



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Pengertian Analisis

Karya tulis ini akan mengandung banyak sekali hasil analisis yang telah dikumpulkan dari berbagai narasumber. Berikut adalah berbagai pengertian mengenai apa itu analisis:

1. Analisis adalah penguraian suatu pokok atas berbagai bagiannya dan penelaahan bagian itu sendiri serta hubungan antar bagian untuk memperoleh pengertian yang tepat dan pemahaman arti keseluruhan. (Kamus Besar Bahasa Indonesia, 2002).
2. Analisis adalah langkah pertama dari proses perencanaan. (Gregory, 2000).
3. Analisis adalah penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan dapat diusulkan perbaikan-perbaikannya (Kusbianto, 2010).

Begitu banyak pola pikir, maupun definisi yang diterapkan oleh berbagai sumber yang beredar, tetapi dari semuanya dapat ditarik suatu garis besar dimana analisis merupakan proses pengkajian yang dilakukan pada proses perencanaan.

Pelaku yang melakukan analisis sebuah sistem disebut sebagai analis sistem. Analis sistem adalah orang yang menganalisis sistem (mempelajari masalah-masalah yang timbul dan menentukan kebutuhan-kebutuhan pemakai sistem) untuk mengidentifikasi pemecahan yang beralasan. (Kusbianto, 2010).

2.2 Pengertian Perancangan Sistem

Perancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru, jika sistem itu berbasis komputer, perancangan dapat dinyatakan spesifikasi peralatan yang digunakan. (McLeod, 2008).

Setelah melakukan analisis kebutuhan dan analisis sumber daya, lalu dapat dirancang sebuah sistem informasi yang dibutuhkan oleh klien. Pada proses perancangan sistem, dapat menggunakan beberapa tools seperti flowchart, *Entity Relationship Diagram (ERD)*, *Data Flow Diagram (DFD)*, juga dapat menggunakan *Unified Modelling Language (UML)*, seperti *activity diagram*, *sequence diagram*, *use case*, dan sebagainya.

2.3 Konsep Dasar Sistem Informasi

Sebelum ditelaah lebih jauh mengenai definisi serta pengertian dari sistem informasi, perlu ditelaah terlebih dahulu apa itu pengertian sistem serta pengertian informasi itu sendiri.

2.3.1 Pengertian Sistem

Sistem juga diartikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu tugas tertentu. (Jogiyanto, 2005).

Sistem dapat didefinisikan dengan pendekatan prosedur dan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu. (Mustakini, 2009).

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. (Sutabri, 2012).

Mendefinisikan sistem secara umum sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu sebagai satu kesatuan. (Mulyanto, 2009)

2.3.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu sebagai berikut (Jogiyanto, 2005):

a. Komponen-komponen sistem (components)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk satu kesatuan.

b. Batasan sistem (boundary)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

c. Lingkungan luar sistem (environments)

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

d. Penghubung sistem (interface)

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya.

e. Masukan sistem (input)

Masukan sistem adalah energi yang dimasukkan kedalam sistem.

f. Pengolah sistem (process)

Suatu sistem harus memiliki suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

g. Keluaran sistem (output)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna

h. Sasaran sistem (objectives)

Suatu sistem pasti mempunyai sasaran, kalau sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya.

2.3.2 Pengertian Informasi

Informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang. (McLeod, 2008). Lalu apa perbedaan antara data dengan informasi?

Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan. Data bisa berwujud suatu keadaan, gambar,

suara, huruf, angka, matematika, bahasa ataupun simbol-simbol lainnya yang bisa di gunakan sebagai bahan untuk melihat lingkungan, obyek, kejadian ataupun suatu konsep. (McLeod, 2008).

