

MODUL I Pengenalan PHP

PHP (*Hypertext Preprocessor*), merupakan bahasa pemrograman web bersifat *server-side*, artinya bahasa berbentuk *script* yang disimpan dan dijalankan di komputer *server* (WebServer) sedang hasilnya yang dikirimkan ke komputer *client* (WebBrowser) dalam bentuk *script* HTML (*Hypertext Mark up Language*).

Karakteristik *script* PHP dapat diuraikan sebagai berikut :

- file PHP disimpan dengan ekstensi filenya yaitu : *.php3, *.php4, *.php
- *Script* PHP biasanya diawali dengan tag '<?' atau '<?php' dan ditutup dengan tag '?>'
- File PHP dapat menginduk atau disisipkan pada bahasa *script* lainnya atau dapat berdiri sendiri. Contoh skrip PHP yang disisipkan pada HTML :

```
<html>
<head>
<title>Contoh</title>
</head>
<body>
<?php
    echo "Hai, Aku adalah skrip PHP!";
?>
</body>
</html>
```

Sedang bentuk skrip PHP yang berdiri sendiri :

```
<?php
    echo "Hai, Aku adalah skrip PHP!";
?>
```

NB : **echo()**, berfungsi untuk menampilkan output.

Apa yang dapat dilakukan oleh PHP?

Pada level dasar, PHP dapat melakukan semua apa yang dapat dilakukan oleh pemrograman berbasis CGI lainnya, dan juga kekuatan utamanya adalah dalam pembuatan aplikasi web database. Hampir sebagian besar produk software DBMS dapat didukung oleh PHP baik yang berjalan pada system operasi Windows, Linux maupun system operasi lainnya. Sebagian DBMS yang dapat didukung oleh PHP, di antaranya adalah sebagai berikut :

Adabas	D Ingres	Oracle (OCI7 and OCI8)
DBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

Praktik 1:

1. Buatlah skrip PHP dengan **notepad.exe**, untuk menampilkan salam dan tanggal sekarang, adapun skrip PHP adalah sebagai berikut :

```
<html>
<head>
<title>Coba 1</title>
</head>
<body>
<?php
    echo "<B>Selamat Datang!</B>";
    echo "<HR>\n";
    echo "Hari ini tanggal : " . date("d F Y");
?>
</body>
</html>
```

2. Simpan skrip di atas dengan nama file '**coba.php**' di direktori ditunjuk oleh WebServer.
3. Bukalah aplikasi *WebBrowser* (Internet Explorer(IE) atau Netscape Navigator dll). Ketikkan pada bagian **Address** (IE) atau **Location** (Netscape)

<http://localhost/coba.php>

Perhatikan hasil tampilan dari WebBrowser.

4. Apa kesimpulan Anda?

Praktik 2:

1. Kembangkan program pada **Praktik 1**, yaitu tambah file skrip HTML berikut ini

```
<html>
<head>
<title>Isi Nama</title>
</head>
<body>
<form action="coba.php" method="post">
Silakan Masukkan Nama Anda <input type="text" name="nama"><br>
<input type="submit" value="OK!">
</form>
</body>
</html>
```

Simpan skrip html di atas dengan nama file '**isinama.html**' pada direktori yang dengan file '**coba.php**'.

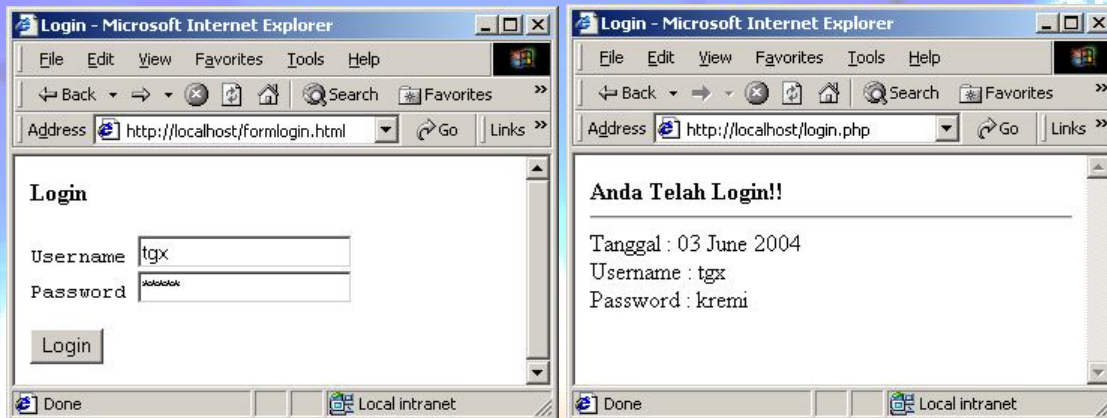
2. Modifikasi file skrip '**coba.php**' di atas

```
<html>
<head>
<title>Coba 2</title>
</head>
<body>
<?php
    echo "<B>Selamat Datang, $nama!</B>";
    echo "<HR>\n";
    echo "Hari ini tanggal : " . date("d F Y");
?>
</body>
</html>
```

