

2. DASAR TEORI

Untuk membuat suatu program aplikasi diperlukan informasi atau teori yang jelas dan tepat mengenai sistem yang dibuat dan bahasa pemrograman yang akan digunakan. Dalam bab ini, menjelaskan mengenai teori-teori dasar pendukung yang digunakan dalam melakukan perancangan dan pembuatan aplikasi *chatting*, konsep dan perancangan basis data, istilah dalam *database*, *data flow diagram*, serta pembahasan teori pemrograman Delphi.

2.1. *Chatting*

Chatting adalah suatu fasilitas dalam Internet untuk berkomunikasi sesama pemakai Internet yang sedang *on-line*. Komunikasi dapat berupa teks atau suara. (“Ngobrol” par.1).

2.1.1. *Internet*

Chatting dapat dilakukan di dua jaringan, yaitu internet dan intranet. Istilah internet berasal dari bahasa latin inter, yang berarti ”antara”. Secara kata per kata internet berarti jaringan antara atau penghubung. Memang itulah fungsinya, internet menghubungkan berbagai jaringan yang tidak saling bergantung pada satu sama lain sedemikian rupa, sehingga mereka dapat berkomunikasi. Sistem apa yang digunakan pada masing-masing jaringan tidak menjadi masalah, apakah sistem DOS atau UNIX.

Sementara jaringan lokal biasanya terdiri atas komputer sejenis (misalnya DOS atau UNIX), internet mengatasi perbedaan berbagai sistem operasi dengan menggunakan bahasa yang sama oleh semua jaringan dalam pengiriman data. Pada dasarnya inilah yang menyebabkan besarnya dimensi internet.

Internet merupakan hubungan antar berbagai jenis komputer dan jaringan di dunia yang berbeda sistem operasi maupun aplikasinya dimana hubungan tersebut memanfaatkan kemajuan media komunikasi (telepon dan satelit) yang menggunakan protokol standar dalam berkomunikasi yaitu protokol TCP/IP. Internet merupakan media komunikasi dan informasi modern (“Internet” par. 1-3).

2.1.2. Intranet

Intranet adalah jaringan komputer yang khusus untuk penggunaan pada lingkungan didalam batasan suatu organisasi. Dilihat dari sudut teknisnya, intranet didefinisikan sebagai penggunaan teknologi Internet dan WWW (*World Wide Web*) didalam sebuah jaringan komputer lokal (LAN). *Local Area Network* (LAN) adalah sekumpulan komputer-komputer yang saling dihubungkan pada suatu daerah atau lokasi tertentu. Intranet memaksimalkan penggunaan LAN tersebut dengan menambahi kemampuan-kemampuan internet kedalamnya. (Odom, 2004).

2.1.3. Perangkat Keras Intranet

Intranet sebenarnya adalah sebuah jaringan komputer lokal (LAN) yang diberikan teknologi internet atau *World Wide Web*. Untuk membangun intranet, LAN (*Local Area Network*) harus memiliki sebuah jaringan komputer lokal terlebih dahulu. Jaringan komputer lokal terdiri dari beberapa komputer, dimana saling terhubung didalam satu lokasi. Semua komputer didalam LAN dapat saling berhubungan dan mengakses sumber daya yang ada pada komputer yang lain. Sumber daya (*resource*) yang dimaksud disini bisa berupa data pada *server*, data pada komputer lain, printer, periferal lain, dan sebagainya.

Client-server merupakan komputer *server* dapat didefinisikan sebagai komputer pusat yang memberikan suatu data atau dokumen, sedangkan komputer yang meminta dokumen disebut sebagai *client*. Komputer *client* biasanya berupa PC biasa dan digunakan oleh pemakai atau pegawai perusahaan untuk meminta informasi dari *server*, sedangkan komputer *server* digunakan untuk menyimpan data dan program serta menyediakan pelayanan network kepada *client*. Jika kedua komputer ini disatukan, maka akan didapat sebuah model jaringan komputer *client server*.

Komponen LAN merupakan komponen perangkat keras yang dibutuhkan untuk membangun intranet, yaitu komputer *server*, komputer *Workstation*, adapter *network* atau *network interface card*, kabel, dan komponen pendukung LAN. Semuanya itu juga dibutuhkan dalam membuat internet.

Server adalah komputer pusat yang mengendalikan seluruh aktivitas jaringan komputer. Juga digunakan sebagai penyimpanan pusat data / program yang dapat diakses oleh komputer *client*.

Workstation adalah komputer yang digunakan oleh setiap orang untuk mengakses komputer pusat tetapi pada lingkungan networking yang ada sekarang ini. Istilah *Workstation* juga sering dipertukarkan dengan istilah *client* yang lebih cocok dengan model networking *client-server*. *Workstation* tidak dapat menjalankan tugas pemrosesan, tugas pemrosesan seluruhnya dilakukan oleh *server* sehingga beban pemrosesan dan lalu lintas *network* terpusat pada *server* tersebut. Cara ini dapat dilakukan apabila lalu lintas data tidak terlalu padat dan *server* mampu menangani seluruh tugas pemrosesan yang diminta oleh setiap *client* (*workstation*).

Network Interface Card (NIC) atau sering juga disebut dengan *adapter card* adalah sebuah kartu elektronik yang dipasang pada semua komputer yang ingin dihubungkan pada suatu *network* (termasuk komputer *server* dan *client*). Kartu NIC inilah yang berfungsi menghubungkan komputer-komputer pada suatu LAN dan mengizinkan semua komputer tersebut dapat saling berkomunikasi.

