

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

Menurut Kurniadi (2013:9) mendefinisikan bahwa “program adalah semacam sebuah metode penyelesaian masalah yang disediakan dalam komputer untuk digunakan oleh manusia dalam mempermudah pekerjaannya / kegiatannya. Pembuatan program adalah dengan memberi “perintah” sedemikian rupa kepada komputer sehingga *terciptalah*”.

Program harus ditulis dalam suatu bahasa yang dimengerti oleh komputer, bahasa pemrograman dibedakan menjadi 2, antara lain :

1. Bahasa tingkat rendah (*low level language*): bahasa yang berorientasi ke mesin.
2. Bahasa tingkat tinggi (*high level language*): bahasa yang berorientasi ke manusia (seperti bahasa Inggris) contoh bahasa Pascal, bahasa C, dll.

Program yang ditulis dalam bahasa pemrograman akan diterjemahkan ke dalam bahasa mesin (dikenal dengan biner digit) dengan menggunakan penterjemah, yaitu :

1. *Interpreter*: menterjemahkan baris per baris intruksi. Contoh bahasa *Basic*.
2. *Compiler*: menterjemahkan setelah seluruh intruksi ditulis. Contoh bahasa *Pascal*, *C*, *C++*, dll.

2.2. Java

Menurut Kadir (2013:15) mendefinisikan bahwa:

Java adalah nama sebuah Bahasa pemrograman yang sangat terkenal. Sebagai Bahasa pemrograman, Java dapat digunakan untuk menulis program. Sebagaimana diketahui, program adalah kumpulan instruksi yang ditujukan untuk komputer. Melalui program, komputer dapat diatur agar melaksanakan tugas tertentu sesuai yang ditentukan oleh *programmer* (orang yang membuat program).

Bahasa Java dikembangkan di *Sun Microsystems* dan mulai diperkenalkan kepada publik pada tahun 1995. Seperti halnya C++, Java juga merupakan bahasa yang berorientasi objek. Dengan demikian, Java juga memudahkan dalam pembuatan aplikasi yang berskala besar.

Sebagai bahasa yang beraras tinggi, yang menggunakan perintah-perintah yang mudah dimengerti oleh orang, Java mempunyai keunggulan yakni bersifat *universal*. Sebagai bahasa yang *universal*, Java bisa dijumpai diberbagai *platform* (Linux, UNIX, Windows, Mac, dan lain-lain). Artinya jika anda menguasai Java di *platform* PC, sangat mudah untuk berpindah di linux ataupun sistem operasi lain. Hal yang menarik lagi, hasil kompilasi Java yang dinamakan *bytecode* dapat dijalankan di berbagai *platform* sepanjang di sistem target memiliki *Java Runtime Environment* (JRE).

2.3. Android

Menurut Safaat H (2015:1) mendefinisikan bahwa:

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google Inc.* membeli *Android Inc.* yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, di bentuklah *Open Handset Alliance*, konsorium dari 34 perusahaan perantikeras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

2.3.1. Android : *Platform* Masa Depan

Android dipuji sebagai *Platform mobile* pertama yang lengkap, terbuka, dan bebas, berikut alasannya :

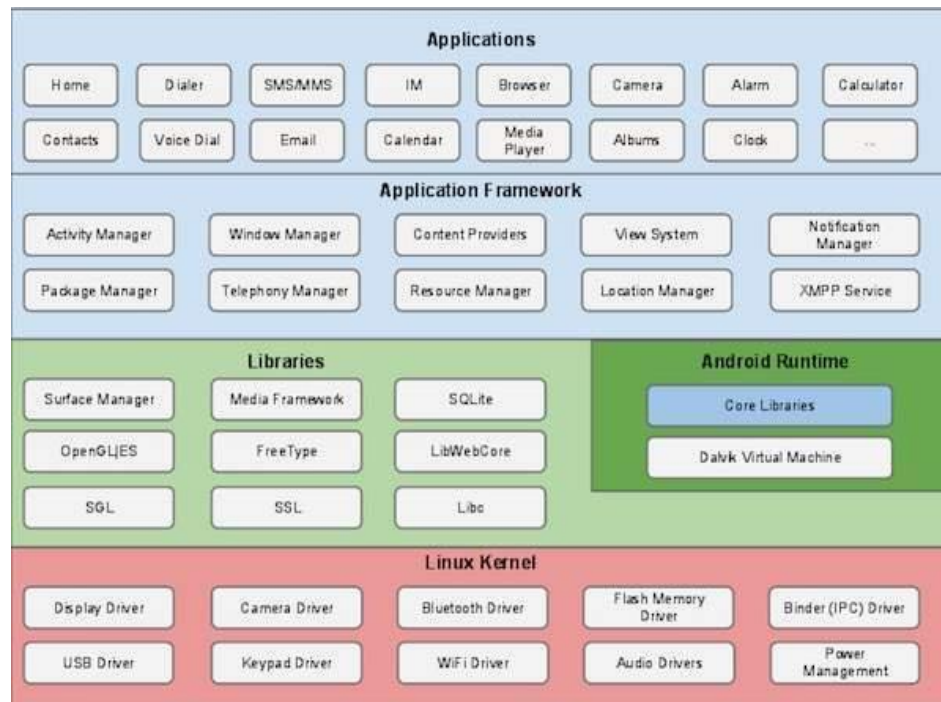
1. Lengkap (*Complete Platform*): Para desainer dapat melakukan pendekatan komprehensif ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.
2. Terbuka (*Open Source Platform*): *Platform* Android disediakan melalui lisensi *open source*. Pengembangan dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6.