3. Simpan dan jalankan dengan mengetikkan alamat berikut pada bagian **address** web browser Internet Explorer.
<http://localhost/isinama.html>
4. Apa kesimpulan anda mengenai praktek 2 ini.

Tugas :

Buatlah **form login** sederhana yang terdiri dari form berbasis HTML dan file skrip PHP yang digunakan untuk menampilkan data login.



MODUL II

Dasar-dasar PHP

Pada dasar-dasar PHP ini akan dijabarkan tentang penulisan komentar, tipe data, konstanta, variabel dan operator.

Komentar

Komentar adalah bagian dari program yang berfungsi sebagai penjelas atau pemberi keterangan dalam program. Komentar ini tidak akan dieksekusi/dikerjakan oleh interpreter.

Untuk mendefinisikan komentar dipergunakan simbol-simbol karakter berikut :

1. dengan symbol **dobel-slash**(//), biasanya untuk komentar satu baris

Contoh :

```
<?php
    // nama program : komentar1.php
    // dibuat tanggal : 3 Juni 2004
    echo "Contoh Komentar dengan '//' ";
?>
```

2. diawali dengan symbol **slash-asterik** (/*) dan ditutup dengan **asterik-slash**(*/), biasanya digunakan untuk memberikan komentar lebih dari satu baris.

Contoh :

```
<?php
    /*
    nama program : komentar2.php
    dibuat tanggal : 3 Juni 2004
    */
    echo "Contoh Komentar dengan '/*' dan '*/' ";
?>
```

baris komentar tidak ditampilkan di halaman webbrowser karena komentar akan diabaikan oleh interpreter.

Tipe Data

Tipe data dasar PHP terdiri dari

- **integer**, termasuk jenis data bilangan bulat
- **double**, termasuk jenis data bilangan pecahan/desimal
- **string**, termasuk jenis data teks/untaian karakter

Contoh :

```
<?php
    $a=10; //variable $a memiliki tipe data integer
    echo $a;
    $b=22.33; //variable $b memiliki tipe data double
    echo $b;
    $c="Skrip PHP"; //variable $c memiliki tipe data string
    echo $c;
?>
```

Variabel

Variabel adalah suatu pengenal dalam program yang berfungsi untuk menyimpan nilai secara sementara dan dapat diubah-ubah nilai.

Untuk mendefinisikan variable, diawali dengan simbol karakter **dollar**('\$') dan diikuti oleh **nama pengenal**.

```
$NamaPengenal = nilai;
```

Adapun aturan dalam menyusun pengenalan :

1. tersusun dari karakter huruf, angka dan underscore(_)
2. tidak boleh mengandung spasi
3. karakter pertama nama pengenalan harus dari karakter huruf atau underscore.
4. huruf kecil dan besar dibedakan

Dalam PHP, tidak diperlukan pendeklarasian variabel dengan tipe datanya seperti bahasa pemrograman pascal. Setiap variabel yang terbentuk dalam program dianggap bertipe variabel, dengan kata lain dapat menampung tipe data dengan jenis apapun.

Contoh :

```
<?php
    $info=10; //variable $info menampung bilangan bulat
    echo $info;
    $info=22.33; //variable $info menampung bilangan pecahan
    echo $info;
    $info="Skrip PHP"; //variable $info menampung data teks/string
    echo $info;
?>
```

Konstanta

Konstanta adalah suatu tetapan nilai dalam program. Konstanta tidak dapat dirubah nilai sewaktu program dijalankan, kalau hal itu dilakukan akan menyebabkan error.

