BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Terdapat dua kelompok pendekatan didalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen – komponen atau elemennya. Pendekatan sistem yang lebih mendekatkan pada prosedur mendefinisikan sistem sebagai berikut ini :

Menurut Jogiyanto (2005 : 1) : "Suatu sistem adalah suatu jaringan kerja dari prosedur – prosedur yang saling berhubungan, berkumpul bersama – sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu".

2.1.1. Karakteristik Sistem

Menurut Jogiyanto (2005 : 4) : Suatu sistem mempunyai karakteristik atau sifat – sifat yang tertentu, yaitu :

1. Komponen sistem

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

2. Batasan Sistem

   Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan.

3. Lingungan Luar Sistem

Adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan berupa energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkunngan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak akan mengganggu kelangsungan dari hidup sistem.

4. Penghubung Sistem

   Merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber – sumber daya mengalir dari subsistem ke subsistem lainnya.

5. Masukan Sistem

   Adalah energi yang dimasukan kedalam sistem masukan dapat berupa masukan perawatan dan masukan sinyal. Masukan perawatan adalah energi yang dimasukan supaya sistem tersebut dapat beroperasi. Sedangkan masukan yang diproses untuk mendapatkann 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 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, Sistem akutansi akan mengolah data – data teransaksi menjadi laporan – laporan lain yang dibutuhkan oleh manajemen.

8. Sasaran sistem

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

2.1.2. Klasifikasi Sistem

Menurut Jogiyanto (2005 : 6) : 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. Contohnya sistem teknologi, yaitu sistem yang berupa pemikiran – pemikiran hubungan antara manusia dengan tuhan. Dan sistem secara fisik merupakan sistem yang ada secara fisik contohnya sistem komputer, sistem akutansi, sistem produksi dan lain sebagainya.

2. Sistem alamiah dan sistem buatan manusia

   Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia misalnya sistem perputaran bumi, sedangkan Sistem buatan manusia adalah sistem yang dirancang oleh manusia, sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan man-machine system.contohnya sistem informasi.

3. Sistem tertentu dan sistem tak tertentu

Sistem terstruktur beroperasi dengan tingkah laku yang sudah dapat diprediksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Contohnya sistem komputer yang tingkah lakunya dapat dipastikan berdasarkan program – program yang dijalankan, sedangkan Sistem tak tertentu adalah sistem yang kondisi masa depannya tidak dapat diproduksi karena mengandung unsur probabilitas.

4. Sistem tertutup dan sistem terbuka

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Dan bekerja secara otomatis tanpa adanya campur tangan dari pihak luarnya, sedangkan Sistem terbuka adalah sistem yang menghubungkan dan terpengaruh dengan lingkungan luarnya.

2.2. **Konsep Dasar Informasi**

Informasi ibarat darah yang mengalir didalam tubuh suatu organisasi, sehingga informasi ini sangat penting didalam suatu organisasi. Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh, kerdil dan akhirnya berakhir.

Menurut Jogiyanto (2005 : 8) : “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya”.

Sumber dari informasi adalah Data adalah kenyataan yang menggambarkan suatu kejadian – kejadian dan kesatuan nyata,

2.2.1. Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui satu model untuk dihasilkan informasi.

Data yang diolah melalui suatu 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.

2.2.2. Kualitas Informasi

Menurut Jogiyanto (2005 : 10) Kualitas informaasi harus :

1. Akurat

   Akurat berarti harus bebas dari kesalahan dan tidak menyesatkan dan harus jelas mencerminkan maksudnya. Ketidak akuratan dapat terjadi karena sumber informasi (*data*) mengalami gangguann atau kesengajaan sehingga merusak, mengubah data – data asli tersebut.

2. Tepat pada waktunya

   Berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak mempunyai nilai lagi karena, informasi merupakan landasan pengambilan keputusan. Berarti informasi harus memiliki manfaat

bagi pemakainya. Relevansi informasi untuk tiap – tiap orang satu dengan yang lainnya berbeda.

