BAB II

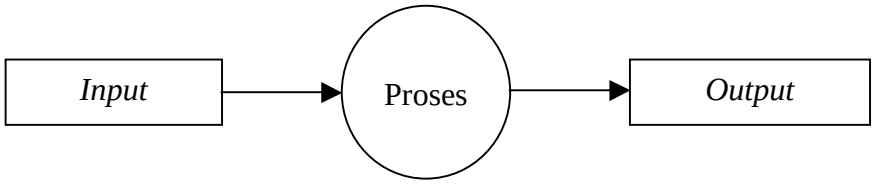
LANDASAN TEORI

2.1 Pengertian Sistem Informasi

Sebelum mengetahui pengertian sistem informasi secara keseluruhan, pertama kali yang harus dikenal adalah istilah–istilah yang tercakup di dalam tiap kata, secara berturut-turut berikut ini akan dijelaskan arti sistem dan informasi.

2.1.1 Bentuk Umum Sistem

Bentuk umum dari suatu sistem terdiri dari atas masukan *(input)*, proses dan keluaran *(output)*, dalam bentuk umum sistem ini terdapat satu atau lebih masukan yang akan diproses dan akan menghasilkan suatu keluaran



Gambar. 2.1 Bentuk Umum Sistem

2.2 Konsep Dasar Sistem Informasi

2.2.1 Sistem

Sistem adalah seperangkat elemen yang membentuk suatu kegiatan/prosedur/bagian pengolahan dalam mencapai tujuan bersama dengan mengoperasikan data pada waktu tertentu untuk menghasilkan suatu informasi.

Beberapa ciri dari sistem yaitu sebagai berikut:

1. Mengarah Pada Tujuan

   Cara kerja sistem ini adalah merangkaikan dan mengkoordinasikan fakta-fakta untuk mencapai tujuan dengan menggunakan aturan-aturan tertentu.

2. Merupakan Suatu Keseluruhan

   Sistem merupakan suatu keseluruhan yang bulat dan utuh, dimana tujuan masing-masing dari bagian yang membentuk sistem akan saling menunjang dan mencapai tujuan dari sistem secara keseluruhan, dan ini berarti bahwa pencapaian tujuan dari salah satu bagian tidak dapat dilakukan dengan mengabaikan pencapaian tujuan dengan bagian yang lainnya.

3. Adanya Keterbatasan

   Sistem memiliki sifat yang terbuka, dimana suatu sistem dapat berinteraksi dengan sistem lainnya yang lebih besar.

4. Adanya Proses Transformasi

   Suatu sistem mempunyai atau melakukan proses transformasi kegiatan yang mengubah suatu *input* atau masukan menjadi suatu *output* untuk mencapai tujuan.

5. Saling Berkaitan

   Sistem terdiri dari elemen-elemen yang saling berkaitan satu elemen dengan elemen yang lain.

2.2.2 Klasifikasi Sistem

Sistem dapat dikelompokkan atau di klasifikasikan menjadi beberapa sudut pandang diantaranya adalah sebagai berikut :

1. Sistem di klasifikasikan sebagai sistem abstrak *(abstract sistem)* dan sistem fisik *(physical sistem),* sistem abstrak terbentuk dari gagasan-gagasan atau konsep-konsep saling berkaitan. Sistem fisik didefinisikan sebagai suatu sistem yang unsur-unsur pendukungnya secara fisik dapat dilihat atau dirasakan dan memiliki keterkaitan antara satu unsur dengan unsur lainnya.
2. Sistem diklasifikasikan sebagai sistem alamiah dan sistem buatan manusia *(human made sistem)*. Yaitu sistem alamiah merupakan sistem yang terjadi melalui proses alam, tidak dibuat manusia, sedangkan sistem buatan manusia merupakan sistem yang dirancang oleh manusia, sistem manusia

yang melibatkan interaksi manusia dengan mesin (Komputer) disebut *human machine system*.