Kabel jaringan (*network*) berfungsi menghubungkan antara NIC-NIC yang dipasang pada setiap komputer. Kabel yang dapat digunakan untuk menghubungkan *network* ada bermacam-macam, bergantung pada kebutuhan dan kecepatan yang anda inginkan

Jenis-jenis kabel yang digunakan untuk *network* diantaranya adalah :

- Shielded twisted pair (STP)
- Unshielded twisted pair (UTP)
- Kabel koaksial
- Kabel serat optik
- *Network* tanpa kebel

2.2. Jaringan Komputer

Jaringan komputer adalah sekumpulan perangkat yang dapat digunakan untuk menyimpan dan manipulasi data elektronis dan pesan-pesan, saling terkait

satu dengan lainnya dimana dengan cara tersebut pengguna dapat menyimpan, menggali dan saling berbagi-pakai terhadap informasi yang tersedia.

Pada umumnya yang dihubungkan tersebut terdiri dari komputer mikro, terminal, printer dan media penyimpan data, serta perangkat jaringan lainnya. Dengan memiliki jaringan komputer memungkinkan anda untuk menggabungkan berbagai tingkat keahlian yang terdapat disegenap staf serta berbagai jenis kapasitas peralatan yang ada, tanpa memperhatikan soal-soal lokasi fisik diantara staf maupun peralatannya. Jaringan memungkinkan pemanfaatan secara bersama diantara para pengguna jaringan terhadap file-file data dan aplikasi, saling berkirim pesan, serta memungkinkan diterapkannya sistem pengaman terhadap instalasi secara keseluruhan.

Istilah jaringan komputer mengartikan suatu himpunan interkoneksi sejumlah komputer yang *autonomous*. Dua buah komputer dikatakan terinterkoneksi bila keduanya dapat saling bertukar informasi. Bentuk koneksinya tidak harus melalui kawat tembaga saja melainkan dapat menggunakan serat optik, gelombang mikro, atau satelit komunikasi.

Untuk memahami istilah jaringan komputer sering kali kita dibingungkan dengan sistem terdistribusi (*distributed sistem*). Kunci perbedaannya adalah bahwa sebuah sistem terdistribusi, keberadaan sejumlah komputer *autonomous* bersifat transparan bagi pemakainya. Seseorang dapat memberi perintah untuk mengeksekusi suatu program, dan kemudian program itu pun akan berjalan dan tugas untuk memilih prosesor, menemukan dan mengirimkan file ke suatu prosesor dan menyimpan hasilnya di tempat yang tepat yang merupakan tugas sistem operasi. Dengan kata lain, pengguna sistem terdistribusi tidak akan menyadari terdapatnya banyak prosesor (*multiprosesor*), alokasi tugas ke prosesor-prosesor, alokasi file ke disk, pemindahan file yang disimpan dan yang diperlukan, serta fungsi-fungsi lainnya dari sistem harus bersifat otomatis.

Pada suatu jaringan komputer, pengguna harus secara eksplisit *log* ke sebuah mesin, secara eksplisit menyampaikan tugasnya dari jauh dan memindahkan file-file dan menangani sendiri secara umum seluruh manajemen jaringan. Pada sistem terdistribusi, tidak ada yang perlu dilakukan secara eksplisit,

semuanya sudah dilakukan secara otomatis oleh sistem tanpa sepengetahuan pemakai.

Dengan demikian sebuah sistem terdistribusi adalah suatu sistem perangkat lunak yang dibuat pada bagian sebuah jaringan komputer. Perangkat lunaklah yang menentukan tingkat keterpaduan dan transparansi jaringan yang bersangkutan. Karena itu perbedaan jaringan dengan sistem terdistribusi lebih terletak pada perangkat lunaknya (khususnya sistem operasi), bukan pada perangkat kerasnya (*Odom, 2004*).

2.3. Macam Jaringan Komputer

Dalam mempelajari macam-macam jaringan komputer terdapat dua klasifikasi yang sangat penting yaitu teknologi transmisi dan jarak. Secara garis besar, terdapat dua jenis teknologi transmisi yaitu jaringan *broadcast* dan jaringan *point-to-point*

Jaringan *broadcast* memiliki saluran komunikasi tunggal yang dipakai bersama-sama oleh semua mesin yang ada pada jaringan. Pesan-pesan berukuran kecil, disebut paket, yang dikirimkan oleh suatu mesin akan diterima oleh mesin-mesin lainnya. *Field* alamat pada sebuah paket berisi keterangan tentang kepada siapa paket tersebut ditujukan. Saat menerima paket, mesin akan mengecek *field* alamat. Bila paket tersebut ditujukan untuk dirinya, maka mesin akan memproses paket itu, bila paket ditujukan untuk mesin lainnya, mesin tersebut akan mengabaikannya.

Jaringan *point-to-point* terdiri dari beberapa koneksi pasangan individu dari mesin-mesin. Untuk mengirim paket dari sumber ke suatu tujuan, sebuah paket pada jaringan jenis ini mungkin harus melalui satu atau lebih mesin-mesin perantara. Seringkali harus melalui banyak *route* yang mungkin berbeda jaraknya. Karena itu algoritma *route* memegang peranan penting pada jaringan *point-to-point*.

