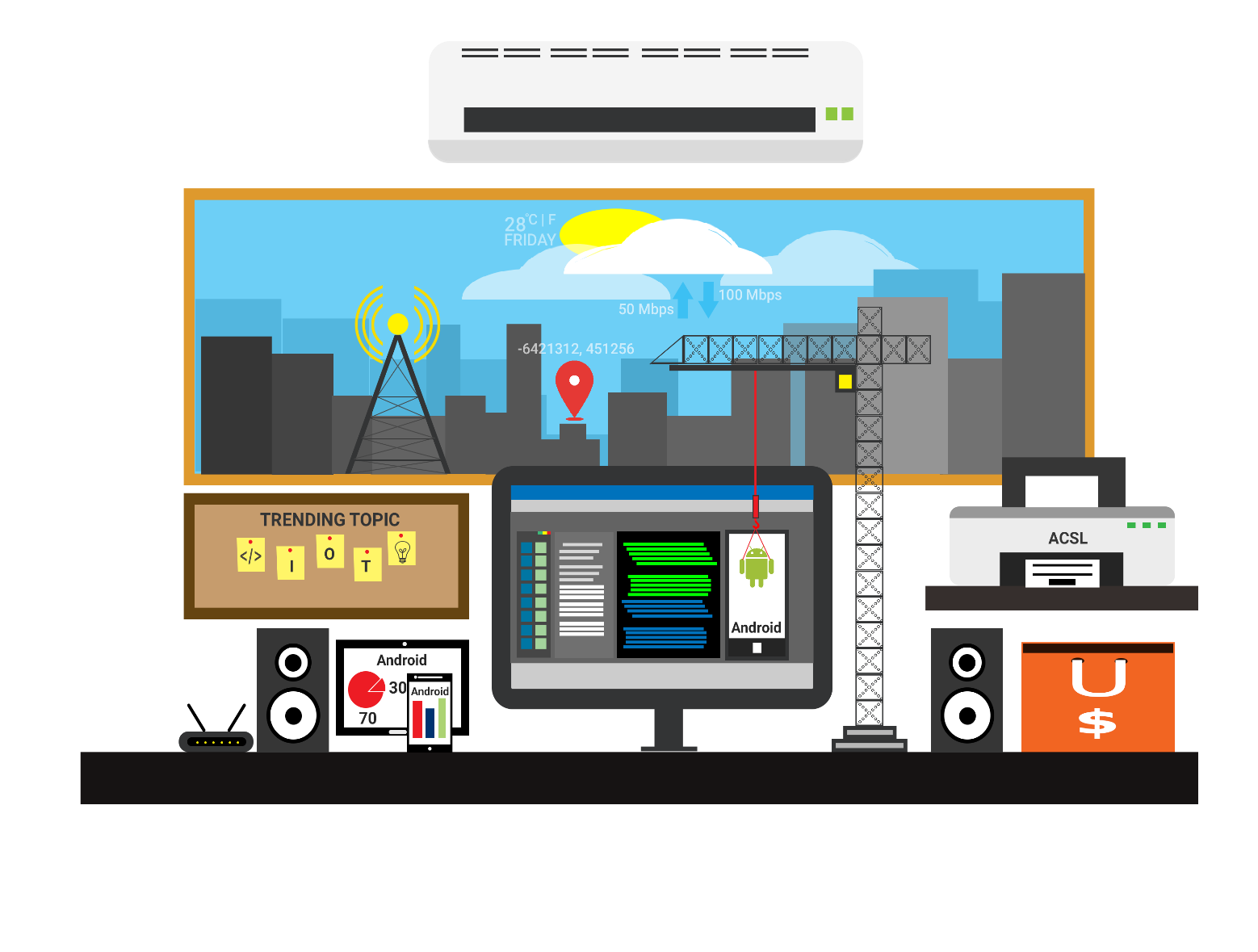


**MODUL PANDUAN**

**MOBILE COMPUTING SYSTEM**

**ATA 2018/2019**



**LABORATORIUM LANJUT SISTEM KOMPUTER**

**UNIVERSITAS GUNADARMA**

**D**

# Daftar Isi

[Daftar Isi i](#_Toc507049688)

[Briefing 1](#_Toc507049689)

[Mobile Computing System (MCS) 1](#_Toc507049690)

[Android 1](#_Toc507049691)

[Arsitektur Android 2](#_Toc507049692)

[Versi Android 6](#_Toc507049693)

[JDK (Java Development Kit) 8](#_Toc507049694)

[Android Studio IDE 8](#_Toc507049695)

[Android SDK (Software Development Kit) 11](#_Toc507049696)

[Gradle 14](#_Toc507049697)

[Android Application Programming Interface 17](#_Toc507049698)

[Layout XML 17](#_Toc507049699)

[Java Classes 20](#_Toc507049700)

[Main Class 24](#_Toc507049701)

[JSON 25](#_Toc507049702)

[Bab 1 : Basic Input / Output Interface 29](#_Toc507049703)

[Interface Editor 29](#_Toc507049704)

[Tampilan Layout 30](#_Toc507049705)

[Menggunakan XML 32](#_Toc507049706)

[Basic Component XML Android 33](#_Toc507049707)

[Material Design 34](#_Toc507049708)

[Bab 2 : Multimedia Interface/Read Data From Sdcard With Json 35](#_Toc507049709)

[JSON (JavaScript Object Notation) 35](#_Toc507049710)

[Audio 35](#_Toc507049711)

[Video 35](#_Toc507049712)

[Image 35](#_Toc507049713)

[Bab 3 : Database Programming (SQLite) 36](#_Toc507049714)

[SQLite 36](#_Toc507049715)

[Create & Close a connection 36](#_Toc507049716)

[Create a database & table 36](#_Toc507049717)

[Insert data to database 36](#_Toc507049718)

[Read data from database 36](#_Toc507049719)

[Update data from database 36](#_Toc507049720)

[Delete data from database 36](#_Toc507049721)

[Delete database & table 36](#_Toc507049722)

[Simple project 36](#_Toc507049723)

[Bab 4 : Google Maps API v2.0 37](#_Toc507049724)

[Basic Viewport 37](#_Toc507049725)

[Current Location 37](#_Toc507049726)

[Custom Icons 37](#_Toc507049727)

[Distance of 2 Object 37](#_Toc507049728)

[Acceleration/Velocity 37](#_Toc507049729)

[Time Calculations 37](#_Toc507049730)

[Bab 5 : Library Cryptography 38](#_Toc507049731)

[Cryptography 38](#_Toc507049732)

[Encryption 38](#_Toc507049733)

[Decription 38](#_Toc507049734)

[MD5 38](#_Toc507049735)

[AES 38](#_Toc507049736)

[Bab 6 : Network Programming (Socket Programming) 39](#_Toc507049737)

[Datagram Socket 39](#_Toc507049738)

[Single Client 39](#_Toc507049739)

[Broadcast to Multiple Client 39](#_Toc507049740)

[TCP Socket 39](#_Toc507049741)

[Single Client 39](#_Toc507049742)

[Multiple Client 39](#_Toc507049743)

[Web Server 39](#_Toc507049744)

[Simple Web Browser 39](#_Toc507049745)

[Simple Web Server 39](#_Toc507049746)

[Bab 7 : Internet of Things (IoT) 40](#_Toc507049747)

[Sensor Signals 40](#_Toc507049748)

[Input Signal 40](#_Toc507049749)

[Output Signal 40](#_Toc507049750)

[Working with Wireless Sensor Network (WSN) 40](#_Toc507049751)

[Client-Server 40](#_Toc507049752)

[Web Based 40](#_Toc507049753)

[Working with Database (SQL) 40](#_Toc507049754)

[Bab 8 : Wireless Sensor Network 41](#_Toc507049755)

[Arsitektur WSN 41](#_Toc507049756)

[Node WSN 41](#_Toc507049757)

# Briefing

**B**

## Mobile Computing System (MCS)

Mobile Computing System adalah kemampuan teknologi untuk menghadapi perpindahan/pergerakan manusia dalam penggunaan komputer secara praktis [Wikipedia].

## Android



Android logo (2014).svg

Gambar B.1 Logo Android

Android merupakan sistem operasi yang dikembangkan untuk perangkat mobile berbasis Linux yang dirancang untuk perangkat mobile dengan layar sentuh seperti smartphone dan komputer tablet. Sistem operasi Linux yang mendasari Android dilisensikan di bawah GNU, General Public License Versi 2 (GPLv2), yang dikenal dengan istilah copyleft. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler.

Dari perkembangan sistem operasi, Android ini yang sekarang menjadi sangat populer karena bersifat *open source,* menjadikannya sebagai sistem operasi yang banyak diminati oleh banyak pengguna. Adapun berberapa kelebihan dari sistem operasi Android adalah sebagai berikut :

1. *Complete Platform*

Sistem operasi Android adalah sistem operasi yang banyak menyediakan *tools* yang berguna untuk membangun sebuah aplikasi yang kemudian aplikasi tersebut dapat lebih dikembangkan lagi oleh para developer.