3. Sistem diklasifikasikan sebagai sistem tertentu *(deterministic system)* dan sistem tak tentu *(probabilistic system)*, sistem tertentu beroperasi dengan tingkah laku yang sudah diprediksi. Interaksi diantara bagian-bagian nya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. Dan yang terakhir adalah klasifikasi sistem sebagai sistem tertutup (*closed system)* dan sistem terbuka *(open system)*, dimana sistem tertutup tidak terpengaruh oleh lingkungan lainnya, sistem ini bekerja secara otomatis tanpa ada campur tangan dari pihak luar. Sebaliknya sistem terbuka merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luar.

Pengertian sistem itu sendiri sangatlah luas dan bermacam-macam disesuaikan dengan pendekatan dalam mendefinisikan sistem itu sendiri.

2.2.3 Informasi

Definisi umum informasi dalam pemakaian sistem informasi adalah data yang telah diolah menjadi sebuah bentuk yang lebih berguna dan lebih berarti bagi si pemakai dalam mengambil keputusan. Jadi sistem pengolahan informasi yaitu

pengolahan data dari bentuk tak berguna menjadi berguna (informasi) bagi penerimanya.

Dalam mendapatkan suatu informasi, informasi tersebut harus yang berkualiatas, yaitu yang memiliki ciri-ciri sebagai berikut:

1. Akurat

   Berarti informasi harus bebas dari kesalahan-kesalahan yang biasanya terjadi dan selain itu harus jelas maksud dan tujuannya sehingga *output/*keluaran bisa dipertanggung jawabkan.

2. Tepat Waktu

   Informasi yang dihasilkan atau dibutuhkan tidak boleh terlambat (usang) informasi yang usang tidak akan mempunyai nilai yang baik, sehingga jika digunakan sebagai dasar dalam pengambilan keputusan akan berakibat fatal atau mengalami kesalahan dalam pengambilan keputusan dan tindakan

3. Relevan

   Berarti infornasi tersebut mempunyai manfaat dan informasi yang diterima si pemakai dan pemakai lainnya bisa berbeda-beda.

Dari uraian tersebut diatas maka sistem informasi dapat didefinisikan sebagai suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan data (kejadian), mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu sistem dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi dapat dikatakan sebagai suatu sistem

yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam suatu sistem organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.

Kegiatan sistem informasi mencakup :

1. *Input*, menggambarkan kegiatan untuk menyediakan data untuk diproses.
2. Proses, menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.
3. *Output*, suatu kegiatan untuk menghasilkan laporan proses diatas tesebut.
4. Penyimpanan, suatu kegiatan untuk memelihara dan menyajikan data.
5. Kontrol, suatu aktifitas untuk menjamin bahwa sistem informasi tersebut sesuai yang diharapkan.

2.3 Komponen Sistem Informasi

Sistem informasi menpunyai beberapa komponen yaitu :

1. Perangkat keras *(Hardware)* yaitu semua alat komputer yang secara fisik dapat dilihat dengan jelas seperti: *keyboard, harddisk, monitor*, alat *printer*, *mouse*, dan lain-lain.
2. Perangkat lunak *(Software)* yaitu semua perangkat komputer yang dapat membuat perangkat keras komputer bekerja menjalankan fungsinya.
3. Manusia *(Brainware)* yaitu orang yang mengoperasikan komputer.

2.4 Alat-alat Pengembangan Sistem

Pengembangan sistem merupakan hal yang penting bagi kelangsungan sistem itu sendiri. Pengembangan sistem adalah suatu upaya untuk menjaga efektivitas sistem dalam memenuhi kebutuhan pengguna sistem. Pengembangan sistem dapat berarti menyusun suatu sistem yang baru untuk mnggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang ada.

