BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem, Informasi dan Sistem Informasi

Berikut adalah rincian dan penjelasan mengenai Konsep Dasar Sistem, Informasi dan Sistem Informasi yang dikutip dari Jogiyanto Hartono. 2005. *Analisis dan Desain Sistem Informasi* , yaitu :

2.1.1 Konsep Dasar Sistem

Istilah sistem bukanlah hal yang asing bagi kebanyakan orang. Suatu sistem terdiri dari sistem-sistem bagian *(subsistem)* dan masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil lagi. Berikut penjelasan mengenai sistem.

2.1.1.1 Definisi Sistem

Untuk dapat memahami atau dapat mendefinisikan sebuah sistem terdapat dua pendekaan yang dapat digunakan untuk menerangkannya yaitu dengan pendekatan :

1. Prosedur

   Yaitu : “Suatu jaringan kerja dari prosedur-prosedur yang berupa urutan kegiatan yang saling berhubungan, berkumpul bersama-sama untuk mencapai tujuan tertentu“.

Prosedur adalah "Rangkaian operasi klerikal (tulis menulis), yang melibatkan beberapa orang di dalam suatu/lebih departemen yang digunakan untuk menjamin penganganan yang seragam dari transaksi-transaksi yang terjadi serta untuk menyelesaikan suatu kegiatan tertentu".

Urutan kegiatan digunakan untuk menjelaskan apa *(what)* yang harus dikerjakan, siapa *(who)* yang mengerjakannya, kapan *(when)* dikerjakan dan bagaimana *(how)* mengerjakannya.

2. Komponen/elemen

   Yaitu : "Kumpulan komponen yang saling berkaitan dan bekerja sama untuk mencapai suatu tujuan tertentu".

Menurut Jogiyanto (2005 : 2) yang dimaksud dengan sistem adalah suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan.

2.1.1.2 Karakteristik Sistem

Menurut Jogiyanto (2005 : 3) Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu mempunyai :

a. Komponen *(components)*

Komponen-komponen dapat terdiri dari beberapa subsistem/subbagian, dimana setiap subsistem tersebut memiliki fungsi khusus dan akan mempengaruhi proses sistem secara keseluruhan.

b. Batas Sistem *(boundary)*

Batas sistem merupakan daerah yang membatasi antara satu sistem dengan system lainnya/dengan lingkungan luarnya. Batas system ini memungkinkan suatu sistem dipandang sebagai satu batas suatu sistem menunjukkan ruang lingkup *(scope)* dari sistem tersebut.

c. Lingkungan Luar Sistem *(environments)*

Lingkungan luar sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar dapat bersifat menguntungkan dan merugikan. Lingkungan yang menguntungkan harus tetap dijaga dan dipelihara, sebaliknya lingkungan yang merugikan harus ditahan dan dikendalikan jika tidak ingin terganggu kelangsungan hidup sistem.

d. Penghubung *(interface)*

Penghubung merupakan media penghubung antar subsistem, yang memungkinkan sumber-sumber daya mengalir sari satu subsistem ke subsistem lainnya. Keluaran *(output)* dari satu subsistem akan menjadi masukan *(input)* untuk subsistem lainnya melalui penghubung disamping sebagai untuk mengintegrasikan subsistem-subsistem menjadi satu kesatuan.

e. Masukan *(input)*

Masukan adalah energi yang dimasukkan ke dalam sistem, yang dapat berupa masukan perawatan *(maintenance input)* dan masukan sinyal *(signal input)*. Masukan perawatan : energi yang dimasukkan supaya system dapat beroperasi, sedangkan masukan sinyal adalah energi yang diproses untuk mendapatkan keluaran. Sebagai contoh di dalam sistem komputer, program adalah

maintenance input yang digunakan untuk mengoperasikan komputer dan data adalah *signal input* untuk diolah menjadi informasi.

f. Keluaran *(output)*

Keluaran adalah hasil dari energi yang diolah dan diklsifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain. Misalnya untuk sistem komputer, panas yang dihasilkan merupakan keluaran yang tidak berguna dan hasil sisa pembuangan sedangkan informasi adalah keluaran yang dibutuhkan.