2.2.3. Nilai Informasi

Nilai informasi ditentukan dari dua hal yaitu : manfaat dan biaya mendapatkannya, suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

2.3. Konsep Dasar Sistem Informasi

Menurut Jogiyanto (2005 : 11) : 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

Menurut Jogiyanto (2005 : 12) komponen sistem terdiri dari :

1. Blok Masukan

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

2. Blok Model

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

3. Blok Keluaran

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

4. Blok Teknologi

   Teknologi merupakan "kata lain" (*Tool box)* dalam sistem informasi, teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

5. Blok Basis Data

   Basis Data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan didalam basis data untuk keperluan penyedian informasi lebih lanjut.

6. Blok Kendali

   Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperature, air debu, kegagalan sistem itu sendiri.

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 Akademik

AKADEMIK adalah seluruh lembaga pendidikan formal baik pendidikan anak usia dini, pendidikan dasar, pendidikan menengah, pendidikan kejuruan maupun perguruan tinggi yang menyelenggarakan pendidikan vokasi dalam satu cabang atau sebagian cabang ilmu pengetahuan, teknologi, dan/atau seni tertentu.

2.4.2 Definisi Sistem Informasi Akademik

Sistem Informasi Akademik adalah sebuah sistem khusus untuk keperluan pengeloaan data-data Akademik dengan penerapan teknologi komputer baik ‘hardware’ maupun ‘software’. Yang dimaksud ‘hardware’ (perangkat keras) adalah peralatan-peralatan seperti computer (PC Computer), Printer, CD ROM, HardDisk, dan sebagainya. Sedang ‘Software’ (perangkat lunak) merupakan program computer yang memfungsikan ‘hardware’ tersebut yang dibuat khusus untuk keperluan pengelolaan data-data Akademik .‘Hardware’ komputer yang akan digunakan dapat dijumpai (dibeli) di pasaran, di tempat-tempat penjualan komputer. Sedang software, harus dibuat dengan teknik pemrograman tertentu.

2.5. Pengertian Basis Data (*Database*)

basis data (*database*) adalah kumpulan data yang saling berhubungan yang disimpan di tempat penyimpanan dan dapat dimanfaatkan kembali dengan cepat dan mudah.

Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur, dan batasan dari data atau informasi yang akan disimpan.

Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna. Penyusunan basis data meliputi proses memasukkan data kedalam media penyimpanan data, dan diatur dengan menggunakan perangkat basis data (*Database Management System*–DBMS). Manipulasi basis data meliputi pembuatan pernyataan (*query*) untuk mendapatkan informasi tertentu, melakukan pembaharuan atau penggantian (*update*) data, serta pembuatan report dari data. Tujuan utama DBMS adalah untuk menyediakan tinjauan abstrak dari data bagi user. Jadi sistem menyembunyikan informasi mengenai bagaimana data disimpan dan dirawat, tetapi data tetap dapat diambil dengan efisien. Pertimbangan efisiensi yang digunakan adalah bagaimana merancang struktur data yang kompleks, tetapi tetap dapat digunakan oleh pengguna yang masih awam, tanpa mengetahui kompleksitas struktur data.

Basis data menjadi penting karena munculnya beberapa masalah bila tidak menggunakan data yang terpusat, seperti adanya duplikasi data, hubungan antar data tidak jelas, organisasi data dan *update* data menjadi rumit.

Jadi tujuan dari pengaturan data dengan menggunakan basis data adalah :

1. Menyediakan penyimpanan data untuk dapat digunakan oleh organisasi/perusahaan saat sekarang dan masa yang akan datang.
2. Cara pemasukkan data sehingga memudahkan tugas operator dan menyangkut pula waktu yang diperlukan oleh pemakai untuk mendapatkan data serta hak-hak yang dimiliki terhadap data yang ditangani.

