BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Pada dasarnya Sistem menurut Andri Kristanto (2008:1) yang dimaksud dengan sistem adalah "jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul, bersama-sama untuk melakukan suatu kegiatan atau menyelsaikan suatu sasaran tertentu".

2.1.1. Pengertian Dasar Sistem

Menurut Tata Sutabri (2007: 2) sistem merupakan suatu kumpulan atau himpunan dari unsur-unsur komponen atau variabel yang terorganisir, saling berintegrasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu.

Menurut Indrajit (2001: 2) mengemukakan bahwa sistem mengandung arti kumpulan-kumpulan dari komponen-komponen yang dimiliki unsur keterkaitan antara satu dengan lainnya.

Dari definisi sistem menurut beberapa para ahli tersebut dapat disimpulkan bahwa sistem adalah kumpulan dari komponen-komponen yang saling berhubungan untuk mencapai suatu tujuan.

2.1.2. Elemen Sistem

Suatu sistem, dapat terbentuk dari berbagai macam elemen, elemen-elemen tersebut adalah:

a. Masukan (input)

Masukan *(input)* sistem adalah energi yang di masukan ke dalam sistem. Masukan dapat berupa masukan perawatan *(maintenance input)* dan masukan sinyal *(signal input).* Maintenance input adalah energi yang di masukan supaya sistem tersebut dapat beroprasi. Signal input adalah energi yang di proses untuk didapatkan keluaran.

b. Keluaran (Output)

Keluaran *(output)* merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

c. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna.

d. Mekanisme Pengendalian

Mekanisme pengendalian diwujudkan dengan menggunakan umpan balik *(feedback).* Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

e. Umpan Balik

Umpan balik adalah Masukan atau Pendapat yang diberikan oleh masyarakat mengenai sistem informasi yang kita buat sehingga kita dapat memperbaiki dan meningkatkan kinerja dari sistem informasi yang kita buat tersebut.

2.1.3. Karakteristik Sistem

Menurut Jogiyanto (2005:3) Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen - komponen (*components*), batas (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) , dan sasaran (*objectives*) atau tujuan (*goal*). Berikut adalah penjelasan dari karakteristik sistem :

a. Komponen Sistem (*component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan.

b. Batas sistem (*boundary)*

Batas sistem merupakan daerah yang membatasi antara satu sistem dengan sistem lain atau dengan lingkungan luar.

c. Lingkungan luar sistem (*environment*)

Lingkungan luar sistem dari satu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

d. Penghubung sistem (*interface*)

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lain.

e. Masukan sistem (*input*)

Masukan (*input)* adalah energi yang dimasukan ke dalam sistem. Masukan dapat berupa masukan perawatan *(maintenance input)* dan masukan sinyal *(signal input). Maintenance input* adalah energi yang dimasukan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk mendapatkan keluaran.

f. Keluaran sistem (*output*)

Keluaran *(output)* adalah hasil dari energi yang diolah dan diklarifikasi menjadi keluaran yang berguna dan sisa pembuangan.

g. Pengolahan system (*proses*)

Suatu sistem bias mempunyai suatu bagian pengolah yang akan mengubah masukan menjadi keluaran.

h. Sasaran sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (objektif). Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali, masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.1.4. Klasifikasi Sistem

Sistem dapat diklasifikasikan dalam beberapa sudut pandangan, diantaranya sebagai berikut:

1. Sistem abstrak (*abstract system*) dan sistem fisik (*physical system*).

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik sedangkan sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*).

   Sistem alamiah adalah sistem yang terjadi karena proses alam tidak dibuat oleh manusia (ditentukan dan tunduk kepada kehendak sang pencipta alam). Sedangkan sistem buatan manusia adalah sistem yang dirancang oleh manusia.

3. Sistem deterministik *(deterministic system)* dan sistem probabilistik *(probabilistic system).*

   Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem tertentu relatif stabil atau konstan dalam jangka waktu yang lama. Sedangkan sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem terbuka (*open system*) dan sistem tertutup (*closed system*).

