

Spesifikasi:

Ukuran: 14x21 cm

Tebal: 193 hlm

Harga: Rp 37.800

Terbit pertama: Januari 2005

Sinopsis singkat:

Problem utama dalam pembuatan aplikasi web seringkali terletak pada pencetakan laporan yang tidak rapi. Jika menggunakan default tampilan di web browser hasil pencetakan tidak terlalu memuaskan.

Pemisahan halaman seringkali tidak pada batas yang diinginkan. Namun jika mengandalkan pemanfaatan aplikasi pengolah kata (word processor) yang berjalan di salah satu sistem operasi, maka sistem operasi lain tentunya tidak dapat membaca laporan tersebut dengan format yang sama, bahkan mungkin tidak ada aplikasi yang terhubung dengan laporan tersebut. Untuk itu diperlukan bentuk laporan yang universal dan dapat digunakan oleh komputer dengan sistem operasi apapun.

Laporan PDF dianggap cukup universal dan telah tersedia aplikasi pembacanya di berbagai sistem operasi, baik Windows maupun Linux. Untuk itu diperlukan kemampuan tambahan agar PHP dapat membuat laporan dalam format PDF.

Di dalam buku ini membahas mengenai pembuatan laporan berbasis PDF untuk aplikasi web. Trik dalam pembuatan laporan PDF dapat ditemukan di sini untuk mempermudah membuat bentuk-bentuk laporan yang diinginkan.

BAB 2

PHP 5.0 DAN MYSQL 5

Pada bab ini, pembahasan PHP 5.0 hanya akan menjadi dasar untuk dapat lebih memahami pembahasan pada bab selanjutnya, bukan memperdalam pemanfaatan pustaka-pustaka yang dimilikinya atau penggunaan SQLite yang merupakan hal baru dalam PHP 5.0. Begitu juga dengan pembahasan MySQL 5.0, sebatas pada pengelolaan data, bukan pada store procedure yang merupakan hal baru bagi MySQL.

Pada akhir bab ini, pengelolaan data MySQL 5.0 akan memanfaatkan PHPMyAdmin yang cukup mudah dan familiar karena didukung oleh tampilan grafis berbasis web. Dengan adanya dukungan Bahasa Indonesia pada PHPMyAdmin, pengelolaan data MySQL akan semakin mudah.

2.1 Dasar-Dasar PHP

Berikut ini terdapat contoh script PHP sederhana yang dapat memberikan sedikit gambaran akan betapa mudahnya mempelajari PHP. Script PHP di bawah ini:

```
<html>
<head><title>Judul Web</title></head>
<body>
<?
echo "Percobaan pertama!";
echo "<br>";
?>
```

```
</body>  
</html>
```

akan sama hasilnya dengan file HTML dengan source seperti berikut ini:

```
<html>  
<head><title>Judul Web</title></head>  
<body>  
Percobaan pertama!<br>  
</body>  
</html>
```

Perintah echo tidak terlihat pada source HTML di atas karena perintah tersebut telah diproses di dalam web server. Hasil proses tersebutlah yang akan dibaca oleh kita melalui browser sehingga script PHP yang ditulis tidak dapat diketahui oleh kita (client). Inilah yang disebut sebagai server-side scripting, script yang diproses di sisi server dan berbeda dengan Javascript yang diproses oleh komputer client (client-side scripting), dimana script yang dibuat tetap dapat dilihat oleh client melalui menu View | Source.

Script PHP dapat disisipkan di bagian manapun dari HTML, asalkan ditulis di antara tag `<? dan ?>` atau `<?php dan ?>`. Tag tersebutlah yang memberi tanda bahwa di dalamnya terdapat kode-kode PHP yang sebagian besar mirip dengan bahasa yang digunakan pada pemrograman bahasa C. Setiap akhir perintah diakhiri dengan tanda titik koma (semicolons). Jika Anda sudah pernah mempelajari pemrograman dalam bahasa C, tentu akan memudahkan dalam pemahaman PHP. Tetapi jika belum, tidak akan menghalangi Anda dalam memahami PHP.

Listing program berikut ini merupakan latihan perhitungan aritmateka sederhana. Materi ini selalu menjadi bagian dasar dari berbagai bahasa pemrograman. Pada contoh berikut, kita akan mencoba untuk memproses perhitungan yang paling sederhana dan mudah dimengerti pada 2 (dua) variabel. Dengan begitu, operasi aritmateka pada PHP dapat kita pahami terlebih dahulu.

2.1.1 Perhitungan dengan Dua Variabel

Variabel yang pertama-tama diberi nilai awal adalah a dan b dengan isi masing-masing adalah 10 dan 3. Variabel c, d, e, f, dan g hanya dikenalkan sebagai variabel integer tanpa diisi dengan nilai apapun sehingga nilai dari variabel c, d, e, f, dan g masih null. Nantinya nilai dari kelima variabel terakhir ini akan diisi dengan hasil perhitungan antara variabel a dan b. Perhitungan aritmateka yang diinginkan adalah :

c = a ditambah dengan b	(c = a + b)
d = a dikurangi dengan b	(d = a - b)
e = a dikalikan dengan b	(e = a * b)
f = a dibagi dengan b	(f = a / b)
g = sisa dari pembagian antara a dengan b	(g = a % b)

latih01.php

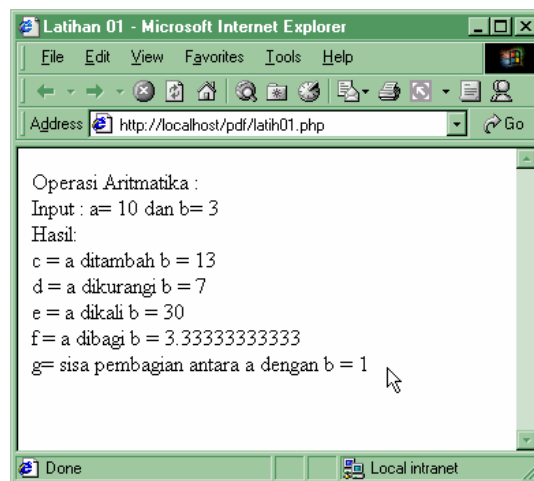
```
<html>
<head>
<title>Latihan 01
</title>
</head>
<body>
Operasi Aritmateka :<br>
<?
$a=10;
$b=3;
?>
Input : a= <? echo $a; ?> dan b= <? echo $b; ?>
<br>
<?
$c=$a+$b;
$d=$a-$b;
$e=$a*$b;
$f=$a/$b;
$g=$a%$b;
?>
Hasil:<br>
c = a ditambah b = <? echo $c; ?>
<br>
d = a dikurangi b = <? echo $d; ?>
<br>
e = a dikali b = <? echo $e; ?>
```

```

<br>
f = a dibagi b = <? echo $f; ?>
<br>
g= sisa pembagian antara a dengan b = <? echo $g; ?>
<br>
</body>
</html>

```

Dari listing tersebut, Anda akan melihat hasil seperti pada gambar di bawah ini. Nilai dari masing-masing variabel telah terisi sesuai dengan operasi aritmateka yang telah dibuat.



Gambar 2.1 Hasil operasi aritmateka terhadap dua variabel

2.1.2 Penggunaan Operasi Boolean

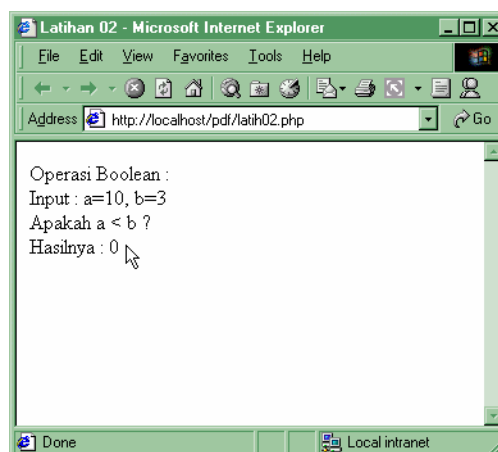
Boolean merupakan variabel yang sering dipakai dalam pemrograman untuk menghasilkan suatu keputusan dalam menjalankan perintah bercabang. Di bawah ini adalah contoh penggunaan operasi boolean dalam PHP. Dari contoh tersebut, dapat kita lihat bahwa cek merupakan variabel boolean yang akan menghasilkan nilai true (benar) atau bernilai 1 (satu) jika nilai a lebih kecil dari b. Sebaliknya, boolean akan menghasilkan nilai false (salah) atau nilai 0 (nol), jika ternyata nilai a ternyata tidak lebih kecil dari b.

Contoh:

Pernyataan	Hasil Boolean	Nilai Integer
10 < 2	false (salah)	0
5 < 10	true (betul)	1

latih02.php

```
<html>
<head>
<title>Latihan 02
</title>
</head>
<body>
Operasi Boolean :<br>
<?
$a=10;
$b=3;
?>
Input : a=<? echo $a; ?>, b=<? echo $b; ?>
<br>
Apakah a < b ?
<br>
<?
$cek=(int) ($a<$b) ;
?>
Hasilnya : <? echo $cek; ?>
<br>
</body>
</html>
```



Gambar 2.2 Operasi boolean yang menghasilkan nilai true atau false

Oleh karena nilai a adalah 10 dan b adalah 3, tentu saja pernyataan a lebih kecil dari b adalah salah sehingga variabel cek akan menghasilkan nilai false dan apabila dikonversi ke nilai interger akan menghasilkan angka 0. Hasil tersebut kemudian akan dicetak dengan menggunakan perintah `<?php echo $cek; ?>`. Hasil eksekusi script di atas dapat dilihat pada Gambar 2.2.

2.1.3 Penggabungan Dua String

Jika Anda ingin untuk menggabungkan dua kata atau kalimat yang berbeda menjadi satu kesatuan, Anda dapat menggunakan tanda titik di antara dua string. Hal ini akan banyak berguna apabila string yang dihasilkan berasal dari sumber data yang berbeda. Contoh lain dalam pemanfaatan penggabungan string dapat Anda lihat pada `latih04.php`.

