

BAB II

TEORI PENUNJANG

Sebelum membahas istilah basis data, perlu untuk membahas istilah data dan informasi terlebih dahulu. Karena kedua istilah ini sering dipertukarkan atau dianggap sama.

Data adalah bentuk jamak dari datum. Data merupakan kenyataan/fakta dari suatu objek, dimana objek itu bisa berupa orang, benda, kejadian atau yang lain. Data merupakan bentuk masih mentah yang belum bisa bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah dalam suatu model sehingga akan dihasilkan informasi. Informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya [Jogiyanto, 1998:8]. Dalam era informasi, informasi menjadi sumber penting untuk melakukan pengambilan keputusan. Sehingga dengan informasi dapat mengurangi ketidakpastian dan mempermudah pengambilan keputusan.

2.1. Basis Data

Basis data adalah kumpulan file-file berelasi, yang biasanya relasi tersebut ditunjukkan dengan kunci dari tiap file yang ada. Definisi lain, basis data adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data [Fabbri & Schwab:1992]. *File* adalah kumpulan record-record sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda-beda data valuenya. *Record* adalah kumpulan elemen-elemen yang saling berkaitan

yang menginformasikan tentang suatu *entity* secara lengkap. **Entity** adalah orang, tempat, kejadian, atau konsep yang informasinya direkam. Setiap *entity* mempunyai **atribut/sebutan** untuk mewakilinya, yang biasa disebut dengan **field**. Lebih jelasnya seperti terlihat pada tabel 2.1

Tabel 2.1. Tabel Identitas Mata Kuliah

The diagram shows a table with five columns and two rows. The columns are labeled 'Kd_MK', 'Mata_Kul', 'SKS', 'Semest', and 'Kel'. The rows contain data for two courses. The word 'Field' is written above the table with arrows pointing to each of the five columns. The word 'Record' is written to the right of the table with an arrow pointing to the second row.

Kd_MK	Mata_Kul	SKS	Semest	Kel
MAT103	Kalkulus I	4	1	MKK
SBE121	Bahasa Inggris	2	1	MKU

2.1.1. Kegunaan Basis Data

Menurut Korth [1986] penyusunan basis data digunakan untuk mengatasi masalah-masalah, yaitu:

◆ Redudansi dan inkonsistensi data

Jika file-file dalam program aplikasi diciptakan dengan konsep yang berbeda-beda, maka ada beberapa bagian data yang mengalami penggandaan. Ketidakkonsistenan data diakibatkan oleh adanya perubahan terhadap data yang sama, tetapi tidak semuanya diubah. Penggandaan atau kerangkapan data dapat menyebabkan timbulnya masalah dalam pengaksesan data, dan menyebabkan data tidak konsisten.

◆ **Kesulitan pengaksesan data**

Pada sistem basis data terdapat data-data fleksibel yang ditujukan untuk memberikan kemudahan dalam menampilkan kembali data-data yang dipilih dan diperlukan dalam basis data dan merepresentasikan data-data tersebut dalam format yang berbeda.

◆ **Banyak pemakai (*multiple user*)**

Sistem basis data yang dibuat ditujukan untuk dimanfaatkan oleh banyak pemakai. Baik untuk memperbaharui data ataupun sekedar memanfaatkan informasi yang terdapat di dalamnya.

◆ **Masalah Keamanan (*security*)**

Keamanan data diperlukan untuk melindungi data terhadap akses yang tidak berhak dari pihak-pihak yang tidak berwenang yang bermaksud merugikan atau bahkan merusak data.

◆ **Masalah kesatuan (*integrasi*)**

Integritas berarti bahwa data selalu dalam keadaan valid. Sistem basis data berisi file-file yang saling berkaitan. Pemrogram basis data memungkinkan hubungan antar file maupun antar data dapat berlangsung dengan baik melalui field kunci yang mengaitkan file-file tersebut.

◆ **Masalah kebebasan data (*data independence*)**

Jika data dikelola dengan baik sebagai sumber daya yang independen, maka pemisahan file-file data dari program-program aplikasi merupakan hal yang sangat penting

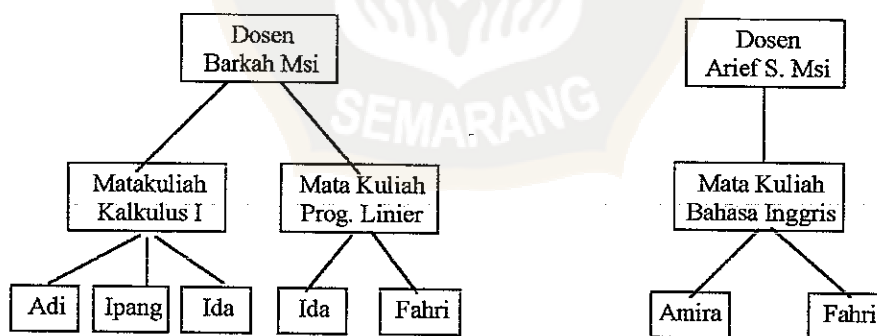
2.1.2. Model Basis Data

Model basis data menyatakan hubungan antar rekaman yang tersimpan dalam basis data. Model dasar yang paling umum ada 3 macam, yaitu:

2.1.2.1. Model hirarkis

Model ini menggunakan pola seperti pohon, dengan pola hubungan orangtua-anak. Pada model ini ada istilah seperti: **simpul** yang menyatakan sekumpulan field, **orangtua** yang menyatakan simpul yang terhubung ke simpul di bawahnya, dan **anak** yang merupakan simpul di bawah simpul orangtua. Setiap orangtua bisa memiliki hubungan lebih dari beberapa anak, tetapi setiap anak *hanya* boleh memiliki satu orangtua.

Diagram yang menunjukkan Model hirarkis ditunjukkan oleh gambar 2.1



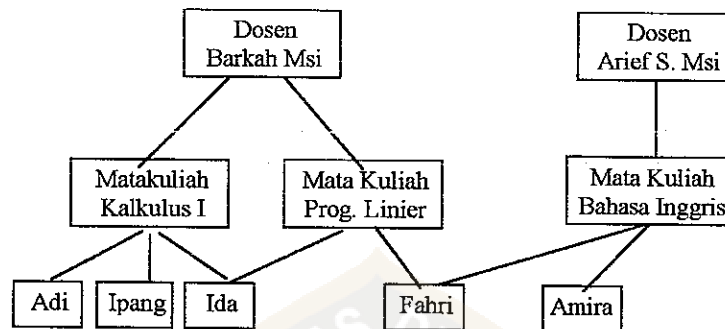
Gambar 2.1. Diagram Model Hirarkis

2.1.2.2. Model Jaringan

Model ini menyerupai model hierarkis, dengan perbedaan suatu simpul anak bisa memiliki lebih dari satu orang tua, sehingga model ini bisa menyatakan

hubungan 1:1 (satu orangtua ke satu anak), 1: M (satu orangtua ke banyak anak), maupun M:M (beberapa anak bisa memiliki beberapa orang tua).

Diagram yang menunjukkan Model Jaringan ditunjukkan oleh gambar 2.2.

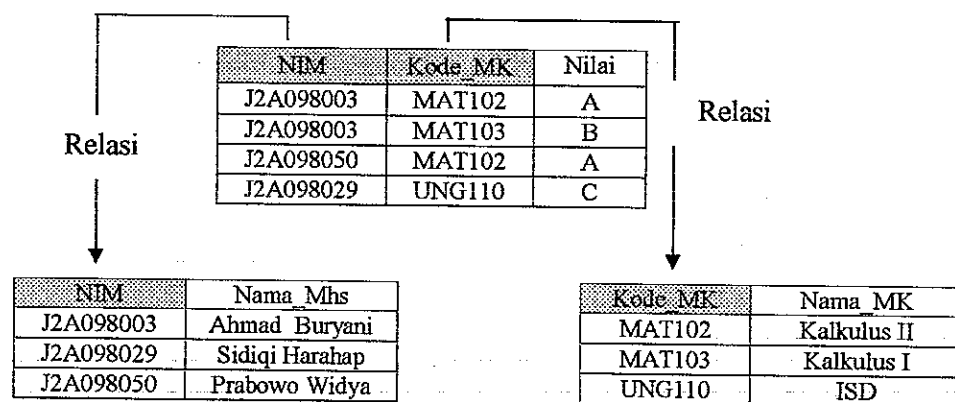


Gambar 2.2. Diagram Model Jaringan

2.1.2.3. Model Relasional

Model ini menggunakan sekumpulan tabel (atau bisa disebut relasi), dengan masing-masing relasi tersusun atas tupel dan field. Relasi dirancang sedemikian rupa sehingga data dapat efektif (tidak ada kemubaziran data) dan menggunakan kunci tamu untuk berhubungan dengan relasi yang lain.

Diagram yang menunjukkan Model Relasional ditunjukkan oleh gambar 2.3.



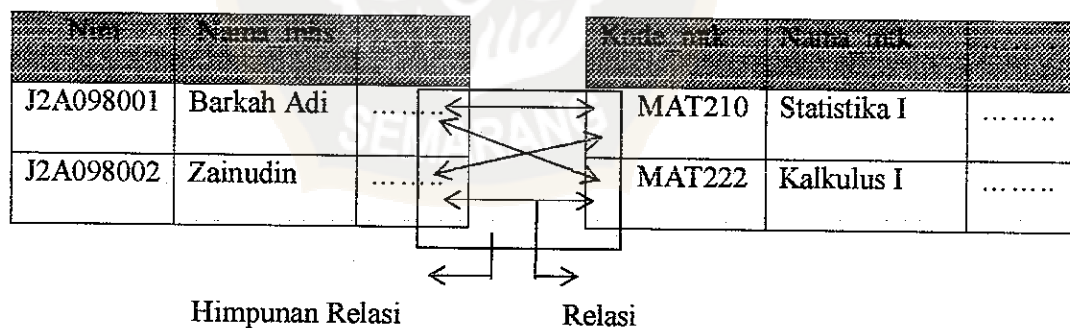
Gambar 2.3. Diagram Model Relasional

Untuk selanjutnya, dalam penyusunan tugas akhir ini menggunakan Model Relasional dalam penyusunan *databasenya* karena merupakan model yang paling sederhana sehingga mudah digunakan dan dipahami.