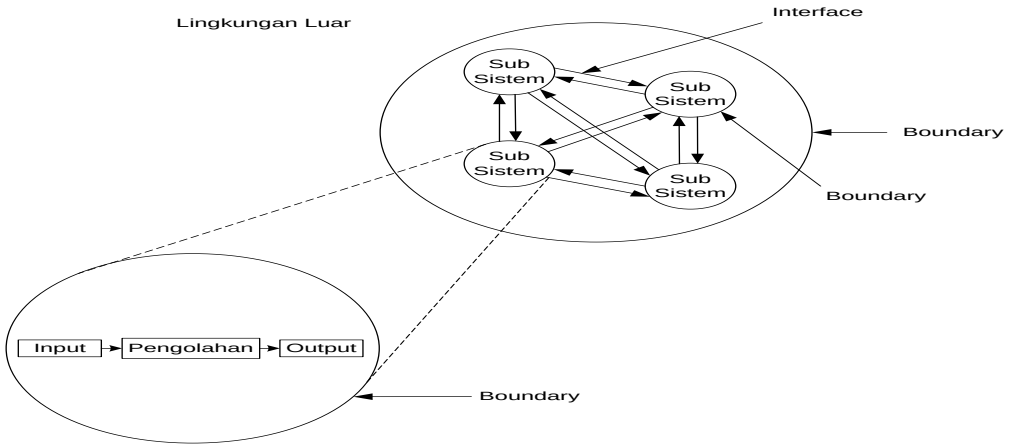
g. Pengolah *(process)*

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan mengubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan lain yang dibutuhkan oleh manajemen.

h. Sasaran *(objectives)* atau Tujuan *(goal)*

Suatu sistem pasti mempunyai tujuan dan sasaran. Jika suatu 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. Suatu sistem dikatakan berhasil bila mengenai sasaran/tujuannya.

Karakteristik atau sifat-sifat suatu sistem dapat kita lihat seperti pada Gambar 2.1 berikut ini :



Gambar 2.1 Karakteristik Sistem

(Sumber : Jogiyanto 2005)

2.1.1.3 Klasifikasi Sistem

Menurut Jogiyanto (2005 : 6) Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

a. Sistem diklasifikasikan sebagai sistem abstrak *(abstract system)* dan sistem phisik *(physical system)*.
   - Sistem Abstrak adalah sistem yang berupa pemikiran/ide yang tidak tampak secara fisik. Contoh : Sistem Teologia, Organisasi.
   - Sistem Fisik adalah sistem yang dapat dilihat wujudnya. Contoh : Kendaraan bermotor.

b. Sistem diklasifikasikan sebagai sistem alamiah *(natutal system)* dan sistem buatan manusia *(human made system)*.
   - Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Contoh : Sistem Rotasi Bumi.
   - Sistem buatan manusia adalah sistem yang dirancang oleh manusia yang melibatkan interaksi dengan mesin. Contoh : Sistem Informasi, Sistem Informasi Manajemen, dll.

c. Sistem diklasifikasikan sebagai sistem tertentu *(deterministic system)* dan sistem tak tentu *(probabilistic system)*.
   - Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Contoh : Sistem Komputer.
   - Sistem tak tentu adalah sistem yang kondisi depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

d. Sistem diklasifikasikan sebagai sistem tertutup *(closed system)* dan sistem terbuka *(open system)*.
   - Sistem dikatakan tertutup bila aktivitas didalam sistem tersebut tidak terpengaruh oleh perubahan yang terjadi di lingkungannya.
   - Sistem dikatakan terbuka menurut bila aktivitas didalam sistem tersebut dipengaruhi oleh lingkungannya.

2.1.2 Konsep Dasar Informasi

Berikut adalah uraian singkat mengenai informasi yang merupakan salah satu sumber daya penting dalam manajemen modern atau dalam suatu sistem atau organisasi.

2.1.2.1 Definisi Informasi

Menurut Jogiyanto (2005 : 8) yang dimaksud dengan informasi adalah hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian *(event)* yang nyata *(fact)* yang digunakan untuk pengambilan keputusan.

Sumber informasi adalah data. Data adalah bentuk jamak dari bentuk tunggal atau data-data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Data sering kali disebut sebagai bahan mentah informasi. Melalui suatu proses transformasi, data dibuat menjadi bermakna.

