BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Pengertian sistem yang menekankan pada prosedur di definisikan oleh Jerry Fitzgerald dalam buku yang dialih bahasakan oleh Jogiyanto adalah sebagai berikut : " *Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk menyelesaikan sesuatu sasaran yang tertentu* ".

Kumpulan elemen tersebut terdiri dari *Hardware*, *Software*, Prosedur, Sumber daya manusia dan data yang saling berhubungan dan saling melengkapi satu sama lain untuk mencapai suatu tujuan tertentu**.**

Suatu sistem mempuntai karakteristik atau sifat-sifat tertentu. Karakteristik sistem tersebut adalah sebagai berikut :

1. Komponen Sistem

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

2. Batas Sistem

   Batas (*Boundary*) merupakan daerah yang membatasi antara satu sistem dengan sistem yang lain atau lingkungan luarnya.

3. Penghubung Sistem

   Penghubung (*Interface*) merupakan media penghubung antara satu subsistem dengan subsistem lainnya yang memungkinkan sumber-sumber daya mengalir antar subsistem tersebut.

4. Masukan Sistem

   Masukan (*Input*) adalah energi yang dimasukan kedalam sistem. Masukan dapat berupa masukan perawatan (*Maintenance input*) dan sinyal (*Signal input*). Masukan Perawatan yaitu energi yang dimasukan supaya sistem tersebut beroperasi, sedangkan Masukan Sinyal yaitu energi yang diproses untuk mendapatkan pengeluaran.

5. Keluaran Sistem

   Keluaran (*Output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

6. Pengolahan Sistem

   Suatu sistem dapat mempunyai suatu bagian pengolahan atau proses yang akan merubah masukan (*Input*) menjadi keluaran (*Output*).

7. Sasaran Sistem

   Suatu sistem mempunyai tujuan (*Goal*) atau sasaran (*Objective*). Apabila suatu sistem tidak mempunyai sasaran maka operasi sistem tidak akan ada manfaatnya. Sasaran dari sistem sangat menentukan sebuah masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

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.

2.1.1 Klasifikasi Sistem

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

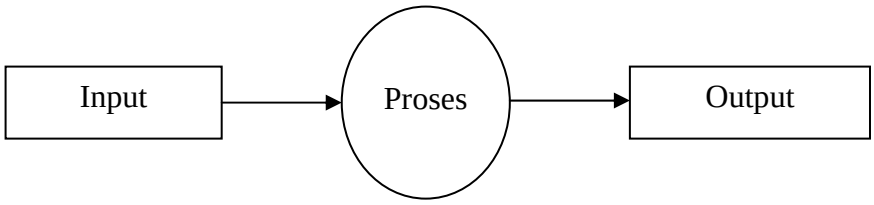
1. Sistem di klasifikasikan sebagai sistem abstrak *(abstract sistem)* dan sistem fisik *(physical sistem)*, sistem abstrak terbentuk dari gagasan-gagasan atau konsep-konsep saling berkaitan. Sistem fisik didefinisikan sebagai suatu sistem yang unsur-unsur pendukungnya secara fisik dapat dilihat atau 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 macine sistem*.
3. Sistem diklasifikasikan sebagai sistem tertentu *(deterministic sistem)* dan sistem tak tentu *(probabilistic sistem)*, sistem tertentu beroprasi 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 sistem)* dan sistem terbuka *(open sistem)*, 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.

2.1.2 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 Informasi

Informasi dapat didefinisikan sebagai berikut :

“ Informasi adalah data yang diolah menjadi bentuk yang akan lebih berarti bagi yang menerimanya “.

Sumber dari Informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal data atau data item. Data adalah kenyataan yang menggambarkan sauatu kejadian-kejadian dan kesatuan nyata. Kejadian (Event) adalah sesuatu yang terjadi pada saat tertentu.

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 Pengertian Sistem Informasi

Informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan, informasi diperoleh dari sistem informasi (*information systems*) atau sering juga disebut dengan *processing systems* atau *information processing systems* atau *information generating system*. Sistem informasi didefinisikan oleh **Robert A Leitch** dan **K. Roscoe Davis** dalam **Jogiyanto HM** sebagai berikut :

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

2.4 Model-model Metode Perancangan Sistem

1. *Waterfall*

Model ini menawarkan cara pembuatan perangkat lunak secara lebih nyata.

Tahapan dalam model *waterfall* terdiri dari:

1. *Sistem Enginering*

   Tahapan ini untuk mendefinisikan sistem, batasan-batasan, maksud dan tujuan dari kebutuhan pembangunan sistem.

2. *Analysis*

   Menentukan kebutuhan yang difokuskan pada perangkat lunak, pemahaman tentang domain informasi, fungsi, kelakuan, performasi, interaksi sesuai dengan kebutuhan dan keinginan pemakai.

3. *Design*

   Merupakan proses multitahap yang difokuskan pada atribut-atribut program yaitu struktur data, arsitektur perangkat lunak, representasi antar muka, procedural algoritma detail, proses perancangan menterjemahkan kebutuhan-kebutuhan hasil analisis

kedalam representasi perangkat lunak sebelum pembuatan kode pemrograman.

4. *Coding / Implementation*

   Tahap penterjemahan hasil perancangan kedalam program-program yang mengguanakan bahasa pemrograman yang sesuai.

5. *Testing*

   Program yang telah dibuat harus diisi, proses pengujian harus difokuskan pada kebenaran logika interal perangkat lunak dan fungsional sistem serta interaksi antara sistem dengan pemakai.

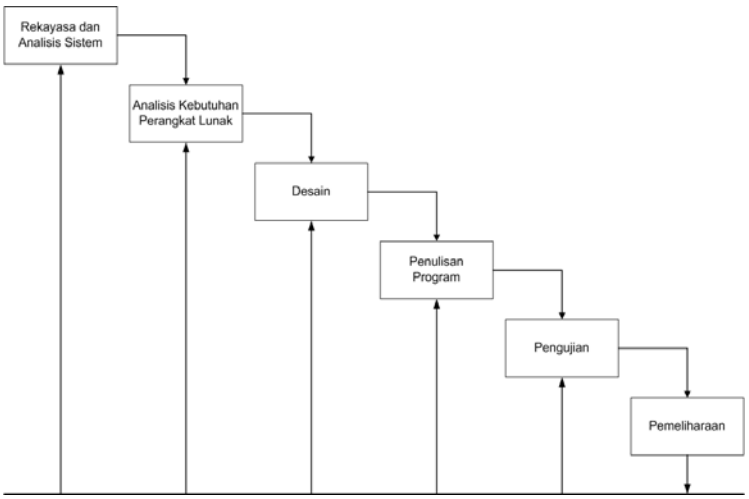
Kelemahan model ini adalah:

1. Proyek-proyek nyata jarang mengikuti alur sekuen yang diusulkan model. Meskipun linear model dapat mengakomodasikan iterasi, namun model melakukan secara tidak langsung. Sebagai hasilnya, perubahan-perubahan dapat menyebabkan kebingungan saat tim pengembangan melakukannya.
2. Sering *customer* sulit menyatakan semua kebutuhannya secara eksplisit. Model ini memerlukan pernyataan eksplisit itu dan sulit mengakomodasi ketidakpastian yang terdapat di awal dari kebanyakan proyek.
3. *Customer* harus memiliki kesabarann. *Versi* yang dapat bekerja dari program tidak akan tersedia sampai akhir dari proyek. Kesalahan besar utama, jika tidak terdeteksi sampai pada program kerja dikaji ulang, maka kesalahan itu dapat mengakibatkan

program sama sekali tidak dapat digunakan. Menyiapkan banyak sumber daya.

