

## BAB II

### LANDASAN TEORI

#### 2.1 Internet

*Internet* berasal dari kata *Interconnection Networking* yang mempunyai arti hubungan berbagai komputer dengan berbagai tipe yang membentuk sistem jaringan yang mencakup seluruh dunia (jaringan komputer global) dengan melalui telekomunikasi seperti telepon. [BUS99]

*Internet* bukanlah suatu jaringan tunggal, tetapi terbentuk dari jaringan regional seperti SuraNet, PrepNet, NearNet, AARNET, yang saling dikoneksikan bersama sebagai satu kesatuan dengan menggunakan TCP/IP protocol. Seluruh aktifitas dilakukan secara *real time*. Jaringan-jaringan komputer tersebut saling berkomunikasi melalui *gateway*, atau terkadang disebut dengan *router*. Jaringan kecil yang terhubung tersebut lazim disebut dengan *subnetwork*.

TCP/IP lazim disebut dengan *open system* karena tidak terbatas oleh satu vendor khusus, serta spesifikasinya tersedia untuk publik. Teknologi ini telah didesain untuk sistem komputer dan sistem operasi yang berbeda. TCP/IP digunakan untuk berkomunikasi antar jaringan yang saling diinterkoneksi. *National Science Foundation (NSF)*, *Departement of Defense (DOD)*, *Departement of Energy (DOE)*, *Health and Human Service Agency (HH)*, *National Aeoroetics and Space Administration (NASA)*, menggunakan TCP/IP untuk membentuk *internetworking* yang akhirnya dikenal dengan nama *DARPA NSF Internet*, yang lebih jauh lagi dikenal sebagai *Internet*. [SUT93]

### 2.1.1 FTP (*File Transfer Protocol*)

Melalui *software* FTP, dapat mentransfer data atau *file* dari satu komputer ke komputer lain. Proses mentransfer *file* dari sebuah komputer ke komputer pengguna tersebut disebut *download*, sedangkan proses mentransfer *file* dari komputer pengguna ke komputer lain disebut dengan *up-load*.

### 2.1.2 URL (*Universal Resource Locator*)

Cara mengacu dokumen dalam WWW, contoh: <http://www.Service.com/PAW/home/html>. Bagian pertama menyatakan cara akses, bagian kedua adalah nama *host*, dan bagian ketiga adalah *path*

### 2.1.3 WWW (*World Wide Web*)

Awal mulanya *Web* pada tahun 1989 di CERN, pusat penelitian nuklir di Eropa dimana sejumlah ilmuwan negara-negara Eropa melakukan penelitian tentang fisika partikel. Kelompok ini memiliki anggota yang berasal dari berbagai negara, sebagian besar eksperimennya sangat rumit dan memerlukan perencanaan serta konstruksi yang besar. *Web* tumbuh dari kebutuhan yang diperlukan oleh kelompok yang besar yang terdiri dari peneliti-peneliti yang tersebar bekerja sama menggunakan sejumlah laporan, cetak biru, gambar, foto, dan dokumen lainnya yang selalu berubah.

Pada dasarnya *web* merupakan sistem *client-server*, dimana para pemakai sebagai *client* sedangkan *server* sebagai penyedia informasi. Yang dimaksud dengan *web server* adalah jaringan *client-server* yang menggunakan fasilitas WWW (*hypertext* dan grafik secara bersamaan untuk menampilkan informasi). Pada sudut pandang pengguna, *web* terdiri dari sekumpulan dokumen yang

tersebar di seluruh dunia yang biasanya disebut dengan *page* (halaman). Setiap halaman dapat berisi *link* (pointer) ke halaman lainnya di mana pun tempatnya di dunia. Pengguna dapat mengikuti *link*-nya yang kemudian akan membawa pengguna ke halaman yang dituju. Proses ini dapat terjadi berulang-ulang. Halaman yang menunjukkan ke halaman lainnya disebut dengan *hypertext*. Halaman-halaman ini dapat dilihat dengan program yang disebut dengan *browser*.

## 2.2 Hypertext Markup Language (HTML)

Halaman-halaman *web* biasanya ditulis di dalam bahasa yang disebut dengan HTML (*hypertext mark-up language*). HTML merupakan salah satu format yang digunakan dalam dokumen dan aplikasi yang berjalan di *web browser*. Fungsi dari HTML sendiri untuk memperindah *file* teks biasa untuk ditampilkan pada program *web browser*.

### 2.2.1 Dasar-dasar HTML

HTTP (*Hypertext Transfer Protocol*) merupakan protokol yang digunakan untuk mentransfer data antara *web server* ke *web browser*. Protokol ini mentransfer dokumen-dokumen *web* yang ditulis atau berformat HTML.

