BAB II

LANDASAN TEORI

Pada Bab ini akan diuraikan teori yang menunjang dalam pemecahan masalah yang berkaitan dengan permasalahan yang tengah dihadapi oleh pihak perusahaan.

2.1 Konsep Dasar Sistem

Perancangan suatu program aplikasi terdiri dari satu kesatuan sistem. Terdapat dua kelompok pendekatan didalam mendefinisikan sistem, yaitu yang menekankan pada prosedur dan yang menekankan pada komponen

Suatu sistem pada dasarnya merupakan suatu susunan yang teratur dari kegiatan yang berhubungan satu sama lain dan prosedur-prosedur yang berkaitan yang melaksanakan dan mempermudah pelaksanaan kegiatan utama dari suatu organisasi. "*Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran terntentu*."(Jogiyanto H.M,2001 :1)

Komponen-komponen atau subsistem dalam suatu sistem tidak dapat berdiri lepas sendiri-sendiri. Komponen-komponen atau subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai.

2.1.1 Karakteristik Sistem

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

1. Komponen Sistem (*components*)

   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 suatu subsistem, atau bagian-bagian dari sistem.

2. Batas Sistem (*Boundary*)

   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 satu kesatuan. Batas suatu sistem menujukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environtment*)

   Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan menggangu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*System Interface*)

   Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Dengan

penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem (*System Input*)

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

6. Keluaran Sistem (*System Output*)

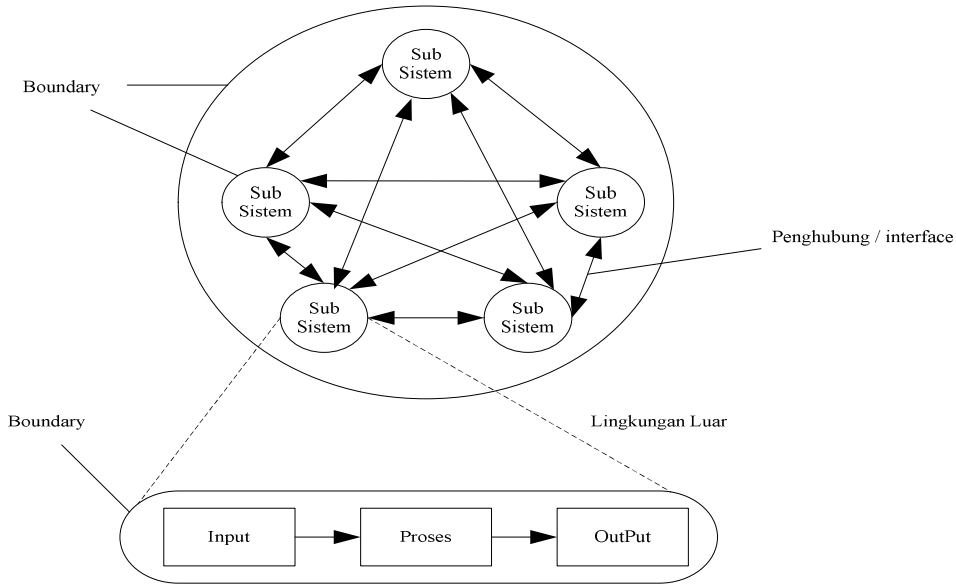
   Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

7. Pengolah Sistem (*System Processing*)

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

8. Sasaran Sistem (*Objectives System*)

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



Gambar 2.1 : Karakteristik Sistem

[Sumber : Jogiyanto HM, 2000, *Analisis dan Desain Sistem Informasi*, Andi Offset, Yogyakarta]

2.2 Pengertian Informasi

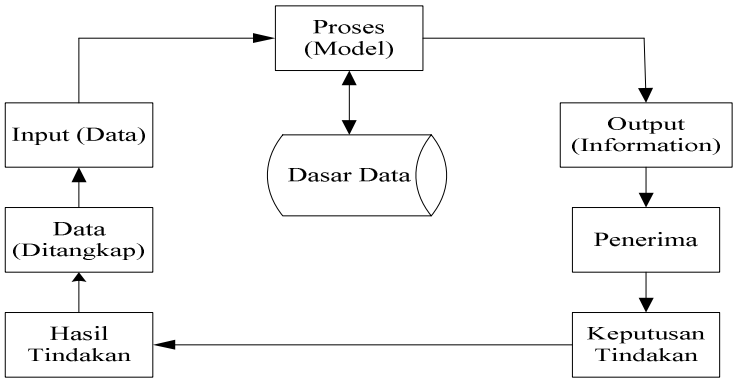
Informasi ibarat darah yang mengalir dalam tubuh suatu organisasi sehingga informasi menjadi sangat penting untuk sebuah organisasi, Menurut Jogianto, HM (2005), pengertian dari informasi adalah: *"Data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima dan membutuhkannya"*.