Untuk mendefinisikan konstanta digunakan :

```
define (NamaPengenal, nilai_konstanta);
```

Contoh :

```
<?php
    // konstanta Judul="Hitung Luas Lingkaran"
    define ("Judul", "Hitung Luas Lingkaran");

    // konstanta PHI=3.14
    define ("PHI", 3.14);

    echo Judul;
    $r=10;
    echo "<BR>Jari-jari : $r<BR>\n";

    $luas=PHI * $r * $r;
    echo "Luas Lingkaran = $luas";
?>
```

Operator

Operator adalah suatu symbol yang berfungsi untuk menyusun sebuah ekspresi maupun operasi. Sedangkan yang dioperasikan operator disebut dengan operand. Adapun macam-macam operator yaitu :

1. Operator Aritmetika
Merupakan symbol-simbol operator untuk melakukan operasi matematis.

Operator	Fungsi	Prioritas
+	Penjumlahan	Ketiga
-	Pengurangan	Ketiga
*	Perkalian	Kedua
/	Pembagian	Kedua
%	Sisa Pembagian	Kedua
++	Penaikan	Pertama
--	Penurunan	Pertama

Contoh :

```
<?php
$bil1 = 200;
$bil2 = 33;

$hasil = $bil1 + $bil2;
echo "$bil1 + $bil2 = $hasil<BR>\n";

$hasil = $bil1 - $bil2;
echo "$bil1 - $bil2 = $hasil<BR>\n";

$hasil = $bil1 * $bil2;
echo "$bil1 * $bil2 = $hasil<BR>\n";

$hasil = $bil1 / $bil2;
echo "$bil1 / $bil2 = $hasil<BR>\n";

$hasil = $bil1 % $bil2;
echo "$bil1 % $bil2 = $hasil<BR>\n";

$hasil = $bil1++;
echo "$bil1++ = $hasil<BR>\n";

$hasil = $bil2--;
echo "$bil2-- = $hasil<BR>\n";
?>
```

2. Operator Perbandingan

Merupakan simbol-simbol operator untuk melakukan perbandingan antara dua buah operand. Hasil perbandingan bernilai **salu**(1) jika benar dan bernilai **nol**(0) jika salah.

Operator	Fungsi
<	Lebih kecil
>	Lebih besar
<=	Lebih kecil atau sama dengan
>=	Lebih besar atau sama dengan
==	Sama dengan
!=	Tidak sama dengan

Contoh :

```
<?php
$bil1 = 100;
$bil2 = 20;
$teks1 = "PHP";
$teks2 = "php";
printf("%d == %d adalah %d<BR>\n", $bil1, $bil2, $bil1 == $bil2);
```

```
printf("%d != %d adalah %d<BR>\n", $bil1, $bil2, $bil1 != $bil2);
printf("%d >= %d adalah %d<BR>\n", $bil1, $bil2, $bil1 >= $bil2);

printf("%s == %s adalah %d<BR>\n", $teks1, $teks2, $teks1 == $teks2);
printf("%s != %s adalah %d<BR>\n", $teks1, $teks2, $teks1 != $teks2);

?>
```

3. Operator Logika

Merupakan symbol-simbol operator untuk menyusun kalimat ekspresi/ungkapan logika. Hasil operasi ini akan didapatkan nilai **salu(1)** jika bernilai benar atau **nol(0)** jika bernilai salah.

Operator	Fungsi
AND atau &&	Operasi logika and
OR atau 	Operasi logika or
XOR	Operasi logika eksklusif or
!	Ingkaran/negasi

Untuk lebih jelasnya mengenai penggunaan operator-operator di atas, perhatikan table kebenaran sebagai berikut :

\$p	\$q	\$p and \$q	\$p or \$q	\$p xor \$q	! (\$p and \$q)
1	1	1	1	0	0
1	0	0	1	1	1
0	1	0	1	1	1
0	0	0	0	0	1

Contoh :

```
<?php
$bil1 = 100;
$bil2 = 20;
$teks1 = "PHP";
$teks2 = "php";
$hasil = ($bil1 <> $bil2) or ($teks1 == $teks2);
printf("%d <> %d) or (%s == %s) adalah %d<BR>\n",
$bil1, $bil2, $teks1, $teks2, $hasil);
$hasil = ! ($teks1 == $teks2);
printf("! (%s == %s) adalah %d<BR>\n", $teks1, $teks2, $hasil);

?>
```

4. Operator String

Dalam PHP juga tersedia operator string, yaitu untuk operasi penggabungan teks. Adapun symbol yang digunakan yaitu berupa karakter **titik/point (.)**.