Alat –alat yang digunakan dalam pengembangan sistem antara lain adalah:

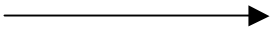
1. *Data Flow Diagram* (DFD)
2. Kamus Data (*Data Dictionary*)

2.4.1 Diagram Arus Data (*Data Flow Diagram*)

Data Flow Diagram (DFD) digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan. DFD memperlihatkan suatu sistem dalam komponen–komponennya serta *interface* (penghubung) antara komponen tersebut. Dalam memperlihatkan aliran data dan pengembangan suatu sistem yang ditinjau dari segi data yang ditampilkan dengan simbol dan aturan tertentu.

a. Arus Data (*Data Flow*)

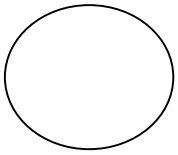
Arus data ini menunjukan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. Pada DFD, arus data digambarkan dengan tanda panah pada gambar berikut :



Gambar 2.2 Data *Flow*

b. Proses

Proses adalah kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Pada DFD, di bawah ini proses digambarkan dalam simbol lingkaran



Gambar 2.3 Proses

c. Kesatuan Luar *(External Entity)*

Kesatuan luar merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang akan memberikan masukan *(input)* atau menerima keluaran *(output)* dari sistem. Di bawah ini kesatuan luar digambarkan dengan simbol kotak persegi panjang.



Gambar 2.4 *External Entity*

d. *File*

Digambarkan dengan dua garis *horizontal*

Gambar 2.5 *File*

Konsep dasar DFD dapat dilakukan dengan analisa *Top Down*, yaitu pemecahan sistem yang besar menjadi beberapa sub-sub sistem yang lebih kecil DFD terdiri dari :

1. Diagram Konteks

Merupakan gambaran sistem secara garis besar dan hanya memperlihatkan kelompok data *input* dan *output*. Diagram konteks adalah suatu diagram yang menggambarkan hubungan sistem dengan sejumlah komponen menjadi bagian penting dalam mendukung keberadaan sistem tersebut, yang akan memberikan masukan atau diberikan keluaran dari sistem sehingga data yang masuk dapat menjadi informasi bagi komponen-komponen tadi. Konteks diagram merupakan *level* teratas dari diagram arus data .

2. Midle Level

Merupakan pemecahan dari tiap–tiap proses yang mempunyai fungsi sama. Pada *middle level* diagaram 0 dipecah menjadi diagram 1,2,3 dan seterusnya yang merupakan penguraian dari diagram konteks.

3. Lowest Level

Merupakan pemecahan dari data *flow* yang ada pada *middle level*.Pemecahan tersebut masih tetap mempunyai fungsi yang sama dari *level* sebelumnya. Untuk *Lowest Level*, pemberian nomor diagram terdiri dari bagian *middle level.*

2.4.2 Kamus Data (Data Dictionary)

Kamus *data (Data Dictionary)* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dapat mendefinisikan data yang mengalir pada sistem dengan lengkap. Kamus data dapat digunakan pada tahap analisa dan perancangan sistem. Pada tahap perancangan sistem, kamus data digunakan untuk merancang masukan *(input)*, merancang laporan–laporan dan *database* .

Dengan adanya kamus data, didapat definisi-definisi dari bentuk–bentuk yang tidak dimengerti dalam DFD yaitu aliran data, *file*, proses dan elemen-elemen data. Arus data pada DFD bersifat *global*, hanya ditunjukan nama arus datanya saja. Keterangan lebih lanjut tentang struktur dari arus data , secara lebih lengkap dapat dilihat di kamus data.

2.5 Pengertian Basis Data (*Database*)

Basis data adalah kumpulan *file-file* yang mempunyai kaitan antara satu *file* lain dengan *file* lain sehingga membentuk suatu bangunan data untuk

menginformasikan suatu perusahaan/instansi dalam batasan tertentu basis data merupakan salah satu komponen penting dalam sistem informasi karena basis data adalah dasar untuk menyediakan informasi bagi para pemakai.

2.5.1 Tujuan Basis Data

Tujuan dari basis data erat hubungannya dengan masalah-masalah yang timbul dalam *file* basis data, yaitu:

1. Penyediaan akses yang *fleksibel*, yaitu untuk memberikan kemudahan dalam menampilkan semua informasi yang diperlukan
2. Pemeliharaan integritas data, basis data berisi *file* yang saling berkaitan yaitu dengan adanya *field* kunci yang menghubungkan kedua *file* tersebut
3. Keamanan data, basis data dapat mendefinisikan prosedur otoritas untuk memastikan bahwa data hanya dapat diakses oleh *user* yang berhak dan dapat mengijinkan *user* yang berbeda untuk melakukan tiap akses yang berbeda pada tahap data yang sama.
4. Menghilangkan ketergantungan data, Program yang berinteraksi dengan *DBMS (Database Management System)* relatif mandiri terhadap data aktual dalam basis data. Hanya perubahan yang mungkin dilakukan terhadap struktur data tanpa membutuhkan perubahan terhadap program aplikasi yang telah ada