Alasan kelemahan model ini adalah:

1. Kebutuhan harus telah ditetapkan di awal di siklus hidup.
2. Kebutuhan divalidasi terlalu lambat.



Gambar 2.2 Model *Waterfall.*

2. *Prototyping*

Langkah-langkah dalam metode *prototyping* meliputi :

1. Mengidentifikasi kebutuhan pemakai

   Pada tahap ini, analisis sistem akan melakukan studi kelayakan dan studi terhadap kebutuhan pemakai baik meliputi model interface, teknik prosedural maupun dalam teknologi yang akan digunakan.

2. Pengembangan *prototyping*

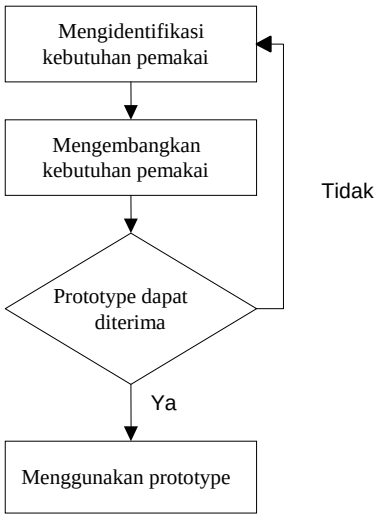
   Pada tahap kedua ini, analisis sistem bekerja sama dengan pemrograman mengembangkan *prototyping* sistem yang akan dibangun.

3. Menentukan *prototyping*

   Apakah dapat dikirim oleh pengguna atau pemakai dan mengidentifikasi sejauh mana pemodelan yang dibuat dapat diterima oleh pemesan.

4. Penggunaan *prototyping*

   Pada tahap ini analisis sistem akan menyerahkan pemrograman untuk mengimplementasikan pemodelan yang dibuatnya menjadi sebuah sistem.



Gambar 2.3 Model Prototyping.

3. *System Development Life Cycle (SDLC)*

System Development Life Cycle (SDLC) merupakan sebuah metode siklus sistem yang terus berputar, untuk menjaga sistem tersebut biasanya digunakan sebagaimana fungsinya (mempermudah pengolahan informasi).

SDLC terdiri dari beberapa fase, namun secara garis besar SDLC terbagi kedalam 4 tahapan yaitu :

1. Perencanaan

   Pada fase ini adalah fase kebijaksanaan atau kesiapan pra proyek dalam merancang/mengembangkan suatu sistem.

2. Analisis

   Fase ini adalah fase yang sangat penting dimana fase ini akan diputuskan sistem apa yang akan dibuat, menilai masih layak atau tidaknya suatu sistem dengan memodelkan sistem dan memilih alternatif sistem yang paling baik.

3. Perancangan

   Fase perancangan adalah fase lanjutan dari analisis, dimana alternatif sistem yang dipilih di desain bai untuk input, output, database dan file semuanya diintegrasikan agar bisa terinplementasikan secara baik.

4. Implementasi

   Pada fase ini secara garis besar terdiri dari tiga bentuk kegiatan yaitu :

   a. *Programing and testing*

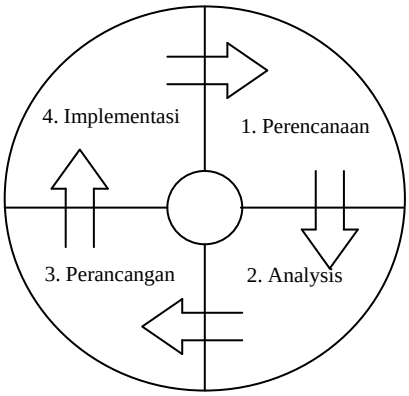
      yaitu mengkonfersikan hasil perancangan ke dalam kegiatan oprasi dengan menggunakan bahasa pemrograman tetentu dan mencoba semua program serta memastikan semua fungsi dapat berjalan secara benar.

b. *Training*

yaitu pelatihan sekaligus pengenalan sistem yang baru kepada user dengan modul panduan sistem.

c. *System hangover*

yaitu penggantian sistem lama ke dalam sistem yang barudan ini merupakan tanggung jawab tim desainer.



Gambar 2.4 Struktur model CDLC

Paradigma perancangan yang digunakan dalam perancangan perangkat lunak untuk tugas akhir ini adalah *Waterfall*. Meskipun terdapat kelemahan, namun model ini memberikan daftar lengkap jumlah aktivitas minimal yang perlu terdapat di model pengembangan perangkat lunak. Umumnya, model-model pengembangan yang lain merupakan superset dari model pengembangan model *waterfall*, memuat aktivitas-aktivitas yang lebih banyak daripada model *waterfall* ini.

2.5 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data, kemana tujuan data yang keluar sistem, dimana data disimpan, proses apa yang dihasilkan data tersebut dan interaksi antara data yang tersimpan dari proses yang dikenakan pada data tersebut.

Data Flow Diagram (DFD) sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misal lewat telepon, surat dan lain-lain) atau lingkunagn fisik dimana data tersebut akan disimpan, misalnya file kertas, hardisk, tape, disket dan lain sebagainya.

DFD merupakan alat yang cukup popular saat ini, karena dapat menggambarkan arus data didalam sistem dengan terstruktur dan jelas. Lebih lanjut DFD merupakan dokumentasi dari sistem yang baik.

Beberapa simbol yang digunakan di DFD antara lain:

a. Kesatuan Luar (*Entity Eksternal*)

   Setiap sistem pasti mempunyai batasan sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima iput dan menghasilkan output kepada lingkungan luarnya. Kesatuan luar (*entity eksternal*) merupakan kesatuan dilingkungan luar sistem, dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan input atau menerima output dari sistem. Kesatuan luar ini kebanyakan adalah salah satu dari yang berikut ini:

1. Suatu kantor, departemen atau divisi dalam perusahaan tetepai diluar sistem yang sedang dikembangkan.
2. Orang atau sekelompok orang di organisasi tetapi diluar sistem yang sedang dikembangkan.

b. Aliran Data (*Data Flow*)

Aliran data si DFD diberikan symbol suatu panah. Aliran data ini mengalir diantara proses (p*rocess*), simpanan data (*data storage*) dan kesatuan luar (*entity eksternal*). Aliran data ini menunjukan aliran dari data yang dapat berupa masukan dri sistem atau hasil dari proses sistem.

c. Proses (*Process*)

Proses adalah kegiatan yang dilakukan oleh orang, organisasi, mesin atau computer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar dari proses yang digambarkan secara umum. Suatu proses dapat ditunjukan dengan simbol lingkaran atau dengan symbol empat persegi panjang tegak dengan sudut- sudutnya yang tumpul. Proses digambarkan dengan simbol lingkaran.

d. Berkas / Simpanan Data (*Data Storage*)

Berkas atau simpanan data merupakan simpanan dari data yang dapat berupa:

1. Suatu file atau database disistem computer.
2. Suatu arsip atau catatan manual.
3. Suatu kotak tempat data di meja seseorang.
4. Suatu tabel acuan manual.