Sehingga dapat ditarik kesimpulan bahwa data adalah suatu produk mentah, sebuah bahan baku, yang tujuan utamanya adalah diolah untuk menjadi informasi. Di sini dapat melihat bahwa informasi adalah data yang sudah diolah, yang sudah mempunyai suatu nilai dan kegunaan yang dapat digunakan untuk membuat suatu kesimpulan, maupun membantu dalam membuat sebuah keputusan.

Data yang telah diolah dan telah melalui proses analisis akan menghasilkan sebuah informasi, apabila dapat dinarik sesuatu yang dapat digunakan dan berguna dari suatu data, maka dapat dikatakan data tersebut sudah bernilai sebuah informasi.

2.3.3 Kualitas Informasi

Kualitas informasi adalah ukuran terhadap nilai dari suatu kebutuhan yang ada dan telah ditetapkan dari data yang terorganisasi dan diproses sehingga menciptakan suatu bentuk yang berarti bagi *user*.

Saat berbicara mengenai informasi, hal ini berarti data yang telah diolah menjadi sesuatu yang bermanfaat, yang dapat ditelaah isinya. Namun, ada suatu masalah. Apabila data yang dikumpulkan dan diolah tidak bersih (*clean data*), bersih di sini artinya data tersebut sesuai dengan isinya, kemudian merupakan data yang sebenarnya, bukan hasil rekayasa maupun kesalahan pencatatan, sesuai dengan waktunya (*up-to-date*), sehingga informasi yang dihasilkan pun mencerminkan keadaan yang sebenarnya. Ada beberapa hal yang menjadi suatu indikator bagi informasi mana yang baik, dan informasi mana yang kualitasnya buruk. Berikut adalah daftarnya:

a. Akurat

Informasi yang akurat adalah informasi yang tidak memiliki banyak bias atau penyimpangan data yang ekstrim. Hal ini akan membuat informasi yang dimiliki tepat, tidak menimbulkan banyak kesalahan.

b. Tepat pada waktunya

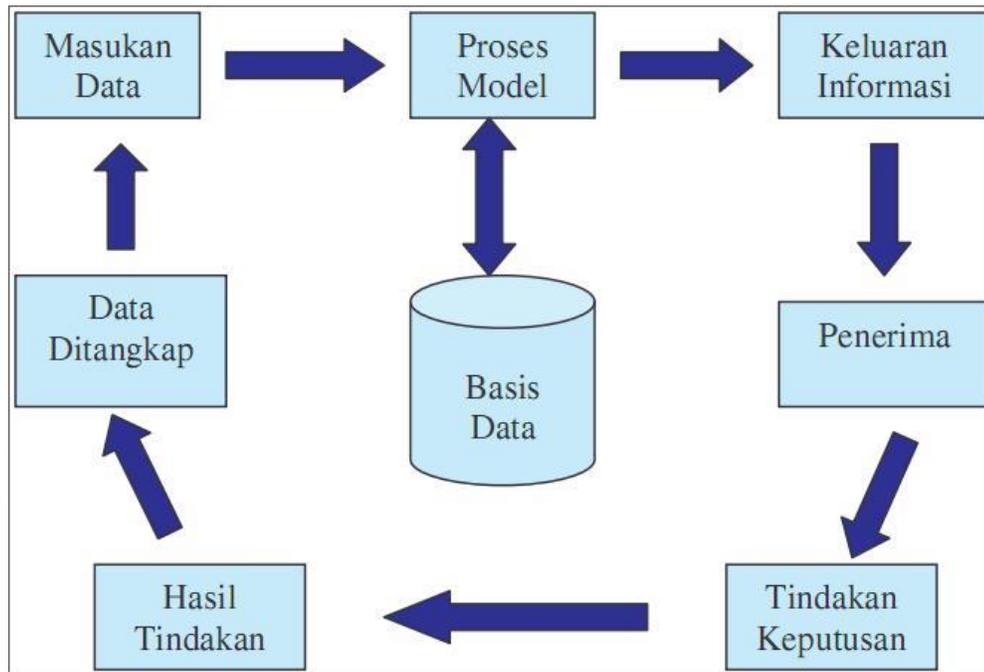
Informasi pun memiliki umur atau periode di mana informasi tersebut masih dapat dimanfaatkan secara maksimal. Contoh apabila terdapat informasi mengenai tipe ponsel yang laku pada tahun 2011, yang sudah terbukti akurat dan tepat, tetapi dimanfaatkan pada tahun 2014. Apa yang terjadi? Tentu saja informasi yang tadinya akurat tersebut, sekarang berkurang keakuratannya, atau bahkan sepenuhnya berbeda dengan kenyataan saat ini. Hal ini dikarenakan informasi yang dimiliki sudah ketinggalan zaman.