3. Pengendalian data untuk setiap siklus agar data selalu *up-to-date* dan dapat mencerminkan perubahan spesifik yang terjadi disetiap sistem.
4. Pengamanan data terhadap kemungkinan penambahan, modifikasi, pencurian dan gangguan-gangguan lain

2.6. Pengertian Jaringan Komputer

Menurut (Buddhi Irawan : 2005) adalah "himpunan interkoneksi antara dua komputer *autonomous* atau lebih yang terhubung dengan media transmisi kabel atau tanpa kabel (wireless)". Bila sebuah komputer dapat membuat komputer lainnya restart, shutdown, atau melakukan control lainnya, maka komputer-komputer tersebut bukan *autonomous* (tidak melakukan control terhadap computer lain dengan akses penuh).

Dua unit komputer dikatakan terkoneksi apabila keduanya bias saling bertukar data atau informasi, berbagai resource yang dimilki, seperti file, printer, media penyimpanan (hardisk, floppy disk, cd-rom, flash disk).

2.6.1. Jenis-Jenis Jaringan Komputer

Berdasarkan luasnya jangkauan, jaringan komputer terdiri dari:

1. Work Group

   Yaitu jaringan yang menghubungkan beberapa komputer dalam jumlah sedikit dalam sebuah ruangan.

2. Local Area Network (LAN)

   Yaitu jaringan komunikasi data yang luas jangkauannya meliputi suatu area lokal tertentu.

Keuntungan LAN :

a. Memungkinkan pemakaian sumber daya secara bersama-sama.
b. Meningkatkan produktifitas serta melindungi investasi yang ada.
c. Memungkinkan pengiriman data yang lebih banyak dan kompleks serta pertukaran informasi yang lebih baik.

Kerugian LAN :

a. Pembuatan instalasi jaringan tidak sederhana.
b. Perlunya software khusus yang dirncang untuk *multi user.*
c. Perlunya pengaturan dan keamanan data di dalam jaringan/*network.*
d. Virus dapat menyebar ke seluruh jaringan.

3. Metropolitan Area Network (MAN)

   Yaitu jaringan komunikasi data yang luas jangkauannya meliputi area dalam satu kota.

4. *Wide Area Network* (WAN)

   Yaitu jaringan komunikasi data yang luas jangkauannya meliputi antar kota atau antar negara.

2.6.2. Topologi Jaringan Komputer

Topologi jaringan adalah gambaran perencanaan hubungan antar komputer dalam Local Area Network yang umumnya menggunakan kabel (sebagai media transmisi), dengan konektor, ethetnet card, dan perangkat pendukung lainnya.

Ada beberapa jenis topologi yang terdapat pada hubungan komputer pada jaringan local area, seperti :

1. Topologi Bus

   Topologi bus merupakan jenis arsitektur yang paling sederhana. Dalam penerapannya, topologi ini sering digunakan untuk membangun jaringan yang hanya terdiri dari beberapa unit komputer, misalnya 2 sampai 4 unit komputer.

   Topologi ini disebut topologi bus karena jenis arsitekturnya menyerupai bus (kendaraan umum). Hal tersebut didasarkan pada setiap *node* (workstation) yang diibaratkan seperti kursi yang ada pada bus kota.

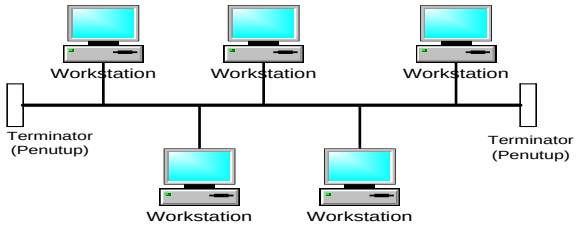
   Beberapa kelebihan dari topologi bus :

   a. Topologi bus merupakan arsitektur jaringan yang paling sederhana dibanding jenis arsitektur lainnya.
   b. Dikatakan sangat sederhana karena hanya memiliki anggota workstation yang sedikit.
   c. Biaya yang dikeluarkan sangat murah karena media transmisi yang digunakan adalah kabel coaxial.
   d. Karena menggunakan satu kabel yang menjadi pusat, pengiriman data pun lebih cepat.