3. Bebas (*Free Platform*): Android adalah *platform/aplikasi* yang bebas untuk *developer*. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Aplikasi untuk Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari Android dilisensikan di bawah GNU, *General Public* Lisensi Versi 2 (GPLv2), yang sering dikenal dengan “*copyleft*” lisensi dimana setiap perbaikan pihak ketiga harus jatuh di bawah *terms*. Android didistribusikan di bawah lisensi *Apache Software* (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (produsen handset khususnya) dapat memilih untuk meningkatkan *platform* tanpa harus memberikan perbaikan mereka kemasyarakat *open source*. Sebaiknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka di bawah lisensi apapun yang mereka inginkan. Pengembang aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan.

2.3.2. Arsitektur Android

Secara garis besar arsitektur Android dapat digambarkan dan dijelaskan sebagai berikut:



Sumber: www.tutorialspoint.com/android/images/architecture.jpg

Gambar II.1

Arsitektur Android

1. *Application dan Widgets*

Application dan Widgets ini adalah *layer* di mana berhubungan dengan aplikasi saja, dimana kita biasanya *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

2. *Application Frameworks*

Android adalah “*Open Development Platform*” yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk

mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status *notifications*, dan sebagainya. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*).

Sehingga bisa kita simpulkan *Applications Frameworks* ini adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada layer ini aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa sms dan panggilan telepon.

Komponen-komponen yang termasuk di dalam *Applications Frameworks* adalah sebagai berikut:

- a. *Views*
- b. *Content Provider*
- c. *Resource Manager*
- d. *Notification Manager*
- e. *Activity Manager*

3. *Libraries*

Libraries ini adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan diatas karnel, layer ini meliputi berbagai library C/C++ inti seperti *Libc* dan *SSL*, serta:

- a. *Libraries* media untuk pemutaran media audio dan video.
- b. *Libraries* untuk manajemen tampilan.
- c. *Libraries graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
- d. *Libraries SQLite* untuk dukungan *database*.
- e. *Libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*.
- f. *Libraries LiveWebcore* mencakup modern *web browser* dengan *engine embeded web view*.
- g. *Libraries 3D* yang mencakup implementasi OpenGL ES 1.0 API's.

4. *Android Runtime*

Layer yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakan implementasi linux. *Dalvik Virtual Machine*(DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android, didalam *Android runtime* dibagi menjadi dua bagian yaitu:

- a. *Core libraries*: Aplikasi Android dibangun dalam bahasa Java, sementara Dalvik sebagai virtual mesinnya bukan *Virtual machine* Java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh *Core libraries*.
- b. *Dalvik Virtual Machine*: Virtual mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux karnel untuk melakukan *threading* dan manajemen tingkat rendah.

5. Linux Kernel

Linux kernel adalah *layer* dimana inti dari sistem operasi Android itu berada. Berisi file-file system yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi Android lainnya. Linux kernel yang digunakan Android adalah linux kernel release 2.6.

2.3.3. Komponen Aplikasi

Aplikasi Android ditulis dalam bahasa pemrograman Java. Kode Java dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, di mana prosesnya di *package* oleh *tools* yang dinamakan “*apt tools*” kedalam paket Android sehingga menghasilkan *file* dengan ekstensi apk. File apk itulah yang kita sebut dengan aplikasi, dan nantinya dapat di *install* di perangkat mobile.

