

# 1 Pengenalan Android

---

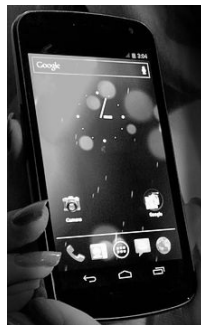
---

**Bab ini membahas:**

- ❖ Perkembangan Android.
  - ❖ Bagaimana Membuat Program di Android?
  - ❖ Konfigurasi Eclipse.
  - ❖ Membuat Program Android Pertama Kali.
  - ❖ Bahasa Pemrograman Android.
- 

**M**embicarakan tentang teknologi mobile sekarang ini tentu amat menyenangkan. Ini karena perkembangan device mobile sangat cepat dan menarik. Salah satunya device Android. Perkembangan Android yang cepat ini selain karena kecanggihannya juga karena Android merupakan sistem operasi mobile yang gratis dan *open source* atau kode sumbernya terbuka.

Android adalah sistem operasi berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Sebagai aplikasi terbuka, Android adalah platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang bisa dipakai oleh bermacam peranti bergerak.



**Gambar 1.1** Sebuah mobile phone dengan sistem operasi Android

Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android lebih lanjut, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache (salah satu jenis lisensi open source), sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

## **Perkembangan Android**

Semua dimulai pada bulan Juli 2000, di mana Google bekerja sama dengan Android Inc., perusahaan internet terkemuka yang berada di Palo Alto, California Amerika Serikat. Kenapa bisa bekerja sama? Itu karena beberapa pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White.



*Gambar 1.2 Logo Android robot seperti manusia*

Ketika itu Android banyak disepelekan sebagai perangkat lunak pada telepon seluler. Karena ketika itu sudah banyak OS untuk mobile, seperti Symbian dan OS komersil milik beberapa pabrikan. Tapi selang beberapa waktu kemudian, Android tahu-tahu sudah berkembang dengan pesat dan menjadi fenomenal.

## Produk Awal Android

Sekitar September 2007 Google mengajukan hak paten aplikasi telepon seluler yang akhirnya menjadi cikal bakal Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada tanggal 5 Januari 2010.

Pada tanggal 9 Desember 2008, diumumkan bahwa ada anggota baru yang bergabung dalam program kerja Android, antara lain ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc.

Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, yaitu Android untuk perangkat bergerak (mobile) yang menggunakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru hingga sekarang ini.

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Lalu pada akhir tahun 2009 diperkirakan paling sedikit terdapat 18 jenis perangkat ponsel yang menggunakan Android.

## Android Versi 1.1

Google merilis Android versi 1.1 pada tanggal 9 Maret 2009. Apa saja fiturnya? Android versi ini dilengkapi dengan tampilan yang lebih cantik pada aplikasi, penambahan fitur pada jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

## Android Versi 1.5 (Cupcake)

Google kembali merilis perangkat smartphone menggunakan Android dan SDK (Software Development Kit) versi 1.5 (Cupcake) pada pertengahan Mei 2009. Ada beberapa pembaruan di sini, di antaranya penambahan beberapa fitur seperti kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa secara langsung dari telepon. Ada juga Bluetooth A2DP yang memungkinkan hubungan secara otomatis ke headset Bluetooth. Ditambahkan juga adanya animasi layar, serta keyboard pada layar yang dapat disesuaikan dengan sistem.



*Gambar 1.3 Android Cupcake*

## Android Versi 1.6 (Donut)

Versi 1.6 ini diberi nama Donut dan dirilis pada bulan September di tahun yang sama dengan versi 1.5. Di sini ada fitur baru yang antara lain proses pencarian yang lebih baik dibanding sebelumnya, adanya indikator baterai indikator dan kontrol applet VPN.

Fitur lainnya adalah galeri yang memungkinkan pengguna memilih foto yang akan dihapus. Fitur kamera, camcorder dan galeri juga diintegrasikan. Dukungan ke standar telekomunikasi yang lain, seperti CDMA/EVDO, 802.1x, VPN, Gesture, dan Text-to-speech engine. Juga ada kemampuan dial kontak; dan teknologi *text to change speech* (tidak tersedia pada semua ponsel), serta dukungan terhadap resolusi VWGA.



*Gambar 1.4 Android Donut*

### Android Versi 2.0/2.1 (Eclair)

Pada tanggal 3 Desember 2009, Open Handset Alliance kembali meluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah optimasi hardware, adanya Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan penuh terhadap standar HTML5, daftar kontak diperbarui serta adanya dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

Google juga mengadakan kompetisi aplikasi mobile terbaik (killer apps - aplikasi unggulan). Kompetisi ini berhadiah \$25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka pada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android di antaranya Angrybirds, Shazam, Backgrounds, dan WeatherBug.

Sistem operasi Android juga mendukung berbagai layanan Internet seperti social network, contohnya MySpace dan Facebook.



*Gambar 1.5 Android Éclair*

## Android Versi 2.2 (Froyo: Frozen Yoghurt)

Pada tanggal 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan dibandingkan versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, memiliki kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi JavaScript engine versi 8 yang dipakai Google Chrome sehingga mempercepat kemampuan rendering browser. Ada juga pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.



*Gambar 1.6 Android Froyo*

## Android Versi 2.3 (Gingerbread)

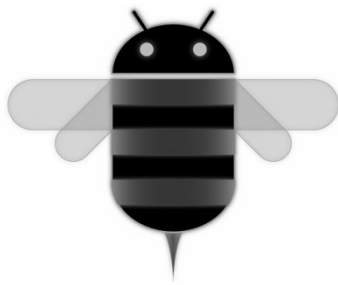
Pada tanggal 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan yang ada di Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antarmuka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, ada juga efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), serta dukungan kemampuan Near Field Communication (NFC) di perangkat ponsel, serta dukungan jumlah kamera yang lebih dari satu.



*Gambar 1.7 Android Gingerbread*

## Android Versi 3.0/3.1 (Honeycomb)

Android Honeycomb dirancang khusus untuk bisa mengakomodasi 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 juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom.



*Gambar 1.8 Android Honeycomb*

## Android Versi 4.0 (ICS: Ice Cream Sandwich)

Versi Android ini diumumkan pada tanggal 19 Oktober 2011, dengan fitur baru termasuk bisa membuka kunci dengan pengenalan wajah, jaringan data kontrol pemantauan penggunaan, integrasi dengan social network, perangkat tambahan fotografi, bisa mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.



*Gambar 1.9 Ikon Ice cream sandwich*

## Arsitektur Android

Android terdiri atas kernel yang berbasis kernel Linux. Di mana di dalamnya ada middleware, library dan API yang ditulis di C dan software aplikasi yang berjalan di framework aplikasi, yang termasuk library java compatible yang berbasis Apache Harmony.

Android memakai Dalvik virtual machine dengan just-in-time compilation untuk menjalankan Dalvik dex-code (Dalvik Executable), yang biasanya diterjemahkan dari Java bytecode. Platform hardware utama untuk Android adalah arsitektur ARM. Ada juga versi x86 dari Android.

Ada beberapa fitur standar dari Android sekarang ini, antara lain:

- Layout handset: Platformnya bisa diterapkan di layar yang lebih lebar, VGA, library 2D yang lebih lengkap. Library 3D berbasis OpenGL. Dan bisa mengakomodasi layout tradisional smartphone.
- Storage: Ada SQLite, database relasional yang cukup ringan dan bisa digunakan untuk penyimpanan data.



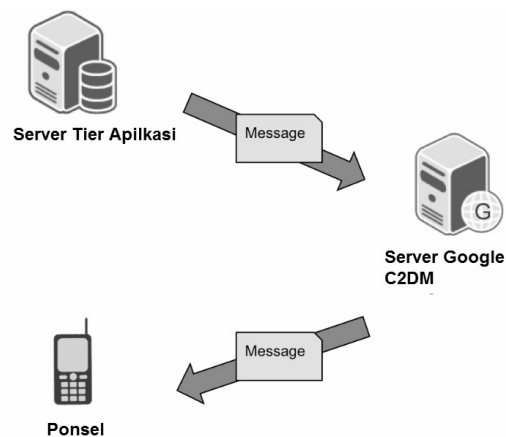
*Gambar 1.10 SQLite sudah ada secara default di Android*

- Konektivitas: Mendukung berbagai teknologi seperti GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC dan WiMAX.
- Java support: walaupun aplikasi Android ditulis di Java, tapi tidak ada Java Virtual Machine (JVM) di platform ini, sehingga java byte code tidak bisa dieksekusi. Class java akan dikompilasi ke dalvik executable dan dijalankan di Dalvik, sebuah virtual machine khusus yang didesain secara spesial untuk Android dan dioptimalkan penggunaan baterainya sehingga cocok dijalankan di perangkat mobile dengan kapasitas memori dan baterai yang ter-



batas. Tapi dukungan J2ME bisa disediakan melalui aplikasi pihak ketiga.

- Messaging: SMS and MMS tersedia. Mendukung juga Android Cloud To Device Messaging (C2DM) yang merupakan bagian dari Android Push Messaging service.



*Gambar 1.11 Dukungan Android Cloud 2 Device*

- Dukungan multi bahasa: Android bisa mendukung multi bahasa.
- Web browser: Browser yang tersedia di Android adalah yang berbasis WebKit Layout engine di mana ada juga versi Chrome V8 JavaScript engine. Di Android 4.0, nilai browser ini sempurna 100/100 dari tes Acid3.



*Gambar 1.12 Berbagai browser untuk sistem operasi Android*

- Dukungan media: Android mendukung banyak format audio/video/gambar diam, seperti: WebM, H.263, H.264 (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4 or 3GP container), MP3, MIDI, Ogg Vorbis, FLAC, WAV, JPEG, PNG, GIF, BMP, WebP.
- Dukungan Streaming media: Mendukung RTP/RTSP streaming (3GPP PSS, ISMA), HTML progressive download (HTML5 <video> tag). Adobe Flash Streaming (RTMP) dan HTTP Dynamic Streaming, Apple HTTP Live Streaming didukung oleh RealPlayer untuk Android, dan sistem operasi Android 3.0 (Honeycomb).
- Dukungan hardware tambahan: Android bisa menggunakan kamera video/gambar, touchscreen, GPS, accelerometer, gyroscope, barometer, magnetometer, kontrol game khusus, dan sensor tekanan. Ada juga termometer, blit 2D bit (dengan orientasi hardware, scaling, pixel dan akselerasi grafis 3D).
- Multi-touch: Android mendukung multitouch yang awalnya dibuat di handset seperti HTC Hero. Fitur ini awalnya dinonaktifkan di level kernel agar tidak melanggar paten apple di teknologi touch screen. Tapi kemudian Google sudah merilis update untuk Nexus one dan Motorola Droid yang memungkinkan multitouch dalam kondisi standar.