   Sedangkan kekurangan pada topologi bus, yaitu :

   a. Karena menggunakan satu kabel yang dijadikan pusat, akibatnya sering terjadi tabrakan data.
   b. Apabila ada salah satu workstation *error*, maka akan mengakibatkan kerusakan pengiriman data dari komputer lain.

Untuk lebih jelasnya dapat dilihat pada gambar berikut:



Gambar 2.1 Topologi Bus

Sumber : Budhi Irawan (2005 : 26)

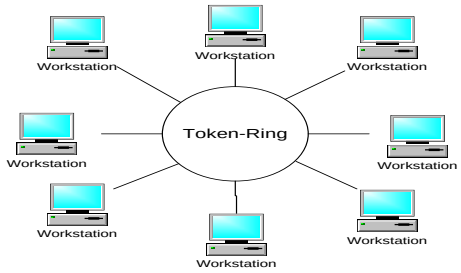
2. Topologi Cincin (Ring)

   Topologi cincin merupakan desain jaringan yang dapat dikatakan cukup sederhana dibandingkan dengan bintang. Dalam pemasangan jaringan, arsitektur ini akan menggunakan kabel yang dipasang melingkar dengan sistem tertutup. Kabel yang sering digunakan pada jenis topologi ini adalah kabel coaxial. Keuntungan yang didapat dari topologi cincin :

   a. Implementasinya sangat sederhana karena arsitektur ini merupakan bentuk pengembangan dari topologi bus.
   b. Peralatan yang digunakan sama seperti topologi bus, yaitu menggunakan media transmisi kabel coaxial.
   c. Pada jenis ini anda tidak lagi memerlukan terminator karena kedua ujung kabel akan disambungkan dengan ujung kabel yang lain.
   d. Transfer data dilakukan dalam satu arah, sehingga kemungkinan terjadi tabrakan sangat jarang.

Kerugian yang mungkin akan didapat adalah kegagalan pengiriman data karena topologi jenis ini sangat dipengaruhi oleh *node* (workstation) yang lain. Pengiriman data dilakukan dalam satu arah. Apabila ada salah satu workstation yang rusak, proses pengiriman data akan terputus atau gagal.

Untuk lebih jelasnya dapat dilihat pada gambar berikut:



Gambar 2.2 Topologi Ring

Sumber : Budhi Irawan (2005 : 28)

3. Topologi Star (Bintang)

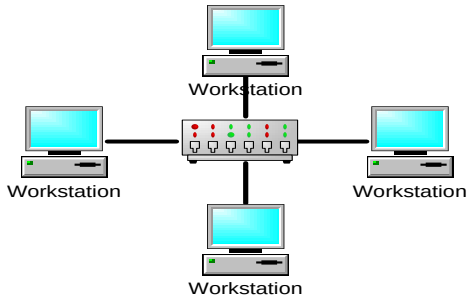
Topologi ini dinamakan topologi bintang karena bentuk arsitekturnya dapat dianalogikan seperti bentuk bintang. Tentu saja bentuk bintang yang dimaksud tidak seperti bentuk bintang yang anda ketahui selama ini.

Bentuknya seperti bintang karena pada perancangannya arsitekturnya memiliki *node* (titik) terpusat, yang kemudian dihubungkan dengan *node-node* atau workstation yang lain. Dalam implementasinya, pengembangan arsitektur jaringan ini akan selalu memerlukan sebuah peralatan yang dijadikan sebagai terminal yang menghubungkan antara komputer satu dengan

komputer yang lain. Terminal yang dipakai dapat berupa *hub* atau *switch.* Banyak keuntungan yang akan didapat dari topologi star, diantaranya :

a. Sanggup memuat banyak workstation dalam satu jaringan LAN.
b. Sangat jarang terjadi tabrakan data.
c. Karena jarang terjadi tabrakan data, transfer data akan lebih cepat.
d. Apabila salah satu workstation terputus atau rusak, workstation lain tidak akan mengalami gangguan.
e. Kerusakan kabel, misalnya putusnya transmisi kabel pada salah satu workstation, tidak akan mengakibatkan kerusakan jaringan secara menyeluruh.
f. Memiliki teknik kerja yang terpusat, maksudnya semua workstation yang melakukan pengiriman data akan dikirimkan melalui media transmisi menuju terminal. Selanjutnya data tersebut akan diforward oleh terminal ke alamat tujuan pengiriman.

