BAB II

LANDASAN TEORI

2.1. Sistem

2.1.1 Definisi Sistem

Pengertian sistem saat ini banyak sekali karena banyak pihak medefinisikan sistem sesuai kebutuhanya, berikut merupakan pengertian sistem berdasarkan para ahli :

Menurut Andi Kristanto (2008 : 1) yang dimaksud dengan sistem adalah " jaringan kerja prosedure – prosedure yang saling berhubungan, berkumpul bersama – sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu."

Menurut Raymond Mcleod (2001 : 11) sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan.

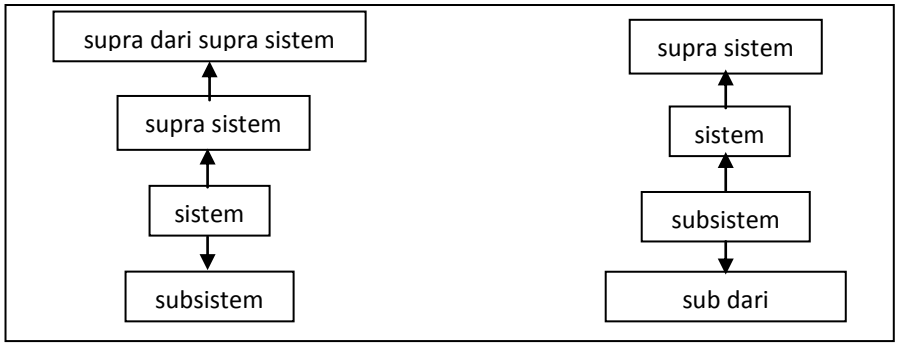
Menurut Al-barha bin ladjamudin (2005: 2) sistem yang lebih menekankan pada elemen dan komponennya adalah :

"kumpulan komponen yang saling berkaitan dan bekerja sama untuk mencapai tujuan tertentu. Suatu sistem dapat terdiri dari beberapa subsistem. Subsistem-subsistem tersebut dapat pula terdiri dari beberapa subsisten yang lebih kecil".

2.1.2 Elemen / Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu

mengandung komponen-komponen atau subsistem-subsistem. Setiap *subsistem* mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu yang lebih besar yang disebut dengan *supra system*. Misalnya suatu perusahaan dapat disebut dengan suatu sistem dan industri yang merupakan sistem yang lebih besar disebut dengan *supra system*. Kalau dipandang industri adalah suatu sistem maka perusahaan adalah *subsistem* nya.



Gambar 2.1 *Subsistem, sistem, supra sistem*

[Sumber: Jogiyanto, "Pengenalan Komputer Dasar Ilmu Komputer, Pemrograman, Sistem Informasi dan Intelegensi Buatan]

2.1.3 Klasifikasi Sistem

Menurut Tata Sutabri (2004 : 14) " sistem merupakan suatu bentuk inregrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi di dalam sistem tersebut."

Oleh karena itu sistem dapat diklasifikasikan dari beberapa sudut pandangan yang antara lain sebagai berikut :

1. Sistem Abstrak dan Sistem Fisik.

   Sistem abstrak (*abstrack system*) adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sedangkan sistem fisik (*physical system*) adalah sistem yang ada secara fisik dapat dilihat.

2. Sistem Alamiah dan Sistem Buatan Manusia.

   Sistem alamiah (*natural system*) adalah sistem yang terjadi melalui proses alam (tidak dibuat oleh manusia). Sedangkan sistem buatan manusia (*human made system*) adalah sistem yang melibatkan hubungan manusia dengan mesin, yang disebut dengan *human machine system*.

3. Sistem Deterministik dan Sistem Probabilistik

   Sistem deterministik (*deterministic system*) adalah suatu sistem yang beroperasi dengan tingkah laku yang dapat diprediksi. Sedangkan sistem yang bersifat probabilistik (*probabilistic system*) adalah sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

4. Sistem Terbuka dan Tertutup

   Sistem tertutup (*closed system*) adalah sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya, sistem ini bekerja secara otomatis tanpa ada campur tangan dari pihak luar. Sedangkan sistem terbuka (*open system*) adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

2.1.4 Karakteristik Sistem

Menurut Tata Sutabri (2004 : 12), model umum sebuah sistem terdiri *input*, proses dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifatsifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen sistem *(Component)*

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut dengan supra sistem.