2.2.1 Siklus Informasi

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

membangun sebuah media penyimpanan sebagai tempat penyimpanan data-data akademik dan dapat diakses dengan cepat apabila dibutuhkan



Gambar 2.2 Siklus Informasi

(Sumber **: Al-Bahra. 2005.** ***Analisis dan Desain Sistem Informasi.*****Graha Ilmu.Yogyakarta)**

2.2.2 Kualitas Informasi

Kualitas dari suatu informasi tergantung pada tiga hal, yaitu :

1. Akurat (*Accurate*)

   Akurat artinya informasi harus mencerminkan keadaan yang sebenarnya dan dapat diandalkan.

2. Tepat Waktu (*Timelines*)

   Tepat waktu artinya informasi itu harus tersedia atau ada pada saat informasi tersebut diperlukan.

3. Relevan (*Relevance*)

Relevan artinya informasi diberikan harus mempunyai manfaat sebagai dasar pengambilan keputusan sesuai dengan yang dibutuhkan. Kalau kebutuhan informasi ini untuk suatu organisasi maka informasi tersebut harus sesuai dengan kebutuhan inforimasi diberbagai tingkatan dan bagian yang ada dalam organisasin tersebut

2.2.3 Konsep Dasar Sistem Informasi

Pengertian sistem informasi menurut Robert A. Leitch dan K. Roscoe Davis yang dikutip dari buku karangan Jogiyanto HM berjudul Analisis dan desain sebagai berikut "Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertamukan kebutuhan pengolahan transaksi harian, mendukung operasi bersifat manajerial dan kegiatan strategi-strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan".

2.2.4 Komponen Sistem Informasi

Sistem informasi dapat terdiri dari komponen-komponen, yaitu :

1. Blok Masukan

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

2. Blok Model

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

3. Blok Keluaran

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

4. Blok Teknologi

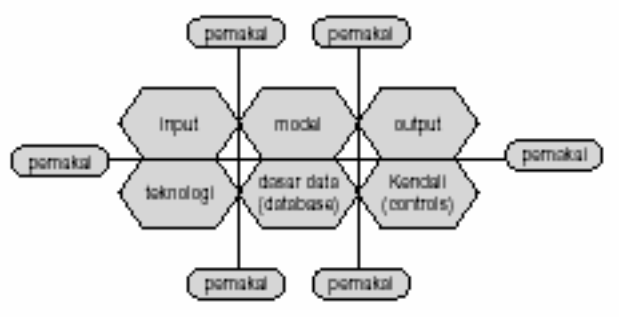
   Digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, mengasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan

5. Blok Basis Data

   Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam dasar data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

6. Blok Kendali

   Untuk supaya sistem informasi dapat berjalan sesuai dengan yang diingikan, maka perlu diterapkan pengendalian-pengendalian. Beberapa pengendali perlu dirancang dan diterapkan untuk menyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atapun bila terlanjur kesalahan-kesalahan dapat langsung cepat diatasi.



Gambar 2.3 Blok sistem informasi yang berinteraksi

[Sumber : http://www.dwiantoro.com/documents/Modul_2_PTI.pdf]

2.3 Konsep – Konsep Didalam Basis Data (*Database*)

Basis data (*Database*) adalah sekumpulan data atau informasi yang saling terhubung atau terkait dan disimpan secara bersamaan dalam media penyimpanan. Sekumpulan data ini harus mengandung informasi untuk mendukung suatu sistem. Penyimpanan data pada *database* dikelola oleh satu atu banyak file. *Database* merupakan bagian penting dari sistem informasi. Penyimpanan – penyimpanan data secara konvensional akan memakan kerugian maka disusunlah *database.*

Kriteria – kriteria dalam *database* adalah :

1. Database harus bersifat *object oriented* bukan *program oriented* maksudnya adalah dalam aplikasi data apabila diakses secara bersamaan, tidak terjadi perubahan dalam struktur database.
2. Dapat dikembangkan dalam hal volume dan struktur
3. Kerangkapan data (*data redundancy*) harus seminimal mungkin

Bahasa yang digunakan dalam sistem *database* adalah SQL *(Structure Query Language*). Elemen – elemen yang terkait dalam sistem database adalah :

1. *Database*
2. Perangkat keras (*hardware*)
3. Perangkat lunak *(software*)
4. Personil pengguna basis data (*brainware*), yaitu user dan sistem analisis / administrator data.

2.3.1 *Database Management Sistem* (DBMS)

Manajemen Sistem Basis Data (*Database Managgement Sistem* / DBMS) adalah perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar. DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, semisal penyimpanan data dalam file dan menulis kode aplikasi yang spesifik untuk pengaturannya.