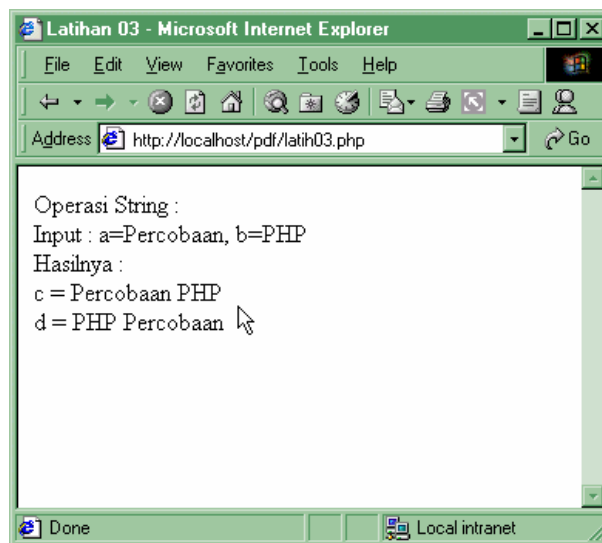
Contoh:

`a = "Menyanyi" . "dan" . "Menari"`, sama hasilnya dengan `a="MenyanyidanMenari"`

`latih03.php`

```
<html>
<head>
<title>Latihan 03
</title>
</head>
<body>
Operasi String :<br>
<?
$a="Percobaan";
$b="PHP";
?>
Input : a=<? echo $a; ?>, b=<? echo $b; ?>
<br>
<?
$c=$a . " " . $b;
$d=$b . " " . $a;
?>
Hasilnya : <br>
c = <? echo $c; ?>
<br>
d = <? echo $d; ?>
<br>
</body>
</html>
```

Kedua string pada program di atas akan digabungkan menjadi satu dengan penambahan spasi di dalamnya sehingga akan dihasilkan kalimat “Percobaan PHP” dan “PHP Percobaan”.



Gambar 2.3 Penggabungan dua kata menjadi satu

2.1.4 Mencetak Tag HTML di dalam PHP

Jika sebelumnya kita hanya mencetak nilai dari suatu variabel dengan PHP yang disisipkan ketika diperlukan, pada listing berikut ini kita akan mencetak isi HTML menggunakan perintah echo. Bukan hanya nilai variabel saja yang dapat dicetak, tetapi juga tag-tag HTML di dalamnya karena pada dasarnya echo berfungsi untuk mencetak.

latih04.php

```
<html>
<head>
<title>Latihan 04
</title>
</head>
<body>
<?
$a=10;
```



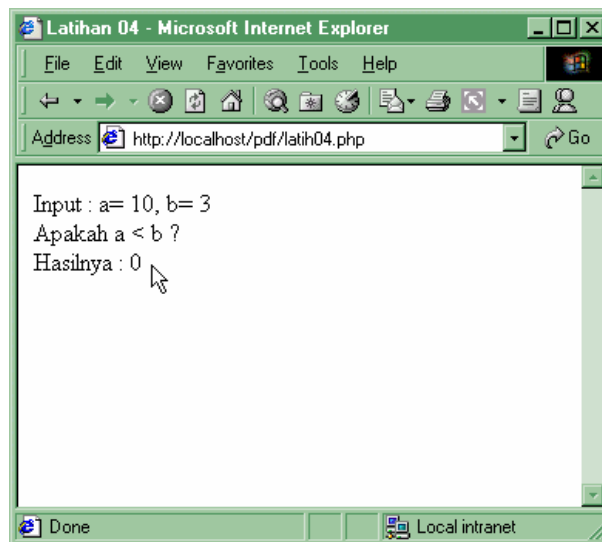
```

$b=3;
echo "Input : a= " . $a . ", b= " . $b;
echo "<br>";
echo "Apakah a < b ?";
echo "<br>";
$cek = (int) ($a<$b);
echo "Hasilnya : ";
echo $cek;
echo "<br>";
?>
</body>
</html>

```

Listing program di atas sebetulnya sama dengan listing latih02.php dengan penggunaan echo untuk mencetak beberapa tag di dalamnya. Perbedaan antara latih02.php dan latih04.php terletak pada pencetakan tag-tag HTML yang bergabung menjadi satu di dalam tag PHP `<?>` dan `?>`.

Tampilan hasil eksekusi program di atas dapat dilihat pada Gambar 2.4:



Gambar 2.4 Mencetak ke Layar Browser

2.1.5 Pengujian Kondisi Bersyarat

Seringkali hasil yang diperoleh dari sebuah halaman web yang satu dengan yang lain dapat berbeda. Hal ini disebabkan karena alur program terdiri dari beberapa alternatif yang ditentukan oleh beberapa kondisi tertentu. Untuk mengatur arah dari alur percabangan suatu program dibutuhkan uji kondisi yang nantinya dapat menentukan kemana program akan berjalan.

Terdapat 2 (dua) cara dalam melakukan pengujian kondisi pada program, yaitu menggunakan if dan switch-case. Keduanya mempunyai fungsi yang sama meskipun penggunaannya tergantung pada selera dan kebutuhan. Bila dibandingkan dengan penggunaan if, hasil yang akan diperoleh dalam pengujian kondisi switch-case lebih dapat diprediksi secara pasti. Jika Anda menggunakan if, hasil yang diperoleh cenderung merupakan hasil pengujian nilai benar dan salah pada suatu pernyataan yang Anda buat.

Di bawah ini terdapat contoh penggunaan if untuk pengujian kondisi terhadap pernyataan $a < b$. Jika pernyataan benar, pada layar browser akan tampil teks “Betul!!!”. Sebaliknya jika salah, pada layar browser akan tampil teks “Salah donk...”.

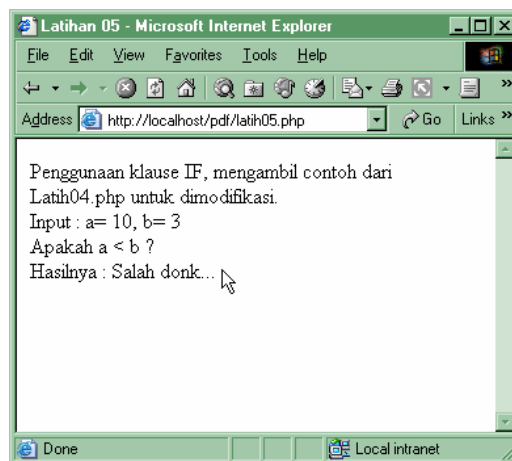
latih05.php

```
<html>
<head>
<title>Latihan 05
</title>
</head>
<body>
Penggunaan klausa IF, mengambil contoh dari
Latih04.php untuk dimodifikasi.<br>
<?
$a=10;
$b=3;
echo "Input : a= " . $a . ", b= " . $b;
echo "<br>";
echo "Apakah a < b ?";
echo "<br>";
echo "Hasilnya : ";
if ($a<$b) {
    echo "Betul!!!";
} else {
    echo "Salah donk...";
}
```

```
?>  
</body>  
</html>
```

Pernyataan yang benar ditulis setelah klausa `then` dan akan muncul jika kondisi yang disyaratkan (a lebih kecil dari b) dianggap sesuai. Pernyataan yang salah ditulis setelah klausa `else` dan akan muncul jika kondisi yang disyaratkan tidak sesuai.

Hasil eksekusi dari listing di atas dapat dilihat pada gambar di bawah ini:



Gambar 2.5 Penggunaan klausa IF

Seperti yang sudah disinggung sebelumnya, pada penggunaan `switch` hasil dari suatu pernyataan sudah dapat diprediksi sebelumnya. Namun demikian, seringkali tidak semua hasil bisa sesuai dengan prediksi. Apabila hasil yang diperoleh dari uji kondisi tidak sesuai dengan prediksi, maka selanjutnya akan ditangani oleh satu pernyataan yang ditulis dalam default.

latih06.php

```
<html>  
<head>  
<title>Latihan 06  
</title>  
</head>  
<body>
```

```

Penggunaan switch dan Case<br>
<?
$a=10;
$b=3;
echo "Jika 10 dibagi 3.. ";
$sisas=$a % $b;
switch($sisas){
  case 0: echo "tidak sisa";
    break;
  case 1: echo "sisa satu";
    break;
  case 2: echo "sisa dua";
    break;
  case 3: echo "sisa tiga";
    break;
  default: echo "sisa lebih dari tiga";
}
?>
</body>
</html>

```

Hasil eksekusi listing program latih06.php dapat dilihat pada gambar di bawah ini. Layar akan menampilkan pesan bahwa hasil perhitungan mempunyai sisa satu.



Gambar 2.6 Menggunakan switch-case

2.2 Membuat Perulangan

Perulangan sangat penting untuk membuat pernyataan yang berulang-ulang dalam suatu program. Jumlah perulangan dapat diatur sesuai jumlah tertentu maupun sesuai dengan kondisi tertentu. Apabila jumlah perulangan sudah dirancang atau diketahui sebelumnya, Anda dapat memastikannya dalam bentuk nilai angka. Namun demikian, jika jumlah perulangan belum diketahui secara pasti dan didasarkan pada suatu pengujian kondisi maka Anda dapat menggunakan uji kondisi dalam nilai akhirnya.

Terdapat beberapa cara dalam membuat perulangan, di antaranya adalah penggunaan for, while, dan do...while. Pada contoh program di bawah ini, jumlah perulangan sudah dipastikan akan terjadi lima kali.

Contoh:

```
for ($a=1;$a<=5;$a++) {  
    echo "percobaan";  
}
```

Dari potongan program di atas, pernyataan yang berada di dalam pernyataan for..next akan diulang sampai 5 (lima) kali. Lima kali dihitung dari awal=1 dan akhir=5 dengan penambahan a sebanyak 1 pada setiap perulangannya (\$a++). Jika Anda menginginkan penambahan lebih dari 1 (satu) angka, contoh di bawah ini dapat memberikan gambaran kepada Anda. Nilai variabel a akan bertambah atau meloncat 2 (dua) angka dari sebelumnya (\$a+=2) sehingga teks percobaan yang tercetak hanya 5 (lima) kali saja.

Contoh:

```
for ($a=1;$a<=10;$a+=2) {  
    echo "percobaan";  
}
```

latih07.php

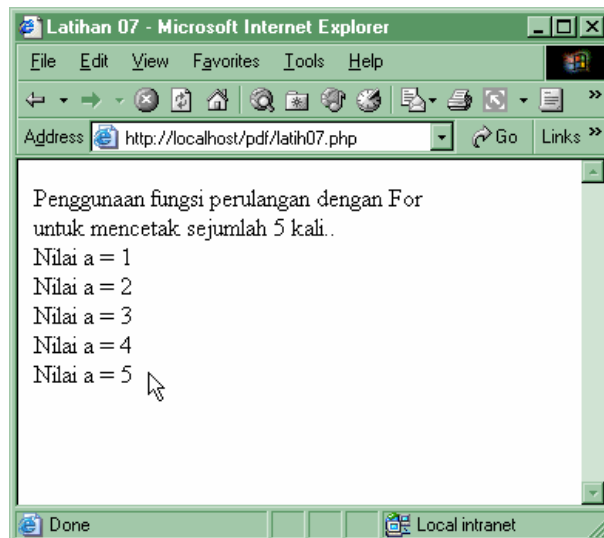
```
<html>  
<head>  
<title>Latihan 07  
</title>  
</head>  
<body>  
Penggunaan fungsi perulangan dengan For<br>
```

```

untuk mencetak sejumlah 5 kali..<br>
<?
for ($a=1;$a<=5;$a++) {
    echo "Nilai a = $a";
    echo "<br>";
}
?>
</body>
</html>

```

Hasil dari eksekusi program di atas akan menampilkan nilai a pada setiap prosesnya. Program dapat dikatakan berjalan benar jika tampil seperti gambar di bawah ini.



