BAB II

LANDASAN TEORI

2.1. Pengertian Perancangan

Perancangan adalah penggambaran, perencanaan dan pembuatan *sketsa* atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi sebagai perancangan sistem dapat dirancang dalam bentuk bagan alir sistem (*system flowchart*), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukan urutan-urutan proses dari sistem. Syifaun Nafisah, (2003 : 2).

2.1.1. Pengertian Perancangan Aplikasi

Konsep merancang *multimedia* merupakan aplikasi *multimedia* yang akan dibuat. Untuk dapat merancang konsep dalam membuat aplikasi *multimedia* dibutuhkan kreatifitas. Kreatifitas adalah kemampuan untuk menyajikan gagasan atau ide baru. Sedangkan inovasi merupakan aplikasi dari gagasan atau ide baru tersebut. Untuk menciptakan ide yang *orisinil* tidaklah mudah, maka dapat digunakan beberapa teknik untuk menciptakan ide, yaitu penyesuaian (adaptasi), *Multimedia* yang telah ada dianggap belum sesuai dengan lingkungan yang dituju.

Merancang konsep analisis sistem bekerjasama dengan pemakai, mungkin juga bekerjasama dengan profesional komunikasi seperti *produser*, *sutradara*, penulis naskah, *editor* elektronik terlibat dalam merancang konsep yang menentukan keseluruhan pesan dan membuat aliran (urutan) pada

aplikasi *multimedia* yang akan dibuat. Untuk dapat merancang konsep dalam membuat aplikasi *multimedia* dibutuhkan kreativitas.

Multimedia yang telah ada dianggap belum sesuai dengan lingkungan yang dituju walaupun isinya telah sesuai dengan lingkungan. Misalnya *multimedia* ditujukan ke kalangan kawula muda namun *multimedia* yang ada hanya sesuai untuk kalangan orang tua (dari sisi desain tampian, bahasa dll) pembesaran *(maksimasi), Multimedia* yang ada dianggap sangat sederhana, sehingga perlu untuk dikembangkan lebih kompleks pengecilan *(minimasi), Multimedia* yang ada dianggap terlalu rumit dan sulit untuk difahami, dimengerti, dioperasikan dll. Sehingga perlu untuk dilakukan. Penyederhanaan pembalikan *(inversi),* Multimedia yang telah ada dianggap memiliki isi yang keliru, sehingga diperlukan perubahan isi *multimedia* secara menyeluruh terhadap pengembangan *multimedia* tersebut perubahan *(modifikasi),* pengaturan kembali dan perpaduan *(kombinasi). Multimedia* dianggap belum sesuai dengan kebutuhan dan dirubah dengan menggunakan teknik perubahan lebih dari satu cara pengembangan.

2.1.2. Pengertian Perancangan Sistem

Ada beberapa pengertian perancangan sistem menurut beberapa ahli antara lain :

1. *Verzello / John Reuter* III Tahap setelah analisis dari siklus pengembangan sistem : Pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi : menggambarkan bagaimana suatu sistem dibentuk.

2. *John Burch & Gary Grudnitski* Desain sistem dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan *sketsa* atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

3. *George M. Scott* Desain sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan tahap ini menyangkut mengkonfigurasikan dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem, sehingga setelah *instalasi* dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem.

2.1.3. Pengertian Aplikasi

Adapun pengertian aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

1. Aplikasi *software spesialis*, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

Menurut Hengky w. Pramana aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, *game*, pelayanan masyarakat, periklanan dan hampir semua proses kegiatan.

2.1.3.1. Perangkat lunak aplikasi

Yang dimaksud perangkat lunak aplikasi adalah suatu *sub* kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *Open Office*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antar muka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan

tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.1.3.2. Aplikasi Komputer

Aplikasi Komputer atau Aplikasi *Software* adalah Program komputer yang ditulis dalam suatu bahasa pemrograman dan dipergunakan untuk menyelesaikan masalah tertentu.

Adapun beberapa contoh aplikasi komputer :

1. Aplikasi *Grafik*

   Aplikasi komputer *grafik* awal perkembangannya dimulai pada akhir tahun 1950-an dan awal 1960-an. Merupakan suatu program komputer yang digunakan dengan menarik sebuah gambar yang memberikan dimensi baru pada bidang komputer saat itu.

2. Aplikasi *Web*

   Merupakan bagian dari *internet* sebagai komunitas jaringan komputer yang memberikan pelayanan *www* (*world wide web*). Dengan demikian, definisi teknis dari *world wide web* adalah semua sumber daya dan semua pengguna di

internet yang menggunakan *HTTP* (*Hypertext Transfer Protocol*).

a. *WWW* adalah aplikasi yang paling menarik di *Internet* dan seperti *e-mail*, aplikasi ini sangat penting dan banyak digunakan. Aplikasi ini kadang disebut "*The Killer Application*" atau "*the world is at your fingertip*" karena kita bisa mendapatkan hasil dengan begitu mudah dan bukan sekadar teks yang bisa kita dapatkan namun gambar (*images*), maupun *multimedia*.

