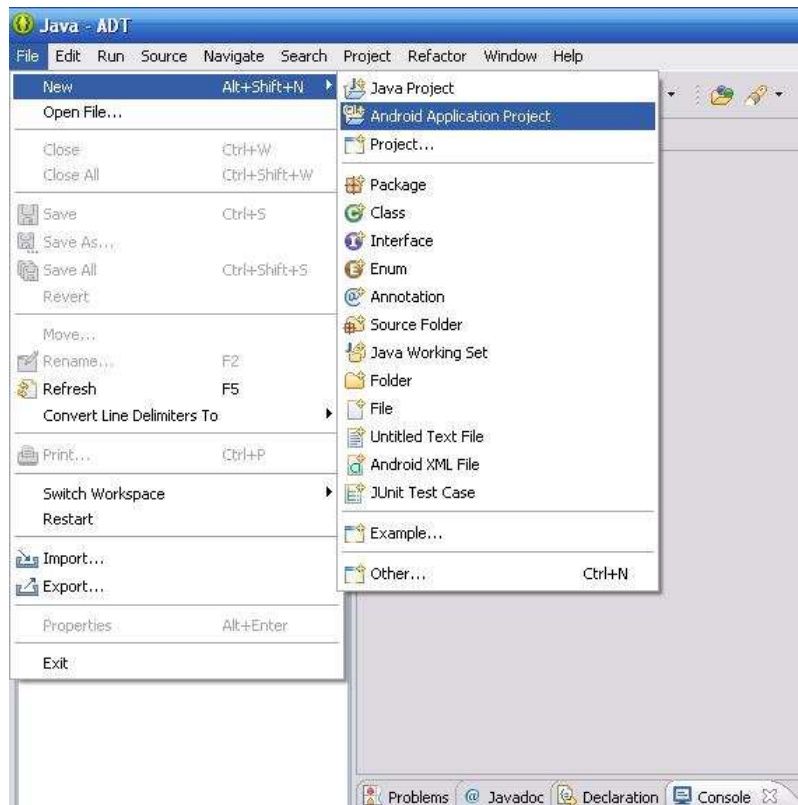
Gambar 16 AVD(Android Virtual Device) Manager



Gambar 17 Create new AVD



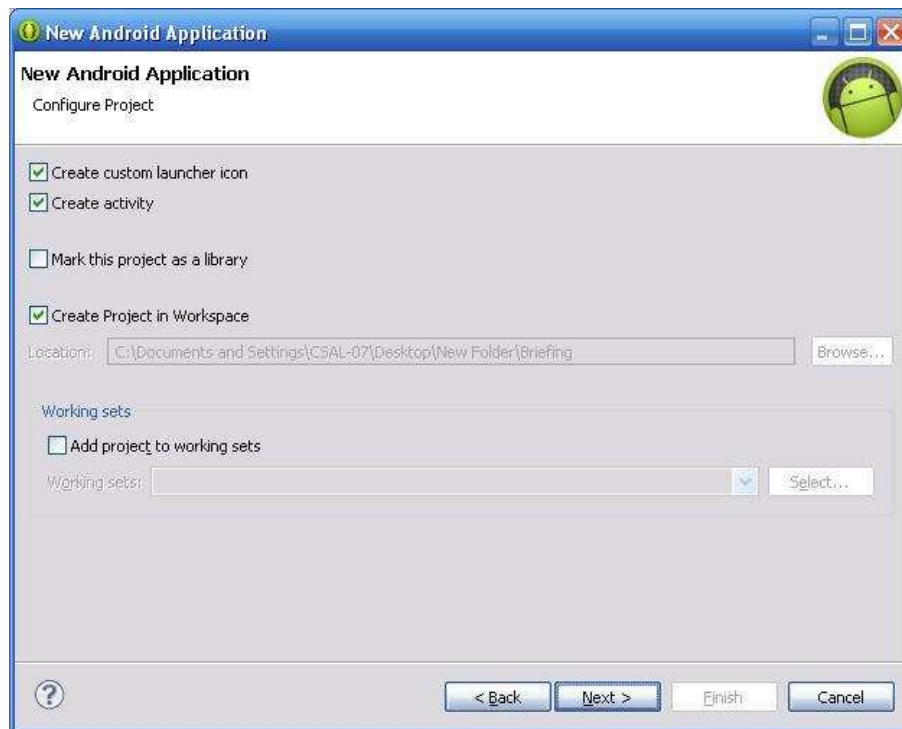
Gambar 18 AVD(Android Virtual Device) Manager