Gambar 2.7 Membuat perulangan dengan for

Selanjutnya Anda bisa membuat perulangan dengan menggunakan while. Jika menggunakan while maka selama kondisi yang diujikan masih benar program akan berjalan terus.

Contoh:

```

$a=1;
while ($a<=5) {
    echo "Percobaan";
    $a++;
}

```

Contoh di atas dapat diartikan sebagai perintah pencetakan teks Percobaan selama a masih bernilai di bawah atau sama dengan 5 ($a \leq 5$). Jika nilai a terus bertambah dan melebihi angka 5 maka pencetakan teks Percobaan akan berhenti.

latih08.php

```
<html>
<head>
<title>Latihan 08
</title>
</head>
<body>
Penggunaan fungsi perulangan dengan while<br>
untuk mencetak sejumlah 5 kali..  
<?
$a=1;
while ($a<=5) {
    echo "Nilai a = $a";
    echo "<br>";
    $a++;
}
?>
</body>
</html>
```



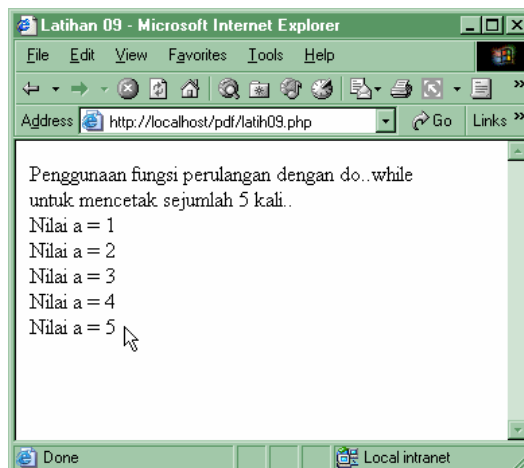
Gambar 2.8 Membuat perulangan selama kondisi masih true

Latihan ke-8 sama hasilnya dengan latihan sebelumnya. Perbedaan yang mendasar terletak pada uji kondisi yang berada di kurung while. Jumlah perulangan didasarkan pada nilai kebenaran dari pengujian tersebut.

Bentuk perulangan berikut ini juga akan menampilkan hasil yang tidak berbeda dengan listing sebelumnya. Penggunaan `do...while` sebetulnya tidak jauh berbeda dengan penggunaan `while` pada `latih08.php`. Perbedaannya, perulangan dengan `do...while` akan mengakibatkan pengujian dilakukan setelah pernyataan di atasnya dieksekusi terlebih dahulu, berbeda dengan `latih08` yang menguji kondisi terlebih dahulu sebelum menjalankan pernyataan-pernyataan di dalamnya.

`latih09.php`

```
<html>
<head>
<title>Latihan 09
</title>
</head>
<body>
Penggunaan fungsi perulangan dengan do<br>
untuk mencetak sejumlah 5 kali..<br>
<?
$a=1;
do {
    echo "Nilai a = $a";
    echo "<br>";
    $a++;
} while ($a<=5)
?>
</body>
</html>
```



Gambar 2.9 Membuat perulangan dengan `do...while`

Pada Gambar 2.9 dapat dilihat bahwa dengan ketiga cara yang berbeda hasil perulangan dapat dibuat serupa. Anda perlu memperhatikan logika masing-masing perulangan. Jika logika sudah sesuai, program dapat berjalan sebagaimana mestinya.

2.2.1 Membuat Array

Seperti halnya dengan setiap bahasa pemrograman, PHP juga mendukung pembuatan variabel bertipe array. Dengan adanya indeks penomoran pada variabel, Anda dapat memilih elemen dari sebuah array. Contoh pendeklarasian array adalah sebagai berikut:

```
$n=array(1, 2, 3, 4, 5);
```

atau

```
$n[0]=1;  
$n[1]=2;  
$n[2]=3;  
$n[3]=4;  
$n[4]=5;
```

Deklarasi di atas akan mengalokasikan variabel array `n` yang bertipe string sebanyak 5 elemen. Perlu diketahui, bahwa indeks array dimulai dari urutan ke-0 sehingga indeks terakhir dari array berakhir pada urutan ke-4.

Untuk menghitung panjang dari array, Anda dapat membuat satu variabel, misalkan `maks` yang diisi dengan nilai `count($n)` sehingga `maks` akan mempunyai nilai dari panjang array `n`. Perintah `count` akan menghasilkan nilai maksimum dari array yang bersangkutan sehingga akan memudahkan kita untuk mengetahui panjang array saat itu.

```
$maks = count($n);
```

latih10.php

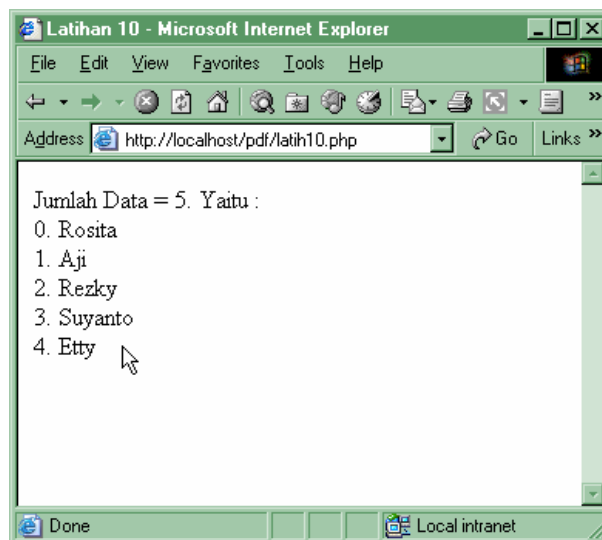
```
<html>  
<head>  
<title>Latihan 10  
</title>  
</head>  
<body>  
<?  
$n[0]="Rosita";  
$n[1]="Aji";
```

```

$n[2]="Rezky";
$n[3]="Suyanto";
$n[4]="Etty";
$maks = count($n);
echo "Jumlah Data = $maks. Yaitu :<br>";
for ($i=0;$i<$maks;$i++) {
    echo "$i. " . $n[$i] . "<br>";
}
?>
</body>
</html>

```

Dari Gambar 2.10, dapat dilihat bahwa dari alokasi array sebanyak 5 elemen dihasilkan index dari 0 sampai 4. Total dari index 0 sampai 4 adalah 5 elemen.



Gambar 2.10 Bekerja dengan array

2.2.2 Membuat Fungsi

Dalam membuat program, seringkali dibutuhkan pernyataan-pernyataan yang sebelumnya sudah ada di bagian lain. Agar kerja program dapat lebih baik, pernyataan-pernyataan yang sering dipakai dapat dibentuk menjadi fungsi sehingga sewaktu-waktu dapat dipanggil sesuai dengan kebutuhan.

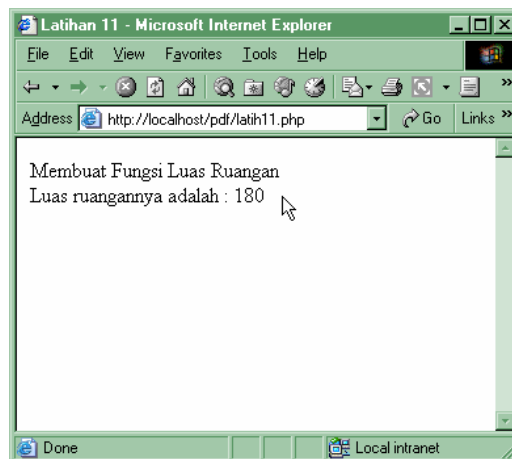
Contoh:

```
function luasRuang($Panjang,$Lebar) {  
    return $Panjang*$Lebar;  
}
```

Dari contoh di atas, dapat dilihat bahwa apabila fungsi luasRuang dipanggil, nilai yang dikembalikan adalah nilai dari Panjang dikalikan Lebar. Penulisan fungsi dapat disertai parameter jika diperlukan untuk membantu proses di dalam function. Penulisan parameter berada di belakang nama fungsi.

latih11.php

```
<html>  
<head>  
<title>Latihan 11  
</title>  
</head>  
<body>  
Membuat Fungsi Luas Ruangan<br>  
<?  
function luasRuang($Panjang,$Lebar) {  
    return $Panjang*$Lebar;  
}  
echo "Luas ruangnya adalah : ";  
echo luasRuang(12,15) ;  
?>  
</body>  
</html>
```



Gambar 2.11 Membuat fungsi luas ruang

Perintah `echo luasRuang(12,15)` akan menghasilkan keluaran berupa perkalian antara 12 dan 15. Pada Gambar 2.11, Anda dapat melihat nilai keluaran yang dihasilkan adalah 180 yang merupakan proses perkalian kedua nilai yang dimasukkan dalam fungsi.

2.2.3 Membuat Prosedur

Prosedur merupakan kumpulan beberapa pernyataan yang dapat digunakan untuk kepentingan tertentu. Sama seperti fungsi, prosedur dapat dipanggil sewaktu-waktu oleh program tergantung dari kebutuhan. Pembuatan prosedur cukup membantu jika proses di dalamnya cukup sering digunakan.

Terdapat perbedaan yang mendasar antara fungsi dan prosedur. Fungsi menghasilkan keluaran berupa nilai tertentu yang dapat digunakan atau dicetak, sedangkan prosedur tidak. Prosedur menghasilkan keluaran berupa hasil proses pernyataan-pernyataan yang berada di dalamnya. Pemrograman prosedural merupakan teknik yang cukup populer dalam pemrograman terstruktur. Programmer tidak harus mengulang banyak pernyataan jika memang bisa ditulis menjadi sebuah prosedur. Format dasar penulisan prosedur adalah seperti di bawah ini:

```
<?
function nama_prosedur() {
    pernyataan-pernyataan
}
?>
```

Untuk menjalankan atau memanggil kembali prosedur di atas, Anda hanya perlu menyebutkan `nama_prosedur`:

```
<?
nama_prosedur();
?>
```

Contoh prosedur yang dibuat pada listing berikut ini berguna untuk menampilkan nama-nama hari berdasarkan tanggal tertentu yang ditulis dalam parameter. Jika parameter diisi dengan tanggal hari ini (`time()`) maka yang akan tampil adalah nama hari ini.