*Gambar 1.13 Contoh multitouch di Android*

- Bluetooth: Mendukung A2DP, AVRCP, pengiriman file (OPP), pengaksesan phone book (PBAP), voice dialing dan mengirimkan kontak antara telepon. Dukungan keyboard, mouse dan joystick juga ada di Android 3.1+ dan di versi sebelumnya menggunakan kostumisasi dari pabrikan dan aplikasi pihak ketiga.



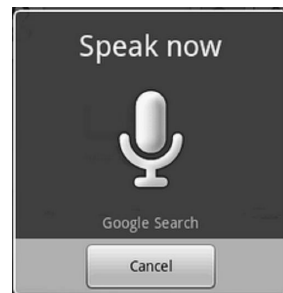
*Gambar 1.14 Bluetooth di Android bisa untuk komunikasi wireless antar-perangkat jarak dekat, contohnya untuk nge-game*

- Video calling: Awalnya Android tidak mendukung video calling. Tapi beberapa handset punya sistem operasi khusus yang mendukungnya. Baik melalui jaringan UMTS (seperti Samsung Galaxy S) atau melalui IP. Video calling melalui Google Talk tersedia di Android 2.3.4 dan seterusnya. Gingerbread memungkinkan Nexus S untuk melakukan internet call melalui akun SIP. Skype 2.1 memungkinkan video calling di Android 2.3 termasuk dukungan untuk kamera depan.



*Gambar 1.15 Video calling di Android*

- Multitasking: Dukungan multitasking tersedia secara default.
- Voice based: Google search bisa melakukan pencarian berdasarkan kata yang Anda ucapkan. Anda juga bisa melakukan telepon, kirim pesan sms, navigasi melalui voice. Ini didukung sejak versi Android 2.2.



*Gambar 1.16 Voice search di Android*

- Tethering: Android mendukung tethering yang memungkinkan telepon digunakan sebagai hotspot Wi-Fi. Sebelum Android 2.2, ini hanya didukung oleh aplikasi pihak ketiga.



*Gambar 1.17 Wifi Tethering Android*

- Screen capture: Android mendukung screenshot dengan klik tombol Power+Volume Down. Tapi untuk Android 4.0

ke bawah, metode untuk screenshoot menggunakan software DDMS dan koneksi ke PC menggunakan DDMS developer tool.

- External storage: Kebanyakan peranti Android berisi micro SD slot. Dan bisa membaca kartu micro SD yang diformat dengan FAT 32, Ext3FS dan Ext4FS. Kalau mau menggunakan flash disk USB, beberapa tablet Android juga berisi port USB A.



*Gambar 1.18 Pembaca storage eksternal di Android*

## **Bagaimana Membuat Program di Android?**

Ketika Android market (sekarang Google Play) di-launching, maka hampir 9 ribu aplikasi per bulan ada di Android Market. Dan ini semakin meningkat dengan cepat. Tak heran bahwa jumlah aplikasi Android yang bertambah pesat, dan juga perangkat Android yang keluar maka pemrograman di Android akan makin dibutuhkan dan digemari.

Tentu Anda ingin mengetahui bagaimana menulis aplikasi di Android bukan? Aplikasi Android sebenarnya ditulis menggunakan bahasa Java. Di mana bahasa java adalah bahasa yang sangat mudah dipelajari, user friendly dan gampang dimengerti oleh para developer pemula.

Sebelum bisa menulis aplikasi Android, Anda perlu terlebih dahulu mengetahui bagaimana cara kerja pemrograman di Android. Pertama sama seperti bahasa pemrograman lainnya, program di Android juga dikembangkan di komputer PC Windows, Mac atau Linux. Di Indonesia, yang paling lazim digunakan adalah di Windows.

Setelah dikembangkan di PC, kemudian dikompilasi dan dikirim ke device untuk dicoba. Apakah berarti Anda harus punya perangkat Android dulu sebelum bisa memprogram? Tentu saja tidak, karena tersedia emulator yang bisa mensimulasikan peranti Android di komputer, artinya Anda tetap bisa mengembangkan aplikasi Android, atau game Android tanpa memiliki perangkat Android fisik.



*Gambar 1.19 Ikon Eclipse*

Pertama untuk proses development, Anda membutuhkan Eclipse. Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak yang dapat dijalankan di semua platform (*platform-independent*). Berikut ini sifat Eclipse:

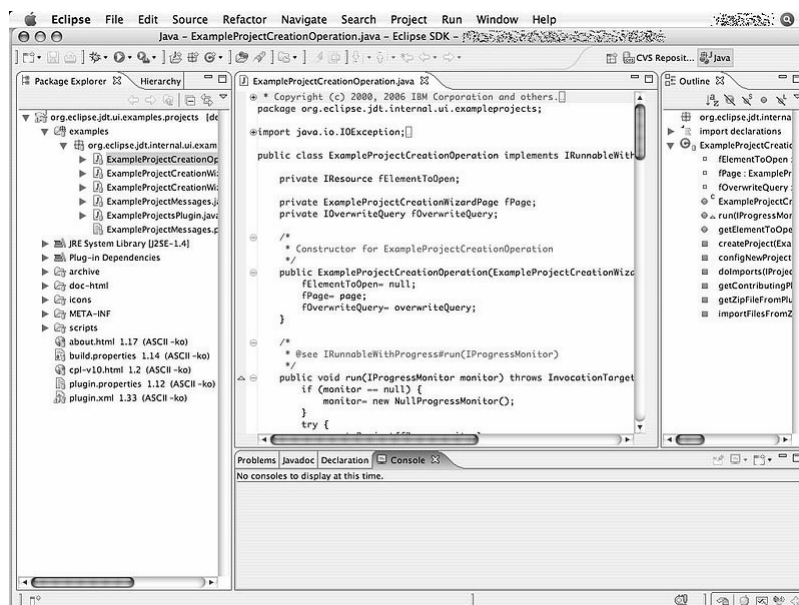
- **Multi-platform:** Eclipse dapat dijalankan di Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- **Mult-language:** Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
- **Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi, Eclipse bisa juga dipakai untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit karena sifatnya yang gratis (*free*) dan *open source*, artinya setiap orang

boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan lebih lanjut oleh pengguna dengan komponen yang dinamakan *plug-in*.

Untuk tutorial di dalam buku ini akan digunakan Eclipse. Kenapa? Karena Eclipse adalah yang paling mudah digunakan. Untuk memperoleh Eclipse Anda bisa men-download-nya dari <http://www.eclipse.org/downloads/>.

File yang Anda download biasanya berupa file .zip sehingga Anda perlu melakukan *unzip*. Setelah itu barulah Anda bisa menggunakannya secara langsung. Sebagai catatan Anda bisa men-download Eclipse untuk sistem operasi Anda, kalau pakai Linux, Anda tinggal download versi Linux, kalau pakai Mac tinggal download versi Mac, kalau pakai Solaris, tinggal download versi Solaris.



Gambar 1.20 Contoh User interface dari Eclipse

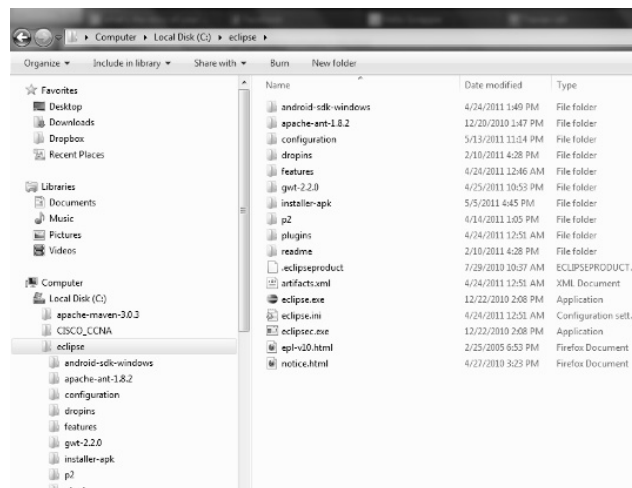
Penjelasan lengkap berkaitan dengan download dan proses awal penggunaan Eclipse bisa Anda lihat pada langkah berikut:

1. Download file **Eclipse Classic** (ukuran kurang lebih 171 MB) dari alamat <http://www.eclipse.org/downloads/>.
2. File hasil download berbentuk .zip. Lakukan ekstrak file tersebut menggunakan tool seperti WinZip atau WinRar.



*Gambar 1.21 Download file Eclipse*

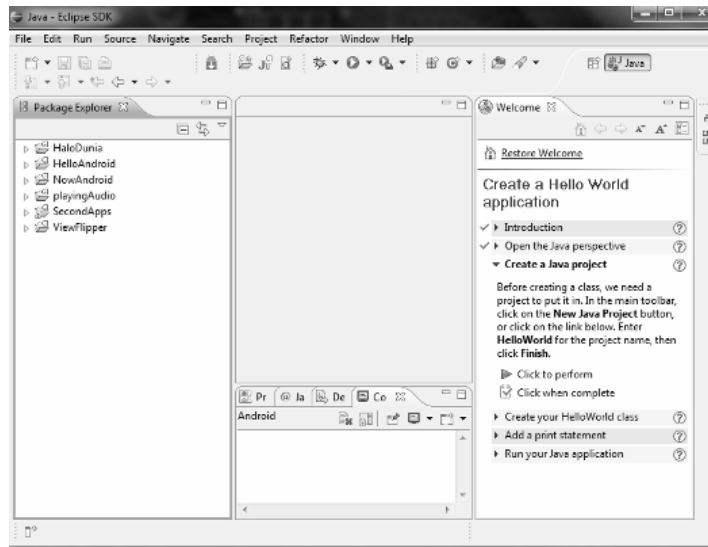
3. Setelah diekstrak, akan dihasilkan folder hasil ekstraksi. Selanjutnya pindahkan folder tersebut di lokasi yang sekiranya mudah Anda akses.



*Gambar 1.22 Hasil ekstraksi Eclipse*



4. Eclipse tidak memerlukan wizard instalasi. Dalam folder akan ditemukan file **eclipse.exe**. Untuk membuka program Eclipse tersebut cukup lakukan klik ganda file **eclipse.exe** sehingga terbuka jendela berikut.



*Gambar 1.23 Tampilan halaman awal eclipse di Windows*

Setelah Eclipse sudah siap, tahap kedua adalah dengan men-download Java JDK. Anda bisa memperolehnya dari alamat <http://java.sun.com/javase/downloads/>. Sama seperti Eclipse, download-lah JDK sesuai sistem operasi Anda.



*Gambar 1.24 Ikon Java*

Anda bisa men-download versi JDK 5, 6, atau 7. Anda tidak perlu memakai versi terakhir, karena Anda bisa memakai dari versi sebelumnya. Anda tinggal menginstalnya dengan men-download dan kemudian menjalankan **Setup** untuk menjalankan program ini.