2. Batasan Sistem *(Boundary)*

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan luar sistem *(Environtment)*

   Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem dapat menguntungkan dan dapat juga

merugikan sistem tersebut. Lingkungan luar yang mengutungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar sistem tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan, karena kalau tidak maka akan mengganggu kelangsungan hidup dari sistem tersebut.

4. Penghubung sistem *(Interface)*

Sebagai media yang menghubungkan sistem dengan subsistem yang lain Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melalui penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem *(Input)*

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan *(maintenance input)* dan sinyal *(signal input).* Sebagai contoh, di dalam suatu unit sistem komputer, ”program” adalah maintenance input yang digunakan untuk mengoperasikan komputer sementara ” data” adalah signal input yang akan diolah menjadi informasi.

6. Keluaran Sistem *(Output)*

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. keluaran ini merupakan masukan bagi subsistem yang lain, seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, yang mana informasi ini dapat digunakan sebagai masukan

untuk pengambilan keputusan atau hal - hal lain yang meupakan input bagi subsistem lainnya.

7. Pengolah sistem (Proses)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi, sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran sistem *(Objective)*

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada guanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.2. Sistem Informasi

2.2.1 Definisi Informasi

Menurut Davis (Abdul kadir 2003 : 31), informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.

Sedangkan menurut Jogyanto (2001 : 8), informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya.

Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum atau data-item. Data adalah kenyataan yang menggambarkan suatu kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu.

Kualitas dari suatu informasi (*quality of information*) tergantung dari tiga hal, yaitu informasi harus akurat, tepat pada waktunya dan relevan.

1) Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan.
2) Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat.
3) Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya.

2.2.2 Definisi Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Pertanyaannya adalah darimana informasi tersebut bisa didapatkan? Informasi dapat diperoleh dari sistem informasi (*information systems*) atau disebut juga dengan *processing systems* atau *information generating systems.*

Sistem Informasi menurut Jogyanto (2002 : 14) dapat didefinisikan sebagai berikut :

"Informasi adalah data yang telah diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya, alat pengolah informasi dapat meliputi elemen komputer, elemen non komputer atau kombinasinya."

Sistem Informasi didefinisikan oleh *robert* A. *Letic* dan K. *Rescoe* Davis, dalam bukunya yang berjudul *Accounting Informastion system* menurut Jogiyanto (2002 : 28) adalah sebagai berikut :

"Sistem Informasi adalah suatu sistem didalam organisasi yang mempertemukan kebutuhan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan bagi pihak luar tertentu dengan laporan yang diperlukan".

2.2.3 Kualitas Informasi

Tidak semua informasi berkualitas. Oleh karena itu, sudah seharusnya dilakukan penyaringan terhadap informasi yang beredar atau yang dapat ditangkap. Menurut Oetomo (2002) Kualitas ditentukan oleh beberapa faktor, yaitu:

1. Keakuratan dan teruji kebenaranya

Artinya informasi harus bebas dari kesalahan-kesalahan, tidak bias, dan tidak menyesatkan. Kesalahan-kesalahan itu dapat berupa kesalahan perhitungan maupun akibat gangguan *(noise)* yang dapat mengubah dan merusak informasi tersebut.

2. Kesempurnaan informasi

Untuk mendukung faktor pertama diatas, maka kesempurnaan informasi menjadi faktor penting, dimana informasi disajikan lengkap tanpa pengurangan, penambahan, atau pengubahan.

3. Tepat waktu

Informasi harus disajikan dengan tepat waktu, mengingat informasi akan menjadi dasar dalam pengambilan keputusan. Keterlambatan informasi akan mengakibatkan kekeliruan dalam mengambil keputusan.

4. Relevansi

Informasi akan memiliki nilai manfaat yang tinggi, jika informasi tersebut diterima oleh mereka yang membutuhkan, dan menjadi tidak berguna jika diberikan kepada mereka yang tidak membutuhkan.

5. Mudah dan murah

Kini, Cara dan biaya untuk memperoleh informasi menjadi bahan pertimbangan sendiri. Bilamana cara dan biaya untuk memperoleh informasi sulit dan mahal, maka orang menjadi tidak berminat untuk memperolehnya, atau mencari alternatif subtitusinya. Biaya mahal yang dimaksud disini, jika bobot informasi tidak sebanding dengan biaya yang harus dikeluarkan. Dengan melalui teknologi internet, kini orang atau perusahaan dapat memperoleh informasi dengan mudah dan murah.