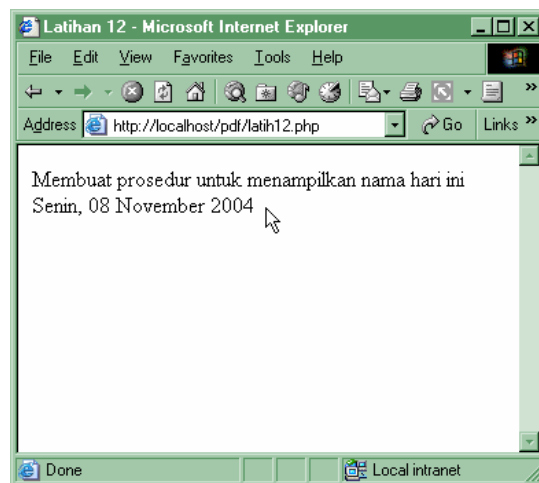
```
<html>
<head>
<title>Latihan 12
```

```

</title>
</head>
<body>
Membuat prosedur untuk menampilkan nama hari ini<br>
<?
function tampilhari($waktu) {
    $hari=date("w",$waktu);
    switch($hari) {
        case 0: echo "Minggu";
            break;
        case 1: echo ("Senin");
            break;
        case 2: echo ("Selasa");
            break;
        case 3: echo ("Rabu");
            break;
        case 4: echo ("Kamis");
            break;
        case 5: echo ("Jumat");
            break;
        case 6: echo ("Sabtu");
            break;
    }
}

tampilhari(time());
echo " , " . date("d F Y", time());
?>
</body>
</html>

```



Gambar 2.12 Prosedur untuk menampilkan nama hari

Perintah `tampilhari(time())`; digunakan untuk memanggil prosedur `tampilhari`. Oleh karena parameter diisi dengan `time()`, nama hari yang akan muncul akan sesuai dengan tanggal saat ini. Perintah `date("w", $waktu)` digunakan untuk menghasilkan angka 0 sampai dengan 6 untuk mewakili hari Minggu sampai dengan Sabtu.

2.2.4 Berpindah ke Halaman Lain

Kebutuhan untuk berpindah dari satu halaman ke halaman lain secara otomatis tanpa intervensi dari pengunjung kadang diperlukan untuk mempercepat proses dan demi alasan keamanan. Untuk itu dibutuhkan perintah untuk meminta browser web untuk langsung berpindah ke halaman lain sesuai dengan alamat yang dituju. Terdapat 2 (dua) alternatif teknik untuk berpindah ke halaman lain, yaitu dengan perintah:

```
echo "<META http-equiv='refresh' content='0; url=alamat_lain'>";
```

atau

```
header ("Location: alamat_lain ");
```

Jika `alamat_lain` yang dituju berupa alamat situs lain, Anda cukup menuliskan name-address atau IP-address lengkap dengan http-nya.

Contoh:

```
echo "<META http-equiv='refresh' content='0; url=http://www.yahoo.com'>";
```

atau

```
header ("Location: http://www.yahoo.com");
```

Namun demikian apabila lokasi file berada dalam satu direktori (satu lokasi) maka `alamat_yang_dituju` cukup ditulis sesuai dengan nama halaman dan ekstensinya saja. Untuk lokasi file yang tidak berada dalam satu direktori, Anda harus mengetikkan lokasi tersebut dengan pedoman direktori saat ini. Inilah yang disebut dengan alamat relatif, alamat file yang ditulis didasarkan pada direktori halaman utama yang akan digunakan untuk berpindah ke halaman lain. Untuk lebih jelasnya, mari kita lihat ilustrasi di bawah ini.



Gambar 2.13 Pohon Direktori

Pada Gambar 2.13, diperlihatkan 4 (empat) direktori/folder yang akan menjadi tempat tujuan untuk berpindah. Dimisalkan nama file tujuan adalah file.php sedangkan dokumen PHP yang berisi perintah untuk berpindah halaman mempunyai lokasi awal di direktori latihan. Dengan empat tujuan, dapat dihasilkan kombinasi sebagai berikut:

Direktori/ Folder Asal	Direktori/Folder Tujuan	Penulisan Perintah
latihan	latihan	header("Location: file.php");
latihan	webfiles	header("Location: ../file.php");
latihan	proyek	header("Location: ../proyek/file.php");
latihan	images	header("Location: images/file.php");

Untuk lebih jelasnya, di bawah ini terdapat contoh program yang menggunakan perintah di atas untuk berpindah ke file latih12.php. Listing latih13.php berlokasi di direktori yang sama dengan latih12.php.

latih13.php

```

<?
    header ("Location: latih12.php";
?>
<html>
<head>
<title>Latihan 13
</title>
</head>

```

```
<body>  
</body>  
</html>
```

Hasilnya dapat dilihat seperti di bawah ini. Pada Gambar 2.14 tampak bahwa meskipun pada kotak alamat (address) tertulis latih13.php, isi dan judul dokumennya adalah latih12.php (dapat dilihat pada judul yang muncul pada browser).



Gambar 2.14 Meskipun alamat tertulis latih13.php, namun judul dan isi milik latih12.php

2.2.5 Menggabungkan Halaman Lain

Seringkali beberapa halaman web dapat Anda gabungkan menjadi satu untuk alasan efisiensi, contohnya penggunaan template halaman web atau penggunaan prosedur-prosedur. Anda tidak perlu menuliskannya lagi pada setiap halaman untuk menampilkan header web yang sama. Header web cukup dibuat menjadi satu file, kemudian digabungkan pada setiap halaman yang membutuhkan dengan memanfaatkan perintah include. Anda juga tidak perlu menuliskan prosedur yang sama pada tiap

halaman web jika prosedur tersebut dibuat dalam satu file tersendiri kemudian digabungkan dengan include. Struktur penulisan untuk include adalah :

```
<?
include "nama_file";
?>
```

Seperti halnya dengan perintah perpindahan halaman, aturan penulisan nama_file pada include juga menggunakan aturan penulisan alamat relatif. Akibatnya, penulisan lokasi dimulai dari halaman PHP utama yang akan menggabungkan halaman lain. Sebagai contoh, mari kita lihat ilustrasi di bawah ini:



Gambar 2.15 Pohon Direktori

Apabila lokasi file PHP yang akan menggabungkan file lain berada di direktori latihan, akan terdapat 4 (empat) lokasi file yang bisa digabungkan berdasarkan Gambar 1.19. Dengan empat tujuan, dapat dihasilkan kombinasi sebagai berikut:

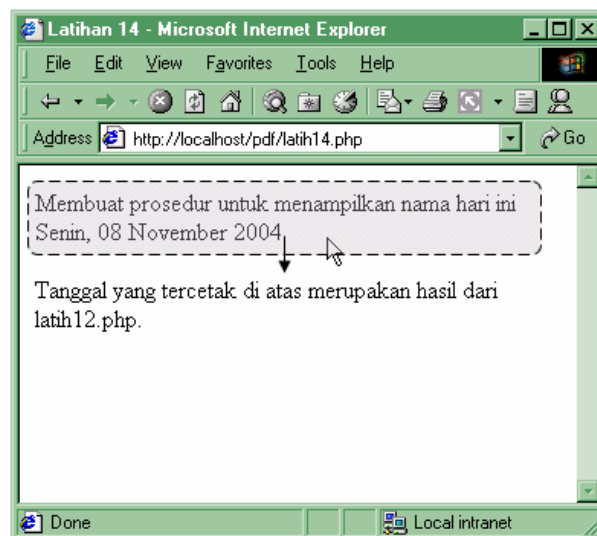
Direktori/Folder Halaman Utama	Direktori/Folder halaman yang akan digabung	Penulisan Include File
latihan	latihan	include file "file.php";
latihan	webfiles	include file "../file.php";
latihan	proyek	include file "../proyek/file.php";
latihan	Images	include file "images/file.php";

Sebagai contoh penggunaan include, kita dapat menggabungkan file latih12.php dengan latih14.php:

latih14.php

```
<head>
<title>Latihan 14
</title>
</head>
<body>
<?
include "latih12.php";
?>
<p>
Tanggal yang tercetak di atas merupakan hasil dari latih12.php.
</body>
</html>
```

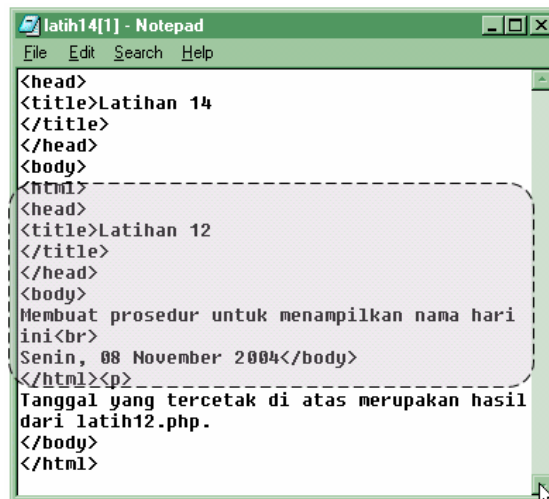
Hasil yang akan diperoleh dari eksekusi latih14.php dapat dilihat pada Gambar 2.16. Kode hasil penggabungan file tersebut dapat dilihat melalui menu View | Source (Gambar 2.17).



Gambar 2.16 Latih14.php akan memunculkan teks dari file Latih12.php

Pada Gambar 2.17 terlihat bahwa kode-kode yang berada di dalam latih12.php sudah menyatu dengan latih14.php. Lokasi penyisipan

juga tergantung pada lokasi di mana Anda meletakkan perintah include dalam sebuah file.



```
<head>
<title>Latihan 14
</title>
</head>
<body>
<include src="latih12.php">
</include>
</body>
</html>
```

Membuat prosedur untuk menampilkan nama hari ini

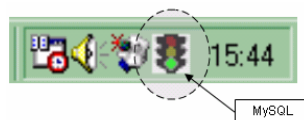
Senin, 08 November 2004</body>
</html>

Gambar 2.17 Kode hasil gabungan antara latih14.php dan latih12.php

Pembahasan di atas beserta contohnya merupakan dasar dari pemrograman PHP. Banyak perintah dan fungsi lain yang mungkin kita bisa temukan guna membuat berbagai macam script dasar untuk kepentingan percobaan. Selanjutnya, kita akan membahas pembuatan database untuk kepentingan koneksi database dengan PHP.

2.3 MySQL 5

Pertama-tama, pastikan bahwa server MySQL sudah aktif melalui tray icon di sebelah kanan bawah layar Windows.



Gambar 2.18 Tray icon MySQL sudah muncul

Jika belum, maka jalankan winmysqladmin.exe atau mysqld-opt.exe di direktori c:\mysql\bin melalui Command Prompt. Untuk sistem operasi Linux, gunakan perintah berikut ini melalui terminal/console:

```
# /etc/init.d/mysql start
```

2.3.1 Koneksi Pertama ke Server MySQL

Untuk bisa masuk ke server MySQL diperlukan otorisasi berupa username dan password. Jika server MySQL baru pertama kali digunakan setelah instalasi, hanya user root yang bisa masuk ke dalam server karena memang user tersebutlah yang pertama kali terdaftar dalam server MySQL (default). Pertama kali masuk ke dalam server, Anda tidak akan ditanya mengenai password koneksi ke server. Password root saat itu masih dalam keadaan kosong. Namun berhati-hatilah, jangan pernah menciptakan peluang bagi orang lain untuk dapat masuk ke dalam server! Anda harus menambahkan password untuk root setelah berhasil masuk ke dalam server agar tidak seorang pun selain Anda, yang bisa login dengan hak yang tidak terbatas. Jika orang lain mempunyai hak untuk login ke server sebagai root, ia akan bebas mengatur siapa saja yang berhak masuk dan hak yang dimiliki oleh masing-masing user selain pengaturan hak akses terhadap database. Dapat disimpulkan bahwa user root berfungsi sebagai administrator server MySQL.