3. Aplikasi Pendidikan

a. *Computer Assisted Instruction (CAI)*

Komputer secara langsung digunakan dalam proses belajar, sebagai pengganti pengajar ataupun buku. Beberapa aplikasi *CAI* adalah :

- *Drill and Practice*
- Tutorial
- Simulasi

b. *Computer Managed Instruction (CMI)*

Para pengajar memanfaatkan komputer untuk merencanakan kuliah, disesuaikan dengan kondisi

para siswa, yang terdiri dari acara belajar dengan bantuan komputer,membaca,dan ujian.

2.2. Konsep Dasar Sistem

Definisi sistem menurut buku analisis desain dan informasi dapat dilihat dari kelompok pendekatan yaitu yang menekankan pada prosedur dan yang menekankan pada komponen atau elemennya.

1. Berdasarkan Penekanan Prosedur

   Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. Jogiyanto (2008 : 1)

2. Berdasarkan Penekanan Komponen

   Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Jogiyanto (2008 : 2)

2.2.1.Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen *(components)*, batas Sistem *(boundary)*, lingkungan luar sistem *(environments)*, penghubung *(interface)*, masukan *(input)*, keluaran *(output)*, pengolah *(process)* dan sasaran *(objectives)* atau tujuan *(goal)*. Jogiyanto (2008 : 3)

1. Komponen Sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu

kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sub-sub sistem mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Jogiyanto (2008 : 4)

2. Batas Sistem

   Batas sistem *(boundary)* merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukan ruang lingkup *(scope)* dari sistem tersebut.

3. Lingkungan Luar Sistem

   Lingkungan luar *(environment)* dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan. Lingkungan luar yang menguntungkan merupakan energi dari sistem dengan demikian harus dijaga dan dipelihara, sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan kalau tidak maka akan menggangu kelangsungan hidup dari sistem.

4. Penghubung Sistem

   Penghubung *(interface)* merupakan media penghubung antara *subsistem* dengan *subsistem* lainnya. Melalui penghubung ini

memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Keluaran *(output)* dari subsistem akan menjadi masukan *(input)* pada sistem lainnya dengan penghubung satu subsistem dapat berintegrasi dengan subsistem lainnya membentuk satu kesatuan. Jogiyanto (2008 : 5)

5. Masukan Sistem

   Masukan *(input)* sistem adalah energi yang dimasukan ke dalam sistem. Masukan dapat berupa masukan perawatan *(maintenance input)* dan masukan *(signal input)*. *maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluarannya.

6. Keluaran Sistem

   Keluaran *(output)* adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada *supra* sistem.

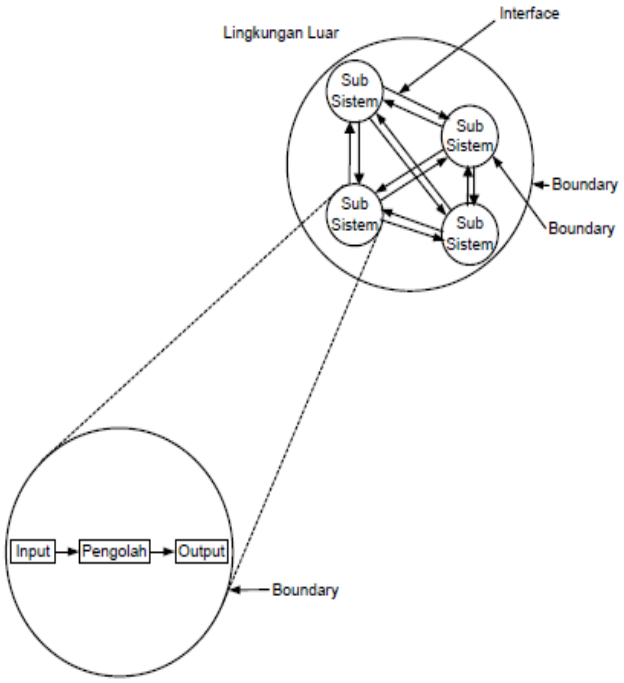
7. Pengolah Sistem

   Suatu sistem dapat mempunyai suatu bagian pengolahan atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

   Suatu sistem pasti mempunyai tujuan *(goal)* atau sasaran

(objektif). Kalau suatu sistem tidak mempunyai sasaran maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran aturan tertentu.



Gambar 2.1. Karakteristik Suatu Sistem

Sumber : Analisis dan Design. Jogiyanto (2008 : 6)

2.2.2. Klasifikasi Sistem

Suatu sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya adalah sebagai berikut ini. Jogiyanto (2008 : 6)

a. Sistem menurut bentuk fisiknya :

1. Sistem Abstrak *(abstract system)*

Sistem abstrak adalah sistem yang berupa pemikiran atau ide – ide yang tidak tampak secara fisik. Contoh : Sistem *teologia* adalah sebuah susunan gagasan mengenai Tuhan, manusia dan alam. Jogiyanto (2008 : 7)

2. Sistem Fisik *(physical system)*

Sistem fisik merupakan sistem yang ada secara fisik.