Contoh :

```
<?php
$teks1 = "Aku Sedang Belajar";
$teks2 = "Pemrograman WEB";
$teks3 = "PHP 4";
$hasil = $teks1 . $teks2 . $teks3;
printf("hasil : %s<BR>\n", $hasil);
$hasil = $teks1 . " " . $teks2 . " " . $teks3;
printf("hasil : %s<BR>\n", $hasil);

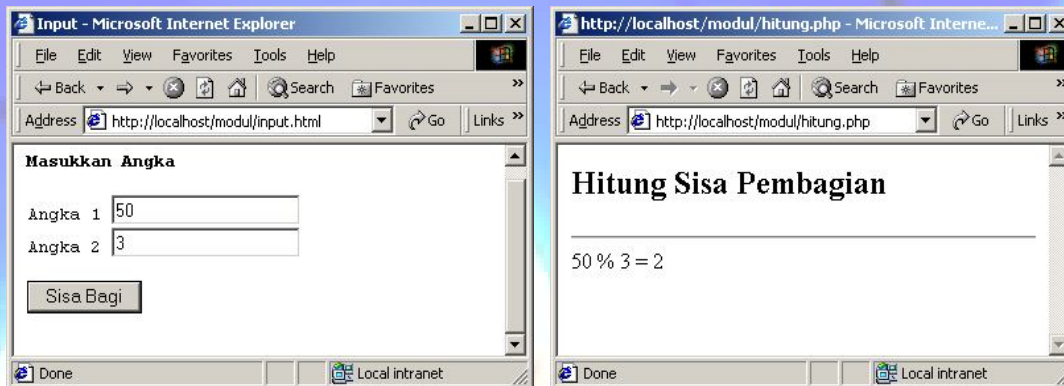
?>
```

Praktik :

1. Ketikkan contoh-contoh program di atas dengan **notepad.exe**, dan jalankan.
2. Amati dan analisa hasil tampilan(*output*) setiap program di atas.
3. Berikan kesimpulan mengenai program-program di atas.

Tugas :

Buatlah program untuk menghitung sisa pembagian antara **angka1** dengan **angka2**.



MODUL III PERNYATAAN SELEKSI

Apa yang dimaksud pernyataan seleksi?

Sebagian besar bahasa pemrograman mengandung pernyataan seleksi. Pada dasarnya pernyataan seleksi adalah suatu mekanisme suatu pernyataan akan dikerjakan atau tidak, hal tergantung pada kondisi yang dirumuskan. Dalam PHP atau bahasa pemrograman lainnya pernyataan seleksi diterapkan dengan menggunakan statemen **if**.

if

Bentuk umum pernyataan **if**, adalah sbb:

```
If (kondisi) {  
    statement;  
}
```

Prinsip Kerja : *Statement* di atas akan dikerjakan apabila kondisi bernilai **TRUE** (1) sedangkan jika kondisi bernilai **FALSE** (0), maka *statement* di atas tidak akan dikerjakan.

Contoh : Mengabsolutkan nilai bulat

```
<?php  
    $a = -10;  
    if ($a < 0) {  
        $a = $a * -1;  
    }  
    echo $a;  
?>
```

else

sebenarnya pernyataan **else** merupakan bagian dari pernyataan **if**, yang mana pernyataan **else** ini digunakan untuk memberikan alternatif perintah apabila kondisi bernilai **FALSE** (0). Bentuk umum pernyataan :

```
If (kondisi) {  
    Statement1;  
} else {  
    Statement2;  
}
```

Contoh :

```
<?php  
    $nilai = 40;  
    if ($nilai >= 60) {  
        echo "Nilai Anda : $nilai. Selamat, Anda Telah Lulus!!";  
    } else {  
        echo "Nilai Anda : $nilai. Sorry, Anda Tidak Lulus!!";  
    }  
?>
```

Contoh di atas akan menguji kondisi $\$nilai \geq 60$, ternyata kondisi bernilai salah (FALSE) sehingga yang dikerjakan adalah statemen dibawah else, yaitu akan ditampilkan hasil output program “Nilai Anda : 40. Sorry, Anda Tidak Lulus!!”.

elseif

Jika pernyataan else memberikan alternatif pilihan kedua pada pernyataan di atas, maka pernyataan elseif adalah untuk merumuskan banyak alternatif pilihan (lebih dari dua pilihan). Adapun bentuk umumnya :

```
if (kondisi1) {
    statement_1;
} elseif (kondisi2) {
    statement_2;
} .....
} else {
    statement_n;
}
```

