

# BAB 1

## Pengenalan PHP

---

Pada bab ini Anda akan dikenalkan dengan PHP sebagai salah satu server side programming, pengertian server side programming dan client side programming, serta alasan mengapa memilih PHP. Tak lupa pula akan dijelaskan tool apa saja yang diperlukan untuk belajar PHP.

### Tentang Client Side dan Server Side Programming

PHP atau merupakan singkatan rekursif dari PHP : Hypertext Preprocessor adalah suatu bahasa pemrograman yang termasuk kategori server side programming. Wah... apaan tuh maksudnya server side programming? OK... server side programming adalah jenis bahasa pemrograman yang nantinya script/program tersebut akan dijalankan oleh server. Selanjutnya hasil pengolahan script/program tersebut akan dikirim ke client sebagai output.

Sebagai gambaran dari server side programming adalah sebagai berikut ini.

Misalkan Anda ingin membuat script/program untuk menjumlahkan dua buah bilangan dengan menggunakan PHP. Nah... andaikan Anda telah membuat form untuk input kedua bilangan dan membuat script PHP untuk mengolah penjumlahan kedua bilangan. Dalam hal ini, keduanya (form dan script PHP) sama-sama diupload ke server.

Selanjutnya misalkan ada seorang user yang ingin menjumlahkan dua bilangan, maka user tadi akan mengakses ke server. Apa yang diakses? Tentunya form input bilangan yang akan dijumlahkan. Dalam hal ini posisi si user adalah sebagai client. Nah... setelah si user mengisi bilangan pada form, selanjutnya dia mensubmitnya. Trus... kemana mensubmitnya? Karena Anda tadi membuat script PHP dan form input telah diarahkan ke script PHP tersebut, maka data input akan disubmit ke server untuk diolah menggunakan script PHP yang telah Anda buat tadi. Setelah input diolah (dijumlahkan), maka hasilnya akan kembali dikirim ke client sebagai output sehingga si user mengetahui hasil penjumlahan dua bilangan yang diinputkan melalui form tadi.

Penjelasan detail tentang cara mengkaitkan antara form dan script PHP sebagai pengolahnya akan dijelaskan di bab tentang pemrosesan form.

Trus... selain server side programming, apakah ada jenis yang lain? Oh.. Ada donk yaitu client side programming. Jenis programming ini merupakan kebalikan dari server side

programming. Untuk client side programming, script/program akan diproses di dalam client sendiri.

Untuk menggambarkan client side programming, kita ambil contoh kasus yang sama dengan sebelumnya yaitu script menjumlahkan dua bilangan. OK... misalkan Anda membuat form input untuk memasukkan dua bilangan, serta script untuk menjumlahkannya. Selanjutnya keduanya ditaruh/diupload ke server. Misalkan ada user yang ingin menjumlahkan dua bilangan, maka dia akan mengakses ke server form input.

Karena Anda menggunakan script pengolah berjenis client side programming, maka script tersebut juga akan terakses oleh si user. Biasanya script ini diletakkan dalam satu halaman yang sama dengan form. Oleh karena itu bila si user mengakses form, maka script juga otomatis akan terakses. Setelah si user memasukkan bilangan ke dalam form, lalu dia mensubmitnya. Trus... kemana submitnya? Yap... submit ini tidak ditujukan ke server melainkan ke client itu sendiri dan diolah oleh script yang ikut terakses tadi. Kemudian hasilnya ditampilkan sebagai output.

Salah satu contoh bahasa pemrograman web yang berjenis client side programming adalah Javascript.

Untuk melihat perbedaan yang kentara antara bahasa pemrograman web yang berjenis server side programming dan client side programming adalah bisa atau tidaknya script/program itu ditampilkan ke dalam browser ketika mengaksesnya. Apa maksudnya?

Cobalah akses suatu halaman situs yang di dalamnya terdapat Javascript, misalnya <http://detik.com>. Ketika Anda lihat source halaman web tersebut (klik kanan, klik View Page Source pada Firefox atau klik kanan, klik View Source pada IE), maka dapat Anda lihat isi kode script Javascriptnya, yang biasanya ditandai dengan

```
<script type="text/javascript" language="javascript">
```

dan diakhiri dengan tanda

```
</script>
```

Mengapa kode Javascript itu bisa terlihat? Ya... karena script itu nantinya akan diproses oleh komputer client, sehingga kode itu ibaratnya harus didownload terlebih dahulu ke dalam client.

Selanjutnya bandingkan 'penampakkannya' ☺ bila Anda mengakses script PHP, misalkan script yang telah saya buat untuk percobaan (<http://rosihanari.net/test.php>). Ketika Anda lihat dibrowser, maka yang tampil adalah hanyalah berupa teks pesan saja. Dan

apabila Anda lihat sourcenya, maka tidak tampak perintah-perintah yang berupa kode PHP. Padahal isi dari script test.php ini adalah sebagai berikut:

```
<?php  
  
echo "Hello apa kabar?<br />";  
echo "Terimakasih atas kepercayaan Anda pada Rosihan Ari sebagai pembimbing belajar  
PHP Anda";  
  
?>
```

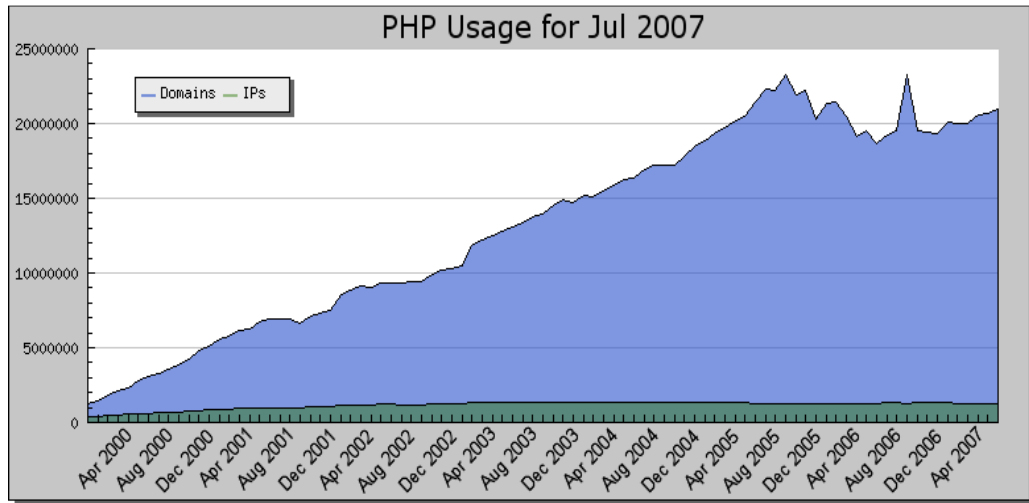
Mengapa bisa demikian? Ya... karena script test.php ini telah dijalankan atau diproses di server dan bukan di client. Dalam hal ini client hanya menerima output hasil pengolahan script di server.

Mudah-mudahan dari uraian di atas Anda sudah bisa membedakan apa itu server side programming dan client side programming.

## Kenapa PHP?

Di samping PHP, sebenarnya ada beberapa jenis pemrograman lain yang termasuk server side programming yaitu JSP (Java Server Pages), ASP (Active Server Pages), Python, dan Perl, serta masih ada lagi beberapa yang lain. Namun mengapa saya rekomendasikan PHP untuk Anda pelajari? Ya... karena PHP memiliki beberapa kelebihan antara lain:

1. Bahasa pemrograman PHP sangat mudah dipelajari karena mirip dengan bahasa C/C++
2. PHP mudah diimplementasikan karena software PHP servernya 100% Free
3. Kebanyakan server web hosting mensupport PHP
4. PHP paling banyak digunakan oleh web programmer di seluruh dunia

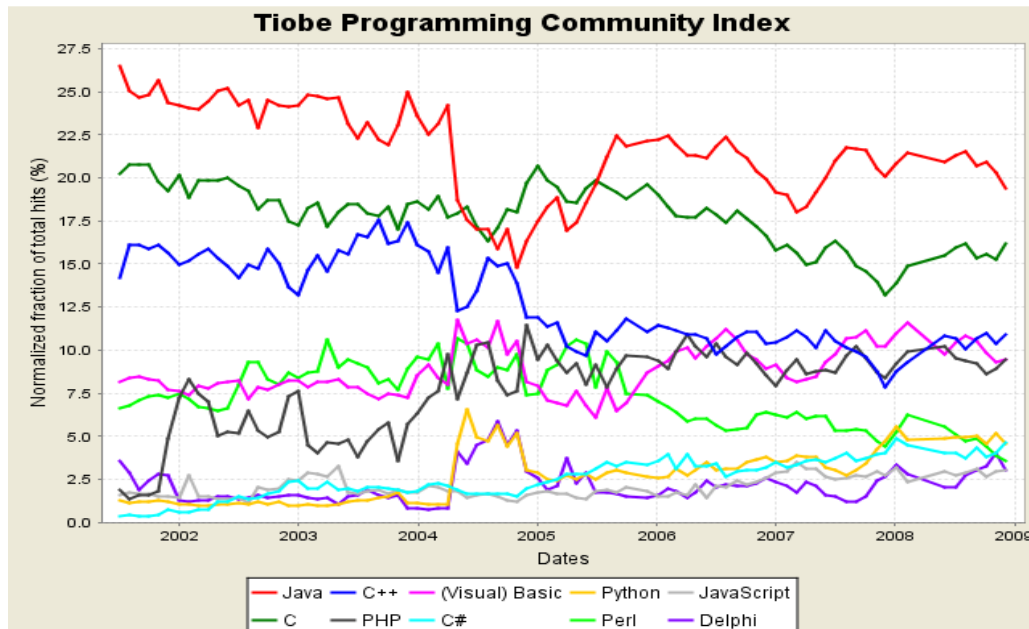


Statistik situs yang menggunakan PHP (sekitar 20 juta situs)

Sumber : <http://www.php.net/usage.php>

Menurut TIOBE index, PHP merupakan bahasa pemrograman yang memiliki tingkat rating no. 4 setelah Java, C dan C++ di akhir tahun 2008 ini (Sumber: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>).