Contoh : Sistem peredaran darah, sistem sekolah, sistem transportasi, sistem komputer.

b. Sistem menurut terjadinya sistem :

1. Sistem Alamiah *(natural system)*

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia.

Contohnya : pergantian siang dan malam, erosi dan bencana alam.

2. Sistem Buatan Manusia *(human made system)*

Sistem buatan manusia adalah sistem yang dirancang oleh manusia.

Contohnya : Sistem Komputer dan sistem irigasi.

c. Sistem menurut kejadian masa depan

1. Sistem Tertentu *(deterministic system)*

Sistem tertentu adalah sistem yang beroperasi dengan

tingkah laku yang sudah dapat di prediksi.

Contoh : Hasil pertadingan sepak bola, dan prestasi.

2. Sistem Tak Tentu *(probabilistic system)*

Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat di prediksi karena mengandung unsur probabilitas.

Contoh : Sistem kematian.

d. Sistem menurut sifatnya

1. 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 diluarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar – benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tetapi tidak benar – benar tertutup).

Contoh : Sistem adat masyarakat baduy.

2. Sistem Terbuka *(open system)*

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

Contoh : Sistem musyawarah. Jogiyanto (2008 : 7)

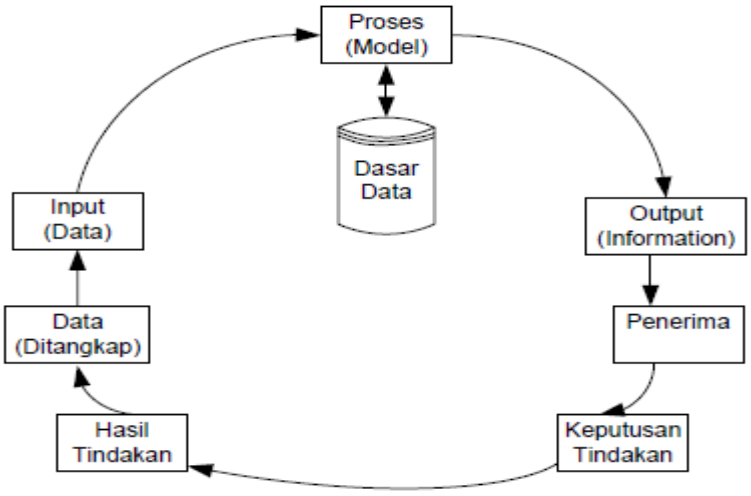
2.3. Konsep Dasar Informasi

2.3.1.Definisi Sistem Informasi

Sistem informasi menurut *Robert N. Anthony* dan *John Dearden* adalah suatu data yang diolah manjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Jogiyanto (2008 : 8)

2.3.2.Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus.



Gambar 2.2. Siklus Informasi

Sumber : Analisis dan Design. Jogiyanto (2008 : 9)

2.3.3.Kualitas Informasi

Kualitas dari suatu informasi menurut buku analisis desain dan informasi tergantung dari tiga hal, yaitu informasi harus akurat, tepat pada waktunya dan relevan. Jogiyanto (2008 : 10)

1. Akurat

   Informasi harus akurat berarti informasi harus bebas dari kesalahan- kesalahan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai penerima informasi kemungkinan banyak terjadi gangguan *(noise)* yang dapat merubah atau dapat merusak informasi tersebut.

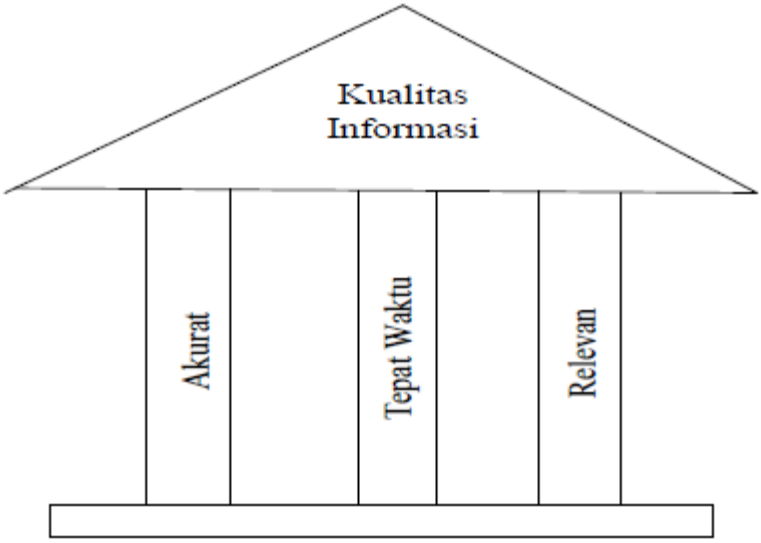
2. Tepat Pada Waktunya

   Informasi harus tepat pada waktunya berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi, karena informasi merupakan landasan didalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi.