c. Relevan

Informasi yang relevan berarti informasi yang sesuai dengan kebutuhan. Contohnya adalah dalam penjualan produk kecantikan, tetapi informasi yang digunakan adalah informasi mengenai jenis mobil yang laku di pasaran. Hal ini sangat tidak berhubungan sehingga kendati terdapat suatu informasi yang bagus dan akuratnya sampai bagaimanapun, tetap informasi tersebut tidak berguna untuk penjual produk kecantikan.

UMMN

2.3.4 Siklus Informasi



Gambar 2.1 Siklus pembentukan dan pemanfaatan informasi

(Sumber: http://widuri.raharja.info/images/4/42/Siklus_informasi.jpeg)

Pada gambar 2.1 digambarkan proses dihasilkannya sebuah informasi dan pemanfaatan informasi. Data yang ditangkap akan ditampung di dalam basis data, dan akan dikerjakan di sebuah model proses guna menghasilkan keluaran informasi yang akan dimanfaatkan oleh penerima sehingga dapat menghasilkan sebuah tindakan keputusan. Dari hasil tindakan tersebut akan menghasilkan data baru lagi yang dapat ditangkap untuk menghasilkan informasi baru lainnya.

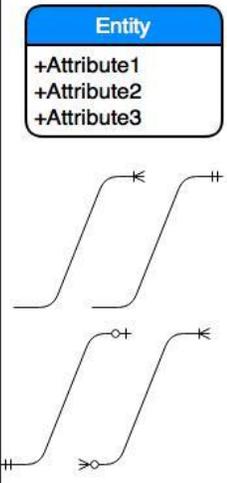
Dalam siklus informasi, komponen-komponen informasi saling berkesinambungan dalam menjalankan proses pembentukan informasi. Contohnya pada saat memulai memasukkan data, data melalui proses model, berintegrasi dengan basis data membentuk hasil keluaran informasi. Setelah hasil keluaran informasi ini diterima, si penerima akan melakukan suatu tindakan atas dasar informasi yang telah didapatkan sebelumnya, tindakan yang telah dilakukan oleh si penerima tersebut akan

menghasilkan suatu data baru, yang kemudian dapat dikumpulkan untuk menjadi masukan baru dan menghasilkan informasi baru.

2.4 Entity Relationship Diagram (ERD)

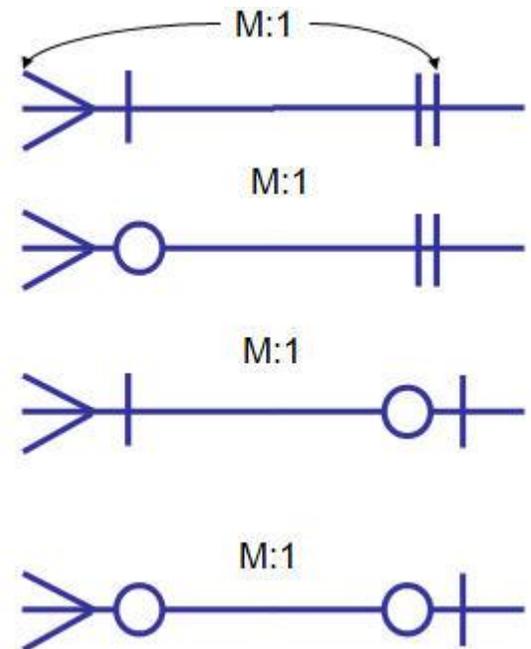
Suatu pemodelan konseptual yang didesain secara khusus untuk mengidentifikasi entitas yang menjelaskan data dan hubungan antar data. (Kroenke, 2006)