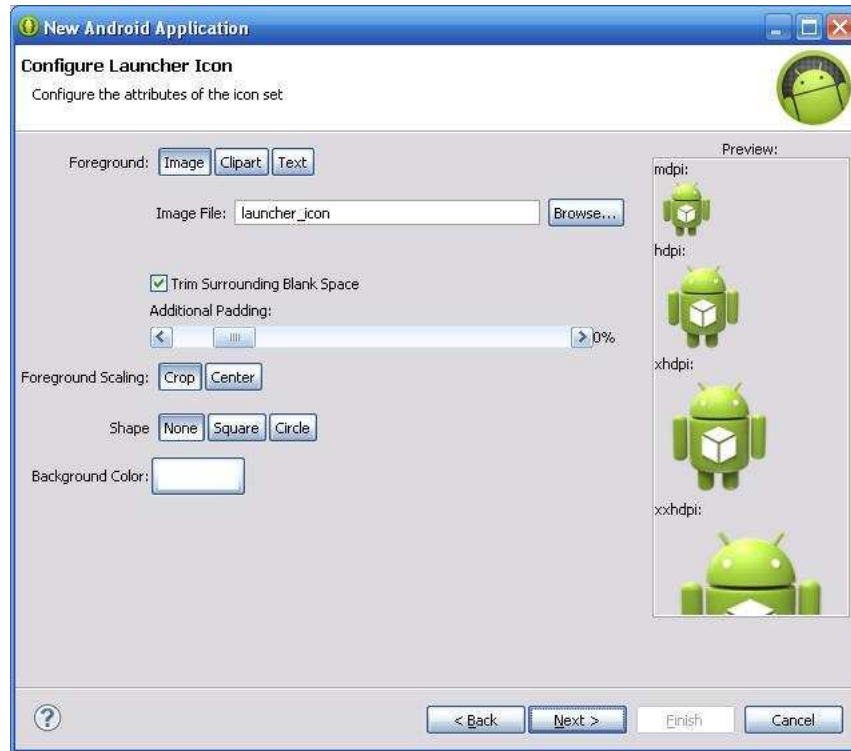
Gambar 19 New Android Project



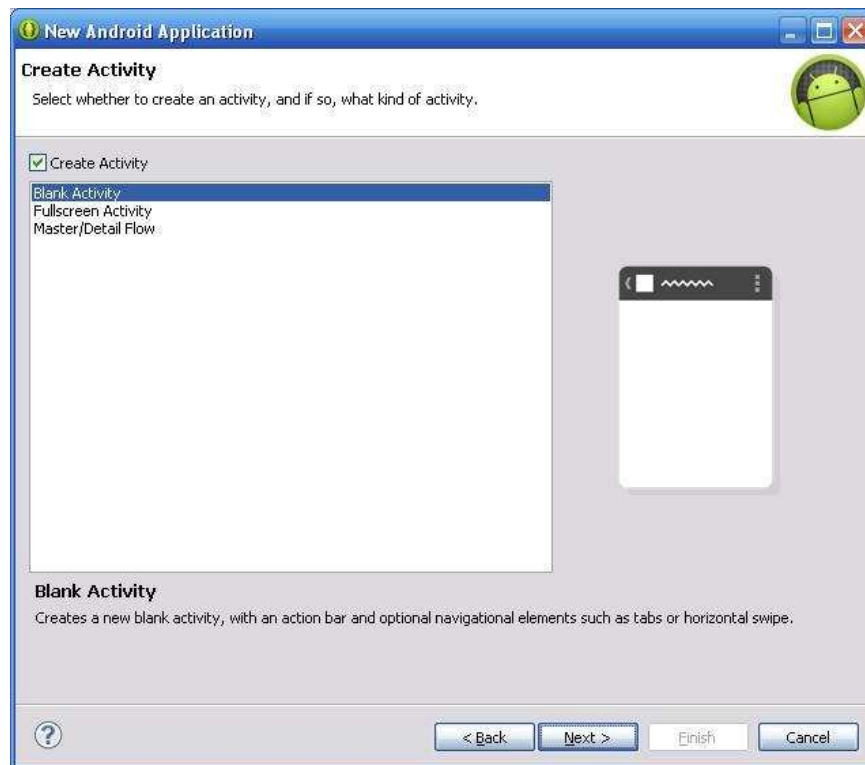
Gambar 20 New Android Application



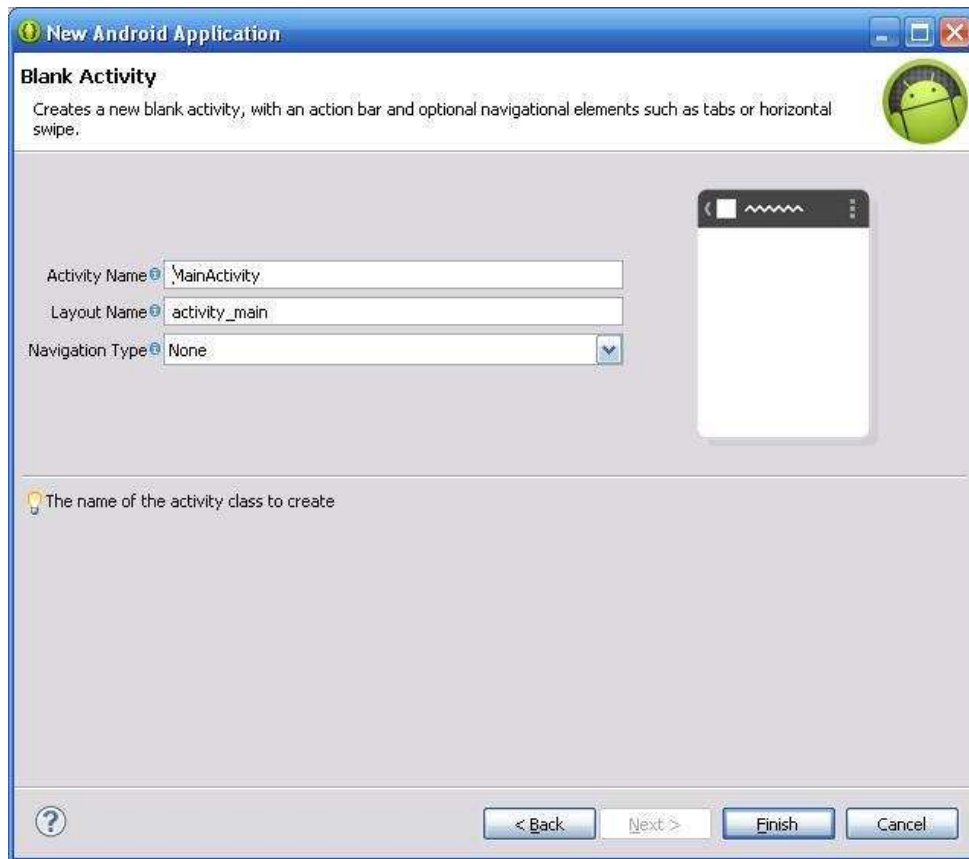
Gambar 21 New Android Application



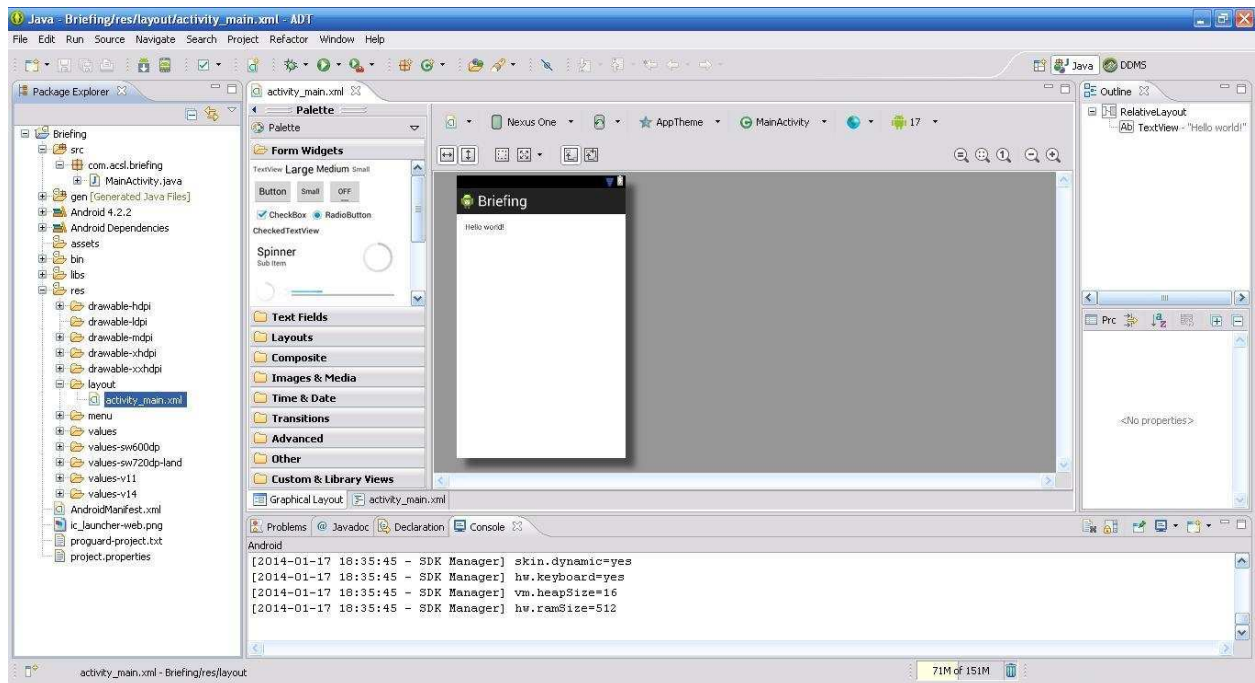
Gambar 22 New Android Application



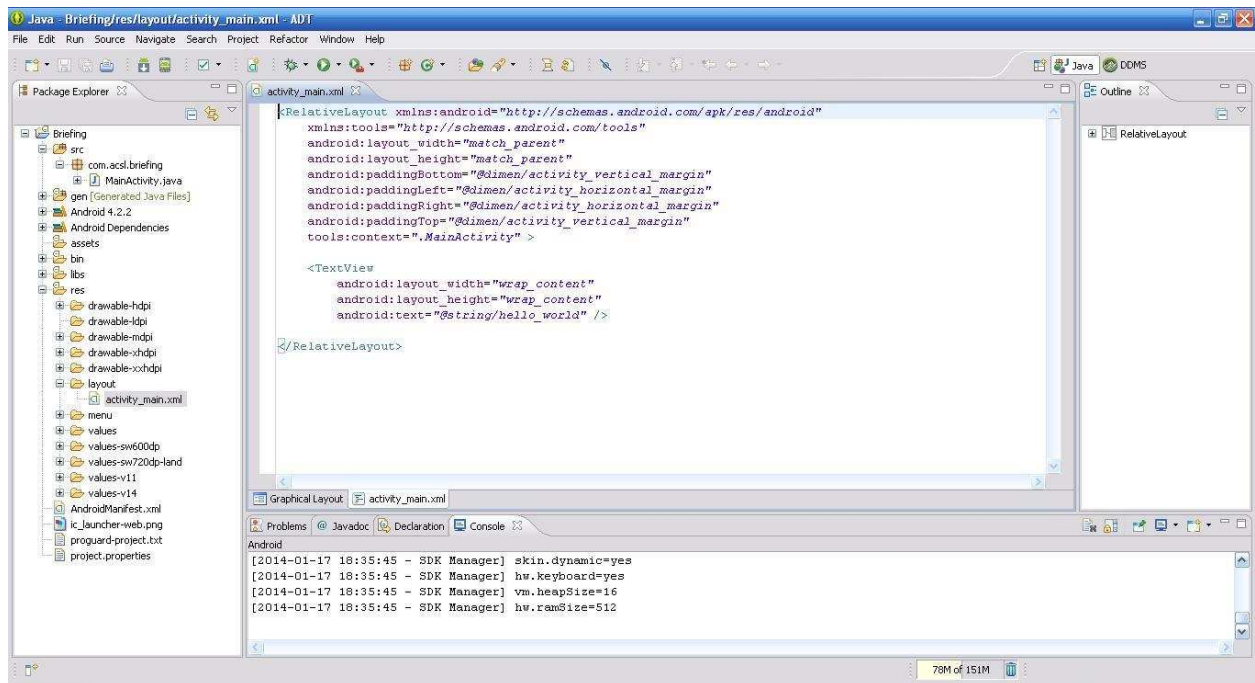
Gambar 23 New Android Application



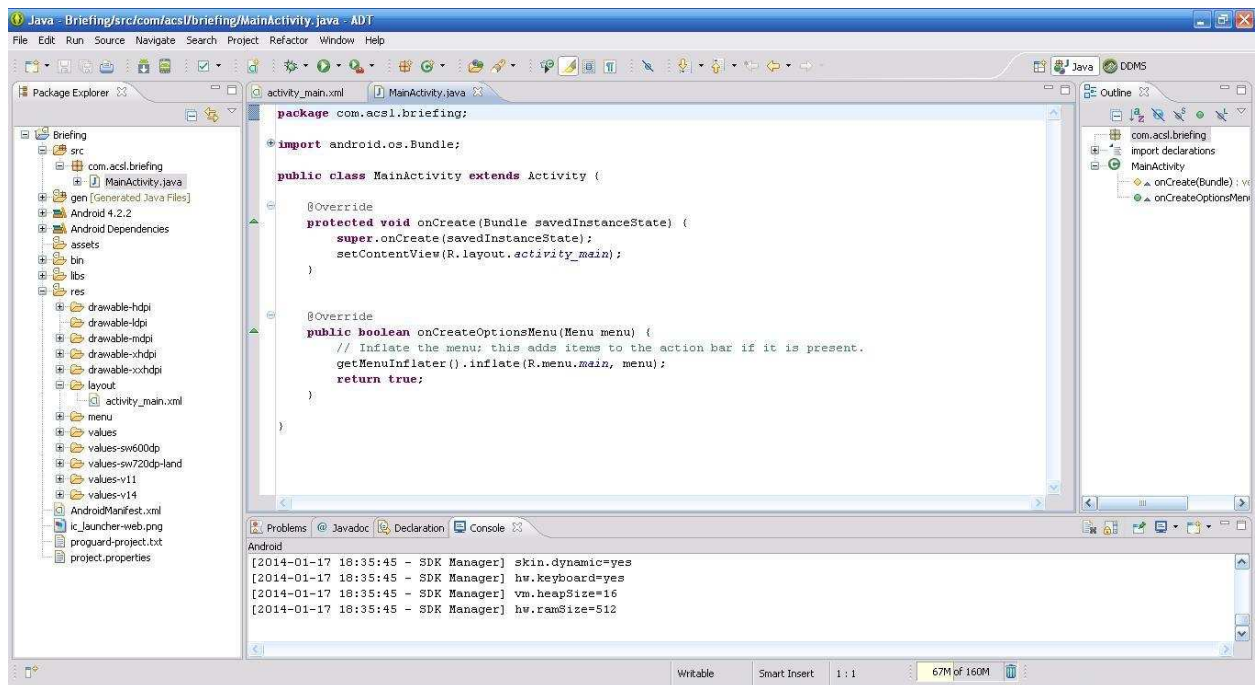
Gambar 24 New Android Application



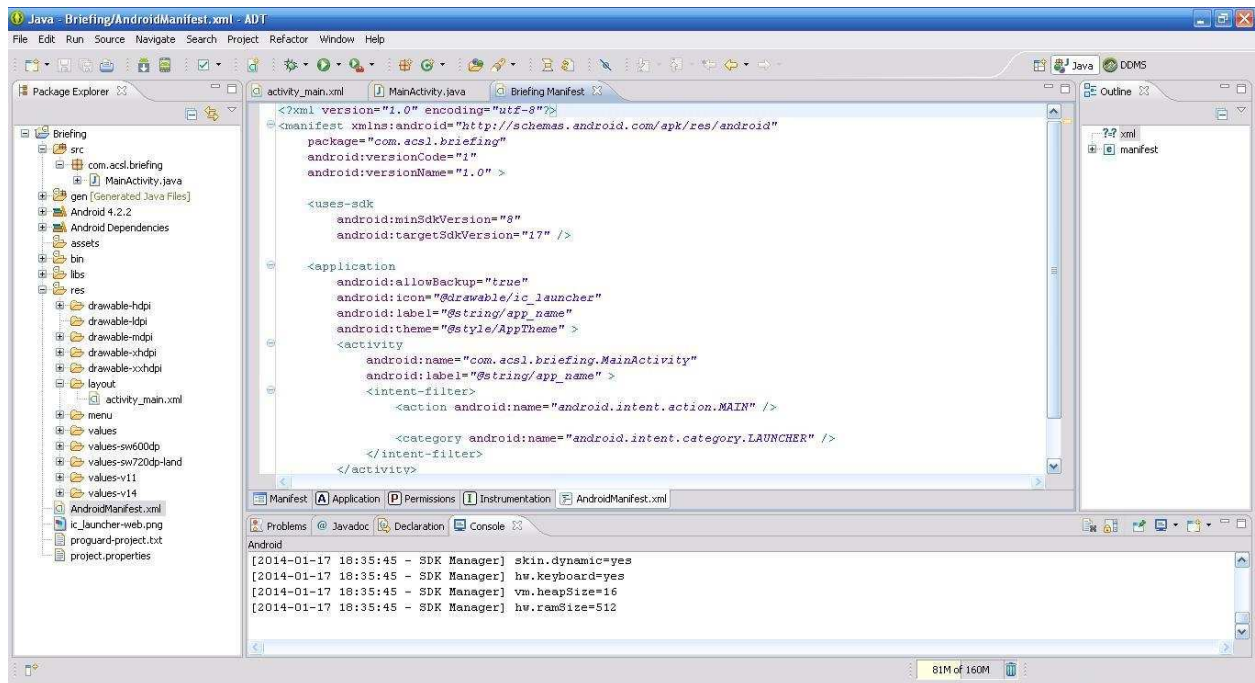
Gambar 25 activity_main.xml GUI



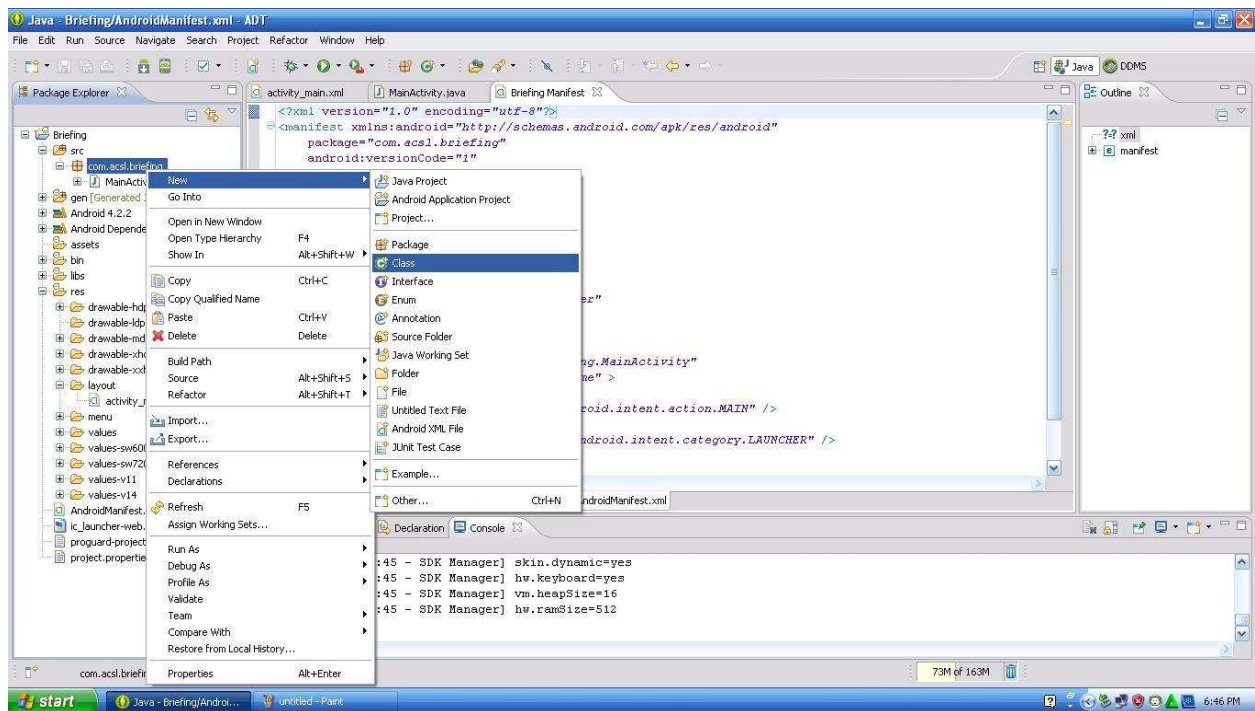
Gambar 26 activity_main.xml



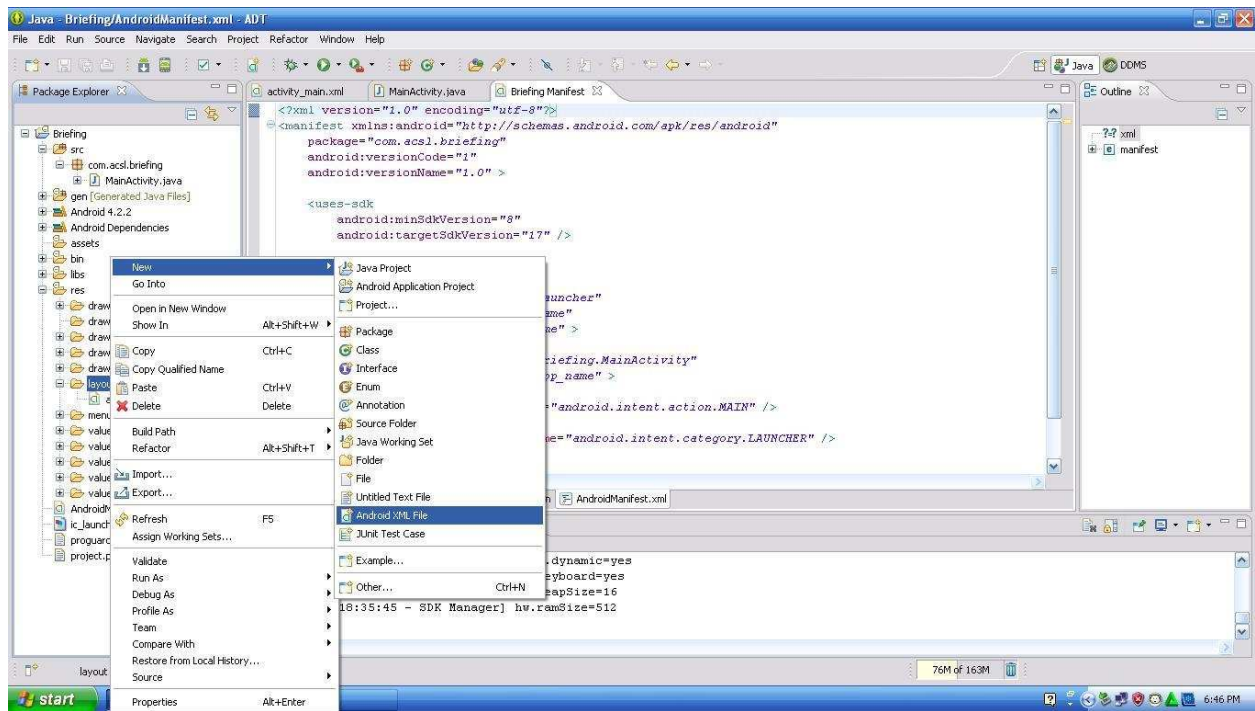
Gambar 27 MainActivity.java



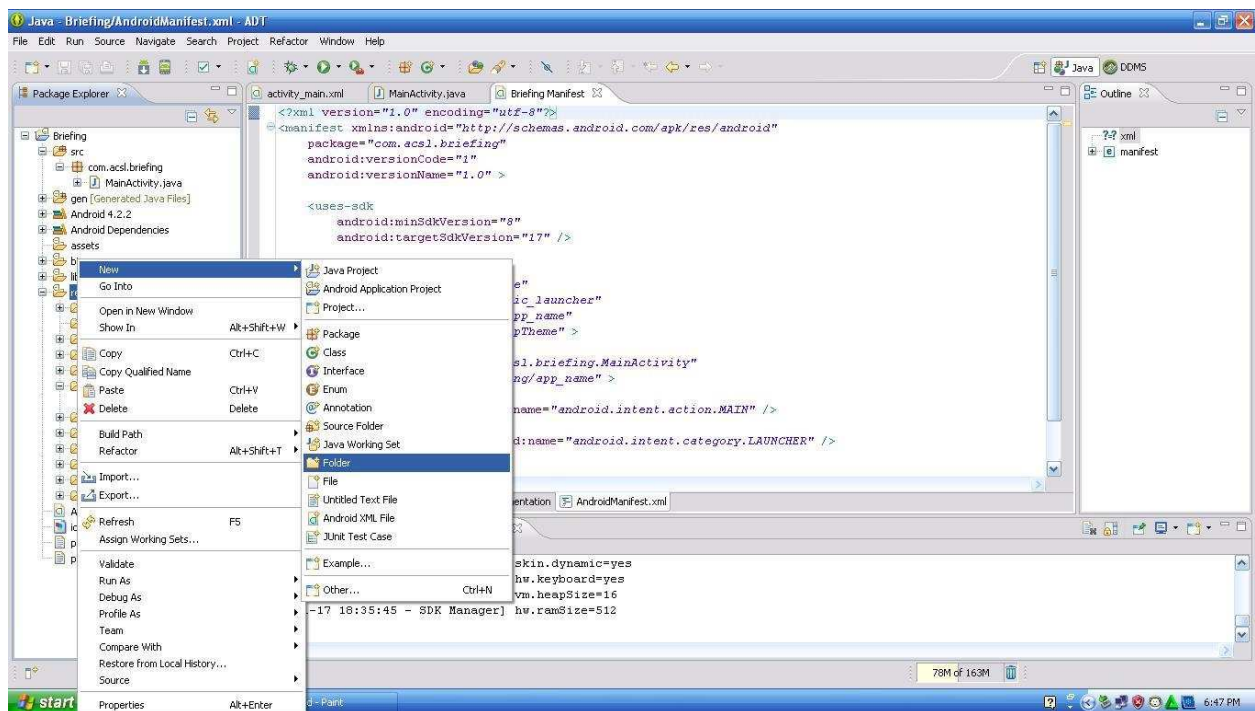
Gambar 28 Briefing Manifest.xml



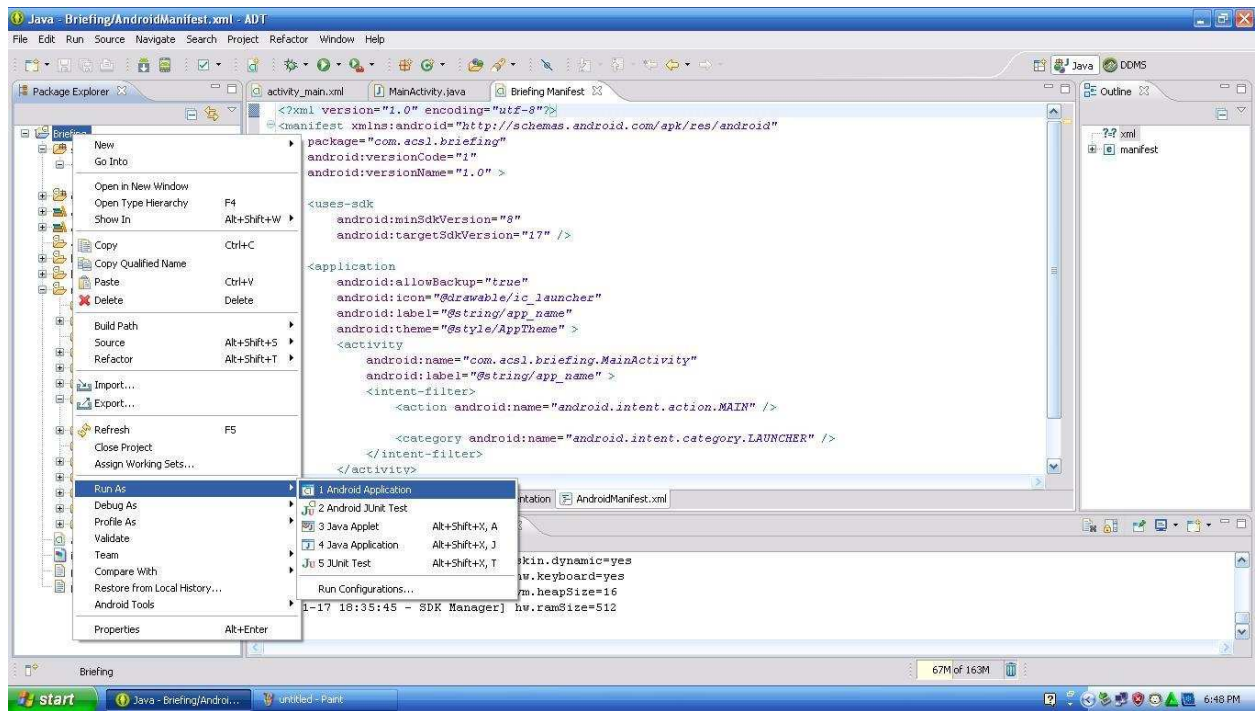
Gambar 29 Create New Java File



Gambar 30 Create New XML File



Gambar 31 Create New Folder



Gambar 32 Run Android Application

1.1 Android

Android merupakan sebuah sistem operasi terbuka yang diperuntukkan untuk perangkat bergerak (mobile device). Android dikembangkan oleh Open Handset Alliance yang terdiri dari pengembang software, hardware dan provider seperti Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan NVIDIA yang bertujuan membuat sebuah standar terbuka untuk perangkat bergerak (mobile device). Pada Juli 2005 Android telah diakuisisi oleh Google dan pada 5 November 2007 barulah secara resmi Android dirilis oleh Google.

Dalam pengembangan aplikasi, Android menyediakan Android SDK yang menyediakan tools dan API untuk para pengembang aplikasi dengan platform Android. Android menggunakan Java sebagai bahasa pemrogramannya.

1.1.1 Fitur Android

Berikut ini berbagai fitur dari sistem operasi Android :

1. Rancangan *handset*

Platform disesuaikan dengan kebutuhan VGA (Video Graphics Adapter) yang lebih besar, library grafis 2D dan 3D yang berdasarkan pada spesifikasi OpenGL ES 1.0 serta layout smartphone yang tradisional.

2. Penyimpanan

Android menggunakan software database SQLite sebagai penyimpanan data.

3. Konektivitas

Android mendukung berbagai teknologi konektivitas, seperti GSM (Global System for Mobile Communications) / EDGE (Enhanced Data rates for GSM Evolution), CDMA (Code Division Multiple Access), EV-DO (Evolution-Data Optimized), UMTS (Universal Mobile Telecommunications System), bluetooth dan Wi-Fi (Wireless Fidelity).

4. Pesan

Android mendukung pengiriman pesan dalam bentuk SMS (Short Message Service) dan MMS (Multimedia Messaging Service).

5. Web browser

Web browser yang tersedia di Android berdasarkan pada framework aplikasi open source WebKit.

6. Dukungan Java

Software yang ditulis dalam bahasa Java dapat dikompilasi dan dieksekusi pada mesin virtual Dalvik, yang merupakan implementasi dari VM (Virtual Machine) yang dirancang khusus untuk penggunaan perangkat bergerak.

7. Dukungan media

Android mendukung beberapa format audio/videoseperti: H.263, H.264 (dalam kontainer 3GP atau MP4), MPEG-4 SP, AMR, AMR-WB (dalam kontainer 3GP), AAC, HE-AAC (dalam kontainer MP4 atau 3GP), MP3, MIDI, OGG Vorbis, WAV, JPEG, PNG, GIF dan BMP.

8. Dukungan hardwarelainnya

Android mendukung penggunaan kamera, layar sentuh, GPS (Global Positioning System), accelerometers, magnetometers, akselerasi 2D bit blits(dengan orientasi hardware, scaling, konversi format piksel) dan akselerasi grafis 3D.

9. Lingkungan Development

Android menyediakan perangkat emulator, debugger, memori dan profil kinerja yang tersedia pada pluginEclipse IDE (Integrated Development Environment).

10. Market

Mirip dengan App Storepada iPhone OS, Android Marketadalah sebuah katalog aplikasi yang dapat di-downloaddan diinstal pada telepon seluler secara online, tanpa menggunakan PC (Personal Computer). Awalnya hanya aplikasi gratis saja yang didukung. Dan sejak tanggal 19 Februari 2009 aplikasi berbayar telah tersedia di Android Marketuntuk Amerika Serikat.

11. Multi-touch

Android memiliki dukungan bawaan untuk multi-touchyang tersedia pada handsetterbaru, seperti HTC Hero. Pada awalnya fitur tersebut dinonaktifkan pada levelkernel (mungkin untuk menghindari pelanggaran paten terhadap teknologi layar sentuh Apple). Sejak Google

merilis update untuk Nexus One dan juga berencana untuk merilis update untuk Motorola Droid yang memungkinkan multi-touch.

1.2 Android SDK (Software Development Kit)

Android SDK mencakup perangkat tools pengembangan yang komprehensif. Android SDK terdiri dari debugger, libraries, handset emulator, dokumentasi, contoh kode program dan tutorial. Saat ini Android sudah mendukung arsitektur x86 pada Linux (distribusi Linux apapun untuk desktop modern), Mac OS X 10.4.8 atau lebih, Windows XP atau Vista. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan plugin Android Development Tools (ADT), dengan ini pengembang dapat menggunakan IDE untuk mengedit dokumen Java dan XML serta menggunakan peralatan command line untuk menciptakan, membangun, melakukan debug aplikasi Android dan pengendalian perangkat Android (misalnya reboot, menginstal paket perangkat lunak).

Android SDK telah dirilis pada tanggal 12 November 2007. Dan pada tanggal 15 Juli 2008 tim Android Developer Challenge dengan sengaja mengirimkan e-mail ke semua pendatang baru di Android Developer Challenge untuk mengumumkan bahwa rilis SDK terbaru telah tersedia pada halaman download. E-mail tersebut juga ditujukan kepada pemenang Android Developer Challenge putaran pertama. Sebuah pernyataan bahwa Google telah menyediakan rilis SDK terbaru untuk beberapa pengembang dan bukan untuk orang lain.

Pada tanggal 18 Agustus 2008, Android SDK 0.9 beta dirilis. Rilis ini menyediakan API yang telah diperbarui dan diperluas, perbaikan pada tools pengembangan dan desain terbaru untuk tampilan home. Petunjuk untuk meng-upgrade SDK sudah tersedia pada rilis sebelumnya. Pada tanggal 23 September 2008, Android 1.0 SDK telah dirilis. Pada tanggal 9 Maret 2009, Google merilis versi 1.1 untuk telepon seluler Android. Rilis terbaru tersebut termasuk dukungan untuk pencarian dengan suara, harga aplikasi, perbaikan jam alarm, perbaikan pengiriman e-mail pada gmail, perbaikan notification dan peta.

Pada pertengahan Mei 2009, Google merilis versi 1.5 (Cupcake) pada sistem operasi Android dan SDK. Pembaruan ini termasuk pada dukungan beberapa fitur baru seperti perekaman video, bluetooth, sistem keyboard pada layar dan pengenalan suara. Rilis ini juga membuka AppWidget framework kepada para pengembang yang memungkinkan untuk

membuat widget sendiri pada tampilan home. Pada September 2009 versi 1.6 (Donut) dirilis yang dapat menampilkan hasil pencarian yang lebih baik dan penggunaan indikator baterai.

1.3 Android Development Tools (ADT)

Untuk membuat aplikasi pada Android dengan menggunakan Eclipse, Android menawarkan custom plugin untuk Eclipse IDE yang disebut Android Development Tools (ADT). ADT dirancang untuk memberikan lingkungan yang terintegrasi untuk membangun aplikasi Android. ADT meningkatkan kemampuan Eclipse untuk membuat pengguna lebih cepat membuat project Android, membuat aplikasi UI, menambahkan komponen berdasarkan Android Framework API dan men-debug aplikasi Anda menggunakan SDK Android tools.