2.1.2.2 Kualitas Informasi

Istilah kualitas informasi *(quality of information)* terkadang juga digunakan untuk menyatakan informasi yang baik. Kualitas dari suatu informasi sering kali diukur dan tergantung pada tiga hal, yaitu :

a. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai

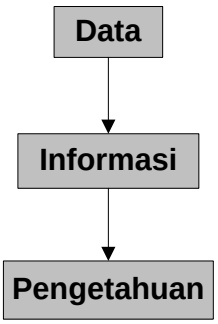
ke penerimanya, informasi kemungkinan banyak terjadi gangguan *(noise)* yang dapat merubah atau merusak informasi tersebut.

b. Tepat pada waktunya, berarti informasi yang datang pada penerimanya tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan.

c. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk setiap orangsatu dengan yang lainnya berbeda.

2.1.2.3 Hirarki Informasi

Informasi memiliki urutan atau tingkatan yang disebut sebagai hirarki. Makna informasi, yaitu tingkatan informasi dilihat dari kegunaan atau asas manfaatnya.



Gambar 2.2 Hirarki Informasi

(Sumber : Jogiyanto, 2004)

Dari gambar diatas, dalam hirarki informasi terdapat 3 (tiga) bagian utama, yaitu :

1. Data, yaitu gambaran dari suatu kejadian nyata.

2. Informasi, yaitu suatu hasil dari pengolahan data yang memiliki nilai guna bagi penerima yang membutuhkan.

3. Pengetahuan, yaitu nilai lebih yang didapat dari sebuah informasi.

2.1.3 Konsep Dasar Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi suatu manajemen di dalam pengambilan keputusan. Informasi tersebut dapat diperoleh dari suatu sistem informasi *(information system)* atau disebut juga dengan *processing systems* atau *information processing systems* atau *information-generating systems.* Berikut adalah uraian mengenai sistem informasi.

2.1.3.1 Definisi Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, member sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdik. Jogiyanto (2005 : 11)

Menurut Jogiyanto (2005: 12) faktor-faktor yang menentukan kehandalan dari suatu sistem informasi atau informasi dapat dikatakan baik jika memenuhi kriteria-kriteria sebagai berikut :

1. Keunggulan (*usefulness*)

Keunggulan bererti suatu sistem harus dapat menghasilkan informasi yang tepat dan relevan untuk mengambil keputusan manajemen dan personil operasi dalam organisasi.

2. Ekonomis

   Kemampuan sistem yang mempengaruhi sistem harus bernilai manfaat minimal sebesar biayanya.

3. Kehandalan (*Reliability*)

   Keluaran dari sistem harus mempunyai tingkat ketelitian tinggi dan sistem tersebut harus beroperasi secara efektif.

4. Pelayanan (*Customer Service*)

   Suatu sistem memberikan pelayanan yang baik dan efisien kepada para pengguna sistem pada saat berhubungan dengan organisasi.

5. Kapasitas (*Capacity*)

   Setiap sistem harus mempunyai kapasitas yang memadai untuk menangani setiap periode sesuai yang dibutuhkan.

6. Sederhana dalam kemudahan (*Simplicity*)

   Sistem tersebut lebih sederhana (umum) sehingga struktur dan operasinya dapat dengan mudah dimengerti dan prosedur mudah diikuti.

7. Fleksibel (*Fleksibility*)

   Sistem informasi ini harus dapat digunakan dalam kondisi yang bagaimana yang diinginkan oleh organisasi tersebut atau pengguna tertentu.

2.1.3.2 Komponen Sistem Informasi

Menurut Jogiyanto (2005 : 12) sistem informasi terdiri dari 6 (enam) komponen yang disebut blok bangunan *(building block)* yang saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya. Enam komponen tersebut antara lain :

a. 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 dimasukan, yang dapat berupa dokumen-dokumen dasar.

b. Blok Model *(model block)*

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

c. 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.

d. Blok Teknologi *(technology block)*

Teknologi merupakan "kotak alat" *(tool-box)* dari pekerjaan sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan

membantu pengendalian dari sistem keseluruhan. Teknologi terdiri dari 2 bagian utama, yaitu perangkat lunak *(software)* dan perangkat keras *(hardware)*.

e. Blok Basis Data *(database block)*

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Organisasi basis data yang baik berguna untuk efisiensi kapasitas penyimpannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS *(Database Management System).*

f. Blok Kendali *(control block)*

Untuk supaya sistem informasi dapat berjalan sesuai dengan yang diinginkan, maka perlu diteraokan pengendalian-pengendalian di dalamnya. Beberapa pengendali 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.