Perintah untuk masuk ke server MySQL dengan user root adalah sebagai berikut :

```
# mysql -h localhost -u root
```

localhost menyatakan bahwa komputer yang sedang kita jadikan sasaran untuk koneksi ke server MySQL adalah komputer server itu sendiri (lokal). Jika Anda melakukan koneksi di luar komputer server, isian paramater localhost harus diganti dengan alamat IP dari komputer server. Perintah di atas hanya dapat dilakukan jika user belum dilindungi password. Jika telah dilindungi password, perintah yang harus dilakukan adalah:

```
# mysql -h localhost -u root -p
```

Jika atribut `-p` tidak ditulis sedangkan user sudah diberi password untuk masuk ke server maka koneksi akan ditolak. Sebaliknya, bila atribut `-p` ditambahkan pada perintah di atas maka akan ditanyakan password untuk koneksi ke server. Jika password yang dimasukkan cocok, user dapat masuk ke server MySQL. Password yang dimasukkan tidak akan terlihat oleh user lain karena tanda asterik (*) menyembunyikan penulisan password itu sendiri. Berikut ini hasil eksekusi perintah koneksi untuk user dengan password:

```
# mysql -h localhost -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 5.0.0-alpha

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> _
```

Password dapat saja ditulis langsung di belakang parameter `-p` tanpa spasi, namun ini tidak disarankan mengingat orang lain dapat langsung melihat password yang Anda ketik.

```
# mysql -u root -ppwd001
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 5.0.0-alpha

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> _
```

2.3.2 Koneksi Menggunakan User Selain Root

Selain root sebagai administrator server, user dapat masuk ke server dengan nama yang lain bergantung pada isi tabel user di database MySQL. Dalam tabel user, semua nama beserta hak akses ke server telah diatur sehingga keberadaan tabel user sangat vital. Untuk membuat user baru atau mengubah hak akses ke server, orang tersebut harus mempunyai hak untuk mengakses tabel user di database MySQL. Dengan kenyataan tersebut, jangan pernah memberikan hak untuk mengakses database MySQL kepada orang lain karena di situlah pusat keamanan server MySQL.

Jika user selain root sudah terbuat, perintah untuk masuk ke dalam server bagi user yang baru tidak jauh berbeda, hanya saja

atribut `-u` yang semula diisi dengan nama root diganti dengan nama user yang telah dibuat.

Struktur penulisan:

```
mysql -h alamat_host -u user -p
```

```
# mysql -h localhost -u sanjaya -p
```

Catatan:

Tidak setiap user mempunyai password dalam koneksi ke server (meskipun disarankan untuk memberikan password kepada setiap user-nya). Jika user tidak mempunyai password, maka tidak diperlukan `-p` dalam penulisan di atas.

2.3.3 Keluar dari Server

Untuk keluar dari server, user cukup menuliskan perintah `quit`, maka user akan dibawa kembali ke prompt masing-masing sistem operasi.

```
mysql> quit  
Bye  
#_
```

2.3.4 Membuat User Baru

Sebagai seorang administator server MySQL (root), Anda mempunyai hak untuk membuat user baru beserta hak akses di dalam server. Seperti yang telah disinggung sebelumnya, pembuatan user dapat dilakukan jika Anda mempunyai hak untuk mengakses tabel user yang ada di database MySQL. Untuk itu, login-lah ke server terlebih dahulu sebagai root, kemudian panggil database MySQL.

```
# mysql -u root  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 1 to server version: 5.0.0-alpha  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql> use mysql;
```

Database MySQL mempunyai table user dengan struktur sebagai berikut:

Field	Type	Null	Key	Default	Extra
host	char(60)		PRI		
user	char(16)		PRI		
password	char(16)				
select_priv	enum('N','Y')			N	
insert_priv	enum('N','Y')			N	
update_priv	enum('N','Y')			N	
delete_priv	enum('N','Y')			N	
create_priv	enum('N','Y')			N	
drop_priv	enum('N','Y')			N	
reload_priv	enum('N','Y')			N	
shutdown_priv	enum('N','Y')			N	
process_priv	enum('N','Y')			N	
file_priv	enum('N','Y')			N	
grant_priv	enum('N','Y')			N	
references_priv	enum('N','Y')			N	
index_priv	enum('N','Y')			N	
alter_priv	enum('N','Y')			N	
show_db_priv	enum('N','Y')			N	
super_priv	enum('N','Y')			N	
create_tmp_table_priv	enum('N','Y')			N	
lock_tables_priv	enum('N','Y')			N	
execute_priv	enum('N','Y')			N	
repl_slave_priv	enum('N','Y')			N	
repl_client_priv	enum('N','Y')			N	
ssl_type	enum('', 'ANY', 'X509', 'SPECIFIC')				

	ED')				
ssl_cipher	blob				
x509_issuer	blob				
x509_subject	blob				
max_questions	int(11) unsigned			0	
max_updates	int(11) unsigned			0	
max_connections	int(11) unsigned			0	

Pada MySQL versi sebelumnya, table user hanya mempunyai 17 field. Akan tetapi pada MySQL versi 5 jumlah field berkembang sampai sejumlah 31 field di dalam tabel user dengan 28 field di antaranya digunakan untuk mengatur hak akses (privileges). Tiga field lainnya digunakan untuk pembatasan host dari komputer yang digunakan sebagai akses server, nama user, dan password. Untuk membuat user baru dengan nama sanjaya dan mempunyai password pwd001, perintah yang harus ditulis adalah:

```
mysql> insert into user (host,user,password) values
-> ('localhost','sanjaya',password('pwd001'));
mysql> flush privileges;
```

Dari hasil pernyataan di atas, terbuat satu user baru yang dapat mengakses server MySQL dengan nama sanjaya dan password pwd001. User ini hanya dapat mengakses database dari komputer server saja (localhost), namun tidak mempunyai hak apapun terhadap perintah-perintah SQL di server database. Agar user mempunyai hak membaca (select), hak tersebut dapat diperbarui (update) melalui perintah:

```
mysql> update user set select_priv='Y' where
-> user='sanjaya';
mysql> flush privileges;
```

Hak tersebut langsung ditambahkan melalui perintah:

```
mysql> insert into user (host,user,password,select_priv)
-> values('localhost','sanjaya',
-> password('pwd001'),'Y');
mysql> flush privileges;
```


Selain `select_priv`, Anda dapat saja menambahkan hak lain yang terdaftar pada tabel, terutama untuk hak-hak yang mendasar untuk pengelola web seperti `update_priv`, `insert_priv`, dan `delete_priv`.

2.3.5 Mengubah Host

Posisi user dalam mengakses server MySQL dapat saja diperluas atau dipersempit. Jika host diisi dengan “%” maka user mempunyai hak yang sangat luas dalam lokasi akses. User dapat berpindah dan menggunakan komputer apapun untuk login ke server. Akan tetapi jika host diisi dengan `localhost` atau IP tertentu seperti `192.168.1.1` maka user mempunyai hak yang sangat sempit, yaitu hanya pada komputer dengan alamat IP yang dimaksudkan. Untuk memperluas host, lakukan:

```
mysql> update user set host='%'
      -> where user='sanjaya';
mysql> flush privileges;
```

Jika Anda ingin mempersempit hak akses `sanjaya` ke server MySQL hanya dari komputer dengan IP `192.168.1.1` saja, gunakan perintah berikut:

```
mysql> update user set host='192.168.1.1'
      -> where user='sanjaya';
mysql> flush privileges;
```

2.3.6 Mengubah Nama User

Selain host, Anda juga dapat mengubah nama user yang salah atau tidak diinginkan. Di bawah ini perintah untuk mengganti nama user `sanjaya` menjadi `ridwan`.

```
mysql> update user set user='sanjaya'
      -> where user='ridwan';
mysql> flush privileges;
```

2.3.7 Mengubah Password

Jika Anda ingin mengubah password karena lupa atau pun sebagai antisipasi keamanan, berikut ini cara untuk mengubahnya:

```
mysql> update user set password=password('pwd002')
      -> where user='ridwan';
mysql> flush privileges;
```

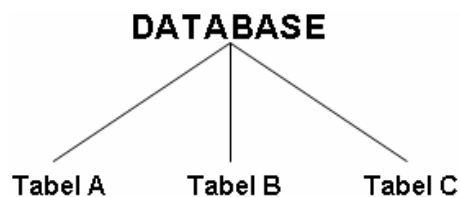
2.3.8 Menghapus User

Apabila user sudah tidak dipakai, sebaiknya nama user dibuang untuk memudahkan administrasi server database. Perintah untuk menghapus user ridwan adalah:

```
mysql> delete from user  
-> where user='ridwan';  
mysql> flush privileges;
```

2.3.9 Mengelola Database

Database merupakan kumpulan dari beberapa tabel sehingga untuk dapat membuat tabel, user harus membuat database lebih dahulu sebagai “induk” yang memayungi tabel-tabel di bawahnya.



Gambar 2.19 Database adalah “payung” dari tabel di dalamnya

2.3.10 Membuat Database

Dalam pembuatan database, perhatikan bahwa dalam penulisan nama database tidak diperbolehkan untuk menggunakan spasi dan karakter nonstandar. Bentuk penulisan perintah untuk membuat database baru adalah:

```
create database nama_database;
```

Berikut ini adalah perintah untuk membuat database latihan:

```
mysql> create database latihan;
```

Catatan:

- ☒ Jangan menggunakan spasi dan karakter khusus lainnya dalam penulisan nama database.
- ☒ Jangan menggunakan angka pada awal nama database.

2.3.11 Melihat Keseluruhan Database

Perintah untuk melihat keseluruhan database yang ada dalam server adalah sebagai berikut:

```
mysql> show databases;
+-----+
| Database |
+-----+
| latihan  |
| mysql   |
| test    |
+-----+
3 rows in set (0.04 sec)
```

2.3.12 Memanggil Database

Setelah database dibuat, untuk bisa menggunakannya diperlukan perintah use untuk memanggil database yang akan digunakan. Perintah ini harus selalu digunakan jika Anda ingin mengakses tabel-tabel yang ada di dalam database.