Ada empat jenis komponen pada aplikasi Android yaitu :

1. Activity

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi Android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain kita dapat melakukannya dengan *event*, misalnya *click* tombol, memilih opsi atau menggunakan *triggers* tertentu. Secara hirarki sebuah *windows activity* dinyatakan dengan *method*

Activity.setContentView(). *ContentView* adalah objek yang berada pada *root* hirarki.

2. *Service*

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Misalnya, *media player* sedang memutar lagu dari list yang ada, Aplikasi ini akan memiliki dua atau lebih *activity* yang memungkinkan *user* untuk memilih lagu misalnya, atau menulis *sms* sambil *player* sedang jalan. Untuk menjaga musik tetap dijalankan, *activity player* dapat menjalankan *service*. *Service* dijalankan pada *thread* utama dari proses aplikasi.

3. *Broadcast Receiver*

Broadcast Receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh *broadcast* seperti notifikasi zona waktu berubah, baterai *low*, gambar telah selesai diambil oleh kamera, menginisiasi *broadcast* misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah di unduh ke perangkat dan siap untuk digunakan.

Broadcast receiver tidak memiliki *user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang mereka terima, atau mungkin menggunakan *Notification manager* untuk memberitahu kepada pengguna, seperti lampu latar atau *vibrating* (getaran) perangkat, dan lain sebagainya.

4. *Content Provider*

Content Provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam *file* sistem seperti *database* SQLite. *Content Provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta (*Map*), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi *content provider*.

2.3.4. **Versi Android**

1. Versi 1.0. *Astro (Alpha)*

Sebelum Android memberikan nama-nama kudapan sebagai nama untuk versi OS nya, Android sempat memiliki 2 versi awal dengan nama *Android Alpha* dan *Beta*. Nama untuk versi pertama ini sendiri sebenarnya adalah *Android Astro*, namun karena alasan hak cipta (*trademark*), nama ini tidak jadi digunakan. Di versi awal ini belum ada perangkat dengan system operasi Android yang dijual secara komersil.

2. Versi 1.1 *Bender (Beta)*

Versi ini dirilis pada tanggal 5 November 2007, sama seperti versi awalnya, nama *Bender* juga tidak jadi digunakan karena alasan hak cipta (*trademark*). Kemudian lahirlah telepon seluler pertama dengan sistem operasi Android yang dijual secara komersial yakni HTC Dream.



Sumber: [techijau.com/wp-](http://techijau.com/wp-content/uploads/2015/09/hape_android_pertama_htc_dream.jpg)

[content/uploads/2015/09/hape_android_pertama_htc_dream.jpg](http://techijau.com/wp-content/uploads/2015/09/hape_android_pertama_htc_dream.jpg)

Gambar II.2

HTC Dream

3. Versi 1.5 (*Cupcake*)

Versi Android ini merupakan versi pertama yang menggunakan nama makanan manis sebagai kode nama untuk tiap versi Android yang kemudian tradisi untuk menamai versi Android dengan nama makanan manis masih diteruskan hingga saat ini. Android *Cupcake* dirilis pada tanggal 30 April 2009.



Sumber: techijau.com/wp-content/uploads/2015/09/android_cupcake.jpg

Gambar II.3

Android 1.5 (*Cupcake*)

4. Versi 1.6 (*Donut*)

Versi Android ini dirilis tidak sampai setahun setelah perilisan Android *Cupcake*, yakni pada tanggal 15 September 2009. Versi ini dihadirkan untuk menutupi bug pada versi sebelumnya, sekaligus untuk penambahan beberapa fitur seperti misalnya dukungan untuk perangkat dengan ukuran layar yang lebih besar.



Sumber: techijau.com/wp-content/uploads/2015/09/android_donut.jpg

Gambar II.4

Android 1.6 (*Donut*)

5. Versi 2.0 - 2.1 (*Eclair*)

Versi Android ini juga dirilis tidak sampai satu tahun setelah kedua versi sebelumnya dirilis, yakni pada tanggal 26 Oktober 2009. Mereka masih berfokus untuk menutupi bug yang ada dan juga menambahkan beberapa fitur seperti *Bluetooth*, *flash* pada kamera, fitur *digital zoom* pada kamera, *multi-touch*, *live wallpaper*, dan lainnya. Hadirnya perangkat seri Nexus dari Google yang pertama kali muncul yakni HTC Nexus One juga menggunakan versi OS Android *Éclair*.