Pada umumnya jaringan yang lebih kecil dan terlokalisasi secara geografis cenderung memakai *broadcasting*, sedangkan jaringan yang lebih besar

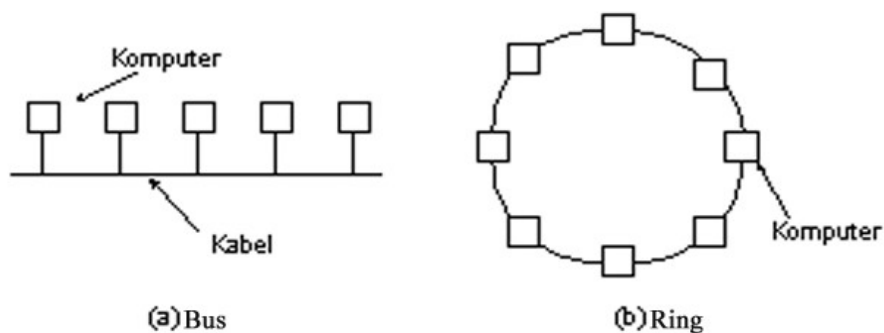
menggunakan *point-to-point*. Kriteria alternatif untuk mengklasifikasikan jaringan adalah didasarkan pada jaraknya (Odom, 2004).

2.3.1. Local Area Network

Local Area Network (LAN) merupakan jaringan milik pribadi didalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor perusahaan atau pabrik-pabrik untuk memakai bersama *resource* lainya (misalnya, printer, scanner) dan saling bertukar informasi. LAN dapat dibedakan dari jenis jaringan lainnya berdasarkan tiga karakteristik: ukuran, teknologi transmisi dan topologinya.

LAN mempunyai ukuran yang terbatas, yang berarti bahwa waktu transmisi pada keadaan terburuknya terbatas dan dapat diketahui sebelumnya. Dengan mengetahui keterbatasannya, menyebabkan adanya kemungkinan untuk menggunakan jenis desain tertentu. Hal ini juga memudahkan manajemen jaringan.

LAN seringkali menggunakan teknologi transmisi kabel tunggal. LAN tradisional beroperasi pada kecepatan mulai 10 sampai 100 Mbps (megabit/detik) dengan *delay* rendah (puluhan *mikrosecond*) dan mempunyai faktor kesalahan yang kecil. LAN - LAN modern dapat beroperasi pada kecepatan yang lebih tinggi, sampai ratusan megabit/detik.



Gambar 2.1. Dua Jenis Jaringan Broadcast

Terdapat beberapa macam topologi yang dapat digunakan pada LAN *broadcast*. Gambar 2.1 menggambarkan dua diantara topologi-topologi yang ada.

Pada jaringan *bus* (yaitu kabel liner), pada suatu saat sebuah mesin bertindak sebagai master dan diijinkan untuk mengirim paket. Mesin-mesin lainnya perlu menahan diri untuk tidak mengirimkan apapun. Maka untuk mencegah terjadinya konflik, ketika dua mesin atau lebih ingin mengirimkan secara bersamaan, maka mekanisme pengatur diperlukan. Mekanisme pengatur dapat berbentuk tersentralisasi atau terdistribusi. IEEE 802.3 yang populer disebut *Ethernet* merupakan jaringan *broadcast bus* dengan pengendali terdesentralisasi yang beroperasi pada kecepatan 10Mbps sampai dengan 100Mbps. Komputer-komputer pada *Ethernet* dapat mengirimkan kapan saja yang mereka inginkan, bila dua buah paket atau lebih bertabrakan, maka masing-masing komputer cukup menunggu dengan waktu tunggu yang acak sebelum mengulangi lagi pengirimannya.

Sistem *broadcast* yang lain adalah ring, pada topologi ini setiap bit dikirim ke daerah sekitarnya tanpa menunggu paket lengkap diterima. Biasanya setiap bit mengelilingi ring dalam waktu yang dibutuhkan untuk mengirimkan beberapa bit, bahkan seringkali sebelum paket lengkap dikirim seluruhnya. Seperti sistem broadcast lainnya, beberapa aturan harus dipenuhi untuk mengendalikan access simultan ke ring. IEEE 802.5 (token ring) merupakan LAN ring yang populer yang beroperasi pada kecepatan antara 4Mbps sampai dengan 16Mbps.

Berdasarkan alokasi *channel*nya, jaringan *broadcast* dapat dibagi menjadi dua, yaitu statik dan dinamik. Jenis alokasi statik dapat dibagi berdasarkan waktu interval - interval diskrit dan algoritma round robin, yang mengijinkan setiap mesin untuk melakukan *broadcast* hanya bila slot waktunya sudah diterima. Alokasi statik sering menyia-nyiakan kapasitas *channel* bila sebuah mesin tidak mempunyai lagi kerjaan pada saat slot alokasinya diterima. Karena itu sebagian besar sistem cenderung mengalokasi *channel*-nya secara dinamik (yaitu berdasarkan kebutuhan).

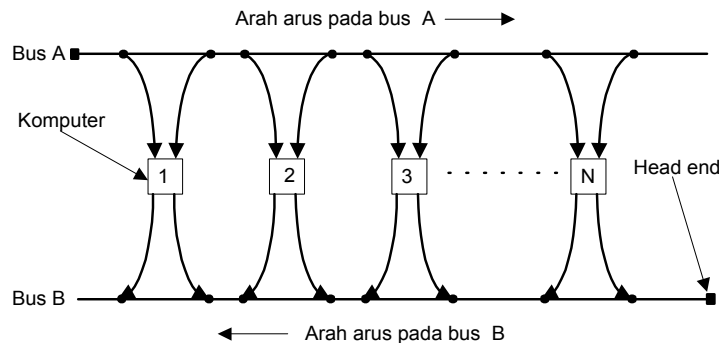
Metode alokasi dinamik bagi suatu *channel* dapat tersentralisasi ataupun terdesentralisasi. Pada metode alokasi *channel* tersentralisasi terdapat sebuah *entity* tunggal, misalnya unit *bus* pengatur, yang menentukan siapa giliran berikutnya. Pengiriman paket ini bisa dilakukan setelah menerima giliran dan membuat keputusan yang berkaitan dengan algoritma internal. Pada metode aloksi

channel terdesentralisasi, tidak terdapat *entity* sentral, setiap mesin harus dapat menentukan dirinya sendiri kapan bisa atau tidaknya mengirim.