2.2 Definisi dari Kasus Yang Dianalisis

Dalam suatu penelitian pada perusahaan tidak lepas dari kasus yang di analisis pada perusahaan tersebut. Berikut adalah beberapa definisi dari kasus yang di analisis.

2.2.1 Definisi Penjualan

Dalam dunia bisnis, penjualan adalah proses bisnis yang penting dan tidak bisa dihilangkan. Penjualan adalah proses, cara, membuat menjual barang. [1]

2.2.2 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.

2.3 Definisi Internet, Website dan *World Wide Web (WWW)*

Dalam penelitian ini dibutuhkan beberapa teori yang mendukung dalam penyajian informasi, hal ini dianggap perlu untuk memudahkan dalam pembuatan sistem informasi khususnya yang berbasis *web.*

2.3.1 Definisi Internet

Internet atau *Interconnection Networking* mempunyai arti hubungan berbagai komputer dan berbagai tipe *(platform)* komputer yang membentuk sistem jaringan yang mencakup seluruh dunia dengan melalui jalur telekomunikasi, seperti : telpon, wireless, bahkan teknologi satelit. [2]

2.3.2 Definisi *Website*

Situs *web* atau sering disebut *website* merupakan suatu layanan yang memberikan informasi kepada semua pengunjung *website* dengan menggunakan konsep *hyperlink* yang bertujuan untuk mempermudah melakukan penelurusan informasi di internet.

Informasi yang bisa didapat dari suatu *website* sangat beraneka ragam, seperti : teks, gambar, imajinasi, suara dan bentuk informasi lain. [3]

2.3.3 Definisi *WWW (World Wide Web)*

WWW (World Wide Web) adalah sebuah ruang informasi yang sangat berguna dan diakses oleh *user* melalui *browser*, dengan adanya *www* kita bisa menerima berbagai macam infomasi dalam berbagai format. *WWW* sering disebut juga sebagai protokol pada sebuah URL *(Uniform Resource Locator).*[4]

Browser merupakan suatu aplikasi atau program yang berfungsi untuk menampilkan informasi bak teks, gambar, suara, animasi, bahkan sekarang telah banyak bermunculan *video streaming* untuk menampilkan acara televisi dan musik. *Browser* pertama kali dibuat berbasiskan teks. Untuk menyatakan suatu alamat *(link)*, dibuat sebaris nomor yang mirip dengan suatu menu.

2.4 Arsitektur Aplikasi

Arsitektur aplikasi terdiri dari definisi jaringan komputer, jenis-jenis jaringan komputer, topologi jaringan komputer dan manfaat jaringan komputer.

2.4.1 Definisi Jaringan Komputer

Jaringan komputer *(computer network)* adalah sebuah sitem, yang terdiri atas komputer, software dan perangkat jaringan lainnya yang bekerja sama untuk mencapai suatu tujuan yang sama. [5]

2.4.2 Jenis-Jenis Jaringan Komputer

Berikut adalah jenis-jenis jaringan komputer menurut rentang geografis.

1. LAN *(Local Area Network)*

   LAN adalah jaringan komputer yang mencakup area dalam satu ruang, satu gedung atau beberapa gedung yang berdekatan. Sebagai contoh, jaringan dalam satu kampus yang terpadu atau di sebuah lokasi perusahaan tergolong sebagai LAN. LAN umumnya menggunakan media transmisi berupa kabel. Namun, ada juga yang tidak menggunakan kabel dan disebut sebagai *wireless LAN*. Kecepatan LAN berkisar dari 10 Mbps – 1 Gbps.

2. MAN *(Metropolitan Area Network)*

   MAN adalah jaringan yang mencakup area satu kota atau dengan rentang sekitar 10 – 45 km. Jaringan yang menghubungkan beberapa Bank yang terletak di dalam satu kota atau kampus yang tersebar dalam beberapa lokasi tergolong sebagai MAN. Jaringan seperti ini umumnya menggunakan media transmisi dengan mikrogelombang atau gelombang radio.

3. WAN *(Wide Area Network)*

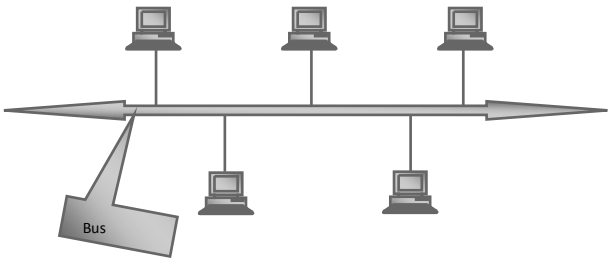
   WAN adalah jaringan yang mencakup antarkota, antarprovinsi, antarnegara dan bahkan antarbenua. Contoh : Internet.