Dikatakan *markup language* karena HTML berfungsi untuk memperindah *file* teks biasa untuk ditampilkan pada program *web browser*. Hal ini dilakukan dengan menambahkan elemen atau sering disebut dengan *tag* pada *file* teks biasa tersebut.

*Tag* HTML biasanya berupa *tag-tag* yang berpasangan dan ditandai dengan simbol < dan >. Pasangan dari sebuah *tag* ditandai dengan tanda '/'. Misalnya pasangan dari *tag* <contoh> adalah </contoh>. Dalam hal ini

<contoh> disebut sebagai elemen dan biasanya dalam suatu elemen terdapat atribut-atribut untuk mengatur elemen ini. Jadi misalnya elemen <contoh> bila ditulis dengan atributnya adalah sebagai berikut:

```
<contohatribut1="nilai_atribut1"
atribut2="nilai_atribut2"...>
```

Dalam penulisan *tag* HTML tidaklah *case sensitive* artinya penggunaan huruf kecil ataupun kapital tidaklah menjadi masalah.

### 2.2.2 Struktur Dasar Dokumen HTML

Setiap dokumen HTML memiliki struktur dasar atau susunan *file* sebagai berikut:

```
<html>
<head>
  <title>berisi judul</title>
</head>
<body>
  Berisi tentang text, gambar, atau apapun yang ingin
  Anda tampilkan pada dokumen Anda ada pada bagian ini
</body>
</html>
```

Seperti terlihat, struktur HTML diawali dengan sebuah *tag* <html> dan ditutup dengan *tag* </html>. Di dalam *tag* ini terdapat dua buah bagian besar, yaitu yang diapit oleh *tag* <head> ... <head> dan <body> ... <body>.

Bagian yang diapit oleh *tag* *HEAD* merupakan header dari halaman HTML dan tidak ditampilkan pada browser. Bagian ini berisi *tag-tag header* seperti <title> ... <title> yang berfungsi untuk mengeluarkan judul pada *title bar window web browser*.

Bagian kedua, yang diapit oleh *tag* *BODY* merupakan bagian yang akan ditampilkan pada *web browser*. Pada bagian akan ditulis semua jenis informasi berupa teks dengan berbagai format maupun gambar yang diinginkan.

### 2.3 Alasan memilih web

*Web*, jumlah audien yang dapat dicapai adalah sangat luas, karena secara teoritis siapapun yang dapat mengakses Internet dapat mengakses Web tersebut. Cakupan geografi adalah 'borderless' lintas negara dan benua. Fleksibilitas isi sangat menakjubkan, bisa dirubah sesuai kondisi terakhir dengan kualitas mendekati 'real time' dan dapat di'customize' sesuai keinginan user. Untuk membaca web memerlukan sarana yang paling kompleks dibandingkan media lainnya. Sarana yang diperlukan komputer, modem, telpon line, akses ke ISP (Internet Service Provider) dan beberapa program penunjang terutama untuk web browser. Tetapi teknologi akhir-akhir ini seperti WebtopTV, WAP memudahkan pengaksesan media Web tanpa membutuhkan perangkat komputer biasa.

Hal di atas menyebabkan beberapa pengguna komputer mengembangkan penggunaan suatu sistem alat bantu misalnya, alat bantu belajar dan tes-tes IQ. Web menawarkan suatu sarana yang baru, dimana akan semakin cepatnya kita untuk mendapatkan suatu informasi dalam membantu pekerjaan kita, karena *web* memiliki ruang yang tak terbatas.

### 2.4 Personal Home Page (PHP)

PHP/FI merupakan nama awal dari PHP. PHP adalah *Personal Home Page* yaan terdiri atas *engine parser* yang hanya mengerti beberapa makro khusus dan sejumlah utilitas yang sering digunakan pada halaman *web*, FI adalah *Form Interface*. Di buat pertama kali pada musim gugur tahun 1994 oleh Rasmus Lerdoff. PHP awalnya merupakan program CGI yang dikhususkan untuk

menerima *input* melalui *form* yang ditampilkan di *browser*. Software ini disebar dan dilisensikan sebagai perangkat *open source*. Parser di program ulang pada pertengahan 1995 dan diberi nama PHP/FI versi 2.0. FI berasal dari paket Rasmus lainnya yang ditulis untuk menginterpretasi data dari form, yang kemudian dikombinasikan dengan *tool Personal Home Page* dan ditambahkan dukungan *mSQL* (mini SQL). Perkembangan selanjutnya pertengahan 1997, pengembangan dilakukan oleh tim yang terorganisasi bukan oleh Rasmus saja. *Parser* dikembangkan oleh Zeev Suraski dan Andi Gutmans yang kemudian menjadi dasar untuk PHP versi 3.0. Pada PHP versi 4.0 (terakhir) menggunakan *engine script Zend* untuk lebih meningkatkan kinerja (*performance*) dan mempunyai dukungan yang banyak berupa ekstensi dan fungsi dari *library* pihak ketiga (*third party*), dan berjalan seolah modul asli (*native*) dari berbagai *server web* yang populer.