2.3.2. Metropolitan Area Network

Metropolitan Area Network (MAN) pada dasarnya merupakan versi LAN yang berukuran lebih besar dan biasanya memakai teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang berdekatan dan dapat dimanfaatkan untuk keperluan pribadi (swasta) atau umum. MAN biasanya mampu menunjang data dan suara, dan bahkan dapat berhubungan dengan jaringan televisi kabel. MAN hanya memiliki sebuah atau dua buah kabel dan tidak mempunyai elemen switching, yang berfungsi untuk mengatur paket melalui beberapa output kabel. Adanya elemen *switching* membuat rancangan menjadi lebih sederhana.

Alasan utama memisahkan MAN sebagai kategori khusus adalah telah ditentukannya standart untuk MAN, dan standart ini sekarang sedang diimplementasikan. Standart tersebut disebut DQDB (*Distributed Queue Dual Bus*) atau 802.6 menurut standart IEEE. DQDB terdiri dari dua buah kabel *unidirectional* dimana semua komputer dihubungkan, seperti ditunjukkan pada gambar 2.2 Setiap *bus* mempunyai sebuah *head-end*, perangkat untuk memulai aktivitas transmisi. Lalulintas yang menuju komputer yang berada disebelah kanan pengirim menggunakan *bus* bagian atas. Lalulintas ke arah kiri menggunakan *bus* yang berada dibawah (*Odom, 2004*).



Gambar 2.2. Arsitektur MAN DQDB

2.4. Analisis dan Desain Sistem

2.4.1. Prinsip dan Konsep Analisis

Pemahaman yang utuh tentang kebutuhan *software* adalah dasar dari keberhasilan usaha pengembangan *software*. Minimumnya analisis dan spesifikasi program akan mengecewakan *user* dan membawa masalah bagi *developer*.

Analisis kebutuhan adalah sebuah proses penemuan, perbaikan, pemodelan, dan spesifikasi. Proses analisis berusaha menjembatani *gap* antara *sistem engineering* dan *software design*. *Sistem Engineer* sendiri mempunyai tugas untuk menspesifikasikan fungsi dan kinerja *software*, menspesifikasikan *interface software* dengan elemen lain, menspesifikasikan batasan yang akan dihadapi, dan memperbaiki alokasi *software* dan membangun model data, fungsi dan domain perilaku yang akan diperlakukan terhadap *software*. Analisis kebutuhan menyediakan *Desainer software* sebuah model yang dapat ditranslasikan kedalam bentuk data, arsitektur, *interface* dan Desain prosedur.

Analisis Kebutuhan membagi area kerjanya kedalam 5 kelompok usaha (Jogiyanto, 2001) :

1. Pengenalan Masalah
2. Evaluasi dan Sintesa
3. Pemodelan
4. *Specification*
5. *Review*

2.4.2. Konsep Desain Sistem

Desain merupakan langkah awal dalam fase pengembangan untuk proses rekayasa atau sistem. Definisi Desain menurut Taylor, E.S adalah suatu proses yang mengaplikasikan bermacam-macam teknik dan prinsip dengan tujuan untuk mendefinisikan peralatan, proses atau sistem yang cukup detail untuk membentuk realisasi fisiknya.

Desain perangkat lunak merupakan teknik utama dalam proses rekayasa perangkat lunak dan diterapkan tanpa menghiraukan model perangkat lunak yang digunakan. Desain perangkat lunak adalah langkah awal dari beberapa aktifitas

teknis yaitu desain, *code generation*, *testing*. Setiap elemen dari model analisis menyediakan informasi yang dibutuhkan untuk membuat model desain. Kebutuhan perangkat lunak, yang ditunjukkan dengan data, fungsi dan perilaku model menjadi bagian utama untuk tahap desain. Berikut adalah beberapa hasil dari metode desain yaitu :

1. Desain Data: mentransformasikan model domain informasi yang dibuat selama tahap analisa kedalam struktur data yang akan dibutuhkan dalam implementasi perangkat lunak.
2. Desain Arsitektur: mendefinisikan relasi diantara struktur utama elemen-elemen tiap program.
3. Desain Interface: mendefinisikan bagaimana proses komunikasi perangkat lunak didalamnya, perangkat lunak dengan *user*.
4. Desain Prosedur: mentransformasikan struktur elemen dari arsitektur program ke dalam definisi prosedural komponen perangkat lunak.

Desain perangkat lunak adalah sebuah proses iterasi yang terus berlangsung dimana kebutuhan-kebutuhan sistem ditranslasikan kedalam *blueprint* untuk membangun konstruksi perangkat lunak. Tanpa Desain, perangkat lunak jadi bisa tidak stabil.

2.4.3. Alat dan Teknik Dalam Pengembangan Sistem

Alat-alat yang digunakan dalam suatu metodologi pengembangan sistem umumnya berupa suatu gambar atau diagram atau grafik. Untuk metodologi terstruktur gambar/grafik yang digunakan adalah *Structured Chart*. Di samping bagan-bagan yang bersifat umum untuk semua metodologi yaitu bagan untuk aktivitas antara lain:

- Bagan Alir Sistem
- Bagan Alir Program
- Bagan Alir Kertas Kerja
- Bagan Relasi *Database*
- Bagan Alir Proses

2.5. Konsep dan Perancangan Sistem Basis Data

2.5.1. *Pengertian Basis Data dan Sistem Basis Data*

Basis data adalah sekumpulan data yang saling berelasi. Basis data merupakan komponen penting dalam teknologi informasi, dimana sebuah teknologi paling dramatis tingkat perkembangannya dan penerapannya bagi umat manusia.