5. Suatu agenda-agenda atau buku.

2.6 Entity Relationship Diagram (ERD)

ERD merupakan notasi grafik dalam pemodelan data konseptual yang mendeskripsikan hubungan antar penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relative kompleks. Dengan ERD kita dapat menguji model dengan menjabarkan proses yang harus dilakukan. ERD menggunakan sejumlah notasi dan symbol untuk menggambarkan struktur dan hubungan antar data. Pada dasarnya ada 3 macam simbol yang digunakan yaitu :

a. Entity

   Merupakan suatu objek yang dapat diidentifikasikan dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat.

b. Atribut

   Entity mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakter entity.

c. Hubungan

   Relationship sebagaimana halnya entitas maka dalam hubungan pun harus dibedakan antar hubungan atau bentuk hubungan antar entity dengan isi dari hubungan itu sendiri.

Relasi antar 2 tabel atau 2 file dapat dikategorikan / dibedakan menjadi 3 macam yaitu :

1. One To One Relationship

   Yang berarti, entitas pada himpunan entitas A berhubungan paling banyak dengan 1 entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan B berhubungan paling banyak 1 dengan entitas pada himpunan entitas B.

2. One To Many Relationship

   Yang berarti, entitas pada himpunan entitas A berhubungan dengan banyak pada himpunan entitas B, tetapi tidak sebaliknya setiap entitas pada himpunan entitas B, hubungan paling banyak dengan satu entitas pada himpunan entitas A.

3. Many To Many Relationship

   Yang berarti, entitas pada himpunan entitas A berhubungan dengan banyak pada himpunan entitas B dan begitu juga sebaliknya, setiap entitas pada himpunan entitas B berhubungan banyak dengan himpunan entitas B.

2.7 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.

2.8 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.

2.8.1 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 nama-nama data secara rinci.

2.8.2 Tahap Perancangan Basis Data

Perancangan basis data merupakan langkah untuk menentukan basis data yang diharapkan dapat mewakili seluruh kebutuhan pengguna. Perancangan basis data terdiri atas perancangan basis data secara konseptual, perancangan basis data secara logis, dan perancangan basis data secara fisis.

Beberapa komponen yang terdapat pada perancangan basis data secara konseptual antara lain:

1. Entitas

   Entitas terkadang disebut tipe entitas atau kelas entitas. Entitas adalah objek yang dapat dibedakan dari objek-objek lainnya.

2. Atribut

   Atribut adalah item data yang menjadi bagian dari suatu entitas. Istilah lain dari attribut adalah properti.

3. Hubungan

   Hubungan adalah asosiasi atau kaitan antara dua entitas.

4. Kekangan

   Kekangan digunakan untuk melindungi integritas data (misalnya, melindungi kesalahan sewaktu pengisian data).

5. Domain

   Domain adalah himpunan yang berlaku bagi suatu atribut. Kekangan domain mendefinisikan nama, tipe, format, panjang, dan nilai masing-masing item data.

6. Integritas Referensial

   Integritas referensial adalah aturan-aturan yang mengatur hubungan antara kunci primer dengan kunci tamu milik tabel-tabel yang berbeda dalam suatu basis data relasional untuk menjaga konsistensi data.

2.9 Sistem Pengelola Basis Data (*Database Manajemen Sistem* / DBMS)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus/ spesifik. Perangkat lunak inilah (sebagai DBMS) yang akan menentukan bagaimana data diorganisasikan, disimpan, diubah dan diambil kembali. Ia juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, keakuratan data dan sebagainya.

Perangkat lunak yang termasuk DBMS seperti dBase III+, dBase IV, FoxBase, Ms-Access, Borland Paradox, Ms-SQL Server, Oracle Borland-

interbase dan lain sebagainya. Salah satu tujuan DBMS adalah untuk menyediakan fasilitas/ antar muka (*interface*) dalam melihat data (yang lebih ramah atau *user friendly*) kepada pemakai.

2.10 Bahasa Basis Data

DBMS merupakan perantara bagi pemakai, dengan basis data dalam disk. Cara berinteraksi / berkomunikasi antar pemakai dengan basis data tersebut diatur dalam suatu bahasa khusus yang diterapkan oleh perusahaan pembuat DBMS. Bahasa ini dapat kita sebut sebagai bahasa basis data yang terdiri dari sejumlah perintah yang diformulasikan dan dapat diberikan user serta dapat dikenali/ diproses oelh DBMS untuk melakukan suatu aksi atau pekerjaan tertentu.

Sebuah bahasa basis data dapat dipilah dalam 2 bentuk yaitu:

1. *Data Definition Language* (DDL)
2. *Data Manipulation Language* (DML)

Struktur atau skema basis data yang menggambarkan desain basis data secara keseluruhan dispesifikasikan dengan bahasa khusus yang disebut *Data Definition Language* (DDL). Dengan bahasa inilah dapat dibuat tabel baru, membuat indeks, merubah tabel, menentukan struktur penyimpanan tabel dan sebagainya, yang mana hasil dari kompilasi perintah *Data Definition Language* (DDL) adalah kumpulan tabel yang disimpan dalam file khusus yang disebut kamus data (*Data Dictionary*).

Sedangkan *Data Manipulation Language* (DML) merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. Manipulasi data dapat berupa:

a. Penyisipan atau penambahan data baru dari suatu basis data.
b. Penghapusan data dari suatu basis data.
c. Pengubahan dari suatu basis data.

Data Manipulation Language (DML) merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data.

2.11 Arsitektur Aplikasi

Untuk mendukung kelancaran program aplikasi yang akan di bangun ini, maka diperlukan beberapa unit komputer, dimana semua komputer tersebut saling berhubungan dan melakukan komunikasi data, agar proses komunikasi data tersebut dapat berjalan dengan baik maka dibutuhkan sebuah media atau perantara yang disebut jaringan komputer.

2.11.1 Pengertian Jaringan Komputer

Jaringan komputer adalah sekelompok komputer otonom yang dihubungkan satu dengan yang lainnya dengan menggunakan protokol komunikasi melalui media transmisi atau media komunikasi sehingga dapat saling berbagi data informasi, program–program, *share* perangkat keras seperti *printer* atau *harddisk*. Tujuan dibangunnya suatu jaringan komputer adalah untuk

membawa data informasi dari sisi pengirim menuju penerima secara cepat dan tepat tanpa adanya kesalahan melalui media transmisi atau media komunikasi tertentu.