1. *Open Source Platform*

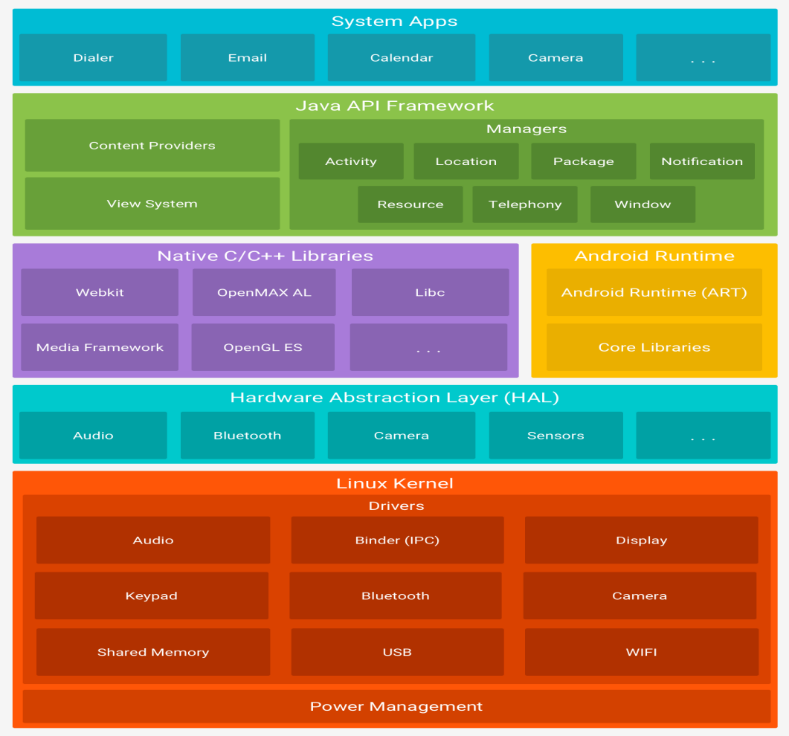
Platform Android yang bersifat *open source* menjadikan sistem operasi ini mudah dikembangkan oleh para developer karena bersifat terbuka.

1. *Free Platform*

Developer dengan bebas bisa mengembangkan, mendistribusikan dan memperdagangkan sistem operasi Android tanpa harus membayar royalti untuk mendapatkan license.

## Arsitektur Android

Penggambaran sebuah arsitektur dari sistem operasi Android, jika dilihat secara garis besar adalah sebagai berikut:



Gambar B.2 Arsitektur Android

1. *Linux Kernel*

Landasan platform Android adalah kernel Linux. Misalnya, Android Runtime (ART) bergantung pada kernel Linux untuk fungsionalitas dasar seperti threading dan manajemen memori tingkat rendah.

Menggunakan kernel Linux memungkinkan Android memanfaatkan fitur keamanan utama dan memungkinkan produsen perangkat mengembangkan driver perangkat keras untuk kernel yang terkenal.

1. *Hardware Abstraction Layer (HAL)*

Lapisan abstraksi perangkat keras menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras perangkat ke kerangka kerja API tingkat tinggi. HAL terdiri dari beberapa modul perpustakaan, yang masing-masing mengimplementasikan sebuah antarmuka untuk jenis komponen perangkat keras tertentu, seperti modul kamera atau bluetooth. Ketika API kerangka membuat panggilan untuk mengakses perangkat keras perangkat, sistem Android memuat modul perpustakaan untuk komponen perangkat keras tersebut.

1. *Android Runtime*

Untuk perangkat yang menjalankan Android versi 5.0 (tingkat API 21) atau lebih tinggi, setiap aplikasi berjalan dalam prosesnya sendiri dan dengan turunannya dari Android Runtime (ART). ART ditulis untuk menjalankan beberapa mesin virtual pada perangkat memori rendah dengan mengeksekusi file DEX, sebuah format bytecode yang dirancang khusus untuk Android yang dioptimalkan untuk jejak memori minimal. Bangun toolchains, seperti Jack, kompilasi source Java ke dalam bytecode DEX, yang bisa berjalan di platform Android.

Beberapa fitur utama ART adalah sebagai berikut:

* Kompilasi Ahead-of-time (AOT) dan just-in-time (JIT)
* Pengumpulan sampah yang dioptimalkan (GC)
* Dukungan debug yang lebih baik, termasuk profiler sampling khusus, pengecualian diagnostik terperinci dan pelaporan kecelakaan, dan kemampuan untuk mengatur titik pandang untuk memantau bidang tertentu

Sebelum Android versi 5.0 (tingkat API 21), Dalvik adalah runtime Android. Jika aplikasi *User* berjalan dengan baik pada ART, maka itu harus bekerja pada Dalvik juga, namun sebaliknya mungkin tidak benar.

Android juga menyertakan seperangkat perpustakaan runtime inti yang menyediakan sebagian besar fungsi bahasa pemrograman Java, termasuk beberapa fitur bahasa Java 8, yang menggunakan kerangka Java API.

1. Native C/C++ Libraries

Banyak komponen dan layanan sistem Android utama, seperti ART dan HAL, dibangun dari kode asli yang memerlukan perpustakaan asli yang ditulis dalam bahasa C dan C ++. Platform Android menyediakan API kerangka Java untuk mengekspos fungsionalitas beberapa perpustakaan asli ini ke aplikasi. Misalnya, *User* dapat mengakses OpenGL ES melalui kerangka Java OpenGL API kerangka untuk menambahkan dukungan untuk menggambar dan memanipulasi grafis 2D dan 3D di aplikasi *User*.

Jika *User* mengembangkan aplikasi yang memerlukan kode C atau C ++, *User* dapat menggunakan Android NDK untuk mengakses beberapa perpustakaan platform asli ini langsung dari kode asli *User*.

1. Java API Framework

Seluruh fitur-set OS Android tersedia untuk *User* melalui API yang ditulis dalam bahasa Jawa. API ini membentuk blok bangunan yang *User* butuhkan untuk membuat aplikasi Android dengan menyederhanakan penggunaan ulang komponen, komponen dan layanan komponen modular, yang mencakup hal berikut:

* Sistem Tampilan yang kaya dan dapat diperluas yang dapat *User* gunakan untuk membuat UI aplikasi, termasuk daftar, grid, kotak teks, tombol, dan bahkan browser web yang dapat disematkan
* Resource Manager, menyediakan akses ke sumber daya non-kode seperti string lokal, grafik, dan file tata letak
* Pengelola Notifikasi yang memungkinkan semua aplikasi menampilkan lansiran ubahsuaian di baris status
* Pengelola Aktivitas yang mengelola siklus hidup aplikasi dan menyediakan tumpukan belakang navigasi yang umum
* Penyedia Konten yang memungkinkan aplikasi mengakses data dari aplikasi lain, seperti aplikasi Kontak, atau untuk berbagi data mereka sendiri

Pengembang memiliki akses penuh ke API kerangka kerja yang sama yang digunakan aplikasi sistem Android.

1. System Apps

Android hadir dengan satu set aplikasi inti untuk email, pesan SMS, kalender, penjelajahan internet, kontak, dan lainnya. Aplikasi yang disertakan dengan platform tidak memiliki status khusus di antara aplikasi yang dipilih pengguna untuk dipasang. Jadi aplikasi pihak ketiga bisa menjadi browser web default pengguna, SMS messenger, atau bahkan keyboard default (beberapa pengecualian berlaku, seperti aplikasi Pengaturan sistem).

Aplikasi sistem berfungsi baik sebagai aplikasi untuk pengguna dan untuk menyediakan kemampuan kunci yang dapat diakses pengembang dari aplikasi mereka sendiri. Misalnya, jika aplikasi *User* ingin mengirimkan pesan SMS, *User* tidak perlu membangun fungsionalitas itu sendiri-*User* malah dapat meminta aplikasi SMS mana pun yang sudah terpasang untuk mengirimkan pesan ke penerima yang *User* tentukan.

