BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Menurut Jogiyanto Hartono (2004:683) mendefinisikan bahwa : Suatu sistem dapat terdiri dari sistem – sistem, bagian *(subsystems)*. Masing – masing subsistem dapat terdiri dari komponen – komponen. Subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai.

2.1.1 Karakteristik Sistem

Menurut Jogiyanto Hartono (2004:684) menyatakan karakteristik dari sistem adalah, sebagai berikut :

1. Komponen Sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi,yang artinya saling bekerja sama membentuk satu kesatuan. Komponen – Komponen sistem dapat berupa suatu subsistem atau bagian – bagian dari sistem. Setiap subsistem mempunyai sifat – sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem

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

sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem

   Lingkungan luar sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. lingkungan luar yang mengutungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem .

4. Penghubung Sistem

   Penghubung merupakan media perantara antara satu subsistem. Dengan sistem lainnya. Melalui penghubung ini memungkinkan sumber daya mengalir dari satu subsistem ke subsistem lainnya. Output dari satu subsistem akan menjadi input untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berinteraksi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem

   Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa *maintenance input* dan *sinyal input. Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Sinyal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem

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

7. Pengolahan Sistem

   Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

8. Sasaran Sistem

   Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau 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 atau tujuannya.

2.1.2 Klasifikasi Sistem

Menurut Jogiyanto Hartono (2004:687) menyatakan Sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya adalah sebagai berikut:

1. Sistem abstrak dan sistem fisik.

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologi, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan.

   Sedangkan, *Sistem fisik* adalah sistem yang ada secara fisik. Misalnya sistem komputer, sistem operasi, sistem penjualan, dan lain sebagainya.

2. Sistem alamiah dan sistem buatan manusia.

   Sistem alamiah adalah sistem yang terjadi karena proses alam tidak dibuat oleh manusia (ditentukan dan tunduk kepada kehendak sang pencipta alam). Misalnya sistem perputaran bumi, sistem pergantian siang dan malam, sistem kehidupan umat manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia.

   Sedangkan, Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebut dengan *man-machine system*. Sistem informasi merupakan contoh *man-machine system*. Karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. 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 dan sistem tersebut dapat diramalkan dan relatif stabil atau konstan dalam jangka waktu yang lama. Contoh: Sistem komputer.

Sedangkan, *Sistem Tak tentu* adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

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

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya walaupun sebenarnya bersifat *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).

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

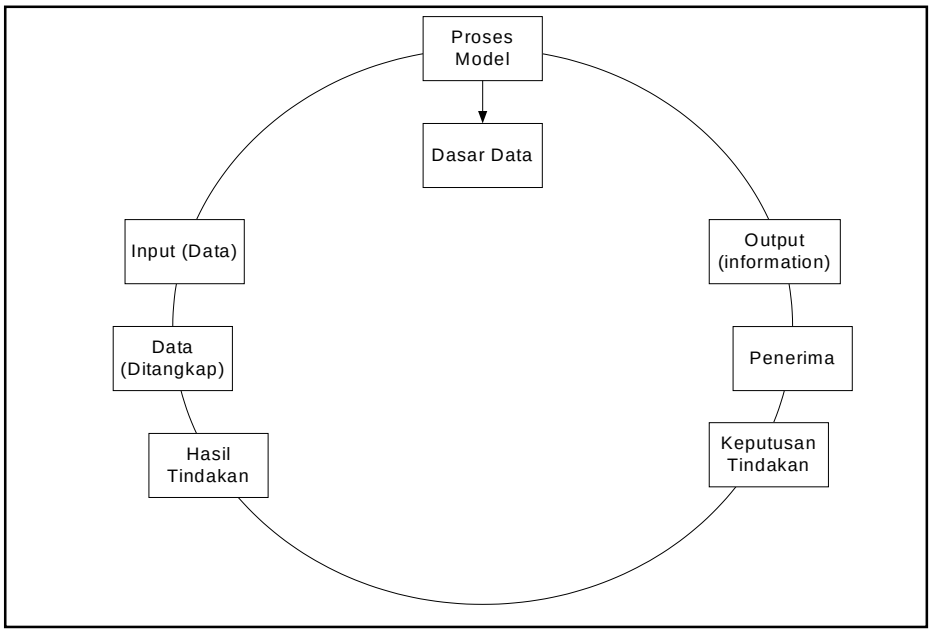
2.2 Konsep Dasar Informasi

Menurut Jogiyanto Hartono (2004:695) menyatakan Informasi dapat didefinisikan sebagai 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 dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal atau data – idem. Data adalah kenyataan yang menggambarkan suatu kejadian – kejadian dan kesatuan nyata. Kejadian – kejadian *(event)* adalah sesuatu yang terjadi pada saat tertentu sedangkan kesatuan nyata *(fact)* adalah suatu objek nyata seperti tempat, benda, dan orang yang betul – betul ada dan terjadi.