Selain itu trend penggunaan PHP sebagai bahasa pemrograman selama kurun waktu 2002 s/d 2008 adalah sebagai berikut



Sumber : <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

Dari grafik di atas tampak bahwa PHP masuk dalam 10 besar bahasa pemrograman yang paling banyak digunakan, dan PHP termasuk urutan nomor 2 pemrograman favorit.

5. Anda bisa melakukan instalasi PHP server sendiri dengan mudah di komputer pribadi Anda untuk belajar.
6. Integrasi antara PHP dan database mudah untuk diimplementasikan karena PHP mensupport banyak database (DBMS), seperti MySQL, MS. Access, Oracle, DB2, Dbase, Interbase dll.

## Tool Apa Saja Yang Diperlukan Untuk Belajar PHP?

Untuk memudahkan Anda belajar PHP, berikut ini beberapa tool yang sebaiknya Anda miliki :

1. Software web server, bisa dipilih salah satu antara Apache, IIS, PWS (100 % free)
2. Software PHP server (100% free)
3. Software database, sebaiknya pilih MySQL karena 100% free. Bila sudah menguasai PHP + MySQL bisa dikembangkan dengan database yang lain.
4. Text Editor, misalkan Notepad atau Notepad++. Dalam hal ini saya merekomendasikan Notepad++ karena selain software ini 100% free, juga mensupport untuk memudahkan penulisan bahasa program.
5. Web Editor, misalnya Dreamweaver atau Frontpage (sifatnya optional) sebagai pendukung saja.

Untuk tool no. 1 s/d 3 sebenarnya Anda tidak perlu mencarinya atau menginstalasi secara terpisah, karena ada software paket yang di dalamnya sudah tersedia web server Apache, PHP server dan MySQL. Dalam hal ini saya merekomendasikan **AppServ** (<http://www.appservnetwork.com>), XAMPP atau EasyPHP sebagai software-nya, yang kesemuanya *absolutely FREE*.

## BAB 2

# Instalasi PHP dan Software Pendukungnya

---

Pada bab ini Anda akan mempelajari bagaimana cara melakukan instalasi PHP server dan juga pendukungnya.

Agar script PHP yang Anda buat bisa berjalan, Anda terlebih dahulu harus sudah menginstall PHP server di komputer Anda. Oya... dalam hal ini kita akan belajar menginstal PHP server ini di komputer Anda sendiri untuk keperluan belajar. Anda tidak perlu menjalankan script PHP di server khusus yang spesifikasinya besar atau di server hosting, tapi cukup di komputer Anda pribadi.

OK... kita mulai masuk ke pokok bahasan utama. Untuk bisa menjalankan PHP, Anda butuh beberapa software pendukungnya antara lain: Web Server, dan juga database sebagai pelengkap. Database di sini diperlukan ketika Anda ingin membuat aplikasi web yang diintegrasikan dengan database.

Ada beberapa macam software yang bisa digunakan sebagai web server, diantaranya adalah : Apache, IIS, dan Personal Web Server. Sedangkan untuk software database, atau sering disebut dengan istilah DBMS (Database Management System), Anda bisa menggunakan MySQL, SQL Server, MS. Access, dll.

Nah... di sini saya hanya akan mengajarkan bagaimana melakukan instalasi PHP server, Apache sebagai web servernya dan juga MySQL sebagai DBMS nya. Mengapa saya memilih Apache dan MySQL? Ya... karena hampir seluruh server hosting telah mendukung kedua software. Selain itu keduanya adalah free sehingga tidak perlu khawatir lagi terkait lisensinya.

OK... untuk menginstal PHP dan pendukungnya, Anda tidak perlu melakukan instalasi satu persatu, namun bisa langsung sekaligus semuanya. Lho kok bisa? Ya... karena saat ini banyak tersedia software bundling yang di dalamnya sudah ada PHP server, Apache dan MySQL nya. Beberapa di antaranya adalah AppServ, XAMPP dan EasyPHP. Tapi dalam hal ini saya sangat merekomendasikan AppServ karena konfigurasinya yang sangat mudah dibandingkan yang lain.

## Instalasi AppServ

Berikut ini cara melakukan instalasi AppServ di komputer Anda.

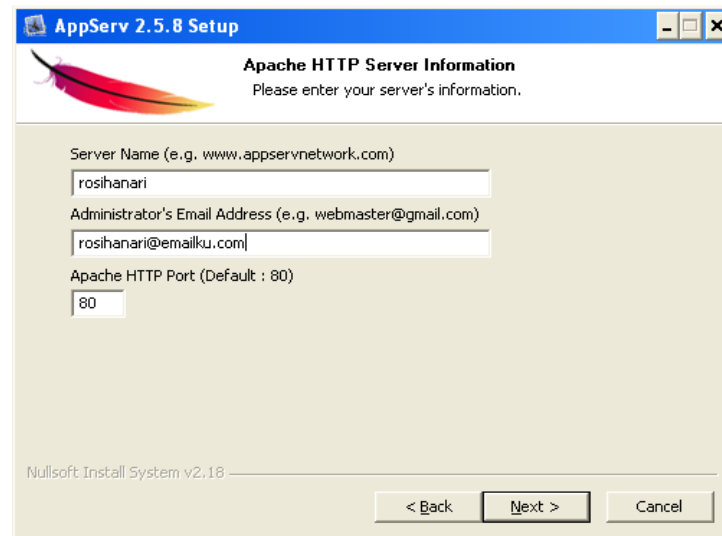
1. Download software AppServ ini di <http://appservnetwork.com>
2. OK... setelah Anda mendownloadnya, silakan double click file hasil downloadnya.
3. Secara default, letak file AppServ untuk instalasi ini adalah di C:\AppServ
4. Bila pada instalasi AppServ muncul kotak dialog seperti ini, maka pilihlah semua option yang ada



Option tersebut menunjukkan software apa saja dalam AppServ yang ingin diinstal, dalam hal ini pilihlah semuanya (Apache, MySQL, PHP server dan phpMyAdmin).

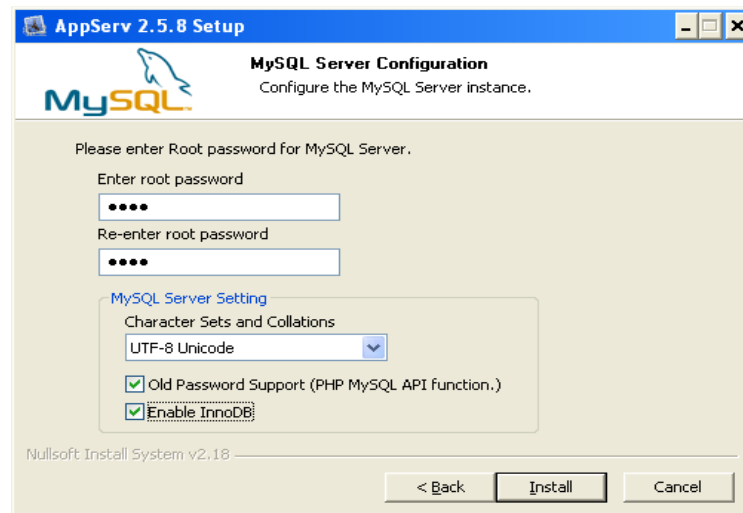
Catatan: phpMyAdmin adalah software untuk mempermudah pengelolaan database MySQL

5. Bila sampai pada tahap tampilan instalasi seperti di bawah ini



Isikan Name Server nya dengan nama Anda atau apalah sebarang. Nama itu nantinya dipakai sebagai nama server webnya. Sebaiknya jangan ada spasi untuk nama servernya. Nama server nanti bisa dipanggil di browser, misal kalau nama servernya 'rosihanari', maka Anda dapat mengakses <http://rosihanari> di browser Anda.

6. Selanjutnya apabila instalasi telah sampai pada tampilan berikut ini

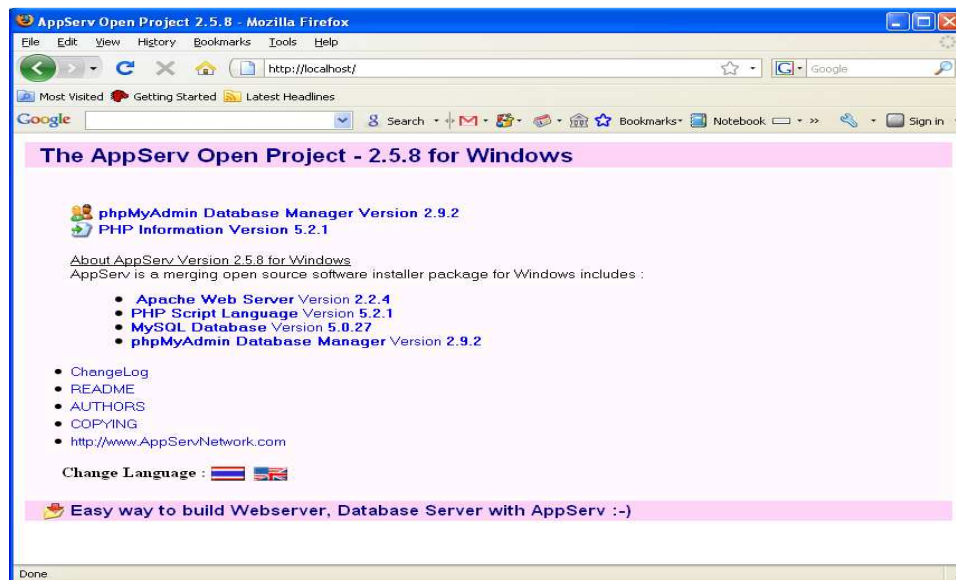


Isikan password untuk username: root guna mengakses database MySQL. Oya... sebaiknya option Enable InnoDB Anda aktifkan supaya MySQL bisa mensupport relasi dan integritas antara tabel. Apa kelebihan InnoDB? Silakan baca artikel saya di



<http://blog.rosihanari.net/foreign-key-relationship-dan-referential-integrity-di-mysql/>

7. Apabila proses instalasi selesai, maka selanjutnya cobalah buka browser Anda dan arahkan URLnya ke <http://namaserver> atau <http://localhost> yang telah Anda berikan sebelumnya pada proses instalasi. Bila browsernya menampilkan halaman berikut ini, maka instalasi AppServ sukses.