Struktur penulisan:

```
use nama_database;
```

Contoh:

```
mysql> use latihan;
```

Perintah di atas menyatakan pemanggilan terhadap database latihan. Jika database ada di server, akan muncul pemberitahuan bahwa database sudah digunakan. Namun demikian jika database tidak ditemukan maka akan muncul pesan kegagalan.

2.3.13 Menghapus Database

Untuk menghapus database tersebut, Anda dapat menggunakan perintah drop. Struktur Penulisan:

```
drop database nama_database;
```

Contoh:

```
mysql> drop database latihan;
```

2.3.14 Membuat Tabel

Dalam pembuatan tabel, perhatikan apakah database sudah dipanggil sebelumnya? Jika belum, panggil database tersebut atau buat database baru jika belum tersedia. Hal ini dilakukan agar tabel yang Anda buat mempunyai tempat dalam database. Tabel sebagai tempat penampungan data mempunyai 2 bagian utama, yaitu nama dan tipe data. Tipe data yang bisa digunakan dalam MySQL dapat Anda lihat dalam tabel di bawah ini.

Struktur penulisan:

```
create table nama_tabel (nama_field type, ...);
```

Contoh:

```
create table berita (tanggal date, berita text);  
create table produk (nama varchar(25), harga bigint);
```

Tipe Data	Keterangan
tinyint	Ukuran terkecil dari integer Signed range: -128 sampai dengan 127 Unsigned range: 0 sampai dengan 255
smallint	Ukuran kecil dari integer Signed range: -32768 sampai dengan 32767 Unsigned range: 0 sampai dengan 65535
mediumint	Ukuran menengah dari integer Signed range: -8388608 sampai dengan 8388607 Unsigned range: 0 sampai dengan 16777215
int, integer	Integer Signed range: -2147483648 sampai dengan 2147483647 Unsigned range: 0 sampai dengan

	4294967295
bigint	<p>Ukuran terbear dari integer</p> <p>Signed range: -9223372036854775808 sampai dengan 9223372036854775807</p> <p>Unsigned range: 0 sampai dengan 18446744073709551615</p>
float	<p>Presisi tunggal</p> <p>Jangkauan: -3.402823466E+38 sampai dengan -1.175494351E-38, 0, 1.175494351E-38 sampai dengan 3.402823466E+38</p>
double, double precision, real	<p>Presisi ganda</p> <p>Jangkauan: -1.7976931348623157E+308 sampai dengan -2.2250738585072014E-308, 0, dan 2.2250738585072014E-308 sampai dengan 1.7976931348623157E+308</p>
decimal, numeric	Pecahan dengan range seperti tipe data DOUBLE
date	<p>Tanggal</p> <p>Format: YYYY-MM-DD</p>
datetime	<p>Kombinasi tanggal dan jam</p> <p>Jangkauan: 1000-01-01 00:00:00 sampai dengan 9999-12-31 23:59:59</p> <p>Format: YYYY-MM-DD HH:MM:SS</p>
timestamp	<p>Kombinasi tanggal dan jam yang berisi waktu saat tabel diakses</p> <p>Jangkauan: 1970-01-01 00:00:00</p>

	sampai dengan tahun 2037 Format: YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD atau YYMMDD
time	Jam Jangkauan: -838:59:59 sampai dengan 838:59:59 Format: HH:MM:SS
year	Tahun Jangkauan: 1901 sampai dengan 2155 (4 digit), 1970-2069 (2 digit) Format: YYYY
Char	Karakter dengan panjang tetap sesuai saat pembuatan tabel, dengan karakter spasi dihilangkan pada saat penyimpanan Jangkauan: 1 sampai dengan 255 karakter
nchar, national char	Karakter dengan panjang tetap sesuai saat pembuatan tabel, namun karakter spasi tidak dihilangkan pada saat penyimpanan Jangkauan: 1 sampai dengan 255 karakter
varchar	Karakter dengan panjang sesuai saat panjang karakter yang ditulis dengan karakter spasi dihilangkan pada saat penyimpanan. Jangkauan: 1 sampai dengan 255 karakter
tinyblob	TEXT/BLOB dengan panjang maksimal

tinitext	255 karakter
mediumblob mediumtext	65535
longblob longtext	16777215
enum('value1','value2',...)	Obyek string yang hanya boleh diisi dari daftar pilihan value yang diberikan, NULL, atau error value Value maksimum: 65535
set ('value1','value2',...)	Obyek string yang hanya boleh diisi dari daftar pilihan value yang diberikan, NULL, atau error value Value maksimum: 64

Apabila tidak diisi dalam suatu field/kolom, maka nilai default dapat diberikan dengan:

```
mysql> use latihan;
Database changed
mysql> create table produk(kode varchar(25) not null
-> primary key, harga int default 0);
Query OK, 0 rows affected (0.17 sec)
```

Hasilnya adalah tabel produk dengan dua field/kolom. Kolom pertama, yaitu kode, mempunyai spesifikasi: tipe data varchar(25), harus diisi (not null), dan merupakan field kunci (primary key). harga mempunyai spesifikasi: tipe data integer dan jika tidak diisi maka secara otomatis (default) diisi dengan angka nol. Berikut ini susunan dari tabel produk yang berhasil dibuat:

Field	Type	Null	Key	Default	Extra
kode	varchar(25)		PRI		
harga	int(11)	YES		0	

2.3.15 Melihat Struktur Tabel

Perintah `describe nama_tabel;` digunakan untuk melihat struktur tabel yang telah dibuat. Namun demikian, pastikan Anda sudah berada pada database yang mempunyai tabel tersebut. Perhatikan contoh di bawah ini:

```
mysql> use latihan;
Database changed

mysql> show tables;
+-----+
| Tables_in_latihan |
+-----+
| produk            |
| berita            |
| toko              |
| supermarket       |
+-----+
4 rows in set (0.07 sec)

mysql> describe toko;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| buku  | varchar(25)   | YES  |     | NULL    |       |
| harga | bigint(20)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.10 sec)
```

2.3.16 Mengubah Struktur Tabel

Apabila pada struktur yang Anda buat terjadi kesalahan atau Anda menginginkan penggantian, gunakan perintah `alter` untuk mengubahnya. Terdapat empat macam perubahan dalam struktur tabel, yaitu :

1. Perubahan nama field/kolom

Perubahan yang terjadi hanya terbatas pada nama field/kolom saja. Nama field/kolom yang lama diganti dengan nama field/kolom yang baru.

Struktur penulisan:

```
alter table nama_tabel change nama_field_lama nama_field_baru
tipe_data;
```

Contoh :

```
mysql> alter table toko change buku nama varchar(25);
```


Hasil:

Field	Type	Null	Key	Default	Extra
nama	varchar(25)	YES		NULL	
harga	bigint(20)	YES		NULL	

Semula, nama field yang paling atas adalah kolom buku. Akibat perintah pada contoh di atas, field paling atas berubah menjadi kolom nama.

2. Perubahan tipe data

Perubahan yang terjadi hanya terbatas pada tipe data yang digunakan oleh field/kolom tertentu. Tipe data yang baru langsung disebutkan di belakang nama field/kolom tanpa harus menyebutkan tipe data yang lama.

Struktur penulisan:

```
alter table nama_tabel modify nama_field tipe_data_baru;
```

Contoh:

```
mysql> alter table toko modify nama longtext;
```

Hasil:

Field	Type	Null	Key	Default	Extra
nama	longtext	YES		NULL	
harga	bigint(20)	YES		NULL	

Semula, tipe field yang paling atas (nama) adalah varchar(25). Akibat perintah pada contoh di atas, berubah menjadi longtext.

3. Penambahan field

Struktur tabel akan berubah dengan bertambahnya field/kolom baru di dalamnya.

Struktur penulisan:

```
alter table nama_tabel add nama_field tipe_data_baru;
```

Contoh:

```
mysql> alter table supermarket add stok int;
```

Hasilnya:

Field	Type	Null	Key	Default	Extra
nama	longtext	YES		NULL	
harga	bigint(20)	YES		NULL	
stok	int(11)	YES		NULL	

Semula, hanya terdapat dua kolom saja (buku, harga) di dalam tabel. Akibat perintah pada contoh di atas, jumlahnya bertambah menjadi tiga kolom. Tambahan kolom tersebut adalah field stok.

4. Penghapusan field

Struktur tabel mengalami perubahan akibat berkurangnya field/kolom tertentu.

Struktur penulisan:

```
alter table nama_tabel drop column nama_field;
```

Contoh:

```
mysql> alter table supermarket drop column stok;
```

Hasil:

Field	Type	Null	Key	Default	Extra
nama	longtext	YES		NULL	
harga	bigint(20)	YES		NULL	

Pada contoh sebelumnya, sudah terdapat tiga kolom (buku, harga, stok) di dalam tabel. Akibat perintah pada contoh di atas, jumlah kolom berkurang menjadi dua kolom. Kolom yang hilang adalah field stok.

2.3.17 Mengganti Nama Tabel

Seringkali penggantian tabel juga meliputi nama tabel itu sendiri. Oleh karena masih dalam obyek database, perintah penggantian nama tabel juga bisa dilakukan dengan DDL. Perintah DDL untuk kepentingan tersebut adalah alter.

Struktur penulisan:

```
alter table nama_table_lama rename nama_table_baru
```

Contoh:

```
mysql> alter table toko rename supermarket;
Query OK, 0 rows affected (0.06 sec)

mysql> show tables;
+-----+
| Tables_in_latihan |
+-----+
| produk            |
| berita            |
| toko              |
| supermarket -----> sebelumnya adalah tabel toko
+-----+
4 rows in set (0.07 sec)
```

2.3.18 Menghapus Tabel

Tabel dapat saja dihapus karena sudah tidak dibutuhkan lagi atau karena terjadi kesalahan. Perintah untuk menghapus tabel adalah drop table.

Struktur penulisan:

```
drop table nama_tabel;
```

Contoh:

```
mysql> drop table supermarket;
```

2.3.19 Memasukkan Data (Insert)

Terdapat dua perintah yang dapat digunakan untuk memasukkan data ke dalam tabel:

```
insert into nama_tabel values (isi_field_1, isi_field_2, ...,
isi_field_n);
```

Contoh:

```
insert into produk values ('SG100',2500);
```

```
insert into nama_tabel (nama_field_1, nama_field_2, ...,  
nama_field_n) values (isi_field_1, isi_field_2, ..., isi_field_n);
```

Contoh:

```
insert into produk (kode,harga) values ('SG100',2500);
```

2.3.20 Menampilkan Data (Select)

Perintah untuk menampilkan data akan banyak kita gunakan untuk menampilkan informasi berdasarkan kriteria dan urutan tertentu. Berikut ini perintah untuk menampilkan data:

Struktur Penulisan:

```
select [fields] from [nama_tabel]  
where [kondisi]  
order by [nama_field]  
group by [nama_field] asc | desc  
limit [batasan];
```

Setelah penulisan select, [fields] dapat diganti dengan menyebutkan satu nama field saja atau beberapa field sekaligus yang dipisah dengan tanda koma (.). Penulisan field-field tersebut digunakan untuk memunculkan data dari kolom mana saja yang akan ditampilkan. Jika seluruh kolom akan ditampilkan, Anda dapat menggunakan tanda asterik (*) untuk mewakilinya.