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Sedangkan

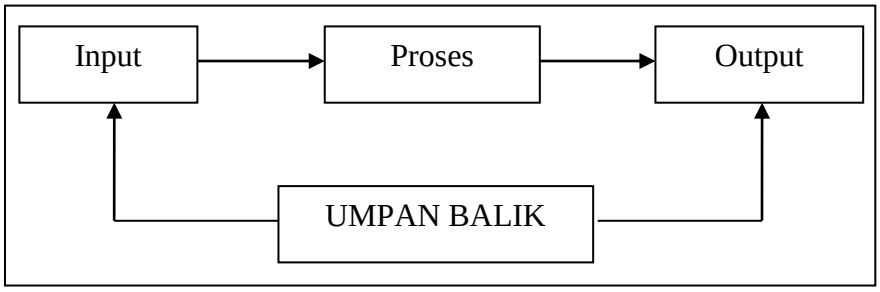
sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lain.

2.2. Konsep Dasar Informasi

Dalam buku Al-bahra (2005:8) Sumber informasi adalah data. Data adalah kenyataan yang menggambarkan kejadian – kejadian dan kesatuan nyata. Kejadian adalah suatu yang terjadi pada saat tertentu. Informasi diperoleh setelah data-data mentah di proses atau diolah.

2.2.1. Pengertian Informasi

Menurut Andri Kristanto (2007 : 10). Data yang masih merupakan bahan mentah apabila tidak diolah maka data tersebut tidak berguna. Data tersebut akan berguna dan menghasilkan suatu informasi apabila diolah melalui suatu model. Model yang digunakan untuk mengolah data tersebut disebut dengan model pengolahan data atau lebih dikenal dengan nama siklus pengolahan data.



Gambar 2.1 Siklus Pengolahan Informasi
(Sumber : Andri Kristanto, 2007 : 10)

Dari gambar diatas dapat dijelaskan bahwa data yang merupakan suatu kejadian yang menggambarkan kenyataan yang terjadi dimasukan melalui elemen input kemudian data tersebut akan diolah dan diproses menjadi suatu output, dan

output tersebut adalah informasi yang dibutuhkan. Informasi tersebut akan diterima oleh pemakai atau penerima, kemudian penerima akan memberikan umpan balik yang berupa evaluasi terhadap informasi tersebut dan hasil umpan balik tersebut akan menjadi data yang akan dimasukan menjadi input kembali. Begitu seterusnya alur pengolahan data.

2.2.2. Kualitas Informasi

Dalam buku Al-Bahra (2005:11) Kualitas informasi sangat dipengaruhi atau ditentukan oleh hal-hall sebagai berikut:

1. Relevan (relevancy), yaitu sejauh mana tingkat relevansi informasi tersebut terhadap kenyataan kejadian masa lalu, masa sekarang dan kejadian yang akan datang.
2. Akurat (accuracy), yaitu suatu informasi dikatakan berkualitas jika seluruh kebutuhan informasi telah tersampaiakan serta pesan yang disampaiakan sudah lengkap sesuai dengan yang diinginkan oleh user.
3. Tepat Waktu (timelines), informasi yang datang pada penerima tidak boleh terlambat. Suatu informasi harus sesuai dengan keadaan saat itu.
4. Ekonomis (economy), informasi yang dihasilkan harus mempunyai daya jual yang tinggi dan biaya operasional yang harus dikeluarkan untuk menghasilkan informasi tersebut harus minimal, informasi tersebut juga mapu memberikan dampak yang luas terhadap laju pertumbuhan ekonomi dan teknologi informasi.

5. Efisien (efficiency), informasi yang berkualitas harus memiliki kalimat yang sederhana dan mudah dimengerti, tapi bisa memberikan makna yang mendalam.
6. Dapat dipercaya (reliability), informasi yang didapat harus dari sumber yang bisa dipercaya.

2.2.3. Nilai Informasi

Dalam buku Al-Bahra (2005:12) Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya dan sebagian besar informasi tidak dapat ditaksir keuntungannya dengan satuan nilai uang, tetapi dapat ditaksir nilai efektivitasnya. Pengukurannya dapat menggunakan alanisis cost effectiveness dan cost benefit.