PHP merupakan *script* untuk pemrograman *script web server-side* (sintaks dan perintah-perintah yang dijalankan di *server web*) yang disisipkan pada HTML. Pembuatan *web* ini merupakan kombinasi dari PHP sebagai bahasa pemrograman dan HTML sebagai pembangun halaman *web*. Ketika pengunjung membuka halaman *web*, server akan memproses perintah PHP dan mengirimkan hasilnya ke browser pengunjung tersebut, juga pada ASP (*active server page*) atau *ColdFusion*.

Tetapi tidak seperti ASP atau *ColdFusion*, PHP mampu lintas *platform*, berjalan di Windows NT, beberap versi UNIX, dapat dibangun sebagai modul pada *webservice apache*, dan sebagai *binary* yang dapat berjalan sebagai CGI.

PHP dapat mengirim HTTP *header*, dapat mengatur *cookies*, *authentication* dan *redirect users*. PHP mampu menawarkan koneksitas yang baik dengan beberapa basis data antara lain Oracle, Sysbase, mSQL, MySQL, Solid, PostgreSQL, Adabas, *Filepro*, Velocis, dBASE, UNIX dbm, dan tak terkecuali semua *database* dengan *interface* ODBC. Dan juga integrasi beberapa *library* eksternal yang dapat membuat segalanya dari dokumen PDF, hingga membentuk *parse* XML. PHP juga mendukung komunikasi dengan layanan melalui protocol IMAP, SNMP, NNTP, POP3 atau bahkan HTTP. Hampir seluruh aplikasi berbasis *web* dapat dibuat dengan PHP.

Dengan menggunakan PHP maka *maintenance* suatu situs menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP. [BET01]

#### 2.4.1 Penulisan Script PHP

Di dalam PHP untuk menuliskan script PHP ada 3 cara:

1. Menggunakan *tag* `<? ... ?>`

Format:            `<?>`  
                       `Script PHP Anda`  
                       `?>`

2. Menggunakan *tag* `<?php ... ?>`

Format:            `<?php>`  
                       `Script PHP Anda`  
                       `?>`

3. Menggunakan *tag* `<script language = "php"> ...`

`</script>`

Format:            `<script language = "php">`  
                       `Script PHP Anda`  
                       `</script>`

Cara pertama akan sering digunakan karena lebih ringkas. Cara kedua digunakan dengan kombinasi XML, sebuah bahasa yang merupakan pengembangan dari HTML. Cara yang ketiga digunakan untuk mengantisipasi *editor-editor* yang tidak dapat menerima kedua cara di atas. Berikut ini contoh sederhana pemakaian bahasa PHP dalam halaman *web*:

```
<html>
  <head>
    <title>Contoh</title>
  </head>
  <body>
    <? Echo " Hai, saya script PHP"; ?>
  </body>
</html>
```

## 2.5 MySQL

MySQL adalah *multi user database* yang menggunakan bahasa *structured query language* (SQL). Dalam operasi *client-server*, melibatkan *server daemon* MySQL disisi *server* dan berbagai macam program serta *library* yang berjalan di sisi *client*. MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku mampu menyimpan data lebih dari 40 *database* dengan 10.000 tabel dan 7 juta baris, totalnya kurang lebih 100 *Gygabytes* data. Perkembangan hingga sekarang MySQL dengan versi 3.23.

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



### 2.5.1 Database, Tabel, Baris, dan Kolom

Dalam konteks bahasa SQL pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi yang terdiri atas baris-baris data yang berada dalam satu atau lebih kolom. Baris pada tabel sering disebut dengan *instance* dari data, sedangkan kolom disebut sebagai *attributes* atau *field*. Keseluruhan tabel itu dihimpun dalam satu kesatuan yang disebut dengan *database*.

### 2.5.2 Tipe Data pada MySQL

Berikut ini adalah tabel tipe data yang digunakan untuk *field-field* tabel pada *database* MySQL.