2.4.3 Topologi Jaringan Komputer

Topologi jaringan adalah gambaran secara fisik dari pola hubungan antara komponen-komponen jaringan, yang meliputi *server, workstation, hub* dan pengkabelannnya. Secara garis besar dapat dibagi menjadi tiga, yaitu : bus, *star* dan ring/cincin.

1. *Linear Bus* (Garis Lurus)

   Topologi *Linear Bus* (Garis Lurus) satu kabel utama menghubungkan tiap simpul ke saluran tunggal komputer. Masing-masing simpul dihubungkan ke dua simpul lainnya, kecuali simpul yang terletak diujung. Jaringan-jaringan *Ethernet* dan *Local Talk* menggunakan topologi linear ini.

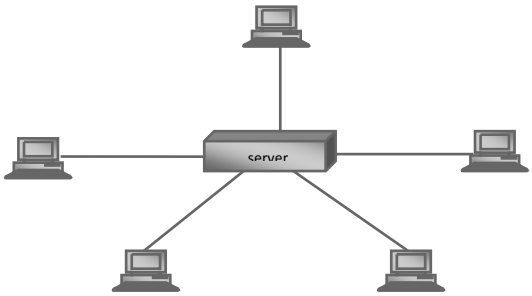


(Sumber: Adi Nugroho, 2005: 205)

Gambar 2.4 Topologi Jaringan komputer Linear *Bus*

2. *Star* (Bintang)

   Tiap simpul terhubung ke file server tunggal terpusat dengan menggunakan segmen kabel sendiri. Topologi ini memiliki keunggulan dalam minimalnya lintas data (hanya dari simpul ke server) sehingga dapat di capai kinerja yang optimal. Tetapi karena hanya ada satu server yang harus mengkoordinir seluruh komunikasi data maka topologi ini memerlukan server yang kuat dan tentu saja mahal.

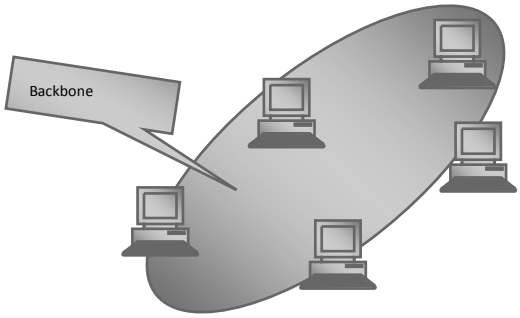


***(Sumber:* Adi Nugroho, 2005*: 207*)**

Gambar 2.5 Topologi Jaringan komputer *Star*

3. *Ring* (Cincin)

   Topologi ini mirip dengan topologi linier bus, kecuali simpul terhubung dalam satu lingkaran dengan menggunakan segmen kabel. Dalam topologi ini, tiap simpul secara fisik hanya terhubung hanya ke dua simpul lain. Masing-masing simpul mengirim informasi ke simpul berikutnya hingga tiba di sasaran.



(Sumber: **Adi Nugroho, 2005*****: 206)***

Gambar 2.6 Topologi Jaringan komputer *Ring*

2.4.4 Manfaat Jaringan Komputer

Manfaat yang didapat dari membangun jaringan komputer adalah sebagai berikut:

1. *Sharing Resources*

   Sharing resources bertujuan agar seluruh program, peralatan/periperal lainnya dapat dimanfaatkan oleh setiap orang yang ada pada jaringan komputer tanpa terpengaruh pada jaringan komputer tanpa terpengaruh oleh lokasi maupun pengaruh dari pemakai. Dengan kata lain, seorang pemakai yang letaknya jauh sekalipun dapat memanfaatkan data maupun informasi yang lainnya tanpa mengalami kesulitan.

2. Media Komunikasi

   Jaringan komputer memungkinkan terjadinya komunikasi antara pengguna, baik untuk *teleconference* maupun untuk mengirim pesan/ informasi yang penting lainnya.