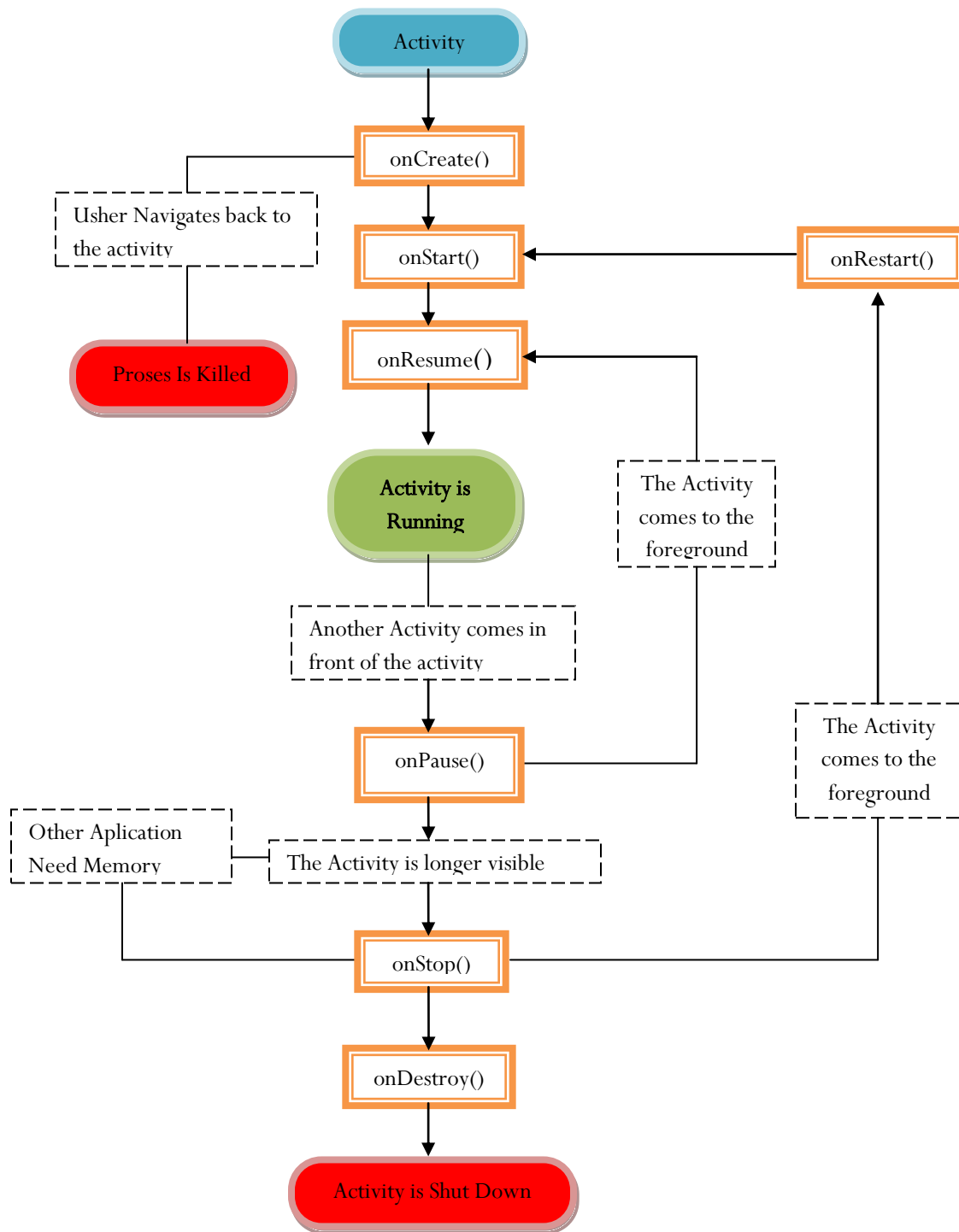
1.4 Pengembangan Aplikasi Android

Ada 4 jenis komponen yang harus kita pahami dalam membangun aplikasi berbasis Android yaitu :

1.4.1 Activities

Activities adalah tampilan grafis yang kita lihat ketika menjalankan sebuah aplikasi. Aplikasi dapat memiliki lebih dari satu Activity. Aplikasi mungkin terdiri hanya dari satu Activity, tetapi ada banyak aplikasi yang terdiri dari beberapa Activity. Activity apa saja dan berapa banyak Activity tersebut, tentu saja tergantung pada aplikasi dan desain.

Biasanya satu Activity akan dipakai sebagai satu UI yang pertama diperlihatkan pada pengguna saat aplikasi diaktifkan. Untuk pindah dari satu Activity ke Activity yang lainnya, kita dapat melakukan satu event misalnya mengklik tombol atau memilih opsi tertentu.



Gambar 1.1 : Siklus Aktivitas aplikasi Android

1.4.2 Service

Service adalah layanan yang bekerja di belakang layar (background) untuk jangka waktu tak terbatas dan tidak memiliki visual interface (UI). Misalnya, service mungkin memainkan musik pada background atau mungkin mengambil data melalui jaringan dan memberikan hasil untuk Activity yang membutuhkannya. Setiap service meng-extends kelas induk Service.

Sebagai contoh ketika media player memutar lagu dari daftar lagu, aplikasi player akan mungkin memiliki satu atau lebih Activity yang memungkinkan pengguna untuk memilih lagu dan memainkannya. Namun player itu sendiri tidak akan ditangani oleh sebuah Activity karena pengguna akan berharap musik tetap bermain bahkan setelah mereka meninggalkan player dan memulai sesuatu yang lain, misalnya mengirim pesan sms. Untuk menjaga musik tetap berjalan, Activity media player dapat menjalankan service untuk membuat musik tetap berjalan. Sistem kemudian akan membuat service tetap running, bahkan setelah Activity yang menjalankannya ditutup dari layar.

1.4.3 *Content provider*

Content provider memungkinkan sebuah aplikasi untuk dapat menyimpan dan menerima data dari database. Data dapat disimpan dalam file sistem, dalam sebuah database SQLite atau cara lainnya. Content provider meng-extends kelas dasar ContentProvider untuk menerapkan standar method yang memungkinkan aplikasi lain untuk mengambil dan menyimpan data dari jenis yang dikontrolnya. Namun aplikasi ini tidak memanggil metode langsung.

Sebaliknya mereka menggunakan Objek ContentResolver dan memanggil metode tersebut. ContentResolver dapat berkomunikasi dengan setiap Content provider yang bekerjasama dengan provider untuk mengelola setiap komunikasi interprocess yang terlihat.

Content provider menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu Activity misalnya ketika kita menggunakan suatu aplikasi berbasis peta (map) . Setiap kali ada permintaan yang harus ditangani oleh komponen tertentu, Android akan memastikan bahwa proses aplikasi dari komponen berjalan, memulainya jika diperlukan, dan mengumumkan ketersediaan objek instant dari komponen, juga menciptakan objek instant jika diperlukan.

1.4.4 *Broadcast receiver*

Broadcast receiver merupakan komponen yang tidak melakukan apapun selain menerima dan bereaksi untuk menyiarkan notifikasi (announcements). Banyak broadcast berasal dari dalam sistem kode misalnya notifikasi zona waktu yang telah berubah , bahwa baterai low, gambar telah diambil oleh kamera, atau pengguna mengubah preferens bahasa. Aplikasi juga dapat menginisiasi broadcast, misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah didownload ke perangkat dan tersedia bagi mereka untuk digunakan.

Aplikasi dapat mempunyai sejumlah broadcast receivers untuk merespon setiap notifikasi dan menanggapi jika penting. Semua receivers meng-extend kelas induk BroadcastReceiver. Broadcast receivers tidak menampilkan UI. Namun mereka dapat memulai sebuah Activity untuk merespon informasi yang mereka terima, atau mungkin menggunakan NotificationManager mengingatkan pengguna. Notifikasi bisa membuat pengguna melihat informasi yang disampaikan dalam berbagai cara- flashing lampu latar (backlight), menggetarkan (vibrating) perangkat, pemutaran suara, dan lainnya.

1.5 Pemrograman Android

a. Deklarasi *Package*

Deklarasi package digunakan untuk mengelompokkan class-class. Sebuah package dapat mempunyai satu atau lebih sub-package sehingga dapat menyusun sebuah hirarki.

Bentuk umum :

```
package<namapackage>.<namasubpackage>.<namasubsubpackage>;
```

b. Deklarasi *import*

Deklarasi import bersifat opsional, digunakan untuk menunjukkan package atau class yang digunakan dalam sebuah program Java. Untuk Java, bentuk umumnya :

```
import java.<package>.<subpackage>
```

Untuk Java Android, bentuk umumnya :

```
import android.<package>.<subpackage>
```

No	Package	Method	Keterangan
1	android.app.Activity		Mendukung sebuah <i>class</i> sebagai <i>activity</i> pada aplikasi
		onCreate()	dipanggil saat <i>activity</i> pertama kali dibuat
		setContentView()	mengatur sebuah <i>activity</i> menggunakan sebuah <i>layout</i> pada <i>resource</i>
		onSaveInstanceState()	Dipanggil untuk mengambil per- <i>instance state</i> dari suatu <i>activity</i> sebelum dimatikan sehingga <i>state</i> dapat dikembalikan dalam <code>OnCreate(Bundle)</code> atau <code>onRestoreInstanceState(Bundle)</code> (<code>Bundle</code> dipopulasikan <i>method</i> ini akan diteruskan untuk keduanya).
		startActivity()	membuka sebuah <i>activity</i> baru
		finish()	dipanggil untuk menutup sebuah <i>activity</i>

2	android.app.AlertDialog	Mendukung penggunaan <i>alert dialog</i>	
		show()	menampilkan <i>alert dialog</i>
		Builder()	membuat sebuah <i>alert dialog</i>
		setMessage()	Mengatur isi pesan pada <i>alert dialog</i>
		setTitle()	mengatur judul <i>alert dialog</i>
		setCancelable()	mengatur jika tombol pada <i>alert dialog</i> ditekan tidak akan terjadi apa-apa
		setNeutralButton	menambahkan tombol yang hanya memiliki satu pilihan
3	android.content.DialogInterface	Mendukung penggunaan Antarmuka Dialog	
		showDialog()	menampilkan dialog
4	android.media.MediaPlayer	Mendukung penggunaan <i>Media Player</i>	
		create()	membuat sebuah <i>media player</i>
		start()	digunakan untuk menjalankan <i>media player</i>
5	android.os.Bundle	Mendukung penggunaan <i>class</i> dapat berjalan pada sistem operasi Android	
6	android.view.View	Mendukung penggunaan sebuah <i>file</i> xml sebagai tampilan	
		findViewById()	Menemukan <i>view</i> yang

			diidentifikasi oleh atribut <i>id</i> dari XML yang kemudian diproses dengan <code>OnCreate()</code>
7	<code>android.view.View.OnClickListener</code>	Mendukung sebuah <i>view</i> dapat menerima perintah eksekusi klik	
		<code>setOnClickListener()</code>	menerapkan penanganan <i>event</i> klik sebuah <i>view</i>
8	<code>android.widget.AdapterView</code>	Mendukung penggunaan sebuah <i>array adapter</i>	
9	<code>android.widget.AutoCompleteTextView</code>	Mendukung penggunaan <i>widget autocomplete</i>	
10	<code>android.widget.Button</code>	Mendukung penggunaan <i>widget Button</i>	
11	<code>android.widget.ImageView</code>	Mendukung penggunaan <i>widget ImageView</i>	
		<code>setImageResource()</code>	Digunakan untuk mengatur sumber gambar
12	<code>android.widget.TextView</code>	Mendukung penggunaan <i>widget TextView</i>	
13	<code>android.app.ProgressDialog</code>	Mendukung penggunaan <i>Progress Bar (ProgressDialog)</i>	
		<code>show()</code>	Menampilkan <i>Progress Dialog</i>
		<code>dismiss()</code>	Mengabaikan dialog yang sebelumnya ditampilkan melalui <code>showDialog()</code>
14	<code>android.content.Intent;</code>	Mendukung penggunaan <i>Intent</i>	
15	<code>android.widget.RadioButton</code>	Mendukung penggunaan <i>widget RadioButton</i>	

16	android.content.ContentValues	Mendukung penggunaan <i>Content Values</i>
17	android.content.Context	Mendukung penggunaan <i>Context</i> untuk menerapkan sebuah <i>class</i> pada <i>class</i> lain
18	android.database.Cursor	Mendukung penggunaan sebuah kursor basis data
	query()	memberikan <i>query</i> pada tabel, lalu mengembalikan hasilnya sebagai <i>cursor</i>
	rawQuery()	menjalankan SQL yang diberikan dan mengembalikan kursor sebagai hasilnya.
19	android.database.SQLException	Mendukung <i>Exception</i> untuk <i>statement</i> SQL
20	android.database.sqlite.SQLiteDatabase;	Mendukung penggunaan SQLite sebagai basis data
	execSQL()	mengeksekusi satu <i>statement</i> SQL yang bukan <i>query</i>
	open()	membuka <i>database</i>
	close()	menutup <i>database</i>
21	android.database.sqlite.SQLiteOpenHelper;	Mendukung penggunaan OpenHelper SQLite
22	java.util.ArrayList;	digunakan untuk mendukung penggunaan sebuah <i>arraylist</i>

23	import java.util.List;	digunakan untuk mendukung penggunaan sebuah <i>list</i>
		add() menambahkan nilai ke dalam <i>list</i> .

Tabel 2.1 Daftar *Package* beserta *method* bawaannya

1.6 Android Manifest

File AndroidManifest.xml diperlukan oleh setiap aplikasi Android. Diletakkan di folder root aplikasi, dan mendeskripsikan nilai global paket aplikasi kita. Termasuk komponen yang ada pada aplikasi (activity, service, dll) yang memang diekspos untuk sistem diluar aplikasi yang membutuhkan. Termasuk juga data pada tiap Activity kita, bagaimana menanganinya dan meluncurkan (launched).

Hal terpenting yang harus diperhatikan dari file AndroidManifest.xml adalah intentFilters. intentFilter menjelaskan dimana dan kapan activity dapat dijalankan. Ketika Activity (atau sistem operasi) akan melakukan sesuatu, misalnya membuka halaman web atau membuka layar kontak, itu akan membuat objek Intent.

Objek Intent ini akan menangani beberapa informasi yang mendeskripsikan apa yang akan dilakukan oleh aplikasi dan data apa yang dibutuhkan untuk itu, selain itu juga beberapa informasi lain. Android membandingkan informasi pada objek Inten dengan intent filter, yang dibuka (exposed) oleh setiap aplikasi dan mencari Activity (aplikasi) yang tepat untuk menangani data atau aksi (action) yang dibutuhkan oleh Activity yang memanggilnya (caller). Jika terdapat aplikasi lebih dari satu yang menangani intent, pengguna (user) akan ditanya (melalui pesan/notifikasi) mana yang akan dipilih.

Selain mendeklarasikan Activity Aplikasi, Content Provider, Service dan Intent Receivers, kita juga harus menetapkan permissions didalam AndroidManifest.xml. Contoh sederhana dari AndroidManifest.xml :

```

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

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".HelloActivity"
            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>

```

Gambar 1.2 : Kode xml dari AndroidManifest

- Hampir setiap AndroidManifest.xml (seperti juga file Android XML yang lain) akan memasukkan deklarasi namespace (`xmlns:android=http://schemas.android.com/apk/res/android`) pada elemen pertama. Ini membuat beberapa atribut standar Android tersedia didalam file, yang akan digunakan untuk mensupply kebanyakan data untuk element didalam file.
- Hampir semua manifest memasukkan satu tag `<application>`, yang berisi beberapa tag yang menjelaskan tentang Application, IntentReceiver, dll yang ada pada aplikasi.
- Jika kita ingin Activity dapat langsung dibuka oleh user, kita harus membuatnya mendukung MAIN dan LAUNCHER.

Penjelasan dari masing tag pada struktur file AndroidManifest :

<manifest>

Ini adalah titik root utama dari AndroidManifest.xml. berisi atribut paket (package-atribute), yang menjadi titik dari semua paket pada Activity. Path Activity lain secara relative akan mengacu pada nilai ini.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.hello"
```

<uses-permission>

Menjelaskan tentang security permission, yang harus diberikan agar aplikasi / Activity bisa secara besar beroperasi (misal ketika mengirim sms atau menggunakan daftar kontak tlp). Permissi ini diberikan kepada user ketika menginstal aplikasi.

```
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

<permission>

Mendeklarasikan security permission yang bisa digunakan untuk membatasi aplikasi yang mana yang bisa diakses komponennya atau fiturnya didalam paket aplikasi kita (atau di paket aplikasi lain).

<instrumentation>

Mendeklarasikan kode dari instrumen yang tersedia untuk menguji fungsionalitas dari paket aplikasi atau paket aplikasi yang lain.

<application>

Elemen root yang berisi deklarasi dari level aplikasi komponen yang terdapat pada paket aplikasi. Elemen ini juga bisa berisi global dan/atau atribut default dari aplikasi, seperti label, ikon, theme, permission yang dibutuhkan, dll.

```
<application android:icon="@drawable/icon"
```

<activity>

Activity adalah hal yang mendasar (primer) pada sebuah aplikasi untuk berinteraksi dengan user. Catatan : setiap Activity harus memiliki sebuah <activity> pada file manifest apakah ini untuk diekspor keluar atau diperuntukkan hanya untuk paket aplikasi itu sendiri. Jika Activity tidak memiliki tag yang sesuai didalam manifest, kita tidak bisa meluncurkannya.

<intent filter>

Mendeklarasikan intent seperti apa yang didukung oleh komponen. Sebagai tambahan kita bisa menempatkan nilai pada elemen ini, atribut bisa diberikan disini untuk mensupply label, ikon dan informasi lain untuk aksi (action) akan dilakukan.

```
<intent-filter>
```

<action>

Tipe aksi (action-type) yang didukung oleh komponen. Contoh :

```
<actionandroid:name="android.intent.action.MAIN"/>
```

<category>

Kategori (category-type) yang didukung oleh komponen.

```
<categoryandroid:name="android.intent.category.LAUNCHER"/>
```

<data>

Tipe MIME, skema URI, authority URI, atau path URI yang didukung komponen

<meta-data>

Menambahkan bagian baru (new piece) metadata kepada activity, dimana client akan mendapatkan nilainya kembali.

```
ComponentInfo.metadata.
```

<receiver>

Sebuah IntentReceiver mengijinkan apliasi untuk diberikan informasi mengenai perubahan data atau aksi yang terjadi, bahkan jika sedang tidak berjalan (running).

```
<receiver android:name=".SMSReceiver">
```

<service>

Service adalah komponen yang dapat berjalan pada background. Seperti tag activity, secara optional kita dapat menempatkan satu atau lebih elemen <intent-filter> yang didukung oleh service atau nilai <meta-data>.

<provider>

ContentProvider adalah komponen yang mengelola data (persistent data) dan mempublisnya untuk diakses oleh aplikasi lain. Kita secara optional dapat meletakkan satu atau lebih nilai <meta-data>, seperti yang dijelaskan pada activity <meta-data>. Semua tag harus ditutup, misal </application>, </intent-filter>, </activity> atau ditutup secara langsung <action android:name="android.intent.action.MAIN"/>.

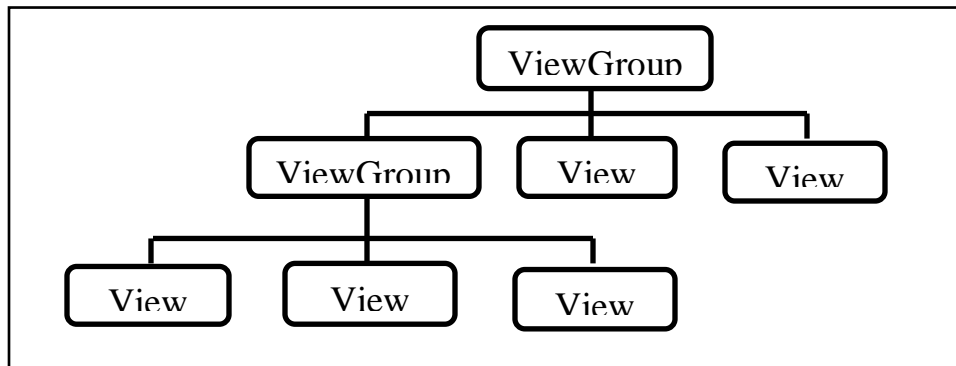
Pada android terdapat package android.view dan android.widget, pada dasarnya keduanya memiliki fungsi sama. Pertama yang akan dibahas pada bab ini adalah mengenai class view untuk pembuatan layout.

2.1 View

UI pada aplikasi Android dibangun dengan menggunakan Object View dan ViewGroup. Ada banyak tipe dari View dan ViewGroup, setiap dari tipe tersebut merupakan turunan dari tipe Views. Object view merupakan unit dasar dari pada platform Android, kelas View menjadi kelas induk (base class) dari subkelas yang bernama “widget”, sebuah object UI yang bisa dipakai dan tersedia secara lengkap. Seperti Teks Field, tombol, dll. Sedangkan kelas ViewGroups tersedia sebagai kelas induk untuk subkelas yang bernama “layout”, yang menyediakan arsitektur layout seperti linier, tabular, relative.

Objek View merupakan struktur data yang menyimpan properties dari parameter layout dan konten secara spesifik untuk area layar (rectangular area). Objek view menangani ukuran, layout, drawing, perubahan fokus, scrolling, dan interaksi key/gesture dari area layar. Sebagai sebuah objek pada UI, View juga merupakan sebagai titik interaksi bagi user dan tempat menerima interaksi even.

Pada platform Android, kita membuat UI Activity dengan menggunakan Hirarki View dan ViewGroup. Pohon hirarki dapat sederhana atau rumit dan kompleks sesuai yang kita butuhkan, dan kita bisa membangun dengan widget dan layout yang telah ada pada android (Predefined), atau dengan View yang kita buat sendiri.