5. mengurangi kerangkapan data
6. penggunaan data bersama-sama, data yang sama dapat diakses atau digunakan oleh beberapa *user* pada saat bersamaan.Hal ini menggunakan sistem basis data, tiap aplikasi mempunyai *file* tersendiri, sehingga suatu data tunggal dalam basis data dapat digunakan untuk beberapa kegunaan
7. Standarisasi data, yaitu memberikan fasilitas-fasilitas kamus data untuk mendefiniskan nam-nama data secara rinci

2.5.2 Perancangan Basis Data

Pada perancangan model konseptual, penekanan tinjauan dilakukan pada struktur data dan relasi antar *file*. Pendekatan yang dilakukan pada perancangan model konseptual adalah menggunakan model data *relational*.

Terdapat dua buah teknik perancangan basis data, yaitu

1. Teknik Normalisasi
2. Teknik *Entity Relationship*

2.5.3 Teknik Normalisasi

Proses normalisasi merupakan pengelompokan data elemen menjadi table-tabel yang menunjukan *entity* dan relasinya. Pada proses normalisasinya selalu diuji pada beberapa kondisi. Apabila terdapat kesulitan pada saat menambah/*insert*. menghapus/*delete*, mengubah/*update*, membaca/*retrieve* pada suatu *database* maka relasi tersebut dipecahkan pada beberapa tabel lagi.

Bentuk–bentuk normalisasi :

1. Bentuk tidak normal *(unnormalizd form)*

Pada bentuk tidak normal, tidak ada keharusan mengikuti suatu format tertentu, data mungkin tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

2.* Bentuk Normal kesatu (*1NF / first Normal Form)

Bentuk normal kesatu mempunyai ciri-ciri sebagai berikut :

a. Data dibentuk dalam *flat file (file* datar/rata)
b. Data dibentuk dalam satu *record* demi satu *record* dan nilai dari *field-field* berupa *atomic value*
c. *Set attribute* tidak boleh berulang-ulang atau bernilai ganda *(Multi value)*

3.* Bentuk Normal Kedua (*2NF / Second Normal Form)

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukan kunci harus bergantung secara fungsi pada kunci utama / *primary key*. Sehingga untuk membentuk normal kedua harus ditentukan kunci-kunci *field*. Kunci *field* tesebut harus unik dan dapat mewakili *attribute* lain yang menjadi anggotanya.

4.* Bentuk Normal Ketiga (*3 NF / Thrid Normal Form)

Untuk membentuk normal ketiga maka relasi harus dalam bentuk normal kedua dan semua *attribute* bukan primer harus bergantung hanya pada *primary key* secara menyeluruh.

5. *Boyce –Codd Normal Form* (*BCNF*)

Pada *Boyce – Codde Normal Form*, relasi harus dalam bentuk normal kesatu dan setiap *attribute* harus bergantung fungsi pada *attribute superkey*.

2.5.4 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram dibuat untuk merancang suatu basis data dengan memperlihatkan hubungan antara *entity* atau objek yang terlibat beserta atributenya. Untuk membantu gambaran relasi secara lengkap ada tiga macam relasi dalam hubungan *attribute* dalam suatu *file* :

1. ***One to One Relationship***

   Hubungan antara *file* kesatu dengan *file* kedua adalah satu berbanding satu. Dengan simbol :



Gambar 2.6 *One to one relationship*

2. ***One to many Relationship***

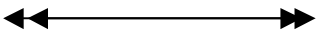
   Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik banyak berbanding satu. Dengan simbol :



Gambar 2.7 *One to many relationship*

3. ***Many to many Relationship***

   Hubungan antara *file* pertama dan *file* kedua adalah banyak ke banyak. Dengan simbol :



Gambar 2.8 *Many to many relationship*

2.6 Sekilas Tentang *Visual Basic*

Program *Visual Basic* adalah bahasa pemrograman berbasis *MS-Windows*. Sebagai bahasa pemrograman yang mutakhir, *Visual Basic* didesain untuk dapat memanfaatkan fasilitas *MS-Windows*, khususnya *MS-Windows 95/97/97* dan *MS-Windows NT*.