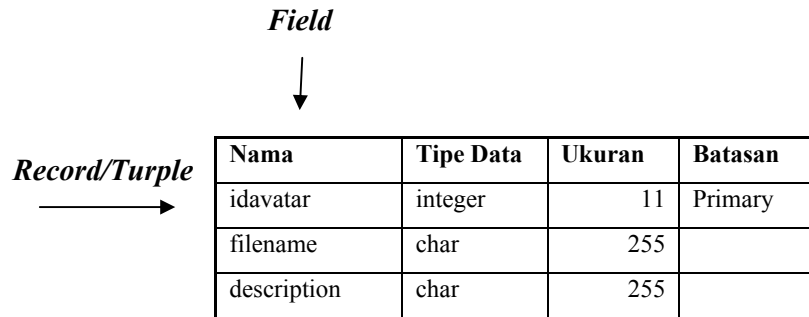
Basis data sendiri merupakan representasi digital dari kenyataan fisik dan logik dari sebuah sistem. Mentransformasikan kenyataan dari sebuah sistem ke basis data bukanlah sebuah pekerjaan sederhana. Kita membutuhkan media perantara, orang menyebutnya sebagai model data., sebagai wujud dari perancangan basis data. Setiap orang yang terlibat didalam pembangunan dan penerapan teknologi informasi harus menyadari bukan hanya pentingnya basis data dan model data, tetapi juga pada pentingnya hal membuat model data yang baik.

Basis data dibentuk sedemikian rupa sehingga memudahkan untuk diakses oleh program lain dengan mengikuti struktur bentuk dari basis data tersebut. Sebuah sistem basis data adalah sekumpulan basis data yang mempunyai keterkaitan antara yang satu dengan yang lainnya.

2.5.2. *Istilah Dalam Database*

Ada berbagai macam istilah dalam *database*, yaitu:

1. Entitas, adalah orang, tempat, kejadian, atau konsep yang informasinya disimpan. Contoh : karyawan, tim keamanan, dan lain sebagainya.
2. Data *Value* (nilai atau isi data), adalah data aktual atau informasi yang disimpan pada tiap data elemen atau atribut. Atribut nama pemasok menunjukkan tempat dimana informasi nama pemasok disimpan, sedangkan data *value* adalah Irwan, Rudi, dan lain sebagainya.
3. *Record* atau *Tuple*, kumpulan-kumpulan elemen yang menginformasikan suatu *entity* secara lengkap. Satu *record* mewakili suatu data atau informasi tentang seseorang atau sesuatu, misalnya idavatar, filename dan lain sebagainya.



Gambar 2.3. Contoh Gambar Ilustrasi *Record/Tuple*

4. *File*, kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda-beda dalam nilainya.
5. *Database*, kumpulan *file-file* yang mempunyai kaitan antara satu *file* dengan *file* yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. Bila terdapat *file* yang tidak dapat dipadukan atau dihubungkan dengan *file* yang lainnya berarti *file* tersebut bukanlah kelompok dari satu *database*, ia tidak dapat membentuk satu *database* sendiri.
6. Atribut, setiap entitas mempunyai atribut atau sebutan untuk mewakili suatu *entitas*. Contoh : nama, alamat, dan lain sebagainya.
7. *Domain* (Tipe Data) adalah kemampuan penyimpanan data yang mungkin bagi suatu atribut secara fisik, dengan melihat batas-batas nilai yang diperbolehkan bagi suatu atribut, dilihat dari kenyataan yang ada. Contohnya atribut *stock_pengaman* yang memiliki tipe data integer (-1, 0 ...100), tetapi kenyataannya *stock_pengaman* hanya memiliki nilai 3 dan 5 saja. Sehingga dapat disimpulkan bahwa *domain* untuk atribut stok adalah 3 dan 5.
8. Ketergantungan Fungsional adalah ketergantungan suatu atribut dengan atribut yang lain.

2.5.3. *Model Database*

Model data dapat didefinisikan sebagai kumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantik data dan batasan data. Ada dua cara merepresentasikan model data secara umum dalam perancangan basis data, yaitu :

1. Model Logik Data Berdasarkan Objek(*Object-Based Logical Models*) yang terdiri atas :
 - Model keterhubungan Entitas(*Entity-Relationship Model*)
 - Model Berorientasi Objek(*Object-Oriented Model*)
 - Model Data Semantik(*Semantic Data Model*)
 - Model Data Fungsional (*Functional Data Model*)
2. Model Logik Data Berdasarkan Record (*Record-Based Logical Models*) yang terdiri atas
 - Model Relasional (*Relational Model*)
 - Model Hirarkis (*Hierarchical Model*)
 - Model Jaringan (*Network Model*)

2.5.4. Abstraksi Data

Abstraksi data adalah tingkatan atau *level* dalam bagaimana melihat data dalam sebuah sistem basis data. Dalam abstraksi data terdapat tiga *level* sebagai berikut :