ERD adalah sebuah bentuk skema yang menggambarkan bagaimana beberapa tabel di dalam sebuah basis data saling berhubungan satu sama lain. ERD berbeda dengan DFD. ERD atau *Entity Relationship Diagram* menunjukkan hubungan antar tabel di dalam basis data saja, tidak menunjukkan hubungan antara basis data dengan entitas, maupun proses. Namun, ERD mengupas lebih dalam hubungan di dalam basis data, seperti penentuan *primary key*, *foreign key*, juga penentuan hubungan antar tabel seperti satu ke satu, satu ke banyak, banyak ke satu, banyak ke banyak, dan sebagainya yang akan dijelaskan lebih banyak di gambar di bawah ini.

Notasi	Deskripsi
	<p>Entiti adalah sebuah objek yang mewakili sebuah table di dalam database</p> <p>Beragam jenis panah ini menunjukkan hubungan antar tabel, yang akan dijelaskan lebih lengkap di bawah</p>

Gambar 2.2 Contoh tipe hubungan relasi antar basis data

- one through many notation on one side of a relationship and a one and only one
- zero through many notation on one side of a relationship and a one and only one
- one through many notation on one side of a relationship and a zero or one notation on the other
- zero through many notation on one side of a relationship and a zero or one notation on the other.



Gambar 2.3 Contoh tipe hubungan relasi antar basis data

(Sumber: www.philblock.info)

- Baris pertama, menunjukkan contoh hubungan antara hubungan 1 sampai banyak (1 ...*) di ujung kiri, dan di ujung kanan terdapat notasi 1 dan hanya boleh 1 (1). Hal ini bermaksud untuk menunjukkan hubungan bahwa setiap *record* tabel di bagian kiri dapat memiliki hubungan dengan 1 dan hanya boleh satu (tidak boleh nol, tidak boleh lebih dari satu) terhadap *record* yang berada di tabel yang kanan, sebaliknya juga setiap *record* dari tabel di bagian kanan, dapat memiliki hubungan dengan 1 sampai banyak *record* tabel yang sebelah kiri (artinya minimal 1, tetapi boleh lebih). Contohnya adalah akar pohon, setiap pohon pasti minimal memiliki 1 akar, tetapi boleh lebih. Sedangkan setiap akar pohon, pasti memiliki 1 dan hanya 1 induk pohon.
- Notasi di baris kedua menunjukkan untuk setiap *record* tabel di bagian kiri dapat memiliki hubungan dengan 1 dan hanya boleh satu (tidak boleh nol, tidak boleh lebih dari satu) terhadap *record* yang berada di tabel yang kanan, dan setiap *record* tabel di sebelah kanan berhubungan dengan *record* tabel di

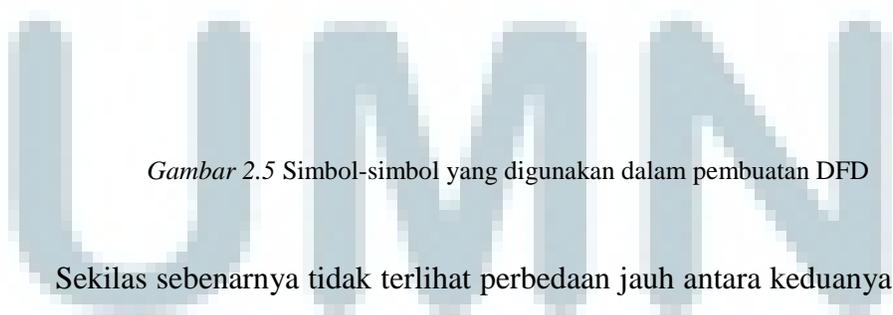
sebelah kiri dengan aturan boleh nol *record*, tetapi juga boleh banyak. Contohnya adalah hubungan antara orang tua dan anak, sepasang orang tua boleh memiliki banyak anak, tetapi boleh juga tidak memiliki anak sama sekali, tetapi setiap anak pasti punya 1 dan hanya 1 orang tua.