Penggunaan DBMS untuk suatu aplikasi tergantung pada kemampuan dan dukungan DBMS yang beroperasi secara efisien. Sehingga agar bisa menggunakan DBMS dengan baik, perlu diketahui cara kerja dari DBMS tersebut. Pendekatan yang dilakukan untuk menggunakan DBMS secara baik meliputi : implementasi DBMS dan arsitektur secara mendetail untuk dapat memahami desain dari suatu basis data

2.3.2 Structured Query Language

Inti dari terjadinya interaksi pengelolaan data antara produk yang berbeda sebenarnya dipelopori oleh berkembangnya suatu bahasa pengolahan data yang dinamakan SQL (structured Query Language) dimana bahasa tersebut sekarang telah menjadi bahasa standar pengolahan data untuk setiap produk database.

Namun demikian, dala hal – hal tertentu yang sifatnya spesifik terhadap beberapa perbedaan, akan tetapi secara umum gramatikal dari bahasa SQL ini sudah disepakati secara bersama – sama oleh seluruh vendor *database*. Sebagai konsekuensinya *vendor* yang tidak menyertakan bahasa SQL dalam produk *database*-nya akan menanggung resiko tinggi karena tidak akan mampu berinteraksi dengan produk dari *vendor* yang berbeda.

Kriteria pembagian perintah umum dari SQL adalah :

1. *Data Definition Language* (DDL)
2. *Data Manipulation Language* (DML)
3. *Transaction Processing Language* (TPL)
4. *Data Control Language* (DCL)

2.4 Pengertian Sistem Informasi Akademik

Menurut kamus bahasa indonesia Akademik berasal dari kata akademis yang berarti lembaga pendidikan tinggi yang bersifat ilmiah, ilmu pengetahuan dan teori [Har00, kamus B.Indonesia lengkap, jilid 1,hal 22]. Dari pengertian akademik, pengertian sistem, dan pengertian informasi diatas maka penulis mengambil kesimpulan mengenai pengertian sistem informasi akademik, sebagai berikut :

Sistem informasi akademik adalah sekumpulan elemen-elemen baik fisik maupun non fisik dan prosedur yang saling berhubungan satu sama lain menjadi satu kesatuan dan bekerja sama untuk mengolah data akademik menjadi informasi yang berguna bagi pemakainya.

Secara umum data-data yang diolah oleh sistem informasi akademik pada suatu sekolah meliputi data siswa, data guru, data jadwal mata pelajaran, data nilai, data registrasi siswa, dan data-data lain yang berhubungan dengan proses keakademikan dari mulai proses pendaftaran sampai kelulusan.

Dalam mengembangkan sistem akademiknya setiap lembaga pendidikan mempunyai kebijakan tersendiri dan dapat mendukung proses akademik di lembaga pendidikan tersebut. Sehingga proses pengolahan data akademik lembaga pendidikan yang satu dengan lembaga pendidikan yang lain akan berbeda.

2.5 Jaringan Komputer

Jaringan komputer adalah sebuah system yang terdiri dari atas komputer dan perangkat lainnya yang bekerja bersama-sama untuk mencapai suatu tujuan. Tujuan dari jaringan komputer adalah :

a. Membagi sumber daya, Contohnya berbagi pemakaian CPU, Harddisk, Memory
b. Komunikasi, Contohnya Surat elektronik, instant messaging, chatting
c. Akses Informasi, Contohnya Browsing

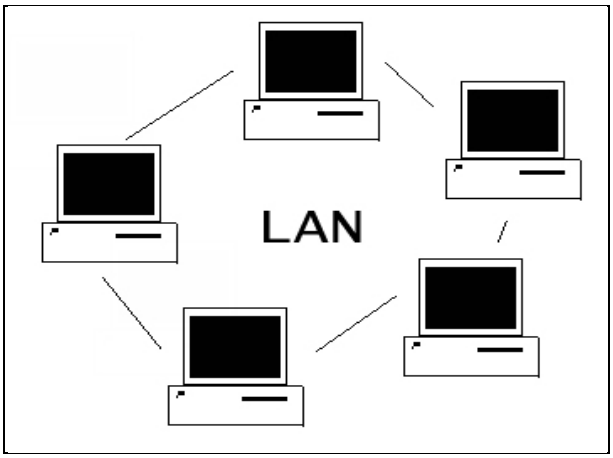
Agar dapat mencapai tujuan yang sama, setiap bagian dari jaringan komputer meminta dan memberikan layanan (service). Pihak yang meminta layanan disebut klien (client) dan yang memberikan layanan disebut pelayan (server). Arsitektur ini disebut dengan sistem client-server, dan digunakan pada hampir seluruh aplikasi jaringan komputer. [http://laksamana-embun.blogspot.com/jaringan.html]

2.5.1 Jenis Jaringan Komputer

Ada empat kategori utama jaringan komputer yang dibedakan berdasarkan cakupan geografisnya sebagai berikut :

1. LAN (*Local Area Network)*

   LAN adalah suatu jaringan yang digunakan untuk menghubungkan komputer yang berada dalam satu area yang kecil seperti dalam satu ruang, satu gedung, atau beberapa gedung yang berdekatan. LAN umumnya menggunakan media transmisi berupa kabel tetapi ada juga yang tidak menggunakan kabel dan disebut dengan wireless LAN atau LAN tanpa kabel. Jarak antara komputer yang dihubungkannya bisa mencapai 5 sampai 10 kilometer. Kecepatan LAN berkisar dari 10 sampai 100 Mbps.