   Untuk lebih jelasnya dapat dilihat pada gambar berikut:



Gambar 2.3 Topologi Star

Sumber : Budhi Irawan (2005 : 27)

2.6.3. Manfaat Jaringan Komputer

Manfaat yang didapat dari membangaun jaringan komputer menurut (Buddhi Irawan : 2005) adalah sebagai berikut :

1. Sharing Resources

   Sharing Resources bertujuan agar seluruh program, peralatan/ peripheral 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, seoranng 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 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 *hardisk* dan peralatan lainnya. Misalnya untuk memberikan perlindungan serangan virus maka pemakai cukup memusatkan perhatian pada hardisk yang ada di komputer pusat.

5. Keamanan Data

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

6. Sumber Daya Lebih Efisien dan Informasi Terkini

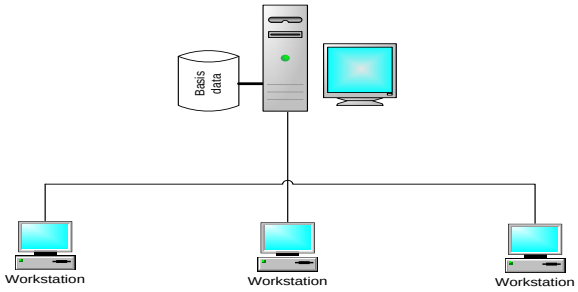
   Dengan adanya pemakaian sumber daya secara bersama-sama maka pemakai bisa mendapatkan hasil dengan maksimal dan kualitas yang tinggi. Selain itu data atau informasi yang di akses terlalu terbaru, karena setiap ada perubahan yang terjadi dapat secara langsung di ketahui oleh setiap pemakai.

2.6.4. Pengertian Client/Server

Model hubungan Client-Server memungkinkan jaringan untuk mensentralisasi fungsi dan aplikasi kepada satu atau dua *dedicated* file server. Sebuah file server menjadi jantung dari keseluruhan sistem, memungkinkan untuk mengakses sumber daya, dan menyediakan keamanan. Workstation yang berdiri sendiri dapat mengambil sumber daya yang ada pada file server. Model hubungan ini menyediakan mekanisme

untuk mengintegrasikan seluruh komponen yang ada di jaringan dan memungkinkan banyak pengguna secara bersama-sama memakai sumber daya pada file server.

Di dalam jenis ini, klien dan server akan sangat berhubungan erat. Apabila ada sebuah komputer yang selalu menyediakan sumber daya dan digunakan oleh komputer lain, komputer tadi disebut komputer server.



Gambar 2.4 Model Hubungan Client Server

Sumber : Budhi Irawan (2005 : 30)

2.7. Konsep Dasar Website

Adapun konsep dasar website dapat dilihat pada penjelasan di bawah ini.

2.7.1. Sekilas Mengenai Website

Situs Web (Website) adalah kesatuan dari dokumen-dokumen yang terhubung dengan atribut yang sama misalnya topik, fungsi maupun desain yang sama. Jadi sebuah situs web memiliki beberapa dokumen yang biasanya disebut dengan halaman web, serta komponen-komponennnya. Untuk membuat atau mendesain situs web, yang pertama harus dilakukan adalah membuat rumusan-rumusan yang diperlukan.

2.7.2. Tujuan Pembuatan Website

Prioritas utama membuat situs web adalah merumuskan suatu tujuan, untuk apa situs web tersebut dibuat. Sebagai contoh, instansi atau lembaga yang bergerak

dibidang pendidikan membuat suatu website dengan tujuan untuk memperkenalkan dan mempromosikan lembaga pendidikan tersebut.