3. Relevan

   Informasi harus relevan berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

   John Burch dan *Gary Grudnitski* menggambarkan kualitas dari informasi dengan bentuk bangunan yang ditunjang oleh tiga pilar.



Gambar 2.3. Kualitas Informasi

Sumber : Analisis dan Design. Jogiyanto (2008 : 10)

2.3.4.Nilai Informasi

Suatu informasi dikatakan bernilai bila informasi lebih efektif dibandingkan dengan biaya mendapatkannya. Kegunaan informasi adalah untuk mengurangi hal ketidakpastian didalam proses pengambilan keputusan tentang suatu keadaan. Nilai dari Informasi ditentukan dari dua hal yaitu manfaat dan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu sistem, informasi umumnya digunakan untuk beberapa kegunaan. Pengukuran suatu nilai informasi biasanya dihubungkan dengan analisis *cost effectivennes* atau *cost benefit*. Jogiyanto (2008 : 11)

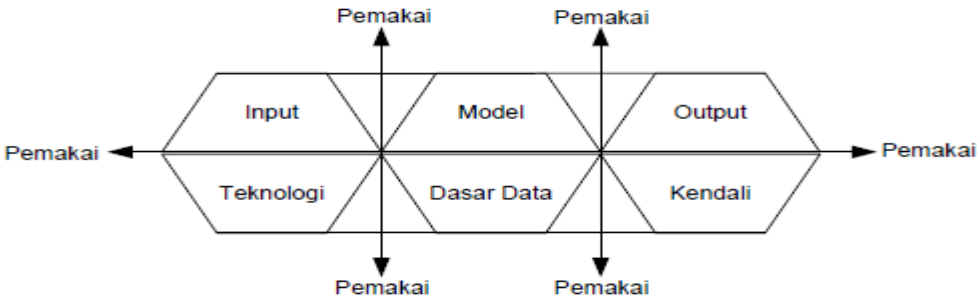
2.4. Konsep Dasar Sistem Informasi

Suatu informasi dapat diperoleh dari sistem informasi *(information system)* atau disebut juga dengan *processing systems* atau *information processing* atau *information-generating systems*.

Sistem informasi didefinisikan oleh *Robert A. Leitch* dan *K. Roscoe Davis* adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Jogiyanto (2008 : 11)

2.4.1. Komponen Sistem Informasi

John Burch dan *Gary Grudnitski* mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah *building block* yaitu blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Sebagai suatu sistem, keenam blok tersebut membentuk suatu kesatuan untuk mencapai sasaran. Jogiyanto (2008 : 12)



Gambar 2.4. Blok sistem informasi yang berinteraksi

Sumber : Analisis dan Design. Jogiyanto (2008 : 12)

1. Blok Masukan

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

2. Blok Model

   Blok model ini terdiri dari kombinasi prosedur logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan didasar data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan. Jogiyanto (2008 : 13)

3. Blok Keluaran

   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

   Teknologi merupakan "kotak alat" *(toolbox)* dari pekerjaan sistem informasi, teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendaliaan dari sistem keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi *(humanware atau brainware)*, perangkat lunak *(software)* dan perangkat keras *(hardware).*

5. Blok Basis Data

   Basis data *(database)* merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan didalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data didalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat yang disebut dengan *DBMS (Data Base Management Sistem)*.

6. Blok Kendali

   Supaya sistem informasi dapat berjalan sesuai dengan yang diinginkan, maka perlu diterapkan pengendalian-pengendalian didalamnya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan dapat langsung cepat diatasi. Jogiyanto (2008 : 13)

2.4.2. Sistem Informasi Manajemen

Sistem informasi manajemen *(management information system)* atau sering dikenal dengan singkatan (MIS) merupakan penerapan sistem informasi di dalam organisasi untuk mendukung informasi-informasi yang dibutuhkan oleh semua tingkatan manajemen. SIM

(Sistem Informasi Manajemen) didefinisikan oleh beberapa para ahli. Jogiyanto (2008 : 14)

Menurut para ahli :

1. Menurut *George M. Scott*

   Suatu sistem informasi manajemen adalah kumpulan dari interaksi- interaksi sistem informasi yang menyediakan informasi baik untuk kebutuhan manajerial maupun operasional.

2. Menurut *Barry E. Cushing*

   Suatu SIM adalah kumpulan dari manusia dan sumber-sumber daya modal di dalam suatu organisasi yang bertanggung jawab mengumpulkan dan mengolah data untuk menghasilkan informasi yang berguna untuk semua tingkatan manajemen di dalam kegiatan perencanaan dan pengendalian.

3. Menurut *Frederick H. Wu*

   Sistem informasi manajemen adalah kumpulan dari sistem-sistem yang menyediakan informasi untuk mendukung manajemen.