## Versi Android

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama kode** | **Nomor versi** | **Tanggal rilis awal** | **API level** |
| N/A | 1.0 | 23 September 2008 | 1 |
| N/A | 1.1 | 9 Februari 2009 | 2 |
| [Cupcake](https://en.wikipedia.org/wiki/Android_Cupcake) | 1.5 | 27 April 2009 | 3 |
| [Donut](https://en.wikipedia.org/wiki/Android_Donut)  tidak didefinisikan | 1.6 | 15 September 2009 | 4 |
| [Eclair](https://en.wikipedia.org/wiki/Android_Eclair)  tidak didefinisikan | 2.0 – 2.1 | 26 Oktober 2009 | 5–7 |
| [Froyo](https://en.wikipedia.org/wiki/Android_Froyo)  tidak didefinisikan | 2.2 – 2.2.3 | 20 Mei 2010 | 8 |
| [Gingerbread](https://en.wikipedia.org/wiki/Android_Gingerbread)  tidak didefinisikan | 2.3 – 2.3.7 | 6 Desember 2010 | 9–10 |
| [Honeycomb](https://en.wikipedia.org/wiki/Android_Honeycomb)  tidak didefinisikan | 3.0 – 3.2.6 | 22 Februari 2011 | 11–13 |
| [Ice Cream Sandwich](https://en.wikipedia.org/wiki/Android_Ice_Cream_Sandwich)  tidak didefinisikan | 4.0 – 4.0.4 | 18 Oktober 2011 | 14–15 |
| [Jelly Bean](https://developer.android.com/about/versions/jelly-bean.html)  tidak didefinisikan | 4.1 – 4.3.1 | 9 Juli 2012 | 16–18 |
| [KitKat](https://developer.android.com/about/versions/kitkat.html)  tidak didefinisikan | 4.4 – 4.4.4 | 31 Oktober 2013 | 19–20 |
| [Lollipop](https://developer.android.com/about/versions/lollipop.html)  tidak didefinisikan | 5.0 – 5.1.1 | 12 November 2014 | 21–22 |
| [Marshmallow](https://developer.android.com/about/versions/marshmallow/index.html)  tidak didefinisikan | 6.0 – 6.0.1 | 5 Oktober 2015 | 23 |
| [Nougat](https://developer.android.com/about/versions/nougat/index.html)  tidak didefinisikan | 7.0 | 22 Agustus 2016 | 24 |

## JDK (Java Development Kit)



JDK adalah sebuah perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java ke bytecode yang dapat dimengerti dan dapat dijalankan oleh JRE (Java Runtime Envirotment). Alasan utama untuk menginstal JDK adalah Pada dasarnya Aplikasi Android ini dikembangkan menggunakan bahasa pemrograman Java, oleh karena itu JDK menjadi sangat penting untuk diinstall ketika kita ingin membangun aplikasi Android.

## Android Studio IDE

Android Studio adalah sebuah IDE yang bisa digunakan untuk pengembangan aplikasi Android, dan dikembangkan oleh Google. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio direncanakan untuk menggantikan Eclipse ke depannya sebagai IDE resmi untuk pengembangan aplikasi Android.



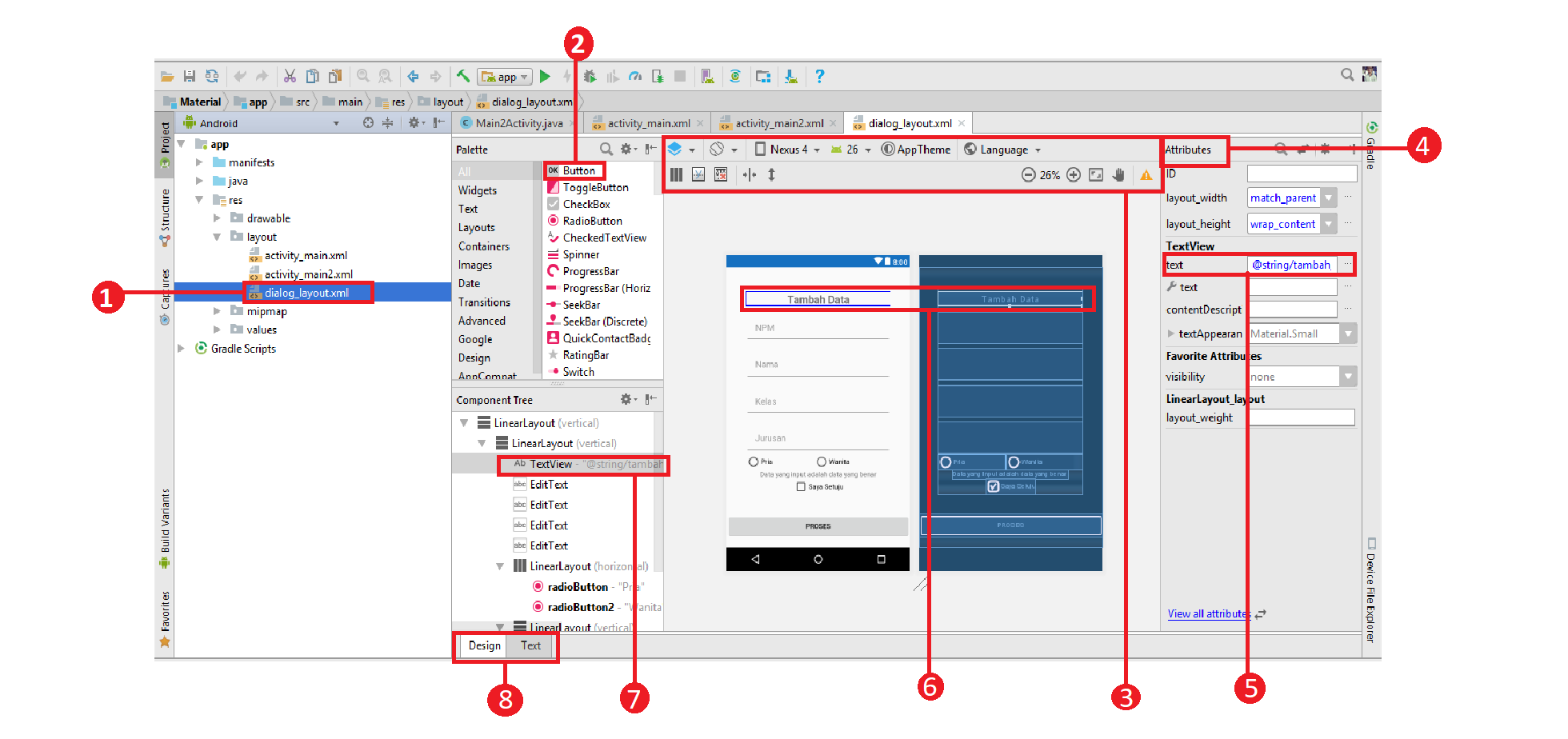
Gambar B.3 Logo Android Studio

Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan ADT, Android Studio menggunakan Gradle sebagai build environment. Fitur-fitur lainnya adalah sebagai berikut :

* Menggunakan Gradle-based build system yang fleksibel.
* Bisa mem-*build multiple* APK.
* *Template support* untuk Google *Services* dan berbagai macam tipe perangkat.
* Layout editor yang lebih bagus

Editor Layout Android Studio

Pada pembuatan user interface, android studio menyediakan layout yang digunakan untuk menambahkan beberapa komponen UI yang diperlukan. Mode yang disediakan untuk pembuatan UI ada 2, yaitu Mode Design yang mana mode ini memungkinkan kita dapat *drag and drop* komponen UI yang diperlukan ke sebuah layout.



Gambar B.4 Editor Layout Android Studio

Dalam gambar di atas:

1. File XML layout. File layout XML, biasanya diberi nama file activiy\_main.xml. Klik dua kali untuk membuka editor layout.
2. Palet elemen UI (tampilan). Panel Palette menyediakan daftar elemen UI dan layout. Tambahkan elemen atau layout ke UI dengan menyeretnya ke panel desain.
3. Bilah alat desain. Bilah alat panel desain menyediakan tombol untuk mengonfigurasi penampilan layout dalam panel desain dan untuk mengedit properti layout. Lihat gambar di bawah ini untuk detail.
4. Panel Properties. Panel Properties menyediakan kontrol properti untuk tampilan yang dipilih.
5. Kontrol properti. Kontrol properti sesuai dengan atribut XML. Yang ditampilkan dalam gambar adalah properti Text dari TextView yang dipilih, yang disetel kedalam file string dengan nama string yaitu tambah.
6. Panel desain. Seret tampilan dari panel Palette ke panel desain untuk memosisikannya di layout.
7. Component Tree. Panel Component Tree menampilkan hierarki tampilan. Klik tampilan atau grup tampilan dalam panel ini untuk memilihnya. Gambar menampilkan TextView yang dipilih.
8. Tab Design dan Text. Klik Design untuk melihat editor layout, atau Text untuk melihat kode XML.

## Android SDK (Software Development Kit)

Android SDK adalah tools API (*Application Programmming Interface)* yang diperlukan untuk memulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Saat ini disediakan Android SDK merupakan alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java.

Sebagai platform aplikasi-netral, Android memberi *user* kesempatan untuk membuat Aplikasi yang kita butuhkan yang bukan merupakan bawaan dari *Smartphone/Handphone.* ada beberapa fitur-fitur android yang peting seperti, *Framework,* Mesin Virtual Dalvik, Integrated browser, Grafis yang dioptimalkan, SQLite, Media Support, Bluetooth, EDGE, WIFI, lingkungan development dan sebagainya

Android *Manifest*



Android manifest berfungsi untuk mendeklarasikan class-class yang digunakan, memanggil kelas yang pertama dijalankan, serta dapat merubah versi Android sesuai yang diinginkan.