- Baris ketiga menunjukkan setiap *record* di tabel sebelah kiri, maka boleh memiliki 1 atau tidak memiliki sama sekali hubungan dengan *record* tabel di sebelah kanan (kalau memiliki hanya boleh memiliki 1, tetapi boleh juga tidak memiliki sama sekali). Sedangkan untuk setiap *record* dari tabel di bagian kanan, dapat memiliki hubungan dengan 1 sampai banyak *record* tabel yang di sebelah kiri (artinya minimal 1, tetapi boleh lebih). Contohnya adalah baju. Baju yang belum dibeli maka pemiliknya bisa dikatakan nol (tidak ada yang punya), tetapi saat dibeli, umumnya hanya didedikasinya untuk satu orang saja (jadi bisa nol, atau satu saja). Namun, setiap orang pasti minimal punya satu baju, dan boleh juga memiliki lebih dari satu baju.
- Baris keempat mengartikan untuk setiap *record* di tabel sebelah kiri, maka boleh memiliki 1 atau tidak memiliki sama sekali hubungan dengan *record* tabel di sebelah kanan (kalau memiliki hanya boleh memiliki 1, tetapi boleh juga tidak memiliki sama sekali). Namun, untuk setiap *record* tabel di sebelah kanan berhubungan dengan *record* tabel di sebelah kiri dengan aturan boleh nol *record*, tetapi juga boleh banyak. Contoh untuk hal ini adalah ponsel misalnya. Sebuah ponsel yang belum laku, maka tidak memiliki pemilik sama sekali. Namun, umumnya saat sudah dibeli, sebuah ponsel hanya dimiliki oleh satu orang saja. Tetapi setiap orang boleh juga memiliki lebih dari satu ponsel, boleh juga tidak punya sama sekali.

2.5 Data Flow Diagram (DFD)

Data Flow Diagram adalah sebuah bentuk diagram yang digunakan untuk menunjukkan arah aliran atau arus data di dalam sebuah sistem. Contohnya saat pegawai membutuhkan data barang, dari manakah data tersebut akan diambil? Atau saat terjadi transaksi, basis data mana sajakah yang akan mengalami perubahan? Hal ini perlu diidentifikasi di dalam DFD. Karena tanpa adanya DFD sulit membuat sebuah sistem informasi yang baik, secara logika tentu sistem informasi yang baik butuh arus informasi yang baik. Jika tidak memiliki suatu skema yang menunjukkan arus data yang kelak akan diolah menjadi informasi, bagaimana informasi yang didapatkan bisa tepat dan bersih? Maka dari itu penting untuk mengetahui skema arus data, yang dapat dilihat dari DFD.

Seperti dapat dilihat pada gambar 2.4 dapat ditemukan 2 jenis DFD yang secara prinsip sama, tetapi menggunakan simbol yang berbeda. Ada yang menggunakan simbol dari Gane dan Sarson, dan ada juga yang menggunakan simbol dari Yourdon dan De Marco. Sebenarnya secara prinsip kerja sama, hanya dibedakan oleh lambang saja.