3. Integrasi Data

   Pembangunan jaringan komputer dapat mencegah ketergantungan pada komputer pusat. Setiap proses data tidak harus dilakukan satu komputer saja, melainkan dapat didistribusikan ke tempat lainnya oleh sebab inilah maka dapat terbentuk data yang terintegrasi sehingga dengan demikian memudahkan pemakai untuk memperoleh dan mengolah informasi setiap saat.

4. Pengembangan dan Pemeliharaan

   Dengan adanya jaringan komputer ini, maka perkembangan peralatan dapat dilakukan dengan mudah dan menghemat biaya. Jaringan komputer bisa memudahkan pemakai dalam merawat *harddisk* dan peralatan lainnya. Misalnya untuk memberikan perlindungan serangan *virus* maka pemakai cukup memusatkan perhatian pada *harddisk* yang ada di komputer pusat.

5. Keamanan Data

   Sistem jaringan komputer memberikan perlindungan terhadap data jaminan keamanan data tersebut di berikan melalui pengaturan hak akses para pemakai dan password, serta teknik perlindungan terhadap harddisk sehingga data mendapatkan perlindungan yang efektif.

6. Sumber Daya Lebih Efisien dan Informasi Terkini

   Dengan adanya pemakaian sumber daya secara bersama-sama maka pemakai bisa mendapatkan hasil dengan maksimal informasi yang di akses

selalu terbaru, karena setiap ada perubahan yang terjadi dapat secara langsung di ketahui oleh setiap pemakai.

2.5 Perangkat Keras Pendukung

Menurut Adi Nugroho (2005 : 213) beberapa komponen perangkat keras yang mempengaruhi sistem informasi adalah :

1. Motherboard adalah sirkuit utama dimana komponen-komponen lainnya seperti CPU, RAM, kartu grafis, kartu suara, kartu jaringan, dan sebagainya ditancapkan.

2. CPU (Central Processing Input) adalah otak komputer, naik jika komputer tersebut berfungsi sebagai work station maupun server. Tanpa CPU tidak mungkin kita mengharapkan suatu pekerjaan akan dapat dikerjakan oleh komputer.

3. Memory (RAM / Random Access Memory) setiap aplikasi pada saat pemrosesan oleh CPU dilakukan, akan diletakkan dalam memori internal (RAM) komputer yang bersifat sementara (volatile) agar pemrosesan berlangsung dengan lebih cepat.

4. Layar Monitor, bagian untuk menampilkan program.

5. Hardisk, Media penyimpanan data.

6. Peripheral Tambahan (Mouse dan disk drive), mouse adalah alat bantu petunjuk, dan disk drive adalah media penyimpanan lain selain hardisk.

7. Modem, alat penghubung komputer dengan jalur telepon.

8. Printer, alat cetak dokumen

9. NIC (Network Interface Card), peralatan yang menandakan alamat dalam jaringan.

2.6 Perangkat Lunak Pendukung

Komputer tidak berguna tanpa keberadaan perangkat lunak (*software*). Komputer bekerja atas dasar instruksi. Sekumpulan instruksi diberikan untuk mengendalikan perangkat keras komputer. Sekumpulan instruksi inilah yang dikenal dengan sebutan program atau program komputer. Secara lebih umum, program komputer inilah yang disebut perangkat lunak.

Perangkat lunak dikelompokkan menjadi dua, yaitu program aplikasi (*application program*) dan program sistem (*system program*). Program aplikasi adalah program yang dibuat oleh pemakai yang ditujukan untuk melakukan suatu tugas khusus. Program seperti ini biasa dikelompokan menjadi dua, yaitu program aplikasi serbaguna dan program aplikasi spesifik.

Sedangkan, program sistem adalah program yang digunakan untuk mengontrol sumber daya komputer, seperti CPU dan piranti masukan/keluaran. Kedudukan program ini adalah sebagai perantara antara program aplikasi dan perangkat keras komputer. Itulah sebabnya, peran program sistem sering kali tidak terlihat secara langsung. Program sistem dapat dikelompokkan menjadi tiga golongan, yaitu program pengendalian sistem, program pendukung sistem dan program pengembangan sistem.

2.6.1 Dreamweaver 8

Dreamweaver merupakan salah satu *software* dari kelompok Macromedia yang banyak digunakan untuk mendesain situs *web*. Adapun macromedia dreamweaver itu sendiri adalah sebuah HTML editor profesional yang berfungsi untuk mendesain secara visual dan mengelola situs web maupun halaman web.