2.3. Konsep Dasar Sistem Informasi

Menurut Al-bahra (2005:78) Sistem informasi memberikan nilai tambah terhadap proses, produksi, kualitas, manajemen, pengambilan keputusan dan pemecahan masalah serta keunggukan kompetitif yang tentu saja sangat berguna bagi kegiatan bisnis.

2.3.1. Pengertian Sistem Informasi

Dikutip dari buku Al-Bahra (2005:13), sistem informasi dapat di definisikan sebagai berikut:

1. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.

2. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan / atau untuk mengendalikan organisasi.
3. Suatu sistem dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.3.2. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*). Sebagai suatu sistem, blok bangunan tersebut masing-masing berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya. Blok bangunan tersebut terdiri dari :

1. Blok Masukan (*Input Block*)

   Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

   Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Terdiri dari 3 bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).

5. Blok Basis Data (*Database Block*)

   Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management Systems*).

6. Blok Kendali (*Controls Block*)

   Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.4. Pemesanan

Menurut kamus besar bahasa indonesia arti dari Pesan adalah suruhan (perintah, nasihat, permintaan, amanat) yang harus dilakukan atau disampaikan kepada orang lain. Pesanan adalah permintaan hendak membeli (supaya dikirim, dibuatkan dan sebagainya). Pemesanan adalah orang yang memesan (tempat, barang dan sebagainya).

2.5. Pengertian Hotel

Hotel berasal dari bahasa Perancis yaitu hostel artinya "tempat penampungan buat pendatang" atau "bangunan penyedia pondokan dan makanan untuk umum". Oleh sebab itu, keberadaan hostel untuk meladeni kebutuhan masyarakat sebagai tempat tinggal sementara.

Adapun pengertian hotel berdasarkan Keputusan Menteri Pariwisata, Pos dan Telekomunikasi adalah sebagai berikut: "hotel adalah suatu jenis akomodasi yang mempergunakan sebagian atau seluruh bangunannya untuk menyediakan jasa pelayanan, penginapan, makan dan minum serta jasa lainnya bagi umum yang dikelola secara komersial serta memenuhi ketentuan persyaratan yang ditetapkan didalam keputusan pemerintah".

Hotel juga dapat diartikan sebagai tempat / bangunan yang diorganisasikan secara komersial, dengan sistem yang baik dan fasilitas yang memuaskan kepada setiap tamu yang ada.

2.6. Teknologi *Smartphone*

Smartphone (ponsel pintar) merupakan sebuah *high-endmobile phone* di mana memiliki beberapa fungsi seperti *personal digital assistant*, alat telepon, multimedia, dan aplikasi komputer *mobile* yang telah dikombinasikan menjadi satu. *Smartphone* bisa menjalakan aplikasi yang memiliki kompleksitas yang lebih tinggi daripada *phone* biasa.*Smartphone* menyediakan sebuah *host* di mana akan memudahkan para *user* ketika ingin melakukan *update* software dan data *plan* yang sudah disediakan untuk beberapa pengguna tertentu. Secara lebih luas ini disebut sebagai ekosistem dari sistem development sebuah *smartphone* (M. Wargo, John,2010).

2.7. Sejarah Android

Android adalah sebuah *platform* pertama yang betul-betul terbuka dalam pengembangannya dan komprehensif untuk perangkat *mobile*, semua perangkat lunak yang ada difungsikan menjalankan sebuah *device mobile* tanpa memikirkan kendala kepemilikan yang menghambat inovasi pada teknologi *mobile* (Meier, 2008). Dalam definisi lain, android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware*, dan aplikasi inti yang dirilis oleh *Google*.

Sedangkan *Android SDK (Software Development Kit)* menyediakan *Tools* dan *API* yang diperlukan untuk mengembangkan aplikasi pada *platform Android* dengan menggunakan bahasa pemrograman *Java*. *Android* dikembangkan bersama antara *Google, HTC, Intel, Motorola, Quallcomm, T-Mobile, NVIDIA*

yang tergabung dalam *OHA (Open Handset Alliance)* dengan tujuan membuat sebuah standar terbuka untuk perangkat bergerak (*Mobile Device*) (Mulyadi, 2010).