2.2.1 Siklus Informasi

Menurut Jogiyanto Hartono (2004:695) menyatakan Siklus Informasi adalah Data yang diolah melalui model menjadi informasi, penerima kemudian menerima informasi tersebut. Membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini oleh Jhon Burch disebut dengan siklus informasi *(information cycle)* atau ada yang menyebutkan dengan istilah siklus pengolahan data *(Data processing cycles)*.



Gambar 2.1 Siklus Informasi

[Sumber Jogiyanto Hartono:Pengenalan Komputer-Edisi Kedua; Penerbit Andi,2004:695]

2.2.2 Kualitas Informasi

Menurut Jogiyanto Hartono (2004:696) menyatakan Kualitas informasi tergantung dari 3 hal, yaitu informasi harus akurat, tepat waktu dan relevan :

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan masudnya.
2. Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat.

3. Relevan, berarti informasi tersebut menpunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

2.2.3 Nilai Informasi

Menurut Jogiyanto Hartono (2004:696) Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Kegunaan informasi adalah untuk mengurangi hal ketidakpastian di dalam proses pengambilan keputusan tentang sesuatu keadaan.

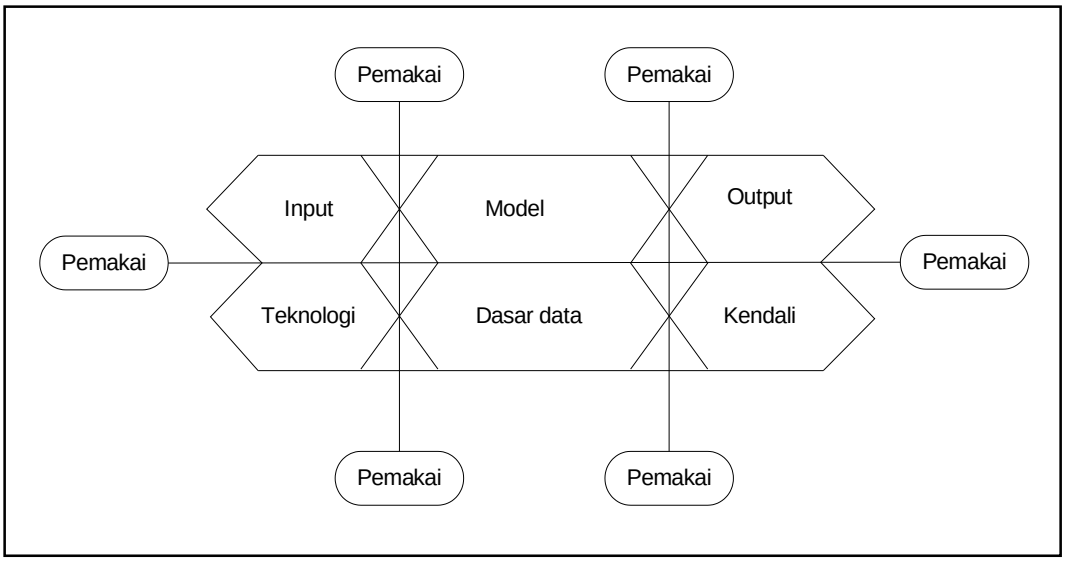
2.3 Konsep Dasar Sistem Informasi

Definisi dari Sistem Informasi sendiri ialah" Suatu Sistem terintegrasi yang mampu menyediakan Informasi yang bermanfaat bagi penggunanya.Atau Sebuah Sistem terintegrasi atau Sistem manusia-mesin, untuk menyediakan Informasi untuk mendukung operasi, manajemen dalam suatu organisasi". Sistemini memanfaatkan perangkat keras dan perangkat lunak komputer, prosedur manual,model manajemen dan basis data.

Sedangkan **Menurut Robert A. Leitch ;** Sistem Informasi adalah suatu Sistem didalam suatu 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.

2.3.1 Komponen Sistem Informasi

Sistem Informasi dapat terdiri dari komponen – komponen yang disebut dengan istilah blok bangunan *(building block)*. Sebagai suatu sistem, keenam blok tersebut masing – masing sling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.



Gambar 2.2 Blok Sistem informasi yang berinteraksi

[Sumber Jogiyanto Hartono:Pengenalan Komputer-Edisi Kedua; Penerbit Andi,2004:698]

Berikut penjelasan dari masing – masing blok bangunan, yaitu :

1. Blok Input

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

2. Blok Model

   Kombinasi prosedur, logika, dan model matemetik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diingiinkan.