Tabel 2. 1 Tipe numerik

Tipe Data	Keterangan
TINYINT[(M)]	Data numeris yang tidak punya komponen desimal atau disebut bilangan Bulat. Signed range -2147483648 sampai 2147483647. Unsigned range 0 sampai - 4294967295.
BIGINT[(M)]	Integer ukuran besar. Signed range - 9223372036854775808. Unsigned range 0 sampai - 16777215 sampai 9223372036854775807.
SMALLINT[(M)]	Integer yang kecil. Signed range -32768 sampai 32767. Unsigned range 0 sampai - 65535
MEDIUMINT[(M)]	Integer ukuran sedang. Signed range -8388608 sampai 8388607. Unsigned range 0 sampai - 16777215
FLOAT(Precision)	Bilangan pecahan. FLOAT(4) adalah single precision dan FLOAT(8) adalah double precision. Range -

	3.402823466E+38F sampai -1.175494351E-38,0,-1.175494351E-38 sampai 3.402823466E+38F
FLOAT[(M,D)]	Bilangan floating point/pecahan kecil. Tidak boleh unsigned. Range -3.402823466E+38F sampai -1.175494351E-38, 0, -1.175494351E-38 sampai 3.402823466E+38F
DOUBLE PRECISION[[M,D]]	Bilangan floating point atau pecahan normal / bilangan real. Tidak boleh unsigned.
REAL[(M,D)] DECIMAL[(M,D)]	Sama seperti DOUBLE. Tidak boleh unsigned. Range sama seperti DOUBLE. Berkelakuan seperti CHAR.
INT[(M)]	Data numeris yang tidak punya komponen desimal atau disebut bilangan Bulat. Signed range -2147483648 sampai 2147483647. Unsigned range 0 sampai -4294967295.

Tabel 2. 2 Tipe Date and time

Tipe Data	Keterangan
DATE	Untuk menyimpan informasi tanggal. Format: 'YY-MM-DD', 'YYY-MM-DD', dan 'YYMMDD'. Range 000-00-00 to 9999-12-31.
TIME	Untuk menyimpan informasi waktu. Format: 'HH:MM:SS', 'HHMMSS', 'HHMMSS', 'HHMM', 'HH'.
DATETIME	Untuk menyimpan informasi tanggal dan waktu. Format: "YYYY-MM-DD HH:MM:SS", Range '0000-01-01- 00:00:00' – '9999-12-31 23:59:59'.
YEAR	Informasi tahun.

Tabel 2. 3 Tipe *string*

Tipe Data	Keterangan
CHAR[M]	String dengan panjang tetap. Tersimpan dengan panjangnya saat itu. Range 1 - 255 karakter. Seluruh space case insentive kecuali jika diberikan binary keyword.
VARCHAR[M]	String dengan panjang variabel, selalu tersimpan sesuai dengan panjang yang ditentukan. Range 1 - 255 karakter. Seluruh space akhirnya dibuat saat di-retrieve. Di sortir dan dibandingkan secara case insentive kecuali jika diberikan binary keyword.
TINYBLOB TINYTEXT	Text dengan ukuran pendek. Dengan panjang maksimum 255 karakter.
BLOB TEXT	Deretan karakter yang panjang. Biasanya dipakai pengganti format data text karena keterbatasan ukuran text. Dengan panjang maksimum 255 karakter..
LONGBLOB LONGTEXT	Teks dengan ukuran panjang. Dengan panjang maksimum 4294967295 karakter.
ENUM('Value1','Value2','Value3',...)	Obyek string yang hanya boleh punya satu nilai dari nilai yang dimungkinkan (atau NULL).
SET('Value1','Value2','Value3',...)	Obyek string yang hanya boleh punya satu nilai dari nilai yang dimungkinkan (atau NULL).

Atribut yang digunakan untuk tipe data *integer*:

- AUTO INCREMENT, digunakan untuk membuat *identifier* yang unik dan biasanya digunakan sebagai *primary key*. Nilai *auto increment* biasanya dimulai dengan satu kemudian ditambah satu untuk setiap barisnya. Apabila

*auto increment* bersamaan dengan nilai *bull* maka MySQL akan menambahkan nilai maksimum saat itu dengan satu. Atribut *auto increment* dapat berpasangan dengan *not null* dan *primary key* apabila nilai suatu kolom unik akan dibuat unik.

- PRIMARY KEY, digunakan untuk membedakan antara *record* yang satu dengan *record* yang lain, karena tidak ada dua atau lebih *record* yang memiliki *primary key* sama.
- UNSIGNED, atribut ini tidak memperbolehkan nilai negatif. Dengan atribut *unsigned* tidak akan merubah jangkauan dari tipe data tersebut.