**Namespace dan tag aplikasi Android**

Manifes Android dikodekan di XML dan selalu menggunakan namespace Android:

xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.android.helloworld">

Ekspresi package menampilkan nama paket unik aplikasi baru. Jangan mengubah ini setelah aplikasi dipublikasikan.

<application

...

</application>

Tag <application , dengan tag </application> penutupnya, mendefinisikan setelan manifes untuk keseluruhan aplikasi.

**Cadangan otomatis**

Atribut android:allowBackup memungkinkan pencadangan data aplikasi otomatis:

...

android:allowBackup="true"

...

Menyetel atribut android:allowBackup ke true memungkinkan aplikasi dicadangkan secara otomatis dan dipulihkan jika perlu. Pengguna menggunakan waktu dan upaya untuk mengonfigurasi aplikasi. Beralih ke perangkat baru bisa membatalkan seluruh konfigurasi yang dibuat dengan susah payah. Sistem melakukan cadangan otomatis ini untuk hampir seluruh data aplikasi secara default, dan melakukannya tanpa mengharuskan developer menulis kode aplikasi tambahan lainnya.

Untuk aplikasi dengan versi SDK target berupa Android 6.0 (API level 23) dan yang lebih tinggi, perangkat yang menjalankan Android 6.0 dan yang lebih tinggi secara otomatis membuat cadangan data aplikasi ke awan karena atribut android:allowBackup default ke true jika dihilangkan. Untuk aplikasi < API level 22, Anda harus secara eksplisit menambahkan atribut android:allowBackup dan menyetelnya ke true.

**Ikon aplikasi**

Metode android:icon menyetel ikon untuk aplikasi:

...

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

...

Atribut android:icon menetapkan ikon di folder **mipmap** (di dalam folder **res** di Project: tampilan Android) untuk aplikasi. Ikon muncul di Peluncur untuk meluncurkan aplikasi. Ikon juga digunakan sebagai ikon default untuk komponen aplikasi.

**Sumber daya string dan label aplikasi**

Seperti yang bisa Anda lihat di gambar sebelumnya, atribut android:label menampilkan string "Hello World" yang disorot. Jika Anda mengeklik string ini, maka string berubah untuk menampilkan sumber daya string @string/app\_name:

...

android:label="@string/app\_name"

...

**Tip**: Ctrl - klik atau klik-kanan app\_name di panel edit untuk melihat menu konteks. Pilih **Go To > Declaration** untuk melihat lokasi sumber daya string dideklarasikan: dalam file strings.xml. Bila Anda memilih **Go To > Declaration** atau membuka file dengan mengeklik dua kali **strings.xml** di Project: Tampilan Android (di dalam folder **values**), isinya muncul di panel pengeditan.

Setelah membuka file strings.xml, Anda bisa melihat bahwa nama string app\_name disetel ke Hello World. Anda bisa mengubah nama aplikasi dengan mengubah string Hello World ke hal lain. Sumber daya string dijelaskan dalam pelajaran terpisah.

**Tema aplikasi**

Atribut android:theme menyetel tema aplikasi, yang mendefinisikan penampilan elemen antarmuka pengguna seperti teks:

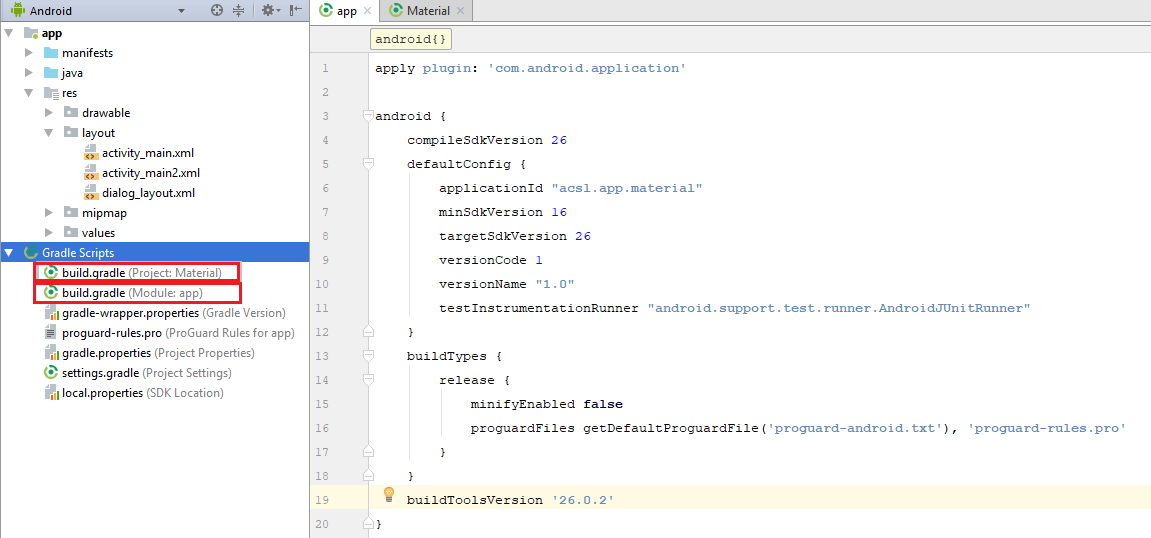
...

android:theme="@style/AppTheme">

...

Atribut theme disetel ke tema standar AppTheme. Tema dijelaskan dalam pelajaran terpisah.

## Gradle



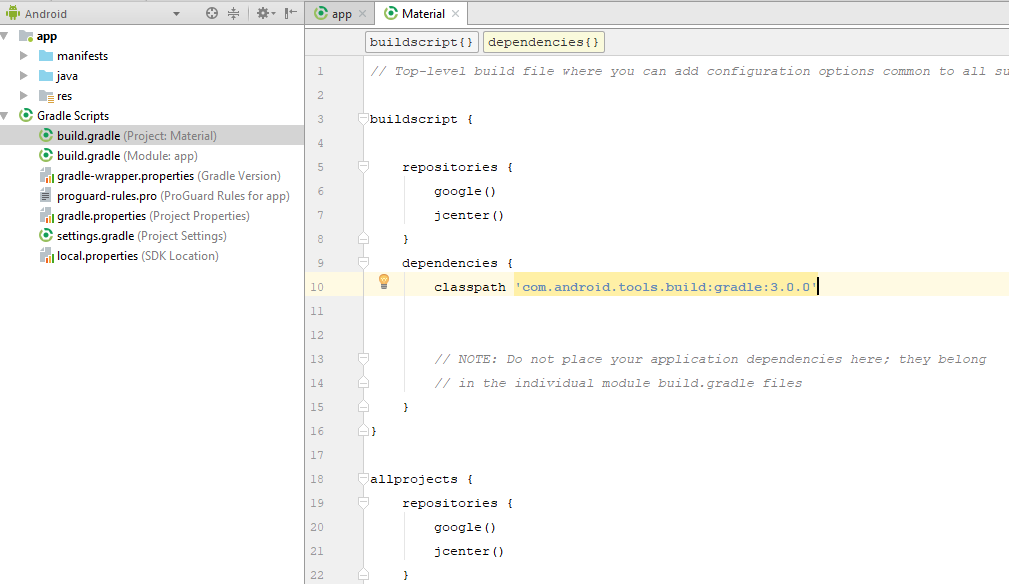
Gambar B.5 Gradle

Gradle adalah build tools yang digunakan pada Android Studio, untuk mencompile-menjalankan project Aplikasi yang sedang dikembangkan, berdasarkan settingan atau konfigurasi yang berada di bagian Gradle Script.[4] Jika di Eclipse build tools yang digunakan adalah Ant, Maka Gradle merupakan pengembangan yang di bangun dari Ant, Maven, dan repositori lvy. Android Plugin untuk Gradle berjalan secara independent (bebas) di Android Studio. Yang berarti bisa membangun Aplikasi Android di dalam Android Studio, serta dapat membangun Aplikasi melalui Command Line menggunakan Gradle dengan mempunyai fitur yang spesifikasinya diperuntukan untuk Aplikasi Android ,Menentukan build tipe ,flavors ,Signing configurations dan Menambahkan library project.