2.3. Arsitektur Jaringan

Untuk mendukung terlaksananya program aplikasi yang akan dibangun ini maka diperlukan beberapa unit komputer, dima komputer tersebutn saling berhubungan dan melakukan komunikasi data agar proses komunikasi data dapat berjalan lancar dengan baik melalui media atau perantara yang disebut dengan jaringan komputer.

2.3.1 Konsep Dasar Jaringan

Jaringan komputer memungkinkan suatu organisasi untuk menggunakan sistem pengolahan data yang terdistribusi yang meggunakan PC dan dapat saling mengakses satu dengan yang lainnya.

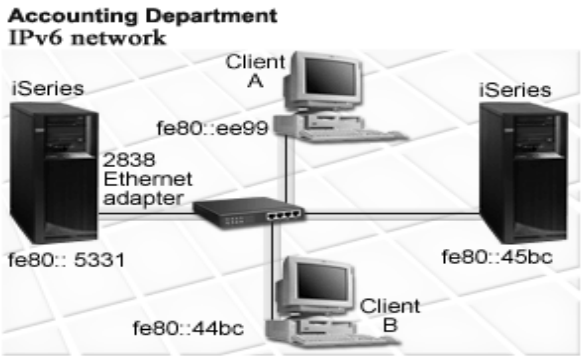
Menurut Andi Kristanto (2003 : 2) jaringan komputer merupakan " sekelompok komputer otonom yang saling berhubungan antara satu dengan yang lainnya menggunakan protokol komunikasi melalui media kounikasi sehingga dapat saling berbagi informasi, programa-program, penggunaan bersama perangkat keras seperti printer, harddisk, dan sebagainya."

2.3.2 Jenis Jaringan Komputer

Budhi Irawan (2005 : 19), membedakan jaringan komputer berdasarkan cakupan geografisnya. Ada empat kategori utaa jaringan komputer, yaitu :

1. LAN (local Area Network)

   LAN digunakan untuk menghubungkan komputer yang berbeda didalam suatu area yang kecil, misalnya di dalam suatu gedung perkantoran atau kampus. Jarak antar komputer yang dihubungkan bisa mencapai 5 sampai 10 km. Suatu LAN biasanya bekerja pada kecepatan mulai 10 Mbps sampai 100 Mbps. LAN menjadi populer karena memungkinkan banyak pengguna untuk memakai sumber daya secara bersama-sama. Contoh dari sumber daya yang digunakan itu misalnya suatu *mainframe, file server,* printer, dan sebagainya.



Gambar 2.2 Local Area Network

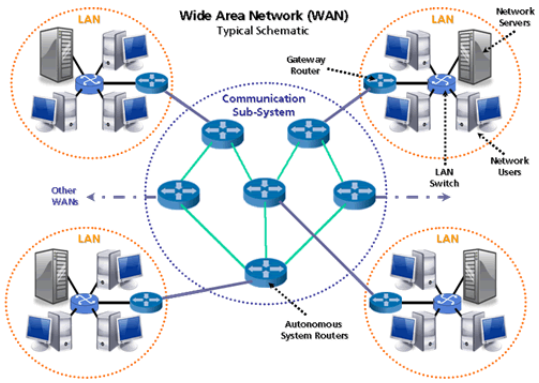
[Sumber : http://publib.boulder.ibm.com/infocenter/iseries/v5r3/topic/rzai2/rzai2500.gif]

2. MAN (Metropolitan Area Network)

   MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. MAN menghubungkan LAN-LAN yang lokasinya berjauhan. Jangkauan MAN bisa mencapai 10 km sampai beberapa ratus km. Suatu MAN biasanya bekerja pada kecepatan 1,5 sampai 150 Mbps.

3. WAN (Wide Area Network)

   WAN dirancang untuk menghubungkan komputer-komputer yang terletak pada suatu cakupan geografis yang luas, seperti hubungan dari satu kota ke kota lain di dalam suatu negara. Cakupan WAN bisa meliputi 100 km sampai 1000 km, dan kecepatan antar kota bisa bervariasi antara 1,5 Mbps sampai 2,4 Gbps. Dalam WAN, biaya untuk peralatan transmisi sangat tinggi, dan biasanya jaringan WAN dimiliki dan dioperasikan sebagai suatu jaringan publik.