2.11.2 *Local Area Network (LAN)*

Sebuah LAN adalah jaringan yang dibatasi oleh area yang relatif kecil, umumnya dibatasi oleh area lingkungan seperti sebuah perkantoran di sebuah gedung, atau sebuah sekolah, dan biasanya sekitar 1 km persegi.. Komputer-komputer yang terhubung ke dalam jaringan itu biasanya disebut dengan *workstation.* Kebanyakan LAN menggunakan media kabel untuk menghubungkan antara satu komputer dengan komputer lainnya.

2.12 Tipe Jaringan

Dalam jaringan komputer terdapat dua tipe jaringan, yaitu :

2.12.1 Model *Peer to Peer*

Pada model jaringan terdapat hanya dua komputer yang saling terhubung. Masing-masing komputer mempunyai kedudukan yang setara atau setingkat sehingga komputer manapun dapat memulai (*initiate*) dan mengendalikan hubungan dalam jaringan. Model Jaringan ini merupakan jenis yang paling sederhana.



Gambar 2.5 Model Jaringan *Peer to peer*

2.12.2 Model *Client Server*

Konsep Jaringan *client-server* ini membedakan dengan jelas kedudukan suatu komputer, mana yang dapat memberikan layanan jaringan (*server*) dan mana yang hanya meminta layanan (*client*). Jika suatu komputer di*install* sebagai *server* yang memberikan segala sumber daya (*resource*), contohnya : *printer*, *modem* kepada komputer lain (*client)* yang terhubung ke jaringan. Untuk dapat saling berkomunikasi antara *server* dan *client*, maka digunakan suatu aplikasi jaringan yang dinamakan *client-server program*, dimana *server* menggunakan aplikasi jaringan yang disebut *server program*, sementara *client* menggunakan *client program* untuk dapat berkomunikasi dengan *server program* pada *server*.

Beberapa sifat dari *client-server* diantaranya sebagai berikut :

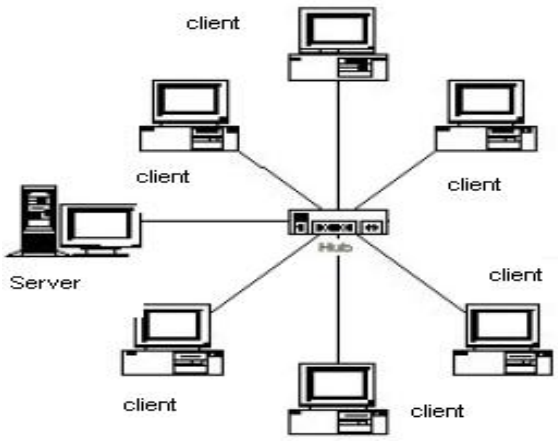
a. *Server* dan *Client* berada pada posisi serta proses yang berbeda.
b. *Server* dan *Client* dapat dijalankan pada mesin yang sama atau berbeda.
c. Setiap proses dapat menyembunyikan informasi tersendiri.
d. Komunikasi yang dilakukan bisa bersifat protokol *peer to peer*.

Koneksitas *client ke server* :

a. *Server* harus terlebih dahulu dijalankan (*running*).
b. *Server* harus menentukan *port* untuk koneksi yang masuk.

c. *Client* harus mengetahui nama *host* dari *server* beserta *port* yang digunakan untuk koneksi.

d. *Client* mengirim pesan ke *server* untuk melakukan suatu pekerjaan (*task*).



Gambar 2.6 Model Jaringan *Client Server*

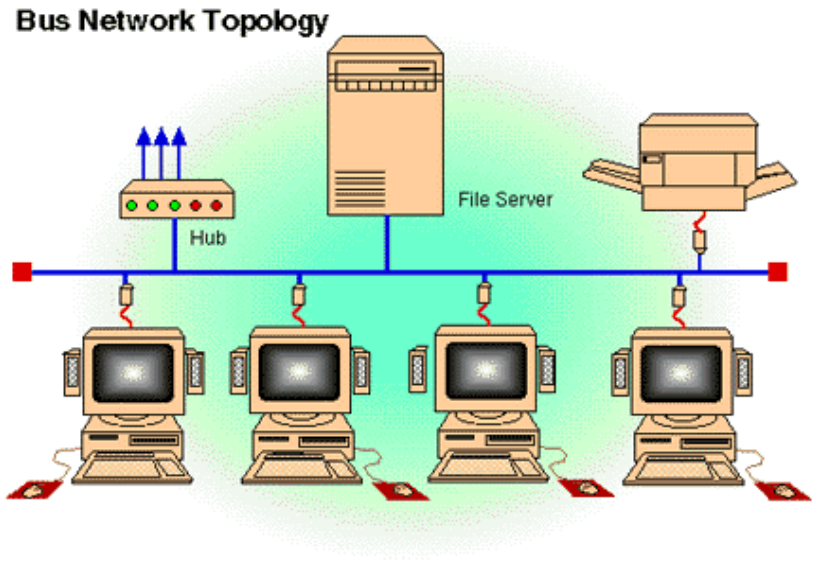
2.12.3 Topologi Jaringan

Topologi jaringan merupakan suatu struktur atau bagaimana menghubungkan antara simpul dan pusat dalam suatu jaringan. Tiap struktur mempunyai keuntungan dan kerugiannya masing-masing. Topologi yang umumnya sebagai berikut :

2.12.3.1 Topologi *Linear Bus*

Topologi sebuah *Linear Bus* / garis lurus terdiri dari satu jalur kabel utama (*backbone*) dimana pada masing-masing ujungnya diberikan sebuah *terminator*. Semua *nodes* pada jaringan (*file server, workstation*, dan perangkat lainnya)

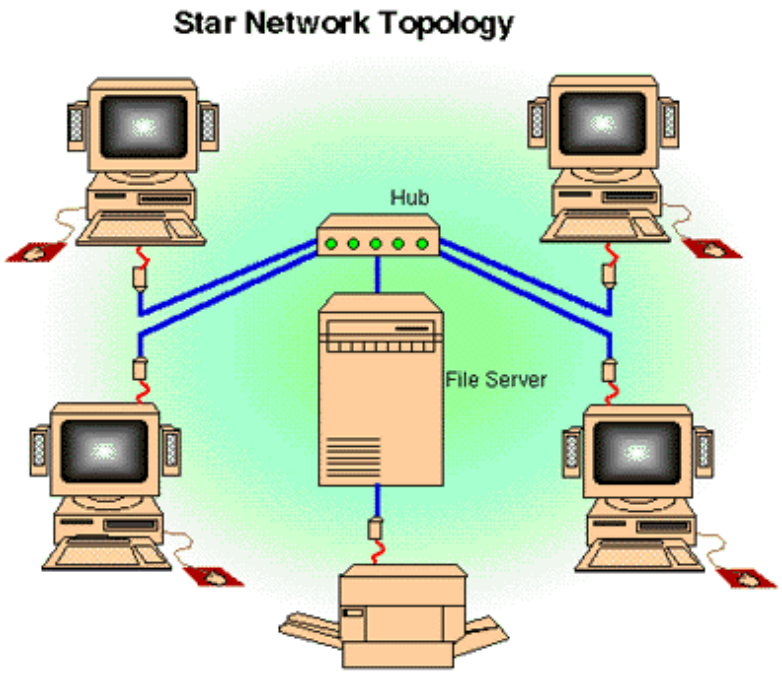
terkoneksi ke sebuah kabel utama. Jaringan-jaringan *Ethernet* dan *Localtalk* menggunakan topologi *linear* ini.