Dreamweaver 8 merupakan versi terbaru yang memiliki performa yang lebih baik dan memiliki tampilan yang memudahkan siapapun untuk membuat halaman *web*, baik ketika sedang bekerja dalam jendela desain maupun dalam jendela kode rumus. Dreamweaver 8 didukung untuk pengembangan menggunakan CSS, XML dan RSS serta kemudahan lain yang diperlukan. Bisa mendapatkan kecepatan dengan memaksimalkan penggunaan XML, tombol-tombol tampilan untuk memadukan bagian kedalam pekerjaan dan mengganti sensasi dari XML ke bahasa HTML. Siapapun juga dapat memadukan sumber data XML, seperti bagian RSS ke halaman *web* hanya menggunakan cara klik tahan geser mouse secara manual dan dapat bekerja pada tampilan kode untuk mengembangkan perubahan dengan membaca kode pertunjuk dari XML dan XSLT.

Adapun fasilitas yang ada dalam dreamweaver 8, sebagai berikut :

1. Dreamweaver 8 masih dapat bekerja meskipun file sedang di *upload* ke jaringan.
2. Desain dapat dikontrol dengan fasilitas baru *Zoom*. *Zoom In* memungkinkan untuk meneliti gambar atau pekerjaan dengan *layout table* yang lebih komplek

atau luas. Adapun *Zoom Out* untuk menampilkan contoh hasil tampilan halaman.

3. Tampilan *Guides* dengan satuan pixel yang sempurna sangat membantu mendapatkan ukuran yang lebih akurat dalam mengukur layout tampilan.
4. *Coding Toolbar*, toolbar baru ini ada pada jendela kode yang digunakan untuk mempermudah pekerjaan pada jendela tersebut.
5. Tampilan baru dalam dreamweaver 8 membuat pekerjaan lebih cepat selesai.
6. *Paste special*, dapat mengambil teks dari Microsoft Word beserta formatnya.

2.6.2 PHP

PHP (*Personal Home Page tool*) adalah bahasa pemrograman *server side* yang sudah banyak digunakan saat ini, terutama untuk pembuatan *website* dinamis. Untuk hal-hal tertentu dalam pembuatan *web*, bahasa pemrograman PHP memang diperlukan, misalnya saja untuk memproses data yang dikirimkan oleh pengunjung *web*. Contoh nyata adalah penggunaanya untuk buku tamu halaman login. Agar bisa berjalan dengan baik maka PHP harus dikolaborasikan dengan *database engine*, misalnya MySQL, PostgresSQL dll.

Pada saat ini banyak *Website* yang menggunakan program PHP sebagai dasar pengolahan data. Beberapa keunggulan yang dimiliki program PHP adalah :

1. Tingkat akses PHP yang lebih cepat dan memiliki tingkat keamanan yang tinggi.
2. PHP memiliki tingkat *Lifecycle* yang cepat, sehingga selalu mengikuti perkembangan terknologi *Internet*.

3. PHP mampu berjalan dibeberapa *Server* yang ada, misalnya *Apache, Microsoft IIS, PWS, AOLserver, phttpd, fhttpd* dan *Xitami.*
4. PHP mampu berjalan di *Linux* sebagai *Platform* system operasi utama bagi PHP, tetapi juga dapat berjalan di *FreeBSD*, *Unix*, *Solaris, Windows* dan yang lain.
5. PHP juga mendukung akses kebeberapa *Database* yang sudah ada, baik yang bersifat *free*/gratis ataupun komersial. *Database* itu antara lain *MySQL, PostgreSQL, mSQL, Informix, Microsoft SQL Server*.
6. PHP bersifat *free*/gratis.

2.6.3 MySQL

MySQL merupakan *software* sistem manajemen *database* (*Database Management System*-DBMS) yang sangat populer dikalangan programer *web*, karena MySQL merupakan *database* yang stabil dalam penyimpanan data. MySQL mampu menangani data yang cukup besar yaitu sekitar 40 *database* dan 10.000 tabel serta jutaan baris yang bisa ditampung oleh MySQL.

Selain kemampuan dalam penyimpanan data yang stabil, MySQL juga telah mendukung *Relasional Database Manajemen Sistem* (RDBMS). MySQL merupakan *software* yang sifatnya gratis artinya tidak dikenakan biaya.