4. Menurut *Gordon B. Davis*

   Sistem informasi manajemen adalah sistem manusia atau mesin yang menyediakan informasi untuk mendukung operasi manajemen dan fungsi pengambilan keputusan dari suatu organisasi. Dari definisi diatas dapat disimpulkan bahwa sistem informasi manajemen adalah:

a) Kumpulan dari interaksi sistem-sistem informasi menghasilkan informasi yang berguna bagi semua tingkatan.

b) Jadi sistem informasi manajemen adalah seperangkat alat yang digunakan oleh para pengambil keputusan dalam membuat suatu keputusan dalam rangka melaksanakan kegiatan-kegiatan manajemen dengan cara memberikan informasi yang didasarkan oleh pandangan secara sistema sehingga mencapai tindakan optimal. Jogiyanto (2008 : 15)

2.5. Definisi Penjualan

Penjualan adalah Penerimaan yang diperoleh dari pengiriman barang dagangan atau dari penyerahan pelayanan dalam bursa sebagai barang pertimbangan. Pertimbangan ini dapat dalam bentuk tunai peralatan kas atau harta lainnya. Pendapatan dapat diperoleh pada saat penjualan, karena terjadi pertukaran, harga jual dapat ditetapkan dan bebannya diketahui.

Sistem (http://www.scribd.com/doc/11320689/Definisi-Penjualan/ 21 April 2010)

2.5.1.Definisi Sistem Informasi Penjualan

Sistem informasi penjualan adalah suatu sistem informasi yang mengorganisasikan serangkaian prosedur dan metode yang dirancang untuk menghasilkan, menganalisa, menyebarkan dan memperoleh informasi guna mendukung pengambilan keputusan mengenai penjualan dan promosi.

2.6. Perangkat Lunak Pendukung

2.6.1. *Hypertext Markup Language (HTML)*

Hypertext Markup Language (HTML) adalah bahasa yang digunakan untuk menulis halaman *web*. *HTML* merupakan pengembangan dari standar

pemformatan dokumen teks yaitu *Standard Generalized Markup Language (SGML). HTML* sebenaranya adalah dokumen *ASCII* atau teks biasa, yang dirancang untuk tidak tergantung pada stuatu sistem operasi tertentu. *HTML* dibuat oleh Tim *Berners-Lee* ketika masih bekerja untuk *CERN* dan dipopulerkan pertama kali oleh *browser Mosaic*. Selama awal tahun 1990 *HTML* mengalami perkembangan yang sangat pesat. Setiap pengembangan *HTML* pasti akan menambahkan kemampuan dan fasilitas yang lebih baik dari versi sebelumnya. Namun perkembangan tersebut tidak sampai mengubah cara kerja dari *HTML*. *HTML* 2.0 secara resmi dikeluarkan pada bulan November 1995 oleh *IETF* (*Internet Engineering Task Force)*. *HTML* 2.0 ini merupakan penyempurnaan dari HTML+ (1993).

2.6.2. PHP (Personal Home Page)

Menurut Andi setiawan dan Ade Irma (2006:16) yang dimaksud dengan *PHP (Personal Home Page)* adalah sebuah bahasa yang *HTML*

(Hypertext Markup Language) embedded, artinya perintah-perintah dalam *PHP* dapat menyatu dengan *tag-tag HTML (Hypertext Markup Language)* yang didalamnya kita masukan perintah-perintah tertentu. Menurut dokumen resmi *PHP*, *PHP* singkatan dari *Personal Hypertext preprocessor*. *PHP* merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*. Hasilnya adalah dikirimkan ke *klien*, tempat penggunaan memakai *browser.*

2.6.3. XAMPP for Window

XAMPP merupakan sebuah *tool* yang menyediakan beberapa paket perangkat lunak ke dalam satu buah paket. Dalam buku ini, penulis menggunakan *XAMPP*-win32- versi 1.7.3 yang tersedia dalam CD yang disertakan. *XAMPP* versi 1.7.3 terdiri atas : Arief Ramadhan & Hendra Saputra (2005:2)

1. *Apache* 2.0.53
2. *MySQL* 4.1.9
3. *PHP* 5.0.3
4. *SQLite* 2.8.15
5. *PHPMyAdmin* 2.6.1
6. *ADODB* 4.60
7. *Zend Optimizer* 2.5.7, dan lain-lain.

Dengan mengisntal *XAMPP*, kita tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache*, *PHP* 5, dan *mySQL* secara manual. *XAMP* akan mengintalasi dan mengonfigurasikannya

secara otomatis.