Contoh :

```
<?php
$nilai = 50;
if (($nilai >= 0) && ($nilai < 50)) {
    $indek = "E";
} elseif (($nilai >= 50) && ($nilai < 60)) {
    $indek = "D";
} elseif (($nilai >= 60) && ($nilai < 75)) {
    $indek = "C";
} elseif (($nilai >= 75) && ($nilai < 85)) {
    $indek = "B";
} elseif (($nilai >= 85) && ($nilai < 100)) {
    $indek = "A";
} else {
    $indek = "Nilai diluar jangkuan";
}
echo "Nilai Anda : $nilai, dikonversi menjadi '$indek' ";
?>
```

switch

Serupa dengan pernyataan if – elseif – else, pernyataan switch juga memberikan banyak alternatif pilihan pernyataan. Adapun bentuk umum switch :

```
switch (nilai_ekspresi) {
case nilai_1 : statement_1; break;
case nilai_2 : statement_2; break;
default :
    statement_n;
}
```

contoh :

```
<?
$angka = 6;
switch ($angka){
case 0: $terbilang = "NOL"; break;
```

```
case 1: $terbilang = "SATU"; break;
case 2: $terbilang = "DUA"; break;
case 3: $terbilang = "TIGA"; break;
case 4: $terbilang = "EMPAT"; break;
case 5: $terbilang = "LIMA"; break;
case 6: $terbilang = "ENAM"; break;
case 7: $terbilang = "TUJUH"; break;
case 8: $terbilang = "DELAPAN"; break;
case 9: $terbilang = "SEMBILAN"; break;
default: $terbilang = "Nilai diluar jangkuan!!";
}
printf("Bentuk terbilang dari angka '%d' adalah '%s' ", $angka, $terbilang);
?>
```

Praktik :

1. Ketikkan setiap contoh-contoh program di atas dengan **notepad.exe** dan kemudian jalankan!
2. Amati dan analisa hasil tampilan atau output program!
3. Modifikasilah dengan memberikan input nilai yang berbeda, kemudian amati dan analisa hasil tampilan(output) program!
4. Apa yang anda dapat simpulkan terhadap pratikum kali ini?

MODUL IV PERNYATAAN PERULANGAN

Pernyataan perulangan digunakan memproses/mengeksekusi pernyataan atau statemen lebih dari satu kali. Macam-macam pernyataan perulangan di PHP meliputi :

while

Pernyataan while akan melakukan memproses secara berulang terhadap pernyataan berdasarkan atas kondisi.

Bentuk Umum while

```
while (kondisi) {  
    Statemen;  
}
```

Prinsip Kerja : pertama kali pernyataan while akan menguji kondisi yang dirumuskan, jika kondisi bernilai TRUE (1) maka statemen di bawahnya akan dikerjakan sekali lagi. Namun jika kondisi bernilai FALSE (0) maka perulangan akan dihentikan. Setiap kali statemen di bawahnya selesai dikerjakan kondisi akan selalu diuji.

Contoh :

```
<?php  
$I = 1;  
while ($I <= 7) {  
    echo "<FONT SIZE=$I>Perulangan ke-$I</FONT><BR>\n";  
    $I++;  
}  
?>
```

Hal yang perlu diperhatikan dalam pernyataan while yaitu

- inialisasi nilai konter pada program di atas ditunjukkan pada pernyataan `$I = 1;`
- penetapan kondisi pada program di atas ditunjukkan pada pernyataan `$I <= 7;` yang memiliki maksud, program akan mengulang selama nilai `$I` kurang atau sama dengan 7
- Operasi penaikan konter Hal ini ditunjukkan pada pernyataan `$I++`, maksudnya adalah untuk merubah nilai `$I` setiap kali looping terjadi, sehingga suatu saat perulangan akan dihentikan.

do - while

Seperti halnya pernyataan while, pernyataan do – while memiliki prinsip kerja yang sama dengan pernyataan while, hanya saja pernyataan do – while akan menguji kondisi pada setiap akhir perulangan. Adapun bentuk umum pernyataan do – while :

```
do{  
    Statemen;  
} while (kondisi);
```

Prinsip Kerja : Mula-mula statemen akan dikerjakan pertama kali tanpa perlu persyaratan kondisi, selesai pengerjaan statemen akan diuji kondisi. Bila kondisi bernilai TRUE (1) maka program akan mengeksekusi statemen sekali lagi. Dan jika kondisi bernilai FALSE maka perulangan akan berakhir.

Contoh :

```
<?php
$I = 1;
do{
    echo "<FONT SIZE=$I>Perulangan ke-$I</FONT><BR>\n";
    $I++;
} while ($I <= 7);
?>
```

for

Pernyataan for biasanya digunakan apabila jumlah perulangannya telah pasti.