Dari contoh di atas dapat diambil kesimpulan bahwa website yang akan dibuat adalah website yang dapat memberikan suatu informasi yang lengkap dan akurat tentang sistem pendidikan yang ditawarkan oleh instansi atau lembaga pendidikan tersebut. Maksudnya adalah agar masyarakat yang melihat website tersebut dapat mengenal dengan baik dan berminat untuk mendaftarkan diri ke lembaga pendidikan tersebut.

2.7.3. Menentukan Isi Website

Dengan target yang jelas kita dapat mempersiapkan isi yang nantinya akan dimasukkan atau didesain dalam halaman web, sehingga kita dapat memperkirakan isi dari website seperti profil dan sejarah singkat dari lembaga atau instansi, struktur orgsnisasi lembaga, visi dan misi lembaga, serta gambaran umum tentang sistem.

2.7.4. Menentukan Struktur Website

Struktur suatu situs web memegang peranan penting dalam kemudahan memanajemen situs yang akan dibangun. Struktur yang baik memudahkan web master untuk mengelola situs web tersebut dan tentunya struktur yang kurang baik akan menyusahkan web master untuk mengelolanya.

Struktur sebuah website juga harus disesuaikan dengan isi situs web tersebut. Situs web yang komplit harus memiliki manajemen yang baik karena struktur yang tidak baik akan membingungkan pengunjung.

2.8 Pemodelan

2.8.1 Pengertian UML

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML tidak berdasarkan pada bahasa pemrograman tertentu. Standar spesifikasi UML dijadikan standar defacto oleh OMG (*Object Management Group*) pada tahun 1997. UML yang berorientasikan objek mempunyai beberapa notasi standar.

Spesifikasi ini menjadi populer dan standar karena sebelum adanya UML, telah ada berbagai macam spesifikasi yang berbeda. Hal ini menyulitkan komunikasi antar pengembang perangkat lunak. Untuk itu beberapa pengembang spesifikasi yang sangat berpengaruh berkumpul untuk membuat standar baru. UML dirintis oleh Grady Booch, James Rumbaugh pada tahun 1994 dan kemudian Ivar Jacobson.

2.8.2 Konsep Dasar UML

UML mendeskripsikan OOP (*Object Oriented Programming*) dengan beberapa diagram, diantaranya:

a. 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.

b. 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

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* .

c. Hubungan Antar Class

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 m* enunjukkan 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.

d. Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*). Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung

siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

e. 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, di mana 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 vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

f. 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.

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity* .

g. 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.

h. 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.

i. 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.

2.9. 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.

2.9.1. PHP

PHP yang merupakan singkatan dari PHP : *Hypertext Preprocessor*, adalah bahasa pemrograman yang mana file dan seluruh prosesnya dikerjakan di server, kemudian hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan *browser* (lebih dikenal dengan istilah *server-side scripting*).

PHP bekerja di dalam sebuah dokumen HTML (*Hypertext Markup Language*) untuk dapat menghasilkan isi dari sebuah halaman web sesuai permintaan.

Dengan PHP kita dapat merubah situs kita menjadi sebuah aplikasi berbasis web, tidak lagi hanya sekedar sekumpulan halaman statis yang jarang diperbaharui.

Mengapa PHP, Karena PHP bersifat tidak memiliki ketergantungan terhadap berbagai platform, jadi PHP dapat dijalankan dalam platform apapun, baik itu *Unix*, Windows ataupun *Macintosh*.

Kelebihan lain dari PHP adalah kemudahan melakukan pengkodean, karena perintah-perintah PHP mirip dengan perintah-perintah C. Selain itu kemudahan dari PHP adalah dapat dengan mudah dihubungkan dengan aplikasi database (melakukan *query*), seperti MySQL dan PostgreSQL.