2.6.4. Mysql

Menurut Bunafit Nugroho (2004:29) *MySQL (My Strukture Query Language)* adalah "sebuah program pembuat *database* yang bersifat *open source*". *MySQL* sebenarnya produk yang berjalan pada *platform Linux* karena sifatnya *open source MySQL* dapat dijalankan pada semua *platform* baik *Windows* maupun *Linux.*

Database MySQL adalah database yang sangat *powerfull*, stabil, mudah. MySQL sangat banyak dipakai dalam sistem *database web* dengan menggunakan *PHP*. *PHPTriad* juga memberikan fasilitas *database* yang. Karena *PHPTriad* dilengkapi dengan *database MySQL* maka terdapat tempat untuk menyimpan data *(store)*, dan untuk mengambil kembali data anda *(retrieve)*. Seperti sistem database SQL *(Structured Query Language)* yang lain, MySQL juga dilengkapi dengan perintah-perintah dan sintaks-sintaks SQL, dengan keunggulan sebagai berikut.

a. Konsep *database MySQL* berkecepatan tinggi tentang sistem penyajian data.
b. Harga yang relatif murah, karena ada yang dapat diperoleh secara gratis.
c. Sintaks bahasanya menggunakan perintah yang sederhana.
d. Dapat bekerja dalam beberapa sistem operasi seperti *Windows, Linux, MacOs, Unix (Solaris, AIX, dan DEC Unix), FreeBSD, OS/2,*

Irix.

e. Dukungan penggunaan banyak tersedia

2.6.4.1. Kemampuan *SQL Server*

Menurut Andi setiawan dan Ade Irma (2006:29) yang dimaksud dengan *SQL server* adalah sistem manajemen *database relasional (RDBMS)* yang dirancang untuk aplikasi dengan arsitektur *client/server*. Istilah *client*, *server* dan *clien/server* dapat digunakan untuk merujuk kepada konsep yang sangat umum atau hal spesifikasi dari perangkat keras atau perangkat lunak. Pada level yang sangat umum, sebuah *client* adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya *(resources)* dari komponen sistem lainnya, sedangkan sebuah *server* adalah setiap komponen sistem yang menyediakan layanan atau sumber daya kekomponen lainnya.

Sistem *client/server* dirancang untuk memisahkan layanan basis data dari *client*, dengan penghubungnya dengan menngunakan jalur komunikasi data. Layanan basis data diimplementasikan pada sebuah komputer yang berdaya guna, yang memungkinkan manajemen tersentralisasi, keamanan dan berbagi sumber daya. Oleh karena itu, *server* dalam *client/server* adalah basis data dan layanannya. Aplikasi-aplikasi *client* diimplementasikan pada berbagai *platform*

menggunakan berbagai *tools* pemrograman.

SQL server adalah *server* basis data yang secara fungsional adalah proses aplikasi yang menyediakan layanan basis data. *Client* berinteraksi dengan layanan basis data melalui antarmuka komunikasi tertentu yang bertujuan untuk pengendalian dan keamanan. *Client* tidak mempunyai akses langsung kedata, tetapi selalu berkomunikasi dengan *server* basis data.

2.6.4.2.Macromedia Dreamweaver

Macromedia Dreamweaver adalah salah satu produk dari *Macromedia* yang merupakan salah satu *produsen software* terkemuka. Software ini banyak digunakan oleh programer profesional dalam membangun sebuah aplikasi berbasis *web*. *Macromedia Dreamweaver* memudahkan penggunanya dalam merancang, pengkodean dan pengembangan aplikasi *web*. *Software* ini mendukung berbagai macam *script* pemrograman seperti *HTML, PHP, CSS, JavaScript, CFML, ASP dan JSP*.

2.7. Pendekatan Berorientasi Objek

Menurut Adi Nugroho (2005 : 4) yang dimaksud pendekatan berorientasi objek adalah.

Suatu cara baru dalam berpikir atau berlogika dalam menghadapi masalah-masalah yang akan dicoba atasi dengan bantuan komputer, mencoba melihat permasalahan lewat pengamatan dunia nyata dimana setiap objek adalah *entitas* tunggal yang memiliki kombinasi struktur data dan fungsi tertentu.

Suatu cara pengembangan perangkat lunak dan sistem informasi berdasarkan objek-objek yang ada di dunia nyata.

Teknologi berorientasi objek menganalogikan sistem seperti kehidupan nyata yang didominasi objek. Berikut prinsip dasar dari pemograman berorientasi objek yang terdiri dari :

1. *Abstraksi* : memfokuskan pada karakteristik objek.
2. Kelas : bagian yang dapat menciptakan objek.
3. *Enkapsulasi* (Pembungkusan) : menyembunyikan banyak hal yang terdapat dalam objek yang tidak perlu diketahui objek lain.
4. *Generalisasi* (spesialisasi) : pembagian kelas-kelas yang berprilaku sama.
5. *Polimorfisme* : fungsi yang sama dapat diterapkan dan dimiliki oleh kelas-kelas berlainan.

2.7.1. UML *(Unified Modeling Language)*

Menurut *Martin Fowler* (2004 : 1) yang dimaksud dengan UML adalah keluarga *notasi grafis* yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemograman berorientasi objek.