Gambar 2.1 : Contoh Hirarki View

Agar layar bisa menampilkan tampilan (sesuai dengan hirarki view), Activity yang kita buat harus memanggil method `setContentView()` dan mempassing referensi pada objek yang ingin ditampilkan. Sistem Android akan menerima referensi itu dan menggungkannya sebagai nilai – nilai tampilan yang akan ditampilkan.

Unit dasar dari aplikasi android adalah activity, yaitu sebuah objek dari kelas **android.app.Activity**. activity bisa melakukan banyak hal, tetapi dia sendiri tidak akan terlihat secara langsung dilayar. Untuk membuat activity terlihat dilayar adalah dengan membuat sebuah UI, dimana disini kita akan menggunakan views dan viewgroups (Unit dasar UI dalam platform Android).

a. Views

Views adalah objek dari kelas (base class) **android.view.Views**. Merupakan data struktur yang menyimpan layout dan konten untuk area layar. Objek View menghandle ukuran, drawing, perpindahan fokus, scrolling, key/gestur untuk area layar.

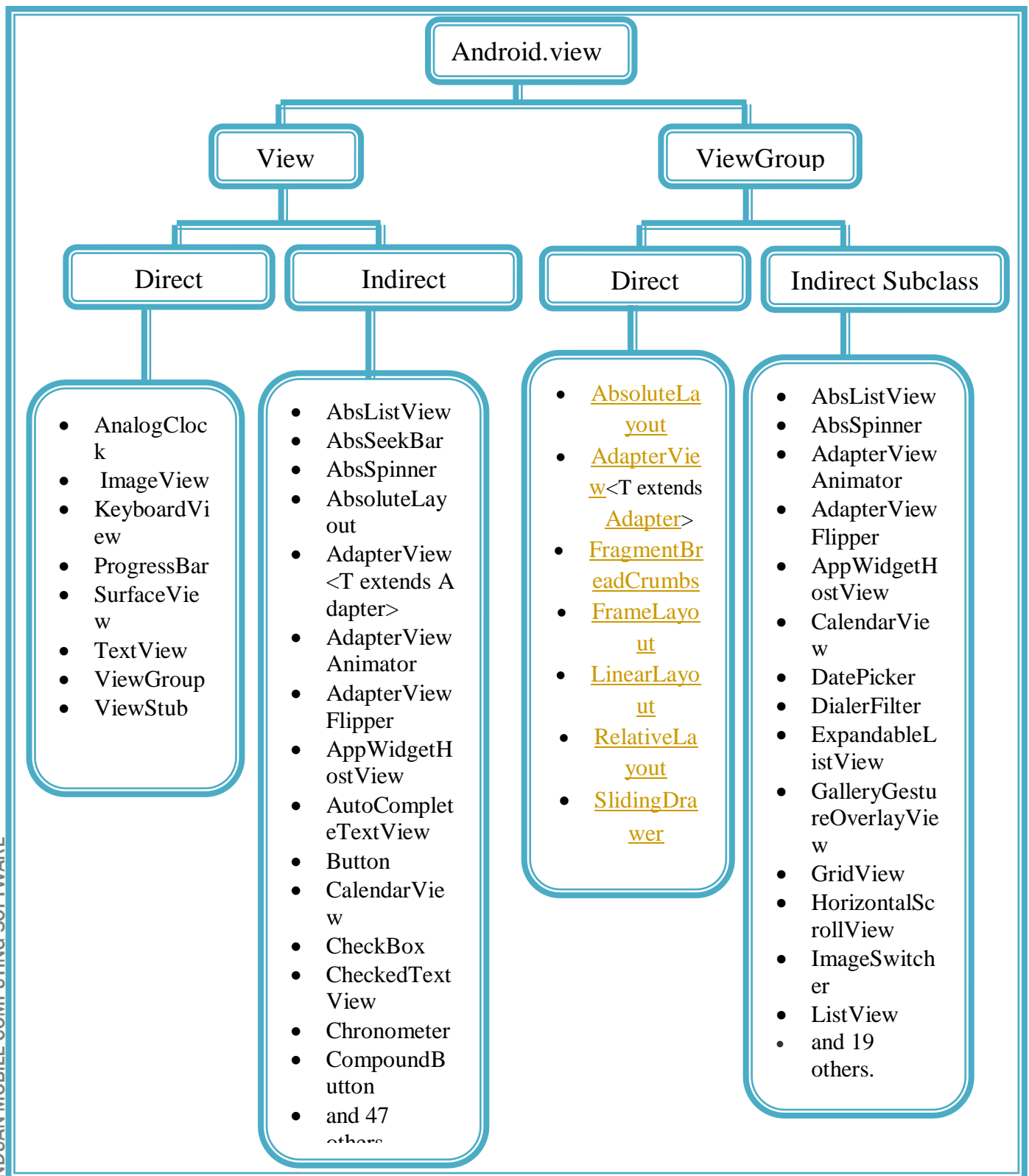
Kelas View tersedia sebagai kelas induk (base class) untuk semua widget - implementasi dari subkelas yang menggambar elemen interaktif pada layar. Widget menangani sendiri bagaimana dia diimplementasikan , sehingga kita bisa memakainya untuk membuat UI secara lebih cepat. List dari widgets yang tersedia meliputi TextView, EditText, Button, RadioButton, CheckBox, scrollView, dan lainnya.

b. ViewGroups

ViewGroups adalah objek dari kelas **android.view.ViewGroup**, dan dari namanya mengindikasikan viewgroup adalah sebuah tipe khusus dari objek view yang berfungsi untuk mewadahi dan mengelola subordinat views dan viewgroups yang lainnya. ViewGroup membolehkan kita untuk menambahkan struktur pada UI dan membuatb elemen layar yang kompleks dan diberikan alamat sebagai satu kesatuan antitas.

Kelas ViewGroups tersedia sebagai kelas induk (baseclass) untuk layout, implementasi dari subkelas yang menyediakan tipe dasar dari layout layar. Dan layotu memberikan cara bagaimana membangun struktur untuk view.

Dibawah ini ditunjukkan diagram class dari package android.view. Pada diagram disusun dari yang paling atas atau tertinggi hingga paling bawah, yaitu masing – masing class di uraikan. Untuk lebih mendalami fungsi dari masing – msing class, kunjungilah website android developer di <http://developer.android.com/reference/android/view/package-summary.html>



Gambar 2.2 : Diagram class dari package *Android.view*

2.2 Layout

Secara mendasar untuk membuat layout dan memperlihatkan hirarki view adalah menggunakan file layout XML. XML memberikan struktur yang mudah dipahami seperti halnya dengan HTML. Setiap elemen XML adalah salah satu objek View atau ViewGroup (atau gabungan keduanya). Objek View seperti sebuah daun pada pohon, dan objek ViewGroup adalah sebuah cabang dari pohon tersebut.

Nama dari elemen – elemen XML adalah merupakan implementasi dari kelas java. Sebuah elemen <TextView> membuat TextView didalam UI. Dan elemen <LinearLayout> adalah membuat viewgroup LinearLayout. Ketika kita meload resource layout, sistem android menginstalasi objek run-time, yang terkait dengan elemen pada layout kita.

Dibawah ini kode xml contoh layout vertikal dengan text view dan tombol :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

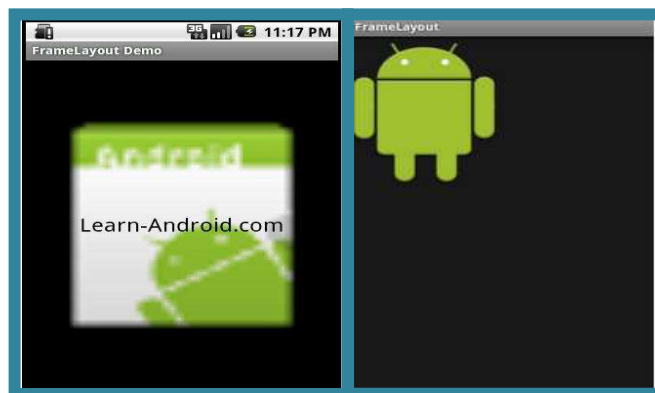
Gambar 2.3 : Kode xml Layout Vertikal dengan Text view

2.3 Layout Manager

Layout Manager	Deskripsi
Linier Layout	Mengatur posisi layout pada layar secara linier (horizontal atau vertical)
Relative Layout	Mengatur tata letak komponen dalam hubungan satu sama lain
Table Layout	Sebuah layout yang diatur dan dikelompokkan kedalam baris dan kolom
Frame Layout	FrameLayout merupakan tampilan dimana widget-widget di dalamnya akan saling menumpuk satu sama lain.
Absolute Layout	Suatu layout yang memungkinkan kita untuk menetapkan lokasi yang tepat dengan menentukan koordinat x dan y

a. FrameLayout

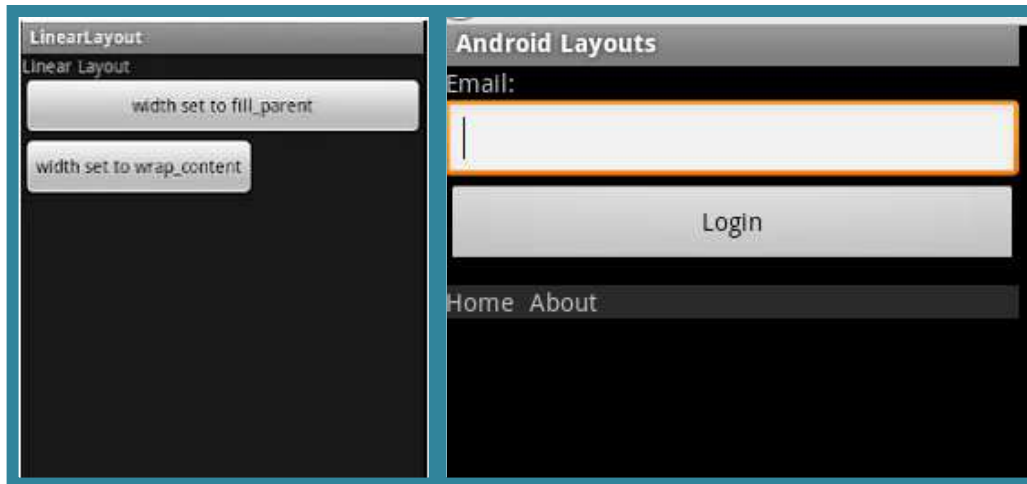
FrameLayout adalah jenis yang paling sederhana dari objek layout. Pada dasarnya merupakan space kosong pada layar yang kemudian dapat diisi dengan objek tunggal misalnya gambar. Semua elemen child view dari FrameLayout akan di letakkan di pojok kiri atas layar



Gambar3.4 :FrameLayout

b. **LinearLayout**

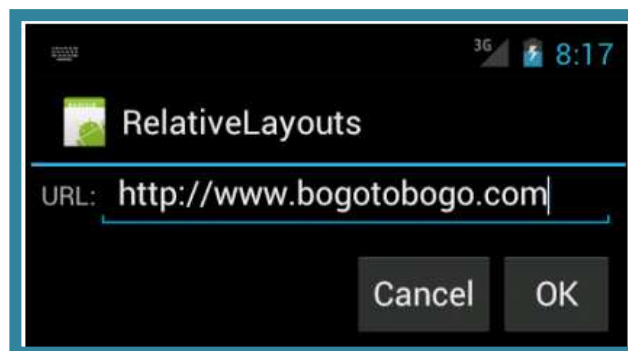
Merupakan salah satu class dari package android.widget yang berfungsi untuk menghasilkan tampilan dimana setiap komponen diletakkan berurutan secara horizontal atau vertikal.



Gambar 3.5 : LinearLayout

c. **RelativeLayout**

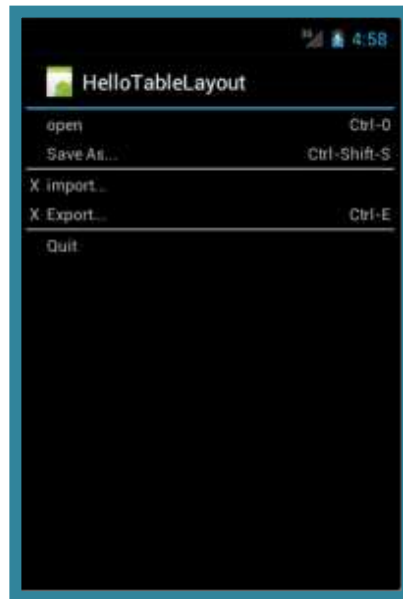
RelativeLayout adalah sebuah layout dimana posisi dari sebuah komponen (simbol, text, dsb) letaknya bisa diatur terhadap komponenlainnya



Gambar 3.5 : RelativeLayout

d. **TableLayout**

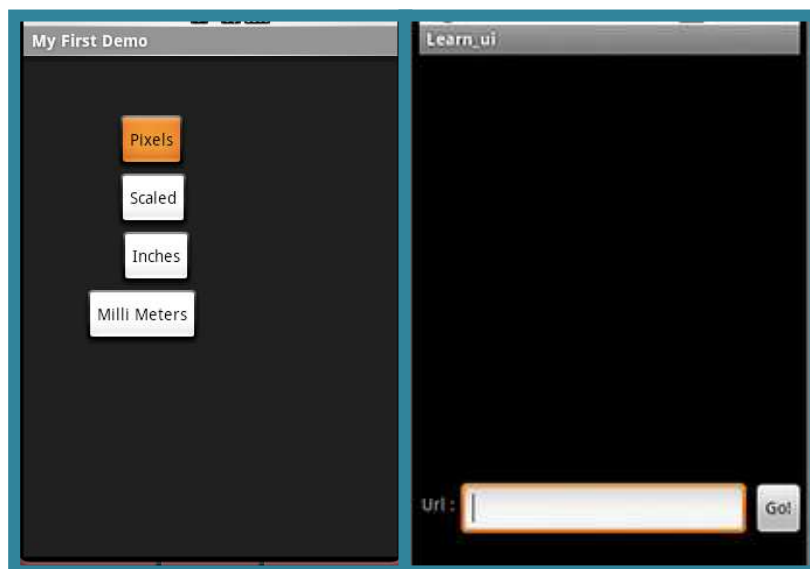
TableLayout adalah sebuah tampilan yang disusun berdasarkanbaris dan kolom. Table layout terdiri atas sejumlahtepinya tidak terlihat.



Gambar 4.6 : TableLayout

e. Absolute Layout

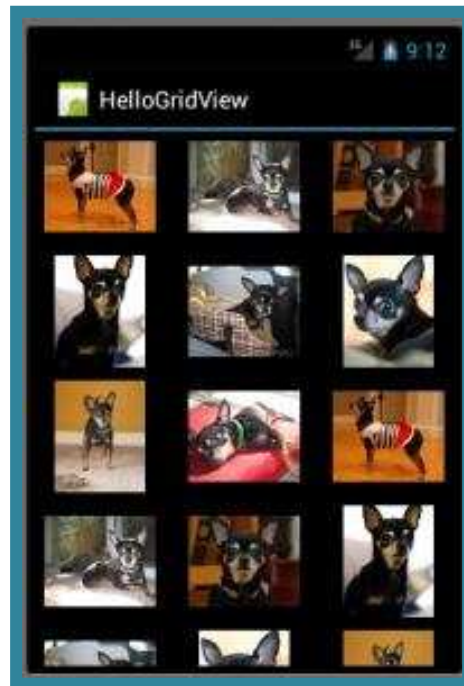
AbsoluteLayout adalah tampilan dimana kita dapat memasukkan posisi x dan y dari widget yang ada di dalamnya secara pasti sehingga penempatan posisi dapat ditentukan secara pasti.



Gambar 4.7 :Absolute Layout

f. GridLayout

GridLayout merupakan salah satu class dari android.widget. Yang dapat menampilkan data dalam bentuk grid. Data yang ditampilkan dapat berupa teks, gambar, dan lain-lain.



Gambar 3.8 : GridLayout

2.4 UI Events

Sekali kita menambahkan beberapa Views atau Widget kedalam UI, kita akan berfikir bagaimana mekanisme interaksi pengguna dengan mereka. Untuk itu kita harus melakukan satu dari dua hal ini :

1. Menetapkan sebuah event listener dan mendaftarkannya dengan View. Ini biasanya akan lebih sering kita lakukan dari pada tidak, karena ini adalah cara bagaimana kita memonitor adanya sebuah event. Kelas View memiliki koleksi sekumpulan interface dan dinamakan On<sesuatu>Listener, dari masing – masing memiliki method untuk mengembalikan nilai (callback) On<sesuatu>().

Sebagai contoh:

- `View.OnClickListener` (*untuk menghandle klik pada View*)
- `View.OnTouchListener` (*untuk menghandle even sentuh di layar pada View*)
- `View.OnKeyListener` (*untuk menghandle keypad pada View*)

Maka jika kita ingin View diberi pesan ketika diklik (seperti ketika sebuah tombol dipilih), implementasikan **OnClickListener** dan definisikan sebuah method yang mengembalikan nilai **OnClick()** (dimana kita melakukan klik), dan mendaftarnya pada View dengan `setOnClickListener()`.

2. Override method Pengembali nilai (Callback) yang ada untuk View. Ini yang harus dilakukan ketika kita mengimplementasikan kelas View kita sendiri dan memonitor spesifikasi even yang terdapat padanya.

Contoh :

- even yang terjadi ketika layar disentuh (`OnTouchEvent()`).
- trackball digerakkan (`OnTrackballEvent()`).
- keypad ditekan (`OnKeyDown()`).

Untuk itu kita dibolehkan untuk mendefinisikan default perilaku (Behavior) setiap even dalam View dan menentukan apakah even akan di pasing kedalam Child View yang lain. Sekali lagi ini kembali ke kelas View, kita hanya diberikan untuk mendefinisikannya hanya ketika kita akan membuat komponen sendiri (Custom Component).

2.5 Mendeklarasikan User Interface (UI)

Android UI bisa dibuat dengan 2 cara, yaitu dengan menggunakan kode XML atau menulisnya melalui kode java. Membuatnya dengan XML sangat direkomendasikan, karena seperti yang kita tahu itu menggunakan konsep MVC (Model-Viewer-Control) yang memisahkan antara UI dengan kode logika program (Logic Program). Dengan begitu, adaptasi kode program aplikasi dari satu format resolusi layar ke format resolusi yang lain menjadi mudah.

Membuat UI di XML sangat mirip dengan membuat dokumen HTML. Sama dengan layout XML Android, semua dalam sebuah struktur rapi dan ditulis pada struktur pohon (tree-struktur).

```
<?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:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    />
</LinearLayout>
```

2.6 Widget

Widget adalah objek View yang bertugas sebagai sebuah antarmuka (interface) untuk interaksi dengan pengguna. Android menyediakan pengimplementasian widget secara penuh, tombol, checkbox, dan field teks, oleh karena itu kita bisa secara cepat membangun UI pada aplikasi android. Selain itu tersedia juga widget yang lebih kompleks seperti date picker, jam, atau tombol zoom yang juga disediakan oleh android. Tetapi kita tidak dibatasi oleh android, disini kita juga dapat membuat sendiri widget dengan membuat sendiri sesuatu yang customized atau elemen – elemen actionable jika dibutuhkan. Dengan mendefinisikan sendiri objek View atau meng-extend dan mengkombinasikan widget yang ada. Beberapa class yang ada di bawah package android.widget adalah [AbsoluteLayout](#), [AnalogClock](#), [Button](#), [CheckBox](#), [EditText](#), [FrameLayout](#), [GridView](#), [ImageButton](#), [ImageView](#), [LinearLayout](#), [ListView](#), [RadioButton](#), [RelativeLayout](#), [Spinner](#), [TableLayout](#), [TextView](#), [ZoomButton](#), dan masih banyak lagi.

Paket widget pada dasarnya merupakan visualisasi dari elemen User Interface (UI) Yang digunakan pada layar aplikasi android dimana kita dapat merancang sendiri sesuai kebutuhan kita.

Widget di dalam android di tampilkan dengan konsep View. Dimana aplikasi android pada umumnya menggunakan widget sebagai layout XML. Untuk mengimplementasikan widget, selain file java kita juga membutuhkan tambahan dua file. Berikut ini adalah file – file yang umumnya kita butuhkan apabila kita membuat widget.

- File Java

Berupa file java dimana merupakan file yang mengimplementasikan aksi dari widget. Jika kita mendefinisikan suatu widget beserta posisinya di layar yang didefinisikan dari file XML, kita harus melakukan koding di file java dimana kita dapat mengambil semua nilai atribut dari tata letak file xml yang didefinisikan.

- File XML

Sebuah file XML yang berada di res / value / yang mendefinisikan komponen elemen – elemen XML yang digunakan untuk inisialisasi widget serta atribut yang mendukungnya (file yang kita kenal dengan main.xml).

- Layout XML [optional]

File ini tidak harus ada, file xml berada di dalam res / layout / yang menggambarkan atau menambahkan keterangan [pada layout widget. Sebenarnya ini kita dapat lakukan di file java (file ini dikenal dengan file String.java) dimana fungsi file ini dapat di handle oleh file java.