Gambar 2.7 Topologi *Linear Bus*

2.12.3.2 Topologi *Star*

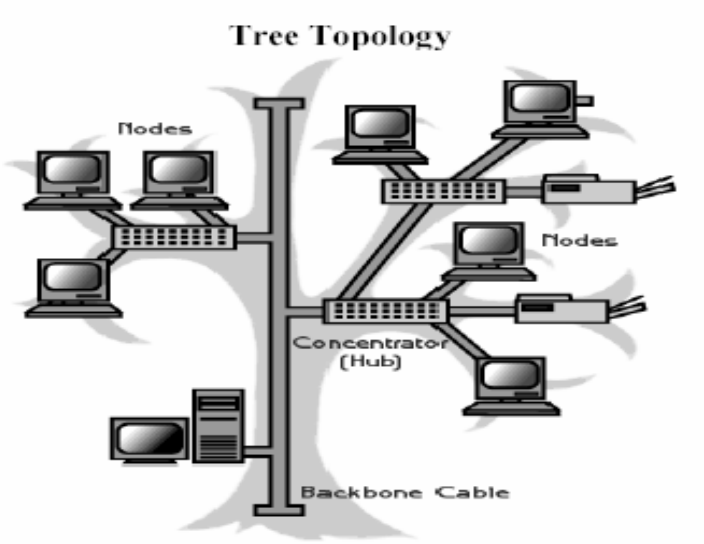
Topologi model ini didesain dimana setap *node* (*file server*, *workstation*, dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *concentrator* baik berupa *Hub* ataupun Switch. Data yang terkirim ke jaringan akan melewati *concentrator* sebelum melanjutkan ke tempat tujuannya. *Concentrator* akan mengatur dan mengontrol keseluruhan fungsi jaringan. Dia juga bertindak sebagai *repeater* dalam skala kecil dari aliran data. Konfigurasi pada jaringan model ini menggunakan kabel *Twisted Pair*. Gambar 2.5 menunjukkan topologi jaringan Bintang / *Star*.



Gambar 2.8 Topologi *Star*

2.12.3.3 Topologi *Tree*

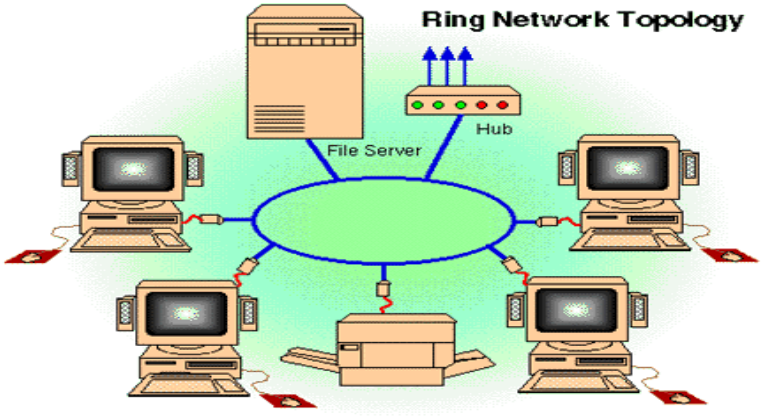
Sebuah topologi model ini merupakan perpaduan antara topologi *Linear Bus* / Garis Lurus dan Bintang, yang terdiri dari kelompok-kelompok dari *workstation* konfigurasi Bintang yang terkoneksi ke kabel utama yang menggunakan topologi *Linear Bus* / Garis Lurus. Topologi ini memungkinkan untuk pengembangan jaringan yang telah ada, dan memungkinkan sebuah perusahaan mengkonfigurasi jaringan sesuai dengan kebutuhannya.



Gambar 2.9 Topologi *Tree*

2.12.3.4 Topologi *Ring*

Dalam topologi *Ring* semua *workstation* dan *server* dihubungkan sehingga terbentuk suatu pola lingkaran atau cincin. Tiap *workstation* ataupun *server* akan menerima dan melewatkan informasi dari satu komputer ke komputer lain, bila alamat- alamat yang dimaksud sesuai maka informasi diterima dan bila tidak informasi akan dilewatkan.



Gambar 2.10 Topologi *Ring*

2.12.4 Jenis Transmisi

Jenis transmisi sinyal data atau informasi dalam suatu media komunikasi dapat dikelompokkan menjadi dua model, yaitu tansmisi paralel dan serial.

2.12.4.1 Transmisi Pararel

Pada transmisi paralel, satu konektor yang terdiri dari tujuh atau delapan bit (*ASCII*) ditransmisikan secara serentak setiap saat, seperti yang ditunjukkan pada gambar 2.7. Misalnya, bila digunakan kode *ASCII*, maka dibutuhkan sebanyak delapan jalur untuk mentransmisikan sekaligus 8 *bit* untuk satu karakter kode *ASCII*.

Pengiriman dengan model paralel ini memiliki kecepatan yang tinggi, karena setiap saat yang ditransmisikan secara paralel adalah bit–bit yang mewakili satu karakter. Komunikasi paralel biasanya digunakan untuk komunikasi jarak dekat, biasanya transmisi jenis ini digunakan untuk mentransmisikan sinyal di dalam komputer atau antara komputer ke *printer*.

2.12.4.2 Transmisi Serial

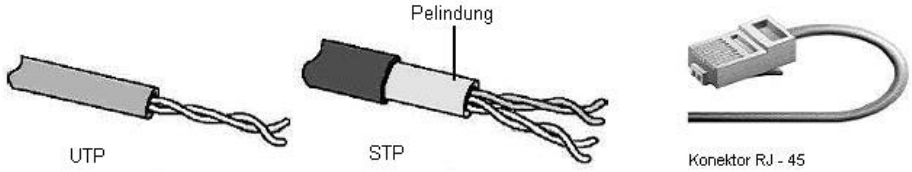
Transmisi model ini merupakan bentuk transmisi yang secara umum dipergunakan. Pada gambar 2.8 menunjukkan transmisi serial yang masing – masing *bit* dari suatu karakter dikirimkan secara berurutan, yaitu *bit* per *bit*, dimana satu *bit* diikuti oleh bit berikutnya. Dalam sistem ini, penerima akan mengumpulkan sejumlah *bit* (untuk sistem ASCII = 8 *bit*) yang dikirimkan oleh pengirim untuk kemudian dirakit menjadi satu karakter. Meskipun kecepatannya

lebih lambat dibandingkan dengan transmisi paralel, namun transmisi model ini paling banyak digunakan dalam lingkungan jaringan komputer.