Pada tahun 2005 *Google* mengakuisisi *Android. Inc* yang pada saat itu dimotori oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Yang kemudian pada tahun itu juga memulai membangun platform *Android* secara intensif. Kemudian pada tanggal 12 November 2007 *Google* bersama *Open Handset Alliance* yaitu sebuah konsorsium perangkat mobile terbuka, merilis *Google Android SDK*, setelah mengumumkannya seminggu sebelumnya. Dan sambutannya sangat luar biasa, hampir semua media berita tentang *IT* dan *Programming* memberitakan tentang dirilisnya *Android SDK (Software Development Kit)* ini.

Ini dikarenakan Ide dari *platform Android* ini sangat menarik untuk *developer* dan *programmer* diseluruh penjuru dunia. Disisi lain, produsen pembuat *handset* ponsel juga segera berlomba-lomba membuat *handset* dengan *platform Android* ini.

Google bersama dengan *OHA* merilis paket *software SDK* yang lengkap untuk mengembangkan aplikasi pada perangkat *mobile*. Yaitu :

Sistem Operasi, *Middleware* dan Aplikasi Utama untuk perangkat *mobile*. Sebagai *programmer* atau *developer* bisa melakukan segalanya, mulai dari membuat aplikasi pengiriman *SMS* hanya dengan dua baris kode, hingga mengganti *event* pada *Home Screen* pearangkat *Android*. Selain itu, bahkan dengan mudah kita

bisa membuat dan merubah Sistem Operasiya atau mengganti semua aplikasi default dari *Google* (Mulyadi, 2010).

2.7.1. Fitur Android

Fitur-fitur yang tersedia pada *platform Android* saat ini diantara lain :

1. *Framework* Aplikasi

   Mendukung penggantian komponen dan *reusable component*

2. *Dalvik Virtual Machine*

   Android berjalan diatas *Linux Kernel* dan dioptimalkan untuk perangkat *mobile*

3. *Integrated Browser*

   Dikembangakan berdasarkan *open source engine WebKit*

4. Grafis

   Dioptimalkan dan didukung oleh *library* grafis *2D* yang terkostumisasi, grafis *3D* berdasarkan spesifikasi *openGL ES 1.0* (Opsional akselerasi *hardware*)

5. *SQLite*

   Media untuk penyimpanan data berupa *database*

6. *Media Support*

   Mendukung *audio, video*, dan gambar (*MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF*)

7. *GSM Telephony*

8. *Bluetooth, EDGE, 3G, dan WiFi*

9. Dukungan Perangkat Tambahan

Android dapat memanfaatkan kamera, layar sentuh, *accelerometers*, *magnetometers*, *GPS*, akselerasi *2D* (dengan perangkat orientasi, *scaling*, konversi format *pixel*) dan akselerasi grafis *3D*

10. Multi Touch

Kemampuan layaknya *handset modern* yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat

11. Lingkungan *Development*

Android lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, dan kinerja memori dan *plugin* untuk *IDE*

12. Market

Seperti kebanyakan *handphone* yang memiliki tempat penjualan aplikasi, *market* pada *android* merupakan katalog aplikasi yang dapat diunduh dan diinstal pada *handphone* melalui *internet.* (Mulyadi, 2010)

2.7.2. Arsitektur Android

1. Linux *Kernel*

   Android bukan *linux* akan tetapi *android* dibangun diatas *Linux Kernel* yatitu versi 2.6 sehingga kehandalannya bisa dipercaya. Untuk inti sistem *service linux* yang digunakan sperti keamanan, manajemen memori, proses manajemen, *network* dan *driver* model. Seperti yang terlihat digambar, *linux kernel* menyediakan *Driver* untuk Layar, Kamera, *Keypad*, *Wifi*, *Flash Memory*, *Audio*, dan *IPC (Interprocess Communication)* untuk mengatur aplikasi dan keamanan. *Kernel* juga

bertindak sebagai lapisan abstrak antara *hardware* dan *software stacknya*.