Gambar 2.4 Arsitektur LAN

[sumber : http://www.galihakmal.com/index.php/basic/3-jenis-jenis-jaringan-komputer]

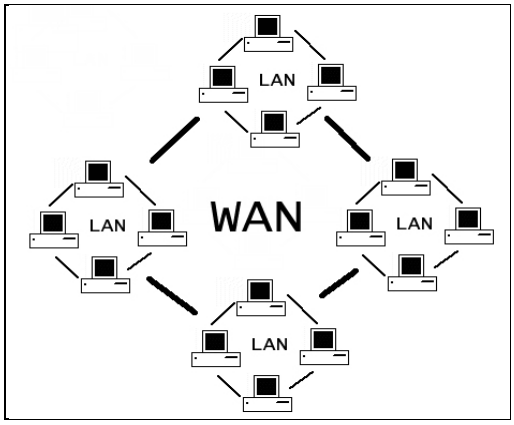
2. MAN *(Metropolitan Area Network)*

   Metropolitan Area Network (MAN), adalah suatu jaringan yang cakupannya meliputi suatu kota atau dengan rentang sekitar 10 sampai 45

kilometer. MAN menghubungkan LAN-LAN yang lokasinya berjauhan. MAN umumnya menggunakan media transmisi dengan mikro gelombang atau gelombang radio, tetapi ada juga yang menggunakan jalur sewa *(leased line)*. Kecepatan MAN berkisar dari 1,5 sampai 150 Mbps.

3. WAN *(Wide Area Network)*

   WAN dirancang untuk menghubungkan komputer-komputer yang terletak pada cakupan geografis yang luas, seperti hubungan dari kota ke kota lain dalam satu Negara. Cakupan WAN meliputi 100 sampai 1000 kilometer dan kecepatan antar kota bisa bervariasi antara 1,5 Mbps sampai 2,5 Gbps.

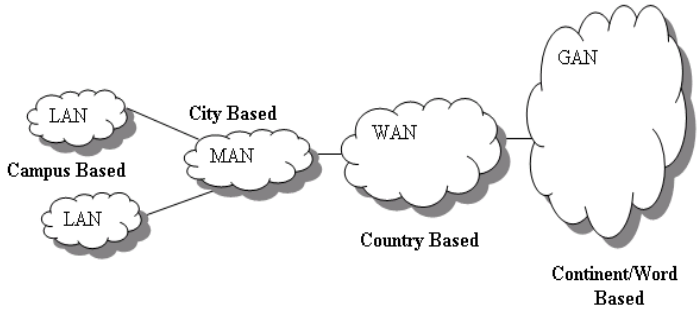


Gambar 2.5 Arsitektur WAN

[sumber : http://www.galihakmal.com/index.php/basic/3-jenis-jenis-jaringan-komputer]

4. GAN (Global Area Network)

   GAN adalah jaringan yang menghubungkan Negara-negara diseluruh dunia. Kecepatan GAN bervariasi mulai dari 1,5 Mbps sampai 100 Gbps, cakupan WAN mencapai ribuan kilometer.



Gambar 2.6 Interkasi antara LAN, MAN, WAN dan GAN.

[sumber : http://www.galihakmal.com/index.php/basic/3-jenis-jenis-jaringan-komputer]

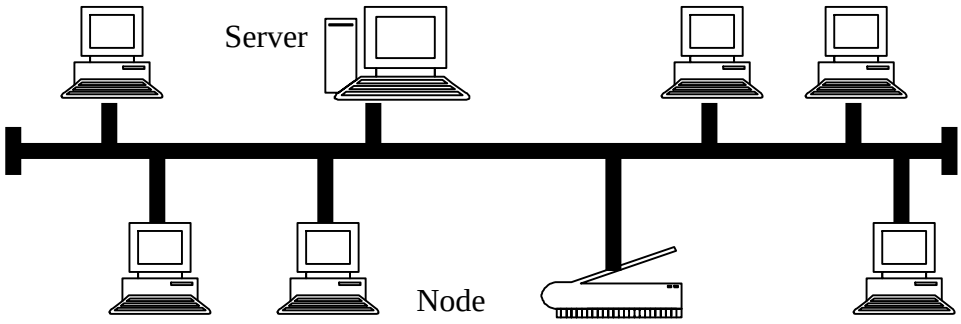
2.5.2 Topologi Jaringan Komputer

Topologi jaringan dalam buku Instalasi dan Konfigurasi Jaringan Komputer karangan Dede Sopandi adalah susunan fisik bagaimana node-node saling dihubungkan. Ada tiga topologi yang digunakan, yaitu :

1. **Topologi *Linear Bus***

   Topologi *Linear Bus* terdiri dari satu jalur kabel utama dimana pada setiap ujungnya diberikan sebuah *terminator*. Sebuah *nodes* pada jaringan (*file server, workstation*, dan perangkat lainnya) terkoneksi sebuah kabel utama (*backbone)*. Jaringan – jaringan *Ethernet* dan *Local Talk* menggunakan topologi *linear* ini.