8. Halaman yang tampil di atas adalah hasil dari akses file **index.php** yang terletak dalam direktori **C:\AppServ\www**. Jadi... file-file web dan juga script PHP yang nantinya akan Anda buat harus diletakkan di direktori tersebut. Apakah bisa diletakkan di tempat lain? Oh bisa saja... bagaimana caranya, simak di Bab 3 tentang bagaimana mengatur konfigurasi di PHP dan Web Server (Apache)

## BAB 3

# Konfigurasi Apache, PHP dan MySQL

---

Pada bagian ini, Anda diharapkan bisa memahami bagaimana cara melakukan konfigurasi pada Apache, PHP dan MySQL.

### Konfigurasi Apache

Secara default, file web dan script PHP yang nantinya Anda buat akan diletakkan di direktori C:\AppServ\www. Akan tetapi jika menghendaki direktori tersebut dipindahkan, maka hal ini bisa dilakukan. Sebagai contoh, misalkan Anda membuat folder di D:\ dengan nama **webku** atau membuat direktori **D:\webku** untuk meletakkan file-file web dan script PHPnya. Nah.. supaya file web dan script PHP ini bisa diakses melalui browser dan web server maka terlebih dahulu Anda harus melakukan setting di bagian konfigurasi Apache.

Bagaimana cara melakukan setting konfigurasi Apache untuk memindahkan default direktori untuk meletakkan file web dan script PHPnya? OK ini dia langkah-langkahnya:

1. Klik START > AppServ > Configuration Server
2. Pilih "Apache Edit The httpd.conf Configuration File"
3. Carilah baris bertuliskan **DocumentRoot "C:/AppServ/www"** pada isi file httpd.conf tersebut. Gunakan fasilitas **Find** pada Notepad
4. Ubahlah baris tersebut dengan **DocumentRoot "D:/webku"**.
5. Cari juga baris <Directory "C:/AppServ/www"> dan ubahlah menjadi <Directory "D:/webku">
6. Cari juga baris <Directory "C:/AppServ/www/cgi-bin"> dan ubahlah menjadi <Directory "D:/webku/cgi-bin">
7. Setelah selesai, simpan kembali file httpd.conf tersebut.
8. Lakukan Restart pada Apache dengan cara klik START > AppServ > Control Server by Service > Restart Apache

Untuk melihat perubahannya, silakan Anda coba buat file web berikut ini pada Notepad.

```
<html>
  <head>
    <title>Ini halaman pertama</title>
  </head>

  <body>
    <h1>Ini halaman web pertamaku</h1>
  </body>
</html>
```

Simpan file tersebut dengan nama **index.htm** dan letakkan ke direktori D:\webku. Selanjutnya panggil file web tersebut menggunakan web server melalui browser dengan URL <http://localhost> atau <http://namaserver>. Bila halaman web muncul tampilan berisi teks Test.. Test sebagai mana yang kita tulis di atas, maka konfigurasi sukses. Sedangkan bila tidak muncul apa-apa atau error, maka coba cek lagi langkah-langkah konfigurasi di atas. Pastikan nama direktorinya benar.

Sekarang coba buat lagi halaman web yang lain.

```
<html>
  <head>
    <title>Ini halaman kedua</title>
  </head>

  <body>
    <h1>Ini halaman web keduaku</h1>
  </body>
</html>
```

Sekarang simpan file di atas dengan nama **hal2.htm** dan letakkan ke D:\webku. Kemudian panggil file web tersebut dengan URL <http://localhost/hal2.htm> atau <http://namaserver/hal2.htm>.

Anda juga bisa membuat folder di dalam D:\webku. Apabila Anda meletakkan file web ke dalam folder tersebut, maka cara mengakses ke file web tersebut melalui browser adalah <http://localhost/namafolder/namafile>.

#### Catatan:

Oya... meskipun Anda bisa langsung membuka kedua file web yang telah dibuat di atas melalui proses *double click* pada file melalui Windows Explorer, tapi proses ini tidak melibatkan web server. Beda prosesnya ketika Anda mengakses file web melalui <http://localhost> seperti di atas. Ketika Anda mengakses file web melalui <http://localhost>, maka pada prinsipnya proses request web yang dilakukan *client* (dalam hal ini Anda) akan diproses oleh web server. Proses yang dimaksud adalah mencari file web yang

diinginkan pada *root direktori* (D:\webku). Setelah ketemu, hasilnya akan dikirim ke web browser untuk ditampilkan.

## BAB 4

# Struktur Dasar Script PHP

---

Pada bab ini kita akan mempelajari struktur dasar script PHP, aturan penulisan statement PHP, cara menyisipkan script PHP ke dalam tag (X)HTML dan juga kebalikannya, tentang variabel, aturan penamaan variabel, sampai dengan penulisan komentar.

### Struktur Script PHP

Kode-kode PHP dituliskan di antara tanda berikut ini:

```
<?php
```

```
...  
...  
...
```

```
?>
```

atau

```
<?
```

```
...  
...  
...
```

```
?>
```

Tips : diantara kedua aturan struktur di atas, sebaiknya gunakan struktur yang pertama (diawali dengan <?php dan diakhiri dengan ?>). Mengapa? Ya... karena tidak semua web server mengenali struktur yang kedua.

## Menyimpan File PHP

Script PHP yang dibuat harus disimpan dalam ekstensi **.php**. Apabila tidak, maka script tidak akan bisa dijalankan dan hanya dianggap teks biasa saja. Dahulu script PHP dapat juga disimpan dalam bentuk ekstensi **.php3**, namun hal ini berlaku untuk PHP versi 3.x saja. Sejak rilis 4.x ke atas, ekstensi tersebut tidak digunakan lagi.

Dimanakah letak script PHP ini harus disimpan supaya bisa dijalankan? Script PHP yang Anda buat harus diletakkan dalam root direktori dari web server. Apabila Anda menggunakan AppServ, script PHP diletakkan dalam direktori **C:\AppServ\www**. Bisa juga Anda mengubah letak root direktori web servernya ke tempat lain. Untuk melakukan hal ini baca kembali Bab III.

Selanjutnya coba buat script PHP pertama Anda berikut ini:

```
<?php  
  
echo "Hello World";  
echo "Hello World";  
echo "Hello World <br />";  
echo "Hello World <br />";  
  
?>
```

Tips :

Dalam membuat script PHP, sebaiknya text editor NOTEPAD++ supaya lebih mudah mengetahui baris mana dalam kode PHP yang salah.

Simpan script PHP tersebut dengan nama **test.php**, dan letakkan ke dalam root direktori web server. Perhatikan apa yang tampak ketika script di atas dibuka pada browser (panggil script di atas dengan URL <http://localhost/test.php>) !

Sekarang kita akan lihat misalkan ada kesalahan perintah atau sintaks dalam kode PHP nya. Misalkan ada perintah echo yang salah dalam penulisannya seperti pada script berikut ini:

```
<?php  
  
eho "Hello World";  
echo "Hello World";  
echo "Hello World <br />";  
echo "Hello World <br />";
```

?>

Bila script di atas dijalankan di browser, maka akan muncul pesan error:

**Parse error:** syntax error, unexpected T\_CONSTANT\_ENCAPSED\_STRING in C:\AppServ\www\test.php on line 3

Dari pesan tersebut tampak adanya kesalahan pada baris ke-3 (on line 3). Dengan demikian Anda harus segera memperbaikinya. Dengan menggunakan Notepad++ Anda dapat dengan cepat menemukan nomor baris dari script PHP yang dibuat.

Selanjutnya jawab dalam hati pertanyaan berikut ini:

1. Apa gunanya perintah echo?
2. Mengapa teks "Hello World" yang kedua terletak sebaris dengan "Hello World" yang pertama?
3. Mengapa teks "Hello World" yang ketiga terletak pada baris yang kedua?
4. Setiap baris statement kode PHP selalu diakhiri dengan tanda semicolon/titik koma. Apa efeknya bila tanda ini dihapus?
5. Bisakah Anda membuat script PHP untuk menampilkan teks dengan format di bawah ini pada browser hanya dengan menggunakan sebuah echo saja?

```
Hello World
Hello World
Hello World
Hello World
```

Simpan script Anda ini dengan nama **test2.php** dan letakkan ke dalam folder bernama **'test'** yang terlebih dahulu dibuat pada root direktori web server. Panggil script ini dengan nama URL <http://localhost/test/test2.php>

## Menyisipkan Script PHP dalam Dokumen (X)HTML

Script PHP dapat juga disisipkan ke dalam dokumen (X)HTML. Berikut ini contohnya:

```
<html>
  <head>
    <title>Halaman PHP pertamaku</title>
  </head>
  <body>
    <h1>Script PHP dalam (X)HTML</h1>
```

```
<?php
echo "Hello World!";
echo "Hello World!";
echo "Hello World!";
echo "Hello World!";
echo "Hello World!";
?>

</body>
</html>
```

Karena dalam dokumen (X)HTML di atas terdapat script PHP, maka file di atas nantinya juga harus disimpan menggunakan ekstensi .php, bukan .htm atau .html.

Coba tulis script di atas dan simpan dengan nama **test3.php**. Letakkan script ke dalam folder '**test**' seperti sebelumnya dan panggil dengan URL <http://localhost/test/test3.php>

Kita juga bisa menyisipkan lebih dari satu script PHP dalam sebuah dokumen (X)HTML seperti contoh di bawah ini

```
<html>
<head>
  <?php echo "<title>Halaman PHP pertamaku</title>"; ?>
</head>
<body>
  <h1>Script PHP dalam (X)HTML</h1>

  <?php
  echo "<p>Ini paragraph yang dibuat dengan script PHP</p>";
  echo "<p>Ini paragraph yang dibuat dengan script PHP</p>";
  ?>

  <p>Ini paragraph yang dibuat tanpa script PHP</p>
  <p>Ini paragraph yang dibuat tanpa script PHP</p>

  <?php
  echo "<p>Ini paragraph yang dibuat dengan script PHP</p>";
  echo "<p>Ini paragraph yang dibuat dengan script PHP</p>";
  ?>

  </body>
</html>
```



Coba simpan file di atas dengan nama **test4.php** dan lihat hasilnya di browser.