2. *Libraries*

   Android menyertakan *library C/C++* yang digunakan oleh berbagai komponen dari sistem *Android*. Kemampuan ini disediakan kepada *developer* aplikasi melalui *framework* aplikasi *android*. Beberapa inti *library* tercantum di bawah ini:

   a. *System C Library*

      Variasi dari implementasi *BSD*. Berasal pelaksanaan sistem standar *C library (libc),* sesuai untuk perangkatan *embedded* berbasis *linux*.

   b. *Media Libraries*

      Paket video berdasarkan *OpenCORE*. *Library* ini mendukung pemutaran rekaman dan populer ke banyak format audio dan video, serta file gambar, termasuk *MPEG4, H.264, MP3, AAC,AMR, JPG* dan *PNG.*

   c. *Surface Manager*

      Mengelola akses ke subsistem layar, lapisan komposit *2D* dan grafis *3D* dari beberapa aplikasi.

   d. *Library Web Core*

      Mesin *web* modern yang *powerfull* yang baik untuk *browser Android* dengan *embedded web.*

   e. *SGL*

      Untuk mesin grafis *2D*

f. 3D Libraries

Library 3D berdasarkan *openGL ES 1.0 API*, *Library* baik untuk digunakan pada perangkat keras dengan akselerasi *3D* atau yang disertakan, sangat optimal pada *3D software rasterizer*.

g. Free Type

Bitmap dan vektor *font rendering*

h. SQLite

Mesin *database* yang kuat dan ringan dan penghubung tersedia untuk semua aplikasi.

2.7.3 Android *Runtime*

Android terdiri dari satu set *core libraries* yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat pada *core library* bahasa pemrograman *Java*. Setiap aplikasi menjalankan prosesnya sendiri dalam *Android*, dengan masing-masing instan dari mesing *Dalvik Virtual Machine*. *Dalvik* dirancang agar perangkat dapat menjalankan *multiple Virtual Machine* secara efisien. Mesin *Virtual Dalvik* mengeksekusi file dalam *Dalvik Executable (.dex)*, sebuah format yang dioptimalkan untuk memori yang kecil. *Dalvik VM* berbasis, berjalan dan dikompilasi oleh kompiler bahasa *Java* yang telah ditransformasi ke dalam *.dex* format oleh tool *"dx"* yang telah disertakan olehn *Dalvik VM* bergantung pada *kernel Linux* untuk berfungsi, seperti *threading* dan manajemen memori tingkat rendahanya.

1. *Framework* Aplikasi

   Pengembang memiliki akses penuh menuju *API framework* yang sama dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (*reuse*) dengan mudah. Setiap aplikasi dapat memanfaatkan kemampuan inti (sesuai dengan batasan kemampuan yang didefinisikan oleh *framework*). Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna. Semua aplikasi merupakan rangakaian set layanan dan sistem termasuk :

   a. Views

   Komponen *user interface* yang dapat digunakan untuk membangun aplikasi termasuk *list, grids,* kotak teks, tombol dan bahkan sebuah *embedded web.*

   b. Content Provider

   Memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak) atau membagi data tersebut.

   c. Resources Manager

   Menyediakan akses ke kode non sumber lokal seperti *string*, gambar, dan tata letak *file.*

   d. Notification Manager

   Memungkinkan semua aplikasi menampilkan alert yang bisa dikostumisasi di dalam status bar.

e. Activity Manager

Mengelola siklus hidup aplikasi dan menyediakan navigasi umum *backstack.*

f. Aplikasi

Android telah menyertakan aplikasi inti seperti *email client, SMS,* kalender, peta, *browser,* kontak. Dan lain-lain. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemrograman *Java*. Pada layer inilah *developer* atau kita menempatkan aplikasi yang dibuat. Yang istimewa adalah pada *android* semua aplikasi baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan pada *layer* aplikasi dengan menggunakan *library API*yang sama. Ini berarti semua aplikasi yang dibuat untuk *android* akan memiliki akses yang setara dalam mengakses seluruh kemampuan *handset,* tanpa membedakan apakah itu merupakan aplikasi inti atau pihak ketiga. Dalam kata lain dengan *platform android* ini, *programmer* atau *developer* secara penuh akan bias mengkostumisasi perangkat *android*nya. (Mulyadi, 2010)