3. Blok Output

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

4. Blok Teknologi

   Teknologi merupakan kotak alat (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara menyeluruh

5. Blok Basis Data

   Merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan diperangkat keras komputer dan perangkat lunak digunakan untuk memanipulasinya. Data didalam basis data perlu diorganisasikan sedemikian rupa supaya informsi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data dapat diakses atau dapat dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*database management system*).

6. Blok Kendali

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

2.3.2 Perancangan Sistem Informasi

Menurut Jogiyanto (2005 : 23), Perancangan sistem adalah tahap yang dilakukan setelah melakukan analisis sistem, pendefinisian kebutuhan–kebutuhan sistem yang akan dibangun, dan persiapan untuk merancang bangun implementasi sistem dengan menggambarkan sistem yang akan dibangun. Perancangan sistem dimulai dengan memahami sistem yang sedang berjalan dan kriteria–kriteria sistem yang akan dibangun biasanya menggunakan pemodelan secara terstruktur yang digambarkan oleh grafik atau diagram.

Hal–hal yang harus diperhatikan dalam merancang dan membangun sebuah sistem menurut Jogiyanto (2005 : 25) adalah :

1. Kebutuhan perusahaan, organisasi, atau lembaga, yaitu dengan cara memahami bidang yang akan dikembangkan, sasaran yang dibidik serta media yang akan digunakan.
2. Kebutuhan operator, yaitu kebutuhan operator untuk memperoleh sistem yang mudah dipahami dan dioperasikan serta tampilan yang interaktif. Kebutuhan operator dapat diperoleh melalui proses wawancara atau kuisioner.

3. Kebutuhan pemakai, yaitu keinginan – keinginan dari si pemakai sistem informasi, seperti jaminan keamanan, standarisasi tampilan, kecepatan akses, dan kemudahan dalam pengoperasian.
4. Kebutuhan teknis, yaitu meliputi arsitektur dan konfigurasi sistem. Secara teknis peralatan dan teknologi yang digunakan, termasuk pertimbangan penggunaan peralatan yang tidak standar, seperti peralatan multimedia, kebutuhan interface, database, dan perangkat lunaknya.

2.3.3 Pengembangan Sistem Informasi

Menurut Jogiyanto (2005 : 35) Menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem yang lama perlu diperbaiki atau diganti.

Menurut Jogiyanto (2005 : 36) Pengembangan sistem dapat berarti penyusunan suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada.

1. Adanya permasalahan-permasalahan yang timbul pada sistem yang lama. Permasalahan yang timbul dapat berupa :
   a. Ketidakberesan

      Ketidakberesan pada sistem yang lama menyebabkan sistem yang lama tidak dapat beroperasi sesuai dengan yang diharapkan.

b. Pertumbuhan Organisasi

Pertumbuhan organisasi yang menyebabkan harus disusunnya sistem yang baru. Pertumbuhan organisasi diantaranya adalah kebutuhan informasi yang semakin luas. *Volume* pengolahan data semakin meningkat, perubahan prinsip akuntasi yang baru. Karena adanya perubahan ini, maka menyebabkan sistem yang lama tidak efektif lagi sehingga sistem yang lama sudah tidak dapat memenuhi lagi semua kebutuhan informasi yang dibutuhkan oleh manajemen.

2. Untuk meraih kesempatan-kesempatan

Teknologi informasi telah berkembang dengan cepatnya. Perangkat keras komputer, perangkat lunak dan teknologi komunikasi telah begitu cepat berkembang. Organisasi mulai merasakan bahwa teknologi informasi ini perlu digunakan untuk meningkatkan penyediaan informasi sehingga dapat mendukung dalam proses pengambilan keputusan yang akan dilakukan oleh manajemen.

Dalam keadaan pasar bersaing, kecepatan informasi atau efisiensi sangat menentukan berhasil atau tidaknya strategi dan rencana-rencana yang telah disusun untuk meraih kesempatan-kesempatan yang ada. Bila pesaing dapat memanfaatkan teknologi ini, maka kesempatan-kesempatan akan jatuh ke tangan pesaing. Kesempatan-kesempatan ini dapat berupa peluang-peluang pasar, pelayanan yang meningkat kepada pelanggan dan lain sebagainya.

3. Adanya instruksi-instruksi

Penyusunan sistem yang baru dapat juga terjadi karena adanya instruksi-instrukai dari atas pimpinan atau dari luar organisasi, seperti peraturan pemerintah dan sebagainya.

Dengan dikembangkannya sistem yang baru ,maka diharapkan akan terjadi peningkatan-peningkatan ini, yaitu sebagai berikut :

1. *Performance* (kinerja)

   Peningkatan terhadap kinerja (hasil kerja) sistem yang baru sehingga menjadi lebih efektif. Kinerja dapat diukur dari :

   a. *Throughput*, yaitu jumlah dari pekerjaan yang dapat dilakukan suatu saat tertentu.
   b. *Response time*, yaitu rata-rata waktu yang tertunda diantara dua pekerjaan tersebut.