   Topologi ini mudah dikembangkan, baik untuk penambahan maupun pengurangan terminal. Banyak terminal yang dapat terhubung tergantung pada jumlah port yang tersedia pada hub yang digunakan.

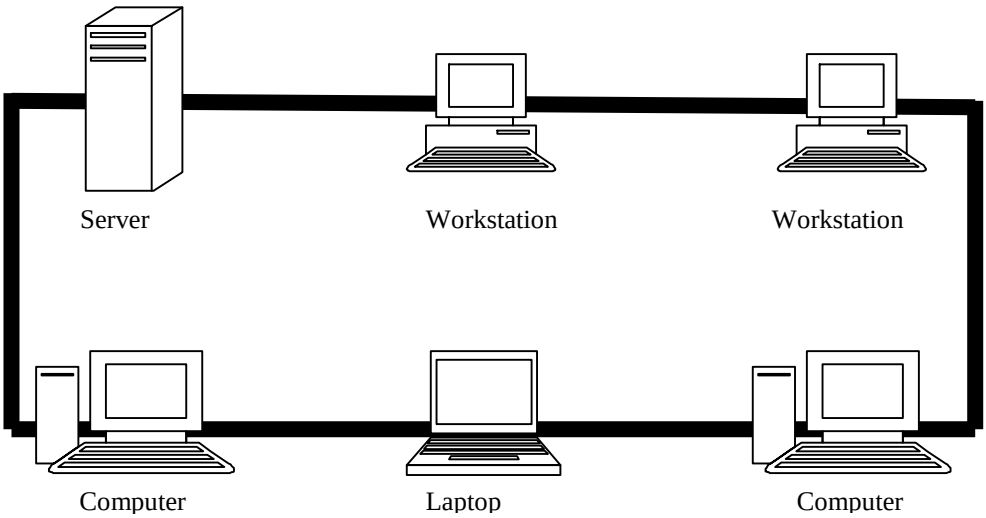


Gambar 2.7 Topologi Bus

[Sumber : Sopandi,Dede. 2005.Jaringan Komputer]

2. Topologi ring

Topologi ini memanfaatkan kurva tertutup, artinya informasi dan data serta traffic disalurkan sedemikian rupa sehingga masing-masing node. umumnya fasilitas ini memanfaatkan fiber optic sebagai sarananya.

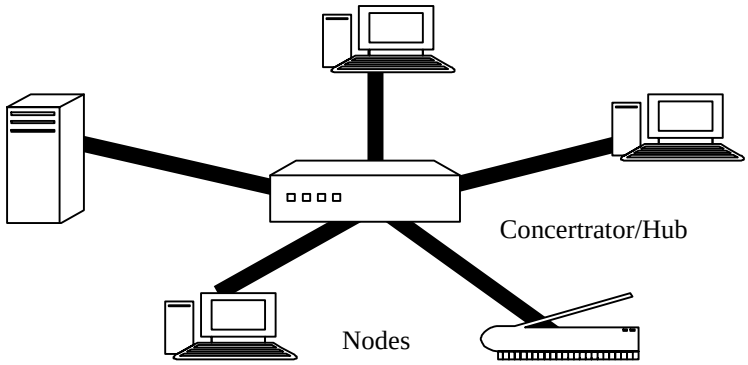


Gambar 2.8 Topologi Ring

(Sumber : Sopandi,Dede. 2005.Jaringan Komputer)

3. Topologi Star atau Hub

Topologi ini banyak digunakan di berbagai tempat, karena kemudahan untuk menambah atau mengurangi serta mudah untuk mendeteksi kerusakan pada sistem jaringan yang ada.