PHP bersifat *free* (bebas dipakai). Kita tidak perlu membayar apapun untuk menggunakan perangkat lunak ini. Untuk versi Windows, kita dapat memperoleh

kode binernya, dan untuk versi Linux, kita mendapatkan kode sumbernya secara lengkap.

Kelahiran PHP bemula saat Rasmus Lerdorf pada tahun 1994 membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya, Skrip-skrip ini selanjutnya dikemas menjadi *tool* yang disebut "*Personal Home Page*". Kumpulan tool inilah yang nantinya menjadi cikal-bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI versi 2. Pada versi inilah pemrogram dapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan database dan melakukan perhitungan-perhitungan yang kompleks sambil jalan.

Pada awalnya, PHP dirancang untuk diintegrasikan dengan webserver Apache. Namun belakangan ini, PHP juga dapat bekerja dengan webserver seperti PWS (*Personal Web Server*), IIS (*Internet Information Server*) dan *Xitami*.

Yang membedakan PHP dengan bahasa pemrograman lain adalah adanya tag penentu, yaitu diawali dengan "<?" atau "<?php" dan diakhiri dengan "?>". Jadi kita bebas menempatkan skrip PHP dimanapun dalam dokumen HTML yang telah kita buat.

Kelebihan lain dari PHP adalah mampu berkomunikasi dengan berbagai database yang terkenal. Dengan demikian, menampilkan data yang bersifat dinamis, yang diambil dari database, merupakan hal yang mudah untuk diimplementasikan. Itulah sebabnya sering dikatakan bahwa PHP sangat cocok untuk membangun halaman-halaman web dinamis.

Pada saat ini PHP sudah dapat berkomunikasi dengan berbagai database, meskipun dengan kelengkapan yang berbeda-beda, beberapa diantaranya adalah :

1. DBASE
2. DBM
3. FilePro (Personix, Inc)
4. Informix
5. Interbase
6. Microsoft Access
7. MSQL
8. MySQL
9. Oracle
10. PostgreSQL
11. Sybase

2.9.2. Database MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal. MySQL menjadi sangat populer karena MySQL bersifat *free* (tidak perlu membayar dalam menggunakannya) pada berbagai jenis platform (*unix*/*windows*).

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Pada MySQL, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

Pada saat instalasi, secara *default* MySQL akan membuat sebuah database bernama mysql. Salah satu isi dari database ini adalah tabel user, tabel ini berisi nama

dan password user yang dapat mengakses data pada database yang dibuat di mysql. Database ini juga berisi hak-hak yang diberikan pada setiap user.

MySQL menerima berbagai macam tipe data, tipe-tipe data ini dibagi menjadi 3, yaitu tipe data untuk bilangan, tipe data untuk tanggal dan jam, dan tipe data untuk karakter

2.9.3 Web Server Apache

Apache merupakan turunan dari web server yang dikeluarkan oleh NSCA yaitu NSCA HTTPD sekitar tahun 1995-an. Pada dasarnya apache adalah "*Apatchy* (*patch*)" dan pengganti dari NSCA HTTPD. Apache web server merupakan tulang punggung dari *Word Wide Web* (www). Web server menunggu permintaan dari klien yang menggunakan browser seperti *Netscape Navigato*, *Modzila*, *Lynk* dan lain-lain. Web server dalam berkomunikasi dengan kliennya, menggunakan protokol HTTP (*Hyper Text Transfer Protocol*).

Apache berada di bawah GNU general public license yang bersifat gratis Saat ini apache banyak digunakan sebagai web server untuk portal-portal besar.

Ada beberapa ciri khas apache, yaitu:

1. Apache sangat cepat dalam merespon klien melebihi server NSCA.
2. Apache mempunyai komponen dasar yang banyak dibandingkan dengan web server lain.
3. Performansi dan konsumsi sumber daya dari web server apache tidak terlalu banyak.

4. Mendukung transaksi yang aman (*secure tranmision*) menggunakan SSL (*Secure Socket layer*)
5. Kompabilitas yang tinggi.