## Penggunaan Variabel

Variabel dalam pemrograman digunakan untuk menyimpan suatu nilai dan jika suatu saat nilai tersebut diperlukan kembali untuk proses perhitungan, maka cukup memanggil nama variable tersebut.

Dalam PHP, nama suatu variabel ditandai dengan tanda dollar (\$). Berikut ini adalah sintaks untuk menyimpan nilai ke dalam suatu variabel.

```
$namaVariabel = nilai;
```

Adapun beberapa aturan penulisan nama variabel adalah sebagai berikut:

- Nama variabel harus diawali dengan huruf atau underscore ( \_ )
- Nama variabel hanya boleh dituliskan dengan alpha numeric a-z, A-Z, 0-9 dan underscore
- Nama variabel yang terdiri lebih dari satu kata, dapat dipisahkan dengan underscore

Berikut ini beberapa contoh menyimpan nilai ke dalam variabel.

```
<?php  
  
$teks = "Hello World!";  
$sebuah_bilangan = 4;  
$bilanganYangLain = 8.567;  
$teks2 = $teks;  
  
?>
```

Keterangan:

- Perintah pertama digunakan untuk menyimpan nilai berupa string ke dalam variabel bernama \$teks.
- Perintah kedua digunakan untuk menyimpan nilai berupa bilangan bulat 4 ke dalam variabel bernama \$sebuah\_bilangan.
- Perintah ketiga digunakan untuk menyimpan nilai berupa bilangan riil 8.567 ke dalam variabel bernama \$bilanganYangLain.
- Perintah keempat digunakan untuk menyimpan nilai yang tersimpan dalam variabel \$teks, dalam hal ini juga "Hello World!".

Catatan Penting:

Besar kecilnya huruf dalam nama variabel sangat berpengaruh. Maksudnya misalnya kita membuat variabel \$a dengan \$A, keduanya adalah berbeda. Sifat ini dinamakan *case sensitive*. Jadi hati-hati dalam menuliskan nama variabel.

## Menampilkan Nilai Variabel

Untuk menampilkan nilai yang telah tersimpan dalam variabel, kita gunakan **echo**. Berikut ini contohnya:

```
<?php

$teks = "Hello World!";
$sebuah_bilangan = 4;
$bilanganYangLain = 8.567;
$teks2 = $teks;

echo $teks;
echo "<br />" . $sebuah_bilangan;
echo "<br />Isi dari variabel \$bilanganYangLain : " . $bilanganYangLain;
echo "<br />Isi dari variabel \$teks2 adalah : " . $teks2;

?>
```

Keterangan:

- Tanda titik (dot) digunakan untuk menggabungkan string. Dalam hal ini, nilai yang akan ditampilkan dianggap sebagai suatu string.
- Tanda backslash (\) di depan \$ pada script di atas digunakan untuk menampilkan tanda \$ ke dalam browser. Coba lihat efeknya apabila tanda backslash ini dihapus.

## Lebih Lanjut Tentang Echo

Sekarang... misalkan kita akan menampilkan tag HTML berikut ini ke dalam browser menggunakan script PHP.

```
<font color="red">I love using PHP!</font>
```

Tag di atas digunakan untuk menampilkan teks 'I love using PHP!' berwarna merah.

Kira-kira bagaimana bentuk script PHP nya? Apakah seperti berikut ini?

```
<?php  
  
echo "<font color=\"red\">I love using PHP!</font>";  
  
?>
```

Bila script di atas dijalankan di browser, maka akan terdapat pesan kesalahan. Mengapa demikian? Ya... karena string yang akan di'echo' kan terdapat tanda petik dua (double quotes).

Trus... bagaimana solusinya? Ada 2 solusi yang bisa digunakan. Solusi pertama adalah ubahlah tanda double quotes pada "red" menjadi single quote 'red'.

```
<?php  
  
echo "<font color='red'>I love using PHP!</font>";  
  
?>
```

Cara yang kedua, tambahkan tanda backslash di depan setiap double quotesnya.

```
<?php  
  
echo "<font color=\"red\">I love using PHP!</font>";  
  
?>
```

## Menyisipkan Komentar dalam Script PHP

Seperti halnya bahasa pemrograman yang lain, komentar dalam suatu kode PHP tidak akan dieksekusi atau diproses. Biasanya komentar digunakan untuk memberikan deskripsi tentang script secara keseluruhan atau memberikan penjelasan pada baris perintah tertentu.

Terdapat dua cara memberikan komentar dalam PHP, yaitu

- Menggunakan tanda `//` di depan teks komentar. Perintah ini hanya bisa berlaku untuk komentar dalam satu baris
- Menggunakan tanda `/*` di depan teks komentar dan diakhiri dengan `*/`. Perintah ini dapat digunakan untuk komentar yang terdiri lebih dari satu baris.

Contoh:

```
<?php
echo "Hello World!"; // perintah ini akan mencetak Hello World!
?>
```

Contoh yang lain:

```
<?php
/*
    Berikut ini adalah perintah
    untuk menampilkan teks Hello World
    pada browser
*/
```

```
echo "Hello World!";
?>
```

Contoh yang lain:

```
<?php
// Berikut ini adalah perintah
// untuk menampilkan teks Hello World
// pada browser
```

```
echo "Hello World!";
?>
```

## BAB 5.

# Assignment & Operator Aritmatika

---

Pada bab ini akan dijelaskan lebih mendalam mengenai proses assignment serta pengenalan beberapa operator aritmatika.

### Assignment

Seperti yang telah disinggung pada bab sebelumnya (Bab IV) bahwa proses pemberian suatu nilai kepada suatu variabel diistilahkan 'assignment'. Tinjau kembali script yang telah diberikan sebelumnya pada Bab IV berikut ini

```
<?php
$teks = "Hello World!";
$sebuah_bilangan = 4;
$bilanganYangLain = 8.567;
$teks2 = $teks;

echo $teks;
echo "<br />" . $sebuah_bilangan;
echo "<br />Isi dari variabel \"$bilanganYangLain : " . $bilanganYangLain;
echo "<br />Isi dari variabel \"$teks2 adalah : " . $teks2;

?>
```

Statement-statement yang saya blok tebal (bold) tersebut merupakan contoh assignment. Apakah proses assignment hanya berupa bentuk tersebut? Yaitu \$namaVariabel = nilai; saja? Ternyata tidak hanya bentuk tersebut. Kita juga bisa membuat proses assignment ke dalam bentuk lain, misalnya:

```
<?php
$a = ($b = 4) + 5;

?>
```

Wah... apa tuh maksudnya? Statement tersebut bermakna : Nilai \$a adalah 9, yaitu diperoleh dari \$b + 5, dimana \$b diassign dengan 4. Dengan demikian sebuah statement

tersebut langsung memuat 2 buah proses assignment, yaitu assignment untuk variabel \$a dan \$b sekaligus.

Untuk membuktikan apakah benar nilai \$a sekarang 9 dan \$b adalah 4, coba tampilkan ke browser nilai kedua variabel tersebut. Bisa kan?

**Catatan:**

Proses assignment memiliki prinsip timpa-menimpa. Wah kok lucu banget istilahnya ☺ Maksudnya adalah, misalkan awalnya kita assign suatu variabel dengan nilai tertentu, katakanlah X. Selanjutnya variabel tadi kita assign lagi dengan nilai yang baru, misalkan Y, maka nilai X yang sebelumnya tersimpan akan hilang, ibaratnya ditimpa dengan nilai Y.

Sebagai bukti akan hal ini, coba jalankan script berikut ini

```
<?php
$a = 10;
$a = 9;
echo "Nilai variabel \ $a adalah : " . $a;
?>
```

Sebelum Anda lihat hasilnya di browser, coba kira-kira menurut Anda akan muncul apa di browser? Trus... coba bandingkan hasil perkiraan Anda tadi dengan apa yang muncul di browser.

**Catatan :**

Wah... lagi-lagi catatan nih... ☺ Satu lagi catatan yang perlu kita tahu, bahwa jangan pedulikan tipe data dari nilai yang diassign ke variabelnya. Secara otomatis tipe data variabel akan menyesuaikan tipe data dari nilai yang diassign-kan. Bingung ya... ??? ☺

Maksudnya begini, misalkan mula-mula kita assign suatu variabel dengan nilai X yang berupa bilangan bulat. Maka dalam hal ini, tipe data variabel tersebut juga bilangan bulat (integer). Selanjutnya kita assign lagi variabel tersebut dengan nilai berupa string, katakanlah Y. Maka secara otomatis tipe data variabel ini akan menjadi string. Tipe data integer sebelumnya tidak berlaku lagi.

Sebagai gambaran, coba jalankan script berikut ini:

```
<?php
$a = 10;

// mengecek apakah variabel $a bertipe string atau bukan
```

```
// jika $a string, maka tampilkan '$a bertipe string'  
// jika $a bukan string, maka tampilkan '$a bukan string'  
  
if (is_string($a)) echo "\$a bertipe string<br>";  
else echo "\$a bukan string<br>";  
  
$a = "Pemrograman PHP is OK";  
  
// mengecek apakah variabel $a bertipe string atau bukan  
// jika $a string, maka tampilkan '$a bertipe string'  
// jika $a bukan string, maka tampilkan '$a bukan string'  
  
if (is_string($a)) echo "\$a bertipe string<br>";  
else echo "\$a bukan string<br>";  
?>
```

*Mmm.. mohon jangan pedulikan dahulu statement **if()** di atas, serta perintah **is\_string()**. Fokus Anda adalah mengerti maksud script di atas dan tujuan dari penjelasan ini.*

Ketika \$a diassign dengan 10, maka script akan memunculkan '\$a bukan string'. Setelah \$a ini diassign kembali dengan nilai suatu string, maka script berikutnya memunculkan '\$a bertipe string'.

## Operator Aritmatik

Dalam bahasa pemrograman secara umum, operator digunakan untuk memanipulasi atau melakukan proses perhitungan pada suatu nilai. Ada beberapa macam operator yang bisa digunakan, diantaranya adalah operator aritmatik, operator penggabungan string dan operator logika.