Visual Basic juga merupakan bahasa pemrograman *Object Oriented Programming* (*OOP*), yaitu pemrograman yang berorientasi objek. *Visual Basic* menyediakan objek-objek yang sangat kuat, berguna dan mudah dipakai. Dengan fasilitas tersebut, membuat *Visual Basic* menjadi begitu diidamkan oleh para *programmer*.

Visual Basic 6.0 merupakan bahasa pemograman yang digunakan untuk membuat aplikasi *Windows* yang berbasis *GUI (Grafhical User Interface*) atau program yang memungkinkan pemakai (*user)* berkomunikasi dengan menggunakan modus grafik atau gambar.

Visual Basic merupakan programming pemrograman terkendali kejadian (*event-drivent)* artinya program menunggu sampai adanya respon dari pemakai (*user*) berupa *event* atau kajian tertentu. Ketika *event* terdeteksi, kode yang berhubungan dengan *event* (*Procedure event*).

2.6.1 Struktur Aplikasi *Visual Basic* 6.0

Aplikasi (*project*) terdiri dari :

1. *Form*

   Windows/jendela dimana anda akan membuat user interval tampilan.

2. Kontrol/*control*

   Tampilan berbasis grafis yang dimasukan pada *form* untuk membuat interaksi dengan memakai (*text box, scroll,* tombol *command*).

3. *Form* dan *Control*

   Objek yang akan ditentukan.

4. Properti/*properties*

   Nilai atau karakteristik yang dimiliki oleh sebuah objek *Visual Basic*.

   Contoh : *Name, Captions, Size, Color, Possition,* dan *Text.*

5. Metode/*methods*

   Serangkaian perintah yang sudah tersedia pada suatu objek yang dapat diminta untuk mengerjakan tugas khusus.

6. Prosedur kejadian/*Event procedure*

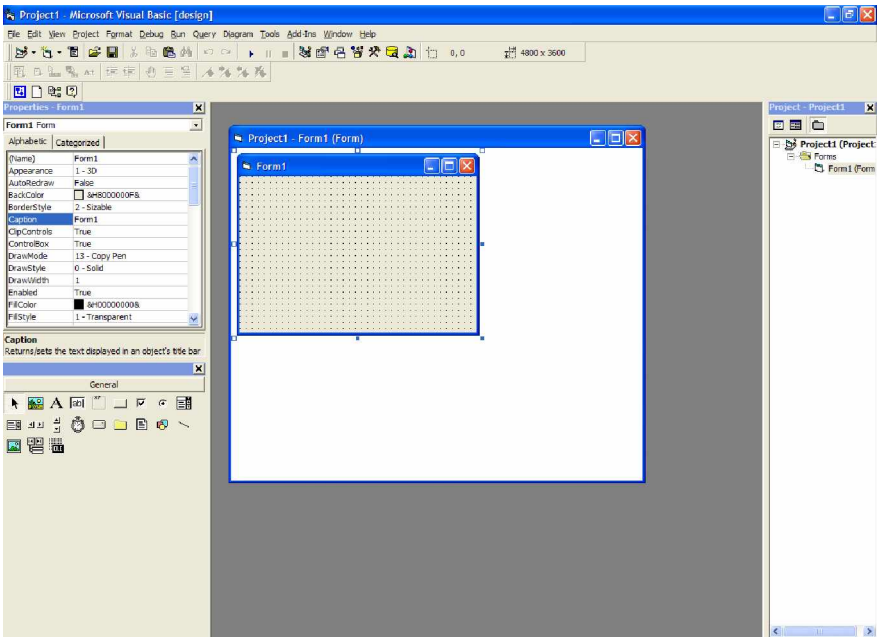
7. Kode yang berhubungan dengan suatu objek.