Gambar 2.3 Wide Area Network

[Sumber : http://www.computernetworks.com/Images/WAN.jpg]

2.4. Definisi penjualan

Menurut Joel G. Siegel dan Joe K. Shim yang diterjemahkan oleh Moh. Kurdi (http://www.scribd.com/doc/11320689/Definisi-Penjualan) :

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

"Penjualan tunai menurut Joel G.Segel dan Joe K.Shim yang diterjemahkan oleh Moh.Kurdi adalah peneriamaan yang diperoleh dari pengiriman barang dagangan atau dari penyerahan pelayanan dalam bursa sebagai bahan pertimbangan."

2.5. Aplikasi Dan Teknologi Web

Bagian ini akan memaparkan tentang aplikasi dan teknologi web yang akan dibahas dalam beberapa *sub* bab.

2.5.1 Pengertian Internet

Internet adalah suatu jaringan komputer global yang terbentuk dari jaringan - jaringan komputer lokal dan regional yang memungkinkan komunikasi data antar komputer yang terhubung ke jaringan tersebut"

Internet awalnya merupakan suatu recana dari Departemen Pertahanan Amerika Serikat (*US Departement of Defenses*) pada sekitar tahun 1960. Dimulai dari suatu proyek yang dinamakan ARPANET atau *Advanced Research Project agency Network*. Beberapa universitas di Amerika Serikat diantaranya UCLA, Stanford, UC Santa Barbara dan Univercity of Utah, diminta bantuan dalam mengerjakan proyek ini dan awalnya telah berhasil menghubungkan empat

komputer di lokasi universitas yang berbeda tersebut. Perkembangan ARPANET ini cukup pesat dilihat perkembangan komputer pada saat itu.

Sebagai gambarannya pada tahun 1977 ARPANET telah menghubungkan lebih dari 100 mainframe komputer dan saat ini terdapat 4 juta *host* jaringan yang terhubung pada jaringan ini.

2.5.2. Istilah - istilah Dalam *Internet*

Teknologi *internet* merupakan teknologi yang sudah tidak asing lagi dalam kehidupan pada masa sekarang ini dimana pemanfaatannya sudah banyak diterapkan pada berbagai bidang. Banyak istilah – istilah dalam *internet* diantaranya :

2.5.2.1 ISP (*Internet Service Provider*)

ISP adalah suatu perusahaan atau badan baik pemerintah maupun swasta yang dapat memberikan fasilitas layanan koneksi ke jaringan Internet bagi jaringan yang terhubung kepadanya.

ISP yang ada pada saat ini merupakan dapat berupa ISP yang bersifat tertutup dan ISP yang bersifat umum. ISP yang bersifat tertutup adalah lembaga atau badan yang hanya melayani atau dapat memberikan fasilitas jaringan internet kepada jaringan lokal dari lembaga atau badan tersebut. Contohnya terdapat pada beberapa departemen pemerintahan, lembaga penelitianm perusahaan, ataupun lembaga. ISP yang bersifat umum adalah sebuah perusahaan yang dapat memberikan layanan kepada masyarakat luas baik pribadi atau lembag yang ingin terkoneksi denagn jaringan internet. ISP seperti ini biasanya bersifat komersial, yaitu menjual jasa layanan koneksi dengan jaringan internet.

2.5.2.2 Bandwidth

Bandwidth atau kecepatan transmisi data, dalam satuan bit per detik.Semakin besar *bandwidth* maka semakin cepat transmisi datanya, contoh untuk komunikasi data dengan jaringan *Ethernet* mempunyai kecepatan 10-100 Mbps.

2.5.2.3 Mail Server

Mail sever adalah komputer yang berperan sebagai ‘Kantor Pos’ bagi sebuah jaringan. Fungsi kerjanya yaitu melayani surat keluar dan surat masuk. Seperti pada contoh kasus jika seorang mengirin *E-mail* maka *E-mail* tersebut akan dikirimkan ke *server* ini.

2.5.2.4 File Transfer Protokol (FTP) Server

File Transfer Protokol (*FTP*) *Server* adalah komputer yang bertindak sebagai penampungn file-file yang dapat diakses oelh jaringan. File-file tersebut dapat di-*transfer* baik untuk disimpan pada file *server* dari jaringan (*upload*) ataupun diambil dari file *server* untuk ke komputer pada jaringan (*download*).

2.5.2.5 Domain Name Service (DNS) Server

Domain Name Service (*DNS*) *Server* adalah komputer yang berfungsi sebagai identitas nama domain yang dimilii oleh jaringan tersebut, dan juga memelihara informasi data dari domain yang lain.