Sumber: techijau.com/wp-content/uploads/2015/09/android_eclair.jpg

Gambar II.5

Android 2.0 - 2.1 (*Eclair*)

6. Versi 2.2 (*Froyo*)

Android versi 2.2 dengan julukan *froyo*, singkatan *Frozen Yogurt* dirilis pada tanggal 20 Mei 2010. Pada versi ini kecepatan kinerja dan aplikasi naik 2 sampai 5 kali lebih cepat. Dan versi ini, ponsel Android memiliki kemampuan untuk menjadi *Hotspot Wi-Fi*.



Sumber: techijau.com/wp-content/uploads/2015/09/android_froyo.jpg

Gambar II.6

Android 2.2 (*Froyo*)

7. Versi 2.3 (*Gingerbread*)

Dirilis pada tanggal 6 Desember 2010 bersamaan dengan diadikannya Nexus S yang merupakan perangkat *smartphone* seri Nexus yang diproduksi oleh Samsung. Versi OS ini juga mengawali kesuksesan Android di jagad *smartphone* meski masih kalah populer dengan BlackBerry OS. Beberapa vendor mulai serius untuk menggarap perangkat OS Android.

Pada saat itu, Samsung dengan Galaxy seriesnya berperan besar dalam kesuksesan Android. Promosi yang luar biasa gencarnya membuat orang awam mulai mengenal sistem operasi Android. Bahkan saat itu sebagian besar orang beranggapan bahwa OS Android adalah milik Samsung karena kuatnya *branding* yang dilakukan oleh Samsung. Ini juga menjadi awal mula terkenalnya Samsung di jagad *smartphone*.



Sumber: techijau.com/wp-content/uploads/2015/09/android_gingerbread.jpg

Gambar II.7

Android 2.3 (*Gingerbread*)

8. Versi 3.0 (*Honeycomb*)

Versi Android ini dirilis pada tanggal 10 Mei 2011 dan dirancang khusus untuk perangkat *tablet*, yang saat itu mulai populer di pasaran, salah satunya berkat promosi Samsung dan juga kepopuleran *Apple iPad*.



Sumber: techijau.com/wp-content/uploads/2015/09/android_honeycomb.jpg

Gambar II.8

Android 3.0 (*Honeycomb*)

9. Versi 4.0 (*Ice Cream Sandwich*)

Versi Android ini dirilis pada tanggal 16 Desember 2011. Bisa dibilang merupakan Android *Honeycomb* yang disempurnakan, dan dioptimalkan untuk pengguna baik *smartphone* maupun *tablet*. Perubahan yang paling terlihat dari versi ini dibandingkan dengan versi sebelumnya adalah dari segi *User Interface* yang terlihat lebih bersih dan elegan. Versi ini juga lebih dioptimalkan untuk urusan *multitasking*.

Bersamaan dengan diperkenalkannya Android ICS, Google juga memperkenalkan perangkat Galaxy Nexus yang merupakan seri *smartphone* Nexus yang diproduksi oleh Samsung. Setelah versi ini, Google kemudian

secara rutin memperkenalkan perangkat seri Nexus pada tiap kali mereka memperkenalkan versi Android terbaru.



Sumber: [techijau.com/wp-](http://techijau.com/wp-content/uploads/2015/09/android_ice_cream_sandwich.jpg)

[content/uploads/2015/09/android_ice_cream_sandwich.jpg](http://techijau.com/wp-content/uploads/2015/09/android_ice_cream_sandwich.jpg)

Gambar II.9 Android 4.0 (*Ice Cream Sandwich*)

10. Versi 4.1 (*Jelly Bean*)

Versi Android ini dirilis pada 9 Juli 2012. Bersamaan dengan diperkenalkannya versi OS 4.1 pada 27 Juni 2012, Google juga memperkenalkan Nexus 7 yang diproduksi oleh ASUS. Nexus 7 (gen 1) merupakan seri Nexus pertama yang merupakan perangkat *tablet*. *Jelly Bean* mengalami 3x update versi yakni 4.1, 4.2 hingga 4.3. Selanjutnya mereka memperkenalkan Android 4.2 bersamaan dengan dihadirkannya Nexus 4, *smartphone* yang diproduksi oleh LG Plus Nexus 10, perangkat *tablet* yang diproduksi oleh Samsung.

Pada saat versi 4.3 dirilis, Google juga merilis Nexus 7 generasi 2 yang masih diproduksi oleh ASUS yang mana ia memiliki beberapa peningkatan seperti misalnya penambahan kamera belakang serta dukungan untuk konektivitas *internet*.