Bentuk Umum pernyataan for adalah sebagai berikut :

```
for (ekspresi1; ekspresi2; ekspresi3) {
    Statemen;
}
```

Keterangan :

- ekspresi1 → inisialisasi/penentuan nilai awal konter
- ekspresi2 → perumusan kondisi
- ekspresi3 → operasi penaikan/penurunan konter

Contoh

```
<?php
for ($I = 1; $I <= 7; $I++) {
    echo "<FONT SIZE=$I>Perulangan ke-$I</FONT><BR>\n";
}
?>
```

Praktik :

1. Ketikkan setiap program contoh di atas dengan **notepad.exe** dan kemudian jalankan!
2. Amati dan analisa hasil setiap contoh program di atas!
3. Berdasarkan logika Anda, apa perbedaan antara pernyataan **while** dengan **do - while** jika kondisi perulangan diubah menjadi **\$I <= 0**.
4. Kita ketahui bahwa setiap program di atas menampilkan output secara urut dari kecil ke besar (Ascending), Apa yang perlu dirubah pada setiap contoh program di atas apabila output harus menampilkan dari besar ke kecil (Descending).
5. Apa kesimpulan anda praktik kali ini.

Tugas :

1. Buatlah program untuk menampilkan deret bilangan kelipatan tiga (3) dari range 0 sampai 100.
2. Buatlah program untuk menampilkan deret bilangan yang habis dibagi dengan bilangan 6 dan 9 dari range deret 0 sampai 100.

MODUL V

Sekilas Tentang MySQL

MySQL merupakan produk software DBMS yang cukup populer dipasaran, karena DBMS ini didistribusikan secara freeware atau gratis dan juga support untuk berbagai platform bahkan sekarang juga telah tersedia paketan untuk platform sistem operasi Windows. Selain gratis DBMS MySQL ini juga memiliki kecepatan akses yang lebih baik selain produk-produk DBMS sekilas sehingga MySQL sangat cocok untuk diterapkan pada aplikasi-aplikasi pemrograman berbasis internet.

Berikut ini akan diuraikan beberapa perintah-perintah untuk mengakses DBMS MySQL yang dapat diketikkan pada aplikasi MySQL Client.

1. Perintah membuat database

Bentuk umumnya:

```
CREATE DATABASE namadatabase
```

contoh :

```
CREATE DATABASE cobadb;
```

apabila perintah di atas dijalankan DBMS Server MySQL akan menciptakan sebuah database dengan nama 'cobadb'. Tentu saja perintah diketikkan setelah berhasil login ke MySQL. Adapun perintah untuk login ke database seperti berikut ini:

```
prompt>mysql -u username -p password
```

contoh :

```
C:\>mysql -u root -p rahasia
```

sehingga apabila berhasil login akan muncul tampilan berikut ini.

```
mysql>_
```

setelah muncul tampilan di atas mysql telah siap menerima perintah-perintah query untuk akses data seperti membuat database di atas.

2. Perintah mengaktifkan database.

bentuk umumnya:

```
USE namadatabase
```

contoh:

```
USE cobadb
```

3. Perintah membuat tabel di database.

bentuk umum:

```
CREATE TABLE namatabel  
(  
  namafield1 type [NOT NULL] [AUTO_INCREMENT],  
  namafield2 type [NOT NULL] [AUTO_INCREMENT],  
  .....  
  [PRIMARY KEY(namafield)]  
);
```

keterangan:

namafield1, namafield2, adalah nama field anggota tabel.

NOT NULL, adalah berfungsi untuk membuat status field sebagai field mandatory field, sehingga apabila saat terjadi penambahan record tabel field ini tidak boleh dikosongkan.

contoh:

```
CREATE TABLE tamu  
(  
  idtamu int NOT NULL AUTO_INCREMENT,  
  nmtamu VARCHAR(35) NOT NULL,  
  email VARCHAR(35) NOT NULL,  
  PRIMARY KEY(idtamu)  
);
```

apabila perintah di atas dijalankan maka haruslah databasenya diaktifkan dulu dengan perintah

```
USE cobadb;
```

tabel di atas memiliki tiga macam field dimana field idtamu berfungsi sebagai field kunci primer.

4. Perintah untuk menghapus struktur tabel

bentuk umumnya:

```
DROP TABLE namatabel;
```

contoh:

```
DROP TABLE tamu;
```

efek perintah jika berhasil dijalankan maka tabel dengan nama 'tamu' akan dihapus secara permanen berikut dengan data yang ada dalam tabel juga akan terhapus.