**Tipe – tipe Gradle**

build.gradle (Project)

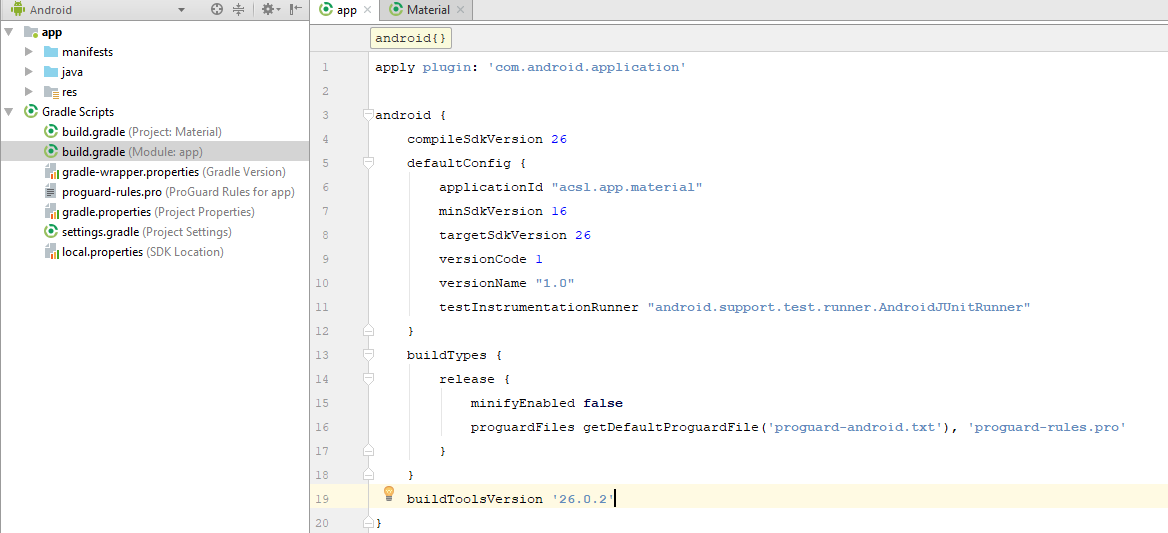
Ini adalah file pembangunan tingkat atas untuk keseluruhan proyek, berada di akar direktori proyek, yang mendefinisikan konfigurasi pembangunan yang berlaku untuk semua modul di proyek Anda.



File ini, yang dihasilkan oleh Android Studio, tidak boleh diedit untuk menyertakan dependensi aplikasi.

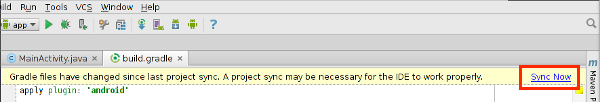
build.gradle (Modul)

Android Studio membuat file build.gradle (Module: app) terpisah untuk setiap modul. Anda bisa mengedit setelan pembangunan guna menyediakan opsi pemaketan khusus untuk setiap modul, seperti tipe pembangunan tambahan dan ragam produk, dan untuk menggantikan setelan di file manifes atau build.gradle tingkat atas.



File ini paling sering adalah file untuk mengedit ketika mengubah konfigurasi tingkat aplikasi, seperti mendeklarasikan dependensi di bagian dependencies.

**Menyinkronkan proyek**



Bila Anda membuat perubahan pada file konfigurasi pembangunan dalam proyek, Android Studio akan mengharuskan Anda untuk melakukan sinkronisasi file proyek sehingga Android Studio bisa mengimpor perubahan konfigurasi pembangunan dan menjalankan beberapa pemeriksaan untuk memastikan konfigurasi tidak akan menimbulkan kesalahan pembangunan.

Untuk menyinkronkan file proyek, klik Sync Now di bilah notifikasi yang muncul saat membuat perubahan, atau klik Sync Project dari bilah menu. Jika Android Studio memperlihatkan kesalahan apa pun dengan konfigurasi — misalnya, jika kode sumber menggunakan fitur API yang hanya tersedia di API level yang lebih tinggi dari compileSdkVersion — jendela Messages muncul untuk menjelaskan masalah. Menyinkronkan File Gradle di Proyek Android Studio

## Android Application Programming Interface

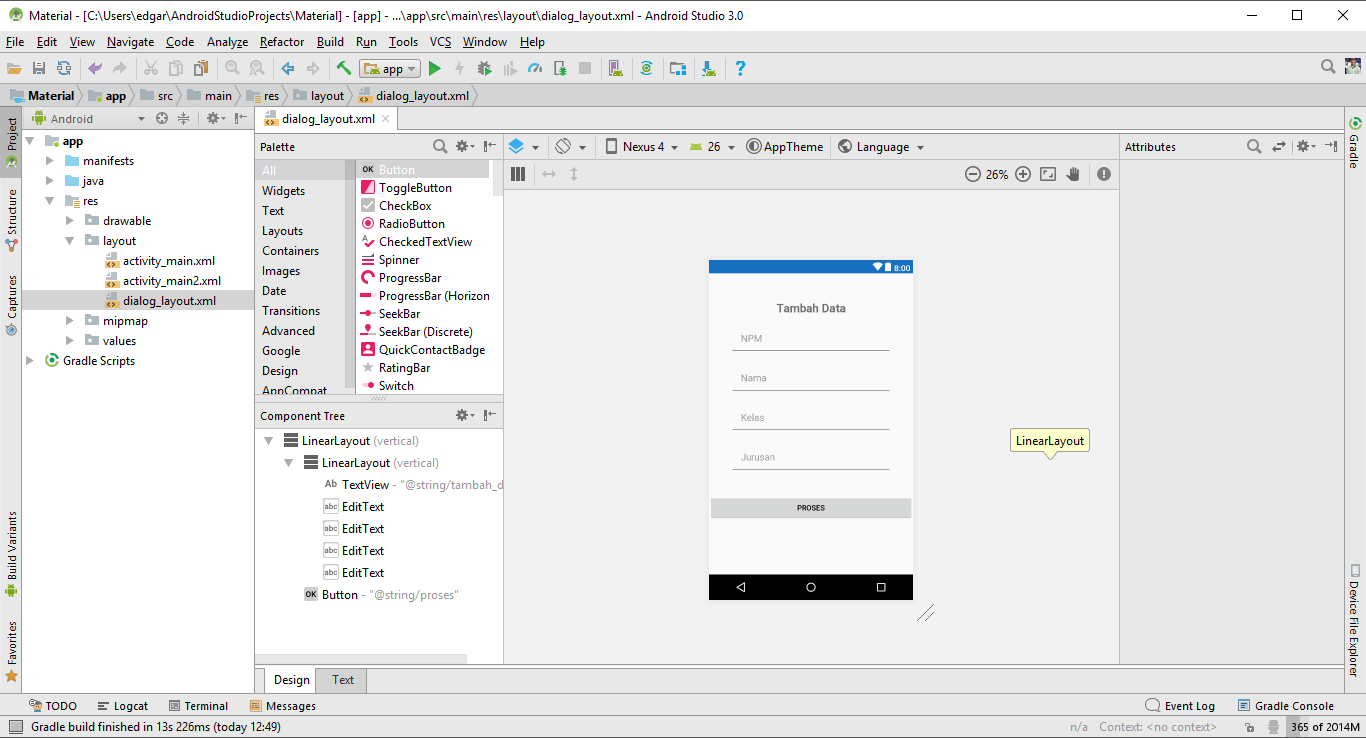
Application Programming Interface (API / Antarmuka Pemrograman Aplikasi) merupakan sekumpulan baris perintah, fungsi dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak. Alasan programmer menggunakan API adalah untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi.

Keuntungan memprogram dengan menggunakan API adalah:

* Portabilitas. Programmer yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah ter- install API tersebut. Sedangkan system callberbeda antar sistem operasi, dengan catatan dalam implementasinya mungkin saja berbeda.
* Lebih Mudah Dimengerti. API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa system call. Hal ini sangat penting dalam hal editing dan pengembangan.

Ada banyak penyedia API yang terdapat pada Android. Contohnya Gmap, Admob, PDF reader, penyedia cuaca dan lain sebagainya.

## Layout XML



Extensible Markup Language (XML) adalah file format yang digunakan untuk mendeskripsikan data. Beda dengan HTML (Hyper Text Markup Language) yang difokuskan untuk menampilkan data.