Sumber: techijau.com/wp-content/uploads/2015/09/android_jelly_bean.jpg

Gambar II.10

Android 4.1 (*Jelly Bean*)

11. Versi 4.4 (*Kitkat*)

Nama *Kitkat* diambil dari sebuah produk cemilan wafer berlapis cokelat yang dimiliki oleh *Nestle*. Sebelum Android versi “K” ini disebut-sebut sebagai *Key Lime Pie*, namun atas beberapa pertimbangan akhirnya Google lebih memilih untuk memberi nama *Kitkat*. Ceritanya, *Kitkat* adalah salah satu cemilan yang tersedia di dapur kantor yang biasanya juga menemani pada *programmer* Google. Hingga seseorang berkata “hei, kenapa kita tidak menamainya sebagai *Kitkat*?”.

Sesaat setelah ide itu muncul, Google segera menghubungi pihak *Nestle* sebagai pemilik merk dagang *Kitkat* dan mereka menyetujui pemberian nama *Kitkat* untuk versi Android “K”. Karyawan Google sendiri tidak mengetahui bahwa Android 4.4 akan diberi nama *Kitkat* karena yang mereka tau versi Android “K” adalah *Key Lime Pie*. Mereka baru mengetahuinya setelah patung maskot Android *Kitkat* diletakan di kantor pusat Google.

Versi ini diklaim lebih ramah terhadap perangkat dengan spesifikasi seadanya. Bahkan perangkat dengan RAM 512mb masih bisa menjalankan OS versi ini dengan mulus.



Sumber: techijau.com/wp-content/uploads/2015/09/android_kitkat.jpg

Gambar II.11

Android 4.4 (*Kitkat*)

12. Versi 5.0 – 5.1 (*Lollipop*)

Dirilis pada tanggal 15 Oktober 2014, versi OS ini mengusung perubahan besar dari segi *UI* yang terlihat lebih *flat* dengan konsep material desain. Versi Android ini sudah mendukung arsitektur 64-bit sehingga memungkinkan untuk pengguna RAM diatas 3gb pada *hardware* perangkat. Pengguna prosesor 64-bit pun semakin banyak diadopsi oleh para *vendor*, mulai dari penerapan pada perangkat *flagship* hingga perangkat kelas menengah kebawah.

Bersamaan dengan dirilisnya versi ini, Google juga memperkenalkan perangkat *smartphone* Nexus 6 yang diproduksi oleh Motorola, yang mana

saat itu Motorola juga masih menjadi milik Google. Perangkat *smartphone* Android pertama yang menggunakan RAM diatas 3gb adalah ASUS Zenfone 2 yang mengusung kapasitas RAM 4gb.



Sumber: techijau.com/wp-content/uploads/2015/09/android_lollipop.jpg

Gambar II.12

Android 5.0 (*Lollipop*)

13. Versi 6.0 (*Marshmallow*)

Versi Android ini resmi dirilis pada bulan September tahun 2015. Bersamaan dengan dirilisnya versi ini, untuk pertama kalinya Google juga memperkenalkan 2 perangkat *smartphone* Nexus sekaligus yang diproduksi oleh 2 vendor yang berbeda.

Nexus 5X adalah versi *smartphone* Nexus kelas menengah dengan ukuran layar 5.2 inch yang diproduksi oleh LG. Sedangkan yang satunya lagi memiliki bentang layar yang lebih lebar yakni 5.7 inch yang diberi nama Nexus 6P yang merupakan *smartphone flagship* hasil kerjasama Google dengan Huawei.



Sumber: techijau.com/wp-content/uploads/2015/09/android_marshmallow.jpg

Gambar II.13

Android 6.0 (*Marshmallow*)

14. Versi 7.0 (*Nougat*)

Resmi diperkenalkan pada akhir Juni 2016. Banyak netizen yang berspekulasi bahwa kemungkinan besar, pemberian nama untuk Android versi “N” ini adalah *Nuttela*. Namun Google menepis kabar tersebut setelah resmi memperkenalkannya bersamaan dengan dipamerkannya patung icon Android yang berdiri diatas potongan *Nougat*.