Java Development Kit (JDK) adalah sebuah produk yang dikembangkan oleh Oracle yang ditujukan untuk para developer java. Sejak Java diperkenalkan, JDK merupakan Java Software Development Kit (SDK) yang paling sering digunakan. Pada tanggal 17 November 2006, Sun mengatakan bahwa akan merilis JDK sebagai *open source* menggunakan lisensi GNU General Public License (GPL), artinya software ini sekarang menjadi free software. Pada tanggal 8 May 2007 Sun mengontribusikan source code-nya ke OpenJDK.

JDK dilengkapi dengan banyak komponen untuk melakukan pemrograman, di antaranya komponen:

- Java: adalah loader untuk aplikasi Java. Tool ini adalah interpreter yang bisa menerjemahkan file class yang di-generate oleh javac compiler. Sekarang sudah menjadi launcher tunggal yang bisa dipakai untuk development dan deployment. Tool deployment lama JRE sudah tidak digabungkan di JDK ini, tapi sudah diganti dengan java loader yang baru.
- Javac: merupakan compiler yang mengonversi source code java ke java bytecode.
- Appletviewer: merupakan tool yang bisa digunakan untuk menjalankan dan debug applet java tanpa web browser.
- Apt: merupakan tool pemrosesan annotation.
- Extcheck: merupakan utility yang bisa mendeteksi konflik di file JAR.
- Idlj: merupakan compiler IDL-ke-Java. Utility ini akan men-generate java binding dari file IDL.
- Javadoc: merupakan generator dokumentasi yang akan menghasilkan dokumentasi secara otomatis dari komentar di source code.

- Jar: merupakan archiver yang akan memaketkan class library ke file JAR tunggal. Tool ini bisa digunakan untuk mengatur file JAR.
- Javah: merupakan C header dan stub generator, digunakan untuk menuliskan metode native.
- Javap: merupakan disassembler file class.
- Javaws: merupakan java Web Start yang berfungsi sebagai launcher untuk aplikasi JNLP.
- Jconsole: merupakan jJva Monitoring and Management Console.
- Jdb: digunakan sebagai debugger.
- Jhat: merupakan Java Heap Analysis Tool (eksperimental).
- Jinfo: merupakan utility untuk memperoleh informasi konfigurasi dari proses java yang running atau crash dump (eksperimental).
- Jmap: utility ini akan memberikan output berupa peta memori dari java dan bisa mencetak dari proses tertentu atau core dump (eksperimental).
- Jps: adalah Java Virtual Machine Process Status Tool yang bisa menggunakan HotSpot Java Virtual Machines (JVM) di sistem target.
- Jrunscript: merupakan Shell Java command-line.
- Jstack: merupakan utility yang akan mencetak stack trace dari thread java.
- Jstat: merupakan Java Virtual Machine statistics monitoring tool.
- Jstatd: merupakan jstat daemon (eksperimental).
- Policytool: merupakan tool policy creation dan management, yang bisa menentukan policy yang dipakai oleh Java runtime.
- VisualVM: merupakan tool visual untuk mengintegrasikan beberapa tool JDK command line.

- Wsimport: berfungsi men-generate artifak JAX-WS portabel.
- Xjc: merupakan bagian dari Java API untuk XML Binding (JAXB). Ini bisa digunakan untuk menerima skema XML dan men-generate class Java.

JDK juga berisi paket Java Runtime Environment yang komplet, biasanya disebut private run time karena biasanya dipisah dari JRE reguler dan dilengkapi dengan konten tambahan. Yaitu terdiri atas Java Virtual Machine dan semua class library yang ada di environment produk. Dan juga library tambahan yang berguna untuk developer. Misalnya library internationalization dan library IDL.

Salinan dari JDK juga berisi beberapa contoh program yang menjelaskan contoh penerapan porsi dari API Java. Form JDK merupakan bagian dari software development kit (SDK). Untuk Java SE, EE, dan ME, Sun mengatakan bahwa JDK merupakan bagian dari SDK yang memiliki tanggung jawab untuk menuliskan dan menggabungkan program java yang sedang berjalan. Bagian SDK lainnya antara lain application server, debugger, dan dokumentasi.

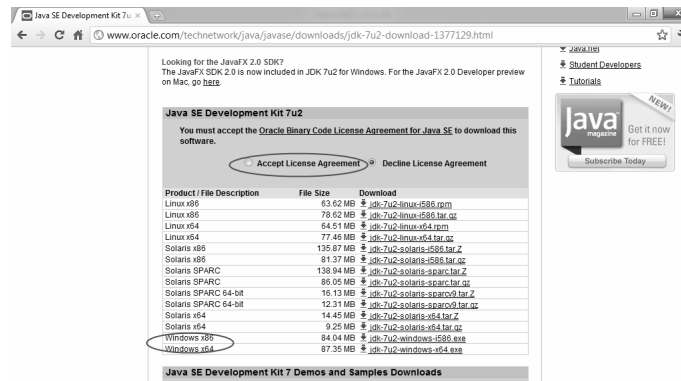
Penjelasan lengkap mengenai download dan instalasi JDK bisa Anda lihat pada langkah berikut:

1. Buka situs download Oracle untuk men-download java JDK dengan URL <http://www.oracle.com/technetwork/java/javase/downloads/>. Anda bisa memilih versi JDK yang Anda inginkan.



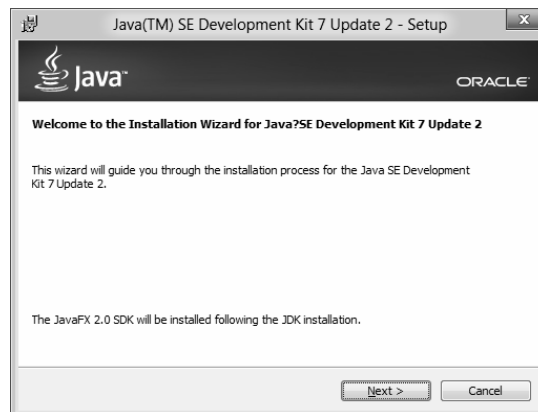
Gambar 1.25 Download versi JDK

- 
2. Klik **Accept License Agreement**, pilih download **Windows x86** atau **Windows x64**.



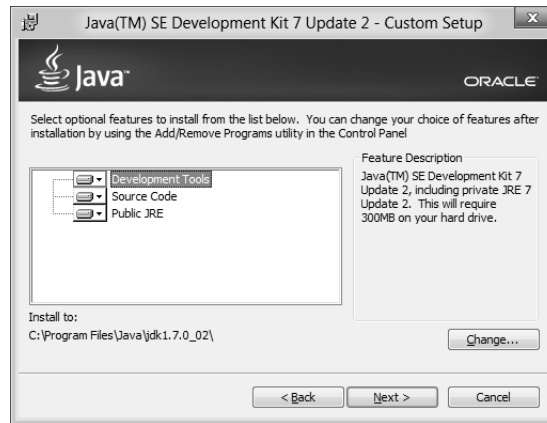
*Gambar 1.26 Men-download versi JDK*

- 
- 
3. Tunggu hingga proses download selesai.
4. Eksekusi file yang sudah Anda download. Pertama akan muncul kotak dialog **Welcome to the Installation Wizard**. Klik **Next**.



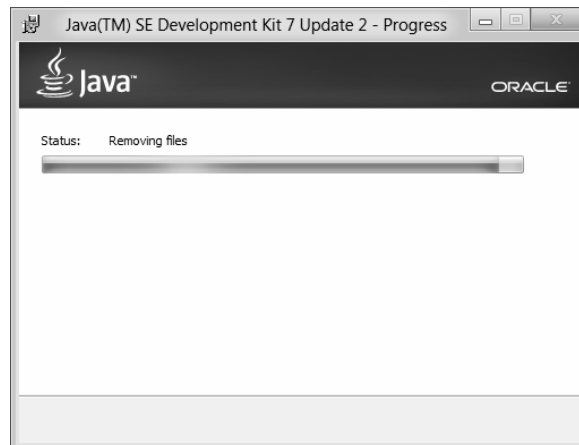
*Gambar 1.27 Welcome to the Installation Wizard*

- 
- 
- 
- 
5. Proses berikutnya Anda bisa memilih paket yang akan diinstal, namun bisa pula Anda biarkan sesuai kondisi default saja. Klik **Next**.



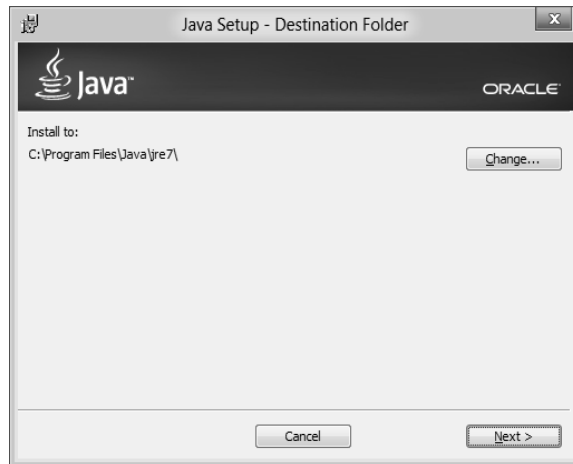
*Gambar 1.28 Pemilihan paket yang akan diinstal*

6. Instalasi akan dilanjutkan. Pada progress bar bisa Anda lihat detail kemajuan instalasi.



*Gambar 1.29 Proses instalasi berlangsung*

7. Pada proses penentuan folder instalasi, Anda dapat langsung klik **Next** sehingga folder instalasi tetap default.



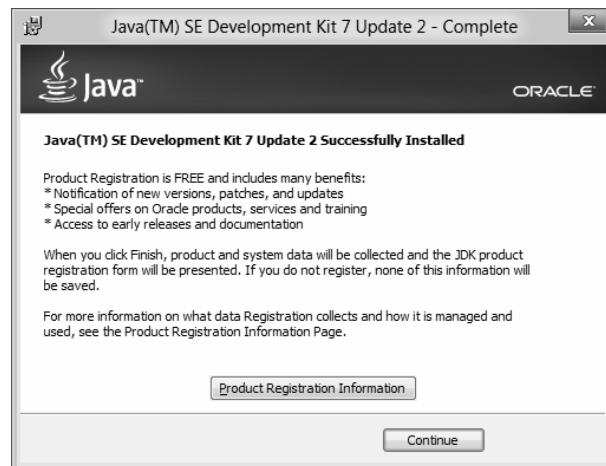
*Gambar 1.30 Penentuan destination folder*