### 2.5.3 Menjalankan MySQL

Sebelum menjalankan MySQL pastikan *daemon* MySQL sudah di-*start* atau belum. Untuk menjalankan MySQL gunakan perintah berikut:

```
# MySQL -u user -p
# password:
```

*User* disini adalah nama *user* pada MySQL, setelah di-*enter* maka diminta memasukan *password*-nya. Jika berhasil maka akan tampil prompt sebagai berikut:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 841 to server version: 3.23.32-
debug
Type 'help;' or '\h' for help. Type '\c' to clear the buffer
mysql>
```

### 2.5.4 Membuat dan Membuka Database

Untuk membuat *database* digunakan perintah berikut ini:

```
mysql> CREATE DATABASE nama_database;
```

untuk membuka *database* digunakan perintah berikut ini:

```
mysql> USE nama_database
```

### 2.5.5 Membuat Tabel

Setelah *database* dibuat, agar data yang dibutuhkan dapat disimpan, maka harus dibuat tabel. Setiap perintah SQL pada MySQL harus diakhiri dengan dengan titik koma (;). Perintah untuk membuat tabel yaitu:

```
mysql> CREATE TABLE nama_table
-> (
-> nama_field1 tipe data,
-> nama_field2 tipe data,
-> ....
-> );
```

### 2.5.6 Perintah *INSERT* dan *SELECT*

Untuk mengisi data pada suatu tabel, digunakan perintah *INSERT*.

Sintaks penulisannya adalah:

```
mysql> INSERT INTO nama_table (field1, field2, ...) VALUES
      (nilai_field1, nilaifield2, ...);
```

Perintah untuk melihat data dari suatu tabel digunakan perintah *SELECT*. Sintaks penulisannya adalah sebagai berikut:

```
mysql> SELECT (field1, field2, ...) FROM nama_tabel;
```

Untuk melihat semua kolom atau *field* data dari satu atau beberapa tabel. Sintaks penulisannya adalah sebagai berikut:

```
mysql> SELECT * FROM nama_tabel;
```

### 2.5.7 Menyaring Hasil *SELECT* dengan *WHERE*

Hasil *SELECT query* dapat dibatasi dengan *keyword WHERE* sehingga *record-record* yang dikeluarkan hanyalah *record* yang sesuai dengan kriteria yang diinginkan. Sintak penulisannya berikut ini:

```
mysql> SELECT (field1, field2, ...) FROM nama_tabel WHERE
      kriteria;
```

### 2.5.8 Mensortir dan Mengelompokan Data

Hasil *query* dapat disortir atau diurutkan sesuai dengan kebutuhan dengan *keyword ORDER BY*. Sintak penulisannya adalah:

```
mysql> SELECT (field1, field2, ...) FROM nama_tabel ORDER BY
        kriteria;
```

Untuk mensortir dengan urutan terbalik, gunakan *keyword* tambahan *DESC*.

Sintak penulisannya adalah:

```
mysql> SELECT (field1, field2, ...) FROM nama_tabel ORDER BY
        kriteria DESC;
```

Untuk mengelompokan berdasarkan salah satu *field* tertentu sehingga menghasilkan satu nilai dari *field* yang berbeda. Sintak penulisannya adalah:

```
mysql> SELECT * FROM nama_tabel GROUP BY fieldx;
```

### 2.5.9 Perintah *DELETE*

Berfungsi untuk menghapus suatu *record* dengan kriteria tertentu. Sintak penulisannya:

```
mysql> DELETE FROM nama_tabel WHERE kriteria;
```

Untuk menghapus seluruh *record* pada suatu tabel, gunakan perintah *DELETE* tanpa memutuskan kriterianya.

```
mysql> DELETE FROM nama_tabel WHERE kriteria;
```

### 2.5.10 Perintah *UPDATE*

Berfungsi untuk memodifikasi nilai kolom atau *field* dari suatu *record*.

Sintak penulisannya adalah:

```
mysql> UPDATE FROM nama_tabel SET nama_field1=nilai_baru1,
        nama_field2=nilai_baru2, ... WHERE kriteria;
```

Untuk memodifikasi nilai suatu kolom atau *field* secara keseluruhan, gunakan perintah *UPDATE* tanpa memutuskan kriterianya.