Sumber: techijau.com/wp-content/uploads/2016/07/android_nougat.jpg

Gambar II.14

Android 7.0 (*Nougat*)

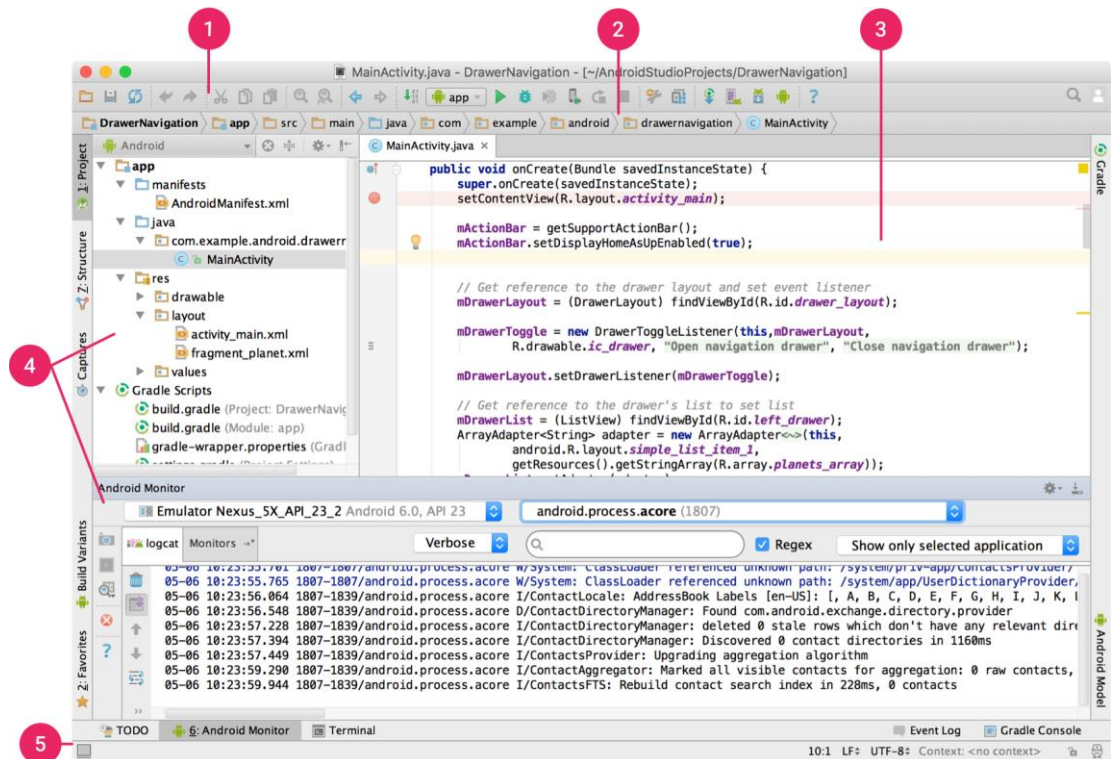
2.4. Peralatan Pendukung (*Tools System*)

2.4.1. Android Studio

Android Studio adalah (<https://developer.android.com/studio/intro/index.html>: 2016) Lingkungan Pengembangan Terpadu – *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan *IntelliJ IDEA*. Selain merupakan editor kode *IntelliJ* dan alat pengembangan yang berdaya guna.

Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas anda saat membuat aplikasi Android, misalnya system pembuatan berbasis *Gradle* yang fleksible, emulator yang cepat dan kaya fitur, lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android, *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru, template kode dan *integrasi GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh, alat penguji dan kerangka kerja yang ekstensif, alat *Lint* untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain, serta dukungan bawaan untuk *Google Cloud Platform*, untuk mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*.

Android Studio memiliki tampilan antarmuka pengguna / *user interface* seperti gambar di bawah ini.



Sumber: developer.android.com/studio/images/intro/main-window_2-1_2x.png?hl=id

Gambar II.15

User Interface / Antarmuka Pengguna Android Studio

Berikut ini penjelasan dari gambar *user interface/ antarmuka pengguna* diatas:

1. Bilah alat yang memungkinkan anda untuk melakukan berbagai jenis tindakan, termasuk menjalankan aplikasi dan meluncurkan alat Android.
2. Bilah navigasi yang akan membantu anda menavigasi di antara proyek dan file yang dibuka untuk pengeditan. Di sini tampilan struktur yang terlihat taplak lebih ringkas daripada jendela *Project*.
3. Jendela editor yang merupakan tempat anda membuat dan mengubah kode. Bergantung pada jenis file saat ini, editor dapat berubah. Misalnya, ketika melihat *file* tata letak, *editor* akan menampilkan *Layout Editor*.

4. Jendela alat yang memberi anda akses ke tugas-tugas spesifik seperti pengelolaan proyek, penelusuran, kontrol versi, dan banyak lagi. Anda bias meluaskan dan juga menciutkannya.
5. Bilah status yang menampilkan status proyek anda dan IDE itu sendiri, serta setiap peringatan atau pesan.