UML dikeluarkan oleh OMG *(Object Management Group, Inc)* yaitu organisasi internasional yang dibentuk pada 1989, terdiri dari perusahaan sistem informasi, *software developer* dan para *user* sistem komputer.

Beberapa konsep dasar dalam OOP *(Object Oriented Programing)* menurut *Martin Flower* :

1. *Object*

   Objek *(object)* adalah "benda", secara fisik atau konseptual, yang dapat kita temui disekeliling kita. *Hardware, software,* dokumen, manusia, dan bahkan konsep semuanya adalah contoh objek. Untuk kepentingan memodelkan perusahaannya, seorang kepala eksekutif akan melihat karyawan, gedung, divisi, dokumen, dan keuntungan sebagai objek. Seorang teknisi mesin akan melihat ban, pintu, mesin, laju tertinggi, dan banyaknya bahan bakar sebagai objek. Dan, seorang *software engineer* akan memandang tumpukan, antrian, jendela *(window),* dan *check box* sebagai objek. Sebuah objek memiliki keadaan sesaat *(state)* dan perilaku *(behavior). State* dari sebuah objek adalah kondisi objek tersebut atau himpunan dari keadaan yang menggambarkan objek tersebut. Sebagai contoh, bola lampu adalah

objek, dan salah satu keadaan nyala atau tidak adalah *state* dari objek bola lampu tersebut. Salah satu *state* dari objek jam adalah waktu saat ini. *State* dinyatakan dengan nilai dari atribut *(attribute)* objeknya. Atribut adalah nilai internal suatu objek yang mencerminkan antara lain karakteristik objek, kondisi sesaat, koneksi dengan objek lain, dan identitasnya. Perubahan *state* dicerminkan oleh perilaku *(behavior)* objek tersebut.

Behavior suatu objek mendefinisikan bagaimana sebuah objek bertindak (beraksi) dan memberi reaksi. Behavior ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan dalam objek itu sendiri. *Behavior* dari sebuah objek dicerminkan oleh *interface, service,* dan *method* dari objek tersebut. Interface adalah pintu untuk mengakses *service* objek. *Service* adalah fungsi yang bisa diemban objek. *Method* adalah mekanisme internal objek yang mencerminkan perilaku *(behavior)* objek tersebut. Sebagai contoh, jika *printer* sebuah merupakan perilaku *(behavior)* atau *service*-nya mencetak apapun yang dia terima.

2. Class

Kelas *(class)* adalah definisi umum (pola, *template* atau catak biru) untuk himpunan objek sejenis. Kelas menetapkan spesifikasi perilaku *(behaviors)* dan atribut objek-objek tersebut. *Class* adalah keniskalan *(abtraksi)* dari *entitas* dalam dunia nyata. Objek adalah "contoh" *(instance)* dari sebuah kelas. Sebagai contoh, atribut

untuk kelas binatang adalah berkaki empat dan memiliki ekor. Perilakunya adalah tidur dan makan. Contoh *(instance)* yang mungkin dari kelas binatang ini adalah kucing, gajah, dan kuda.

3. *Inheritance*

   Seperti yang sudah diuraikan di atas, objek adalah contoh/*instance* dari sebuah *class*. Hal ini mempunyai konsekuensi yang penting yaitu sebagai *instande* sebuah *class*. Sebuah objek mempunyai semua karateristik dari *class*-nya. Ini yang di sebut *inherintance* (perwarisan sifat). Dengan demikian apapun *attribute* dan *operation* dari *class* akan dimiliki pula oleh sebuah objek yang di *inherit* / diturunkan dari *class* tersebut. Sifat ini tidak hanya berlaku untuk objek terhadap *class*, akan tetapi juga berlaku sebagai *class* terhadap *class* lainnya.

4. *Polimorphisme*

   Polimorphisme adalah konsep yang handal bagi pengembangan perangkat lunak untuk pemisahan secara jelas diantara *sub* sistem yang berbeda. Dengan demikian sebuah sistem akan bisa dimodifikasi secara mudah karena hanya dibutuhkan *interface* antar *class*.

5. *Encapsulation*

   Encapsulation adalah proses menyembunyikan *detil implementasi* sebuah objek. Satu – satunya jalan untuk mengakses data objek tersebut adalah melalui *interface. Interface* melalui

internal *state* sebuah objek dari "campur tangan" pihak luar. Oleh karena itu objek sering digambarkan sebagai kotak hitam *(black box)* yang menerima dan mengirim pesan-pesan *(messages).* Dalam *object-orinted programming* kotak hitam tersebut berisi kode (himpunan intruksi dengan bahasa yang dipahami komputer) dan data (informasi dimana intruksi tersebut beroperasi dengannya). Dalam *object-oriented programming*, kode dan data disatukan dalam sebuah benda "benda" yang tersembunyi isinya, yaitu objek. Pengguna objek tidak perlu mgengetahui isi dalam kotak tersebut. Untuk dapat berkomunikasi dengan objek diperlukan pesan *(messages).* Secara formal kita didefinisikan *message* sebagai permintaan untuk objek penerima *(receiver object).* Untuk membawa metode yang ditunjukkan atau perilaku dan mengembalikan *result* dari aksi tersebut kepada objek pengirim *(sender object)*. Sebagai contoh, satu *object* orang mengirim *object* bola lampu sebuah pesan *(message)* untuk menyala (melalui saklar). Objek bola lampu menyalakan dirinya dan menunjukkan kepada objek orang tersebut bahwa *state* barunya adalah menyala.