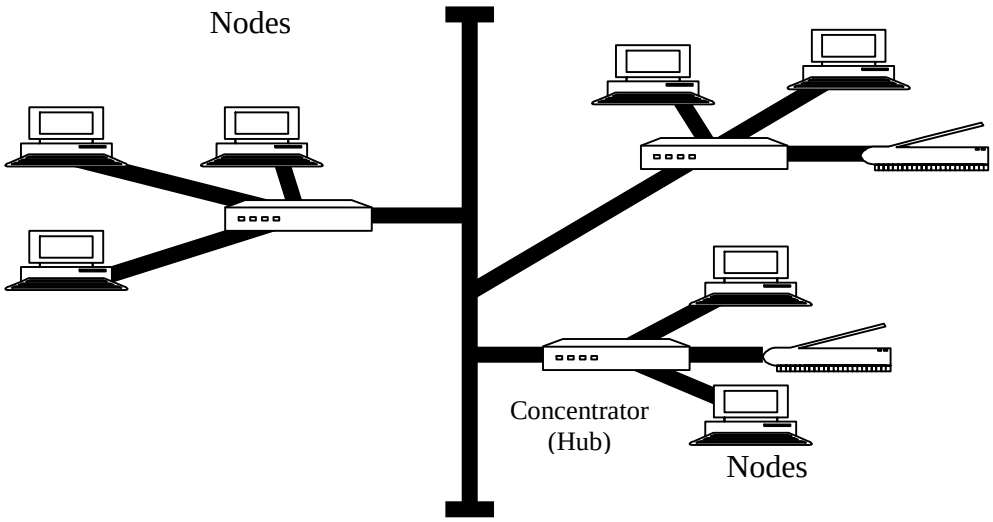


Gambar 2.9 Topologi Star

(Sumber : Sopandi,Dede. 2005.Jaringan Komputer)

4. Topologi Hybrid

Topologi hybrid adalah pada intinya bahwa sebuah jaringan bisa jadi merupakan kombinasi dari dua atau tiga topologi di atas. Topologi ini disebut juga tree topology.



Gambar 2.10 Topologi Hybrid

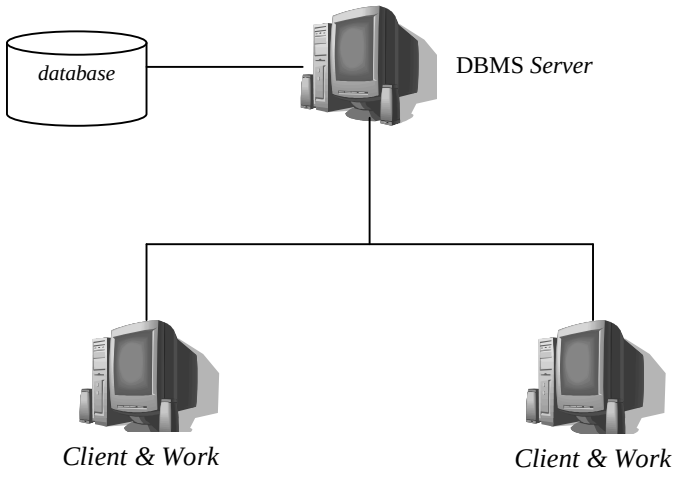
(Sumber :Sopandi,dede.2005. Jaringan Komputer)

2.6 Sistem *Client Server*

Sistem *client-server* atau disebut dengan sistem sentralisasi yang diterapkan dalam sebuah jaringan. *Client-server* terdiri dari dua kompunen, yaitu client dan server. Adapun pengertian client server adalah " *model konektivitas LAN (Local Area Network) yang mengenal adanya client dan server dimana masing-masing memiliki fungsi yang berbeda satu sama lain, dan dalam pelaksanaan tugasnya server sebagai pembagi data pada client* " dikutip dari http://id.wikipedia.org/wiki/apotek/02 april 2010

Client berisi aplikasi basis data dan *server* berisi DBMS dan basis data. Pada sistem *Client Server* untuk memenuhi kebutuhan, *client* akan mengirimkan *message* (perintah) *query* pengambilan data. Selanjutnya, *server* yang menerima *message* tersebut akan menjalankan *query* tersebut dan hasilnya akan dikirimkan

kembali ke *client.* Dengan begitu transfer datanya jauh lebih efisien. Adapun bentuk dari Sistem *Client Server* yang sederhan adalah sebagai berikut :



Gambar 2.11 Sistem *Client Server* Sederhana

(sumber : http://images.google.co.id/imglanding?q=clientserver&imgurl2010)

2.7 Perangkat Lunak Pendukung

Perangkat lunak yang digunakan dalam pembuatan sistem informasi ini adalah sebagai berikut :

2.7.1 Microsoft Visual Basic 6.0

Menurut Suryana, Taryana (2009), pengertian dari visual basic adalah: sebuah aplikasi yang digunakan untuk pengembangan dengan memanfaatkan keistimewaan konsep-konsep antar muka grafis dalam Microsoft Windows. Aplikasi yang dihasilkan visual basic berkaitan erat dengan Windows itu sendiri sehingga dibutuhkan pengetahuan bagaimana cara kerja windows.