Untuk operator penggabungan string, sudah pernah kita gunakan di Bab IV yaitu menggunakan tanda dot (.). Operator logika akan kita bahas pada bab berikutnya karena sering digunakan pada statement kontrol. Fokus kita sekarang adalah pada operator aritmatik.

Operator aritmatik digunakan untuk operasi perhitungan yang melibatkan nilai berupa bilangan. Namun bisa pula diberikan pada string, namun string yang berupa bilangan.

Ada beberapa macam operator aritmatik, diantaranya adalah

Operator	Makna	Contoh
+	Penjumlahan	2 + 4
-	Pengurangan	6 - 2
*	Perkalian	5 * 3
/	Pembagian	15 / 3
%	Modulus/sisa hasil bagi	43 % 10

Contoh script:

```
<?php
$penjumlahan = 2 + 4;
$pengurangan = 6 - 2;
$perkalian = 5 * 3;
$pembagian = 15 / 3;
$modulus = 5 % 2;

echo "Hasil: 2 + 4 = " . $penjumlahan."<br>";
echo "Hasil: 6 - 2 = " . $pengurangan."<br>";
echo "Hasil: 5 * 3 = " . $perkalian."<br>";
echo "Hasil: 15 / 3 = " . $pembagian."<br>";
echo "Hasil: 5 % 2 = " . $modulus;
?>
```

## Tingkat Presedensi

Harap hati-hati dalam menggunakan operator aritmatik, terutama jika kita menggunakan lebih dari satu operator yang berbeda dalam satu statement perhitungan, sebagai contoh script berikut ini:

```
<?php

$a = 3 + 4 * 5 - 6;
echo $a;

?>
```

Apabila script di atas dijalankan, maka hasil yang muncul bukan 29, tapi 17. Mengapa demikian? Ya... karena operasi aritmatik yang dikerjakan terlebih dahulu adalah perkalian (\*). Mengapa? Perkalian memiliki tingkat presedensi yang lebih tinggi daripada + dan -. Setelah perkalian dikerjakan, trus... yang dikerjakan operasi yang mana lagi? +



atau -. Keduanya, + dan -, memiliki tingkat presedensi yang sama. Nah... kalo demikian, maka yang dikerjakan lebih dahulu adalah yang terletak di bagian yang lebih kiri, yaitu +.

Bagaimana dengan operator pembagian (/)? Operator ini memiliki tingkat presedensi yang sama dengan (\*). Keduanya memiliki tingkat presedensi yang lebih tinggi daripada + dan -. Sedangkan operator modulo (%) levelnya juga sama dengan \* dan /.

Trus... dari script di atas, misalkan kita ingin yang dikerjakan adalah yang penjumlahan terlebih dulu bagaimana caranya? Ya... seperti dalam pelajaran matematika di SD, kita berikan tanda kurung.

```
<?php  
  
$a = (3 + 4) * 5 - 6;  
echo $a;  
  
?>
```

Paham ya... ???

## Kombinasi Operator Aritmatik dan Assignment

Selain bentuk operator aritmatik yang dibahas sebelumnya, ada juga operator yang merupakan kombinasi antara operator aritmatik dengan assignment.

Dalam pemrograman seringkali dijumpai proses yang melibatkan proses increment (kenaikan nilai). Misalkan kita menginginkan proses increment dengan tingkat kenaikan 1, maka perintah yang dituliskan dapat berupa

```
$counter = $counter + 1;
```

Maksud dari perintah di atas adalah, nilai variabel \$counter yang baru diperoleh dari nilai \$counter yang lama ditambah 1.

dalam PHP, perintah di atas dapat ditulis dalam satu perintah singkat sebagai

```
$counter += 1;
```

Dari contoh di atas tampak bahwa operator yang digunakan (+=) merupakan gabungan dari operator aritmatik dan assignment. Berikut ini adalah bentuk-bentuk operator lain jenis ini.

Operator	Contoh	Operasi yang ekuivalen
+=	<code>\$x += 2;</code>	<code>\$x = \$x + 2;</code>
-=	<code>\$x -= 4;</code>	<code>\$x = \$x - 4;</code>
*=	<code>\$x *= 3;</code>	<code>\$x = \$x * 3;</code>
/=	<code>\$x /= 2;</code>	<code>\$x = \$x / 2;</code>
%=	<code>\$x %= 5;</code>	<code>\$x = \$x % 5;</code>
.=	<code>\$my_str.="hello";</code>	<code>\$my_str = \$my_str . "hello";</code>

## Operator Pre/Post Increment dan Decrement

Operator jenis ini merupakan pengembangan dari operator jenis sebelumnya. Operator ini hanya digunakan pada proses increment maupun decrement dengan tingkat 1.

Berikut ini adalah operator yang termasuk jenis ini:

- `$x++;`  
ekuivalen dengan `$x += 1;` atau `$x = $x + 1;`
- `$x--;`  
ekuivalen dengan `$x -= 1;` atau `$x = $x - 1;`

Contoh:

```
<?php  
  
$x = 4;  
$x++;  
  
echo "Nilai x yang baru : ". $x;  
  
$x = 4;  
$x--;  
echo "Nilai x yang baru : ". $x;  
  
?>
```

## Studi Kasus Lain:

1. Menghitung komisi salesman (Script 5.1)
2. Menghitung gaji bersih seorang karyawan (Script 5.2)

3. Mengkonversi waktu tempuh ke dalam satuan detik (Script 5.3)
4. Mengkonversi jumlah detik ke satuan jam-menit-detik (Script 5.4)
5. Menghitung selisih dua buah waktu (Script 5.5)

## Tugas Untuk Anda

### Tugas 1

Ada seorang nasabah bank yang menabung di bank X dengan saldo awal Rp. 1.000.000,-. Bank X menerapkan kebijakan bunga 3% perbulan dari saldo awal tabungan. Hitunglah jumlah saldo akhir nasabah tersebut setelah 11 bulan. Lengkapilah script berikut ini untuk mengerjakan kasus ini!

```
<?php
$saldoAwal = 1000000;
$bunga = 0.03;
$bulan = 11;
$saldoAkhir = ...;

echo "Saldo akhir setelah ".$bulan." bulan adalah : Rp. ".$saldoAkhir. ",-";
?>
```

### Tugas 2

Lengkapi script berikut ini untuk menampilkan hasil bagi bilangan 100 dengan 3 beserta sisanya.

```
<?php
$bilangan = 100;
$pembagi = 3;
.
.
.
echo $bilangan." dibagi dengan ".$pembagi." adalah ".$hasilBagi." sisa ".$sisBagi;
?>
```

Tampilan outputnya:

**“100 dibagi dengan 3 adalah 33 sisa 1”**

### Tugas 3

Ibu ingin mengambil uang tabungan sejumlah Rp. 1.575.250,- yang dimilikinya di sebuah bank. Misalkan pada saat itu uang pecahan yang berlaku adalah Rp. 100.000,-; Rp. 50.000,-; Rp. 20.000,-; Rp. 5.000,-; Rp. 100,- dan Rp. 50. Dengan menggunakan script PHP, tentukan banyaknya masing-masing uang pecahan yang diperoleh ibu tadi!

Lengkapi script berikut:

```
<?php
$jumlahUang = 1575250;
.
.
.
echo "Jumlah Rp. 100.000 : ".$a. "<br />";
echo "Jumlah Rp. 50.000 : ".$b. "<br />";
echo "Jumlah Rp. 20.000 : ".$c. "<br />";
echo "Jumlah Rp. 5.000 : ".$d. "<br />";
echo "Jumlah Rp. 100 : ".$e. "<br />";
echo "Jumlah Rp. 50 : ".$f. "<br />";
?>
```

Keterangan :

\$a adalah variabel yang menyatakan jumlah pecahan Rp. 100.000,-  
\$b adalah variabel yang menyatakan jumlah pecahan Rp. 50.000,-  
\$c adalah variabel yang menyatakan jumlah pecahan Rp. 20.000,-  
\$d adalah variabel yang menyatakan jumlah pecahan Rp. 5.000,-  
\$e adalah variabel yang menyatakan jumlah pecahan Rp. 100,-  
\$f adalah variabel yang menyatakan jumlah pecahan Rp. 50,-

---

#### Ketentuan Pengiriman Tugas:

Tugas dikirim dalam bentuk script yang dikompresi menjadi satu lalu dikirim ke email: [rosihanari@gmail.com](mailto:rosihanari@gmail.com) dengan subject: TUGAS-BAB 5 KURSUS (nama anda)

## BAB 6

# POST dan GET Request

---

Pada bab sebelumnya, kita telah mengenal proses assignment. Anda dapat mengassign suatu nilai kepada suatu variabel. Proses assignment yang telah kita pelajari adalah dengan cara memberikan perintah sbb:

```
$namaVariabel = nilai;
```

Di dalam scriptnya secara langsung. Misalnya:

```
<?php  
  
$bilangan1 = 2;  
$bilangan2 = 6;  
  
$jumlah = $bilangan1 + $bilangan2;  
echo "Hasil penjumlahannya adalah ".$jumlah;  
  
?>
```

Nah... sekarang kita coba kembangkan sedikit. Pada script di atas kita hanya bisa menjumlahkan bilangan 2 dan 6. Kalaupun kita ingin menjumlahkan bilangan yang lain, maka kita harus mengedit scriptnya dan mengubah kedua bilangan. Sekarang... bagaimana kalau kita kembangkan script di atas supaya bisa menjumlahkan bilangan-bilangan lain tanpa mengubah scriptnya supaya lebih interaktif.

Untuk bisa melakukan hal di atas, Anda terlebih dahulu harus telah memahami POST dan GET Request.

## POST Request

POST request adalah proses mengirim data (submit) melalui form yang berasal dari (X)HTML. Karena PHP adalah bahasa program atau scripting, maka PHP bisa digunakan untuk menerima data hasil submit tersebut. Data yang diterima nantinya bisa diproses dalam script PHP.