2.5.2.6 Electronic Mail (E-Mail)

Adalah merupakan aplikasi pertama internet, *E-mail* ini merupakan sebuah surat atau pesan yang dapat dikirimkan melalui jaringan komputer ke antar komputer.

2.6. Perangkat Lunak Pendukung

Perangkat lunak pendukung yang digunakan adalah PHP (*Pheriperal Hyper Text Preprocessor*) dan menggunakan Mysql *Server* sebagai mengelola database.

2.6.1 PHP (*Pheriperal Hyper Text Preprocessor*)

PHP secara resmi merupakan kependekan dari *Pheriperal Hyper Text Preprocessor*, merupakan bahasa *script server side* yang disisipkan pada HTML " PHP merupakan script untuk pemrograman script *web server - side*, *script* yang membuat dokumen HTML secara *on the fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakann *editor text* atau *editor HTML*".

PHP berawal saat Rasmus Lerdorf membuat sejumlah skrip *"Perl"* yang dapat mengamati siapa saja yang melihat - lihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip - skrip ini selanjutnya dikemas menjadi *tool* yang disebut *"Personal Home Page"*. Paket inilah yang menjadi cikal bakal PHP. Pada tahun 1995 , Rasmus menciptakan *PHP/FI* Versi 2. Pada Versi inilah pemrogram dapat menempelkan kode tersruktur di dalam tag *HTML (Hypertext Markup Language)* Kode PHP juga bisa berkomunikasi dengan database dan melakukan perhitungan-perhitungan yang kompleks sambil jalan.

Pada saat ini, PHP cukup popular sebagai piranti pemrograman Web, terutama di lingkungan Linux. Walaupun demikian, PHP sebenarnya juga dapat

berfungsi pada server-server yang berbasis *UNIX, Windows NT*. PHP juga dirancang untuk diintegrasikan dengan *web server Apache.*

Dengan menggunakan PHP maka maintenance suatu situs web menjadi lebih mudah. Proses update data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan script PHP, yang akhirnya software ini disebarkan dan dilisensikan sebagai perangkat lunak *Open Source*.

2.6.2 Mysql Server

Mysql merupakan *software database* yang dikembangkan di lingkungan sistem operasi Linux yang menunjang performasi query dari databasenya yang saat ini bisa dikatakan paling cepat, dan jarang bermasalah.

2.6.3 Macromedia dreamweaver 8

Macromedia Dreamweaver 8 adalah sebuah software *web design* yang menawarkan cara mendesain *website* dengan menggunakan dua langkah sekaligus dakam satu waktu, yaitu mendesain dan memprogram. Dreamweaver memiliki satu jendela mini yang disebut *HTML, source* tempat kode-kode *HTML* tertulis. Setiap kali kita mendesain *website* seperti menulis kata-kata, meletakan gambar, membuat table, dan proses lainnya, tag-tag *HTML* akan tertulis secara langsung mengiring proses pengaturan *website.* Artinya kita memiliki kesempatan untuk mendesain *website* sekaligus mengenal lewat tag-tag *HTML* yang membangun *website* itu.

2.7. Alat Bantu Pemodelan

Pemodelan adalah sebuah aktivitas yang telah berlangsung bertahun-tahun dalam pengembangan perangkat lunak. Sebuah model masih diperlukan untuk

menulis sebuah aplikasi baik menggunakan bahasa pemograman yang paling sederhana sampai bahasa yang kompleks. Pemodelan diperlukan untuk mendefinisikan sebuah model membuat mudah untuk memecah aplikasi yang komplek atau sebuah sistem yang besar menjadi sederhana, menjadi bagian-bagian diskrit yang mudah untuk dipelajari. Dengan alasan demikian, pemodelan harus memenuhi syarat:

1. Readability

   Readability membawa kejelasan – kemudahan dalam memahami. Memahami sistem adalah langkah pertama baik dalam membangun atau mengembangkan sebuah sistem yang telah ada. Hal ini melibatkan pengetahuan tentang bagaimana sistem dibuat, bagaimana sistem berperilaku dan sebagainya. Memodelkan sebuah sistem membuat sistem mudah dibaca dan yang paling penting, mudah dalam pendokumentasian. Penggambaran sebuah sistem agar mudah dibaca melibatkan pemahaman struktur sistem dan tingkah laku sistem.