2.1.3. Relasi dan Himpunan Relasi

Ditulis oleh Korth (1986), Relasi adalah hubungan dari beberapa entitas. Himpunan relasi adalah kumpulan relasi dengan tipe yang sama. Secara matematis Himpunan Relasi adalah 2 (dua) atau lebih relasi dalam himpunan-himpunan entitas. Jika E_1, E_2, \dots, E_n adalah himpunan entitas, maka himpunan relasi R adalah himpunan bagian dari $\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$ $n = 1, 2, \dots, n$; dimana (e_1, e_2, \dots, e_n) adalah relasi.

Contoh relasi dan himpunan relasi ditunjukkan pada Gambar 2.4:



Gambar 2.4. Relasi dan Himpunan Relasi

2.1.4 Key dan Atribut Deskriptif

Entitas memiliki atribut yang mendeskripsikan karakteristik (*properti*) dari entitas tersebut. Dalam model E-R, kedudukan atribut dalam entitas harus dapat membedakan atribut yang berfungsi sebagai *key* primer (*primary key*) dan yang berfungsi sebagai atribut deskriptif.

Key adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*row*) dalam tabel secara unik. Artinya, jika suatu atribut dijadikan sebagai *key*, maka tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut tersebut. (Fathansyah, 1999:39)

Ada 3 macam *key* yang dapat diterapkan pada suatu tabel :

- ◆ *Superkey*, merupakan satu atau lebih atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam sebuah tabel secara unik. Bisa terjadi, ada lebih dari 1 kumpulan atribut yang bersifat seperti itu pada sebuah tabel.
- ◆ *Candidate-key*, merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.
- ◆ *Primary-key (key primer)*, adalah *key* yang dipilih dari sejumlah *candidate-key*, yang dipilih berdasarkan:
 - ♣ *Key* tersebut lebih sering (lebih umum) dipakai sebagai acuan
 - ♣ *Key* tersebut lebih ringkas
 - ♣ Jaminan keunikan *key* tersebut lebih baik

Atribut deskriptif adalah atribut-atribut yang bukan merupakan anggota *Key Primer*.

Contoh :

Pada himpunan entitas Dosen dengan atribut (NIP, Nama_dosen, Alamat_dosen, Tgl_lahir, Gol_darah) dapat dianalisa sebagai berikut :

- ◆ *Superkey* yang mungkin dari tabel tersebut adalah :
 - (NIP, Nama_dosen, Alamat_dosen, Tgl_lahir, Gol_darah)
 - (NIP, Nama_dosen, Alamat_dosen, Tgl_lahir)
 - (NIP, Nama_dosen, Alamat_dosen)
 - (NIP, Nama_dosen)
 - (Nama_dosen) jika dapat dijamin tidak ada nilai yang sama untuk atribut ini
 - (NIP)
- ◆ *Candidate Key* yang mungkin adalah :
 - (Nama_dosen), jika dapat dijamin tidak ada nilai yang sama untuk atribut ini
 - (NIP)
- ◆ Atribut *NIP* merupakan *primary key* untuk Himpunan Entitas Dosen, karena *NIP* merupakan Entitas yang paling unik dalam Himpunan Entitas tersebut. Atribut-atribut yang lain (*Nama_dosen*, *Alamat*, *Tgl_lahir*, *Gol_darah*) merupakan atribut deskriptif.

2.1.5. Ketergantungan Fungsional

Ketergantungan Fungsional mendefinisikan relasi antara sebuah/beberapa atribut pada sebuah tabel ke dalam atribut atau beberapa atribut dari tabel yang

lain. Misalkan pada Data Mahasiswa, setiap mahasiswa memiliki NIM yang berbeda-beda. Tidak ada NIM kembar walaupun kemungkinan nama sama.

Fathansyah (1999) memberikan definisi Ketergantungan Fungsional (KF) sebagai berikut :

Diberikan 2 row r_1 dan r_2 dalam tabel T yang memiliki atribut A dan B dengan $A \rightarrow B$, jika $r_1(A) = r_2(A)$ maka $r_1(B) = r_2(B)$.