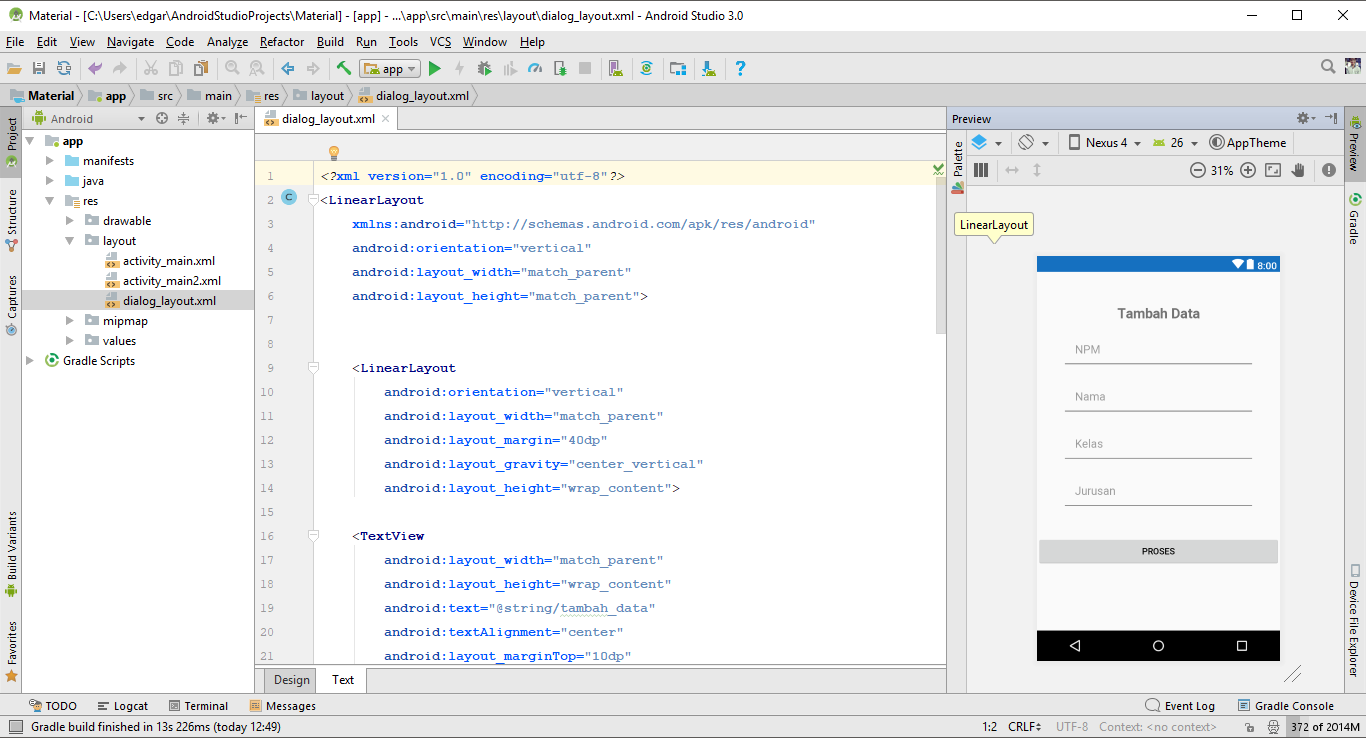
Layout mendefinisikan struktur visual untuk antarmuka pengguna, seperti UI (User Interface) sebuah aktivitas atau widget aplikasi. Anda dapat mendeklarasikan layout dengan dua cara:

* Mendeklarasikan elemen UI dalam XML. Android menyediakan sebuah kosakata XML sederhana yang sesuai dengan kelas dan subkelas View, seperti halnya untuk widget dan layout.
* Buat instance elemen layout saat waktu proses. Aplikasi Anda bisa membuat objek View dan ViewGroup (dan memanipulasi propertinya) lewat program.

Kerangka kerja Android memberi Anda fleksibilitas untuk menggunakan salah satu atau kedua metode ini guna mendeklarasikan dan mengelola UI aplikasi Anda. Misalnya, Anda bisa mendeklarasikan layout default aplikasi Anda dalam XML, termasuk elemen-elemen layar yang akan muncul di dalamnya dan di propertinya. Anda nanti bisa menambahkan kode dalam aplikasi yang akan memodifikasi status objek layar, termasuk yang dideklarasikan dalam XML, saat waktu proses.

Keuntungan mendeklarasikan UI dalam XML adalah karena hal ini memungkinkan Anda memisahkan penampilan aplikasi dari kode yang mengontrol perilakunya dengan lebih baik. Keterangan UI Anda bersifat eksternal bagi kode aplikasi Anda, yang berarti bahwa Anda bisa memodifikasi atau menyesuaikannya tanpa harus memodifikasi dan mengompilasi ulang kode sumber. Misalnya, Anda bisa membuat layout XML untuk berbagai orientasi layar, berbagai ukuran layar perangkat, dan berbagai bahasa.

**Sintaks XML**



Dengan menggunakan kosakata XML Android, Anda bisa mendesain secara cepat layout UI dan elemen layar yang dimuatnya, sama dengan cara membuat laman web dalam HTML dengan serangkaian elemen tersarang.

Tiap file layout harus berisi persis satu elemen akar, yang harus berupa sebuah objek View atau ViewGroup. Setelah mendefinisikan elemen akar, Anda bisa menambahkan objek atau widget layout tambahan sebagai elemen anak untuk membangun hierarki View yang mendefinisikan layout Anda secara bertahap. Misalnya, inilah layout XML yang menggunakan LinearLayout vertikal untuk menyimpan TextView dan Button:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
              android:layout\_width="match\_parent"  
              android:layout\_height="wrap\_content"  
              android:orientation="vertical" >  
    <TextView android:id="@+id/text1"  
              android:layout\_width="wrap\_content"  
              android:layout\_height="wrap\_content"  
              android:text="Halo, saya developer android ACSL" />  
    <Button android:id="@+id/button1"  
            android:layout\_width="wrap\_content"  
            android:layout\_height="wrap\_content"  
            android:text="Tombol" />  
</LinearLayout>

Memasukan source code XML dalam file Java

Saat mengompilasi aplikasi, masing-masing file layout XML akan dikompilasi dalam sebuah sumber daya View. Anda harus memuat sumber daya layout dari kode aplikasi, dalam implementasi callback Activity.onCreate(). Lakukan dengan memanggil setContentView(), dengan meneruskan acuan ke sumber daya layout berupa: R.layout.layout\_file\_name. Misalnya, jika XML layout Anda disimpan sebagai activity\_main.xml, Anda akan memuatnya untuk Activity seperti ini:

public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity\_main);  
}

## Java Classes

Pada pengembangan aplikasi android, untuk mengaktifkan fungsi logic dibutuhkan file java yang dapat mengatasi hal tersebut. Java sendiri adalah bahasa pemrograman yang berorientasi object (Object Oriented Programming : OOP). OOP adalah sebuah metodologi atau cara berpikir dalam melakukan pemrograman dimana pendefinisian tipe data disertai dengan pendefinisian fungsi. Struktur data yang seperti ini disebut dengan istilah object. Paradigma pemrograman OOP dapat dilihat sebagai interaksi sebuah object dalam melakukan tugasnya. Java menggabungkan banyak fitur-fitur canggih dari bahasa-bahasa canggih tersebut, sambil mengatasi beberapa kelemahan mereka. Walaupun demikian, tingkat kecanggihan bahasa pemrograman bergantung pada library mereka. Library ini ada untuk membantu para developer untuk membuat aplikasi.

Class adalah sebuah konsep OOP yang digunakan untuk mencapsulasi /membungkus data dan abstraksi prosedural yang diperlukan dalam menggambarkan isi dan tingkah laku berbagai entitas . Kelas juga merupakan deskripsi tergeneralisir (misal templet, pola, cetak biru) yang menggambarkan kumpulan objek yang sama.

**Konsep – Konsep OOP**

Abtraksi (Abstraction)

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari “pelaku” abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

Enkapsulasi (Encapsulation)

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak, hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

Polimorfisme (Polymorphism)

Melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan, metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan “gerak cepat”, dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

Inheritas (inheritance)

Mengatur polimorfisme dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa haru mengimplementasi ulang perilaku tersebut (bahasa berbasis objek tidak selalu memiliki inheritas.)

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

**Jenis jenis modifier**

Pada pemgrogaman OOP java terdapat prinsip utama yaitu Encapsulation (enkapsulasi) atau yang dikenal juga sebagai information hiding. Yang artinya anda bisa menentukan aksesibilitas apakah suatu method atau variabel dari suatu kelas bisa diakses atau dimodifikasi oleh kelas lain. Pada umumnya di OOP terdapat 3 modifier yang berfungsi untuk mengatur aksesibilitas dari method dan variabel. Modifier tersebut antara lain :

Modifier public

Public artinya kelas, method, maupun variabel yang memiliki modifier ini bisa diakses oleh semua class, baik yang ada di package itu maupun yang bukan. Modifier ini juga memberikan akses kepada objek yang diinstans (dibentuk) di kelas tersebut.

Modifier private

Sifat Private artinya kelas, method, ataupun variabel yang memiliki modifier ini hanya bisa diakses oleh kelas itu sendiri. Kelas lain, maupun objek yang dibentuk oleh kelas lain sama sekali tidak punya akses method, kelas, maupun variabel dengan modifier private. Karena sifatnya inilah, maka method/var private tidak bisa diwariskan ke subkelas.