5. Mengubah struktur tabel

bentuk umum:

```
ALTER TABLE namatabel  
ADD | DROP | MODIFY field
```

contoh, menambah field sex (jenis kelamin) pada tabel tamu.

```
ALTER TABLE tamu  
ADD sex enum('L','P');
```

sedangkan contoh menghapus field 'sex' pada tabel tamu adalah sebagai berikut:

```
ALTER TABLE tamu  
DROP sex;
```

perintah-perintah di atas tergolong perintah DDL(Data Definition Language), yang hanya dijalankan pada saat database akan dikonstruksikan. Sedangkan perintah-perintah untuk memanipulasi data dalam database (DML singkatan dari Data Manipulation Language) akan diuraikan berikut ini:

6. Perintah input data dalam tabel

Bentuk umumnya adalah seperti berikut ini:

```
INSERT INTO namatabel [(field1,field2, ...)]  
VALUES (datafield1, datafield2, ....);
```

contoh:

```
INSERT INTO tamu (nmtamu, email)  
VALUES ('Budiman Raharjo', 'budiman@telkom.net');
```

7. Perintah untuk mengupdate data tabel.

Bentuk Umumnya :

```
UPDATE namatabel  
SET namafield1=databaru1, namafield2=databaru2, ...  
WHERE kondisi
```

keterangan:

namatabel, adalah nama tabel yang akan dilakukan pengupdatean datanya.

namafield1, namafield2 adalah field-field yang akan disunting datanya dengan databarunya

databaru, adalah diisi dengan data baru yang akan mereplace data lama.

kondisi, adalah merupakan penentuan kriteria record-record yang akan diupdate.

contoh:

```
UPDATE tamu  
SET nmtamu='Budiman Sukarno'  
WHERE nmtamu='Budiman Raharjo'
```

8. Menghapus record tabel

Bentuk umumnya:

```
DELETE FROM namatabel  
WHERE kondisi
```

keterangan:

namatabel, adalah nama tabel yang akan dilakukan penghapusan record datanya.
kondisi, adalah merupakan penentuan kriteria record-record yang akan dihapus.

contoh:

```
DELETE FROM tamu  
WHERE idtamu=3
```

9. Perintah untuk menampilkan record-record data tabel

Bentuk umumnya:

```
SELECT * | field1, field2, ...  
FROM nama_table1 [ , nama_table2, ...]  
[ WHERE kondisi ]  
[ GROUP BY ekspresi ]  
[ HAVING kondisi_pencarian ]  
[ ORDER BY ekspresi_pengurutan [ ASC | DESC ] ]
```

Keterangan :

SELECT *

berfungsi untuk menampilkan data seluruh field dari table yang disebutkan setelah klausa FROM

SELECT field1, field2, ...

berfungsi untuk menampilkan data-data dari field-field yang dituliskan secara eksplisit dari table yang disebutkan setelah klausa SELECT

FROM nama_tabel1, nama_tabel2

berfungsi untuk menentukan tabel yang field-field akan ditampilkan data recordnya.

WHERE kondisi

berfungsi untuk menentukan criteria record-record yang akan dihapus.

GROUP BY ekspresi

berisi daftar record-record untuk dijadikan acuan kriteria terhadap record-record data yang dikelompokkan.

HAVING kondisi_pencarian

adalah rumusan kondisi/kriteria record yang digroup yang akan ditampilkan

ORDER BY ekspresi_pengurutan

berisi daftar field-field yang akan dijadikan acuan pengurutan.

ASC

adalah model pengurutan record-record data secara menaik.

DESC

adalah model pengurutan record-record data secara menurun.

Contoh :

```
SELECT * FROM tamu;
```

Perintah di atas jika dijalankan akan menampilkan seluruh record dari seluruh field dalam tabel tamu.

```
SELECT nmtamu, email FROM tamu;
```

perintah di atas jika dijalankan akan menampilkan seluruh data record-record dari field nmtamu dan email.

```
SELECT * FROM tamu  
WHERE nmtamu LIKE 'Budi%';
```

perintah di atas menampilkan data-data record yang memiliki nama dengan awalan 'Budi'

```
SELECT * FROM tamu  
WHERE idtamu=2;
```

perintah di atas akan menampilkan data record tabel yang memiliki idtamu bernilai 2.

contoh lain: untuk menampilkan data dari kombinasi field-field beberapa tabel yang saling berelasi.