Anda bisa menata jendela utama untuk memberi anda ruang layar yang lebih luas dengan menyembunyikan atau memindahkan bilah alat dan jendela alat. Anda juga bisa menggunakan pintasan keyboard untuk mengakses sebagian besar fitur IDE.

Anda dapat menelusuri seluruh kode sumber, basis data, tindakan, elemen antarmuka pengguna, dan seterusnya setiap saat dengan menekan tombol *shift* dua kali, atau mengklik kaca pembesar di sudut kanan atas dari jendela Android Studio. Ini akan sangat berguna misalnya saat anda mencoba menemukan tindakan IDE tertentu yang anda lupakan cara memicunya.

2.4.2. Android Software Development Kit (SDK)

Menurut Safaat H (2015:5) mendefinisikan bahwa “Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java”. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di *release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi-netral, Android memberi anda

kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone/Smartphone*. Beberapa fitur-fitur Android yang paling penting adalah:

1. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*.
2. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat *mobile*.
3. *Integrated browser* berdasarkan *engine open source Webkit*.
4. Grafis yang di optimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (*Optional akselerasi hardware*).
5. SQLite untuk penyimpanan data.
6. *Media Support* yang mendukung audio, video, dan gambar (MPEG4, H.264, AAC, AMR, JPG, PNG, GIF), GSM Telephony (tergantung hardware)
7. *Bluetooth*, EDGE, 3G, dan WiFi (tergantung *hardware*).
8. Kamera, GPS, kompas, dan *accelerometer* (tergantung *hardware*).
9. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan plugin untuk IDE.

2.4.3. **Android Development Tools (ADT)**

Menurut Safaat H (2015:5) mendefinisikan bahwa “*Android Development Tools* (ADT) adalah *plugin* yang didesain untuk IDE yang memberikan kita kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE.” Dengan menggunakan ADT untuk akan memudahkan kita dalam membuat aplikasi project Android, membuat GUI aplikasi, dan menambahkan komponen-komponen

menggunakan android SDK melalui. Dengan ADT juga kita dapat melakukan pembuatan package Android (.apk) yang digunakan untuk distribusi aplikasi yang kita rancang.

Mengembangkan aplikasi Android dengan menggunakan ADT sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi Android.

Berikut adalah versi ADT untuk yang sudah dirilis:

1. ADT 12.0.0 (July 2011)
2. ADT 11.0.0 (June 2011)
3. ADT 10.0.1 (March 2011)
4. ADT 10.0.0 (February 2011)
5. ADT 9.0.0 (January 2011)
6. ADT 8.0.1 (December 2010)
7. ADT 8.0.0 (December 2010)
8. ADT 0.9.9 (September 2010)
9. ADT 0.9.8 (September 2010)
10. ADT 0.9.7 (May 2010)
11. ADT 0.9.6 (March 2010)
12. ADT 0.9.5 (December 2009)
13. ADT 0.9.4 (October 2009)

Semakin tinggi platform Android yang kita gunakan, dianjurkan menggunakan ADT yang lebih terbaru, karena biasanya munculnya platform baru akan diikuti oleh munculnya versi ADT yang terbaru.

2.4.4. Android Virtual Device (AVD)

Android Virtual Device adalah (<https://developer.android.com/studio/run/managing-avds.html>:2016) sebuah *emulator* Android, yang memungkinkan kita untuk mendefinisikan karakteristik dari ponsel Android, *tablet*, *Android Wear*, atau perangkat Android TV yang ingin kita simulasikan di Android Studio.

2.4.5. XML

Menurut Tim EMS (2012:15) mendefinisikan bahwa “XML (*eXtensible Markup Language*) merupakan bahasa web turunan dari SGML (*Standard Generalized Markup*) yang ada sebelumnya. XML hampir sama dengan HTML, di mana kedua-duanya diturunkan dari SGML”.

Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa *meta-markup* yang menyediakan format tertentu untuk dokumen-dokumen yang mempunyai data terstruktur. Bahasa *markup* adalah mekanisme untuk mengenal aplikasi yang siap didistribusikan.

2.5. HIPO

Menurut Jogiyanto (2005:787) mengemukakan bahwa:

Metodologi yang dikembangkan dan didukung oleh IBM. HIPO sebenarnya adalah alat dokumentasi program. Akan tetapi sekarang HIPO juga banyak digunakan sebagai alat desain dan teknik dokumentasi dalam siklus perkembangan sistem. HIPO berbasis pada fungsi, yaitu tiap-tiap modul didalam sistem digambarkan oleh fungsi utamanya.