```
mysql> UPDATE FROM nama_tabel SET nama_field1=nilai_barul;
```

## 2.6 Koneksi PHP dengan MySQL

PHP harus memiliki ekstensi koneksi ke basis data MySQL agar dapat menggunakan fungsi-fungsi koneksi ke basis data MySQL. Sebelum mampu mengakses basis data *server*, terlebih dahulu membuka koneksi basis data. Untuk koneksi ke MySQL *server*, digunakan perintah :

```
<?
$hostname="localhost";
$username="root";
$password="";
    if (!$mysql_link=mysql_connect($hostname,$username,
        $password))
    {
        echo mysql_error();
        exit;
    }
?>
```

Setelah itu variabel `$mysql_link` akan berisi referensi yang akan digunakan untuk proses akses basis data selanjutnya.

### 2.6.1 Query ke Basis Data MySQL

Setelah koneksi sukses dilakukan, sekarang tinggal melakukan *query* ke basis data untuk membaca data, menyisipkan, mengubah, dan sebagainya pada satu atau beberapa tabel dengan menggunakan bahasa SQL. Gunakan perintah berikut untuk mengirimkan *query* (yang dalam contoh ini disimpan dalam variabel `$myquery`) ke *server*:

```
<?
    if (!$mysql_select_db($database, $mysql_link))
    {
        echo mysql_error();
    }
?>
```

```

        exit;
    }
    //membuka tabel
    $myquery = "select * from nama tabel";
    if (!$mysql_result=mysql_query($myquery, $mysql_link))
    {
        echo mysql_error();
        exit;
    }
?>

```

Setelah proses di atas sukses, variabel `$mysql_result` akan berisi *identfier* ke hasil *query* yang setiap digunakan oleh fungsi selanjutnya, misalnya `mysql_fetch_row()`, `mysql_fetch_array`, dan sebagainya.

### 2.6.2 Menampilkan Isi Tabel

Setelah melakukan proses *query*, perlu dilakukan proses untuk menampilkan hasil *query*. Perintah dibawah ini untuk menampilkan dalam bentuk format tabel HTML. Misal hasil *query* dengan tiga buah *field*.

```

<TABLE>
<TR>
    <TD>field1</TD>
    <TD>field2</TD>
    <TD>field3</TD>
    <?
        while($row=mysql_fetch_row($mysql_result))
        {
            echo"</TR>";
            echo"<TD>$row[0]</TD>";
            echo"<TD>$row[1]</TD>";
            echo"<TD>$row[2]</TD>";
        }
    ?>
</TABLE>

```

### 2.6.3 Menambah Record ke Basis Data MySQL

Untuk menyisipkan *record* ke suatu tabel, menggunakan perintah berikut:

```

<?
$myquery ="INSERT INTO nama_tabel(field1, field1, field3)";
$myquery .= "VALUES($field1, $field1, $field3)";
if (!$mysql_result = mysql_query($myquery, $mysql_link))

```



```

        {
            echo mysql_error();
            exit;
        }
    ?>

```

Menyisipkan *record* ke tabel nama\_tabel field1, field2 dan field3 dengan nilai variabel \$field1, \$field2 dan \$field3 dan kriteria dijabarkan dalam variabel kriteria. Biasanya variabel ini adalah variabel eksternal, misalnya dari *form* HTML.

#### 2.6.4 Mengubah *Record* ke Basis Data MySQL

Untuk menyisipkan *record* ke suatu tabel, menggunakan perintah berikut:

```

<?
$myquery ="UPDATE nama_tabel SET field1=$field1";
$myquery .="field2=$field2, field3=$field3";
$myquery .="WHERE kriteria";
if (!mysql_result = mysql_query($myquery, $mysql_link))
{
    echo mysql_error();
    exit;
}
?>

```

Mengubah *record* ke tabel nama\_tabel field1, field2 dan field3 dengan nilai variabel \$field1, \$field2 dan \$field3 dan kriteria dijabarkan dalam variabel kriteria. Biasanya variabel ini adalah variabel eksternal, misalnya dari *form* HTML.

#### 2.6.5 Menghapus *Record* ke Basis Data MySQL

Untuk menghapus *record* ke suatu tabel, menggunakan perintah berikut:

```

<?
$myquery ="DELETE FROM nama_tabel";
$myquery .=" WHERE kriteria";
if (!mysql_result = mysql_query($myquery, $mysql_link))
{
    echo mysql_error();
    exit;
}

```

?>

Menghapus record dari tabel `nama_tabel` dengan kriteria dijabarkan dalam variabel `kriteria`. Biasanya variabel ini adalah variabel eksternal, misalnya dari *form* HTML.

## 2.7 Basis Data

Basis data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebihnya diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. [FAT99]

Basis data sendiri dapat diartikan dalam sejumlah sudut pandang, seperti:

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersamaan sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu untuk memenuhi berbagai kebutuhan.
- c. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik. [FAT99]

### 2.7.1 Objektif Basis Data

Pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan (objektif) seperti berikut ini:

1. Kecepatan dan Kemudahan (*Speed*)

2. Efisiensi ruang penyimpanan (*Space*)
3. Keakuratan (*Accuracy*)
4. Ketersediaan (*Availability*)
5. Kelengkapan (*Completeness*)
6. Keamanan (*Security*)
7. Kebersamaan Pemakaian (*Sharability*)

## **2.8 Sistem Basis Data**

Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi atau tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses / pekerjaan tertentu. Gabungan antara basis data dan program aplikasi (*software*) menghasilkan sebuah sistem.

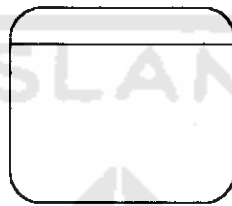
Secara umum sebuah sistem basis data merupakan sistem yang terdiri atas kumpulan *file* (tabel) yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan atau program lain untuk mengakses dan memanipulasi *file-file* (tabel-tabel) tersebut.

### **2.8.1 Data Flow Diagram (DFD)**

Model ini menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data. Sebagai perangkat analisis, model ini hanya mampu memodelkan sistem dari sudut pandang fungsi. Komponen dari fungsi ini adalah:

- a. Proses

Komponen pertama dalam model ini dinamakan proses. Proses menunjukkan transformasi dari masukan menjadi keluaran, dalam hal ini sejumlah masukan dapat menjadi satu keluaran atau sebaliknya. Proses dipresentasikan dalam bentuk lingkaran atau bujur sangkar dengan sudut melengkung. Bentuk proses dapat dilihat pada gambar berikut:



Gambar 2. 1 Bentuk dari proses

b. Aliran

Komponen ini dipresentasikan dengan menggunakan anak panah yang menuju ke atau proses. Aliran digunakan untuk menggambarkan gerakan paket data atau informasi dari satu bagian ke bagian lain dari sistem dimana penyimpanan mewakili lokasi penyimpanan data. Aliran yang digambarkan sebagai panah dengan dua ujung menggambarkan terjadinya dialog. Aliran dapat juga menyebar atau menyatu. Bentuk dari aliran dapat dilihat pada gambar berikut:



Gambar 2. 2 Bentuk dari aliran

c. Penyimpanan

Komponen ini digunakan untuk memodelkan kumpulan data atau paket data. Notasi yang digunakan adalah garis sejajar, segiempat dengan sudut

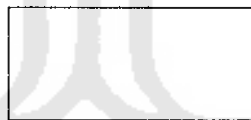
lengkung atau persegi panjang. Bentuk dari penyimpanan dapat dilihat pada gambar berikut:



Gambar 2. 3 Bentuk dari penyimpanan

d. Terminator

Komponen ini di dalam model dipresentasikan menggunakan persegi panjang, yang mewakili entiti dari luar dimana sistem berkomunikasi. Notasi ini melambangkan dari orang atau kelompok orang misalnya organisasi di luar sistem. Pada sejumlah kasus dapat juga merupakan sistem lain. Bentuk dari terminator dapat dilihat pada gambar berikut ini:



Gambar 2. 4 Bentuk dari terminator

## 2.9 Apache Web Server

Web server (atau lebih tepatnya *world wide web*) adalah server Internet yang mampu melayani koneksi transfer data dalam protokol HTTP (hipertext transfer protokol). Web server saat ini merupakan inti dari server-server Internet selain e-mail server, ftp, dan news server. Hal ini belalasan karena web server telah dirancang untuk dapat melayani beragam jenis data, mulai dari text, hypertext, gambar, suara, gambar 3D, plug-in dan sebagainya. Keunggulan ini membuat web dapat diterima tidak saja dikalangan institusi universitas, melainkan hampir semua perusahaan komersial saat ini telah memiliki sebuah atau beberapa web server sekaligus di Internet.