2.7.4 Aplikasi Android

Aplikasi *Android* ditulis dalam bahasa pemrograman *Java,* yaitu kode *Java* yang terkompilasi bersama-sama dengan data dan *file-file* sumber yang dibutuhkan oleh aplikasi yang digabungkan oleh *aapt tools* menjadi paket aplikasi *android,* sebuah *file* yang ditandai dengan akhiran *.apk*. *File* inilah yang didistribusikan sebagai aplikasi dan diinstal pada *handset android*. *File* ini

diunduh oleh pengguna ke perangkat *mobile* mereka. Semua kode dijadikan satu file *.apk,* dan kemudian kita sebut sebagai sebuah aplikasi.

Dalam banyak cara, masing-masing aplikasi *android* aktif dalam lingkungannya sendiri :

1. Secara *default* semua aplikasi berjalan sendiri dalam proses *Linux*, android memulai proses ketika salah satu dari kode aplikasi harus dijalankan, dan menutup proses saat tidak lagi diperlukan dan sumber daya sistem tersedia untuk aplikasi lain.
2. Setiap proses memiliki *virtual machine (VM)* sendiri, maka kode aplikasi yang dijalankan di isolasi dari semua kode aplikasi lain.
3. Secara *default*, setiap aplikasi diberikan sebuah *user ID Linux* yang unik. Perizinan (*Permissions*) diatur agar aplikasi dari *file* tersebut terlihat hanya untuk user yang memiliki izin untuk aplikasi itu sendiri.

Sangat memungkinkan mengatur dua aplikasi yang sama untuk berbagi *user ID* dalam hal ini masing-masing mereka akan dapat melihat file mereka satu sama lain. Untuk menjaga sumber daya sistem, aplikasi dengan *ID* yang sama juga dapat diatur untuk menjalankan proses *Linux* yang sama dan berbagi *Virtual Machine* yang sama.

2.7.5 Komponen Aplikasi

Salah satu fitur utama dari *android* adalah satu aplikasi dapat menggunakan elemen dari aplikasi lain (yang disediakan berdasarkan izin aplikasi itu). Sistem aplikasi bekerja ketika salah satu bagian aplikasi harus dijalankan dengan menginisiasi objek *Java* pada bagian tersebut. Oleh karna itu, tidak seperti

kebanyakan aplikasi lainnya, pada sistem aplikasi *android* tidak ada satu entry point bagi semua yang ada dalam aplikasi (Misalnya : tidak ada fungsi *main()*). Sebaliknya, mereka mempunya komponen penting dalam sistem yang dapat diinisiasi dan dijalankan sesuai dengan kebutuhan. Ada empat jenis komponen, yaitu :

1. *Activity*

   Sebuah *Activity* normalnya akan menyajikan *User Interface* bagi penggunannya, dimana pengguna kemudian bisa melakukan interaksi padanya. Misalnya, sebuah *activity* dapat memberikan daftar menu kepada pengguna dan kemudian pengguna dapat memilih atau menampilkan foto. Sebuah aplikasi pesan teks mungkin memiliki satu *activity* kedua untuk menulis pesan pada kontak yang dipilih, dan *activity* lainnya untuk memeriksa pesan-pesan lama atau mengubah pengaturan. Walaupun mereka terlihat sebagai sebuah aplikasi, tetapi masing-masing *activity* tersebut adalah mandiri. Masing-masing diimplementasikan sebagai subkelas (*subclass*) dari kelas induk (*base class*) yang bernama *Activity*.

   Aplikasi mungkin terdiri hanya dari satu *activity*, tetapi ada banyak aplikasi yang terdiri hanya dari satu *activity*, tetapi ada banyak aplikasi yang terdiri dari beberapa *Activity* seperti aplikasi pengiriman pesan yang sudah disebutkan diatas. *Activity* apa saja dan berapa banyak mereka, tentu saja tergantung pada aplikasi dan desain. Biasanya, satu aplikasi akan di pakai sebagai satu *UI* yang pertama diperlihatkan kepada pengguna saat aplikasi diaktifkan. Untuk pindah dari *Activity* ke *Activity* yang lainnya

kita dapatmelakukan satu *event* misalnya mengklik tombol, memilih opsi tertentu atau melalui *trigger* tertentu.