Notasi $A \rightarrow B$ mempunyai arti A secara fungsional menentukan B , atau B secara fungsional tergantung pada A , jika hanya jika untuk setiap kumpulan baris data (row) yang ada dalam tabel T pasti ada dua row dalam tabel T dengan nilai A yang sama maka nilai untuk B pasti juga sama. (Fathansyah, 1999:44)

Contoh Ketergantungan Fungsional:

Diberikan tabel nilai mahasiswa Matematika (tabel 2.2) dengan atribut *Mata kul*, *Nim*, *Nama mhs*, dan *Indeks nilai*.

Tabel 2.2. Tabel Nilai Mahasiswa Matematika

	Mata_kul	Nim	Nama_mhs	indeks_nilai
Row 1	Kalkulus I	J2A98011	Aswin Sani	B
Row 2	Kalkulus I	J2A98004	Prabowo S	C
Row 3	Basis data	J2A980011	Aswin Sani
Row 4	Basis data	J2A98022	Rudiyanto
Row 5	Kalkulus II	J2A98022	Rudiyanto	A

Dari data di atas, dengan asumsi bahwa data tersebut memadai dan dengan pertimbangan intuisi kita, maka KF yang dapat diajukan adalah :

♣ $nim \rightarrow nama_mhs$

yang berarti bahwa atribut `nama_mhs` hanya tergantung pada atribut `nim`. Hal ini dapat dilihat bahwa untuk setiap nilai `nim` yang sama maka pasti nilai `nama_mhs`-nya juga sama.

♣ $nama_kul \quad nim \rightarrow indeks_nilai$

yang berarti bahwa atribut `indeks_nilai` tergantung pada atribut `nama_kul` dan `nim` secara bersama-sama. KF ini menunjukkan pengertian bahwa setiap indeks nilai diperuntukkan bagi mahasiswa tertentu untuk mata kuliah tertentu yang diambilnya.

2.1.6. Normalisasi

Setelah mengenal pengertian *key*, akan dibahas mengenai normalisasi.

Normalisasi adalah proses konversi dokumen / laporan manual ke dalam struktur tabel dengan menghilangkan elemen yang sama dan data yang berulang-ulang. (Leman, 1999:87)

Kristanto (1999) memberikan definisi normalisasi sebagai suatu proses pengelompokan data menjadi tabel-tabel yang menunjukkan entity dan relasinya.

Tujuan desain adalah mengkonstruksi relasi tanpa redundansi. Kondisi ini didefinisikan dalam terminologi relasi normal (normal relations). (Pohan, 1997:49)

Dalam perspektif normalisasi, sebuah basis data dapat dikatakan baik jika setiap tabel yang menjadi unsur pembentuk basis data tersebut juga telah berada dalam keadaan baik (efisien) atau normal. Adapun sebuah tabel dapat dikategorikan baik (efisien) atau normal jika telah memenuhi tiga kriteria berikut :

1. Jika ada dekomposisi (penguraian) tabel maka dekomposisinya dijamin aman (*Lossless-Join Decomposition*)
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*)
3. Tidak melanggar Boyce-Code Normal Form (BCNF). Atau jika BCNF tidak terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap Ketiga (*3rd Normal Form / 3NF*)

(Fathansyah,1999:46)

Kriteria di atas adalah kriteria utama untuk mendapatkan predikat efisien / normal bagi sebuah tabel. Selain kriteria tersebut, terdapat kriteria-kriteria lain yang juga tercakup dalam kerangka normalisasi, tetapi bukan merupakan kriteria utama, yaitu :

- Bentuk Normal tahap Pertama (*1st Normal Form / 1NF*)

Field yang diberikan dalam record yang diberikan hanya dapat berisi satu nilai. Tidak diperkenankan pengulangan grup dalam record tunggal.

- Bentuk Normal tahap Kedua (*2nd Normal Form / 2NF*)

Setiap Field nonkunci harus bergantung pada semua field dalam kunci primer. Bentuk ini juga menyebutkan bahwa dua tabel atau lebih tidak

bisa memiliki kunci primer yang sama. Jika disain kita memiliki dua tabel dengan kunci primer yang sama, Anda harus menggabungkan tabel-tabel tersebut menjadi satu.

Bentuk Normal tahap Ketiga (3th *Normal Form* / 3NF)

Tidak satupun field non kunci bergantung pada field nonkunci lainnya.

Bentuk Normal tahap Keempat (4th *Normal Form* / 4NF)

Bentuk ini melarang berbagai hubungan yang berdiri sendiri, satu dengan lainnya, diantara field kunci primer dan field-field nonkunci.

Bentuk Normal tahap Kelima (5th *Normal Form* / 5NF)

Bentuk ini sangat ekstrim dan seringkali diabaikan. Bentuk ini butuh pemecahan tabel ke dalam bagian yang lebih kecil untuk mengurangi segala kekosongan.

Selanjutnya akan diterangkan sekilas.

Bentuk Normal Pertama (1NF)

Suatu relasi dikatakan dalam bentuk normal pertama jika dan hanya jika setiap atribut bernilai tunggal untuk setiap baris. Seperti terlihat pada tabel 2.3.