Kemudian, perintah dilanjutkan dengan penulisan from [nama_tabel] yang dimaksudkan untuk menyebutkan nama tabel yang akan digunakan sebagai sumber untuk menampilkan kolom-kolom yang telah disebutkan sebelumnya.

Berikut ini contoh penulisan yang dapat menggambarkan hasil dari penulisan perintah tersebut.

```
select kode from produk;
```

Oleh karena perintah tersebut, yang akan ditampilkan dari tabel produk adalah kolom kode saja:

```
+-----+  
| kode  |  
+-----+
```

```
| SG100 |
| BD234 |
+-----+
```

Jika digunakan untuk menampilkan kolom kode dan harga dalam tabel produk maka dapat ditulis dengan :

```
select kode,harga from produk;
```

Hasilnya akan menampilkan semua data yang berasal dari kolom kode dan harga:

```
+-----+-----+
| kode  | harga |
+-----+-----+
| SG100 | 2500  |
| BD234 | 2250  |
+-----+-----+
```

Untuk menampilkan data dari seluruh kolom yang ada dalam tabel produk, Anda dapat menggunakan perintah:

```
select * from produk;
```

Jika tabel produk berisi kolom kode, nama, harga, dan berat maka yang akan ditampilkan adalah:

```
+-----+-----+-----+-----+
| kode  | nama      | harga | berat |
+-----+-----+-----+-----+
| SG100 | sikat gigi | 2500  | 200   |
| BD234 | pasta gigi | 2250  | 100   |
+-----+-----+-----+-----+
```

Catatan :

- ✓ Gunakan DISTINCT dalam perintah select agar data yang kembar hanya diwakili oleh satu data saja.

Contoh:

```
mysql> select distinct * from produk;
```

Pada baris where [kondisi], dituliskan kriteria apa saja yang disyaratkan untuk menampilkan data. Contoh:

- ✓ Untuk menampilkan produk pasta gigi.

```
select * from produk where nama = 'Pasta Gigi';
```

- ✓ Untuk menampilkan produk pasta gigi dan sikat gigi

```
select * from produk where nama = 'Pasta Gigi' and  
nama='Sikat Gigi';
```

Penulisan kondisi yang lebih rumit dapat mengidentifikasi tampilnya data yang lebih spesifik. Berikut ini beberapa contoh penggunaan klausa where untuk menyatakan kondisi yang lebih detail:

1. Berdasarkan kondisi yang sesuai

Penggunaan klausa where berguna untuk menampilkan data yang sesuai dengan pernyataan dalam kondisi yang disyaratkan.

- ✓ Untuk menampilkan harga produk yang di antara (between) 1000 dan 2000.
 - ❖ `select * from produk where harga between 1000 and 2000;`
 - ❖ atau sama artinya dengan fungsi boolean di bawah ini:
`select * from produk where harga>1000 and harga<2000;`
- ✓ Untuk menampilkan nama produk yang diawali dengan huruf "S".
 - ❖ Awal sebuah teks berada pada satu huruf di sebelah kiri (left), maka dapat ditulis dengan:
`select * from produk where left(kode,1)="S";`
 - ❖ atau sama artinya dengan penulisan sembarang huruf yang huruf depannya adalah huruf "S". Penulisan sembarang huruf dapat diwakili dengan tanda persen (%).
`select * from produk where kode like "S%";`
- ✓ Untuk menampilkan nama produk yang diakhiri dengan huruf "S".
 - ❖ Akhir sebuah teks berada pada satu huruf di sebelah kanan (right), maka dapat ditulis dengan:
`select * from produk where right(kode,1)="S";`

- ❖ atau sama artinya dengan penulisan sembarang huruf yang huruf akhirnya adalah huruf "S". Penulisan sembarang huruf dapat diwakili dengan tanda persen (%).

```
select * from produk where kode like "%S";
```

- ✓ Untuk menampilkan nama produk yang huruf keduanya adalah "e".

- ❖ Untuk menentukan huruf yang berada di tengah-tengah teks, dapat digunakan substring:

```
select * from produk where substring(kode,2,1)="e";
```

- ❖ atau sama artinya dengan penggunaan fungsi mid di bawah ini:

```
select * from produk where mid(kode,2,1)="e";
```

- ❖ atau sama artinya dengan penulisan sembarang huruf di bagian akhir tetapi dengan dua huruf di depannya adalah satu karakter sembarang dan huruf "e". Penggunaan garis bawah (_) adalah untuk menggantikan setiap satu karakter sembarang, sedangkan tanda persen (%) menggantikan semua karakter sembarang yang tersisa.

```
select * from produk where kode like "_e%";
```

2. Berdasarkan kondisi yang tidak sesuai

Penggunaan klausa where untuk menampilkan data yang tidak sesuai dengan pernyataan dalam kondisi yang disyaratkan.

- ✓ Menampilkan data yang tidak diawali dengan huruf "S".

```
select * from produk where left(kode,1)<>"S";
```

- ✓ atau sama artinya dengan penulisan NOT LIKE di bawah ini.

```
select * from produk where kode not like "S%";
```

Selain penggunaan klausa where, HAVING dapat juga dipakai untuk menyatakan kondisi. Di bawah ini contoh yang sama

dengan sebelumnya, dengan menggunakan klausa HAVING untuk menyatakan kondisi :

- ✓ select * from produk having left(kode,1)="e";
- ✓ select * from produk having kode like "_e%";

2.3.21 Mengurutkan Tampilan

Klausa ORDER BY digunakan untuk mengurutkan hasil. Dari hasil query sebelumnya, akan tampak hasilnya tidak berurutan dan tergantung dari urutan yang digunakan dalam memasukkan data. Jika Anda ingin produk ditampilkan urut berdasarkan urutan terkecil ke besar, Anda maka dapat menggunakan ASC (ascending):

- ✓ select * from produk order by kode asc;
- ✓ select * from produk having left(kode,1)="e" order by kode asc;

Untuk mengurutkan data dari yang terbesar ke yang terkecil, gunakan DESC (descending).

- ✓ select * from produk order by kode desc;
- ✓ select * from produk having left(kode,1)="e" order by kode desc;

2.3.22 Menampilkan Gabungan Beberapa Tabel

Dalam penggunaannya sehari-hari, seringkali kita dihadapkan dengan permasalahan dalam menampilkan beberapa tabel sekaligus untuk keperluan tertentu. Penggunaan JOIN dalam query SELECT dapat menghasilkan tampilnya beberapa tabel untuk itu.

Untuk memudahkan Anda memahami penggunaan join, berikut ini contoh data yang akan diolah:

Tabel Items		Tabel Pembelian	
Kode	Nama	Kode	Jumlah
A100	Keyboard	A100	125
A110	Mouse	A210	250
A210	Speaker	A220	100

A220 Monitor

✓ [CROSS] JOIN

Cara ini merupakan pengungkapan hubungan antartabel yang paling sederhana. Dalam penggunaan cross join, Anda tidak diharuskan untuk menuliskan kondisi. Namun demikian Anda akan mendapatkan dua belas baris dari empat baris tabel items yang dikalikan dengan tiga baris tabel pembelian.

```
mysql> select * from items cross join pembelian;
+-----+-----+-----+-----+
| kode | nama      | kode | jumlah |
+-----+-----+-----+-----+
| A100 | Keyboard  | A100 | 125     |
| A110 | Mouse     | A100 | 125     |
| A210 | Speaker   | A100 | 125     |
| A220 | Monitor   | A100 | 125     |
| A100 | Keyboard  | A210 | 250     |
| A110 | Mouse     | A210 | 250     |
| A210 | Speaker   | A210 | 250     |
| A220 | Monitor   | A210 | 250     |
| A100 | Keyboard  | A220 | 100     |
| A110 | Mouse     | A220 | 100     |
| A210 | Speaker   | A220 | 100     |
| A220 | Monitor   | A220 | 100     |
+-----+-----+-----+-----+
12 rows in set (0.03 sec)
```

Hasil tersebut cukup membingungkan dan tidak sesuai dengan harapan. Sesungguhnya yang diinginkan adalah tampilnya data dari tabel item yang sama dengan data dari tabel pembelian. Untuk itu dapat ditambahkan klausa where untuk menyatakan persamaan tersebut.

```
mysql> select * from items cross join pembelian where
items.kode=pembelian.kode;
+-----+-----+-----+-----+
| kode | nama      | kode | jumlah |
+-----+-----+-----+-----+
| A100 | Keyboard  | A100 | 125     |
| A210 | Speaker   | A210 | 250     |
| A220 | Monitor   | A220 | 100     |
+-----+-----+-----+-----+
```

✓ INNER JOIN

Cara ini hampir sama dengan sebelumnya, namun penggunaan on [kondisi] diharuskan dalam penulisan inner join. Jika on [kondisi] tidak dituliskan maka akan

menyebabkan perintah tidak menampilkan data apapun karena error.

```
mysql> select * from items inner join pembelian on
items.kode=pembelian.kode;
```

kode	nama	kode	jumlah
A100	Keyboard	A100	125
A210	Speaker	A210	250
A220	Monitor	A220	100

✓ STRAIGHT JOIN

Cara ini identik dengan join. Namun demikian, pada straight join tidak dikenal klausa where untuk menyatakan kondisi yang ingin ditampilkan.

```
mysql> select * from items straight join pembelian;
```

kode	nama	kode	jumlah
A100	Keyboard	A100	125
A110	Mouse	A100	125
A210	Speaker	A100	125
A220	Monitor	A100	125
A100	Keyboard	A210	250
A110	Mouse	A210	250
A210	Speaker	A210	250
A220	Monitor	A210	250
A100	Keyboard	A220	100
A110	Mouse	A220	100
A210	Speaker	A220	100
A220	Monitor	A220	100

✓ LEFT [OUTER] JOIN

Berbeda dengan cara lainnya, left outer join akan menampilkan tabel di sebelah kanannya (dalam hal ini tabel pembelian) dengan nilai NULL jika tidak terdapat hubungan antara tabel di sebelah kiri (dalam hal ini tabel items).

```
mysql> select * from items left join pembelian on
items.kode=pembelian.kode;
```

kode	nama	kode	jumlah
A100	Keyboard	A100	125
A110	Mouse	NULL	NULL
A210	Speaker	A210	250
A220	Monitor	A220	100

```
+-----+-----+-----+-----+
```

✓ RIGHT [OUTER] JOIN

Kebalikan dengan left outer join, pada right outer join, apabila data di sebelah kanan tabel tidak dapat menemukan hubungannya dengan tabel di sebelah kiri maka akan ditampilkan NULL.