Apache merupakan produk yang istimewa karena berisi hal-hal yang diperlukan oleh sebuah web, dapat diakses, dapat diandalkan dan gratis. Apache merupakan web server yang *open source*.

Ini berarti setiap yang menggunakan produk ini mempunyai akses ke sumber kode (*source code*). Jika ingin menambahkan sesuatu tinggal dituliskan ke dalam sumber kode dan menjalankanya. Apabila ditemukan kesalahan (*bug*) dapat segera diperbaiki atau segera dilaporkan.

Menurut *Netcraft* apache web server telah banyak digunakan melebihi web server lainnya. Dari sekitar tujuh juta website dalam dunia web sekitar empat juta (55%) diantaranya menggunakan apache web server.

2.9.4 Adobe Dreamweaver CS4

Desain dengan cepat dan mudah, mengembangkan, dan memelihara *web*site dan aplikasi *web* dari awal hingga selesai - dengan Adobe Dreamweaver CS4 *software*. Dibangun untuk kedua desainer dan pengembang, Dreamweaver CS4 menawarkan pilihan bekerja di sebuah intuitif visual layout interface atau lingkungan coding efisien.

Intelligent integrasi dengan Adobe Photoshop CS4, Adobe Illustrator CS4, Adobe Fireworks CS4, Adobe Flash CS4 Professional, dan perangkat lunak Adobe CS4 Kontribusi menjamin efisien alur kerja di alat favorit Anda.

Web desainer dapat menemukan sebuah alur kerja luar biasa *web* design dengan Adobe Dreamweaver CS4 *software*. Desain dan memelihara *web*site dari awal hingga selesai dengan pilihan anda, baik yang kuat visual desain antarmuka atau yang efisien coding lingkungan. Merampingkan desain dengan fitur bermanfaat CSS; pratinjau halaman untuk PC, perangkat mobile, dan cetak, dan mudah untuk halaman browser compatibility. Nikmati ketat dengan Adobe Photoshop CS4, Flash CS4 Professional, Fireworks CS4, dan Adobe peralatan lainnya.

Mengembangkan lebih handal, konsisten aplikasi berbasis *web* pada umumnya diterima pengembangan standar dengan Adobe Dreamweaver CS4. Mengambil keuntungan dari dukungan untuk terkemuka perkembangan teknologi

web, termasuk PHP, Adobe ColdFusion, dan Ajax, baik untuk internal dan eksternal proyek. Memastikan konsistensi yang bermanfaat CSS fitur otomatis dan lintas-browser kompatibilitas pengujian. Temukan Dreamweaver CS4 dan lengkap dari fitur-fitur baru, termasuk CSS alat-alat, Ajax komponen untuk membangun user interface dinamis, dan cerdas integrasi dengan perangkat lunak Adobe lainnya.

2.9.5 Adobe Photoshop

Adobe Photoshop telah berada garis terdepan dalam inovasi pengolah gambar (digital imaging), dan mereka telah memberikan banyak kontribusi bagi perkembangan dunia digital. Di tahun 2009, Adobe Corporation meluncurkan kembali program Adobe Photoshop teranyar dengan label Adobe Photoshop CS4. Dalam bundel Adobe Photoshop Creative Suite 3 tersebut, mereka tidak hanya satu program Adobe Photoshop saja melainkan dua pilihan yaitu Adobe Photoshop CS4 dan Adobe Photoshop CS4 Extended.

Adobe Photoshop CS4 memuat semua fitur yang Anda sukai dalam Photoshop CS2, jadi semua fitur yang ada pada versi sebelumnya tidak akan hilang. Photoshop CS4 mempercepat kurva (path) dari imajinasi untuk image. Photoshop CS4 ini sangat ideal untuk fotografer, desainer, dan standar profesional seperti perataan layer secara otomatis dan teknik pencampuran komposisi secara “advance”.

Ideal untuk:

a. Fotografer (*fotographer*)
b. Perancang Grafik (*graphic designer*)
c. Perancang *Web* (*web designer*)
d. Penyedia layanan cetak (*print service provider*)

2.9.6 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.