   Kode ini akan dieksekusi ketika ada respon dari *user* berupa *event* tertentu.

8. Prosedur umum/*general procedure*

   Kode yang tak berhubungan dengan suatu objek. Kode ini harus diminta oleh aplikasi. Modul/*Module* : Kumpulan dari prosedur umum, deklarasi variabel dan didefinisi konstanta yang digunakan oleh aplikasi.

2.6.2 Tampilan Layar *Visual Basic* 6.0



Gambar 2.9 Tampilan *Visual Basic*

1. *Main Windows* (jendela umum) terdiri dari *title bar* (baris judul), *menu bar* dan *toolbar.*
2. *Form Windows* (jendela *form*) adalah pusat dari pengembangan aplikasi *visual basic*.
3. *Project Windows* (jendela proyek) menampilkan daftar *form* dan modul proyek. Proyek merupakan kumpulan dari modul *form*, modul *class*, modul *standart*, dan *file* yang membentuk suatu aplikasi.
4. *Toolsbox*

   kumpulan dari objek yang digunakan untuk membuat *user interface* serta kontrol bagi program aplikasi.

5. *Properties Windows* (*jendela property*) berisi daftar struktur seting *property* yang digunakan pada sebuah abjek terpilih.
6. *Form Layout* Windows (*jendela layout Form*) menampilkan posisi relative terhadap layar *monitor*.

Kemudahan cara pemakaian *Visual Basic* membuat orang awam (pemula) tidak akan terlalu takut untuk memulai mempelajarinya, sedangkan kecanggihannya akan merangsang bagi orang yang sudah mulai mempelajarinya untuk lebih mendalami dan memanfaatkannya.

2.7 Pengenalan Database MySQL

MySQL adalah multi *user database* yang menggunakan bahasa *Strucktured Query Language* (SQL). *MySQL* mampu menangani data yang cukup besar. Perusahaan yang mengembangkan *MySQL* yaitu TcX, mengaku menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data.

SQL adalah bahasa standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industri. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user - friendly* dibandingkan dengan menggunakan *dBASE* atau *Clipper* yang masih menggunakan perintah - perintah pemrograman.

MySQL merupakan *software database* yang paling populer di lingkungan Linux, kepopuleran ini karena ditunjang performa *query* dari *database*-nya yang

saat ini bisa dikatakan paling cepat dan jarang bermasalah. *MySQL* ini juga sudah dapat berjalan pada lingkungan *Windows*.

Perintah untuk mengelola *database* dibagi menjadi 3 kelompok, yaitu:

1. Perintah untuk mendefinisikan data/DDL (*Data Definition Language*).
2. Perintah untuk memanipulasi data/DML (*Data Manipulation Language*).
3. Perintah untuk mengendalikan data/DCL (*Data Control Language*).

2.7.1 DDL (Data Definition Language)

Merupakan Perintah untuk mendefinisikan data. Sebagai contoh, membuat *database*, membuat *table*, membuka *database*, menambah *field* baru, mengubah lebar *field*, menghapus *database*, menghapus *table*, membuat *index*, dan menghapus *index*.

Adapun format penulisan perintahnya adalah sebagai berikut:

1. Membuat *database*.

   Create *Database* <nama*database*>;

2. Menghapus *database.*

   ***Drop Database* <nama*database*>;**

3. Membuat *table.*

   ***Create Table* <namatabel>;**

4. Menghapus *table.*

   ***Drop Table* <namatabel>;**

5. Membuat *file index.*

   ***Create Index* <namaindex> ON <namatabel>**

   (nama*field* 1[,nama*field* 2[,nama*field* 3..]]

6. Menghapus *file index.*

   ***Drop Index* <namaindex> ON <namatabel>;**

7. Menambah *field* baru.

   ***Alter Table*<namatabel>;**

   ***Add* <namafieldbaru> tipe_data;**

2.7.2 DML (Data Manipulation Language).

Merupakan perintah untuk memanipulasi data. Sebagai contoh perintah untuk memanipulasi data adalah memasukan *record* (*insert*), update data (*update*), menghapus *record* (*delete*), dan menampilkan *record* (*select*).