Berikut beberapa contoh class dari widget dapat dilihat di website android developer yaitu di <http://developer.android.com/reference/android/widget/package-summary.html>

2.6.1 Resources dan R.java

Resource dari projek dan R.java sangat berkaitan

a. Resources

Resource adalah file eksternal(non-code file) yang digunakan oleh kode kita dan di compile ke dalam aplikasi pada saat build. Android mendukung banyak macam dari file resource, termasuk file XML, PNG, dan JPEG. File XML mempunyai perbedaan format tergantung bagaimana mereka di deskripsikan.

Resource ditempatkan diluar kode, dan file XMP dikompile kedalam binary, format yang bisa cepat di loading untuk alasan efisiensi. String dikompres kedalam format penyimpanan yang lebih efisien.

Daftar Resource

Tipe Resources dan tempat dimana mereka ditempatkan :

- Layout – files : “/res/layout/”
- Images : “/res/drawable/”
- Animations : “/res/anim/”
- Styles, string dan array : “/res/values/”
 - Nama jangan persis dengan seperti :
 - ‘array.xml’ untuk mendefinisikan array
 - ‘color.xml’ untuk mendefinisikan warna
- #RGB, #ARGB, #RRGGBB, #AARRGGBB
 - ‘dimens.xml’ untuk mendefinisikan dimensi
 - ‘string.xml’ untuk mendefinisikan string
 - ‘styles.xml’ untuk mendefinisikan objek
- Raw file seperti mp3 atau video : “/res/raw/”

Menggunakan Resources didalam kode

Menggunakan resource didalam kode adalah dengan mengetahui ID resource dan tipe dari objek dari resource kita. Dibawah ini syntax untuk mengacu (referring) kepada resource :

```
R.resource_type.resource_name atau  
Android.R.resource_type.resource_name
```

Dimana `resource_type` adalah subkelas (subclass) dari `R` yang merupakan tipe spesifik dari `resource.resource_name` merupakan nama atribut dari `resource` yang didefinisikan pada file XML, atau file name (tanpa ekstensi) untuk `resource` yang didefinisikan oleh tipe file lain. Setiap tipe dari `resource` akan ditambahkan pada subkelas `R` yang spesifik, tergantung dari tipe `resource`.

Resource yang di compile sendiri oleh aplikasi dapat direfer(diacu) tanpa nama paket (name package) (seperti `R.resource_type_resource_name`). android memiliki beberapa standart resource, seperti style layar, dan background button. Untuk merefer pada kode itu, kita harus memastikan kualitasnya dengan Android, seperti contoh: **`android.R.drawable.button_background`**.

Resource Refence

Nilai yang dimasukkan pada atribut (atau resource) dapat juga di refer pada resource lain. Sering kali ini dilakukan pada file layout untuk memasukkan string (sehingga mereka bisa dilocalized) dan image (yang berada pada file lain), walaupun juga bisa dilakukan juga pada tipe resource lain misalnya warna (colors) dan nilai integer.

Contoh untuk resource warna :

```
<EditText  
    Android:layout_width="fill_parent"  
    Android:layout_height="fill_parent"  
    Android:textColor="@color/opeque_red"  
    Android:text="Hello, world!" />
```

Seperti yang terlihat diatas menggunakan prefik “@” untuk memperkenalkan resource reference, text yang menyertai kemudian adalah nama dari resource didalam format @[package:]type/name. Pada kasus ini kita tidak menspesifikasikan paket karena kita

merefensi resource didalam paket kita sendiri. Untuk mereference sistem resource, kita bisa menulisnya seperti ini :

```
<EditText
    Android:layout_width="fill_parent"
    Android:layout_height="fill_parent"
    Android:textColor="@android:color/opaque_red"
    Android:text="Hello, world!" />
```

Contoh 1 t :

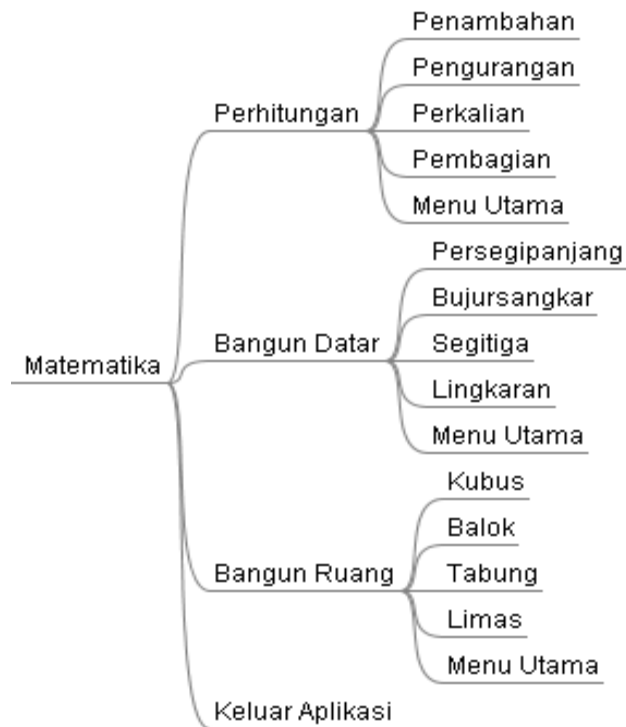
```
<EditText
    Android:layout_width="fill_parent"
    Android:layout_height="fill_parent"
    Android:textColor="@android:color/opaque_red"
    Android:text="@string/Hello, world!" />
```

b. R.java

R.java merupakan file yang digenerate secara otomatis, yang mengindex semua resource dari proyek. Dapat menggunakan kelas ini kedalam kode sebagai cara pintas untuk merefer ke resource yang dimasukkan kedalam proyek. Ini sangat bermanfaat apabila dikaitkan dengan fitur IDE code-completion seperti Eclipse karena ini membuat kita secara cepat dan interaktif menemukan spesifik resource. Selain itu, dapat meningkatkan waktu compile (compile-time) karena resource yang akan dipakai benar – benar ada.

2.6.2 List Bertingkat

List Bertingkat adalah kondisi dimana apabila salah satu menu diklik atau disentuh maka akan muncul list lainnya.



Gambar 2.4 :Gambar Contoh List Bertingkat.

2.6.3 Tab Menu



Gambar 2.5 :Gambar Contoh Tab Menu.

Dalam dunia aplikasi Tab Menu ini termasuk kedalam kelompok horizontal menu. Tab Menu cocoknya digunakan untuk aplikasi yang hanya menggunakan tidak lebih dari 5 Menu, Jika sudah lebih dari 5 Menu Tab Menu bisa dikombinasikan dengan List Menu.

Tab Menu punya kelebihan dan kekurangan :

- Kelebihannya yaitu Menu mudah terlihat sehingga memudahkan user untuk berpindah ke menu lain.
- Kerugiannya yaitu Menu yang ditampilkan membutuhkan ukuran dan ruang khusus

3.1 AUDIO DALAM ANDROID

Audio atau kita kenal dengan istilah **suara** adalah kompresi mekanikal atau gelombang longitudinal yang merambat melalui medium. Medium atau zat perantara ini dapat berupa zat cair, padat, gas. Jadi, gelombang bunyi dapat merambat misalnya di dalam air, batu bara, atau udara. Kebanyakan suara adalah merupakan gabungan berbagai sinyal, tetapi suara murni secara teoritis dapat dijelaskan dengan kecepatan osilasi atau frekuensi yang diukur dalam Hertz (Hz) dan amplitudo atau kenyaringan bunyi dengan pengukuran dalam desibel. Manusia mendengar bunyi saat gelombang bunyi, yaitu getaran di udara atau medium lain, sampai ke gendang telinga manusia. Batas frekuensi bunyi yang dapat didengar oleh telinga manusia kira-kira dari 20 Hz sampai 20 kHz pada amplitudo umum dengan berbagai variasi dalam kurva responsnya. Suara di atas 20 kHz disebut ultrasonik dan di bawah 20 Hz disebut infrasonik.

Socrates pernah berujar bahwa suara adalah imitasi terbaik bagi pikiran maka suara adalah media terbaik untuk menyampaikan informasi. Bagi Socrates text adalah imitasi dari suara, dengan demikian sebagai penyampai pikiran text bukanlah media yang ideal karena ia hanyalah imitasi dari suatu imitasi. Pendapat Socrates mungkin ada benarnya karena suara adalah media yang secara natural telah dimiliki oleh manusia sehingga suara adalah media yang paling alami. Guru di kelas pun lebih banyak mengandalkan suara baik ketika memberikan materi atau melakukan motivasi bagi siswa-siswanya. Jika untuk percakapan secara langsung audio adalah media yang simpel dan alami maka tidak demikian halnya ketika digunakan di dalam komputer. Penggunaan suara di dalam komputer berlangsung belakangan sesudah penggunaan text.

Kelebihan suara di dalam multimedia pembelajaran :

- sangat cocok bila digunakan sebagai media untuk memberikan motivasi.
- Untuk materi- materi tertentu suara sangat cocok karena mendekati keadaan asli dari materi (misal pelajaran mengenai mengenal suara-suara binatang)

- Membantu pembelajar fokus pada materi yang dipelajari karena pembelajar cukup mendengarkan tanpa melakukan aktivitas lain yang menuntut konsentrasi . Bandingkan dengan pembelajar yang melihat teks di layar komputer. Dalam hal ini pembelajar melakukan multi aktivitas yakni : membaca teks pada layar (yang tidak semudah membaca pada buku), mencari kata-kata kunci (keyword) dari materi, dan menggerakkan tangan, seperti melakukan klik mouse untuk menggulung layar saat ingin melihat bagian teks yang tak terlihat pada layar.

Kelemahan audio :

- memerlukan tempat penyimpanan yang besar di dalam komputer.
- Memerlukan software dan hardware yang spesifik (dan mungkin mahal) agar suara dapat disampaikan melalui komputer.

Beberapa method dari MediaPlayer yang kita gunakan dalam aplikasi ini:

Method	Keterangan
Create	Untuk mendaftarkan file suara yang ingin dimainkan
Start	Untuk memulai memainkan file suara
Stop	Untuk menghentikan file suara
setLooping	Menentukan apakah berulang atau tidak
pause	Untuk mempause file suara yang sedang dimainkan
isPlaying	Untuk mengetahui media player sedang memainkan file suara atau tidak

Gambar Table 3.1 Metode-metode yang Ada pada MediaPlayer

3.2 Animasi Pada Android

Android menawarkan beberapa alat animasi berbeda. Animasi yang didasarkan pada kelas `Android.view.animation.Animation`: `RotateAnimation`, `TranslateAnimation`, `ScaleAnimation`, dan `RotateAnimation`. Animasi ini digunakan antar pasangan View. Tipe kedua merupakan subkelas dari `android.graphics.drawable.AnimationDrawable` bisa dimasukkan ke dalam latar belakang widget apa pun untuk menyediakan berbagai macam efek.

1. Animasi Tween

Sebuah animasi tween dapat melakukan serangkaian transformasi sederhana (posisi, ukuran, rotasi, dan transparansi) pada isi dari objek view. Jadi, jika dimiliki objek

textView, objek tersebut dapat dipindahkan, diputar, maupun mengecilkan teks. Jika memiliki gambar latar belakang, gambar latar belakang akan berubah bersama dengan teks.

File animasi XML disimpan di res/anim file harus memiliki elemen root : <alpha>, <scale>, <rotate>, <translate>, atau <set> yang dapat menggabungkan beberapa elemen animasi. Secara default, semua instruksi animasi diterapkan secara bersamaan. Untuk membuat mereka terjadi secara berurutan, harus ditentukan atribut startoffset.

2. Animasi Frame

Ini adalah animasi tradisional dalam arti bahwa hal itu dibuat dengan urutan gambar yang berbeda dimainkan dalam urutan, seperti rol film. Kelas animation drawable adalah dasar untuk animasi frame. Lebih sederhana dilakukan dengan file XML yang berisi daftar frame yang membentuk animasi. Seperti animasi tween di atas, file XML untuk jenis animasi termasuk ke dalam res/drawable. Dalam hal ini, petunjuk urutan dan durasi untuk setiap frame animasi.

Game merupakan permainan elektronik yang menggunakan media computer, phone seluler maupun konsol seperti playstation atau x-box. Bisnis game juga sudah merambah kemana-mana, namun ironisnya content dari game sebagian besar berisi hiburan dan sangat sedikit yang bercontent pendidikan (edukasi). Sebenarnya tanpa disadari game dapat mengajarkan banyak ketrampilan dan game dapat dijadikan sebagai salah satu alternative pendidikan [Buckingham dan Scalon 2002]. Menurut Foreman game merupakan potential learning environments. Bermain game merupakan sebuah literacy baru dalam pendidikan.

Game yang ada sekarang memiliki berbagai macam jenis atau yang dikenal dengan istilah genre. Game bergenre educational ini bertujuan untuk memancing minat belajar anak terhadap materi pelajaran sambil bermain. Sehingga dengan perasaan senang diharapkan anak bisa lebih mudah memahami materi pelajaran yang disajikan. Genre ini sebenarnya lebih mengacu kepada isi dan tujuan game, bukan genre yang sesungguhnya.

4.1 Definisi Game

Menurut Romi Satria Wahono game merupakan aktifitas terstruktur atau semi terstruktur yang biasanya bertujuan untuk hiburan dan kadang dapat digunakan sebagai sarana pendidikan. Karakteristik game yang menyenangkan, memotivasi, membuat kecanduan dan kolaboratif membuat aktifitas ini digemari oleh banyak orang.

4.2 Game Platforms

Game secara umum banyak dibuat di beberapa platform sebagai berikut:

➤ Arcade

Arcade merupakan game dengan sistem coin operated, sampai saat ini platform jenis ini masih sering digunakan. Permulaan platform ini dipelopori oleh Atari. Yang termasuk dalam platform ini adalah:

1. Sega
2. Namco
3. Video Game

➤ **Personal Computer**

Game yang berjalan pada platform ini sangat mendominasi pasar, ada beberapa Operating System yang biasa dipakai untuk membuat game pada platform ini:

1. Windows

Memanfaatkan kemampuan Windows untuk tampilan yang sangat baik dengan kemampuan DirectX yang disediakan Microsoft menjadikan para pengembang game umumnya bermain di platform ini. Tools dan software untuk membuat game sangat banyak tersedia di platform Windows. Baik dengan atau tanpa programming ada di platform ini, dan karena PC adalah peralatan yang banyak dimiliki oleh pengguna rumahan maka kebanyakan pembuat game mengembangkan game pada platform ini.

2. Linux

Linux walaupun terkenal stabil, masih sedikit keraguan untuk mengembangkan game pada platform ini, karenanya masih sedikit game yang dibuat di Linux. Tapi kemudahan lain adalah tersedianya aplikasi yang gratis sehingga walaupun sedikit komunitasnya bisa membuat game yang tidak kalah dari platform Windows. Mungkin hanya kalah bersaing dari jumlah pengembang dan waktu saja, salah satu game engine yang bisa berjalan pada platform ini adalah Torque Engine 3D dengan pemrograman C++.

3. Dos

Masa keemasan platform ini sudah selesai. Dulu hampir semua game komersil berjalan di platform ini. Namun kini sudah tidak dikembangkan lagi karena perpindahan Operating System yang beralih ke Windows. Namun masih banyak resources di internet yang masih menyediakan pembahasan dan programming dengan DOS.

4. Mac

Mac adalah sistem operasi keluaran Apple yang stabil dan fiturnya sangat baik. Sebenarnya Windows menirunya. Tapi nasibnya tidak terlalu beruntung. Kelebihan dari platform ini

adalah fitur grafik dan suaranya yang sangat baik. Jika kita mencari di internet ada beberapa tools dan software yang bisa kita gunakan untuk membuat game di platform ini, tapi terbatas baik jumlah maupun jenis game yang bisa dibuat.

➤ **Hand-held dan Mobile Device (Cell phone)**

Platform ini dipilih karena kepraktisannya. Walaupun tidak sehebat game di komputer, tetapi kemudahan dan jumlah peralatan yang nyaris digunakan untuk alat komunikasi akan membuat platform ini menjadi pilihan pengembang. Umumnya menggunakan java akan tetapi sekarang mulai banyak game mobile yang dikembangkan dengan menggunakan flashlite pada platform ini.

Yang termasuk dalam platform hand held ini adalah:

1. Game?n?Watch
2. Gameboy
3. Gameboy Advance
4. N-Gage
5. Gameboy DS
6. PSP

➤ **Console**

Mesin Console seperti Playstation dan Xbox menjadi saingan bagi platform Windows. Pada awalnya mesin console duluan menjual hiburan ini kepada pemakainya, komputer muncul belakangan. Namun kekuatan pasar awal mesin console dan kemudahan peralatannya menjadikan tetap menjadi pesaing platform PC walaupun sekarang platform PC (Windows) sudah menjadi nomor satu. Pemrograman di mesin console hampir sama dengan platform PC namun kita kurang mendapatkan kemudahan karena umumnya sulit mengembangkan game di platform ini secara individu karena tools dan lisensi ada pada studio pengembang dan harganya sangat mahal demi menjaga mutu dan hak cipta mereka.

4.3 Game Genres

Jenis-jenis game lebih dikenal dengan istilah genre. Samuel Henry mendefinisikan genre sebagai format atau gaya dari sebuah game. Format sebuah game bisa murni sebuah genre atau campuran (hybrid) dari beberapa genre lain dengan maksud membuat unsur permainan lebih bervariasi dan menantang berbagai jenis game beredar dipasaran, beberapa diantaranya adalah:

➤ **Maze Game**

Jenis game ini adalah jenis game yang paling awal muncul. Contoh yang paling populer di Indonesia adalah game pacman dan digger. Pada maze game ini pemain hanya mengitari maze (lorong-lorong yang berhubungan) dan memakan beberapa item untuk menambah tenaga dan kekebalan misalnya. Pemain juga memiliki musuh yang selalu mengejar. Ketika pemain mendapat kekebalan, pemain bisa berbalik mengejar musuh. Permainan ini sederhana tetapi mengasyikkan. Mode permainan ini yang menjadi dasar bagi permainan 3D sekarang. Jika dulu bentuknya 2D maka sekarang menjadi bentuk 3D.

➤ **Board Game**

Jenis game ini sama dengan game board tradisional, seperti monopoly (selain platform PC juga ada pada platform konsol). Sampai saat ini tidak ada variasi yang memunculkan gameplay atau perubahan desain dari versi tradisional ke versi elektronik. Versi elektronik benar-benar hanya memindahkan versi tradisional ke layar komputer. Variasi yang ada hanyalah memindahkan versi 2D menjadi versi 3D (misalnya seperti dalam game catur). Terkadang disisipkan variasi film intro atau animasi lainnya. Umumnya game ini lebih menekankan kepada kemampuan komputer untuk menjadi lawan tanding bagi pemain. Ini melibatkan kemampuan AI (Artificial Intelligence) yang andal untuk bisa menjadikan game ini menantang pemain dengan baik.

➤ **Puzzle Game**

Game ini memberikan tantangan kepada pemainnya dengan cara menjatuhkan sesuatu dari sisi sebelah atas ke bawah. Pemain harus menyusun sedemikian rupa dan tidak ada yang tersisa ketika susunan di atasnya sudah akan dibuat. Susunan ini dilakukan secepat dan sebaik mungkin. Semakin lama akan semakin cepat dan semakin banyak obyek yang jatuh. Contoh yang populer dari jenis ini adalah tetris.

➤ **Fighting Game**

Sesuai dengan namanya game ini mengetengahkan pertarungan. Pada awalnya bersifat 2D namun pada akhirnya banyak mengadopsi sistem 3D disertai animasi. Game ini memberikan kesempatan kepada pemain untuk mengkombinasikan berbagai gerakan dalam pertarungan. Ada yang mengadopsi permainan bela diri, atau ada juga gerakan yang liar. Terkadang musuh bukan manusia tetapi makhluk yang tidak masuk akal sama sekali. Contoh yang populer di Indonesia adalah Street Fighter 2, Samuraidown, Virtua Fighter, Kungfu dan sebagainya.

➤ **Racing Game**

Game balapan, game ini memberikan permainan lomba kecepatan dari kendaraan yang dimainkan oleh pemain. Bisa di dalam arena balap atau diluar arena balap. Beberapa contoh game yang terkenal seperti Need For Speed Underground dan Toca Race Driver.

➤ **Action Game**

Memasuki gua bawah tanah, melompati bebatuan di antara lahar, bergelayutan dari pohon satu ke pohon lain, bergulat dengan ular sambil mencari kunci untuk membuka pintu kuil legendaris atau sekedar mencari telepon umum untuk mendapatkan misi berikutnya, itulah beberapa dari banyak hal yang karakter pemain harus lakukan dan lalui dalam video game jenis ini. Menurut penulis, game jenis ini sudah berkembang jauh hingga menjadi genre campuran action beat-em up juga, dan sekarang, di tahun 2000-an, jenis ini cenderung untuk memiliki visual 3d dan sudut pandang orang ketiga. Tomb rider, grand theft auto dan prince of persia termasuk didalamnya.