1. *Level Fisik(Physical Level)* adalah *level* terendah dalam abstraksi data yang menunjukkan bagaimana sesungguhnya suatu data disimpan.
2. *Level Logik atau Konseptual(Conceptual Level)* adalah *level* setelah *level* fisik yang menggambarkan data apa yang sebenarnya disimpan dalam basis data dan hubungannya dengan data yang lain.
3. *Level Penampakan(View Level)* adalah *level* tertinggi dalam abstraksi data yang hanya menunjukkan sebagian dari basis data

2.5.5. Bahasa Basis Data (Database Language)

Bahasa Basis Data adalah cara berinteraksi atau berkomunikasi antara pemakai dengan basis data yang diatur dalam suatu bahasa khusus yang terdiri atas sejumlah perintah (*statement*) yang diformulasikan dan dapat diberikan *user* dan dikenali atau diproses oleh DBMS untuk melakukan suatu aksi atau pekerjaan tertentu, contohnya *SQL*, *dBASE* dan sebagainya. Bahasa Basis Data dikelompokkan menjadi dua bentuk :

1. *Data Definition Language (DDL)* adalah struktur atau skema basis data yang menggambarkan atau mewakili desain basis data secara keseluruhan dispesifikasi dengan bahasa khusus. Hasil dari kompilasi *DDL* adalah kumpulan tabel yang disimpan dalam *file* khusus yang disebut kamus data. Kamus Data adalah suatu metadata atau superdata yaitu data yang mendiskripsikan data sesungguhnya.
2. *Data Manipulation Language (DML)* adalah bentuk bahasa basis data yang berfungsi untuk melakukan manipulasi dan pengambilan data pada suatu basis data. Contoh manipulasi data :
 - Penyisipan atau penambahan data baru ke suatu basis data.
 - Penghapusan data dari suatu basis data.
 - Pengubahan data di suatu basis data.

DML terbagi menjadi dua jenis :

- 1 Prosedural : mensyaratkan agar pemakai menentukan data apa yang diinginkan serta bagaimana cara mendapatkannya.
- 2 Nonprosedural : membuat pemakai dapat menentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya.

2.5.6. Kunci (Key)

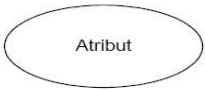
Setiap *file* selalu terdapat kunci dari *file* berupa satu *field* atau satu set *field* yang dapat mewakili *record*. Misalnya nomor karyawan merupakan kunci dari tabel karyawan suatu perusahaan, setiap pencarian cukup dengan menyebut nomor karyawan tersebut. Maka dapat diketahui nama, alamat, dan atribut lainnya mengenai karyawan tersebut.

- *Primary Key*, adalah satu atribut atau satu set minimal atribut yang tidak hanya mengidentifikasi secara unik satu kejadian spesifik, tapi juga dapat mewakili setiap kejadian dari suatu *entity*. Atau dengan kata lain setiap *record* mempunyai satu atribut yang unik yang bersesuaian hanya dengan *record* tertentu saja. Kunci kandidat mempunyai peluang menjadi kunci primer, tapi sebaiknya dipilih satu saja yang dapat mewakili secara menyeluruh *entity* yang ada.

- *Foreign Key*, adalah satu atribut (satu set atribut) yang melengkapi satu relationship yang menunjukkan ke induknya. Kunci tamu ditempatkan pada *entity* anak dan sama dengan kunci primary induk yang direalisasikan.

2.5.7. Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah representasi grafik dari suatu data yang terdapat dalam suatu sistem atau organisasi. ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Dengan ERD, model dapat diuji dengan mengabaikan proses yang dilakukan. ERD mempunyai tiga komponen utama, yaitu entitas, relasi, dan atribut.

Notasi	Keterangan
	Entitas , adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi , menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut , berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah)
	Garis , sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

Gambar 2.4. Notasi ERD (*Analisis Entitas*, 2008)

Dalam ERD, relasi antara 2 himpunan dibagi menjadi 3 golongan, yaitu:

1. One to one / 1-1

Setiap entitas pada himpunan entitas a dapat berelasi dengan paling banyak satu pada himpunan entitas b, berlaku juga sebaliknya.

2. One to many / 1-N

Setiap entitas pada himpunan entitas a dapat berelasi banyak pada himpunan entitas b, tetapi tidak berlaku sebaliknya.

3. Many to many / N-N

Setiap entitas pada himpunan entitas a dapat berelasi banyak pada himpunan entitas b, demikian juga sebaliknya.

2.6. *Flowchart System*

2.6.1. *Definisi Flowchart*

Flowchart adalah representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan simbol, dimana masing-masing simbol merepresentasikan suatu kegiatan tertentu. *Flowchart* diawali dengan penerimaan *input*, pemrosesan *input*, dan diakhiri dengan penampilan *output* (Romney, 2003).

Menurut Romney (2003), ada 2 macam *flowchart* yang menggambarkan proses dengan komputer, yaitu :

1. System Flowchart

System Flowchart adalah suatu gambar yang memperlihatkan urutan prosedur dan proses dari beberapa *file* di dalam media tertentu. Melalui *flowchart* ini, dapat terlihat jenis media penyimpanan yang dipakai dalam pengolahan data. Selain itu juga menggambarkan *file* yang dipakai sebagai *input* maupun *output*.

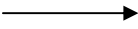
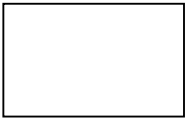
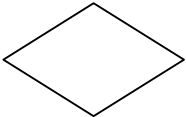
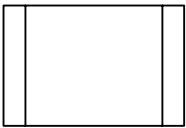
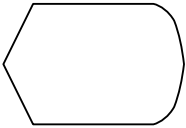
2. Program Flowchart

Program Flowchart adalah bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program. *Program flowchart* ini merupakan langkah awal pembuatan program. Dengan adanya *program flowchart* maka urutan proses di program menjadi lebih jelas.