2.12.5 Media Transmisi

2.12.5.1 Media Transmisi Kabel Twisted Pair

Kabel *Twisted Pair* ini terbagi menjadi dua jenis yaitu STP (*Shielded Twisted Pair*) dan UTP (*Unshielded.Twisted Pair*). *Shielded* adalah jenis kabel yang memiliki selubung pembungkus sedangkan *unshielded* tidak mempunyai selubung pembungkus, seperti ditunjukan pada gambar 2.9. Untuk koneksinya kabel jenis ini menggunakan konektor RJ-11 atau RJ-45. *Twisted pair* umumnya lebih handal (*reliable*) dibandingkan dengan *thin coax* karena HUB mempunyai kemampuan *data error correction* dan meningkatkan kecepatan transmisi. Saat ini ada beberapa kategori dari kabel *twisted pair*. Kategori 5 adalah yang paling *reliable,* memiliki kompabilitas yang tinggi, dan paling disarankan. Kategori 5 berjalan baik pada 10 Mbps dan *Fast Ethernet* (100Mbps). Kabel kategori 5 dapat dibuat *straight-through* atau *crossed*. Kabel *straight-through* digunakan untuk menghubungkan komputer ke HUB. Kabel *crossed* digunakan untuk menghubungkan HUB ke HUB atau komputer dengan komputer. Panjang kabel maksimum kabel *Twisted-Pair* adalah 100 meter.

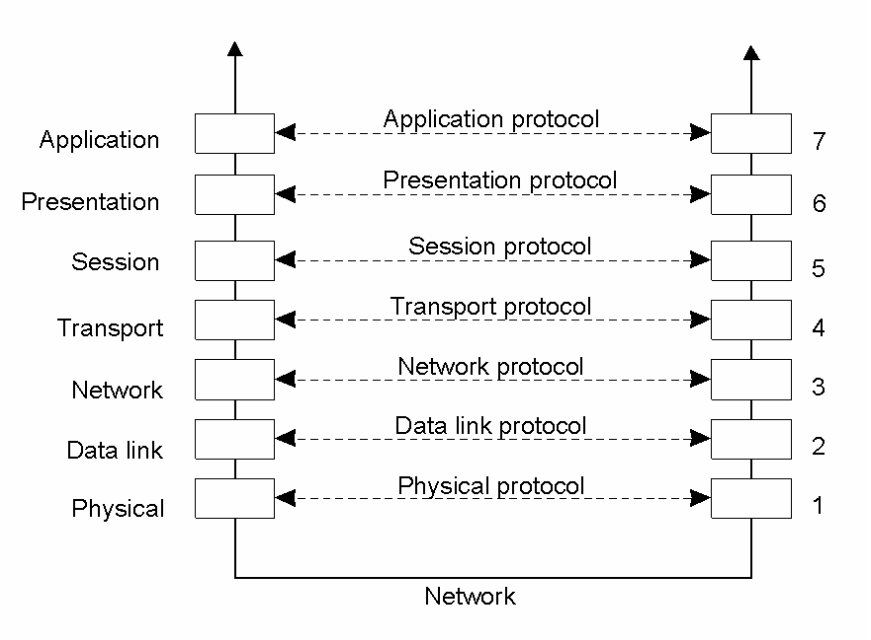


Gambar 2.11 Kabel *UTP, STP* dan *Konektor RJ-45*

2.12.6 Model *OSI*

Model *OSI* merupakan sistem terbuka yang dapat diartikan sebagai suatu sistem yang terbuka untuk berkomunikasi dengan sistem-sistem yang lainnya. Model referensi *OSI* ini memiliki tujuh *layer*. Adapun prinsip-prinsip yang digunakan bagai ketujuh *layer* tersebut adalah:

a. Sebuah *layer* harus dibuat bila diperlukan tingkat abstraksi yang berbeda.
b. Setiap *layer* harus memiliki fungsi-fungsi tertentu.
c. Fungsi-fungsi tersebut sesuai dengan standarisasi internasional
d. Batas-batas layer harus dipilih untuk meminimalkan aliran informasi yang melewati *interface*.



Gambar 2.12 Model Referensi *OSI*

Sebuah *layer* tidak mendefinisikan protokol tunggal, tapi mendefinisikan suatu fungsi komunikasi data yang dapat dilakukan oleh sejumlah protokol. Jadi

setiap *layer* dapat berisi banyak protokol, masing-masing menyediakan *service* yang cocok dengan fungsi *layer* tersebut. Sebagai contoh, *file transfer protocol* (FTP) dan *electronic mail protocol (E-Mail Protocol)* yang keduanya menyediakan *service* pada *user* dan keduanya merupakan bagian dari *layer* aplikasi.

Setiap protokol berkomunikasi dengan *peer* (pasangan) protokol yang sama dalam *layer* yang sama dalam *remote-system* misalnya FTP lokal merupakan *peer* dari FTP *remote*. Jadi secara abstrak setiap protokol hanya peduli terhadap komunikasi dengan *peer*-nya, tak perduli dengan *layer* di bawah dan di atasnya.

Adapun urutan *layer* dari *layer 1* sampai *layer* 7 sebagai berikut :

1. **Lapisan *Physical***
    a. Lapisan ini bertugas mendefinisikan besaran-besaran elektrik yang dipakai untuk berhubungan, misalnya atas tegangan yang benar (*Valid*) logika 0 dan 1, bentuk konektor yang digunakan, cara penyandian isyarat pada kabel dan lain-lain.
    b. Melaksanakan hubungan fisik melalui jalur transmisi .
2. **Lapisan *Data Link***
    a. Lapisan ini bertanggung jawab terhadap pengiriman data yang benar (*error free*) .
    b. Mengirim data secara paket dan memeriksa kesalahan pada pengiriman data.
    c. Menetapkan *prosedur* pengaksesan.

3. **Lapisan *Network***

    a. Lapisan ini menentukan jalur atau *route* yang harus ditempuh oleh paket untuk mencapai tujuannya, yang biasanya disebut *routing layer*.

    b. Pada lapisan ini pula menentukan *prosedur* untuk menghindari terjadinya kemacetan serta mengalihkan perjalanan paket ke jalur yang lain serta memungkinkan bukan jalur yang pendek.

4. **Lapisan *Transport***

    a. Lapisan ini mengatur bagaimana data atau informasi itu akan dibawa ke tempat tujuan termasuk juga jaminan kualitas dari pelayanan pengiriman data.

    b. Lapisan ini pula memilih fasilitas pengiriman yang sebagaimana yang dikehendaki.