Menurut Jogiyanto (2005:788) mengemukakan bahwa:

HIPO menggunakan tiga macam diagram untuk masing-masing tingkatannya, yaitu sebagai berikut :

1. *Visual Table of Contents (VTOC)*

Diagram ini menggunakan hubungan dari fungsi-fungsi di sistem secara berjenjang.

2. *Overview Diagram*

Diagram ini menggunakan secara garis besar hubungan *input*, *process* dan *output*. Bagian *input* menunjukan *item-item* data yang akan digunakan oleh bagian proses. Bagian proses berisi sejumlah langkah-langkah yang menggambarkan kerja dari fungsi. Bagian output berisi item-item data yang dihasilkan atau dimodifikasi oleh langkah-langkah proses.

3. *Detail Diagram*

Diagram ini merupakan diagram tingkatan yang paling rendah di diagram HIPO. Diagram ini elemen-elemen dasar dari paket yang menggambarkan secara rinci kerja dari fungsi.

2.6. Flowhart

Menurut Kurniadi (2013:161) mendefinisikan bahwa:

Flowchart dalam Bahasa Indonesia diterjemahkan sebagai Diagram Alir. Dari dua kata ini, maka dapat kita bayangkan bahwa *flowchart* itu berbentuk diagram yang bentuknya dapat mengalirkan sesuatu. Contoh sederhananya adalah masalah membuat secangkir kopi. Dalam membuat secangkir kopi, tentu saja diperlukan langkah-langkah yang berurutan agar hasilnya dapat sesuai dengan apa yang kita

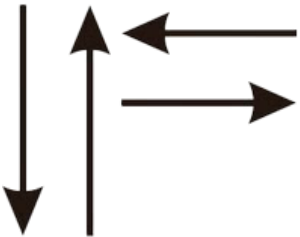
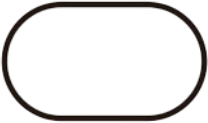
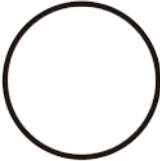
inginkan, yaitu secangkir kopi. Demikian halnya dalam memprogram, diperlukan suatu algoritma (urutan langkah-langkah login penyelesaian masalah yang disusun secara sistematis) agar program yang kita buat dapat berjalan dan memberikan hasil yang valid. Untuk merepresentasikan algoritma itulah kita gunakan *flowchart*.

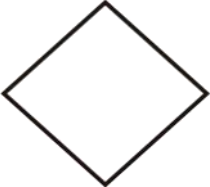
Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

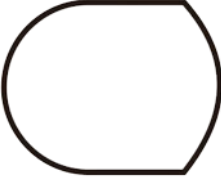
Flowchart biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Tabel II.1

Flowchart

No	Gambar	Deskripsi
1		Flow Direction Symbol , yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga <i>Connecting Line</i>
2		Terminator Symbol , yaitu simbol untuk permulaan (<i>start</i>) atau akhir (<i>end</i>) dari suatu kegiatan
3		Connector Symbol , yaitu simbol untuk keluar - masuk atau penyembungan proses dalam lembar/ halaman yang sama

4		Connector Symbol , yaitu simbol untuk keluar masuk atau penyambungan proses dalam lembar/ halaman yang sama
5		Processing Symbol , yaitu simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer
6		Manual Operation Symbol , yaitu simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer
7		Decision Symbol , yaitu simbol pemilihan proses berdasarkan kondisi yang ada
8		Input-Output Symbol , yaitu simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya
9		Manual Input Symbol , yaitu simbol untuk pemasukan data secara <i>manual on-line keyboard</i>
10		Preparation Symbol , yaitu simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i>

11		Predefine Proses Symbol , yaitu simbol untuk pelaksanaan suatu bagian (<i>sub-program</i>) prosedur
12		Display Symbol , yaitu simbol yang menyatakan peralatan output yang digunakan yaitu layar, <i>plotter</i> , <i>printer</i> , dan sebagainya
13		Disk & On-line Storage Symbol , yaitu simbol yang menyatakan <i>input</i> yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i>
14		Magnetic Tape Unit Symbol , yaitu simbol yang menyatakan <i>input</i> berasal dari pita magnetik atau <i>output</i> disimpan ke pita magnetic
15		Punch Card Symbol , yaitu simbol yang menyatakan bahwa <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu
16		Document Symbol , yaitu simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas

Sumber: <http://www.aryah.id/2014/06/simbol-simbol-flowchart-dan-fungsinya.html>