Untuk menerima data dari proses submit form dalam PHP, kita gunakan statement assignment berikut ini.

```
$namaVariabel = $_POST['nama komponen dalam form'];
```

Dalam hal ini, nama variabel boleh sama atau tidak sama dengan nama komponen dalam form. Namun... sangat disarankan, Anda membuat nama variabel yang berbeda dari nama komponen formnya untuk mencegah hal-hal yang tidak diinginkan, misalnya hacking.

Perhatikan contoh berikut ini:

Dalam contoh ini, kita coba membuat form HTML yang di dalamnya terdapat dua komponen input dua buah bilangan.

#### **input.htm**

```
<html>
<head>
    <title>Contoh Request POST</title>
</head>
<body>
    <h1>Input dua bilangan</h1>

    <form method="post" action="proses.php">
        Masukkan Bilangan 1 <input type="text" name="bil1" /><br />
        Masukkan Bilangan 2 <input type="text" name="bil2" /><br />
        <input type="submit" name="submit" value="Submit" />
    </form>

</body>
</html>
```

Pada form di atas dapat kita lihat bahwa terdapat dua komponen dalam form, yang masing-masing memiliki nama **'bil1'** dan **'bil2'** (perhatikan atribut **name="..."** pada komponen). Keduanya kebetulan merupakan komponen form berbentuk text box. Selanjutnya, perhatikan bagian atribut **action="proses.php"**. Apa maksud dari atribut tersebut? Atribut ini memiliki makna bahwa apabila tombol submit tersebut diklik, maka data yang diisikan pada form nantinya akan dikirim dan diolah pada script bernama **proses.php**.

Sekarang, kita coba buat script proses.php nya. Dalam script ini, kita coba untuk membaca nilai yang diisikan ke dalam form di atas, lalu menampilkannya ke browser.

**proses.php**

```
<?php
$bilangan1 = $_POST['bil1'];
$bilangan2 = $_POST['bil2'];

echo "<p>Anda telah memasukkan bilangan pertama = ".$bilangan1. "</p>";
echo "<p>Anda telah memasukkan bilangan kedua = ".$bilangan2. "</p>";

?>
```

Anda juga menggabungkan script di atas dengan struktur XHTML yang lebih lengkap seperti di bawah ini

**proses.php**

```
<?php
$bilangan1 = $_POST['bil1'];
$bilangan2 = $_POST['bil2'];
?>

<html>
  <head>
    <title>Contoh Request POST</title>
  </head>
  <body>
    <h1>Input dua bilangan</h1>

    <?php
    echo "<p>Anda telah memasukkan bilangan pertama = ".$bilangan1. "</p>";
    echo "<p>Anda telah memasukkan bilangan kedua = ".$bilangan2. "</p>";
    ?>

  </body>
</html>
```

Selain cara di atas yaitu menyimpan nilai dari `$_POST[]` ke variabel tertentu, Anda dapat pula langsung menggunakan `$_POST[]` untuk diproses tanpa disimpan ke variabel tertentu. Perhatikan contoh berikut ini

**proses.php**

```
<html>
  <head>
    <title>Contoh Request POST</title>
  </head>
  <body>
    <h1>Input dua bilangan</h1>

    <?php
    echo "<p>Anda telah memasukkan bilangan pertama = " . $_POST['bil1'] . "</p>";
    echo "<p>Anda telah memasukkan bilangan kedua = " . $_POST['bil2'] . "</p>";
    ?>

    </body>
</html>
```

Sekarang kita tinjau contoh kedua. Pada contoh pertama kita buat script proses.php hanya sekedar menampilkan dua bilangan yang diinputkan pada form. Selanjutnya kita buat script berikutnya, misalkan masih tetap sama nama filenya yaitu proses.php akan tetapi digunakan untuk menampilkan hasil penjumlahan dua bilangan yang diinputkan tadi.

Adapun konsepnya hampir sama, yaitu pertama baca terlebih dahulu bilangan-bilangannya lalu jumlahkan keduanya dan hasil penjumlahannya kita tampilkan.

**proses.php**

```
<?php
$bilangan1 = $_POST['bil1'];
$bilangan2 = $_POST['bil2'];
$jumlah = $bilangan1 + $bilangan2;
?>

<html>
  <head>
    <title>Contoh Request POST</title>
  </head>
  <body>
    <h1>Input dua bilangan</h1>

    <?php
    echo "<p>Anda telah memasukkan bilangan pertama = " . $bilangan1 . "</p>";
    echo "<p>Anda telah memasukkan bilangan kedua = " . $bilangan2 . "</p>";
    echo "<p>Hasil penjumlahannya adalah " . $jumlah . "</p>";
```



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

## GET Request

Apabila POST request adalah proses merequest data melalui form, maka GET request adalah melalui URL.

Terkadang ketika browsing di internet, kita temukan URL atau link yang berbentuk misalnya seperti berikut ini

<http://situs.com/script.php?a=4&b=6>

Perhatikan bagian URL yang saya kasih bold tersebut. Bagian tersebut dinamakan parameter. Konsep GET Request pada prinsipnya mirip dengan POST. Nah... pada contoh URL di atas maksudnya adalah mensubmit atau mengirim data ke script bernama script.php. Apa yang dikirim? Yang dikirim adalah nilai a = 4 dan b = 6. Nilai yang dikirim tersebut nantinya akan dijadikan sebagai input pada proses yang terjadi di script.php.

Selanjutnya bagaimana membaca input dari GET request tersebut? Caranya adalah menggunakan `$_GET['nama parameter'];`

Akan saya berikan contoh script untuk menjumlahkan dua buah bilangan yang diinput melalui GET request.

### proses.php

```
<?php
$bilangan1 = $_GET['bil1'];
$bilangan2 = $_GET['bil2'];
$jumlah = $bilangan1 + $bilangan2;
?>

<html>
<head>
<title>Contoh Request GET</title>
</head>
<body>
<h1>Input dua bilangan</h1>
<?php
echo "<p>Anda telah memasukkan bilangan pertama = ".$bilangan1."</p>";
```

```
echo "<p>Anda telah memasukkan bilangan kedua = ".$bilangan2. "</p>";  
echo "<p>Hasil penjumlahannya adalah ".$jumlah. "</p>";  
?>  
</body>  
</html>
```

Lantas bagaimana menjumlahkan dua bilangan menggunakan script di atas yang menggunakan GET request? Caranya cukup mengetikkan URL berikut ini

<http://localhost/proses.php?bil1=100&bil2=150>

URL di atas akan menjumlahkan dua bilangan yaitu 100 (bil1) dan 150 (bil2).

Catatan:

Tanda ampersand (&) digunakan untuk memisahkan antar parameter.

Anda juga menjalankan script di atas melalui sebuah link.

#### jumlah.htm

```
<html>  
  <head>  
    <title>Contoh Request GET</title>  
  </head>  
  <body>  
    <a href='http://localhost/proses.php?bil1=100&bil2=150'>Klik di sini</a>  
  </body>  
</html>
```

## Studi Kasus yang Lain

1. Script untuk membaca input yang berasal dari berbagai komponen form (Script 6.1)
2. Script menghitung komisi salesman berdasarkan nilai penjualannya yang diinput melalui form (Script 6.2)
3. Script menghitung selisih waktu dalam detik dari 2 buah waktu yang diinput melalui form (Script 6.3)

## Tugas Untuk Anda

1. Kembangkanlah script yang telah diberikan pada Tugas 1 Modul 5 dengan menambahkan form input untuk memasukkan saldo awal, besar bunga perbulan dan juga lama bulan. Jangan lupa untuk menambahkan tombol submit dan reset. Apabila tombol submit diklik, maka semua data input akan diproses ke script PHP untuj diolah menghasilkan saldo akhir pada bulan tertentu.
2. Analog soal no. 1. Terapkan pada soal Tugas 2 Modul 5
3. Analog soal no. 1 dan 2. Terapkan pada soal Tugas 3 Modul 5
4. Rancanglah sebuah form untuk pendaftaran online mahasiswa baru dalam universitas X. Data yang nantinya dimasukkan dalam form pendaftaran adalah: Nama Lengkap (text box), Tempat Lahir (text box), Tanggal Lahir (gunakan 3 buah combo box masing-masing untuk tanggal: 1-31, bulan: 1-12 dan tahun: 1970-1987), Alamat Rumah (text area), Jenis Kelamin (gunakan radiobutton: pria/wanita), Asal Sekolah (text box), Nilai UAN (text box). Tambahkan pula sebuah button dan reset.

Apabila data sudah diisi dan selanjutnya diklik tombol submit, maka kirim data ke sebuah script PHP untuk menampilkan apa yang telah diisikan dalam form tadi.

Contoh output script bila nama yang diisikan adalah 'Joko'.

Terimakasih **Joko** sudah mengisi form pendaftaran.

Nama Lengkap	: Joko
Tempat Lahir	: XXX
Tanggal Lahir	: TGL-BLN-THN
Alamat Rumah	: XXX
Jenis Kelamin	: Pria
Asal Sekolah	: XXX
Nilai UAN	: XXX

5. Setelah Anda selesai membuat script dan form pada no. 4 di atas, coba ubah **method='post'** pada formnya menjadi **method='get'**. Ubah pula `$_POST[]` dalam script PHP nya menjadi `$_GET[]`. Masih bisakah scriptnya bekerja? Coba amati efek perubahan tersebut dan jelaskan apa akibat diberikannya `method='get'` pada form?

**Catatan:**

Anda bisa mempelajari cara membuat komponen form HTML yang lain dalam tutorial HTML yang ada di <http://rosihanari.net/web-tutorial>

## BAB 7

# Statement Kontrol – IF

---

Sekarang kita masuk ke bab yang sangat penting dalam programming yaitu statement kontrol. Mengapa sangat penting? Ya... karena dengan statement ini kita bisa mengontrol aliran program/script.