8. Tunggu hingga status instalasi berlangsung.



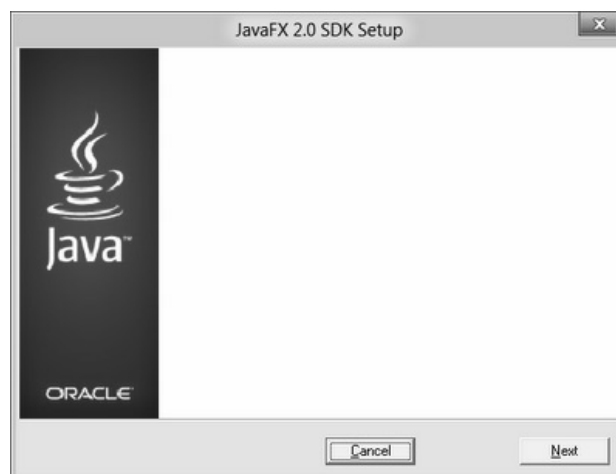
*Gambar 1.31 Instalasi sedang berlangsung*

9. Klik tombol **Continue** untuk melanjutkan proses instalasi JavaFx 2.0 SDK.



*Gambar 1.32 Instalasi JavaFX*

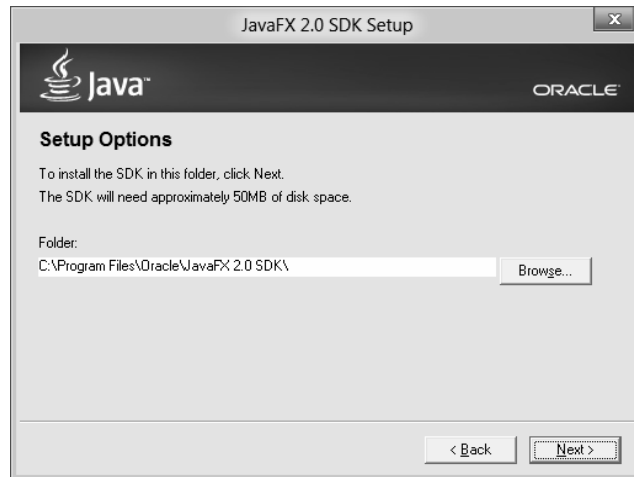
10. Klik **Next** ketika instalasi **JavaFx 2.0 SDK Setup** telah siap.



*Gambar 1.33 Instalasi JavaFX SDK Setup*

11. Langsung saja klik **Next** tanpa perlu Anda ubah direktori instalasi.





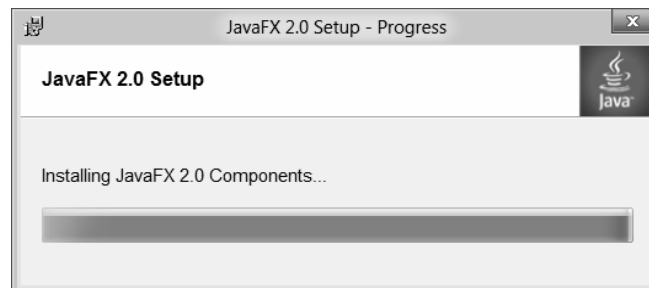
*Gambar 1.34 Setup Options untuk mengatur opsi*

12. Proses instalasi **JavaFX SDK** berlangsung.



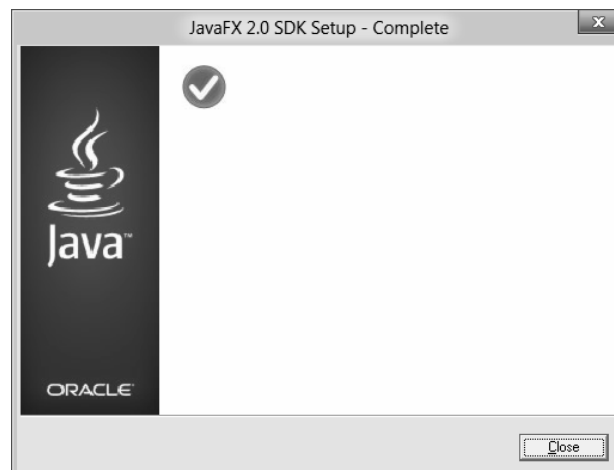
*Gambar 1.35 Proses instalasi berlangsung*

13. Dilanjutkan dengan instalasi komponen **JavaFX**.



*Gambar 1.36 Instalasi komponen JavaFX Setup*

14. Akhiri instalasi dengan klik tombol **Close**.



*Gambar 1.37 Akhir instalasi Java SDK*

15. Selanjutnya pada browser akan dimunculkan **Java Development Kit (JDK) 7 Registration** untuk me-register software Anda.



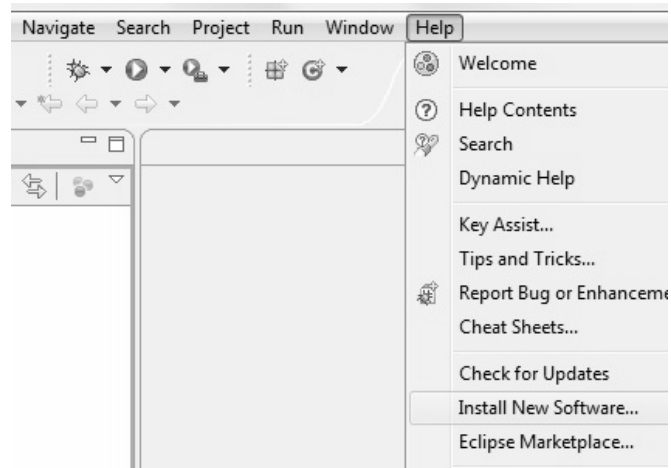
**Gambar 1.38** Browser muncul untuk me-register

16. Terakhir lakukan set path agar Anda bisa menjalankan javac. Jika belum Anda set, maka javac hanya bisa dijalankan dengan mengakses folder `C:\Program Files\Java\jdk1.7.0_02\bin`.

Tahap ketiga setelah Anda menyiapkan Eclipse dan menginstal java JDK adalah dengan men-download tool Android SDK. Situsnya ada di URL <http://developer.android.com/sdk/index.html>. Anda perlu mengakses Android SDK Tool langsung dari Google Unpack. URL ini perlu Anda simpan untuk mengonfigurasi tahap keempat.

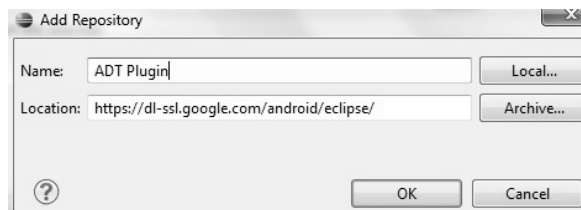
Tahap keempat adalah mengonfigurasi Eclipse untuk Android. Untuk melakukannya ikuti langkah-langkah berikut ini:

1. Setelah Anda download **Android SDK** simpan di hard disk, misalnya "`C:\Android`". Jika Anda sudah men-download-nya, tinggal ekstrak ke lokasi tertentu.
2. Jalankan Eclipse, kemudian klik **Help > Install New Software**. Ini untuk menjalankan SDK di Eclipse.



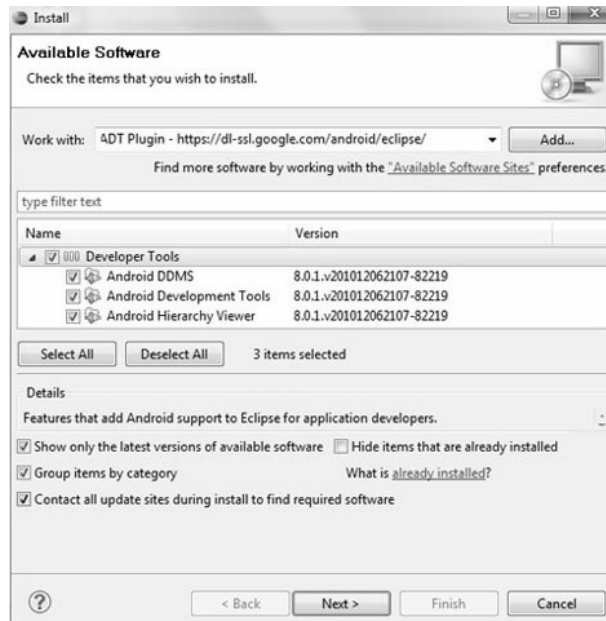
*Gambar 1.39 Menu Help > Install New Software*

3. Pada kotak dialog **Add Repository**, tambahkan/ketikkan nama “ADT Plug-in” pada kolom **Name** dan ketikkan “<https://dl-ssl.google.com/android/eclipse/>” di kolom **Location**. Klik **OK** kemudian.



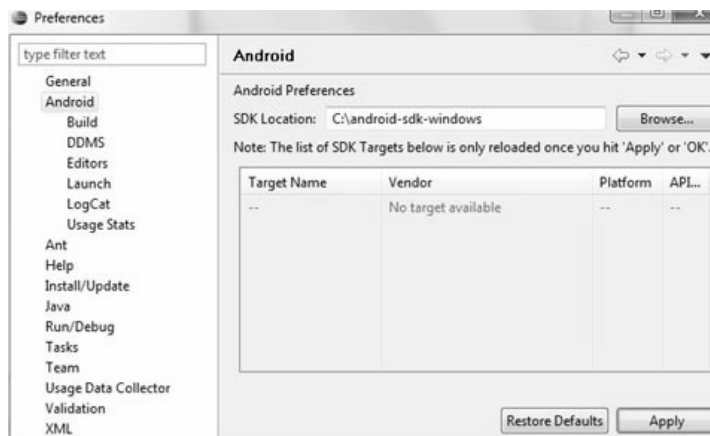
*Gambar 1.40 Penentuan Lokasi plugin ADT*

4. Pada kotak dialog berikutnya, ceklis pada **Developer Tools** kemudian klik tombol **Next**. Pada kotak dialog selanjutnya, Anda akan memperoleh beberapa list dari paket dan tool yang akan di-download. Klik **Next** dan klik tombol **Finish** setelah membaca **License agreement**. Klik tombol **Close** dan restart komputer.