Tabel 2.3. Biodata Mahasiswa Jurusan Matematika

NIM	Nama	Kota Lahir	SKS	KodeMK	Mata Kuliah
J2A098029	M. Irfan Affandi	Bojonegoro	4	MAT001	Riil I
J2A098029	M. Irfan Affandi	Bojonegoro	3	MAT002	Kalkulus II
J2A098029	M. Irfan Affandi	Bojonegoro	2	UNG002	Filsafat
J2A098033	M. Mahmudi	Solo	3	MAT002	Kalkulus II
J2A098033	M. Mahmudi	Solo	4	MAT201	Diskrit
J2A098034	Nuryono Adi	Salatiga	2	UNG003	ISD
J2A098034	Nuryono Adi	Salatiga	4	MAT001	Riil I

Tabel 2.3 di atas sudah dalam bentuk 1NF. Tapi data tersebut akan menyebabkan anomali update, yang berarti kesulitan dalam operasi perubahan data yang meliputi INSERT, DELETE dan UPDATE.

Bentuk Normal Kedua (2NF)

Bentuk normal kedua adalah tabel 1 NF dan setiap field yang bukan kunci bergantung sepenuhnya dengan kunci. Dari tabel 2.3 di atas dapat dipecah menjadi 3 tabel yaitu tabel 2.4, tabel 2.5 dan tabel 2.6 dengan aturan sebagai berikut:

Tabel 2.4. Mata Kuliah

KodeMK	Mata Kuliah
MAT001	Riil I
MAT002	Kalkulus II
UNG002	Filsafat
MAT201	Diskrit
UNG003	ISD

Tabel 2.5. Jumlah SKS

Kode MK	SKS
MAT001	4
MAT002	3
UNG002	2
MAT002	3
MAT201	4
UNG003	2
MAT001	4

Tabel 2.6 Mahasiswa

NIM	Nama	Kota Lahir
J2A098029	M. Irfan Affandi	Bojonegoro
J2A098033	M. Mahmudi	Solo
J2A098034	Nuryono Adi	Salatiga

Bentuk Normal Ketiga (3NF)

Bentuk normal ketiga adalah tabel yang sudah 2NF dan tidak ada non kunci yang tergantung pada elemen bukan kunci.

2.2. Sistem Informasi

Sistem terdiri dari komponen-komponen yang saling berkaitan dan bekerja sama untuk mencapai suatu tujuan. Informasi merupakan proses lebih lanjut dari data dan memiliki nilai tambah, yang memenuhi syarat-syarat : lengkap, akurat,

relevan, dan tepat waktu. Dengan demikian, sistem informasi dapat didefinisikan sebagai suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. (Leman, 1998:3)

Menurut Leman (1998), komponen sistem informasi terdiri dari :

- **Hardware**, terdiri dari komputer, dan perifer (printer)
- **Software**, merupakan kumpulan dari perintah / fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu
- **Data**, merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi.
- **Manusia**, yang terlibat dalam komponen manusia, seperti operator
- **Prosedur**, seperti dokumentasi prosedur / proses sistem, buku penuntun operasional (aplikasi) dan teknis

2.2.1. Pemodelan Sistem

Hal yang dominan ketika merancang sistem adalah memodelkan kebutuhan pemakai. Pembuatan model dilakukan agar jika model yang dibuat salah maka perancang dapat kembali membuat model yang lebih memenuhi kebutuhan pemakai tanpa resiko berarti dibandingkan jika perancang membuat sistem secara langsung. Dalam sebuah sistem, model merupakan kombinasi antara perangkat keras dengan perangkat lunak.

Pohan (1997) memberikan alasan penyebab sebaiknya dilakukan pemodelan sistem, yaitu :

- Dapat memfokuskan perhatian pada hal-hal penting dalam sistem tanpa mesti terlibat terlalu jauh.
- Mendiskusikan perubahan dan koreksi terhadap kebutuhan pemakai dengan resiko dan biaya minimal
- Menguji pengertian penganalisa sistem terhadap kebutuhan pemakai dan membantu pendesain sistem dan pemrogram membangun sistem

2.2.2. Perancangan Sistem

Menurut Bahri (1997), aktifitas perancangan sistem secara terstruktur melingkupi:

- Survey; berfungsi untuk mengetahui kebutuhan-kebutuhan pemakai, kesalahan-kesalahan dalam sistem lama, menetapkan tujuan perancangan dan menyiapkan laporan hasil survey.
- Analisa Sistem; menggabungkan laporan survey dan kebijakan pemakai menjadi spesifikasi yang terstruktur dengan menggunakan permodelan.
- Desain; mengimplementasikan model yang diinginkan pemakai.
- Implementasi; merepresentasikan hasil desain ke dalam pemrograman.
- Uji coba desain; menguji coba seluruh spesifikasi terstruktur.
- Testing akhir; menguji coba sistem secara keseluruhan.
- Deskripsi prosedur; pembuatan laporan teknis tertulis seperti petunjuk pemakaian dan pengoperasian.