2.6.4 Apache

Apache adalah *web* browser *open source* yang tersedia di berbagai *platform*, termasuk Linux dan Windows. Apache dikembangkan berdasarkan kode-kode dan ide dari NCSA httpd 1.3. apache merupakan server HTTP yang paling terkenal pada awal tahun 1995.

Nama apache merupakan singkatan dari "*A PATCHY SERVER*" yang kira-kira berarti *server* yang dibangun dari kode-kode yang telah ada dan file *patch*. Menurut *Neteraft* survey, Apache saat ini adalah server yang paling popular dipakai dalam internet.

2.6.5 HTML *(HyperText Markup Language)*

HTML adalah suatu bahasa pemrograman yang dipergunakan untuk menyusun dan membentuk suatu dokumen agar dapat ditampilkan pada program *browser world wide web* dalam bentuk yang dikehendaki oleh pembuatnya.

HTML juga dipergunakan untuk menciptakan *hypertext link* atau hubungan antara teks dan dokumen lain atau juga bagian dari dokumen lain, HTML hanya berupa file teks yang bisa dibaca dengan suatu editor teks biasa, sesuai dengan namanya bahasa ini menggunakan tanda (*markup*) untuk menandai perintah-perintahnya.

2.6.6 Rational Rose

Rational Rose adalah kakas *(tools)* pemodelan visual untuk pengembangan sistem berbasis objek yang sangat handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem. Adi Nugroho (2005 : 21)

Rational Rose digunakan untuk melakukan pemodelan sebelum pengembang menulis kode-kode dalam bahasa pemrograman tertentu. Pemodelan adalah cara melihat sistem dari berbagai sudut pandang. Ia mencakup semua

diagram yang dikenal dalam UML, aktor-aktor yang terlibat dalam sistem, *use-case*, objek-objek, kelas-kelas, komponen- komponen, serta simpul- simpul penyebaran *(deployment node).* Model juga mendeskripsikan rincian yang diperlukan sistem dan bagaimana ia akan bekerja, sehingga para pengembang dapat menggunakan model itu sebagai *blue-print* untuk sistem yang akan dikembangkan.

2.7 Pendekatan Berorientasi Objek

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

Suatu cara baru dalam berpikir serta 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 aplikasi 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. Generlaisasi (Spesialisasi) : pembagian kelas-kelas yang berperilaku sama.
5. Polimorfisme : fungsi yang sama dapat diterapkan dan dimiliki oleh kelas-kelas berlainan.

2.6.1 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 sisitem yang dibangun menggunakan pemrograman 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 Programming*) 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 cetak biru) untuk himpunan objek sejenis. Kelas menetapkan spesifikasi perilaku (behaviors) dan atribut objek-objek tersebut. Class adalah keniskalan (abstraksi) 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 karakteristik dari classnya. Inilah yang disebut dengan inheritance (pewarisan sifat). Dengan demikian apaun attribute dan operation dari class akan dimiliki pula oleh semua objek yang diinherit/ diturunkan dari class tersebut. Sifat ini tidak hanya berlaku untuk objek terhadap class, akan tetapi juga berlaku untuk 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 melindungi 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-oriented 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" yang tersembunyi isinya, yaitu objek. Pengguna objek tidak perlu mengetahui isi dalam kotak tersebut. Untuk dapat berkomunikasi dengan objek, diperlukan pesan (messages). Secara formal kita definisikan 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 adalah karena programmer dapat mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata. Keuntungan yang lain adalah proses pembuatan software dapat dilakukan dengan lebih cepat karena software dibangun 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 dalam 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.

2.6.2 Alat Bantu UML

Sebagaimana yang telah dijelaskan diatas tentang UML, bahwa untuk mendapatkan banyak pandangan terhadap sistem informasi yang akan dibangun, UML menyediakan beberapa alat bantu diagram visual yang menunjukan berbagai aspek dalam 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* merepresentasikan 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 *requirement* 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-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2. *Class Diagram*

   Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain

berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) 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. Atribut
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 dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

Hubungan Antar *Class* adalah sebagai berikut :

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan 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 metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

3. *Activity Diagram*

   Activity diagrams 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 paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih

menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi 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 *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai 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 aktor, memiliki *lifeline* vertikal. *Message* digambarkan

sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.

5. *Collaboration Diagram*

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

6. *Component Diagram*

   Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya 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/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan

terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat 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) dan *requirement* dapat juga didefinisikan dalam diagram ini.