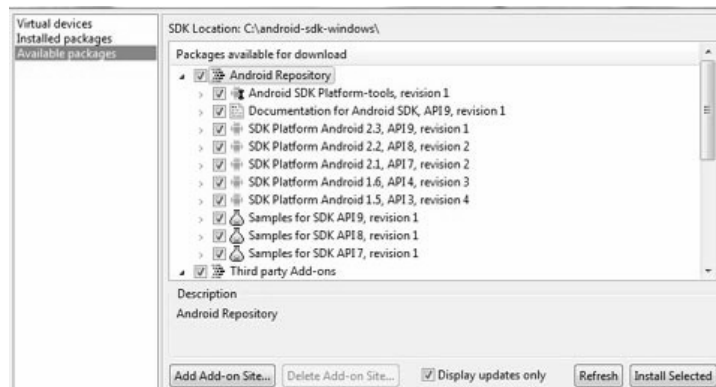
Gambar 1.41 Pengaturan Available software

5. Buka menu **Window** kemudian pilih opsi **Preferences**. Pilih nama "Android" dari direktori dan klik **OK**.



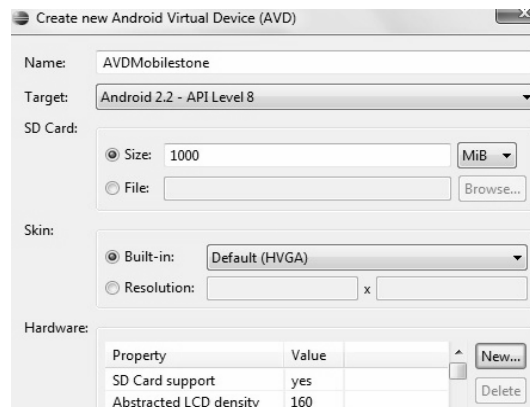
Gambar 1.42 Pemilihan preferensi Android

- Pilih virtual device atau tambahkan AVD (Android Virtual Device) yang baru, Anda bisa mengetes program tanpa harus menggunakan perangkat Android asli.



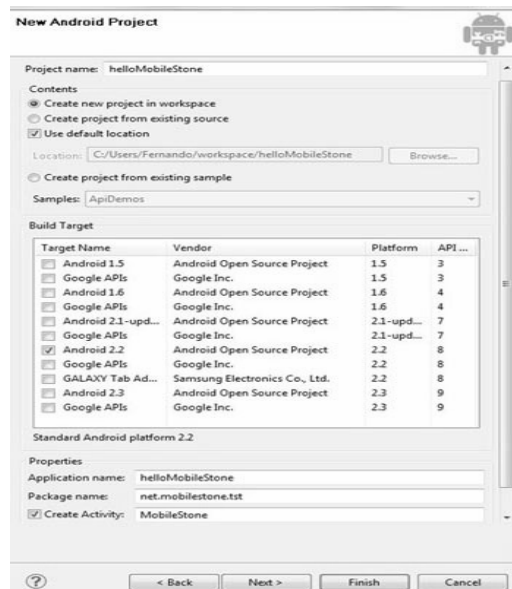
*Gambar 1.43 Android repository*

- Tentukan nama, target dan lain sebagainya dari AVD Anda.



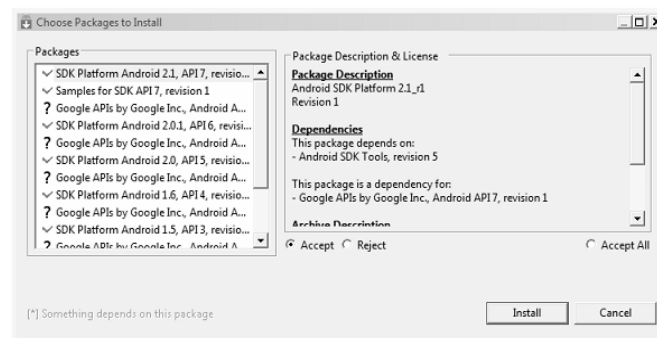
*Gambar 1.44 Pembuatan AVD baru*

- Kemudian klik **File** > **New** dan pilih **Project**. Anda bisa memilih opsi Android. Klik **Finish** maka Anda siap untuk membuat proyek baru.



Gambar 1.45 Pembuatan proyek Android

10. Konfigurasi Android SDK dengan memilih paket yang akan diinstal.

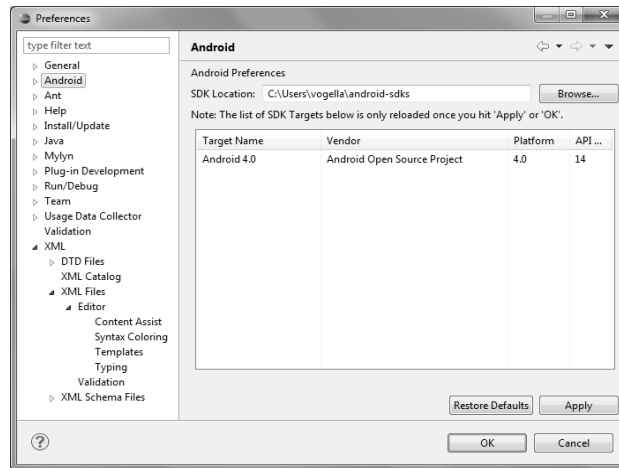


Gambar 1.46 Pemilihan paket yang akan diinstal

11. Buka folder tempat Anda mengekstrak Android sdk, kemudian eksekusi file `sdk setup.exe`.

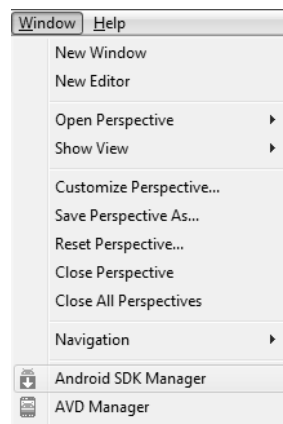
Bisa pula Anda setting manual Android di Eclipse, caranya seperti berikut:

1. Klik **Windows > Preferences**. Pilih **Android** dan masukkan path instalasi dari Android SDK.



*Gambar 1.47 Jendela Preferences > Android*

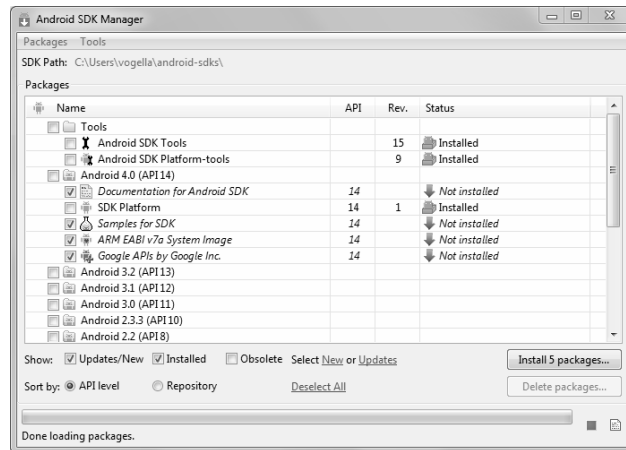
2. Untuk menginstal versi Android tertentu, Anda bisa memasukkan versi tertentu melalui menu **Windows > Android SDK Manager** dari menu Eclipse.



*Gambar 1.48 Android SDK Manager*



3. Muncul kotak dialog **Android SDK Manager**. Pilih **Available Packages**, dan buka **Third party add ons**. Pilih **Google API** sesuai dengan versi Anda. Kemudian klik **Install**.

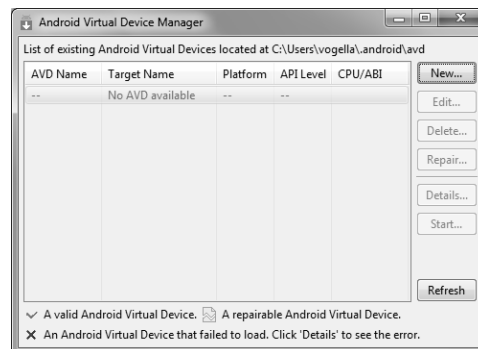


Gambar 1.49 Instalasi versi Android tertentu

## Konfigurasi Eclipse Pertama Kali

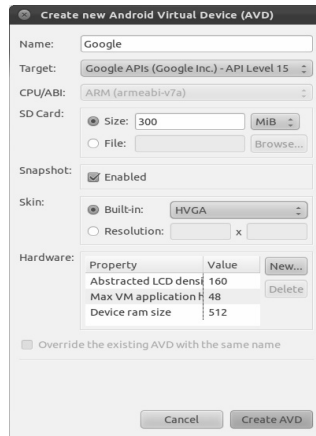
Berikut ini merupakan cara untuk mengonfigurasi Eclipse pertama kali sebelum digunakan untuk menulis program pertama kali di Android:

1. Di **SDK Manager**, klik **Virtual Devices**. Anda perlu membuat peranti Android virtual untuk mengetes.



Gambar 1.50 Jendela untuk buat AVD baru

2. Klik **Create New AVD**.



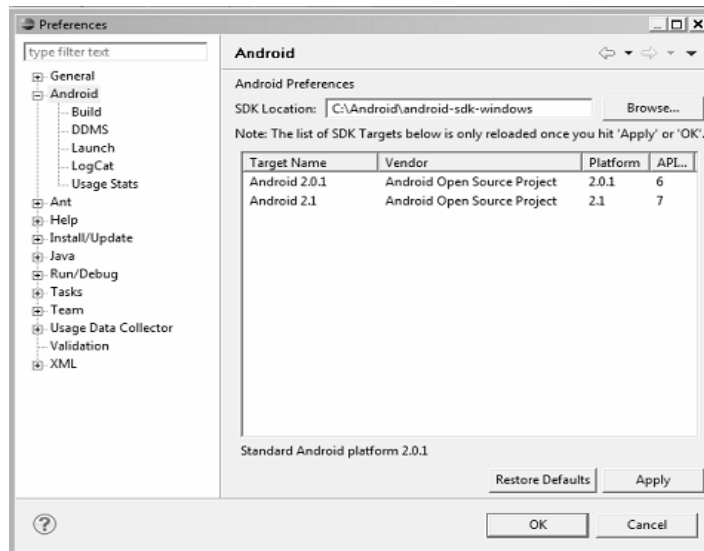
*Gambar 1.51 Pembuatan AVD baru*

3. Setelah itu, coba jalankan AVD Anda.



*Gambar 1.52 Menjalankan AVD Anda*

4. Konfigurasi Eclipse kembali, kemudian klik **Preferences** → **Android**. Pastikan ada **Target Name** di situ.

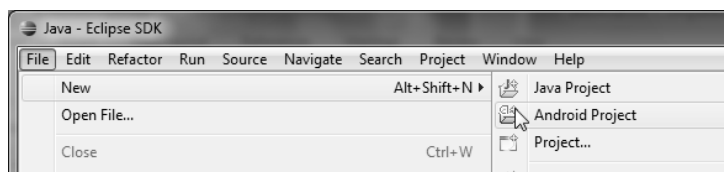


*Gambar 1.53 Target name dari AVD sudah ada di Eclipse*

## Membuat Program Pertama Kali

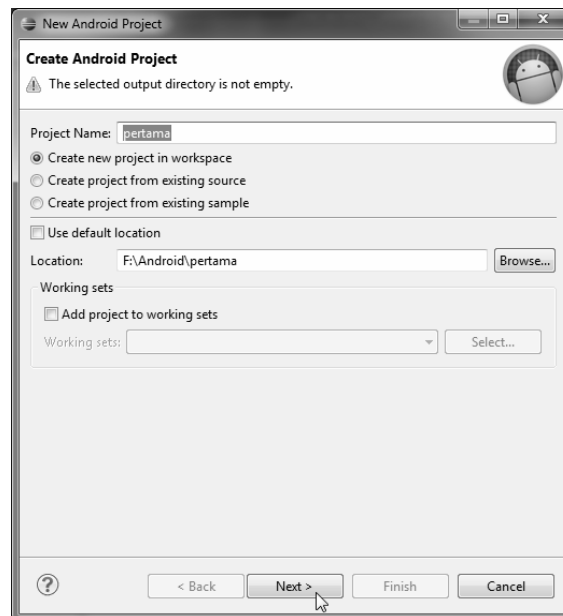
Ketika hendak membuat program, Anda perlu membuat proyek terlebih dahulu. Berikut ini ilustrasi pembuatan proram di Android saat pertama kali:

1. Klik menu **File > New > Android Project** seperti berikut:



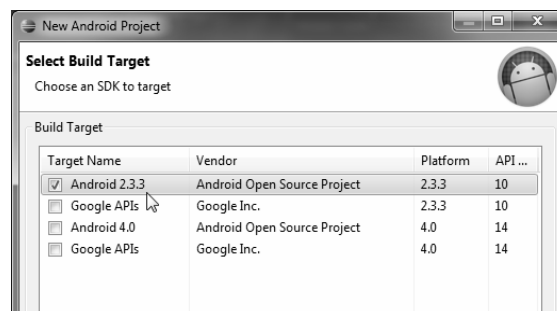
*Gambar 1.54 File > New > Android Project*

2. Muncul kotak dialog **Create Android Project** yang meminta Anda mendeskripsikan project yang hendak dibuat. Klik **Next** kemudian.



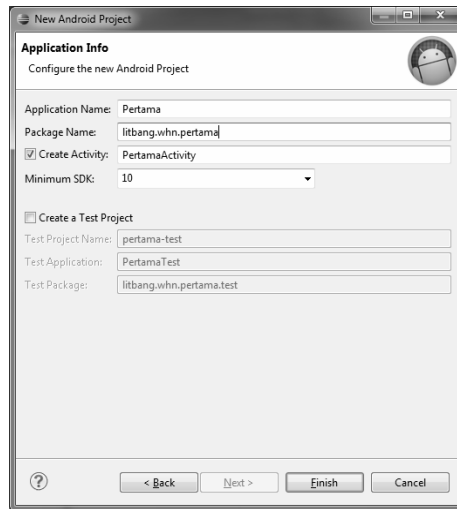
*Gambar 1.55 Pembuatan identitas proyek yang hendak dibuat*

3. Selanjutnya tentukan **Target Name** Android yang diinginkan. Klik **Next** kemudian.



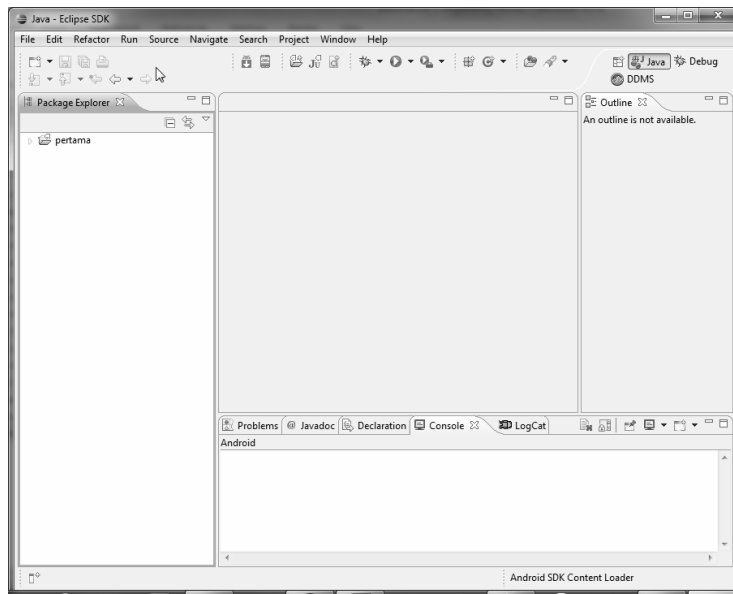
*Gambar 1.56 Target Android yang diinginkan*

4. Kemudian di **Application Info**, masukkan info aplikasi yang dibuat.



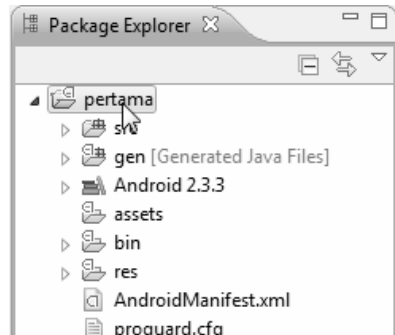
*Gambar 1.57 Pengisian package name*

5. Setelah Anda klik **Finish**, akan dimunculkan jendela **Java - Eclipse SDK** seperti berikut ini.



*Gambar 1.58 Tampilan Eclipse pertama kali*

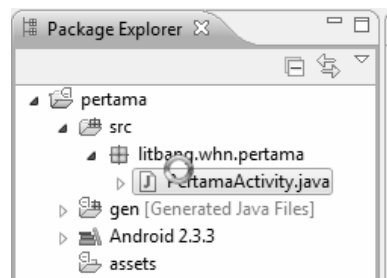
6. Pada **Package Explorer** terlihat file-file proyek yang dibuat.



*Gambar 1.59 File-file di package explorer*

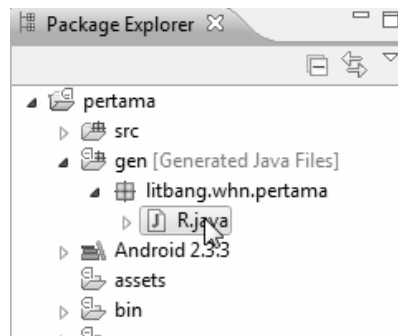
7. Bisa Anda lihat pada gambar di atas, ada contoh struktur program Android. Penjelasan bisa Anda lihat pada daftar berikut:

- Folder `/src` berisi file source java yang akan Anda buat. Di sini ada file **Activity** yang dibuat untuk proyek. File di folder ini akan diatur sesuai dengan struktur paket. Ini mirip dengan folder `/src` yang ada di project java biasa.



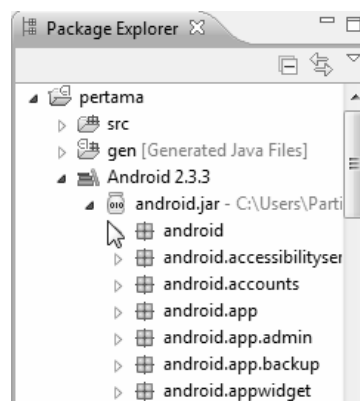
*Gambar 1.60 Isi folder src/*

- Folder `/gen` juga merupakan folder source, ini akan menampung file source java yang akan otomatis di-generate oleh platform Android. Di antara file java yang di-generate otomatis adalah file R class, yang bisa Anda lihat di *screenshot*. Framework akan men-generate file R class dan Anda bisa membaca ini lebih lanjut di dokumentasi Android.



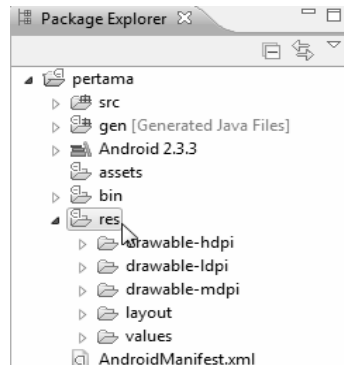
**Gambar 1.61** Isi folder /gen

- Folder **/Android** {nomor versi}: berisi library (jar) yang dibutuhkan untuk project. Anda bisa melihat bahwa ini berisi file jar dari framework. Ini mirip dengan folder **/lib** yang ada di proyek java biasa.



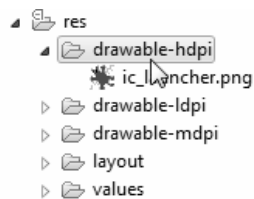
**Gambar 1.62** Folder /Android+versi

- **/res**: direktori ini berisi semua resource eksternal (image, data, file, dan seterusnya) yang digunakan oleh aplikasi Android. Resource eksternal ini akan direferensikan di aplikasi Android. Di dalam /res ini ada beberapa aplikasi. Seperti: /res/drawable, /res/layout dan /res/Values.



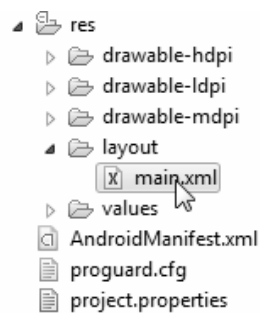
**Gambar 1.63** Isi folder /res

- /res/drawable: berisi gambar dan sebagainya. Jika Anda ingin memasukkan image atau ikon ke aplikasi Android, Anda bisa meletakkannya di folder ini.



**Gambar 1.64** Folder /res/drawable

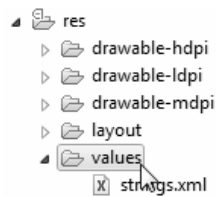
- /res/layout: berisi layout user interface yang akan digunakan di proyek. Layout UI ini akan disimpan dalam file XML. Anda bisa membacanya di dokumentasi Android.



**Gambar 1.65** /res/layout

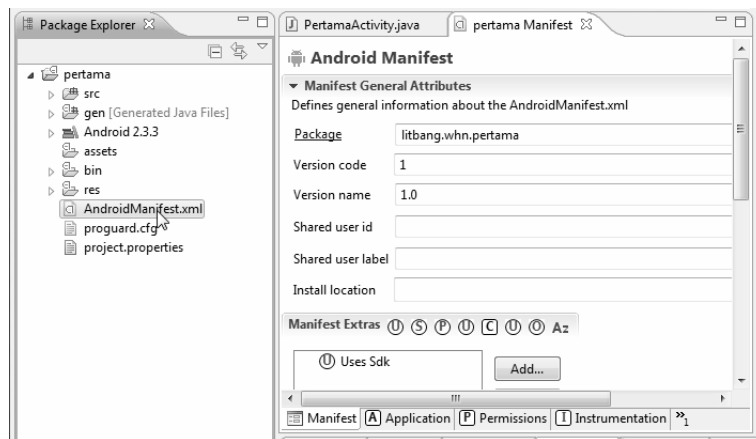


- **/res/Values:** berisi file XML yang berisi value berpasangan yang akan direferensikan di aplikasi. File XML ini mendeklarasikan array, color, dimensi, string dan sebagainya. Salah satu keunggulan pemisahan nilai value dalam file xml tersendiri adalah aplikasi bisa dikostumisasi tanpa harus membongkar kode utamanya.