Modifier protected

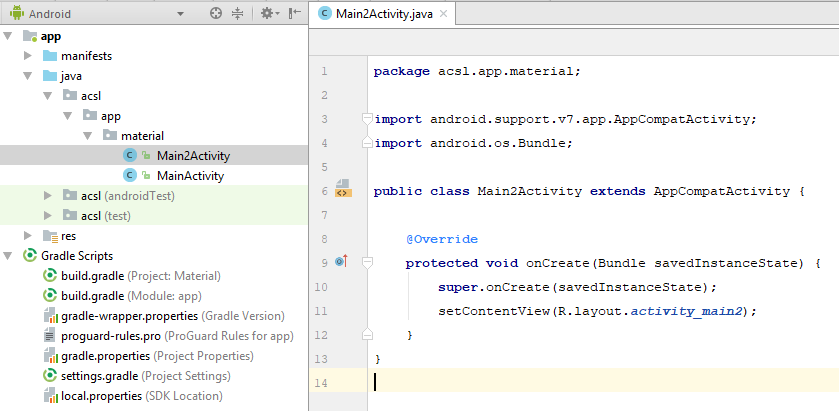
Sama halnya seperti private, protected juga membatasi akses terhadap method/variabel/kelas, namun tidak bersifat total. Modifier ini biasanya digunakan untuk membatasi akses terhadap:

* Subkelas dari suatu kelas
* Kelas lain dari package yang sama

Berikut tabel ringkasan aksesibilitas dari masing-masing modifier

|  |  |  |  |
| --- | --- | --- | --- |
| **Aksesibilitas** | **public** | **private** | **protected** |
| Dari kelas yang sama | Ya | Ya | Ya |
| Dari kelas yang berada dalam package yang sama | Ya | Tidak | Ya |
| Dari kelas di luar package | Ya | Tidak | Tidak |
| Dari subkelas dalam package yang sama | Ya | Tidak | Ya |
| Dari subkelas di luar package | Ya | Tidak | Ya |

## Main Class



Ada yang perlu Anda perhatikan dari kode diatas:

**Package** adalah nama dari proyek aplikasi dan company proyek

**Import**  adalah perintah yang digunakan untuk memasukan library

java / android untuk kebutuhan aplikasi

**public**  adalah sebuah keyword di Java yang menandakan bahwa objek, method, atau atribut dapat diakses dari class lain.

**class**  adalah sebuah keyword di Java yang digunakan untuk membuat sebuah class

**void**  adalah sebuah keyword untuk membuat sebuah method tidak me-return nilai apapun alias kosong

**setContentView** adalah sebuah keyword untuk menghubungkan file XML dengan file Java

## JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

JSON terbuat dari dua struktur:

Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau associative array.

Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini.

JSON menggunakan bentuk sebagai berikut:

**Objek** adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan **{** (kurung kurawal buka) dan diakhiri dengan **}** (kurung kurawal tutup). Setiap nama diikuti dengan **:** (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh **,** (koma).



**Array** adalah kumpulan nilai yang terurutkan. Larik dimulai dengan **[** (kurung kotak buka) dan diakhiri dengan **]** (kurung kotak tutup). Setiap nilai dipisahkan oleh **,** (koma).



**Nilai** (*value*) dapat berupa sebuah **string** dalam tanda kutip ganda, atau *angka*, atau **true** atau **false** atau **null**, atau sebuah *objek* atau sebuah *larik*. Struktur-struktur tersebut dapat disusun bertingkat.



**String** adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.



**Angka** adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Spasi kosong (*whitespace*) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detil *encoding* yang secara lengkap dipaparkan oleh bahasa pemprograman yang bersangkutan.

Contoh source code parsing Json

{

"firstName": "John",

"lastName": "Smith",

"age": 25,

"address":

{

"streetAddress": "21 2nd Street",

"city": "New York",

"state": "NY",

"postalCode": "10021"

},

"phoneNumber":

[

{

"type": "home",

"number": "212 555-1234"

},

{

"type": "fax",

"number": "646 555-4567"

}

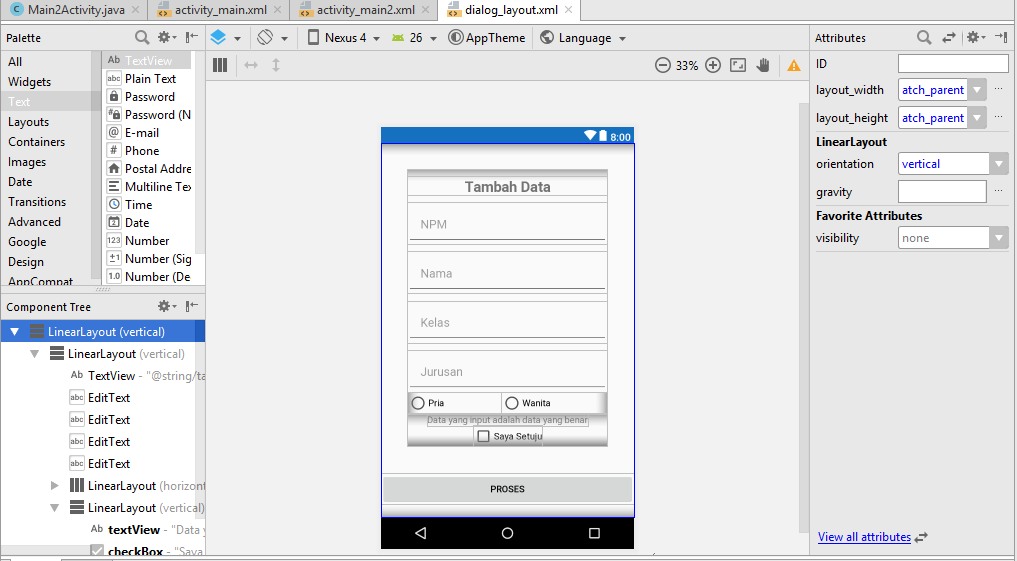
]

}

**1**

# Bab 1 : Basic Input / Output Interface

## Interface Editor



Tampilan antarmuka (interface)

UI (User Interface) terdiri dari hierarki objek yang disebut tampilan, setiap elemen layar adalah tampilan. Class View menyatakan blok pembangunan dasar untuk semua komponen UI, dan kelas dasar untuk kelas yang menyediakan komponen UI interaktif seperti Button, RadioButton, TextView dan Inputan Text.

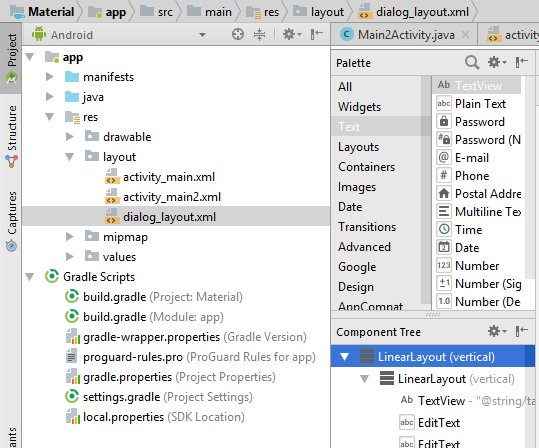
Tampilan memiliki lokasi, yang dinyatakan sebagai pasangan koordinat kiri dan atas, dan dua dimensi, yang dinyatakan sebagai lebar dan tinggi. Unit untuk lokasi dan dimensi adalah piksel yang tidak tergantung perangkat (dp / density pixel).

Sistem Android menyediakan ratusan tampilan yang telah didefinisikan sebelumnya, termasuk yang menampilkan:

* Teks (TextView)
* Bidang untuk memasukkan dan mengedit teks (EditText)
* Pengguna tombol bisa mengetuk (Button) dan komponen interaktif lainnya
* Teks yang bisa digulir (ScrollView) dan item yang bisa digulir (RecyclerView)
* Gambar (ImageView)

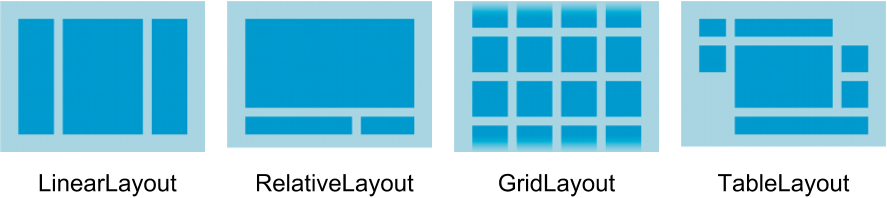
Anda bisa mendefinisikan tampilan untuk muncul di layar dan merespons ketukan pengguna. Tampilan juga bisa didefinisikan untuk menerima masukan teks, atau tidak terlihat hingga diperlukan.

Anda bisa membuat sebuah tampilan interface di file sumber daya layout XML. Sumber daya layout ditulis dalam XML dan dicantumkan dalam folder **layout** di folder **res** dalam Project: Tampilan Android.