- Konversi database; mengkonversi data.
- Instalasi, aspek terakhir yang semestinya dilakukan mencakup serah terima manual, perangkat keras dan pelatihan pemakaian.

2.3 *Web Database*

Web database merupakan sistem penyimpanan data yang dapat diakses oleh bahasa pemrograman tertentu. Namun sistem basis data ini tidaklah seperti sistem database yang selama ini dikenal, *web database* dapat diakses oleh aplikasi web yang tentunya lebih umum dan simpel meski tetap menggunakan aplikasi database konvensional. *Web database* dapat diakses oleh aplikasi-aplikasi web yang dikembangkan dengan *HTML tag*, pemrograman server-side melalui CGI, Microsoft Internet Information Server, atau skrip seperti ASP yang akan digunakan nanti.

Kemampuan inilah yang menjadikan suatu database menjadi *web database*. *Web database* dapat menggantikan sistem informasi umum seperti pemesanan kamar hotel, web pariwisata dengan banyak pilihan tujuan wisata, sampai ke sistem informasi akademik seperti pengisian form pendaftaran mahasiswa baru. Hampir semua web yang ada sekarang mulai menjadi *web database*, karena bisa untuk mengembangkan fasilitas yang dibutuhkan oleh pengunjung. Dari situ berkembang ide untuk ke aplikasi lain diantaranya dalam dunia pendidikan sehingga muncul istilah *e-Education* yang bisa mencakup di dalamnya adalah *e-Book*, *e-News*, *e-Dictionary*, *e-Laboratory* dan lainnya.

2.4. Internet

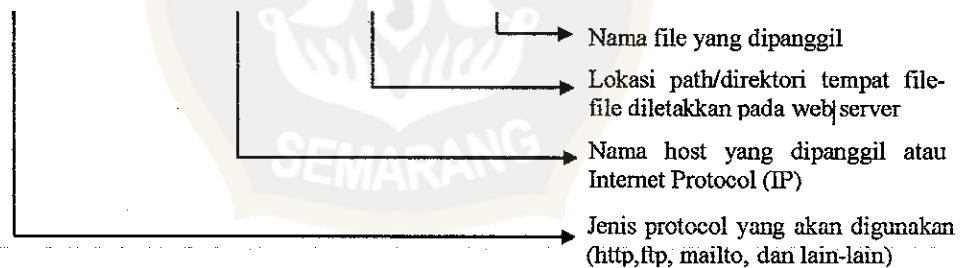
Internet adalah metode untuk menghubungkan berbagai komputer ke dalam satu jaringan komputer global, melalui protokol yang disebut TCP/IP (*Transmission Control Protocol/Internet Protocol*).

2.4.1. Uniform Resource Locator (URL)

Salah satu subyek Internet yang paling awal harus dikenal adalah URL (*Uniform Resource Locator*). URL dapat didefinisikan sebagai sarana untuk menentukan alamat yang akan dipakai untuk mengakses Internet, khususnya *website*.

Secara garis besar URL terdiri dari:

Protocol ://nama-host/path/namafile



Jadi, jika seorang user mengakses suatu website, misalnya

'http://www.geocities.com/dosenku/index.html' maka:

http adalah jenis protocol yang digunakan

'www.geocities.com' adalah nama host yang digunakan

dosenku adalah direktori/path yang dituju

index.html adalah nama file yang diakses

2.4.2. Protocol

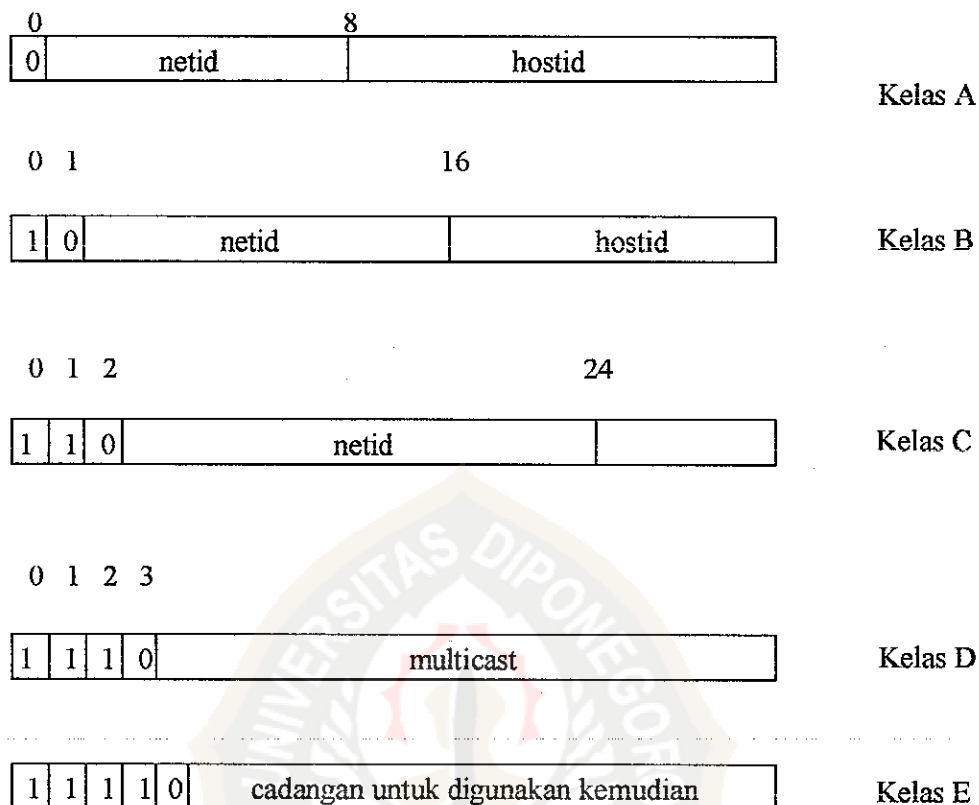
Protocol adalah suatu petunjuk yang menunjukkan pekerjaan yang akan user lakukan dengan Internet, apakah akan mengakses website, melakukan transfer file, mengirim email dan sebagainya. Protocol bisa dibayangkan seperti suatu bahasa yang digunakan untuk berkomunikasi berbagai jenis komputer maupun sistem operasi yang terhubung di Internet.