Pada contoh-contoh script yang sudah saya berikan dalam bab-bab sebelumnya hanya terdiri dari satu aliran program saja, yaitu top – down. Artinya top – down apa? Artinya setiap perintah yang ada dijalankan satu-persatu mulai dari atas sampai dengan paling bawah, dan selesai. Kita ambil contoh script PHP berikut ini yang digunakan untuk menampilkan luas sebuah persegi panjang dengan panjang 10 satuan luas dan lebar 15 satuan luas.

```
<?php
$panjang = 10;
$lebar = 15;

$luas = $panjang * $lebar;
echo "Luas persegi panjang = " . $luas . " satuan luas";
?>
```

Script di atas menggunakan sebuah aliran saja. Pertama jalankan perintah paling atas (\$panjang = 10;), lalu jalankan perintah kedua (\$lebar = 15), kemudian jalankan perintah untuk menghitung luas, dan terakhir tampilkan luasnya, selesai.

Namun... tidak semua script/program hanya terdiri dari sebuah aliran saja yang sederhana seperti di atas. Terkadang atau bahkan sering nantinya kita membuat script yang di dalamnya terdapat perintah yang harus diulang-ulang sampai berapa kali, terkadang juga ada perintah yang hanya dijalankan pada kondisi atau berdasarkan syarat tertentu saja. Hal ini kita bisa atur dengan menggunakan statement kontrol.

Dalam dunia pemrograman umumnya, terdapat 2 jenis statement kontrol yaitu: **statement kontrol kondisional** (bersyarat) dan **statement kontrol perulangan** (looping).

Statement kontrol kondisional adalah statement kontrol yang digunakan untuk mengatur kapan suatu perintah akan dijalankan. Dengan statement ini kita bisa mengatur kapan suatu perintah akan dijalankan, yaitu ketika telah dipenuhinya suatu syarat tertentu.

Sedangkan statement kontrol perulangan digunakan untuk mengatur perintah yang dijalankan secara berulang-ulang.

Dalam PHP, terdapat dua buah statement kontrol yang termasuk statement kontrol kondisional, yaitu IF dan CASE OF. Sedangkan yang termasuk statement kontrol perulangan adalah: FOR, WHILE, DO WHILE dan FOREACH.

Untuk mempermudah pembahasan masing-masing statement kontrol tersebut, akan dibahas secara bab per bab saja. Nah... pada bab 7 ini pembahasan hanya difokuskan pada statement kontrol IF. ... *are u ready to learn?* ☺

OK... pembahasan tentang IF ini akan saya mulai dari pertanyaan berikut ini

Apa yang Anda pahami bila menjumpai pernyataan di bawah ini?

Jika **sore ini hujan**, maka **saya akan menghabiskan waktu di rumah**.

Pernyataan di atas mengandung sebab akibat. Bila Anda lihat, keterangan 'sore ini hujan' adalah merupakan suatu syarat. Apa akibatnya bila syarat tersebut terpenuhi? Ya... 'saya akan menghabiskan waktu di rumah'. Dengan demikian, 'saya akan menghabiskan waktu di rumah' ini hanya akan dilakukan apabila syaratnya terpenuhi atau 'sore ini hujan'.

Dalam dunia logika, istilah 'terpenuhinya suatu syarat' dapat dikatakan sebagai 'syarat tersebut bernilai BENAR atau TRUE'.

Nah... bentuk pernyataan 'jika... maka...' pada contoh di atas dapat diadopsi dalam programming. Untuk menyatakan pernyataan tersebut dalam programming, khususnya PHP, kita bisa menggunakan statement kontrol IF

Adapun sintaks atau aturan penulisan IF nya adalah sebagai berikut:

```
if (syarat)
{
    statement;
}
```

Catatan: Kata 'if' harus ditulis dalam huruf kecil semua.

Dalam sintaks di atas, bagian 'statement' akan dijalankan atau dilakukan jika 'syarat' terpenuhi atau 'syarat' bernilai benar/true. Terus... bagaimana bila 'syarat' tidak terpenuhi? Ya...bagian 'statement' tidak akan dijalankan atau dalam hal ini tidak

melakukan apa-apa. Dengan demikian, nilai 'syarat' haruslah hanya ada dua kemungkinan, yaitu BENAR atau SALAH itu saja.

## Operator Relasional

Terkadang untuk menyatakan suatu syarat, kita menggunakan operator pembandingan atau relasional. Hasil penggunaan operator relasional ini akan diperoleh nilai BENAR atau SALAH. Berikut ini beberapa operator relasional yang dapat digunakan

Simbol	Makna
<	Lebih kecil dari
>	Lebih besar dari
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan
==	Sama dengan
!=	Tidak sama dengan

Perhatikan contoh penggunaan operator relasional pada statement IF berikut ini

```
<?php

$bil = 10;
if ($bil > 5)
{
    echo "Bilangan lebih besar dari 5";
}

?>
```

Pada script di atas, nilai dari variabel \$bil adalah 10. Selanjutnya terdapat statement IF yang di dalamnya terdapat syarat: jika nilai \$bil lebih besar dari 5, maka tampilkan teks 'Bilangan lebih besar dari 5'. Dalam hal ini syarat bernilai BENAR, karena 10 > 5. Karena syarat bernilai BENAR, maka perintah echo " Bilangan lebih besar dari 5"; akan dijalankan.

Namun.. coba perhatikan contoh script berikut ini:

```
<?php

$bil = 10;
if ($bil == 5)
{
```

```
    echo "Bilangan sama dengan 5";  
}  
  
?>
```

Pada script di atas, syarat bernilai SALAH, karena 10 tidak sama dengan 5. Dengan demikian perintah echo "Bilangan sama dengan 5"; tidak akan dijalankan.

## Operator Logika

Untuk menyatakan suatu syarat, kita dapat menggunakan operator logika. Ketika Anda menempuh sekolah SMA kelas 1, tentu Anda pernah belajar tentang logika dalam matematika. Masih ingatkah Anda dengan operator DAN, ATAU dan juga NEGASI, serta tabel kebenaran BENAR-SALAH? Ya... jangan dikira materi tersebut tidak ada gunanya. Materi tersebut akan dipakai dalam konsep pemrograman.

Dalam pemrograman PHP, terdapat beberapa operator logika yang bisa digunakan:

Operator	Makna
&&	AND
	OR
!	Negasi

Sedangkan berikut ini tabel kebenaran dari masing-masing operator

Tabel hasil operasi logika AND

Operand 1	Operand 2	Hasil
BENAR	BENAR	BENAR
BENAR	SALAH	SALAH
SALAH	BENAR	SALAH
SALAH	SALAH	SALAH

Tabel hasil operasi logika OR

Operand 1	Operand 2	Hasil
BENAR	BENAR	BENAR
BENAR	SALAH	BENAR
SALAH	BENAR	BENAR
SALAH	SALAH	SALAH



Tabel hasil operasi logika Negasi (NOT)

Operand	Hasil
BENAR	SALAH
BENAR	SALAH
SALAH	BENAR
SALAH	BENAR

Contoh penggunaan operator logika

```
<?php
$hobi = "makan";

if (($hobi == "makan") || ($hobi == "ngemil"))
{
    echo "Awat... jangan banyak-banyak, ntar kebanyakan kolesterol<br />";
    echo "Serta banyak olahraga ya...";
}

?>
```

Pada script di atas, nilai \$hobi adalah "makan", sedangkan untuk syarat terdapat dua keadaan yaitu \$hobi == "makan" serta \$hobi == "ngemil". Diantara kedua keadaan tersebut terdapat operator OR. Keadaan pertama dalam hal ini bernilai BENAR sedangkan keadaan kedua bernilai SALAH. Sehingga apabila kita lihat tabel kebenaran, BENAR OR SALAH dihasilkan BENAR. Dengan demikian syarat bernilai BENAR dan akibatnya perintah echo "Awat... jangan banyak-banyak, ntar kebanyakan kolesterol"; akan dijalankan.

### Bentuk IF yang lain

Selain bentuk IF di atas, terdapat pula bentuk IF seperti di bawah ini

```
if (syarat)
{
    statement 1
}
else
{
    statement 2
}
```

Bentuk if IF di atas, statement 1 akan dijalankan apabila nilai 'syarat' adalah BENAR. Sedangkan bila 'syarat' bernilai SALAH maka yang dijalankan adalah statement 2.

Contoh script:

```
<?php  
  
$bil = 10;  
  
if ($bil == 10)  
{  
    echo "Bilangan sama dengan 10";  
}  
else  
{  
    echo "Bilangan tidak sama dengan 10";  
}  
  
?>
```

Script di atas akan menampilkan "Bilangan sama dengan 10". Sekarang perhatikan script berikut ini

```
<?php  
  
$bil = 12;  
  
if ($bil == 10)  
{  
    echo "Bilangan sama dengan 10";  
}  
else  
{  
    echo "Bilangan tidak sama dengan 10";  
}  
  
?>
```

Script di atas akan menampilkan "Bilangan tidak sama dengan 10".

Selain bentuk kedua IF di atas, terdapat pula bentuk IF seperti di bawah ini:

```
if (syarat 1)  
{  
    Statement 1;  
}
```

```
else if (syarat 2)
{
    Statement 2
}
else if (syarat 3)
{
    Statement 3
}
.
.
else {
    Statement X
}
```

Pada bentuk IF di atas, Statement 1 akan dijalankan apabila 'syarat 1' bernilai BENAR. Jika 'syarat 1' bernilai SALAH, maka akan dicek 'syarat 2'. Jika 'syarat 2' BENAR maka akan dijalankan Statement 2, begitu seterusnya. Dan apabila ternyata tidak ada satupun syarat yang terpenuhi, barulah Statement X akan dikerjakan.

Contoh script:

```
<?php

$nilai = 70;

if ($nilai > 50)
{
    echo "Nilai lebih besar dari 50";
}
else if ($nilai < 50)
{
    echo "Nilai lebih kecil dari 50";
}
else
{
    echo "Nilai sama dengan 50";
}

?>
```

Script di atas akan menghasilkan "Nilai lebih besar dari 50" karena syarat '\$nilai > 50' bernilai BENAR. Begitu terdapat syarat yang bernilai benar, maka syarat yang terletak di bawahnya tidak akan dicek.