2. Reusability

   Reusability adalah efek dari membuat sebuah sistem mudah dibaca. Bila sebuah sistem telah dimodelkan sedemikian rupa sehingga mudah dipahami, karena para pengembang cenderung untuk mencari kesamaan atau pengulangan, baik dalam hal fungsionalitas, fitur ataupun struktur.

2.7.1 Unified Modeling Language (UML)

UML mengintegrasikan konsep dari *Booch*, OMT, OOSE dan juga *Class-Relation* dengan menggabungkan mereka menjadi suatu kesatuan bahasa

modeling yang bisa berguna bagi siapa saja. UML bertujuan untuk menjadi standar bahasa modeling yang mampu untuk memodelkan sistem yang konkuren dan juga terdistribusi. UML sudah menjadi standar industri yang dibuat dibawah pengawasan *Object Management Group* (OMG). OMG ditugaskan untuk memberi informasi mengenai object-oriented methodology yang mampu untuk membuat bahasa untuk modeling software. Dengan menggunakan UML para pengembang dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa pemrograman berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VisualBasic atau C.

UML bukan merupakan metode dan juga tidak berusaha untuk menjadi suatu metode, namun demikian ia merupakan kebutuhan untuk *Rational Unified Process* yang telah dibuat oleh *Rational Software Corporation*. Metode development lainnya seperti *Dynamic Systems Development Method* juga menggunakan teknik dari UML.

UML menyediakan kemampuan untuk menangkap karakteristik sebuah sistem dengan menggunakan notasi-notasi tertentu. UML menyediakan sederetan notasi sederhana yang mudah dipahami untuk mendokumentasikan sistem berdasarkan prinsip-prinsip perancangan berorientasi objek.. UML sudah menjadi *standardized modelling language* yang terdiri dari kumpulan-kumpulan diagram,

dikembangkan untuk membantu para pengembang sistem dan software agar bisa menyelesaikan tugas-tugas seperti:

1. Spesifikasi
2. Visualisasi
3. Desain Arsitektur
4. Konstruksi
5. Simulasi dan testing
6. Dokumentasi

2.7.2 Model Diagram UML

UML terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang disebut sebuah model. Pada saat ini UML telah mencapai versi 2.1 memiliki banyak perubahan dari versi sebelumnya seperti diagram kolobarasi diganti dengan diagram komunikasi dan terdapat tambahan tiga diagram.

2.7.2.1 Class Diagram

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek.

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 object beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lainlain. Sebuah *Class* memiliki tiga area pokok :

1. Nama, merupakan nama dari sebuah kelas

2. Atribut, merupakan peroperti dari sebuah kelas. Atribut melambangkan batas nilai yang mungkin ada pada obyek dari *class*

3. Operasi, adalah sesuatu yang bisa dilakukan oleh sebuah *class* atau yang dapat dilakukan oleh *class* lain terhadap sebuah *class*. Atribut dan metoda dapat memiliki salah satu sifat berikut :
    a. *Private,* tidak dapat dipanggil dari luar *class* yang bersangkutan.
    b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
    c. *Public,* dapat dipanggil oleh siapa saja.
    d. *Package,* hanya dapat dipanggil oleh *instance* sebuah *class* pada paket yang sama.

Class juga 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 *runtime*.

Relasi atau *relationship* merupakan keterhubungan antar kelas yang muncul pada saat sebuah kelas berinteraksi dengan kelas-kelas lainnya. Didalam *class diagram*, setiap kelas pasti akan berinteraksi dengan baik satu maupun lebih kelas. Relasi yang muncul pada setiap keterhubungan antar kelas juga akan memiliki atribut-atribut yang akan lebih menjelaskan definisi dari sebuah relasi yang terjadi.

1. Asosiasi *(Association)*

Para pemodel menggunakan pemahaman asosiasi adalah pada saat beberapa

kelas saling terhubung satu sama lain secara konseptual. Asosiasi juga dapat menjadi lebih kompleks pada saat beberapa class terhubung ke satu class.

2. Constraint Pada Asosiasi

Kadangkala sebuah asosiasi diantara dua *class* harus mengikuti sebuah aturan dan aturan ini bisa diletakkan dalam sebuah *constraint* pada garis asosiasi dan diletakkan dalam kurung kurawal. Bentuk lain dari tipe *constraint* adalah relasi OR yang ditulis dengan {or} dalam garis putusputus yang menghubungkan dua garis asosiasi. Kondisi OR ini menghadapkan kepada keadaan bahwa sebuah kelas terhubung dengan dua kelas tetapi pada saat mendefinisikan relasinya kelas harus memilih salah satu dari kedua kelas tersebut, sedangkan kondisi untuk memilih keduanya adalah *invalid*.