Setiap *Activity* diberikan default *window* untuk sebagai sebuah *UI*. Biasanya, *window* akan mengisi seluruh layar, tetapi mungkin lebih kecil daripada layar dan berada di atas *window* lain. Sebuah *Activity* juga dapat menggunakan jendela tambahan – misalnya, *pop-up dialog* yang meminta tanggapan pengguna di tengah-tengah pengguna informasi penting ketika mereka memilih item tertentu pada layar.

Isi (*content*) visual dari *window* disediakan oleh hirarki *view* – objek yang berasal dari kelas induk *View*. Setiap *View* mengkontrol setiap ruang didalam *window* (area persegi empat). *Parent view* berisi dan mengatur *child layout*, daun *view* (yang berada dibagian bawah hirarki) di tampilkan di area persegi empat, yang mengkontrol dan merespon terhadap aksi pengguna pada area tersebut. Misalnya, *View* menampilkan sebuah gambaran kecil dan menginisiasi sebuah aksi (*action*) ketika pengguna menyentuh gambar tersebut. *Android* memiliki sejumlah *viewm* siap pake (*ready-made*) yang dapat kita pakai – tombol, *text field*, *scrolls bar*, *menu item*, *check boxes* dan yang lainya. Hirarki *View* ditempaykan dalam sebuah *window Activity* oleh method *Activity.setContentView()*. *Content View* adalah objek yang berada pada *root* hirarki.

2. *Services*

Services tidak memiliki visual *interface* (*UI*), namun berjalan di latar belakang (secara *background*) untuk jangka waktu tak terbatas.

Misalnya, *service* mungkin mengambil data melalui jaringan atau menghitung sesuatu dan memberikan hasilkan untuk *Activity* yang membutuhkannya. Setiap *service* meng-extends kelas induk *Service.*

3. *Broadcast Receiver*

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

4. *Content Provider*

   Content Provider membuat set aplikasi data spesifikasi, tersedia untuk aplikasi lain. Data dapat disimpan dalam *file* sistem, dalam sebuah *databases SQLite*, atau cara lain yang bisa diterima. *Content Provider* meng-*extends* kelas dasar *ContentProvider* untuk menerapkan metode standar yang memungkinkan aplikasi lain untuk mengambil dan menyimpan data dari jenis yang dikontrolnya. Namun, aplikasi ini tidak memanggil metode lansung.sebaliknya mereka menggunakan Objek *ContentResolver* dan memanggil metode tersebut. *ContentResolver* dapat berkomunikasi dengan setiap *content provider*, yang bekerja sama dengan provider untuk mengelola setiap komunikasi *interprocess* yang terlibat.

Content provider menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *Activity* (Aplikasi), misalnya ketika kita menggunakan suatu aplikasi berbasis peta (*Map*), dan navigasi, maka disinilah *content provider* akan berperan. Setiap kali ada permintaan yang harus di tangani oleh komponen tertentu, *Android* akan memastikan bahwa proses aplikasi dari komponen berjalan, memulainya jika diperlukan, dan mengumumkan ketersediaan objek instantdari komponen, juga menciptkan objek instant jika diperlukan.

5. Versi Sistem Operasi *Android*

1. Versi 1.0 dirilis 23 September 2008
2. Versi 1.1 dirilis 9 February 2009
3. Versi 1.5 (*Cupcake*)
4. Versi 1.6 (*Donut*)
5. Versi 2.0 / 2.1 (*Eclair*)
6. Versi 2.2 (*Froyo*)
7. Versi 2.3 (*Gingerbread*) yaitu versi yang digunakan saat membangun aplikasi ini.
8. Versi 3.0 / 3.1 / 3.2 (*Honeycomb*)
9. Versi 4.0 (*IceCreamSandwich*)
10. Versi 4.1 (*Jelly Bean*)
11. Versi 4.2 (*Key Lime Pie*)