5. **Lapisan *Session***

    a. Lapisan ini berfungsi menentukan kapan dimulai dan kapan diakhirinya suatu babak atau *session* komunikasi atau koneksi.

    b. Suatu babak akan dimulai jika babak sebelumnya telah selesai.

    c. Melihat data yang sudah dikirim dan diterima oleh proses.

6. **Lapisan *Presentation***

    a. Lapisan ini mendefinisikan bagaimana aplikasi dapat memasuki jaringan dan mengatur supaya data yang dipertukarkan atau dikirim ke lapisan aplikasi atau pemakai akhir dapat dimengerti bentuknya.

    b. Dalam lapisan ini, peralatan atau sistem yang menggunakan format data yang berlainan dapat saling berkomunikasi .

c. Lapisan ini juga menyediakan *encryption* ataupun kompresi data jika diperlukan.

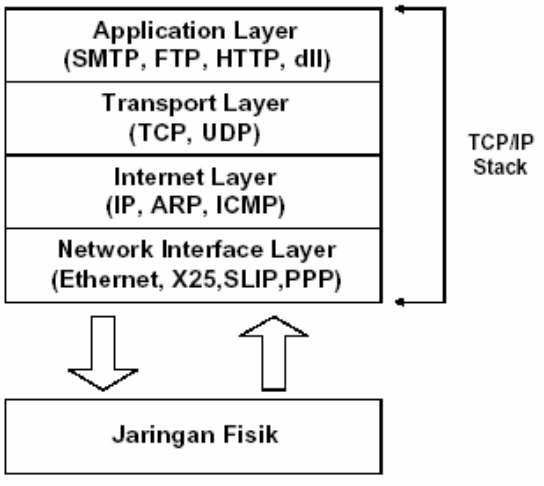
7. Lapisan *Aplication*

a. Lapisan ini berfungsi mengelola interaksi antara program dan pemakai.

b. Lapisan ini juga menerima perintah atau input dari pemakai dan mengembalikan kode *error* atau pesan kesalahan kepada pemakai jika terjadi *error*.

2.12.7 Model *TCP/ IP*

Model ini terdiri dari empat lapisan. Pada model ini tidak memiliki *presentation* dan *session layers*. Fungsi kedua lapisan ini dapat dilakukan sesuai dengan kebutuhan.



Gambar 2.13 Model *TCP/IP*

Berikut merupakan penjelasan lapisan-lapisan atau *layer-layer* pada model *TCP/IP* mulai dari lapisan teratas :

1. *Aplication Layer*
    a. *Layer* ini melingkupi semua proses yang menggunakan protokol *transport layer* untuk mengirimkan data.
    b. Karena *layer* ini berhubungan langsung dengan *service* yang ditawarkan pada pemakai jaringan maka protokol-protokol baru masih ditambahkan yang memperkaya pelayanan pada *user*.
2. *Transport Layer*

Dua protokol terpenting dalam layer ini adalah TCP *(Transmission Control Protocol)* dan UDP *(User datagram Protocol)*. TCP menyediakan pelayanan pengiriman data yang handal dengan deteksi dan koreksi kesalahan dari ujung ke ujung (*end to end*). Sedangkan UDP menyediakan pelayanan pengiriman *datagram* yang *connection-less* dan tanpa dilengkapi deteksi dan koreksi kesalahan. Kedua *protokol* mengirimkan data antara *layer* aplikasi dan *layer internet*. TCP merupakan protokol yang *connection oriented* dan *handshake*.

2.13 Sistem Client-Server

Menurut Gallaugher dan Ramanathan (1996) "*client-server* adalah *client* mengirim permintaan ke *server*, *server* menterjemahkan pesan, kemudian berusaha memenuhi permintaan", sedangkan menurut Blaha dan Premerlani (1998): *client-server* adalah suaktu arsitektur dimana sumber daya *server* menyediakan komputasi untuk banyak komponen *client*, *client* dapat mengakses suatu *server* atau *multiple server*. Client dan *server* bisa berjalan pada mesin yang

sama atau berbeda, ditulis dalam berbagai bahasa dan menggunakan sistem oprasi yang berbeda.

Secara umum *client-server* adalah arsitektur jaringan aplikasi yang memisahkan *client* dari *server*, setiap satuan perangkat lunak *client* berhubungan dengan perangkat lunak *server*. *Client-Server* adalah arsitektur bersekala dimana setiap komputer atau proses pada jaringan berperan sebagai *client* atau *server*, piranti lunak *server* pada umumnya walaupun tidak selalu akan menjalankan komputer dengan kekuatan penuh yang secara khusus menjalankan aplikasi bisnis. Sedangkan piranti lunak *client* pada umumnya berjalan pada *PC* atau *Workstation* biasa. *Client* akan mendapatkan seluruh informasinya dan mengirimkannya kepada perangkat lunak *server* untuk sebuah keperluan, sebagai contoh konfigurasi file, kuota penyimpanan, program aplikasi bisnis atau untuk membebaskan intensifitas pekerjaan komputasi dan mengkondisikan komputer *client* bebas dan menjalankan pekerjaan lainnya.

2.14 Pengertian Obat

Obat adalah senyawa kimia yang dapat dipakai pada atau diberikan pada manusia atau binatang sebagai alat bantu diagnosis.

2.14.1 Manfaat Obat:

a. Sebagai Pengobatan dan pencegahan penyakit atau keadaan abnormal lainnya.
b. Untuk menghilangkan rasa sakit, penderitaan, untuk mengendalikan atau memperbaiki setiap keadaan fisiologik dan patologik.

2.14.2 Obat terbagi menjadi dua macam:

a. Obat Generik

   Obat yang diproduksi oleh perusahaan negeri. Contoh: *Parasetamol, Antalgin, amoxicillin,* dll.

b. Obat Paten

   Obat yang di produksi oleh perusahaan-perusahaan swasta. Contoh: *Lanamol, Neuromach, Amoxsan,* dll.

Dengan macam-macam obat yang tersedia, para pemakai obat (pasien) dapat memilih obat yang sekiranya sesuai dengan kemampuan *Financial*-nya, dikarenakan obat *generic* lebih terjangkau harganya daripada obat paten. Akan tetapi obat *generic* tersebut ada kekurangannya yaitu dari segi kandungan isi obat, sedangkan obat paten kandungan isinya lebih banyak dan lebih baik dibandingkan dengan obat *generic*, akan tetapi obat tersebut harganya jauh lebih mahal dari pada obat *generic.*

2.15 Perangkat Lunak Pendukung

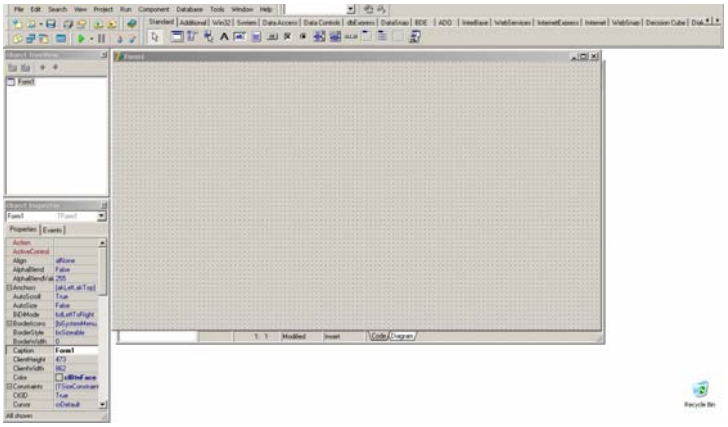
2.15.1 Sekilas Tentang *Borland Delphi 7*

Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual. Keunggulan bahasa pemograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan pemrogramannya yang terstuktur. Keunggulan lain dari Delphi adalah dapat digunakan untuk

merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis Windows.