3. Kelas Asosiasi (*Associations Class*)

Sebuah asosiasi dapat memiliki atribut dan operasi seperti halnya sebuah *class*. Sebuah *association class* sebenarnya diperlukan apabila salah satu dari kelas yang terhubung mempunyai sebuah atau beberapa atribut yang tidak layak dimiliki oleh kelas tersebut, karena secara logis atribut tersebut lebih layak dimiliki oleh asosiasi yang menghubungkan kedua kelas tersebut.

4. Multiplisitas (*Multiplicity*)

Multiplisitas atau kardinalitas adalah jumlah banyaknya obyek sebuah *class* yang berelasi dengan sebuah obyek lain pada *class* lain yang berasosiasi dengan *class* tersebut. Untuk menyatakan multiplisitas anda dapat meletakkannya diatas garis asosiasi berdekatan dengan *class* yang sesuai. Ada banyak multiplisitas yang mungkin untuk dipakai. Tabel berikut menjabarkan multiplisitas yang dapat

digunakan.

Tabel 2.1 Nilai *Multiplisitas Potensial*

Indicator	Meaning
0..1	Zero or one
1	One only
0..*	Zero or more
1..*	One to more
N	Only n (where n > 1)
0..n	Zero to n (where n > 1)
1..n	One to n (where n > 1)

(Sumber : Modul Workshop UML Bab 2)

Sebuah *class* (*child class* atau *subclass*) dapat mewarisi atribut-atribut dan operasioperasi dari *class* lainnya (*parent class* atau *super class*) dimana *parent class* bersifat lebih umum daripada *child class*.

Generalisasi pada konsep *Object Oriented* digunakan untuk menjelaskan hubungan kesamaan diantara *class*. Dengan menggunakan *generalisasi* bisa dibangun struktur logis yang bisa menampilkan derajat kesamaan atau perbedaan diantara *class-class*. Manfaat lain dari struktur hirarkis juga memungkinkan untuk penambahan *subclass* (*child class*) baru tanpa harus merubah struktur yang sudah ada.

Inheritance adalah sebuah mekanisme pengimplementasian *generalisasi* dan spesialisasi. Aturan *inheritance* dapat secara umum bisa diklasifikasikan sebagai berikut :

1. *Subclass* selalu mewarisi semua sifat dari *superclass*-nya.
2. Definisi *subclass* selalu mencakup paling tidak satu detil yang tidak diturunkan dari *superclass*-nya.

Inheritance sangat dekat dengan asosiasinya dengan *generalisasi.*

Generalisasi menjelaskan hubungan logis antar elemen-elemen yang mempunyai karakteristik yang sama. Sedangkan *inheritance* menerangkan mekanisme agar bagi pakai (*sharing*) bisa terjadi.

Didalam UML, ada relasi dengan perlakukan khusus yang disebut dengan "bagian dari (*part of*)" yang menangani antar objek-objek dimana salah satunya adalah bagian dari yang lain. Dengan kata lain sebuah objek terdiri atas objek-objek yang lain. Hal seperti ini di dalam UML disebut *agregasi*. Sebuah *agregasi* adalah kasus khusus dari *asosiasi*. *Agregasi* disimbolkan dengan jajaran genjang yang diletakkan pada *class* yang mengandung objek. Kadangkala relasi OR harus digunakan pada *agregasi*. Untuk memodelkan hal tersebut, sebuah constraint – dengan kalimat atau dalam tanda kurung kurawal pada garis putus-putus harus dibuat untuk menghubungkan dua objek.

Komposit (*composite*) adalah sebuah tipe agregasi yang kuat dimana bagian dari obyek tergantung penuh/secara keseluruhan terhadap objeknya sehingga bila sebuah obyek komposit dibuang maka bagian yang tergantung pada komponen tersebut akan terbuang juga pada saat yang bersamaan. Notasi komposit sama seperti agregasi hanya saja jajaran genjangnya terisi (solid).