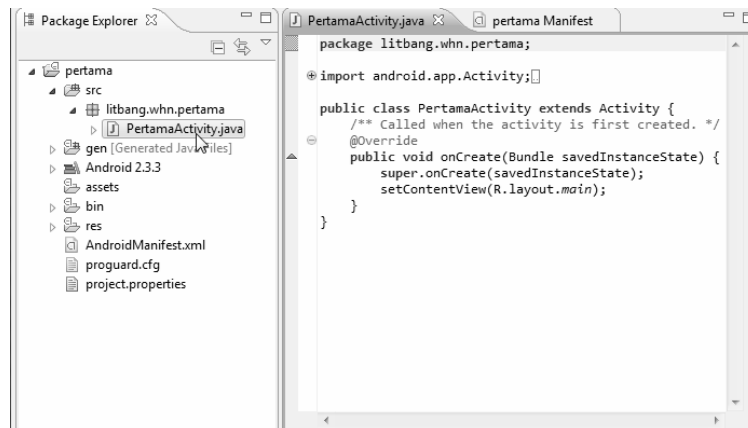
Gambar 1.66 /res/values

- **/assets:** berisi resource eksternal yang digunakan dalam aplikasi. Sama seperti /res, perbedaannya adalah kalau resource disimpan dalam format raw dan bisa dibaca secara programatik saja.
- **AndroidManifest.xml:** ini adalah file xml yang berisi informasi meta mengenai aplikasi Android dan merupakan file yang penting untuk tiap proyek aplikasi Android. Ini berisi informasi mengenai berbagai aktivitas, view, service dan sebagainya. Ini juga berisi list user permission yang diperlukan untuk menjalankan aplikasi Android.



Gambar 1.67 Folder AndroidManifest.xml

8. Selanjutnya Anda buka file **Java** di **/src/nama\_paket**.



*Gambar 1.68 Mengedit file java di /sr/nama\_paket*

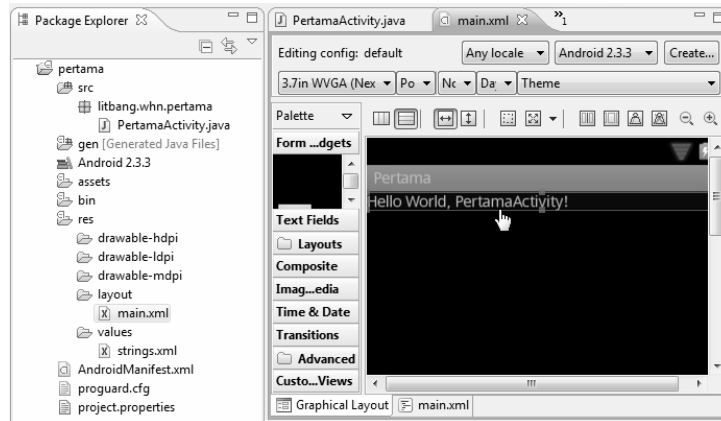
9. Akan diperlihatkan kode seperti berikut ini:

```
package litbang.whn.pertama;
import android.app.Activity;
import android.os.Bundle;

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

10. Kode di atas disertakan pada saat objek diciptakan (onCreate), dan ditampilkan layout di R.layout.main. Karena itu perlu diedit isi dari R.layout.main yang menjelaskan layout dari jendela utama yang akan ditampilkan.

11. Layout standar terlihat seperti berikut.



*Gambar 1.69 File layout standar main.xml di /res/layout/main.xml*

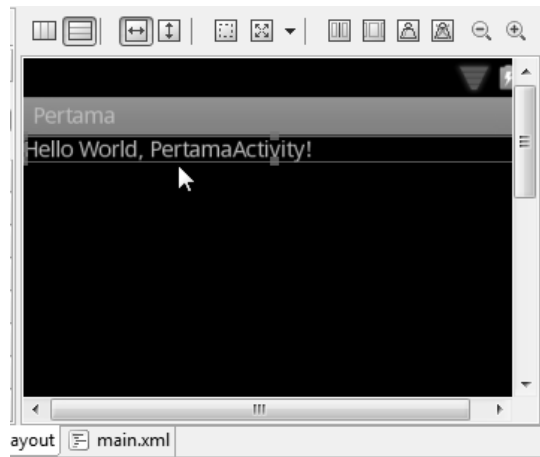
12. Kalau dilihat kodenya, terlihat seperti di bawah (untuk melihat kodenya, klik tab Main.xml).

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

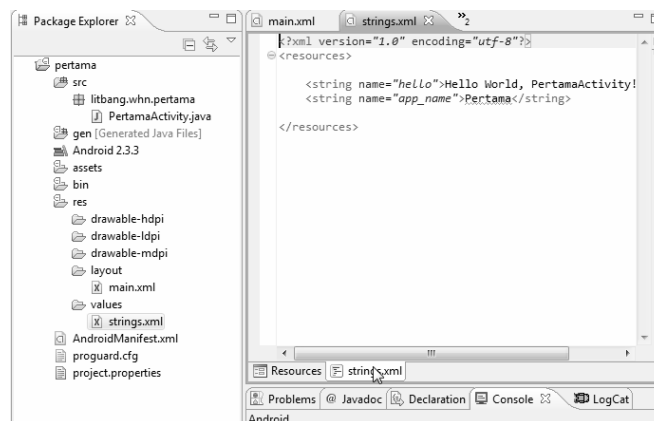
13. Kalau Anda lihat di kode di atas, hanya ada satu widget, yaitu *TextView* yang dibuat menggunakan kode xml `<TextView />`.

14. Widget `<TextView/>` tersebut adalah widget yang menampilkan teks. Jika ditampilkan dalam grafik, widget textview adalah widget yang diklik, dengan teks “Hello World, PertamaActivity”.



*Gambar 1.70 HelloWorld, PertamaActivity*

15. Teks yang ada di TextWidget tersebut didapat dari file `@string/hello`.
16. Nilai ini diperoleh dari `res/values/strings.xml`.



*Gambar 1.71 Nilai di /res/values/strings.xml*

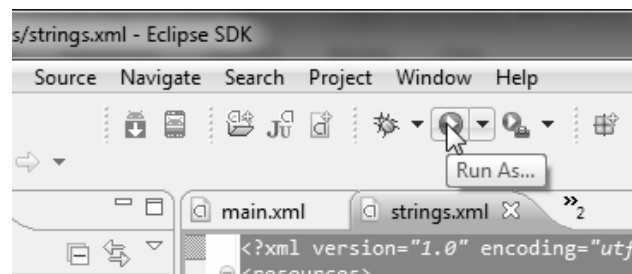
17. Kode dari /res/values/string.xml yaitu sebagai berikut.

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

18. Kalau mau mengganti teks, tinggal ganti di <string name="hello"> </string>. Misalnya menjadi seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Android Programming
  Menyenangkan!</string>
  <string name="app_name">Aplikasi Pertama Gw</string>
</resources>
```

19. Sekarang jalankan program dengan menekan tombol **F11** pada keyboard atau klik tombol panah pada **Run As**.



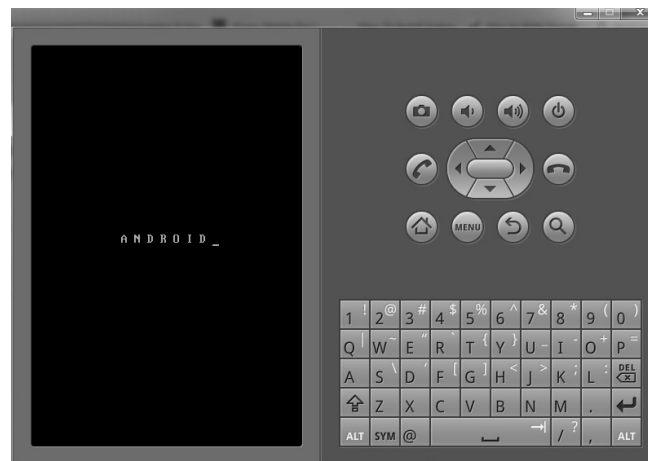
*Gambar 1.72 Klik tombol Run As*

20. Muncul kotak dialog **Run As**, pilih **Android Application** dan klik **OK**.



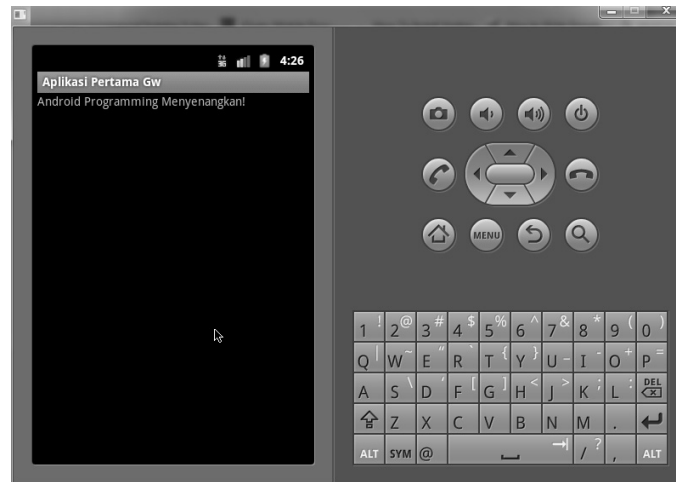
*Gambar 1.73 Run As Android Application*

21. Muncul AVD Anda seperti berikut.



*Gambar 1.74 AVD sedang loading*

22. Tunggu hingga AVD muncul dan Anda bisa melihat program Anda di-upload ke peranti emulator Android ini.



Gambar 1.75 Program pertama sudah dibuat

## Bahasa Pemrograman Android

Aplikasi Android ditulis di Java. Tidak bahasa java sepenuhnya seperti J2EE, tapi bagian dari java tanpa melibatkan class-class yang tidak dipakai di mobile device. Jika Anda sudah berpengalaman di pemrograman java, tentunya Anda akan mudah mempelajari java.

Usahakan memiliki buku panduan pemrograman java, tapi kalau tidak, Anda tetap bisa browsing dengan mudah menggunakan *search engine*. Di source code java, tidak semua library dimasukkan, karena itu Anda perlu memeriksa apakah paket ada di tempat Anda.

### Activity

Aplikasi Android dibuat dari satu atau lebih activity. Aplikasi Anda paling tidak memiliki satu activity. Sebuah aplikasi Android bisa memiliki lebih dari satu activity.