➤ **Strategi Game**

Kebalikan dari game jenis action yang berjalan cepat dan perlu refleks secepat kilat, video game jenis strategi, layaknya bermain catur, justru lebih memerlukan keahlian berpikir dan memutuskan setiap gerakan secara hati-hati dan terencana. Game jenis strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan, sampai kendaraan, bahkan hingga pembangunan berbagai bangunan, pabrik dan pusat pelatihan tempur, tergantung dari tema ceritanya.

Pemain game strategi melihat dari sudut pandang lebih meluas dan lebih ke depan dengan waktu permainan yang biasanya lebih lama dan santai dibandingkan game action. Unsur-unsur permainannya biasanya berkisar sekitar prioritas pembangunan, peletakan pasukan, mencari dan memanfaatkan sumber daya (uang, besi, kayu, minyak, dll), hingga ke pembelian dan peng-upgrade-an pasukan atau teknologi.

➤ **Game Simulasi kendaraan**

game jenis ini memberikan pengalaman atau interaktifitas sedekat mungkin dengan kendaraan yang aslinya, meskipun terkadang kendaraan tersebut masih eksperimen atau bahkan fiktif, tapi ada penekanan khusus pada detil dan pengalaman realistik menggunakan kendaraan tersebut.

4.4 Konsep pengembangan game

Pengembangan game biasanya dilakukan oleh seorang developer, bisa 1 orang atau 1 perusahaan besar. Biasanya, game komersial berskala besar dibuat oleh tim pengembang dalam sebuah perusahaan yang mengkhususkan pada game komputer atau konsol. Video game modern umumnya menghabiskan biaya mulai dari USD \$1.000.000 sampai lebih dari \$20.000.000 untuk dikembangkan. Pengembangan biasanya didanai oleh sebuah penerbit. Sebuah game modern bisa memakan waktu 1-3 tahun untuk dikembangkan, meskipun ada pengecualian lainnya.

Proses pengembangan game bervariasi tergantung pada perusahaan dan proyek. Namun perkembangan game komersial biasanya meliputi tahapan sebagai berikut.

1. Pre-Production

Fase awal perkembangan game sering ditandai oleh rendahnya kualitas grafis. Hal ini benar terutama dari berbagai permainan prototipe. Produsen game bekerja selama pra-produksi yang umumnya terkait dengan perencanaan jadwal, anggaran & perkiraan tugas dalam tim. Melakukan produser ini bertujuan untuk membuat rencana produksi yang padat sehingga produksi dapat dimulai bila diperlukan tanpa penundaan.

2. Production

Produksi mainstream biasanya didefinisikan sebagai periode waktu ketika proyek sepenuhnya dikerjakan. Programmer menulis banyak kode baru, artis game mengembangkan aset permainan seperti sprite, unsur permainan model 3D. Sound insinyur mengembangkan efek suara dan

komposer musik untuk mengembangkan game. Desainer menciptakan tingkat lanjutan dan kenyamanan mata saat memainkan game tersebut dan penulis menulis dialog untuk cutscenes dan NPC.

Sementara itu, desainer game menerapkan dan memodifikasi desain game untuk mencerminkan visi sebuah game. Fitur dan level seringkali dihapus atau ditambahkan. Gaya seni yang ada bisa dikembangkan dan backstory mungkin dirubah. Platform baru mungkin saja menargetkan game dengan demografis baru. Semua perubahan ini harus didokumentasikan dan tersebar ke seluruh tim. Sebagian besar perubahan terjadi sebagai pembaruan dokumen desain.

Dari sudut pandang waktu, level pertama dari game bisa memakan waktu yang lama untuk dikembangkan. Sebagai desainer dan artis menggunakan alat-alat untuk membangun level, mereka meminta fitur dan perubahan yang lebih cepat dan memungkinkan untuk pengembangan dengan kualitas yang lebih tinggi. Memperkenalkan fitur baru dapat menyebabkan level lama menjadi usang, sehingga level awal bisa dikembangkan berkali-kali. Karena lingkungan dinamis pada pengembangan game, desain level awal dapat juga berubah seiring berjalannya waktu. Hal ini tidak jarang menghabiskan lebih dari dua belas bulan pada satu level game yang dikembangkan selama 3 tahun. Level selanjutnya dapat dikembangkan jauh lebih cepat sebagai set fitur yang lebih lengkap dan visi permainan yang lebih jelas dan lebih stabil.

3. Testing (pengujian)

Staf pengujian paling banyak diandalkan pada akhir proyek karena mereka tidak hanya perlu untuk menguji fitur baru yang ditambahkan, level dan perbaikan bug, tapi mereka juga perlu melakukan pengujian regresi untuk memastikan bahwa fitur-fitur yang telah ada ditempatnya selama berbulan-bulan bisa beroperasi dengan benar. Hal ini juga sering ketika fitur dan level sedang selesai pada level tertinggi, sehingga ada lebih banyak materi baru yang akan diuji daripada waktu lainnya dalam proyek. Pengujian regresi merupakan salah satu tugas paling penting yang dibutuhkan untuk pengembangan software yang efektif.

4. Penyelesaian

Setelah game dibuat, beberapa developer akan memberikan waktu kompensasi untuk anggota tim (mungkin sampai satu atau dua minggu) untuk mengkompensasi lembur dalam menyelesaikan permainan, meskipun kompensasi ini bukanlah suatu standar.

5. Pemeliharaan

Konsol game dulu dianggap 100% lengkap sehingga tidak mungkin adanya masalah atau bug. Namun, dengan pengenalan konsol yang bisa online seperti Xbox 360, PlayStation 3 dan Wii sebagian besar game menerima patch dan perbaikan setelah karena adanya bug dan masalah-masalah kecil, seperti game komputer.

Sementara konsol game dapat dikembangkan untuk kumpulan komponen yang berhingga, game komputer dapat memiliki konflik dengan berbagai konfigurasi hardware pengguna. Developer mencoba untuk menjelaskan konfigurasi yang paling umum, tetapi tidak dapat mengantisipasi semua sistem permainan yang mungkin mereka coba. Hal ini merupakan penerapan umum untuk pengembang game komputer untuk merilis patch. Patch ini digunakan untuk dikirimkan kepada pengguna melalui floppy disk, tetapi sekarang umumnya tersedia untuk didownload melalui situs web pengembang.

6. Durasi

Game modern bisa memakan waktu 1-3 tahun untuk menyelesaikannya. Lamanya suatu pengembangan bergantung pada sejumlah faktor, seperti genre, skala, platform pengembangan dan jumlah aset.

Sebagai contoh, sebuah game puzzle yang sederhana dengan menggunakan grafik 2D akan jauh lebih sedikit memakan waktu untuk dikembangkan daripada game role-playing komputer 3D.

Pertimbangan lainnya adalah penggunaan mesin game middleware. Mengembangkan mesin 3D dari bawah ke atas membutuhkan waktu lebih banyak daripada menggunakan paket middleware yang ada Cots (commercial, off-the-shelf) seperti Gamebryo atau RenderWare. Sebagai contoh, Gas Powered Games mengembangkan sebuah mesin 3D kustom untuk permainan mereka Dungeon Siege. Pengembangannya memakan waktu 3 tahun. Firaxis menggunakan mesin game Gamebryo untuk permainan mereka Sid Meier's Pirates! yang dikembangkan selama 2 tahun kurang.

4.5 Game Tebak Kata

Game Tebak Kata yang bertujuan untuk memberikan hiburan dan mengasah otak. Game ini terdiri dari beberapa pertanyaan dimana pada setiap pertanyaan terdapat clue untuk membantu user menjawab pertanyaan tersebut. Game ini memiliki beberapa form, form utama yang

ditampilkan adalah ‘Menu utama’ sebagai halaman awal, yang akan membagi pilihan yang akan diakses. Pada ‘Menu Utama’ akan kita bagi lagi menjadi beberapa pilihan submenu yang dapat dipilih. Setelah memasuki ‘Menu Utama’, pemakai akan dihadapkan pada 4 pilihan, yaitu menu pilihan masuk, petunjuk, tentang dan exit.

Pertama kali yang penulis lakukan adalah memilih bahasa pemrograman yang akan digunakan juga pokok bahasan atau materi-materi yang akan digunakan. Tahap selanjutnya melakukan pengumpulan data-data baik berupa gambar ataupun artikel dari internet yang berkaitan dengan aplikasi.

4.6 Perbedaan Game iOS dengan Game Android

Faktor aplikasi merupakan hal yang paling menonjol ketika membandingkan iOS dengan Android. Saat ini iOS telah memiliki aplikasi termasuk didalamnya terdapat game dalam jumlah besar pada iTunes mereka yang merupakan sumber penghasilan dari si iOS. walaupun awalnya si Android memiliki lebih sedikit jumlah aplikasi maupun game yang ditawarkan kepada konsumen dibandingkan iOS, tapi kini pengguna Android tidak perlu khawatir karna saat ini banyak developer yang bersaing dalam pembuatan aplikasi dan game Android alhasil kini game Android lebih banyak jumlahnya dibandingkan dengan game iOS, selain itu terdapat kata yang amat disukai banyak konsumen yaitu kata “freedom”. Dengan Android kita dapat melakukan instalasi game Android dari sebuah halaman web (tentunya dengan melakukan download) ataupun bisa saja share game buatan sendiri untuk langsung dicoba di smartphone Android masing-masing. Tidak bisa melakukan hal yang sama seperti Android jika kita bergelut dengan si iOS. Di iOS instalasi game atau aplikasi lain tidak dapat dilakukan jikalau tidak dari iStore, para developerpun tidak mendapat akses untuk melakukan modifikasi fungsi iOS itu sendiri.

Game Android

1. Dibuat dengan Foundation berbasis Java.
2. Bersifat Ekonomis.
3. Perangkat dapat dibeli dengan harga yang relative murah.
4. Terdapat market atau forum yang sifatnya gratis.

Game IOS

1. Dibuat dengan Foundation berbasis Objective-C
2. Bersifat Komersial.
3. Perangkat yang dipasarkan diperuntukan untuk kalangan menengah ke atas.
4. Tidak memiliki market gratis melainkan iStore yang statusnya berbayar.

5.1 Pengertian SQLite

SQLiteDatabase merupakan interface yang ada pada system operasi Android yang digunakan untuk membuat relational database. SQLite menyokong implementasi dari SQL yang kaya untuk apapun yang dibutuhkan oleh aplikasi mobile. Setiap aplikasi dapat memiliki databasenya sendiri dengan pengaturan lengkap.

Dengan penggunaan SQLite dapat dibuat sebuah database untuk aplikasi yang digunakan untuk menyimpan dan mengatur data aplikasi terstruktur. Database Android tersimpan di folder `/data/data/<package_name>/database` device/emulator. Secara default, semua database bersifat private yang hanya diakses oleh aplikasi yang membuatnya. Desain database dengan baik termasuk normalisasi cukup penting untuk mengurangi redundancy.

➤ Cursor dan Content Value

Objek Cursor merupakan nilai kembalian apabila dilakukan Query di Android. Cursor merupakan penunjuk dari serangkaian hasil data.cursor melayani pengaturan posisi pada kumpulan hasil dari query database. Beberapa method yang ada pada kelas cursor .

`moveToFirst ()` : memindahkan cursor ke baris pertama dari hasil Query.

`moveToNext ()` : memindahkan cursor ke baris selanjutnya.

`moveToPrevious ()` : memindahkan cursor ke baris sebelumnya.

`getCount ()` : mengembalikan jumlah baris pada kumpulan hasil.

`getColumnIndexOrThrow ()` : mengembalikan indeks dari kolom sesuai dengan nama yang diberikan atau melemparkan exception jika tidak ada kolom sesuai dengan nama tersebut.

`getColumnName ()` : mengembalikan nama dari indeks kolom yang ditentukan.

`moveToPosition ()` : memindahkan cursor ke baris yang ditentukan.

`getPosition ()` : mengembalikan posisi cursor saat ini.

`ContentValues` digunakan untuk menambahkan baris baru ke tabel.

➤ SQLiteOpenHelper

`SQLiteOpenHelper` adalah kelas abstrak yang diimplementasikan untuk membuat, membuka dan meningkatkan database dengan meng-Override method `onCreate ()` dan `onUpgrade ()`. Method `onCreate ()` dipanggil otomatis ketika aplikasi dijalankan pertama kali, biasanya digunakan untuk membuat database. Jika aplikasi yang dibuat merupakan versi terbaru dn database pada phone perlu diperbarui, maka gunakan method `onUpgrade ()`.

➤ Query database untuk mendapatkan data

Setiap query database dikembalikan sebagai cursor. Untuk mengeksekusi query pada database menggunakan method `query ()`.

```
namaSQLiteDatabase.query(tabel, kolom, seleksi, argument seleksi, groupBy, having, orderBy)
```

jika ingin digunakan sintaks dari bahasa SQL, selain method tersebut dapat juga digunakan method `rawQuery ()`.

```
namaSQLiteDatabase.rawQuery (perintah SQL, argument seleksi)
```

penggunaan method method ini akan mengembalikan nilai sebagai hasil cursor. Cursor ini sebagai penunjuk dari hasil query yang telah dilakukan. Untuk mengambil nilai dari cursor, pertama menggunakan method `moveToPosition` (indeks baris) memindahkan cursor ke posisi baris yang diinginkan. Kemudian ambil data yang ingin diperoleh menggunakan method `get`

‘jenis data’ (indeks kolom). Jenis data merupakan jenis dari data yang ingin diperoleh misalnya string, int, float. Untuk memperoleh jenis data dari kolom tersebut.

➤ Query untuk Insert, Edit, dan Delete

Kelas SQLiteDatabase menyediakan perintah untuk mengeksekusi perintah insert, edit, dan delete. Selain itu, method yang disediakan memungkinkan untuk melakukan suatu perintah SQL yang valid pada tabel database yang diinginkan.

- Insert

Untuk membuat baris baru, dengan membangun sebuah objek ContentValues dan menggunakan method put () untuk memberikan nilai bagi setiap kolom. Insert baris baru dengan melewati objek ContentValues menjadi method insert () pada database target bersama dengan nama tabel.

Sintaks : namaSQLiteDatabase.insert (table, nilai inisial, objek ContentValues).

- Edit

Memperbarui baris juga menggunakan ContentValues.

Pertama buat objek ContentValues baru, menggunakan method put () untuk menetapkan nilai baru untuk setiap kolom yang ingin diperbarui. Panggil method update () pada database, dengan melewati nama tabel, nilai nilai diperbarui objek dan klausa where yang menentukan baris yang akan diperbarui.

Sintaks : namaSQLiteDatabase.update (table, objek ContentValues, klausa where, argument where)

- Delete

Menghapus baris dapat dilakukan dengan menggunakan method delete () pada sebuah database, menentukan nama tabel dan baris mana yang ingin dihapus dengan menggunakan klausa where.

Sintaks : `namaSQLiteDatabase.delete (table, klausa where, argument where)`

Klausa where ini dapat langsung digunakan dengan langsung menulis argmen pada klausa tersebut sedangkan apabila ingin menggunakan argument where dapat digunakan.

Sintaks : `namaSQLiteDatabase.delete (table, “namatabel=?”, argmen where)`

Selain menggunakan method tersebut, dapat digunaka method `execSQL ()`. Method ini digunakan untuk mengeksekusi perintah SQL yang diberikan.

Sintaks : `namaSQLiteDatabase.execSQL (perintah SQL).`

6.1 Pengertian Client Server

Client merupakan sembarang sistem atau proses yang melakukan suatu permintaan data atau layanan ke server sedangkan server ialah, sistem atau proses yang menyediakan data atau layanan yang diminta oleh client.

Client-Server adalah pembagian kerja antara server dan client yg mengakses server dalam suatu jaringan. Jadi arsitektur client-server adalah desain sebuah aplikasi terdiri dari client dan server yang saling berkomunikasi ketika mengakses server dalam suatu jaringan.

1. Servis (layanan)

- Hubungan antara proses yang berjalan pada mesin yang berbeda
- Pemisahan fungsi berdasarkan ide layanannya
- Server sebagai provider, client sebagai konsumen

2. Sharing resources (sumber daya): Server bisa melayani beberapa client pada waktu yang sama, dan meregulasi akses bersama untuk share sumber daya dalam menjamin konsistensinya.

3. Asymmetrical protocol (protokol yang tidak simetris): Many-to-one relationship antara client dan server. Client selalu menginisiasikan dialog melalui layanan permintaan, dan server menunggu secara pasif request dari client.

4. Transparansi lokasi: Proses yang dilakukan server boleh terletak pada mesin yang sama atau pada mesin yang berbeda melalui jaringan. Lokasi server harus mudah diakses dari client.

5. Mix-and-Match : Perbedaan server client platforms

6. Pesan berbasis komunikasi; Interaksi server dan client melalui pengiriman pesan yang menyertakan permintaan dan jawaban.

7. Pemisahan interface dan implementasi: Server bisa diupgrade tanpa mempengaruhi client selama interface pesan yang diterbitkan tidak berubah.

- Client Server System
- Client Server Application

6.2 Fungsi Client Server

Dalam konteks basis data, client mengatur interface berfungsi sebagai workstation tempat menjalankan aplikasi basis data. Client menerima permintaan pemakai, memeriksa sintaks dan generate kebutuhan basis data dalam SQL atau bahasa yang lain. Kemudian meneruskan pesan ke server, menunggu response dan bentuk response untuk pemakai akhir. Server menerima dan memproses permintaan basis data kemudian mengembalikan hasil ke client.

Proses-proses ini melibatkan pemeriksaan otorisasi, jaminan integritas, pemeliharaan data dictionary dan mengerjakan query serta proses update. Selain itu juga menyediakan kontrol terhadap concurrency dan recovery.

Ada beberapa keuntungan jenis arsitektur ini adalah :

- Memungkinkan akses basis data yang besar
- Menaikkan kinerja
- Jika client dan server diletakkan pada komputer yang berbeda kemudian CPU yang berbeda dapat memproses aplikasi secara paralel. Hal ini mempermudah merubah mesin server jika hanya memproses basis data.
- Biaya untuk hardware dapat dikurangi
- Hanya server yang membutuhkan storage dan kekuatan proses yang cukup untuk menyimpan dan mengatur basis data
- Biaya komunikasi berkurang
- Aplikasi menyelesaikan bagian operasi pada client dan mengirimkan hanya bagian yang dibutuhkan untuk akses basis data melewati jaringan, menghasilkan data yang sedikit yang akan dikirim melewati jaringan
- Meningkatkan kekonsistenan
- Server dapat menangani pemeriksaan integrity sehingga batasan perlu didefinisikan dan validasi hanya di satu tempat, aplikasi program mengerjakan pemeriksaan sendiri
- Map ke arsitektur open-system dengan sangat alami

Berikut ini adalah ringkasan fungsi client-server

1 Client

- Mengatur user interface
- Menerima dan memeriksa sintaks input dari pemakai
- Memproses aplikasi
- Generate permintaan basis data dan memindahkannya ke server
- Memberikan response balik kepada pemakai
- Menyediakan akses basis data secara bersamaan
- Menyediakan kontrol recovery

6.3 Aplikasi client server

Istilah arsitektur mengacu pada desain sebuah aplikasi, atau dimana komponen yang membentuk suatu system ditempatkan dan bagaimana mereka berkomunikasi.

Macam-macam arsitektur aplikasi Client-Server beserta kelebihan dan kekurangannya yaitu:

1. Standalone (one-tier)

Pada arsitektur ini semua pemrosesan dilakukan pada mainframe. Kode aplikasi, data dan semua komponen sistem ditempatkan dan dijalankan pada host. Walaupun computer client dipakai untuk mengakses mainframe, tidak ada pemrosesan yang terjadi pada mesin ini, dan karena mereka “dump- client” atau “dump-terminal”. Tipe model ini, dimana semua pemrosesan terjadi secara terpusat, dikenal sebagai berbasis-host. Sekilas dapat dilihat kesalahan pada model ini. Ada dua masalah pada komputasi berbasis host: Pertama, semua pemrosesan terjadi pada sebuah mesin tunggal, sehingga semakin banyak user yang mengakses host, semakin kewalahan jadinya. Jika sebuah perusahaan memiliki beberapa kantor pusat, user yang dapat mengakses mainframe adalah yang berlokasi pada tempat itu, membiarkan kantor lain tanpa akses ke aplikasi yang ada.

Pada saat itu jaringan sudah ada namun masih dalam tahap bayi, dan umumnya digunakan untuk menghubungkan terminal dump dan mainframe. Namun keterbatasan yang dikenakan pada user mainframe dan jaringan telah mulai dihapus.