Untuk melihat perbedaan yang kentara, tambahkan lebih dahulu satu data di dalam tabel pembelian yang tidak mempunyai hubungan dengan tabel items.

```
mysql> insert into pembelian values ('A330',20);
```

Kemudian, lanjutkan dengan:

```
mysql> select * from items right join pembelian on  
items.kode=pembelian.kode;
```

```
+-----+-----+-----+-----+  
| kode | nama      | kode | jumlah |  
+-----+-----+-----+-----+  
| A100 | Keyboard | A100 | 125 |  
| A210 | Speaker  | A210 | 250 |  
| A220 | Monitor  | A220 | 100 |  
| NULL | NULL     | A330 | 20 |  
+-----+-----+-----+-----+
```

✓ NATURAL [RIGHT | LEFT [OUTER]] JOIN

Penggunaan natural right outer join sama halnya dengan menuliskan right join tanpa pernyataan kondisi. Pernyataan tersebut secara otomatis menganggap bahwa pada kedua tabel terdapat persamaan.

```
mysql> select * from items natural right join pembelian;
```

```
+-----+-----+-----+-----+  
| kode | nama      | kode | jumlah |  
+-----+-----+-----+-----+  
| A100 | Keyboard | A100 | 125 |  
| A210 | Speaker  | A210 | 250 |  
| A220 | Monitor  | A220 | 100 |  
+-----+-----+-----+-----+
```

Penggunaan natural left outer join sama halnya dengan menuliskan left join tanpa pernyataan kondisi. Oleh karena pernyataan tersebut, secara otomatis dianggap bahwa pada kedua tabel terdapat persamaan.

```
mysql> select * from items natural left outer join pembelian;
```

kode	nama	kode	jumlah
A100	Keyboard	A100	125
A110	Mouse	NULL	NULL
A210	Speaker	A210	250
A220	Monitor	A220	100

2.3.23 Mengubah Data

Apabila data sudah kadaluwarsa, berikut ini perintah yang digunakan untuk mengubah data:

Struktur penulisan:

```
update nama_tabel set nama_field_1=isi_baru_1,
nama_field_2=isi_baru_2, .., nama_field_n=isi_baru_n where
kriteria;
```

Contoh:

```
update produk set harga=3000 where nama='Sikat Gigi';
```

2.3.24 Menghapus Data (Delete)

Apabila terdapat kesalahan data atau data sudah tidak dipakai lagi, hapuslah dengan perintah delete.

Struktur penulisan:

```
delete from nama_tabel where kriteria;
```

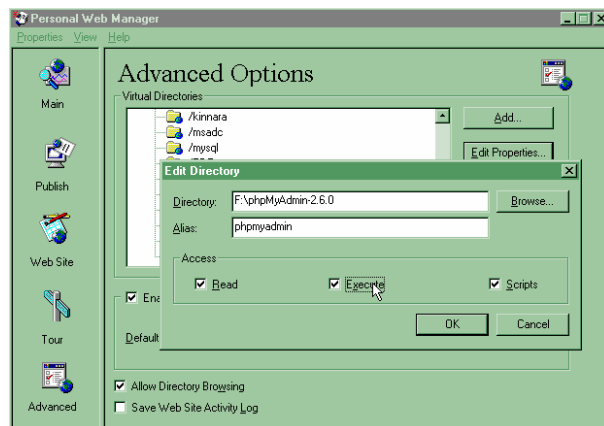
Contoh:

```
delete from produk where nama="Pasta Gigi";
```

2.4 Mengelola Data MySQL dengan PHPMyAdmin

Selain dengan perintah-perintah yang ditulis di dalam server MySQL, Anda juga dapat menggunakan PHPMyAdmin yang mempunyai tampilan grafis berbasis web untuk mengelola data di server MySQL. Anda hanya perlu menyalin direktori phpMyAdmin dari CDROM pendamping ke salah satu direktori dari server web. Jika Anda melakukannya di PWS, lokasi penyimpanan dapat berada di direktori Inetpub\wwwroot, namun jangan lupa untuk

menambahkan hak eksekusi pada virtual directory phpMyAdmin melalui Personal Web Manager.



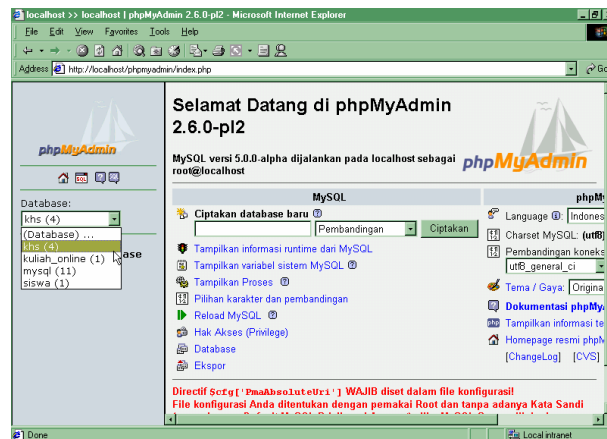
Gambar 2.20 Personal Web Manager

Jika Anda menggunakan Apache sebagai server web, phpMyAdmin dapat diletakkan di dalam direktori htdocs dari Apache atau /var/www/html.

Setelah proses penyalinan selesai, pastikan user dan password di dalam file config.inc.php di direktori phpmyadmin telah sesuai dengan server MySQL. Jika tidak, Anda harus menyesuaikannya. Default setting adalah user root tanpa menggunakan password. Nantinya Anda harus segera mengubahnya demi alasan keamanan. Berikut ini adalah setting user dan password pada file config.inc.php:

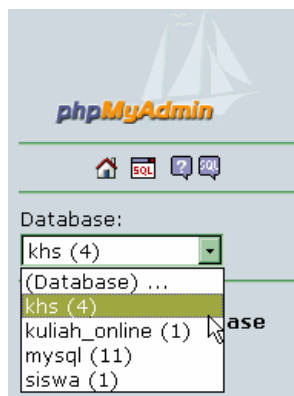
```
$cfg['Servers'][$i]['user']      = 'root';  
$cfg['Servers'][$i]['password'] = '';
```

Apabila semua setting phpMyAdmin telah disesuaikan dengan server MySQL, bukalah jendela browser web dan akseslah alamat phpMyAdmin di <http://localhost>. Dukungan PHPMyAdmin versi 2.6.0 terhadap bahasa Indonesia akan semakin memudahkan Anda dalam mengelola data MySQL melalui PHPMyAdmin.



Gambar 2.21 Tampilan PHPMyAdmin dalam bahasa Indonesia

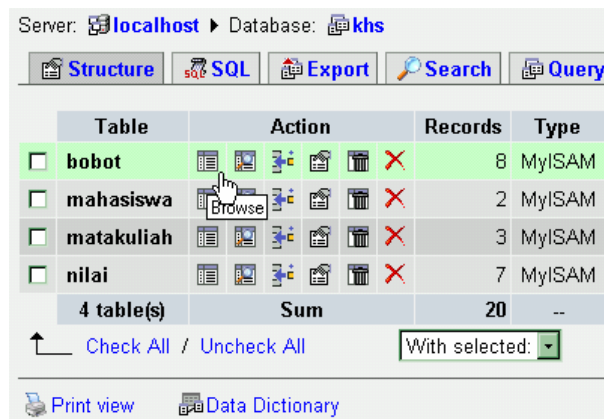
Tampilan PHPMyAdmin dibagi ke dalam 2 (dua) frame. Bagian pertama, frame kiri, merupakan menu untuk memilih database yang akan dikelola. Menu pulldown berisi nama-nama database yang apabila Anda klik akan menampilkan isi tabel di frame sebelah kanan.



Gambar 2.22 Frame kiri untuk memilih database

Pada bagian kedua, yaitu frame kanan, merupakan jendela utama untuk membuat dan mengelola database serta membuat dan mengelola isi tabel. Anda dapat mengerjakan semua pekerjaan

administrator database di server MySQL melalui PHPMyAdmin dengan mengklik mouse.



Gambar 2.22 Frame kiri untuk memilih database

Semakin tinggi frekuensi penggunaan PHPMyAdmin, semakin lancar pula pengelolaan data MySQL dengan PHPMyAdmin.

2.5 Koneksi MySQL dengan PHP

Oleh karena MySQL sudah terhubung saat instalasi dan konfigurasi PHP, Anda dapat langsung memanfaatkan perintah-perintah dalam PHP untuk melakukan koneksi ke server MySQL. Program di bawah ini merupakan contoh sederhana untuk menampilkan data dari tabel user. Anda hanya perlu menyesuaikan SQL, database, user, dan password yang digunakan.

mysql.php

```
<?php
$hostname="localhost";
$username="root";
$password="";
$namadatabase="mysql";
if (! $dbh = mysql_connect($hostname, $username, $password)) {
    echo mysql_error();
    return 0;
}
```

```

if (! mysql_select_db($namadatabase)) {
    echo mysql_error();
    return 0;
}
$sql="select user from user";

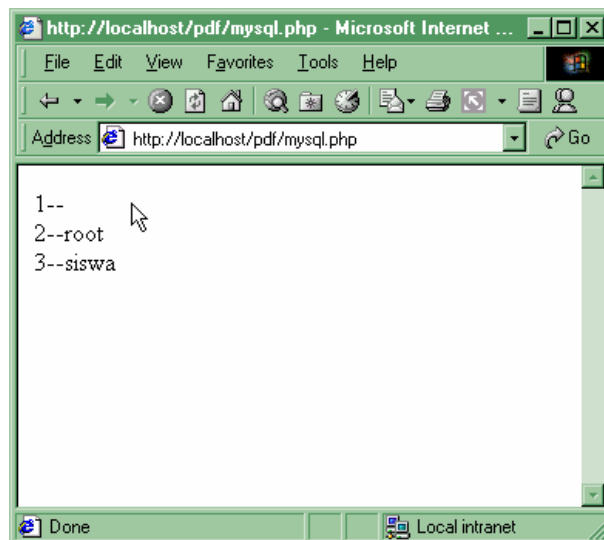
if (! $res=mysql_query($sql, $dbh)) {
    echo mysql_error();
    return 0;
}

$i=0;
while ($row=mysql_fetch_row($res)) {
    echo ($i+1)."--".$row[0]."<br>\n";
    $i++;
}

if ($i==0) {
    echo "Data masih kosong!";
    return 0;
}
?>

```

Hasilnya dapat Anda lihat pada gambar di bawah ini. Terdapat 3 (tiga) user yang berhasil ditampilkan, termasuk user kosong.



Gambar 2.23 Data dari tabel user

Dengan berakhirnya pembahasan pada bab ini, diharapkan Anda sudah siap untuk mengembangkannya dalam pembuatan dokumen PDF yang dibahas pada Bab 3.