Adapun format penulisan perintahnya adalah sebagai berikut:

1. Perintah *insert*.

   ***Insert Into* <namatabel> *values* (nilai1 [,nilai2[,…]]);**

2. Perintah *update*.

   ***Update* <namatabel> set *field*1=nilai1 [,*field*2=nilai2[,…]] [*where* kondisi];**

3. Perintah *delete*.

   ***Delete from* <namatabel> [*where* kondisi];**

4. Perintah *select*.

   ***Select* {* | *field*1 [*,field*2[,…]]} *from* namatabel [*where* kondisi];**

2.7.3 DCL (Data Control Language).

Merupakan perintah untuk memberi hak otorisasi, untuk mengakses *database*. Perintah-perintah DCL, antara lain *Commit, Rollback, Grant* dan *Revoke.*

Adapun format penulisan perintahnya adalah sebagai berikut:

1. Menambah *user*.

   Insert into user (Host,User)

   ***Values* (‘alamatkomputer’,’nama*user*’)**

2. Mendefinisikan hak akses.

   Update User

   Set namakolom1_Priv=’Y’

   [,namakolom2_Priv=’Y’[,namakolom3_priv=’3’[,…]]

   ***Where user* =’nama*user*’**

3. Mendefinisikan pemakai dan hak akses menggunakan DCL.

   Grant [*privilege*][nama-namakolom]

   ON nama*database*.namatabel

   TO nama*user*@hostkomputer

   IDENTIFIED BY “Password”

   [WITH GRANT OPTION]

4. Mencabut hak akses.

   Revoke [privilege] **[nama-namakolom]**

   ON **nama*database*.namatabel**

   FROM nama*user*@hostkomputer

5. Menghapus pemakai.

   Delete from User where User**='nama*user*'**

   Delete from Db where user**='nama*user*'**

   Delete from Tables_Priv where User**='nama*user*'**

   Delete from Columns_Priv where User**='nama*user*'**

2.8 LAN (*Local Area Network)*

Local Area Network (LAN) merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor suatu perusahaan atau pabrik-pabrik untuk memakai bersama sumberdaya (*resource*, misalnya printer) dan saling bertukar informasi dan tidak jauh berjarak 1 km^2.

Beberapa model konfigurasi LAN, satu komputer biasanya dijadikan sebuah *file server* yang digunakan untuk menyimpan perangkat lunak (*software*) yang mengatur aktivitas jaringan ataupun sebagai perangkat lunak yang dapat digunakan oleh komputer-komputer yang terhubung ke dalam jaringan lokal. Komputer-komputer yang terhubung ke dalam jaringan (*network*) itu biasanya

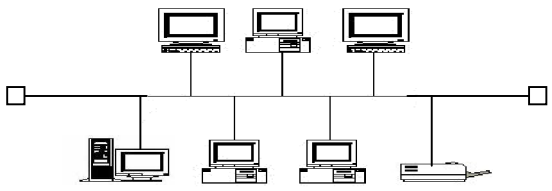
disebut dengan *workstation*. Biasanya kemampuan *workstation* lebih di bawah dari *file server* dan mempunyai aplikasi lain di dalam harddisknya selain aplikasi untuk jaringan. Kebanyakan LAN menggunakan media kabel untuk menghubungkan antara satu komputer dengan komputer lainnya.

2.9 Topologi

Topologi adalah suatu cara menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk jaringan. Masing-masing topologi ini mempunyai ciri khas dengan kelebihan dan kekurangannya sendiri.