Web server bukan hanya dapat melayani file-file yang ada di dunia Internet. Ia juga dapat dikombinasikan dengan dunia mobile wireless Internet. Dengan menggabungkan web server dengan sebuah WAP (*wireless access protokol*) gateway, maka jadilah web server ini sebuah WAP server yang siap melayani akses mobile Internet pada ponsel-ponsel yang telah memiliki fitur WAP. Dalam konteks ini, web server tidak lagi melayani data file HTML, tapi file-file WML (*wireless markup language*) hanya memiliki. Apache merupakan web server yang paling banyak digunakan di Internet, ini disebabkan karena beberapa faktor, seperti kecepatan performansi, dan juga harganya yang gratis. Apache adalah turunan dari web server yang dikeluarkan oleh NCSA (hhhp:hoofoo.ncsa.uiuc.edu), yaitu NCSA HTTPd pada sekitar tahun 1995-an. Pada dasarnya apache adalah “ A PatCHy” (patch) dan pengganti dari NCSA HTTPd.

## 2.10 Intelligence

Ada banyak definisi intelligensi, dan sedikit persesuaian antara para ilmuwan tentang itu. Orang pertama yang mengemukakan ide untuk mengukur intelligensi ialah sir Francis Galton pada tahun 1880 dan dilanjutkan kesempurnaan oleh Prof.C.E.Spearman. Sebelum itu perkataan atau istilah intelligensi sudah digunakan, artinya informasi. Sebagai contoh, korps inteligen dalam tentara ialah bagian dari tentara yang berhubungan dan bertugas untuk mendapatkan informasi. Tapi dalam penggunaannya Galton mengganti perkataan intelligensi itu menjadi kemampuan umum. Hanya secara berangsur-angsur perkataan itu (intelligensi) menjadi biasa dipakai dalam pengertian moderen. Dalam penggunaan perkataan intelligensi dalam hubungannya dengan kemampuan untuk mengumpulkan dan mempergunakan informasi itu. Juga dengan itu tidak dimaksudkan dengan hanya ingatan yang baik maupun unggul. Tapi lebih bertitik berat, atau berhubungan dengan cara penggunaan materi yang ada dalam materi itu.[SER01]

Intelligensi adalah mengenai informasi. Dan informasi selalu "*dienkode*", yaitu memilih dan mengartikan tanda atau lambang-lambang. Kalau kita mengatakan seseorang cerdas atau pandai, kalau dia mempunyai perbendaharaan informasi yang baik (ingatan); dengan baik memadukan informasi yang baru atau informasi yang sudah ada atau tersimpan; dengan baik menyederhanakan, memadatkan dan mencerna informasi itu untuk dapat digunakan dengan lebih efisien; dan akhirnya dengan baik dapat menggerakkan

dan mengatur serta mengolah informasi itu untuk menghasilkan pemecahan-pemecahan masalah. [SER01]

### **2.10.1 Memberi Tes IQ Pada Anak**

Mengukur inteligensi ialah membandingkan skor atau nilai seorang anak dengan skor rata-rata atau pertengahan dari suatu kelompok. Adalah **Alfred Binet**, adalah seorang ahli psikologi Prancis pada awal abad 20 dan seorang pelopor tes intelligensi anak-anak. Tes binet terdiri dari suatu deretan pertanyaan-pertanyaan yang bertingkat yang disesuaikan atau dikaitkan dengan kemampuan anak yang normal untuk melakukannya melihat tingkat umur yang berturut-turut. Hal ini dilaksanakan terhadap sampel anak-anak dari kelompok umur yang berbeda, dan skor rata-rata dari setiap tingkat ditentukan, dan hasilnya disebut sebagai *skala binet*. [SER01]

### **2.10.2 Metode Pengukuran**

Metode tentang tes IQ sebenarnya bermacam-macam, adapun tes yang dijelaskan dalam karya tulis ini merujuk pada buku yang berjudul *Teslah IQ Anak Anda*, karya Victor Serebriakoff dan Dr. Steven Langer. Buku ini dipakai karena banyak beredar dipasaran, sedangkan buku-buku tes IQ anak yang lain kebanyakan tertutup pada kalangan tertentu. [SER01]

### **2.10.3 Mengenai Tes**

Ada dua tipe tes, yaitu yang berujung terbuka dan yang berujung tertutup. Tes yang berujung terbuka memajukan pertanyaan-pertanyaan terhadap mana yang dapat dijawab dengan sejumlah jawaban yang benar. Contohnya “Sebutlah binatang menyusui yang hidup dilaut”. “ikan paus atau ikan lumba-lumba”



keduanya adalah jawaban yang benar. Tes yang berujung tertutup mempunyai beberapa pertanyaan dengan jawaban pilihan yang memberi suatu jumlah tertentu sebagai jawaban yang benar. Contoh "Manakah binatang-binatang dibawah berikut ini sebagai binatang menyusui dan juga yang hidup dilautan?". "beruang laut, lumba-lumba, ikan hiu atau ikan haring?". Dalam kasus ini jawabannya adalah ikan lumba-lumba, karena yang lainnya itu adalah bukan binatang menyusui atau sebagian hidupnya tidaklah dilautan. [SER01]

Jumlah waktu yang disediakan dalam memberikan tes, berkisar antara 50 menit. Soal-soal yang akan diberikan akan dibedakan menurut umur anak antara lain :

1. Anak berumur antara enam setengah tahun dan sepuluh setengah tahun.
2. Anak berumur antara sepuluh setengah dan empat belas setengah tahun.
3. Anak berumur diatas empat belas setengah tahun.

Dalam menentukan tingkat IQ, telah disediakan soal sebanyak 55 soal (dalam lampiran) yang mana setiap soal dibedakan menurut umur anak yang tes, dan yang menentukan tingkatan IQ ditentukan dengan tabel Interpretasi (dalam lampiran).