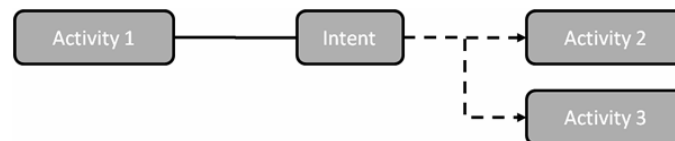
Activity ini bisa diibaratkan seperti wadah untuk UI (*user interface*) Anda. Activity ini akan menampilkan UI dan kode untuk menjalankannya. Jadi, mirip dengan form dalam pemrograman Windows.

## Intent

Intent merupakan sistem pesan untuk core yang menjalankan Android. Intent terdiri atas action yang perlu dilakukan di Android (misalnya view, edit, dial, dan sebagainya) dan data. Action yang dimaksud adalah action umum yang akan dilakukan ketika intent diterima. Dan data adalah data yang akan dioperasikan. Data ini bisa berupa contact item. Intent digunakan untuk memulai activity dan berkomunikasi antara beberapa bagian dari sistem Android. Aplikasi Anda bisa mem-broadcast dan menerima intent.

### Mengirim Pesan Menggunakan Intent

Pada saat Anda mem-broadcast intent, Anda mengirimkan pesan yang akan memberitahukan Android untuk melakukan sesuatu. Intent ini bisa memberi tahu Android untuk memulai activity dari dalam aplikasi, atau memulai aplikasi yang lain.



*Gambar 1.76 Sebuah aplikasi bisa memiliki lebih dari satu activity*

### Registrasi Intent Receiver

Karena Anda mengirim pesan, belum tentu akan ada action yang diambil. Anda harus me-register intent reciver yang akan mendengarkan adanya intent dan memberitahukan Android apa yang harus dilakukan!

Apakah task itu bertugas memulai activity baru, atau memulai aplikasi yang lama. Jika sebuah receiver bisa menerima intent, maka akan dibuatkan sebuah chooser yang memungkinkan user untuk memilih aplikasi yang akan digunakan. Contoh klasik adalah chooser bisa membuat user memilih apakah hendak menggunakan email, messaging, atau aplikasi lainnya.

Jika sistem Android tidak bisa mencari kecocokkan dengan intent, dan chooser tidak bisa dibuat secara manual, maka aplikasi akan



crash karena eksepsi saat run time. Jadi, developer harus hati-hati. Ketika Anda membuat intent di mana Android tidak mengetahui cara menghandelnya, maka aplikasi akan crash dan minta dimatikan.

### **Kontrol Tanpa Kursor**

Tidak seperti PC yang memiliki mouse untuk menggerakkan kursor di layar, perangkat Android memungkinkan Anda menggunakan jari untuk melakukan apa saja yang bisa dilakukan oleh mouse.

Bagaimana dengan klik kanan? Kalau di perangkat Android, klik dan tekan button atau ikon atau layar dalam jangka waktu tertentu. Kalau membuat button, ukurlah button agak besar karena ukuran jari tiap orang berbeda-beda. Sehingga pastikan ukuran button sesuai dengan yang Anda inginkan.



*Gambar 1.77 Kontrol multitouch di Android*

### View dan Widget

Apakah view itu? View adalah komponen dasar dari elemen UI. Sebuah kotak di layar yang bertanggung jawab untuk menampilkan gambar dan menghandel jika ada event. Jadi, ini mirip dengan kontrol dasar seperti label di HTML. Berikut ini contoh beberapa view.

- ContextMenu.
- Menu.

- View.
- SurfaceView.

Sedangkan Widget adalah elemen UI yang lebih *advanced*, seperti checkbox. Dengan inilah, user akan berinteraksi dengan program. Berikut ini beberapa widget:

- Button.
- Checkbox.
- DatePicker.
- DigitalClock.
- Gallery.
- FrameLayout.
- ImageView
- RelativeLayout.
- PopupWindow.

Selain di atas, masih ada beberapa widget lainnya. Anda bisa melihat di dokumentasi Android pada alamat url <http://developer.android.com/reference/android/widget/package-summary.html>.

## Asynchronous Call

Class AsyncTask adalah sebuah class di Android yang memungkinkan Anda menjalankan berbagai operasi pada saat yang bersamaan. Tanpa harus mengatur thread yang berbeda-beda. AsyncTask tidak hanya memungkinkan Anda memulai proses baru tapi juga mengembalikan hasilnya ke activity yang dimulai. Ini memungkinkan Anda memiliki model pemrograman yang clean untuk pemrosesan asinkronus.

Apakah thread itu? Thread adalah sebuah proses yang berjalan terpisah dan simultan ketika semua proses lainnya tetap berlangsung. Kapan Anda menggunakan asynchronous processing? Yaitu ketika memproses task untuk waktu yang lama. Misalnya di komunikasi internet, pemrosesan media atau lainnya yang mengharuskan user untuk menunggu. Jika user perlu menunggu,

maka Anda harus menggunakan asynchronous call dan memanfaatkan juga beberapa elemen UI agar user tahu bahwa Android sedang diproses.

### Background Service

Service adalah aplikasi yang berjalan di background dan belum tentu memiliki antarmuka. Contohnya aplikasi antivirus yang biasanya berjalan di background sebagai service, walaupun tidak terlihat, tapi sebenarnya berjalan.

Contoh lain adalah proses download yang biasanya dilakukan di Android market (google play), sehingga Anda bisa men-download sambil melakukan hal lain di perangkat Android Anda.

### Tool Perangkat Keras

Perangkat Android memiliki banyak tool sehingga developer diberi peralatan lengkap untuk memanfaatkan tool-tool perangkat keras di Android. Perangkat Android memiliki beberapa fitur hardware yang bisa digunakan untuk membangun aplikasi, seperti:

- GPS Radio: mengetahui posisi.
- Kompas built in: mengetahui arah.
- Sensor proximity: mengetahui apakah perangkat menghadap ke atas atau ke bawah.
- Accelerometer: mengetahui apakah perangkat bergerak atau tidak.
- Bluetooth: menggunakan perangkat bluetooth.
- Kamera: merekam video dan gambar.

Mayoritas perangkat Android memiliki peralatan-peralatan di atas. Tapi tiap pabrikan memiliki spesifikasi berbeda. Ada yang tidak memiliki satu atau dua perangkat tertentu. Di samping itu, perangkat-perangkat baru terus bermunculan dengan fitur perangkat yang tidak didukung secara asli oleh Android. Tapi jangan khawatir, karena pabrikan hardware biasanya menyertakan Software development kit (SDK) yang memungkinkan developer untuk mengembangkan fitur unik Android.

Perangkat Android bisa memiliki ukuran yang bervariasi. Dari mulai ponsel, komputer tablet, hingga ebook reader. Banyak implementasi yang lain pula, seperti Google TV dan perangkat rumahan seperti Microwave dan kulkas.

## Touchscreen

Pada umumnya perangkat Android berjenis layar sentuh (touch screen). Walaupun kadang memiliki keyboard fisik juga. Touch screen ini bisa digunakan untuk menambahkan interaksi user dengan aplikasi. User bisa melakukan swipe, flip, drag dan pinch to zoom. Misalnya dengan menggerakkan jari di touch screen. Anda juga bisa menggunakan gesture custom ke aplikasi Anda yang akan membuka lebih banyak kemungkinan.

Android juga mendukung multitouch yang memungkinkan semua layar disentuh oleh lebih dari satu jari. Button di aplikasi Android adalah virtual, sehingga Anda bisa mendesain di manapun yang Anda inginkan.

## GPS

GPS memungkinkan developer untuk mengakses lokasi user pada saat tertentu. Anda juga bisa melakukan tracking pada gerakan user ketika berpindah posisi. Contohnya aplikasi foursquare. Aplikasi ini menggunakan GPS untuk menentukan lokasi perangkat dan kemudian mengakses web untuk menentukan lokasi atau tempat user terdekat berada.

Contoh lainnya aplikasi map yang memungkinkan Anda untuk menunjukkan lokasi sebuah map dan menyediakan direktori ke tujuan Anda. Android yang memiliki hardware GPS bisa menunjukkan posisi telepon dengan sangat tepat.

## Accelerometer

Accelerometer adalah peranti yang mengukur kecepatan. Dengan demikian, Anda bisa menggunakannya untuk banyak hal, misalnya untuk membuat aplikasi game di mana gerakan dari perangkat Android menjadi controller dari accelerometer.

## SDCard

Android menyediakan tool untuk mengakses file di SD Card yang tertancap ke peranti. Jadi, aplikasi bisa menggunakan sd card untuk menyimpan data. Anda bisa men-download aplikasi dari internet dan kemudian menyimpannya ke sd card. Sehingga program bisa memanfaatkan kapasitas simpan yang tidak terbatas.

## Dukungan Audio dan Video

Android memungkinkan user mengakses audio dan video. Tidak hanya itu, bisa menggunakan efek suara, video instruksional, musik background, video streaming, dan audio dari internet, semuanya bisa ditambahkan ke aplikasi dengan mudah. Anda bisa kreatif menggunakannya.

## Menggunakan Android Manifest

File Android manifest adalah file yang mencatat apa saja yang dibutuhkan aplikasi, request dan apa pun yang diperlukan untuk menjalankan program. File Android manifest disimpan pada bagian root dari project dan dinamakan `AndroidManifest.xml`. Tiap aplikasi harus punya Android Manifest di direktori root.

Android manifest menyediakan semua informasi ke sistem Android. Informasi ini harus ada sebelum kode aplikasi Anda bisa dijalankan. File Android manifest menyediakan beberapa hal berikut:

- Nama paket java dari aplikasi. Ini adalah identitas unik untuk aplikasi di sistem Android dan juga di Google Play/Android market.
- Komponen dari aplikasi, misalnya activity dan service background.
- Deklarasi permission dari aplikasi yang perlu dijalankan.
- Level minimum dari Android API yang dibutuhkan aplikasi.

File Android manifest akan mendeklarasikan versi dari aplikasi. Anda wajib memberikan versi pada aplikasi. Jadi, Anda perlu

mempertimbangkan rilis dan menambahkan versinya di masa depan.

## Permission

Misalnya aplikasi Anda perlu mengakses internet, maka Anda perlu mengatur permission karena pada dasarnya secara default Android akan membatasi akses internet Anda. Jadi, agar aplikasi bisa mengakses internet, Anda perlu mengatur set permisison.

Berikut ini beberapa jenis permission dari aplikasi Android:

- Internet: Aplikasi perlu mengakses internet.
- Write External Storage: Aplikasi perlu menuliskan data ke SD card.
- Camera: Aplikasi perlu akses ke kamera.
- Access Fine Location: Aplikasi perlu akses ke lokasi GPS.

Read phone state: Aplikasi perlu akses ke kondisi telepon (seperti ringing).