Visual Basic merupakan bahasa pemrograman berbasis windows yang mempunyai tampilan antarmuka program dengan pengguna yang sudah bersifat

GUI (*Graphical User Interface*) sehingga semakin memudahkan pengguna dalam pembuatan program, menyajikan banyak kemudahan bagi para programmer untuk membuat aplikasi. Microsoft Visual Basic adalah bahasa pemrograman tingkat tinggi (*high level language*) untuk menyusun program aplikasi yang berdasarkan pada bahasa BASIC (*Beginners All-Purpose Symbolic Instruction Code*).

Bahasa BASIC merupakan bahasa pemrograman yang dijalankan dari sistem operasi DOS. Karena itu Microsoft mengembangkan suatu compiler bahasa BASIC untuk pemrograman Windows yang dinamakan Visual Basic. Bahasa Basic mempunyai sifat *event driven* yaitu program yang menunggu sampai adanya respon dari pemakai berupa kejadian tertentu, misalnya tombol diklik. Ketika event terdeteksi, event yang berhubungan akan melakukan aksi sesuai dengan kode yang diberikan.

Microsoft Visual Basic 6.0 mempunyai fleksibilitas dan berintegrasi dengan aplikasi lain yang menggunakan objek OLE *(Object Linking and Embedding).*

2.7.2 Pengertian SQL SERVER 2000

SQL Server adalah tipe database yang bersifat database realational. Pengertian relasional adalah penyimpanan data dalam bentuk tabel, sedangkan tabel dapat disusun sesuai dengan kelompoknya yang mempunyai sifat sama. Tabel berisi dari kumpulan baris-baris dan kolom-kolom yang jelas dapat saling berhubungan satu sama lain. Database SQL Server terdiri dari dua file yaitu data yang disimpan dalam file LDF dan dalam bentuk MDF. Service Manager merupakan salah satu utilitas SQL Server untuk mengetahui apakah status database pada server aktif atau tidak. Untuk menggunakan aplikasi yang

menggunakan database SQL Server maka service manager harus pada posisi on (hijau). Untuk mengaktifkan bisa dilakukan dengan masuk ke program Microsoft SQL Server dan klik Service Manager. Field Server berisi nama server yang akan dimonitor dan Services menampilkan servis-servis yang ada serta tampilan grafis dari status services. Enterprise Manager merupakan utilitas utama dalam SQL Server. Disinilah pusat perintah untuk mengatur database. Seorang administrator akan sangat mudah mengelola database dengan utilitas ini.

2.8 Pengujian Perangkat Lunak

Agar perangkat lunak dapat dioperasikan secara optimal maka pengujian perangkat lunak sangat diperlukan. Berikut ini adalah macam – macam pengujian perangkat lunak

1. **Testing (Pengujian Perangkat Lunak)**

Adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean.

Pentingnya pengujian perangkat lunak dan implikasinya yang mengacu pada kualitas perangkat lunak tidak dapat terlalu ditekan karena melibatkan sederetan aktivitas produksi di mana peluang terjadinya kesalahan manusia sangat besar dan arena ketidakmampuan manusia untuk melakukan dan berkomunikasi dengan sempurna maka pengembangan perangkat lunak diiringi dengan aktivitas jaminan kualitas.

Meningkatnya visibilitas (kemampuan) perangkat lunak sebagai suatu elemen sistem dan “biaya” yang muncul akibat kegagalan perangkat lunak,

memotivasi dilakukannya perencanaan yang baik melalui pengujian yang teliti. Pada dasarnya, pengujian merupakan satu langkah dalam proses rekayasa perangkat lunak yang dapat dianggap sebagai hal yang merusak daripada membangun.

Sejumlah aturan yang berfungsi sebagai sasaran pengujian pada perangkat lunak adalah:

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan
2. Test case yang baik adalah test case yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya

Sasaran itu berlawanan dengan pandangan yang biasanya dipegang yang menyatakan bahwa pengujian yang berhasil adalah pengujian yang tidak ada kesalahan yang ditemukan. Data yang dikumpulkan pada saat pengujian dilakukan memberikan indikasi yang baik mengenai reliabilitas perangkat lunak dan beberapa menunjukkan kualitas perangkat lunak secara keseluruhan, tetapi ada satu hal yang tidak dapat dilakukan oleh pengujian, yaitu pengujian tidak dapat memperlihatkan tidak adanya cacat, pengujian hanya dapat memperlihatkan bahwa ada kesalahan perangkat lunak.