2.4.3. IP address

IP Address adalah alamat logik dengan panjang 32 bit yang didefinisikan oleh Internet Protocol dalam STD 5 RFC 791. Alamat logik merupakan nomor identifikasi yang diberikan oleh administrator. Dibagi dalam dua bagian yaitu field **netid** yang menunjukkan jaringan kemana *host* dihubungkan, dan field **hostid** yang memberikan suatu pengenalan unik pada setiap host pada suatu jaringan. Sehingga angka 32 bit ini bersifat unik untuk seluruh jaringan termasuk jaringan-jaringan yang terhubung.

RFC (*Request for Comments*) merupakan dokumentasi operasi dari protokol-protokol yang mengahiasi sederetan protocol TCP/IP. Sedangkan STD merupakan bagian rangkaian RFC yang menentukan standar-standar Internet. Daftar resmi mengenai standar Internet terdapat dalam STD 1.

Alamat 32 bit ini dikelompokkan menjadi 4 bagian (oktet) yang terdiri dari 8 bit (xxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx) dimana tiap oktet ini mewakili sebuah bilangan desimal yang berkisar dari 0 dan 255. Untuk menentukan jumlah jaringan dan jumlah station untuk tiap jaringan maka disusun kelas jaringan seperti tampak pada gambar 2.5 berikut:



Gambar 2.5. Kelas-kelas IP Address

Pengalamatan IP address ini berupa susunan angka-angka. Padahal manusia cenderung lebih mudah mengingat huruf dibandingkan angka-angka. Untuk mempermudah maka dilakukan *mapping* (pemetaan) IP address ke **name address** seperti ditunjukkan pada tabel 2.7 :

Tabel 2.7. IP Address

Name address	IP address
'www.idola.net.id'	202.152.9.1

Dengan menggunakan *name address* ini maka perubahan konfigurasi terhadap sebuah host tidak mempengaruhi keberadaan host tersebut. 1 mesin dapat menggunakan banyak nama, tapi 1 nama hanya boleh digunakan oleh 1 mesin saja.

2.4.4. Host

Host merupakan suatu teks yang akan membawa ke alamat yang dituju lewat DNS (*Domain Name System*). Syntax dari penulisan host dapat dituliskan:

nama_host.nama.domain

Sebagai contoh, 'www.vcampus.com'

Maka yang disebut *www* adalah nama *hostnya*, *vcampus* adalah namanya, dan *com* adalah domainnya.

2.4.5. Layanan Internet

Di Internet disediakan pelayanan banyak pelayanan teknis, diantaranya seperti berikut:

- HTTP

Pelayanan ini memungkinkan user untuk mengakses suatu halaman web yang dapat berupa teks, gambar, suara, bahkan film. User dapat sekedar melihat informasi yang ada dengan memasuki sebuah alamat pada suatu URL.

- *Electronic Mail (Email)*

Pelayanan untuk mengirim dan menerima surat elektronik ke mailbox yang ada di Internet. Prinsipnya hampir sama dengan pengiriman lewat pos.