Gambar 2.5 Simbol-simbol yang digunakan dalam pembuatan DFD

Sekilas sebenarnya tidak terlihat perbedaan jauh antara keduanya seperti yang terlihat di gambar 2.5, karena memang secara prinsip keduanya memakai prinsip dfd yang sama. Yaitu menentukan entitas, basis data, kemudian nama prosesnya, kemudian arah aliran datanya. Yang membedakan di DFD versi Gane dan Sarson ini adalah

bentuk proses yang apabila menggunakan metode Yourdon dan De Marco lambangnya adalah lingkaran, di versi Gane dan Sarson menggunakan segiempat dengan sudut melengkung. Dan lambang dari basis data di metode Gane dan Sarson, sebelah kirinya seperti ditutup dengan kotak segiempat, di mana pada metode Yourdon dan De Marco, basis data dilambangkan seperti 2 garis yang sejajar saja.

2.6 Android

Android adalah sistem operasi berbasis Linux bagi telepon seluler seperti telepon pintar dan komputer tablet. (Safaat, 2012)

Android adalah sebuah *OS (Operating System)* yang bekerja di mobile gadget seperti *handphone* dan *tablet*. *Android* sendiri dibangun berbasis pada *linux* dan awalnya dikembangkan oleh *Android, Inc.*, dengan dukungan finansial dari *google*, yang kemudian membelinya pada tahun 2005. Secara resmi, sistem operasi ini dirilis pada tahun 2007, *Android* sendiri dilepas ke pasar dengan konsep yang sama seperti *linux*, yaitu *open-source* yang berarti publik diperbolehkan melakukan modifikasi pada *source code android* tersebut.



Gambar 2.6 Lambang android (Sumber:android.com).

Saat ini *android* memiliki sejumlah besar komunitas pengembang aplikasi yang memperluas fungsionalitas perangkat, umumnya ditulis bahasa pemrograman *java*. Pada tahun 2013 tercatat 48 miliar aplikasi telah diunduh melalui “*play store*”, yaitu sebuah portal aplikasi untuk pengguna android, dimana pengguna dapat mengunduh aplikasi yang dapat memperkaya *gadget* mereka.

2.7 Android IDE (Integrated Development Environment)

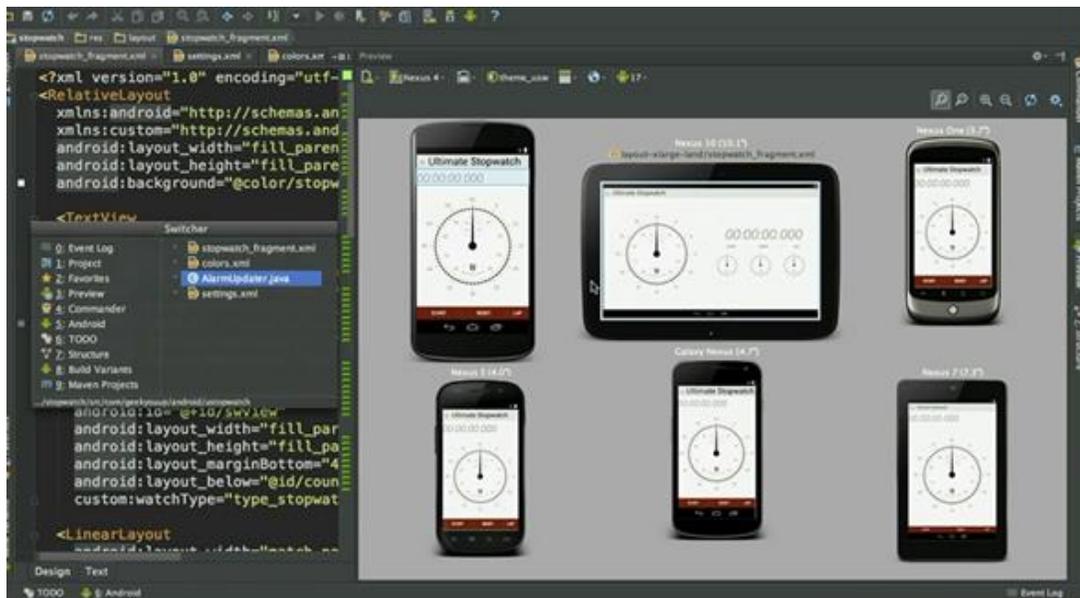
IDE sendiri adalah sebuah *tools* untuk melakukan proses pengembangan sebuah software. Kelebihan dari IDE sendiri adalah tersedianya library berisi beberapa *coding* umum yang dapat membantu dalam mengembangkan proyek aplikasi buatan *programmer* tersebut. Di kesempatan ini terdapat 2 alternatif pengembangan menggunakan IDE *eclipse* dan IDE *android studio*, setiap IDE tersebut memiliki kelebihan masing-masing yang dapat saling melengkapi.