Sebelum mengaplikasikan metode untuk mendesain test case yang efektif, perekayasa perangkat lunak harus memahami prinsip dasar yang menuntun pengujian perangkat lunak, yaitu:

1. Semua pengujian harus dapat ditelusuri sampai ke persyaratan pelanggan, maksudnya mengungkap kesalahan dari cacat yang menyebabkan program gagal.
2. Pengujian harus direncanakan lama sebelum pengujian itu mulai, maksudnya semua pengujian dapat direncanakan dan dirancang sebelum semua kode dijalankan.
3. Prinsip Pareto berlaku untuk pengujian perangkat lunak, maksudnya dari 80% kesalahan yang ditemukan selama pengujian dapat ditelusuri sampai 20% dari semua modul program.
4. Pengujian harus mulai “dari yang kecil” dan berkembang ke pengujian “yang besar”, Selagi pengujian berlangsung maju, pengujian mengubah focus dalam usaha menemukan kesalahan pada cluster modul yang terintegrasi dan akhirnya pada sistem.
5. Pengujian yang mendalam tidak mungkin karena tidak mungkin mengeksekusi setiap kombinasi jalur skema pengujian dikarenakan jumlah jalur permutasi untuk program menengah pun sangat besar.
6. Untuk menjadi paling efektif, pengujian harus dilakukan oleh pihak ketiga yang independent

Dalam lingkungan yang ideal, perekayasa perangkat lunak mendesain suatu program computer, sebuah sistem atau produk dengan testabilitas dalam pikirannya. Hal ini memungkinkan individu yang berurusan dengan pengujian mendesain test case yang efektif secara lebih mudah. Testabilitas adalah seberapa mudah sebuah program computer dapat diuji. Karena sangat sulit, perlu diketahui

apa yang dapat dilakukan untuk membuatnya menjadi lebih mudah. Procedural dan menggunakannya sebagai pedoman untuk menetapkan basis set dari jalur eksekusi.

Sasaran utama desain test case adalah untuk mendapatkan serangkaian pengujian yang memiliki kemungkinan tertinggi di dalam pengungkapan kesalahan pada perangkat lunak. Untuk mencapai sasaran tersebut, digunakan 4 kategori yang berbeda dari tehnik desain test case: *Pengujian white-box, pengujian black-box, Integrasi Bottom-Up* dan *Integrasi Top-Down.*

2. Jenis – Jenis Pengujian

Pengujian white-box berfokus pada struktur control program. Test case dilakukan untuk memastikan bahwa semua statemen pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji. Pengujian basic path, tehnik pengujian white-box, menggunakan grafik (matriks grafiks) untuk melakukan serangkaian pengujian yang independent secara linear yang akan memastikan cakupan.

Pengujian aliran data dan kondisi lebih lanjut menggunakan logika program dan pengujian loop menyempurnakan tehnik white-box yang lain dengan memberikan sebuah prosedur untuk menguji loop dari tingkat kompleksitas yang bervariasi. Pengujian black-box didesain untuk mengungkap kesalahan pada persyaratan fungsional tanpa mengabaikan kerja internal dari suatu program.

Tehnik pengujian black-box berfokus pada domain informasi dari perangkat lunak, dengan melakukan test case dengan menpartisi domain input dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam.

Metode pengujian graph-based mengeksplorasi hubungan antara dan tingkah laku objek-objek program. Partisi ekivalensi membagi domain input ke dalam kelas data yang mungkin untuk melakukan fungsi perangkat lunak tertentu. Analisis nilai batas memeriksaa kemampuan program untuk menangani data pada batas yang dapat diterima.

Metode pengujian yang terspesialisasi meliputi sejumlah luas kemampuan perangkat lunak dan area aplikasi. GUI, arsitektur client/ server, dokumentasi dan fasilitas help dan sistem real time masing-masing membutuhkan pedoman dan tehnik khusus untuk pengujian perangkat lunak.

Integrasi Top-Down adalah pendekatan incremental dengan menggerakkan ke bawah melalui hirarki control, dimulai dengan control utama. Strategi intergrasi top-down memeriksa control mayor atau keputusan pada saat awal di dalam proses pengujian. Pada struktur program yang difaktorkan dengan baik, penarikan keputusan terjadi pada tingkat hirarki yang lebih tinggi sehingga terjadi lebih dulu.

Strategi top-down kelihatannya tidak sangat rumit, tetapi di dalam praktenya banyak menimbulkan masalah logistic. Biasanya masalah ini terjadi jika dibutuhkan pemrosesan di dalam hirarki pada tingkat rendah untuk menguji secara memadai tingkat yang lebih tinggi.

Pengujian Integrasi Bottom-up memulai konstruksi dan pengujian dengan modul atomic (modul pada tingkat paling rendah pada struktur program). Karena modul diintegrasikan dari bawah ke atas, maka pemrosesan yang diperlukan untuk modul subordinate ke suatu tuingkat yang diberikan akan selalu tersedia dan

kebutuhan akan stub dapat dieliminasi. Strategi integrasi bottom-up dapat diimplementasi dengan langkah-langkah:

1. Modul tingkat rendah digabung ke dalam cluster (build) yang melakukan subfungsi perangkat lunak spesifik.
2. Driver (program control untuk pengujian) ditulis untuk mengkoordinasi input dan output test case
3. Cluster diuji
4. Driver diganti dan cluster digabungkan dengan menggerakkannya ke atas di dalam struktur program.