- *Chat*
Pelayanan yang memungkinkan seorang user untuk berhubungan secara langsung dengan user yang lain. Lewat media inilah banyak terdapat suatu *channel* yang memungkinkan banyak user untuk komunikasi interaktif bersama. Setelah koneksi ke Internet user tinggal menjalankan software yang khusus layanan ini dan memilih channel yang dimasukinya.
- *Telnet*
Pelayanan yang memberi kesempatan untuk mengadakan komunikasi tertulis secara interaktif di Internet. Telnet memberi kesempatan dua program bekerja secara kooperatif dengan saling menukar data.
- *File Transfer Protocol (FTP)*
Pelayanan yang memberi kesempatan mentransfer file dari server di Internet (download) atau sebaliknya, untuk mentransfer file ke server (upload).
- *Network File System (NFS)*
Pelayanan akses file-file jarak jauh yang memungkinkan klien-klien untuk mengakses file-file pada host jarak jauh walaupun file-file tersebut disimpan secara lokal.
- *News*
Pelayanan yang digunakan untuk mendapatkan suatu berita dari *web server*.
News terkadang diidentikkan dengan ruang diskusi

2.5. e-Education

e-Education adalah suatu istilah yang digunakan untuk memberi nama pada kegiatan-kegiatan pendidikan yang dilakukan melalui Internet (Oetomo,

2002:92). Seperti ditulis sebelumnya, lahir juga istilah-istilah serba “e”, seperti *e-learning*, *e-consulting*, *e-book*, *e-news* dan berbagai istilah lain. Istilah-istilah ini menunjukkan bahwa kegiatan-kegiatan yang menyertai kegiatan pendidikan tersebut juga memanfaatkan Internet.

2.6. Hypertext Markup Language (HTML)

Agar dapat membangun sebuah web maupun halaman web yang fantastis di Internet, pemahaman HTML sangat diperlukan karena HTML merupakan bahasa utama dalam pembuatan web maupun halaman web di Internet. Ibarat sebuah bangunan, fungsi HTML dalam membangun web adalah sebagai pasir yang merupakan bahan pokok bangunan.

Secara garis besar format HTML dapat digambarkan sebagai berikut:

<HTML>

<HEAD>

...Tempat untuk menuliskan keterangan tentang dokumen / judul...

</HEAD>

<BODY>

....Tempat untuk menuliskan isi dokumen

</BODY>

</HTML>

Keterangan:

- *Head*

Head sebagai kepala dari suatu dokumen HTML yang berisi judul, alamat dasar dari dokumen HTML dan hubungan antara dokumen yang satu dengan yang

lain. Disamping itu, *head* dapat juga berfungsi sebagai pemberi informasi yang tidak akan ditampilkan pada dokumen utama saat HTML dibuka suatu *browser*.

- *Body*

Merupakan bagian utama suatu dokumen yang berisi semua teks yang ditampilkan melalui browser. Di dalam *body* ini terdapat teks dan kode-kode untuk memformat teks tersebut.

2.7. Active Server Pages (ASP)

Microsoft Active Server pages (ASP) merupakan suatu skrip yang bersifat server-side yang ditambahkan pada HTML untuk membuat sebuah web menjadi lebih menarik, dinamis dan interaktif. ASP dapat mengolah data konsumen yang diambil dengan sebuah form, membuat aplikasi-aplikasi tertentu dalam sebuah web, ataupun membuat database dalam sebuah web.

Di atas dikatakan bahwa ASP bersifat *server-side*, yang berarti adalah proses pengerjaan skrip berlangsung di server, bukan di browser/client. Dengan kata lain jika Anda menggunakan sebuah browser untuk memanggil sebuah file ASP, maka browser tersebut mengirimkan permintaan ke web server, kemudian server tersebut mengeksekusi setiap skrip yang ada dan hasilnya dikirimkan kembali ke browser Anda. Karena bersifat *server-side*, maka untuk dapat dijalankan pada sebuah PC biasa yang berbasis *Windows*, PC tersebut perlu disimulasikan menjadi sebuah web server dengan menginstal *Microsoft Personal Web Server (PWS)* atau *Microsoft Internet Information Services (IIS)*

Bahasa skrip standar yang digunakan oleh ASP adalah *Microsoft VBScript* dan *Microsoft JScript*. *Software editornya* dapat menggunakan **notepad** atau

frontpage. *ASP* menggunakan teknologi **VBScript** dan **Jscript** karena *ASP* bukanlah bahasa pemrograman.

File *ASP* berekstensi **.asp** dan berisi *tag HTML*, *text*, dan *script ASP*. Karena *script ASP* merupakan *script* yang berjenis *server-side* (yang berarti segala pengerjaan *script* berlangsung di dalam *server*, bukan di *browser*), sehingga di dalam server mini seperti **PWS (Microsoft Personal Web Server)** terdapat direktori virtual yang telah diberi akses *excute*. Disinilah file *ASP* harus diletakkan supaya dapat dieksekusi dan ditampilkan pada jendela *browser* nantinya.

Untuk membuat database dapat menggunakan **MS Access, Oracle, MS SQL, Visual Basic** atau yang lain.

Aplikasi *database* adalah *database* yang bisa digunakan dengan mudah untuk aplikasi-aplikasi lain. Untuk membuat aplikasi *database* yang bisa dijalankan oleh *web browser* digunakan **ODBC DSN (Open DataBase Connectivity Data Source Name)**. Dengan ODBC, setiap *database* akan dikenali oleh aplikasi lainnya.