Gambar 2.7 Lambang IDE eclipse

(Sumber: www.androidmania.us)

Eclipse sendiri sesungguhnya bukanlah sebuah *framework* yang mengkhususkan diri hanya untuk android saja, karena *eclipse* adalah sebuah *framework* multifungsi yang bisa digunakan untuk melakukan coding berbagai macam bahasa, salah satunya adalah *java*. *Java* inilah yang sebenarnya menjadi dasar dari pemrograman android. Namun, untuk melakukan *coding* di dalam *eclipse* di perlu menginstall android sdk ke dalam *eclipse*.

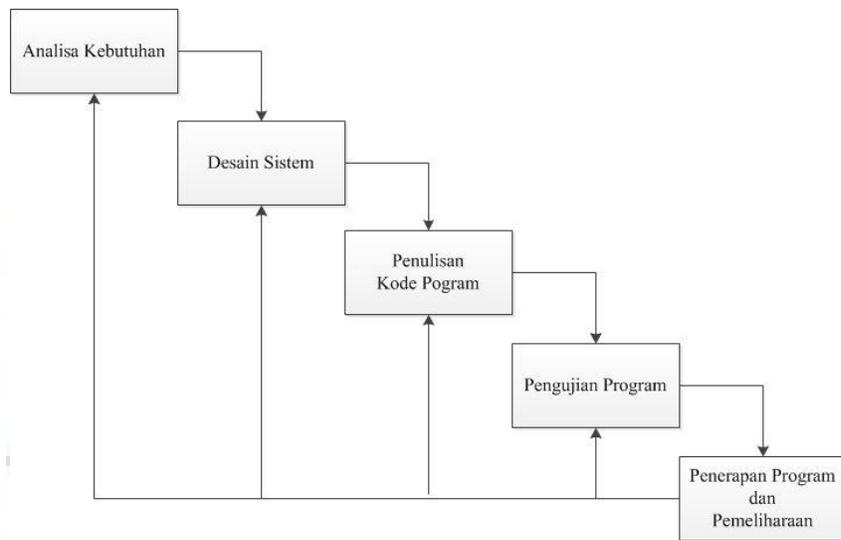


Gambar 2.8 Tampilan IDE *android manager* pada saat melakukan pengembangan aplikasi
(Sumber: www.redmondpie.com)

IDE (*Integrated Development Environment*) *Android Manager* adalah salah satu IDE rekomendasi dari google. Baik *android manager* maupun *eclipse* keduanya merupakan *framework* yang paling umum digunakan oleh *programmer* aplikasi android.

2.8 Metode pengembangan *waterfall*

Metode pengembangan *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. (Pressman, 2010)



Gambar 2.9 Metodologi penelitian waterfall

Metode ini menawarkan solusi yang baik apabila software akan dikembangkan dengan sifat kontinuitas. Di mana software mengalami perubahan yang cukup sering. Karena pada saat pengujian program pun, apabila dirasa ada kebutuhan tambahan dapat langsung kembali ke analisa kebutuhan. Namun, setelah menganalisa kebutuhan dan ditemukan kebutuhannya, tetap harus mengikuti langkahnya menuju desain sistem, penulisan program baru kembali ke tahap pengujian lagi.

UMMN