Khusus untuk pemrograman database, Borland Delphi menyediakan fasilitas objek yang kuat dan lengkap yang memudahkan programmer dalam membuat program. Format database yang dimiliki Delphi adalah format database Paradox, dBase, MS. Access, ODBC, SyBASE, Oracle dan lain-lain.

2.15.2 Tampilan Layar Borland Delphi 7



Gambar 2.14 *Tampilan Borland Delphi*

1. *Main Window, Menubar* dan *Toolbar.*



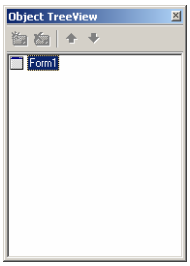
Gambar 2.15 *Menubar*

2. *Form Designer* Merupakan suatu projek yang dipakai sebagai tempat untuk merancang program aplikasi.



Gambar 2.16 *Lembar kerja Form*

3. *Object TreeView* menampilakan diagram pohon dari komponen-komponen yang bersifat visual maupun nonvisual yang telah terdapat dalam *form*, *data module*, atau *frame*. *Object TreeView* juga menampilkan hubungan logika antar komponen. Apabila anda mengklik kanan salah satu item yang terdapat di dalam diagram pohon, Anda dapat melihat konteks menu komponen versi sebelumnya. Untuk mengakses menu secara penuh, kilik kanan pada komponen yang sama dalam *form*, *data module*, atau *frame*.



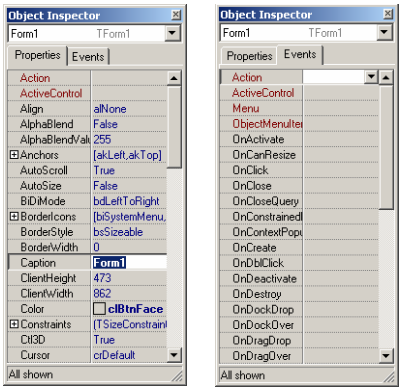
Gambar 2.17 *Object TreeView.*

4. *Component Platte* berisi kumpulan icon yang melambangkan komponen-komponen yang terdapat pada VCL (Visual Component Library).



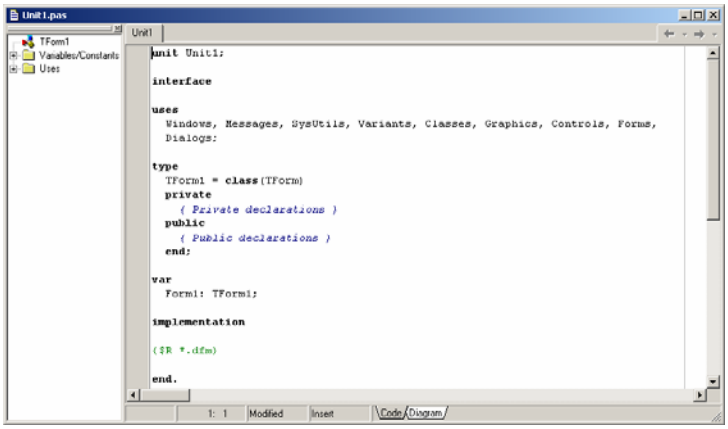
Gambar 2.18 *Component Platte*

5. *Object Inspector* digunakan untuk mengubah properti atau karakteristik dari sebuah komponen. *Object Inspector* terdiri dari dua tab, yaitu Properties dan Events.



Gambar 2.19 *Lembar kerja Object Inspector.*

6. *Code Editor* Merupakan tempat dimana dapat menuliaskan kode program. Pada bagian ini menuliskan pernyataan-pernyataan dalam Object Pascal. Keuntungan bagi pemakai *Delphi* adalah bahwa tidak perlu menuliskan kode-kode sumber, karena *Delphi* telah menyediakan kerangka penulisan sebuah program.



Gambar 2.20 *Lembar kerja Code Editor*

2.15.3 Sekilas Tentang MySQL

MySQL adalah suatu sistem manajemen database. Suatu database adalah sebuah kumpulan data yang terstruktur. Untuk menambahkan, mengakses, dan memproses data yang tersimpan pada suatu database komputer anda memerlukan sistem manajemen database seperti MySQL. Karena komputer sangat unggul dalam menangani sejumlah besar data, sistem manajemen database memainkan suatu peranan yang penting dalam komputasi, baik sebagai utility stand-alone maupun bagian dari aplikasi lainnya.

Suatu database relasional menyimpan data dalam tabel-tabel terpisah. Hal ini memungkinkan kecepatan dan fleksibilitas. Tabel-tabel yang dihubungkan dengan relasi yang ditentukan membuatnya bisa mengkombinasikan data dari beberapa tabel pada suatu permintaan. Bagian SQL dari kata MYSQL berasal dari "Structured Query Language" bahasa paling umum yang dipergunakan untuk mengakses database.

Beberapa perintah dasar SQL yang sering dipergunakan pada MySQL adalah sebagai berikut :

a. Create Database

   Yaitu perintah yang digunakan untuk membuat database baru.

   Sintaks : Create database database_nama database

b. Drop Database

   Yaitu perintah yang digunakan untuk menghapus database.

   Sintaks : Drop Tabel Tabel_name

c. Create Tabel

Yaitu perintah yang digunakan untuk membuat tabel baru.

Sintaks Create Tabel tabel_name (create_definition)

d. Describe

Yaitu perintah yang digunakan untuk mendeskripsikan tabel atau logam

Sintaks Describe (Desc) tabel [colum]

e. Alter Tabel

Yaitu perintah yang digunakan untuk memodifikasi tabel

Sintaks Alter [Ignor] Tabel table_name

f. Drop Tabel

Yaitu perintah yang digunakan untuk menghapus tabel

Sintaks Drop Tabel tabel_name [tabel_name..]

g. Delete

Yaitu perintah yang digunakan untuk menghapus record dri tabel

Sintaks Delete From tabel_name Where Where_definiition

h. Select

Yaitu perintah yang digunakan untuk query ke database

Sintaks select*from tabel_name

Select Field from tabel_name