6. *Association* dan *Aggregation*

Association (asosiasi) adalah hubungan antar objek yang saling membutuhkan. Sedangkan, *aggregation* (agregasi) adalah bentuk khusus dari asosiasi yang menggambarkan seluruh bagian suatu

objek merupakan bagian dari objek lainnya. Sebagai contoh, objek tanggal dapat disusun dari objek hari, objek bulan, dan objek tahun. Satu alasan mengapa *object-oriented* menguntungkan bagi *programmer* dapat mendesai program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyala. Keuntungan yang lain adalah proses pembuatan *software* dapat dilakukan lebih cepat karena *software* di bangun dari *object-object* standar, dapat menggunakan model yang ada, dan dapat membuat model dengan cepat melalui metodologi. Kualitas yang tinggi dari *software* dapat dicapai karena adanya *tested components*. Lebih mudah dari *maintenance* karena perbaikan kode hanya diperlukan pada satu tempat (bukan diurut dari awal). Mudah dalam membangun sistem yang besar karena *subsistem* dapat dibuat dan diuji secara terpisah. Mengubah sistem yang sudah ada tidak memerlukan membangun ulang keseluruhan sistem.

2.7.2. Alat Bantu UML

Sebagaimana telah dijelaskan diatas tentang UML, bahwa untuk mendapatkan banyak pandangan terhadap sistem informasi yang akan dibangun, UML menyediakan beberapa alat bantú diagram visual yang menunjukan berbagai aspek dalm sistem. Ada beberapa alat bantu diagram yang disediakan dalam UML antara lain :

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan :bagaimana". Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah *entitas* manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case* diagram dapat sangat membantu bila kita sedang menyusun *requitment* sebuah sistem. Mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extand use case* lain dengan *behavior*-nya sendiri. Sementara hubungan *generalisasi* antar *use case*

menunjukan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2. *Class Diagram*

Class diagram adalah sebuah spesifikasi yang diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan *(atribut/property)* suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok *:*

1. Nama (dan *stereotype*)
2. *Attribute*
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar class yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja.

Class dapat merupakan implementasi dari sebuah *interface,* yaitu *class abstrak* yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time,* sesuai dengan perkembangan *class* model, *class* dapat dikelompokan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package.*

Hubungan antar class adalah sebagai berikut :

1. *Asosiasi,* yaitu hubungan statis *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa, *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukan arah *query* antar *class*.
2. *Agregasi*, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
3. Pewarisan. Yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya dan menambah fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah *generalisasi*.

4. Hubungan dinamis, yaitu rangkaian pesan *(messege)* yang di-passing dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence* yang akan dijelaskan kemudian.

3. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses pararel yang mungkin terjadi pada beberapa eksekusi. *Activity* diaram merupakan *state* diagram khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya *(internal processing).* Oleh menggambarkan proses-proses dan jalur-jalur aktivitas dari atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagian *actor* menggunakan sistem untuk melakukan aktivitas. Sama seperti state, standar UML menggunkan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan

behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses pararel *(fork dan join)* digunakan titik singkronisasi yang dapat berupa titik, garis horizontal atau vertical. *Activity* diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

4. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *messege* yang digambarkan terhadap waktu. *Sequence* diagram terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence* diagram biasa digunakan untuk menggambarkkan skenario atau rangkaian langkah-langkah yang dilakukan sebagi *respons* dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk *actor*, memiliki *lifeline* vertical. *Messege* digambarkan sebagi garis berpanah dari satu objek ke objek lainya. Pada fase desain berikutnya. *Messege* akan dipetakan menjadi operasi/metoda dari *class*.

5. *Collaaboration Diagram*

   Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence* diagram, tetapi lebih menekankan pada masing-masing objek dan bukan pada waktu penyampaian *messege*. Setiap *messege* memiliki *sequence number*, dimana *messege* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki *refiks* yang sama.

6. *Component Diagram*

   Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan *(dependency)* diantaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *liblary* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time.* Umumna komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

7. *Deployment Diagram*

   Deployment diagram menggambarkan *detail* bagaimana komponen di-*deploy* dalam infrastruktur sistem,

dimana komponenn akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringann pada lokasi tersebut, spesifikasi *server*, daan hal-hal lain yang bersipat fisikal. Sebuah *node* adalah *server*, *Workstation*, atau piranti keras lain yang digunakan untuk, men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP *requirement* dapat juga didefinisikan dalam diagram ini.