Interface adalah satu set *operation* yang memberikan spesifikasi beberapa aspek dari perilaku dan operasi disuatu *class* ke *class* yang lain. Contohnya, *keyboard* pada komputer sebenarnya merupakan *interface* yang bisa dipakai ulang karena tombol-tombol *keyboard* sebenarnya berasal dari mesin ketik, hanya saja mungkin ada beberapa operasionalisasi tombol-tombol yang berbeda yang sudah ditransfer ke sistem yang lain. Pemodelan *interface* sama dengan pemodelan *class*

hanya saja pada *interface* tidak mempunya atribut dan pada penamaannya perlu ditambahkan *stereotype* <<interface>> atau ditambahkan huruf "I" di depan nama interface. Relasi antara *class* dan *interface* disebut *Realization*. Realisasi dituliskan dengan garis putus-putus dengan segitiga yang mengarah ke *interface*.

2.7.2.2 Use Case Diagram

Use case adalah deskripsi fungsi dari sebuah sistem dari *perspektif* / sudut pandang para pengguna sistem. *Use case* mendefinisikan "apa" yang dilakukan oleh sistem dan elemen-elemennya, bukan "bagaimana" sistem dan elemen-elemennya saling berinteraksi. *Use case* bekerja dengan menggunakan *"scenario",* yaitu deskripsi urutan-urutan langkah yang menerangkan apa yang dilakukan penggunan terhadap sistem maupun sebaliknya. *Use case* diagram mengidentifikasikan fungsionalitas yang dipunyai oleh sistem (use-case), *user* yang berinteraksi dengan sistem *(actor)* dan *asosiasi* / keterhubungan antara user dengan fungsionalitas sistem.

Fungsi dari diagram *use case* adalah Menggambarkan Fungsionalitas yang diharapkan dari suatu sistem, dengan menekankan aspek aspek apa yang dilakukan sistem bukan bagaimana sistem melakukannya. Sebuah *use case* merepresentasikan sebuah interaksi antara *actor* dengan sistem.

Berbicara mengenai *use case* diagram tidak akan terlepas dengan hal yang disebut *stereotype. Stereotype* adalah sebuah model khusus yang terbatas untuk kondisi tertentu. Untuk menunjukkan *stereotype* digunakan simbol "<<" diawalnya dan ditutup dengan ">>" diakhirnya. Terdapat 2 *stereotype* paling sering digunakan dalam *use case* diagram yaitu <<*extend*>> dan <<*include*>>.

<<*extend*>> digunakan untuk menunjukkan bahwa satu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu dipenuhi. Sedangkan <<*include*>> digunakan untuk menggambarkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya.

2.7.2.3 Activity Diagram

Activity diagram digunakan untuk mendokumentasikan alur kerja pada sebuah sistem, yang dimulai dari pandangan business level hingga ke *operational level.* Pada dasarnya, *activity* diagram merupakan variasi dari statechart diagram. Activity diagram mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity* diagram bisa mendukung perilaku parallel sedangkan *flowchart* tidak bisa. Berikut adalah notasi *activity diagram* :

2.7.2.4 Sequence Diagram

Sequence diagram menurut Munawar (2005 : 187) adalah grafik dua dimensi dimana obyek ditunjukkan dalam dimensi horizontal, sedangkan *lifeline* ditunjukkan dalam dimensi vertikal.

2.7.2.5 Component Diagram

Komponen perangkat lunak adalah bagian fisik dari sebuah sistem yang menetap di komputer. komponen merupakan implementasi *software* dari sebuah *class*. Komponen bisa berupa tabel, file data, file exe, file DLL, dokumen dan lain-lain.

Component diagram mengandung komponen, *interface* dan *relationship*. Komponen diagram ini digunakan pada saat anda ingin memecah sistem menjadi

komponen-komponen dan ingin menampilkan hubungan-hubungan mereka dengan antarmuka atau pemecahan komponen menjadi struktur yang lebih rendah. Secara umum dapat kita katakan bahwa *component* diagram kita gunakan untuk menjelaskan kebergantungan antar beragam komponen-komponen *software* seperti misalnya kebergantungan antara file-file *executable* dengan file-file sumbernya *(source file*) dll. Berikut adalah notasi dari *component diagram* :

2.7.2.6 Deployment Diagram

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut. *Deployment* diagram dapat digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem. Berikut adalah notasi-notasi yang digunakan pada *deployment diagram* :

2.7.2.7 Collaboration Diagram

Collaboration diagram menurut Munawar (2005 : 101) adalah perluasan dari objek diagram. Objek diagram menunjukkan objek-objek yang hubungannya satu dengan yang lain. *Collaboration diagram* menunnjukkan *mesage-message* objek yang dikirim satu sama lain.