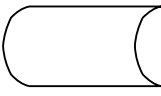
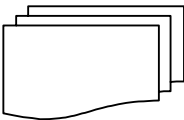
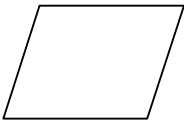
2.6.2. *Flowchart Symbols*

Flowchart disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang digunakan dapat dilihat pada Tabel 2.1.

Tabel 2.1. *Flowchart symbols*

Simbol	Nama Simbol	Keterangan
	<i>Document / Processing Flow</i>	Simbol untuk menunjukkan arah aliran dokumen atau proses.
	<i>Connector</i>	Simbol untuk keluar atau masuk prosedur atau proses pada halaman yang sama.
	<i>Off Line Connector</i>	Simbol untuk keluar atau masuk prosedur atau proses pada halaman yang lain .
	<i>Process</i>	Simbol yang menunjukkan pengolahan yang dilakukan dengan komputer.
	<i>Manual Operation</i>	Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
	<i>Decision</i>	Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban.
	<i>Predefined Process</i>	Simbol untuk menunjukkan suatu proses dalam proses.
	<i>Terminal</i>	Simbol untuk menunjukkan awal, akhir, dan interupsi dalam sistem.
	<i>Display</i>	Simbol yang menyatakan peralatan <i>output</i> yang digunakan yaitu <i>layer</i> , <i>plotter</i> , <i>printer</i> , dan sebagainya.

Tabel 2.2. *Flowchart symbols* (Lanjutan)

	<i>Manual Input</i>	Simbol yang menyatakan <i>input</i> dengan menggunakan mesin yang mempunyai <i>keyboard</i> .
	<i>Stored Data</i>	Simbol yang menggambarkan <i>input</i> yang berasal dari <i>disk</i> atau <i>output</i> yang disimpan pada <i>disk</i> .
	<i>Document</i>	Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas.
	<i>Multiple Document</i>	Simbol yang menyatakan dokumen yang sama dicetak beberapa kali untuk kepentingan tertentu.
	<i>Input/Output</i>	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.

(Sumber : Romney, *Accounting Information Systems* 9/E, p. 168)

2.7. Pemrograman Borland Delphi

Delphi adalah perangkat lunak untuk menyusun program aplikasi yang berdasarkan pada bahasa pemrograman pascal dan bekerja dalam lingkungan sistem operasi windows. Didalam Delphi tidak perlu lagi repot membuat window, kotak dialog maupun perangkat control lainnya seperti tombol perintah (*command button*), menu *pull down*, kotak combo (*combo box*) dan sebagainya.

Kelebihan Delphi adalah dalam hal kecepatan proses kompilasi program. Dapat disebutkan pula sebagai keunggulan Delphi yaitu ukuran file ***.exe** (exe = *execution*, yaitu file yang mampu berjalan sendiri di luar *software* pembangunnya) yang dihasilkan relative sangat kecil sehingga tidak menggunakan ruang disk

yang besar. selain itu, file – file *.dll (dll = *dynamic link library*, yaitu file program bantu pada file execution) yang dihasilkannya mampu berdiri sendiri tanpa program utamanya.

Kegunaan Delphi:

1. Untuk membuat aplikasi windows
2. Untuk merancang aplikasi program berbasis grafis
3. Untuk membuat program berbasis jaringan (*client / server*)
4. Untuk merancang program .Net (berbasis internet)

Keunggulan Delphi:

1. IDE (*Integrated Development Environment*) atau lingkungan pengembangan aplikasi sendiri adalah satu dari beberapa keunggulan delphi, didalamnya terdapat menu – menu yang memudahkan kita untuk membuat suatu proyek program.
2. Proses Kompilasi cepat, pada saat aplikasi yang kita buat dijalankan pada Delphi, maka secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah.
3. Mudah digunakan, source kode delphi yang merupakan turunan dari pascal, sehingga tidak diperlukan suatu penyesuaian lagi.
4. Bersifat *multi purphase*, artinya bahasa pemrograman Delphi dapat digunakan untuk mengembangkan berbagai keperluan pengembangan aplikasi.

2.8. Protokol Transport (TCP/UDP)

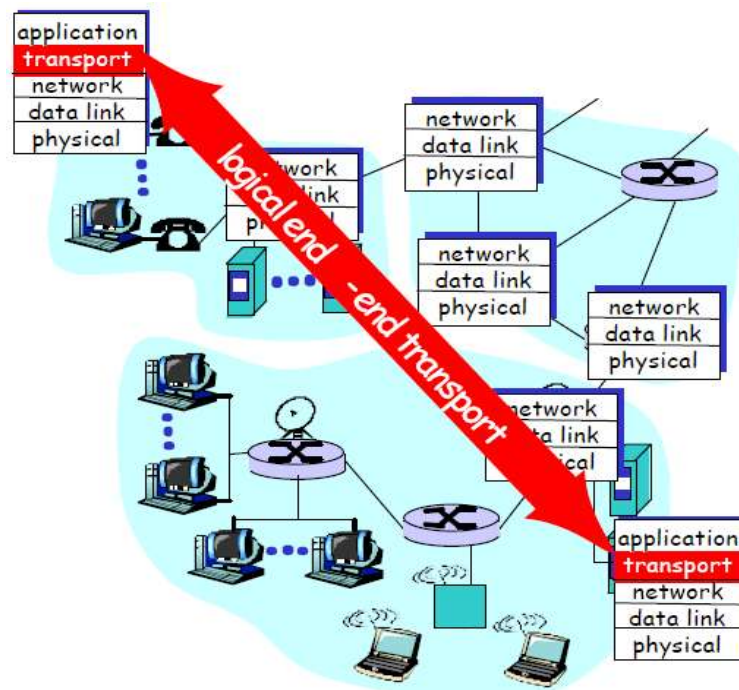
Komunikasi logika pada lapisan Transport dapat berbentuk :