Keuntungan arsitektur standalone (one-tier):

- Sangat mudah
- Cepat dalam merancang dan mengaplikasikan

Kelemahan arsitektur standalone (one-tier):

- Skala kecil
- Susah diamankan
- Menyebabkan perubahan terhadap salah satu komponen diatas tidak mungkin dilakukan, karena akan mengubah semua bagian.
- Tidak memungkinkan adanya re-usable component dan code.
- Cepat dalam merancang dan mengaplikasikan

2. Client/Server (two tier)

Dalam model client/server, pemrosesan pada sebuah aplikasi terjadi pada client dan server. Client/server adalah tipikal sebuah aplikasi two-tier dengan banyak client dan sebuah server yang dihubungkan melalui sebuah jaringan.

Aplikasi ditempatkan pada computer client dan mesin database dijalankan pada server jarak-jauh. Aplikasi client mengeluarkan permintaan ke database yang mengirimkan kembali data ke client-nya.

Model Two-tier terdiri dari tiga komponen yang disusun menjadi dua lapisan : client (yang meminta service) dan server (yang menyediakan service).

Tiga komponen tersebut yaitu :

1. User Interface. Adalah antar muka program aplikasi yang berhadapan dan digunakan langsung oleh user.
2. Manajemen Proses.
3. Database. Model ini memisahkan peranan user interface dan database dengan jelas, sehingga terbentuk dua lapisan.

Kelebihan dari model client/server

- Mudah
- Menangani Database Server secara khusus
- Relatif lebih sederhana untuk di develop dan diimplementasikan.
- Lebih cocok diterapkan untuk bisnis kecil.

Server database berisi mesin database, termasuk tabel, prosedur tersimpan, dan trigger (yang juga berisi aturan bisnis). Dalam system client/server, sebagian besar logika bisnis biasanya diterapkan dalam database.

Server database menangani :

- Manajemen data
- Keamanan
- Query, trigger, prosedur tersimpan
- Penangan kesalahan

Arsitektur client/server merupakan sebuah langkah maju karena mengurangi beban pemrosesan dari komputer sentral ke computer client. Ini berarti semakin banyak user bertambah pada aplikasi client/server, kinerja server file tidak akan menurun dengan cepat. Dengan client/server user dari berbagai lokasi dapat mengakses data yang sama dengan sedikit beban pada sebuah mesin tunggal.

Namun masih terdapat kelemahan pada model ini. Selain menjalankan tugas-tugas tertentu, kinerja dan skalabilitas merupakan tujuan nyata dari sebagian besar aplikasi.

Kekurangan dari model client/server :

- Kurangnya skalabilitas
- Koneksi database dijaga
- Tidak ada keterbaharuan kode
- Tidak ada tingkat menengah untuk menangani keamanan dan transaksi skala kecil.
- Susah di amankan.
- Lebih mahal.

3. Three Tier

Arsitektur Three Tier merupakan inovasi dari arsitektur Client Server. Pada arsitektur Three Tier ini terdapat Application Server yang berdiri di antara Client dan Database Server. Contoh dari Application server adalah IIS, WebSphere, dan sebagainya.

Application Server umumnya berupa business process layer, dimana bisa didevelop menggunakan PHP, ASP.Net, maupun Java. Sehingga kita menempatkan beberapa business logic kita pada tier tersebut. Arsitektur Three Tier ini banyak sekali diimplementasikan dengan menggunakan Web Application. Karena dengan menggunakan Web Application, Client Side (Komputer Client) hanya akan melakukan instalasi Web Browser. Dan saat komputer client melakukan inputan data, maka data tersebut dikirimkan ke Application Server dan diolah berdasarkan business process-nya. Selanjutnya Application Server akan melakukan komunikasi dengan database server.

Biasanya, implementasi arsitektur Three Tier terkendala dengan network bandwidth. Karena aplikasinya berbasis web, maka Application Server selalu mengirimkan Web Application-nya ke computer Client. Jika kita memiliki banyak sekali client, maka bandwidth

yang harus disiapkan akan cukup besar, Sedangkan network bandwidth biasanya memiliki limitasi. Oleh karena itu biasanya, untuk mengatasi masalah ini, Application Server ditempatkan pada sisi client dan hanya mengirimkan data ke dalam database server. Konsep model three-tier adalah model yang membagi fungsionalitas ke dalam lapisan-lapisan, aplikasiaplikasi mendapatkan skalabilitas, keterbaharuan, dan keamanan.

Kelebihan arsitektur Three Tier :

- Segala sesuatu mengenai database terinstalasikan pada sisi server, begitu pula dengan pengkonfigurasinya. Hal ini membuat harga yang harus dibayar lebih kecil.
- Apabila terjadi kesalahan pada salah satu lapisan tidak akan menyebabkan lapisan lain ikut salah
- Perubahan pada salah satu lapisan tidak perlu menginstalasi ulang pada lapisan yang lainnya dalam hal ini sisi server ataupun sisi client.
- Skala besar.
- Keamanan dibelakang firewall.
- Transfer informasi antara web server dan server database optimal.
- Komunikasi antara system-sistem tidak harus didasarkan pada standart internet, tetapi dapat menggunakan protocol komunikasi yang lebih cepat dan berada pada tingkat yang lebih rendah.
- Penggunaan middleware mendukung efisiensi query database dalam SQL di pakai untuk menangani pengambilan informasi dari database.

Kekurangan arsitektur Three Tier :

- Lebih susah untuk merancang
- Lebih susah untuk mengatur
- Lebih mahal

4. Multi Tier

Arsitektur Multi Tier adalah suatu metode yang sangat mirip dengan Three Tier. Bedanya, pada Multi Tier akan diperjelas bagian UI (User Interface) dan Data Processing. Yang membedakan arsitektur ini adalah dengan adanya Business Logic Server. Database Server dan Bussines Logic Server merupakan bagian dari Data Processing, sedangkan Application Server dan Client/Terminal merupakan bagian dari UI. Business Logic Server biasanya masih menggunakan bahasa pemrograman terdahulu, seperti COBOL. Karena sampai saat ini, bahasa pemrograman tersebut masih sangat mumpuni sebagai business process.

Multi-tier architecture menyuguhkan bentuk three – tier yang diperluas dalam model fisik yang terdistribusi. Application server dapat mengakses Application server yang lain untuk mendapat data dari Data server dan mensuplai servis ke client Application.

Kelebihan arsitektur Multi tier :

- Dengan menggunakan aplikasi multi-tier database, maka logika aplikasi dapat dipusatkan pada middle-tier, sehingga memudahkan untuk melakukan control terhadap client-client yang mengakses middle server dengan mengatur seting pada dcomcnfg.
- Dengan menggunakan aplikasi multi-tier, maka database driver seperti BDE/ODBC untuk mengakses database hanya perlu diinstal sekali pada middle server, tidak perlu pada masing-masing client.
- Pada aplikasi multi-tier, logika bisnis pada middle-tier dapat digunakan lagi untuk mengembangkan aplikasi client lain, sehingga mengurangi besarnya program untuk mengembangkan aplikasi lain. Selain itu meringankan beban pada tiap-tiap mesin karena program terdistribusi pada beberapa mesin.
- Memerlukan adaptasi yang sangat luas ruang lingkupnya apabila terjadi perubahan sistem yang besar.

Kekurangan arsitektur Multi tier :

- Program aplikasi tidak bisa mengquery langsung ke database server, tetapi harus memanggil prosedur-prosedur yang telah dibuat dan disimpan pada middle-tier.
- Lebih mahal

Keunggulan client server

- Kecepatan akses lebih tinggi
- Sistem keamanan & administrasi lebih baik
- Sistem backup data lebih baik

Kelemahan Client/Server

- Biaya lebih mahal
- Dibutuhkan komputer dengan spesifikasi khusus untuk menjadi server
- Ketergantungan terhadap server, jika server terganggu maka keseluruhan jaringan terganggu

7.1. Google Maps

Google Maps merupakan sebuah layanan peta dunia virtual berbasis web yang dikembangkan oleh Google. Layanan ini gratis dan dapat dengan mudah ditemukan di <http://maps.google.com>. Google Maps menawarkan peta yang dapat digeser (panned), diperbesar (zoom in), diperkecil (zoom out), dapat diganti dalam beberapa mode (map, satelit, hybrid dan lain-lain), fitur pencarian rute (routing), penunjuk arah dari suatu objek peta ke objek yang lain (direction) dan juga pencari tempat (place). Google Maps akan menampilkan peta sebuah wilayah yang dibuat secara digital. Sehingga, cukup mudah untuk dapat mengetahui peta sebuah wilayah, cukup dengan masuk ke situs google map tersebut dan menekan tombol atau mengetikkan sebuah alamat. Maka, lokasi yang dicari atau diinginkan akan segera ditemukan tanpa menunggu waktu lama.

Google map dapat disertai pada third-party website (website pengguna) dengan menggunakan Google Maps API. Layanan Google Maps ini merangkum peta jalanan, perencanaan rute jalan untuk pejalan kaki, mobil, sepeda dan juga transportasi umum. Selain itu Google Maps juga digunakan sebagai penanda lokasi / alamat dari badan usaha / organisasi di beberapa negara di dunia. Data – data peta yang ada pada Google Maps tersebut tidak terupdate secara Real-Time, namun Google menyimpan data dari gambar satelit tersebut pada database utama dan kebanyakan gambar yang ada pada layanan tersebut diambil kurang dari 3 tahun lamanya.

Google Maps tidak dapat menampilkan secara akurat area di sekitar kutub (baik selatan maupun utara), hal ini berbeda dengan aplikasi Google Earth yang dapat menampilkan fitur tampilan bumi secara keseluruhan, termasuk daerah kutub. Google Maps untuk perangkat mobile adalah yang paling terkenal di aplikasi smartphone, aplikasi tersebut digunakan sebanyak 54% dari total keseluruhan pengguna smartphone yang ada (hingga Agustus 2013).

7.1.1. Satellite view

Google Maps menyediakan gambar udara dari satelit dengan resolusi tinggi dari seluruh dunia. Namun, tidak semua daerah pada gambar satelit tersebut memiliki resolusi yang sama, area yang berpenduduk rendah biasanya tampil dengan detail gambar yang kurang. Walaupun Google menggunakan satelit dalam pengambilan gambar tersebut, kadangkala gambar perkotaan dengan resolusi tinggi diambil dari pesawat terbang di ketinggian 240 meter sampai 460 meter, namun tetap kebanyakan berasal dari satelit.

7.1.2. Directions

Google Maps menyediakan layanan perencana rute dalam fitur “Get Directions” pada aplikasi website. Terdapat 4 tipe mode transportasi yang dapat digunakan dalam area yaitu driving, public Transit, walking dan bicycling. Dalam Google Street View, hal hal seperti parkir, putaran arah dan jalan satu arah dapat dilihat sebelum memulai rute.

7.2. Google Maps API

Google merilis Google Maps API pada bulan Juni 2005 agar pada Developer dapat mengintegrasikan Google Maps ke website. Layanan tersebut gratis dan juga tidak mengandung iklan, tetapi Google menerapkan suatu peraturan dalam penggunaan dimana suatu waktu Google berhak untuk menaruh iklan pada layanan Google Maps di Google Maps API.

Dengan menggunakan Google Maps API, maka dapat meng-embed Google Maps ke situs eksternal lainnya. Google Maps API telah disertakan API untuk Adobe Flash, layanan untuk gambar peta statis, selain itu terdapat fitur geocoding, fitur driving directions. Lebih dari 1 juta website telah menggunakan Google Maps API, sehingga menjadikannya API aplikasi web yang paling banyak digunakan.

Google Maps API juga dapat digunakan secara komersial dengan gratis dengan syarat sebagai berikut:

1. Situs web yang menggunakan dapat diakses secara publik.
2. Tidak mengenakan tarif akses kepada pengguna web tersebut.
3. Tidak men-generate lebih dari 25.000 akses peta dalam sehari.

Apabila website tidak dapat memenuhi persyaratan tersebut diatas, maka diharuskan website tersebut untuk dikenakan biaya penggunaan layanan Google Maps API khusus Bisnis. Kesuksesan Google Maps API telah membangkitkan sejumlah kompetitor alternatif untuk layanan peta berbasis web seperti Yahoo! Maps API, Bing Maps Platform, MapQuest Development Platform, dan juga OpenLayers.

7.3. Google Maps Untuk Perangkat Mobile

Pada Oktober 2005, Google memperkenalkan sebuah aplikasi berbasis java bernama Google Maps for Mobile, yang dijalankan pada ponsel atau perangkat mobile berbasis Java, terdapat banyak fitur dari versi web yang juga disertai pada aplikasi mobile tersebut.

Pada tanggal 4 Oktober 2009, Google Maps Navigation dirilis oleh Google untuk Android 2.0 Éclair pada Motorola Droid dengan fitur voice commands, traffic reports dan street view support. Pada Maret 2011, Vice President of Location Service Google, Marrison Mayer mengumumkan bahwa layanan peta Google telah digunakan oleh 150 juta pengguna.

Pada bulan Juni 2012, Apple mengumumkan bahwa akan mengganti aplikasi Google Maps dengan aplikasi layanan peta dari Apple sendiri dari iOS 6. Namun, pada Desember 2012, Google mengumumkan ketersediaan Google Maps untuk perangkat iOS Apple di Apple App Store. Hanya berlangsung 1 jam sejak dirilis, Google Maps iOS menjadi top free app di App Store.

7.4. Google Maps Android API v2

Untuk membuat aplikasi android baru yang menggunakan Google Maps Android API v2, dibutuhkan beberapa tahap. Beberapa dari langkah langkah tersebut hanya akan dilakukan sekali saja, tetapi beberapa informasi akan menjadi referensi penting untuk pengembangan aplikasi kedepannya. Berikut adalah langkah langkah proses untuk menambahkan peta di Aplikasi Android :

1. Install Android SDK.
2. Download dan konfigurasi Google Play Service SDK yang telah disertai dengan Google Maps Android API melalui SDK Manager. Jika akan menggunakan Google Maps Mobile

SDK untuk Business maka harus men-download versi Google Maps Mobile SDK for Business static library.

3. Men-generate kode API key. Untuk mendapatkan kode API Key, developer harus mendaftarkan proyek pada Google APIs Console, dan mendapatkan tanda certificate untuk aplikasi.
4. Menambahkan beberapa konfigurasi penting pada Android application's manifest (lihat pada bagian 7.4.1. “Menspesifikasi Permissions Pada Android Manifest”).
5. Menambahkan peta ke dalam aplikasi.
6. Mempublikasi aplikasi.

7.4.1. Menspesifikasi Permissions Pada Android Manifest

Untuk dapat menggunakan akses Google Maps pada aplikasi android, maka developer harus menspesifikasikan permissions yang dibutuhkan pada aplikasi, dengan menambahkan tag `<uses-permission>` elements sebagai anak dari tag `<manifest>` element. Berikut adalah sintaks umumnya :

```
<uses-permissionandroid:name="permission_name"/>
```

Sebagai contoh, agar aplikasi dapat mengakses internet maka butuh menambahkan request Internet permission, seperti berikut:

```
<uses-permissionandroid:name="android.permission.INTERNET"/>
```

Selain digunakan untuk bagian dari aplikasi, permission juga diperlukan untuk dapat menggunakan Google Maps Android API. Berikut adalah beberapa permission yang perlu disertakan di AndroidManifest.XML agar dapat menggunakan Google Maps Android API :

- **android.permission.INTERNET**, Digunakan oleh API untuk mendownload proyeksi peta dari Google Maps Server dengan Internet.

- `android.permission.ACCESS_NETWORK_STATE`, Mengizinkan API untuk mengecek status koneksi perangkat untuk dapat mengalokasikan data data yang akan didownload.
- `com.google.android.providers.gsf.permission.READ_GSERVICES`, Memberikan Akses kepada API untuk mengakses layanan Google berbasis web.
- `android.permission.WRITE_EXTERNAL_STORAGE`, Mengizinkan API untuk menyimpan cache map tile data padamedia penyimpanan external perangkat.
- `android.permission.ACCESS_COARSE_LOCATION`, Mengizinkan API untuk dapat menggunakan WiFi ataupun paket data untuk mengirimkan lokasi perangkat.
- `android.permission.ACCESS_FINE_LOCATION`, Mengizinkan API menggunakan Global Positioning System (GPS) untuk mengirimkan lokasi perangkat dari area yang lebih detail.

Berikut adalah contoh penggunaan permission pada AndroidManifest :

```
<uses-permission android:name="android.permission.INTERNET"/>

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>

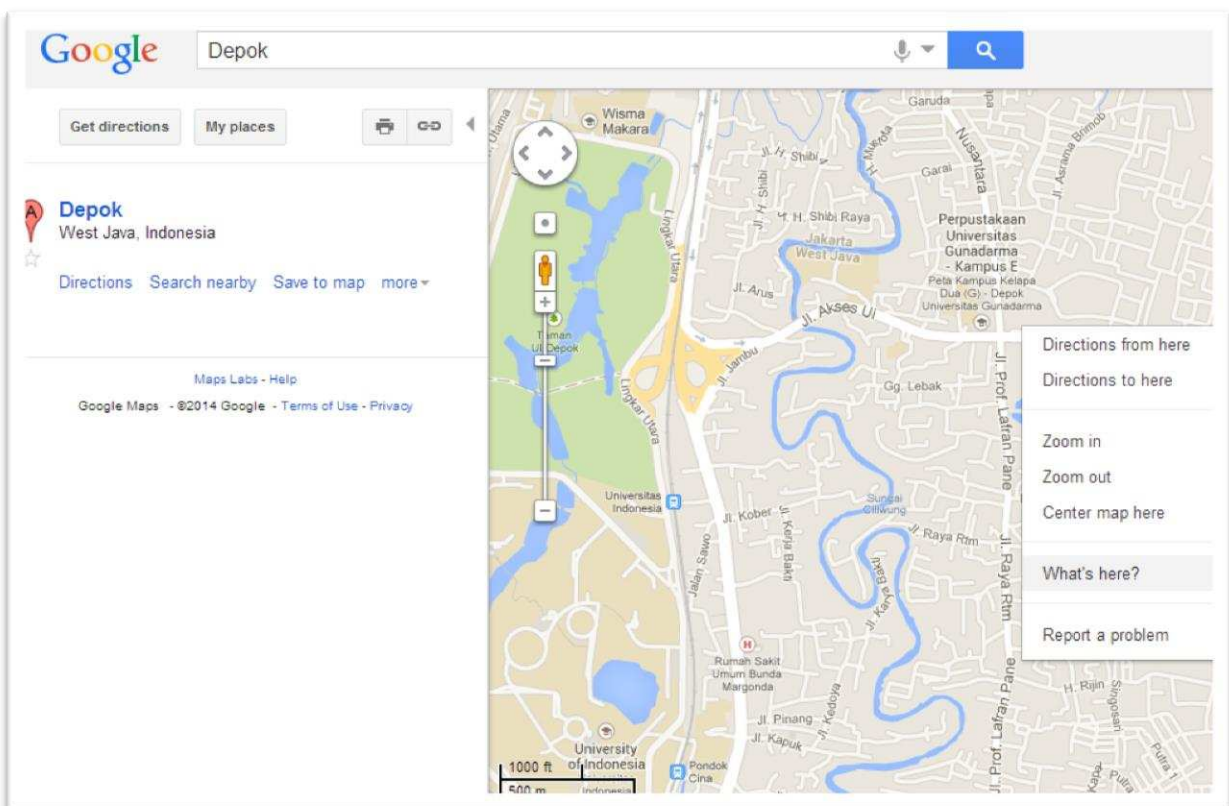
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

7.4.2. Geo Point

Geo Point adalah sebuah kelas tetap yang mewakili sepasang lintang dan bujur, disimpan sebagai angka integer dari microdegrees. Geo Point biasanya digunakan dalam aplikasi pemetaan, untuk memanggilnya dapat digunakan perintah import sebagai berikut:

```
import com.google.android.maps.GeoPoint;
```

GeoPoint bisa didapatkan salah satunya dengan cara melihat pada Google Maps. Misalkan arah tujuan menuju ke Indonesia. Dan disini akan diambil GeoPoint untuk kota Depok dengan titik Universitas Gunadarma. Klik kanan, lalu pilih What's here?



Gambar 7.1 Mendapatkan Geo Point dari Google Maps

Sehingga pada jendela sebelah kiri akan muncul angka GeoPoint dari titik tersebut, atau pada field searching. Untuk Universitas Gunadarma kita mendapatkan GeoPoint dengan deretan angka, -6.36806, 106.833029.

7.4.3. Map Fragment

`MapFragment`, adalah subclass dari `classFragment` di Android, berfungsi untuk menempatkan map ke dalam Android Fragment. Objek `MapFragment` bekerja sebagai penampung dari peta dan menghubungkan akses ke objek `GoogleMap`.

Tidak seperti `View`, `Fragment` memisahkan setiap bagian dari user interface pada Activity. Sehingga dapat menggabungkan beberapa fragment ke dalam single activity untuk membuat tampilan multi panel pada User Interface dan dapat menggunakan ulang fragment tersebut dalam activity lainnya.

7.4.4. Map View

`MapView`, adalah subclass dari `classView`, berfungsi untuk menempatkan map pada view di Android. Sebuah view dapat di representasikan sebagai bentuk persegi dalam area layar, dan membentuk blok-blok pada aplikasi android dengan widget-widget. Seperti halnya `MapFragment`, `MapView` juga berfungsi sebagai container / penampung dari Map untuk `GoogleMap` object.

7.4.5. Tipe – Tipe Google Maps Android API

Ada berbagai macam tipe maps yang tersedia oleh Google Maps Android API. Tipe peta pada dasarnya merepresentasikan peta. Contohnya, sebuah atlas biasanya terdapat peta politik yang focus untuk menunjukkan batas dari negara-negara berdasarkan warna dan peta jalan yang fokus untuk menunjukkan jalan pada suatu kota atau negara.

Google Maps Android API menyediakan berbagai macam peta, yaitu sebagai berikut :

1. Normal

Tipe peta jalanan. Menampilkan fokus jalanan, tempat - tempat penting dan tempat alam seperti sungai. Nama jalan juga ditampilkan.

2. Hybrid

Foto gambaran satelit peta jalanan. Hanya jalan dan nama jalan yang ditampilkan.

3. Satellite

Foto gambaran jalan dari data satelit, namun jalan dan nama jalan tidak ditampilkan.

4. Terrain

Foto topografi dari jalan, peta terdapat warna-warna, corak garis dan label. Beberapa jalan dan nama jalan ditampilkan.

7.4.6. Cara Mengubah Map Type

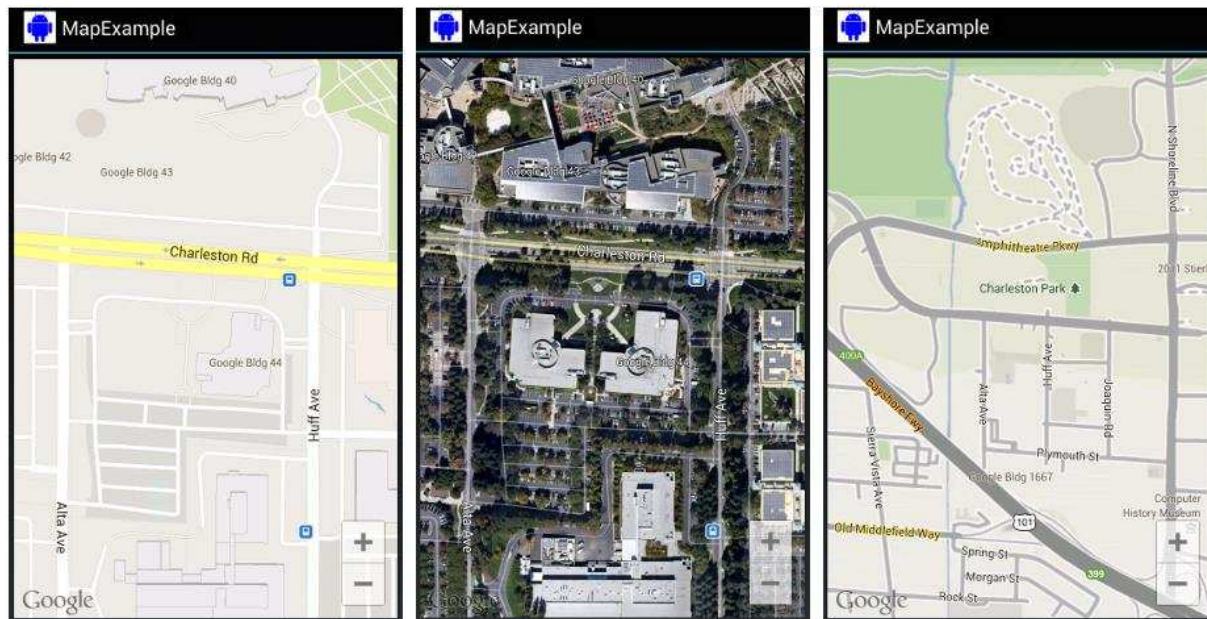
Untuk mengubah tipe peta pada Google Maps API Android adalah dengan memanggil objek `GoogleMap` dengan method `setMapType` dan kemudian mengisi salah satu tipe konstanta dari tipe peta. Sebagai contoh, untuk menampilkan peta tipe hybrid adalah dengan cara sebagai berikut :

```
// deklarasi objek GoogleMap
GoogleMapmap;

...

// men-Set tipe peta menjadi "hybrid"
map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
```

Setelah itu jalankan aplikasi maka akan tampak perbedaan tipe peta tersebut, seperti pada gambar dibawah ini :



Gambar 7.2 Perbedaan Tipe Peta Google Maps

7.4.7. Marker Manager

Google Maps API sebetulnya sudah menyediakan komponen Marker Manager yang berfungsi untuk mengoptimalkan penampilan marker ketika kita menavigasikan peta namun ada banyak juga Marker Manager yang dibuat oleh orang lain yang lebih bagus. Dengan menggunakan komponen ini objek marker pada peta di Google Maps dapat diatur secara otomatis mana yang perlu ditampilkan dan mana yang tidak secara otomatis.

Namun untuk optimalisasi proses penginputan data dari database masih harus dibuat manual, karena jika kita memasukkan data berjumlah besar seperti 500 marker sekaligus akan membuat page load terlalu lama, namun navigasi peta tetap lancar karena telah diatur oleh Marker Manager. Marker adalah simbol yang menandakan suatu lokasi wisata pada peta yang ditampilkan Google Maps. Google Maps API juga mendukung membuat tanda menggunakan Google Map Marker. Tanda Google Map Marker diciptakan melalui kontribusi dari seluruh dunia, dan area yang belum tercakup oleh Google Maps.

8.1 Sejarah Socket

Socket adalah mekanisme komunikasi yang memungkinkan terjadinya pertukaran data antar program atau proses baik dalam satu mesin maupun antar mesin. Gaya pemrograman socket sendiri berawal dari sistem Unix BSD yang terkenal dengan kepeloporannya pada bidang penanganan jaringan, sehingga sering disebut BSD Socket. Socket pertama kali diperkenalkan di sistem Unix BSD versi 4.2 tahun 1983 sebagai kelanjutan dari implementasi protokol TCP/IP yang muncul pertama kali pada sistem Unix BSD 4.1 pada akhir 1981. Hampir setiap variant Unix dan Linux mengadopsi BSD Socket. Pada lingkungan Unix, socket memberikan keleluasaan pemrograman gaya Unix yang terkenal dengan ideologinya, Semua di Unix/Linux adalah file. Komunikasi antar program dapat berlangsung lewat penggunaan deskriptor file standar Unix dengan bantuan socket.

Keunggulan dari penggunaan socket adalah anda dapat melakukan komunikasi antar proses/program melalui jaringan berbasis yang TCP/IP tentunya, bahkan dengan program lain yang berjalan pada platform non-unix seperti Microsoft Windows, sepanjang program tersebut berbicara dalam protokol transfer yang sama. Fasilitas-fasilitas yang disediakan oleh mesin unix seperti rlogin, ssh, ftp, dan lain-lain menggunakan socket sebagai sarana komunikasi mereka. Socket dibentuk dan digunakan dengan cara yang berbeda. Komunikasi socket terutama diciptakan untuk tujuan menjembatani komunikasi antara dua buah program yang dijalankan pada mesin yang berbeda. Jangan khawatir, ini tentu saja berarti dua program pada mesin yang sama dapat juga saling berkomunikasi. Kelebihan lain dari komunikasi socket adalah mampu menangani banyak klien sekaligus (multiple clients).

8.1.1 Jenis Socket

Ada dua golongan socket di Unix yang paling umum dipakai yaitu:

1. Socket Lokal atau AF_UNIX

Socket Lokal adalah socket yang melakukan komunikasi dengan perantara sebuah file yang biasanya diletakkan pada direktori /tmp atau /usr/tmp ataupun /var/tmp. Socket semacam ini digunakan umumnya terbatas untuk komunikasi antar aplikasi dalam satu mesin.

2. Socket Networking atau AF_INET

Socket Networking ditujukan untuk komunikasi antar aplikasi antar mesin dalam lingkungan jaringan TCP/IP. Identifikasi socket dilakukan dengan sebuah service identifieryaitu berupa nomor port TCP/IP yang dapat di sambung oleh client.

Socket Networking memiliki beberapa jenis, yang paling umum digunakan yaitu:

a. Socket Stream atau SOCK_STREAM

Socket Stream adalah socket komunikasi full-duplexberbasis aliran (stream) data. Pada model komunikasi Socket Stream, koneksi dua aplikasi harus dalam kondisi tersambung dengan benar untuk dapat bertukar data. Ini dapat dianalogikan seperti komunikasi telepon. Jika sambungan telepon di salah satu titik putus, maka komunikasi tidak dapat terjadi. Koneksi model seperti ini akan menjamin data dapat dipertukarkan dengan baik, namun memiliki kelemahan dalam hal penggunaan jalur data yang relatif besar dan tidak boleh terputus.

b. Socket Datagram atau SOCK_DGRAM

Socket Datagram berkomunikasi dengan cara yang berbeda. Socket ini tidak membutuhkan koneksi yang tersambung dengan benar untuk mengirimkan dan menerima data. Model koneksi semacam ini tidak dapat menjamin data dapat dipertukarkan dengan baik, namun memiliki keunggulan dalam hal penggunaan jalur data yang minimal. Socket Datagram dapat dianalogikan dengan komunikasi yang terjadi pada kelas, misalnya pada saat guru melakukan broadcastingmateri pelajaran untuk diterima oleh setiap murid. Tidak ada yang dapat menjamin materi pelajaran dapat diterima oleh semua murid dengan baik, kecuali diterapkan metoda rechecking. Rechecking ini dapat dilakukan baik oleh guru maupun murid. Guru bertanya untuk memastikan jawaban dari murid benar, atau murid bertanya untuk memastikan kebenaran materi yang diterimanya. Socket

Datagram pun menggunakan metoda ini untuk menjamin pengiriman data dapat dilakukan dengan baik.

8.1.2 Sekilas Tentang Socket, TCP Dan UDP

1. Mengenal Socket

Pengertian socket adalah interface pada jaringan yang menjadi titik komunikasi antarmesin pada Internet Protocol, dan tentunya tanpa komunikasi ini, tidak akan ada pertukaran data dan informasi jaringan.

Socket terdiri dari elemen-elemen utama sebagai berikut:

1. Protokol.
2. Local IP.
3. Local Port.
4. Remote IP.
5. Remote Port.

Dalam komunikasi antara dua pihak, tentunya harus digunakan kesepakatan aturan dan format yang sama agar komunikasi dapat dimengerti. Seperti halnya dua orang yang menggunakan bahasa yang sama, maka bahasa di sini berfungsi sebagai protokol. Protokol yang digunakan dalam socket dapat menggunakan TCP ataupun UDP.

Contoh komunikasi sederhana adalah komunikasi antara komputer A dan komputer B. Baik komputer A maupun komputer B harus memiliki identitas unik, yang direpresentasikan oleh IP masing-masing. Komunikasi yang terjadi melalui port, sehingga baik komputer A maupun komputer B harus memiliki port yang dapat diakses satu sama lain.

2. TCP dan UDP

Pemrograman socket adalah cara untuk menggunakan komponen/API (Application Programming Interface) socket untuk membuat sebuah aplikasi.

Aplikasi socket umumnya terdiri dari dua kategori berdasarkan pengiriman datanya, yaitu:

- a. Datagram socket (menggunakan UDP).
- b. Stream socket (menggunakan TCP).

Terdapat perlakuan yang berbeda antara UDP dan TCP, walaupun sama-sama berfungsi sebagai protokol pertukaran data.

UDP tidak memerlukan proses koneksi terlebih dahulu untuk dapat mengirimkan data, paket-paket data yang dikirimkan UDP bisa jadi melalui rute yang berbeda-beda, sehingga hasil yang diterima bisa jadi tidak berurutan.

Contohnya jika aplikasi socket pengirim mengirimkan berturut-turut pesan 1, pesan 2, dan pesan 3, maka aplikasi socket penerima belum tentu mendapatkan pesan yang berurutan dimulai dari pesan 1, pesan 2, dan terakhir pesan 3. Bisa saja pesan 2 terlebih dulu diterima, menyusul pesan-pesan yang lain, atau berbagai kemungkinan lainnya. Bahkan, dapat terjadi pesan yang dikirimkan tidak sampai ke penerima karena kegagalan pengiriman paket data.

Tidak demikian halnya dengan stream socket yang menggunakan TCP. Jenis ini mengharuskan terjadinya koneksi terlebih dahulu, kemudian mengirimkan paket-paket data secara berurutan, penerima juga dijamin akan menerima data dengan urutan yang benar, dimulai dari data pertama yang dikirimkan hingga data terakhir. TCP dapat menangani data yang hilang, rusak, terpecah, ataupun terduplikasi.

Dari sekilas perbedaan ini, kita dapat menarik kesimpulan bahwa aplikasi socket yang menggunakan TCP memerlukan pertukaran data dua arah yang valid. Sedangkan, aplikasi socket yang menggunakan UDP lebih memprioritaskan pada pengumpulan data.

Karena itu aplikasi socket dengan TCP sering diterapkan untuk aplikasi chat, transfer file, ataupun transaksi-transaksi penting. Sedangkan aplikasi socket dengan UDP cocok diterapkan untuk aplikasi monitoring jaringan, game online, dan aplikasi-aplikasi broadcast.

3. Socket Programming

Socket adalah sebuah cara untuk berkomunikasi dengan program atau node lain menggunakan file deskriptor. Di UNIX (dimana socket diciptakan) kita sering mendengar slogan: “everything is a file”, jadi untuk berkomunikasi dengan program atau node lain semudah kita membaca dan menulis file deskriptor. Antarmuka socket dan file adalah mirip, jikapada file kita membukanya dengan `open()` sedangkan pada socket kita menggunakan `socket()`. Pada file deskriptor yang menjadi tujuan adalah sebuah file, sedangkan pada socket adalah computer atau node lain. Intinya ketika kita telah terhubung dengan socket(), maka antarmukanya sama saja

dengan sebuah file. Sebuah abstraksi perangkat lunak yang digunakan sebagai suatu “terminal” dari suatu hubungan antara dua mesin atau proses yang saling berinterkoneksi

Penggunaan socket programming memungkinkan adanya komunikasi antara client dan server. Salah satu contoh sederhana penggunaan socket programming adalah pembuatan program untuk chatting. Program tersebut sebenarnya merupakan bentuk aplikasi berupa komunikasi antara client dan server. Ketika seorang user (client) melakukan koneksi ke chat server, program akan membuka koneksi ke port yang diberikan, sehingga server perlu membuka socket pada port tersebut dan “mendengarkan” koneksi yang datang. Socket sendiri merupakan gabungan antara host-adress dan port adress. Dalam hal ini socket digunakan untuk komunikasi antara client dan server.

Socket merupakan fasilitas IPC (Inter Proses Communication) untuk aplikasi jaringan. Agar suatu socket dapat berkomunikasi dengan socket lainnya, maka socket butuh diberi suatu alamat unik sebagai identifikasi. Alamat socket terdiri atas Alamat IP dan Nomer Port. Contoh alamat socket adalah **192.168.29.30: 3000**, dimana nomer 3000 adalah nomer portnya. Alamat IP dapat menggunakan alamat Jaringan Lokal (LAN) maupun alamat internet. Jadi socket dapat digunakan untuk IPC pada LAN maupun Internet.

4. Port dan Winsock

1. Port

Salah satu elemen penting yang digunakan dalam aplikasi socket adalah port. Port merupakan sebuah koneksi data virtual yang digunakan aplikasi untuk bertukar data secara langsung.

Terdapat banyak port di dalam sebuah sistem komputer dengan fungsinya masing-masing. Sebagai contoh, dalam mengirim e-mail digunakan service SMTP yang umumnya menggunakan port 25. Sementara service POP3 untuk menerima e-mail menggunakan port 110, port 80 digunakan untuk HTTP, port 443 digunakan untuk HTTPS, dan seterusnya.

Nomor-nomor port dikategorikan dalam tiga jenis sebagai berikut:

- a. Well-known ports.

Merupakan port yang telah digunakan secara internal oleh sistem Windows, misalnya port untuk koneksi Internet, service FTP, dan seterusnya. Port yang telah digunakan ini adalah port 0 sampai dengan port 1023.

b. Registered ports.

Port ini dapat digunakan dalam aplikasi Anda, range-nya adalah port 1024 hingga port 49151, cukup banyak port yang tersedia yang bebas Anda pilih sehingga Anda tidak perlu khawatir kekurangan port untuk aplikasi Anda.

c. Dynamic/Private ports.

Dari port 49152 sampai dengan port 65535.

2. Winsock

Untuk pemrograman aplikasi socket berbasis Windows, maka komponen API yang sering digunakan adalah Winsock (Win-dows Socket API) yang mendukung interface standar TCP/IP, yang merupakan protokol jaringan paling populer saat ini (contoh protokol jaringan yang lain adalah NetBIOS, IPX dari Novell, AppleTalk dari Apple, dan sebagainya).

Pengertian TCP/IP (TCP over IP) mungkin dapat menjadi sedikit rancu jika diartikan TCP/IP hanya mengizinkan pengiriman TCP (dan tidak UDP), padahal seperti yang telah kita bahas, pengiriman socket dapat melalui TCP maupun UDP.

Pengertian TCP/IP di sini sebenarnya digunakan untuk menunjukkan teknologi jaringan/Internet, termasuk di dalamnya adalah UDP. Jika Anda menggunakan UDP, dapat juga disebut sebagai UDP/IP (UDP over IP), tetapi umumnya istilah ini jarang digunakan dan istilah TCP/IP telah mencakup, baik TCP maupun UDP.

Pada bahasa pemrograman visual seperti Visual Basic/Delphi, Anda dapat menggunakan control Winsock yang telah disediakan untuk mengembangkan aplikasi socket.

Walaupun kita akan mencontohkan aplikasi socket dalam environment Windows, Anda tidak perlu khawatir jika aplikasi socket yang menggunakan Winsock tidak dapat berkomunikasi dengan aplikasi socket berbasis Unix/Linux, karena komunikasi tetap dapat terjadi selama aplikasi tersebut menggunakan protokol jaringan yang sama.

Bagi Anda yang terpaksa hanya menggunakan satu komputer, dapat memanfaatkan alamat local host atau 127.0.0.1 yang mengizinkan dua aplikasi berjalan pada satu mesin komputer dan berkomunikasi satu sama lain.

8.2 Model Aplikasi Client Server



Untuk membuat aplikasi socket yang sederhana diperlukan dua aplikasi. Yaitu, pertama adalah aplikasi server yang akan menerima data, sedangkan aplikasi kedua adalah aplikasi client yang mengirimkan data pada server. Baik aplikasi server dan aplikasi client mendefinisikan port yang sama sebagai jalur komunikasi.

Obyek socket pada sisi client dan server berbeda sedikit. Pada sisi aplikasi server, suatu socket server dibentuk dan melakukan operasi listen/menunggu. Operasi ini pada intinya menunggu permintaan koneksi dari sisi client. Sedangkan pada sisi client, dibentuk suatu socket biasa.

Pada saat socket client, informasi alamat socket server dilewatkan sebagai argumen dan socket client akan otomatis mencoba meminta koneksi ke socket server. Pada saat permintaan koneksi client sampai pada server, maka server akan membuat suatu socket biasa. Socket ini yang nantinya akan berkomunikasi dengan socket pada sisi client. Setelah itu socket server dapat kembali melakukan listen untuk menunggu permintaan koneksi dari client lainnya. Langkah ini umumnya hanya dilakukan jika aplikasi server mengimplementasikan multithreading.

Setelah tercipta koneksi antara client dan server, maka keduanya dapat saling bertukar pesan. Salah satu atau keduanya kemudian dapat mengakhiri komunikasi dengan menutup socket.

Untuk protokol UDP, perbedaannya adalah socket di sisi server sama dengan socket di sisi client, dan tidak ada operasi listen pada sisi server. Kemudian saat paket data dikirimkan, alamat socket penerima harus disertakan sebagai argumen.

8.3 Kelas Tambahan

Kelas java.net.Socket

Kelas ini digunakan untuk melakukan koneksi ke suatu mesin atau proses yang dituju.

Kelas java.net.ServerSocket

Kelas ini berfungsi untuk listen koneksi yang sudah terjadi dan siap menerima permintaan dari proses lain. Kelas ini digunakan ketika pembuatan aplikasi server yang bersifat Connection Oriented. Ketika dibuat obyek baru ServerSocket, untuk dapat menerima dan mengirim melalui stream perlu dibuat InputStream dan OutputStream. ServerSocket siap menerima permintaan client ketika menjalankan method accept() dan akan mengakhiri koneksi ketika menjalankan method close()

Kelas java.net.ServerSocket digunakan oleh Server untuk listen koneksi, sedangkan java.net.Socket digunakan oleh Client untuk inisialisasi koneksi. Setelah client terkoneksi ke server dengan menggunakan Socket, maka ServerSocket akan mengembalikan status server ke client melalui koneksi yang terbentuk sebelumnya.