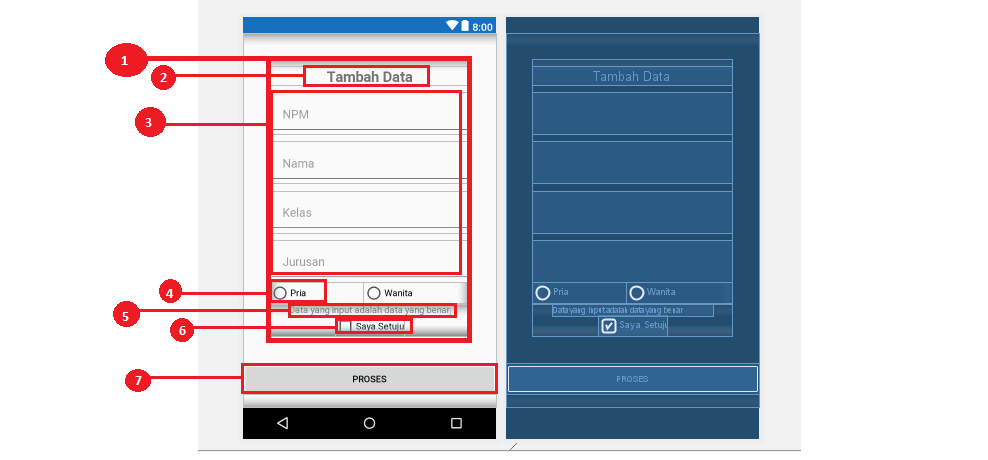
## Tampilan Layout

Beberapa grup tampilan ditandai sebagai layout karena grup tampilan tersebut mengelola tampilan anak dalam cara khusus dan umumnya digunakan sebagai grup tampilan akar. Beberapa contoh layout adalah:



* **LinearLayout**: Grup tampilan anak yang diposisikan dan disejajarkan secara horizontal atau secara vertikal.
* **RelativeLayout**: Grup tampilan anak yang setiap tampilannya diposisikan dan disejajarkan relatif terhadap tampilan dalam grup tampilan. Dengan kata lain, posisi tampilan anak bisa dijelaskan dalam hubungan satu sama lain atau dengan grup tampilan induk.
* **ConstraintLayout**: Grup tampilan anak yang menggunakan titik jangkar, tepi, dan panduan untuk mengontrol cara memosisikan tampilan relatif terhadap elemen lain di layout. ConstraintLayout didesain untuk mempermudah saat menyeret dan melepaskan tampilan di editor layout.
* **TableLayout**: Grup tampilan anak yang disusun ke dalam baris dan kolom.
* **AbsoluteLayout**: Grup yang memungkinkan Anda menetapkan lokasi pasti (koordinat x/y) tampilan anaknya. Layout mutlak bersifat kurang fleksibel dan lebih sulit dikelola daripada tipe layout lainnya tanpa pemosisian mutlak.
* **FrameLayout**: Grup tampilan anak bertumpuk. FrameLayout didesain untuk memblokir area di layar guna menampilkan satu tampilan. Tampilan anak digambar bertumpuk, dengan anak yang baru saja ditambahkan di atas. Ukuran FrameLayout adalah ukuran tampilan anak terbesarnya.
* **GridLayout**: Grup yang menempatkan layar anaknya dalam kotak persegi panjang yang bisa digulir.

Contoh tampilan design + blueprint layout dengan nama file dialog\_layout yang berisikan komponen komponen interface seperti EditText, TextView, Button, RadioButton, dan CheckBox menggunakan LinearLayout



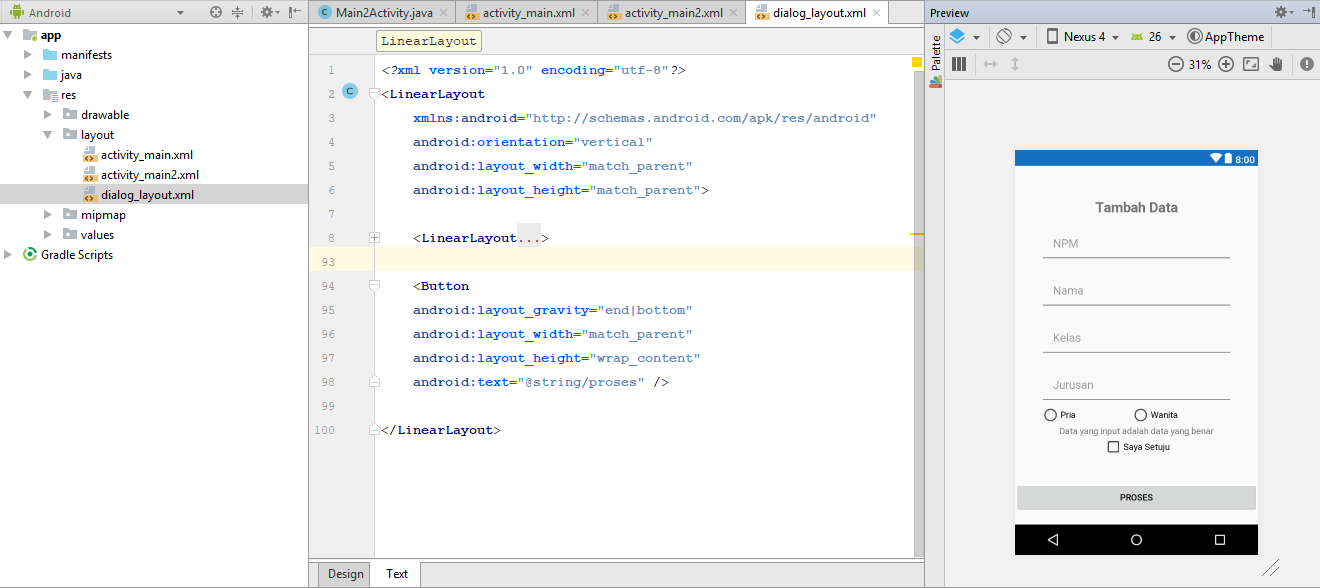
Keterangan pada gambar diatas :

1. LinearLayout, yang berisi komponen komponen UI dengan setelan tampilan vertikal
2. TextView, yang memiliki text “TAMBAH DATA” dengan setelah text = bold dan alignment = center
3. EditText, yang merupakan field untuk menginput data dengan hint sesuai masing- masing inputan
4. RadioButton, yang memiliki value pria dan wanita
5. TextView, dengan text “data yang diinput adalah data yang benar”
6. CheckBox, dengan text “saya setuju”
7. Button, yang memiliki text “proses”

## Menggunakan XML

Terkadang lebih cepat dan lebih mudah mengedit kode XML secara langsung, terutama saat menyalin dan menempelkan kode untuk tampilan serupa.

Untuk menampilkan dan mengedit kode XML, buka file layout XML. Editor layout muncul bersama tab Design di bagian bawah yang disorot. Klik tab Text untuk melihat kode XML. Berikut contoh tampilan kode xml yang sama dengan gambar sebelumnya :



## Basic Component XML Android

* TextView

TextView merupakan komponen android yang digunakan untuk menampilkan tulisan berupa teks dan angka.

* Button

Button adalah komponen yang digunakan dalam pembuatan aplikasi android yang berfungsi untuk menjalankan setiap aksi seperti input, delete, update dll.

* Radio Button

Radio Button adalah komponen yang diperuntunkan untuk memilih salah satu nilai saja diantara 2 komponen. Setiap RadioButton memiliki nilainya masing masing. Contoh penggunaannya seperti pemilihan jenis kelamin dan persetujuan aplikasi.

* Check Box

CheckBox adalah komponen yang memberikan pilihan kepada pengguna lebih dari satu pilihan.

## Material Design

Material Design adalah panduan komprehensif untuk desain visual, gerak, dan interaksi lintas platform dan perangkat. Android kini menyertakan dukungan untuk aplikasi desain bahan. Untuk menggunakan desain bahan di aplikasi Android, mengikuti panduan yang didefinisikan dalam spesifikasi Material Design dan menggunakan komponen dan fungsionalitas baru yang tersedia di Android 5.0 (API level 21) ke atas.

Android menyediakan elemen berikut untuk membangun aplikasi menggunakan Material Design:

* Tema baru
* Widget baru untuk tampilan yang kompleks
* API baru untuk animasi dan bayangan custom

# Bab 2 : Multimedia Interface/Read Data From Sdcard With Json

**2**

## JSON (JavaScript Object Notation)

## Audio

## Video

## Image

# Bab 3 : Database Programming (SQLite)

**3**

## SQLite

## Create & Close a connection

## Create a database & table

## Insert data to database

## Read data from database

## Update data from database

## Delete data from database

## Delete database & table

## Simple project

# Bab 4 : Google Maps API v2.0

**4**

## Basic Viewport

## Current Location

## Custom Icons

## Distance of 2 Object

## Acceleration/Velocity

## Time Calculations

# Bab 5 : Library Cryptography

**5**

## Cryptography

## Encryption

## Decription

## MD5

## AES

# Bab 6 : Network Programming (Socket Programming)

**6**

## Datagram Socket

## Single Client

## Broadcast to Multiple Client

## TCP Socket

## Single Client

## Multiple Client

## Web Server

## Simple Web Browser

## Simple Web Server

# Bab 7 : Internet of Things (IoT)

**7**

## Sensor Signals

## Input Signal

## Output Signal

## Working with Wireless Sensor Network (WSN)

## Client-Server

## Web Based

## Working with Database (SQL)

# Bab 8 : Wireless Sensor Network

**8**

## Arsitektur WSN

## Node WSN

## Topologi Wsan (WSAN - Wireless Sensor Actuator Networks)

## Security

## Kompresi