2.9.1 Topologi *Linear Bus* (Garis Lurus)

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



Gambar 2.10 Topologi *Linear Bus* (Garis Lurus)

Kelebihan dari topologi *linear bus* adalah :

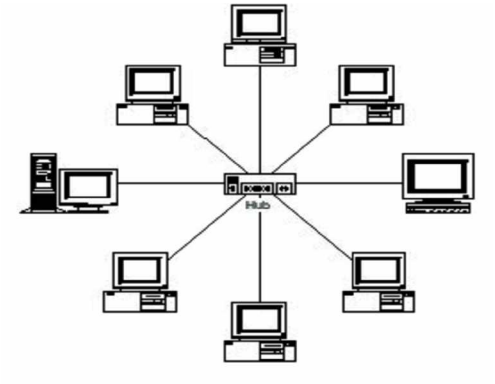
1. mudah di dalam mengkonfigurasi komputer atau perangkat lain ke dalam sebuah kabel utama
2. tidak terlalu menggunakan banyak kabel dibanding dengan topologi star/bintang

Kekurangan dari topologi *linear bus* adalah :

1. seluruh jaringan akan mati apabila ada kerusakan pada kabel utama (*backbone*)
2. membutuhkan terminator pada kedua sisi kabel utamanya
3. sangat sulit mengidentifikasikan permasalahan jika jaringan sedang rusak
4. sangat tidak disarankan dipakai sebagai salah satu solusi pada penggunaan jaringan di gedung besar.

2.9.2 Topologi *Star* (Bintang)

Topologi model ini dirancang yang mana setiap *nodes* (*file server, workstation* dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *concentrator*. Konfigurasi pada jaringan model ini menggunakan kabel *twisted pair* dan dapat juga digunakan pula kabel *coaxciel* dan *fibre optic*.



Gambar 2.11 Topologi *Star* (Bintang)

Kelebihan dari topologi *star* adalah :

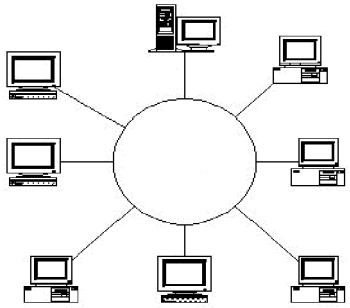
1. mudah didalam pemasangan dan pengkabelan
2. tidak mengakibatkan gangguan pada jaringan ketika akan memasang atau memindahkan perangkat jaringan lainnya.
3. mudah untuk mendeteksi kesalahan atau memindahkan perangkat-perangkat lainnya

Kekurangan dari topologi *star* adalah :

1. membutuhkan lebih banyak kabel daripada topologi *linear bus*
2. membutuhkan *concentrator* dan bilamana *concentrator* tersebut rusak maka semua node yang terkoneksi tidak dapat terdeteksi
3. lebih mahal daripada topologi *linear bus* karena biaya untuk pengadaaan *concentrator*

2.9.3 Topologi *Ring* (Cincin)

Topologi *ring* (cincin) menggunakan teknik konfigurasi yang sama dengan topologi *star* tetapi pada topologi ini terlihat bahwa jalur media transmisi menyerupai suatu lingkaran tertutup menyerupai cincin (lingkaran) sehingga diberi nama topologi bintang dalam lingkaran atau *star-wired ring*.



Gambar 2.12 Topologi *Ring* (Cincin)

Kelebihan dari topologi ring adalah hemat kabel

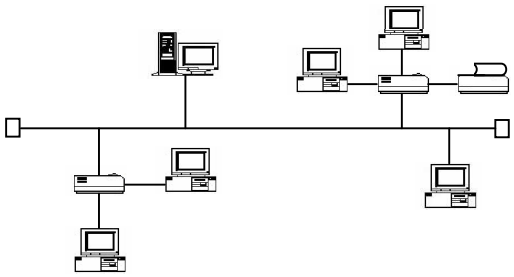
Kekurangan dari topologi ring adalah :

1. peka kesalahan
2. pengembangan jaringan lebih kaku

2.9.4 Topologi *Tree* (Pohon)

Topologi model ini merupakan perpaduan antara topologi *linear bus* dan *star* yang mana terdiri dari kelompok-kelompok dari *workstation* dengan konfogurasi *star* yang terkoneksi ke kabel utama yang menggunakan topologi *linear bus.* Topologi ini memungkinkan untuk pengembangan jaringan yang telah

ada dan memungkinkan untuk mengkonfigurasi jaringan sesuai dengan kebutuhan.



Gambar 2.13 Topologi *Tree* (Pohon)