- *connectionless* atau *connection-oriented*.
- *Reliable* atau *unreliable* : *Reliable* berarti data ditransfer ke tujuannya dalam suatu urutan seperti ketika dikirim. Pengiriman data *Unreliable* sangat menggantungkan diri pada lapisan jaringan di bawahnya, sehingga tidak dapat menyakinkan apakah segment data dapat dikirimkan sampai ditujuannya atau tidak.
- *Stateful* atau *stateless*. Pengiriman data *stateful* berarti informasi yang

dimasukkan pada satu *request*, yang dikirimkan dari pengirim ke penerima, dapat dimodifikasi untuk *request* berikutnya. Sedangkan pengiriman *stateless* berarti informasi dalam satu *request* tidak dapat dikaitkan dengan *request* lainnya, sehingga tidak dapat digunakan untuk *request* lainnya.

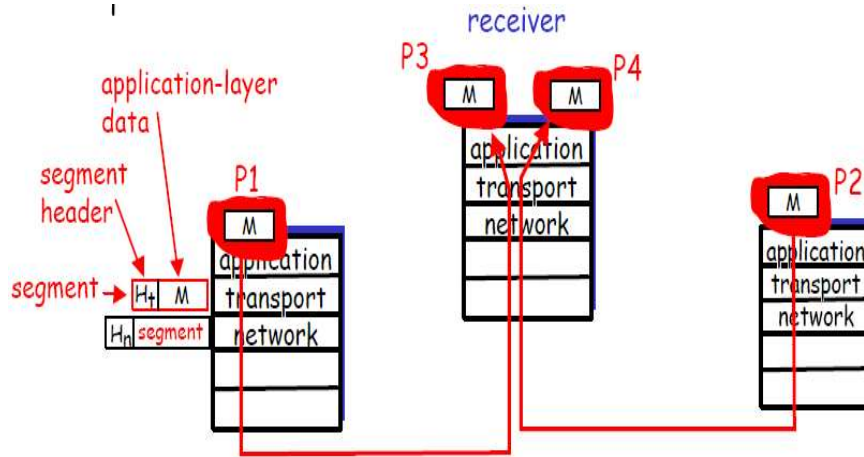
Fungsinya dari lapisan *transport* adalah:

- menyediakan komunikasi logika antar proses aplikasi yang berjalan pada *host* yang berbeda protokol transport berjalan pada *end systems*.
- Perbedaan dengan Lapisan Network :
lapisan *network* : transfer data antar *end-system*.
lapisan *transport* : transfer data antar proses.
- Layanan transport pada Internet :
reliable, pengiriman dalam suatu urutan dengan model *unicast*. Contoh : TCP.
unreliable, pengiriman tidak dalam suatu urutan dengan model *unicast* atau *multicast*. Contoh : UDP.
- Pada layanan *transport*, satuan data yang dipertukarkan disebut sebagai *segment* (TPDU = Transport Protocol Data Unit).



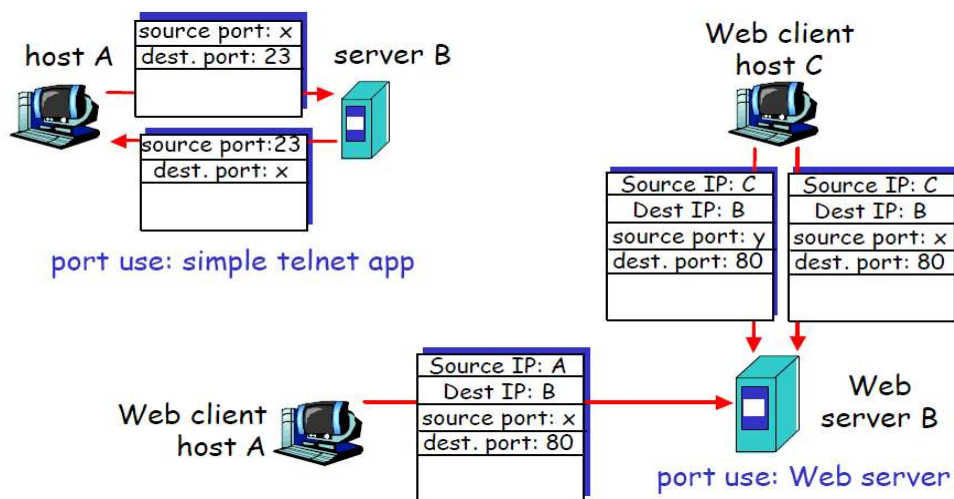
Gambar 2.5. Lapisan Transport

- Layanan transport menyediakan demultiplexing untuk dapat mengirimkan segment ke proses lapisan aplikasi yang sesuai berdasar alamat dan port proses tersebut.



Gambar 2.6. Pengiriman Segmen

- Selain itu juga melakukan multiplexing, yang akan mengambil data dari beberapa proses aplikasi, dan membungkus data dengan header.
- Multiplexing dan Demultiplexing di dasarkan pada pengirim dan nomor port serta alamat IP penerima. Terdapat nomor port sumber dan tujuan pada tiap segment. (Susanto, page 1-2).



Gambar 2.7. Port Source-Destination