2. *Economy* (ekonomis)

   Peningkatan terhadap manfaat-manfaat atau keuntungan-keuntungan atau penurunan-penurunan biaya yang terjadi.

3. *Control* (Pengendalian)

   Peningkatan terhadap pengendalian untuk mendeteksi dan memperbaiki kesalahan-kesalahan serta keuntungan-keuntungan yang akan terjadi.

4. *Effeciency* (efisiensi)

   Peningkatan terhadap efisiensi operasi,yaitu bagaimana sumber daya digunakan dengan pemborosan yang paling minimum.

5. *Service* (pelayanan)

   Peningkatan terhadap pelayanan yang diberikan oleh sistem.

2.4 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.

2.4.1 Definisi Penjualan

Dalam dunia bisnis, penjualan adalah proses bisnis yang penting dan tidak bisa dihilangkan. Penjualan adalah proses menjual barang.

2.4.2 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.

2.5 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*.

2.5.1 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.

2.5.2 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.

2.5.3 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)*.

Browser merupakan suatu aplikasi atau program yang berfungsi untuk menampilkan informasi baik 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.

2.6 Arsitektur Aplikasi

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

2.6.1 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.

2.6.2 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)*

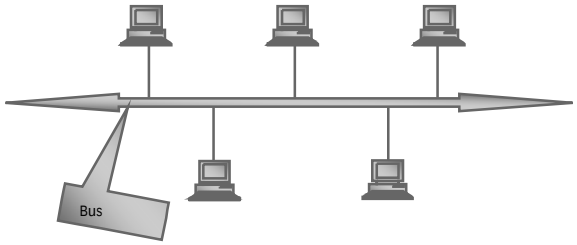
   WAN adalah jaringan yang mencakup antarkota, antarprovinsi, antarnegara dan bahkan antarbenua. Contoh : Internet.

2.6.3 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.

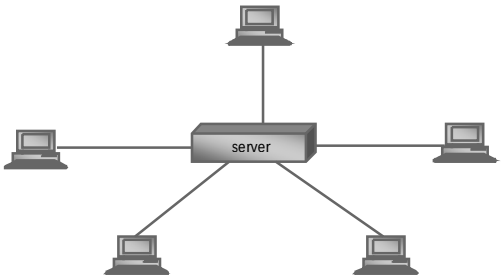


(Sumber: Adi Nugroho, 2005: 205)

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.

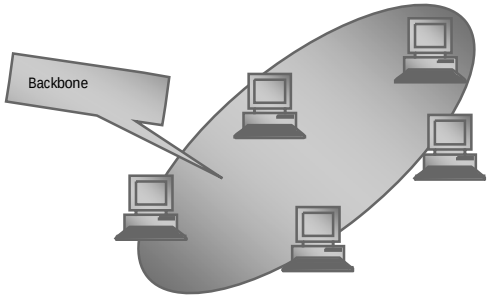


***(Sumber:* Adi Nugroho, 2005*: 207*)**

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*

2.6.4 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.

2.7 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.

2.8 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 bisa 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.

2.8.1 Adobe Dreamweaver CS4

Dreamweaver CS4 membuat lebih mudah untuk bekerja dengan situs saat ini, berpisah karena mereka antara HTML, CSS, dan file script.Tambahkan integrasi yang lebih kuat dengan Photoshop untuk membuat grafis webFitur yang lebih baik untuk bekerja dengan CSS, ditingkatkan Photoshop integrasi Live View baru dan Kode Hidup lebih akurat membuat halaman web dalam Dreamweaver.

Salah satu perkembangan menarik yang muncul setelah Dreamweaver dikirim adalah tersedianya berbagai widget dari kerangka JavaScript lainnya pada situs Bursa Adobe Web, termasuk widget berdasarkan Yahoo Antarmuka Library (YUI), jQuery, dan MooTools. Hal ini menunjukkan bahwa Adobe Dreamweaver ingin memberikan pengguna akses ke hasil kerja pengembang lain. Sebelumnya, jika Anda ingin mengambil keuntungan dari setiap kerangka JavaScript selain sigap, Anda perlu untuk mengintegrasikan dalam Tampilan Kode Dreamweaver. Sekarang, Anda dapat melakukannya tanpa perlu menggali ke

dalam kode, setelah Anda menginstal pihak ketiga widget dengan Extension Manager yang disertakan, mereka muncul di panel Insert Dreamweaver.

2.8.2 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.

2.8.3 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.

2.8.4 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.

2.8.5 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.

2.8.6 *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.

2.9 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.

2.9.1 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.

2.9.2 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.