tabel kategori

```
kdkategori *  
nmkategori
```

tabel barang

kdbarang *
nbarang
jmlbarang
hrgbarang
kdkategori **

berdasarkan struktur tabel-tabel di atas akan dibuat laporan dengan susunan field yang dibutuhkan adalah sebagai berikut:

kdbarang	nbarang	jmlbarang	hrgbarang	nmkategori
...

sehingga bentuk script SQL yang dapat disusun adalah sebagai berikut:

```
SELECT barang.kdbarang, barang.nbarang, barang.jmlbarang,  
barang.hrgbarang, kategori.nmkategori  
FROM barang, kategori  
WHERE barang.kdkategori = kategori.kdkategori;
```

MODUL VI

Koneksi Data PHP dengan MySQL

Berikut ini akan diuraikan pemakaian fungsi-fungsi untuk mengkoneksikan data ke DBMS MySQL dengan PHP.

1. fungsi koneksi ke instance server MySQL

```
integer mysql_connect(nama_server, nama_user, password);
```

keterangan:

nama_server, adalah nama komputer server yang memuat DBMS MySQL yang dituju.

nama_user, adalah nama user DBMS MySQL yang memiliki otoritas untuk login ke database.

password, adalah password user login ke DBMS MySQL.

contoh:

```
<?
$id=mysql_connect("localhost", "root", "");
if (! $id)
    echo "Gagal koneksi dengan Server";
else
    echo "Sukses koneksi dengan Server";
?>
```

program di atas apabila dijalankan akan menampilkan output

Sukses koneksi dengan Server

jika berhasil terkoneksi ke dbms server mysql, namun jika gagal terkoneksi ke database akan ditampilkan output:

Gagal koneksi dengan Server

2. Memilih Database dalam MySQL

```
integer mysql_select_db(string nama_database [, integer
id_koneksi]);
```

keterangan:

nama_database, adalah diisi dengan nama database yang akan dituju dalam DBMS MySQL.

id_koneksi, diisi dengan pengenal koneksi yang telah aktif.

contoh:

```
<?php
```

```
$id=mysql_connect("localhost", "root", "rahasia");
$db=mysql_select_db("test", $id);
if (! $db)
    echo "gagal membuka database test!";
else
    echo "sukses membuka database test!";
?>
```

3. fungsi untuk merequest query ke DBMS MySQL

Bentuk Umumnya:

```
result mysql_query( string perintah_sql [, integer idkoneksi]);
```

keterangan:

perintah_sql, berisi perintah-perintah untuk mengakses data dalam database dalam syntax SQL

idkoneksi, diisi dengan identitas koneksi yang aktif.

4. fungsi mysql_fetch_row()

bentuk umum:

```
array mysql_fetch_row(result r)
```

keterangan:

r, adalah variabel yang menampung hasil eksekusi query pada fungsi mysql_query().

contoh:

```
<?php
$id=mysql_connect("localhost", "root", "rahasia");
mysql_select_db("test", $id);
$sql="SELECT nimhs, nmmhs, prodi FROM msmhs ";
$hasil=mysql_query($sql, $id);
while ($row=mysql_fetch_row($hasil)){
    echo "$row[0], $row[1], $row[2]<br>\n";
}
?>
```

5. Menampilkan Jumlah Record Pada Recordset

Bentuk Umum:

```
integer mysql_num_rows(result r)
```

keterangan:

r, adalah variabel yang menampung hasil eksekusi query pada fungsi mysql_query().

fungsi di atas akan menghasilkan sejumlah record dari recordset.

6. Fungsi *Mysql_Affected_Rows()*

Berfungsi untuk mengetahui jumlah record yang termanipulasi, biasanya fungsi ini digunakan untuk mengidentifikasi keberhasilan perintah query update & delete. Adapun bentuk umumnya adalah:

```
integer mysql_affected_rows(result r)
```

contoh:

```
<?php
$id=mysql_connect("localhost", "root", "rahasia");
mysql_select_db("test", $id);
$sql="DELETE FROM msmhs WHERE kdpst='TI' ";
mysql_query($sql, $id);
printf("Telah terhapus %d record", mysql_affected_rows());
?>
```

Praktik:

Diketahui sebuah tabel tamu dalam Database MySQL adalah sebagai berikut :

Nama Field	Tipe Data	Panjang	Keterangan
idtam	int		identitas tamu, sebagai field kunci primer dan AUTO_INCREMENT
nmtamu	varchar	35	untuk mengisi data nama tamu
email	varchar	40	untuk mengisi data alamat email tamu

Buatlah form entry data tamu dengan menggunakan pemrograman PHP!