Selanjutnya bagaimana dengan script berikut ini?

```
<?php

$nilai = 50;

if ($nilai > 50)
{
    echo "Nilai lebih besar dari 50";
}
else if ($nilai < 50)
{
    echo "Nilai lebih kecil dari 50";
}

?>
```

Pada script di atas, syarat '\$nilai > 50' bernilai SALAH. Karena syarat ini bernilai SALAH, maka selanjutnya akan dicek syarat '\$nilai < 50'. Syarat inipun juga SALAH. Dengan demikian hasil script di atas tidak akan muncul apa-apa, karena dalam hal ini tidak terdapat alternatif terakhir 'else'. Beda halnya pada script berikut ini:

```
<?php

$nilai = 50;

if ($nilai > 50)
{
    echo "Nilai lebih besar dari 50";
}
else if ($nilai < 50)
{
    echo "Nilai lebih kecil dari 50";
}
else
{
    echo "Nilai sama dengan 50";
}

?>
```

Pada script di atas, hasil dari script adalah "Nilai sama dengan 50".

## Contoh-contoh Aplikasi Script dengan IF.

**Script untuk menentukan apakah suatu huruf termasuk huruf vokal (asumsi: huruf yang digunakan adalah kapital)**

```
<?php

$huruf = "A";

if (($huruf == "A") || ($huruf == "I") || ($huruf == "U") || ($huruf == "E") ||
    ($huruf == "O"))
{
    echo "Termasuk huruf vokal";
}
else
{
    echo "Tidak termasuk huruf vokal";
}
?>
```

## Script untuk mencari bilangan terbesar dari 3 buah bilangan: 6, 9 dan 10.

Bagaimana ide untuk mencari bilangan terbesar dari ketiga bilangan tersebut? Idennya adalah kita membandingkan setiap bilangannya.

Langkahnya:

1. Kita cari terlebih dahulu bilangan terbesar di antara bilangan pertama dan kedua (misalkan dinamakan 'max')

Untuk mencari 'max' caranya dengan membandingkan bilangan pertama dengan kedua.

Jika bilangan pertama > bilangan kedua, maka nilai 'max' adalah bilangan pertama. Jika tidak, maka 'max' adalah bilangan kedua.

2. Setelah kita tahu bilangan terbesar di antara bilangan pertama dan kedua ('max'), langkah berikutnya kita cari bilangan terbesar di antara 'max' dengan bilangan ketiga. Nilai terbesar dari ketiga bilangan adalah nilai yang terbesar di antara bilangan ketiga dengan 'max'.

Untuk mencari bilangan terbesar antara 'max' dengan bilangan ketiga, caranya sama seperti langkah pertama.

```
<?php
$bil1 = 6;
$bil2 = 9;
$bil3 = 10;

if ($bil1 > $bil2)
{
    $max = $bil1;
}
else
{
    $max = $bil2;
}

if ($max > $bil3)
{
    $maxSemua = $max;
}
else
{
    $maxSemua = $bil3;
}

echo "Nilai terbesar dari ketiga bilangan adalah ".$maxSemua;

?>
```

### Script untuk mengilustrasikan proses login.

Script berikut ini akan mengilustrasikan proses login. Sebagai contoh, misalkan user yang bisa login hanyalah user bernama 'joko' (password: 'jokokusayang') serta user bernama 'amir' (password: 'amirkusayang').

Konsep login pada prinsipnya pengecekan nama user dan password saja. Jika nama user dan passwordnya cocok maka login sukses.

Pertama kita buat terlebih dahulu form loginnya.

```
<form method="post" action="login.php">
Username <input type="text" name="username"> <br />
Password <input type="password" name="pass"> <br />
<input type="submit" name="submit" value="Submit">
</form>
```

Selanjutnya kita buat script login.php

```
<?php
$username = $_POST['username'];
$password = $_POST['pass'];

if (((($username == "joko") && ($password == "jokokusayang")) ||
    (($username == "amir") && ($password == "amirkusayang"))))
{
    echo "Login sukses";
}
else
{
    echo "Login gagal";
}

?>
```

Script di atas dapat kita pahami bahwa proses login akan sukses bila username yang dimasukkan adalah 'joko' dan passwordnya 'jokokusayang', atau usernamenya 'amir' dan passwordnya 'amirkusayang'. Dengan demikian, andaikan usernamenya benar, misalkan 'joko', namun passwordnya salah, maka login juga gagal.

### Script untuk mengkonversi nilai angka ke huruf.

Misalkan pada suatu universitas menerapkan kebijakan konversi dari nilai angka ke huruf sebagai berikut:

Nilai Angka	Nilai Huruf
80 s/d 100	A
70 s/d 79	B
60 s/d 69	C
50 s/d 59	D
0 s/d 49	E

Nah.. kita akan coba membuat script PHP untuk melakukan konversi tersebut.

Dalam hal ini, nilai angka yang diberikan 0 s/d 100. Dengan kata lain, konversi ke nilai huruf ini akan dilakukan apabila nilai angkanya 0 s/d 100. Untuk itu, dalam script yang akan dibuat terlebih dahulu akan dilakukan pengecekan apakah nilai yang diberikan 0 s/d 100 atau tidak.

```
<?php
$nilaiAngka = 87;
```

```
if (($nilaiAngka < 0) || ($nilaiAngka > 100))
{
    echo "Nilai angka yang diberikan salah";
}
else
{
    if (($nilaiAngka >= 80) && ($nilaiAngka <= 100))
    {
        $nilaiHuruf = "A";
    }
    else if (($nilaiAngka >= 70) && ($nilaiAngka <= 79))
    {
        $nilaiHuruf = "B";
    }
    else if (($nilaiAngka >= 60) && ($nilaiAngka <= 69))
    {
        $nilaiHuruf = "C";
    }
    else if (($nilaiAngka >= 50) && ($nilaiAngka <= 59))
    {
        $nilaiHuruf = "D";
    }
    else if (($nilaiAngka >= 0) && ($nilaiAngka <= 49))
    {
        $nilaiHuruf = "E";
    }

    echo "Nilai hurufnya adalah ".$nilaiHuruf;
}

?>
```

Script di atas akan ekuivalen dengan script berikut ini

```
<?php
$nilaiAngka = 87;

if (($nilaiAngka < 0) || ($nilaiAngka > 100))
{
    echo "Nilai angka yang diberikan salah";
}
else
```



```
{  
  
    if ($nilaiAngka >= 80)  
    {  
        $nilaiHuruf = "A";  
    }  
    else if ($nilaiAngka >= 70)  
    {  
        $nilaiHuruf = "B";  
    }  
    else if ($nilaiAngka >= 60)  
    {  
        $nilaiHuruf = "C";  
    }  
    else if ($nilaiAngka >= 50)  
    {  
        $nilaiHuruf = "D";  
    }  
    else if ($nilaiAngka >= 0)  
    {  
        $nilaiHuruf = "E";  
    }  
  
    echo "Nilai hurufnya adalah ".$nilaiHuruf;  
}  
  
?>
```

## Studi Kasus Lain

1. Beberapa alternatif script untuk mencari nilai terbesar dari 3 buah bilangan (Script 7-1)
2. Script untuk menentukan suatu bilangan termasuk bilangan negatif, positif atau nol (Script 7-2)
3. Mengarahkan submit form ke bagian tertentu dalam suatu script (Script 7-3)
4. Simulasi membuat menu pilihan (Script 7-4)

## Tugas Untuk Anda

1. Buatlah form untuk memasukkan bilangan yang menyatakan tahun. Setelah form tersebut disubmit, maka akan muncul apakah tahun tersebut termasuk tahun kabisat atau tidak. Gunakan script PHP untuk membuat hal ini.
2. Karyawan honorer di perusahaan XXX digaji berdasarkan jumlah jam kerjanya selama satu minggu. Upah per jamnya adalah Rp. 2.000,-. Bila jumlah jam kerja selama satu minggunya lebih besar dari 48 jam, maka sisanya dianggap jam lembur dengan upah per jam lemburnya adalah Rp. 3.000,-

Buatlah form untuk memasukkan jumlah jam kerja selama satu minggu seorang karyawan. Setelah form disubmit, maka akan tampil jumlah upah yang diterima karyawan tersebut. Gunakan script PHP untuk membuat hal ini.

3. Berat badan ideal ada kaitannya dengan tinggi badan seseorang. Untuk menentukan berat badan ideal rumusnya: tinggi badan – 100, lalu hasilnya dikurangi lagi dengan 10% dari hasil pengurangan pertama.

Berdasarkan keterangan tersebut, buatlah form yang digunakan untuk mengisikan tinggi badan seseorang (cm) dan berat badannya (kg). Setelah diisi, apabila form diklik maka akan muncul keterangan “Berat badan Anda ideal” atau “Berat badan Anda tidak ideal”.

Berat badan seseorang dikatakan ideal apabila perbedaan berat badan orang tersebut terhadap berat badan idealnya tidak lebih dari 2 kg. Bila perbedaan keduanya lebih dari 2 kg, maka berat badannya dikatakan tidak ideal.

4. Buatlah form untuk memasukkan usia seseorang (dalam tahun). Setelah form tersebut diisi dan kemudian disubmit, maka akan muncul keterangan apakah usia yang diisikan tersebut termasuk usia tua, dewasa, anak-anak, dan balita dengan kriteria sbb:

Usia	Status
0 – 5	Balita
6 – 16	Anak-anak
17 – 50	Dewasa
> 50	Tua

5. Perhatikan kembali dua script yang ekuivalen untuk mengkonversi nilai angka ke huruf seperti yang pernah diberikan sebelumnya. Analisislah kedua script yang ekuivalen tersebut, dan jelaskan mengapa kedua script ekuivalen!

6. Soal analog no. 2, namun dalam hal ini terdapat 4 jenis upah perjam nya dibedakan berdasarkan golongan:

Golongan	Upah perjam
A	Rp. 4.000,-
B	Rp. 5.000,-
C	Rp. 6.000,-
D	Rp. 7.500,-

Sedangkan upah lemburnya dihitung sama untuk setiap golongan, yaitu Rp. 3.000,- per jamnya.

Buatlah form untuk mengisikan jumlah jam kerja selama seminggu, dan juga memilih golongan karyawannya (gunakan combo box). Apabila form tersebut disubmit maka akan muncul jumlah upah yang diperoleh karyawan.

## BAB 8

# Statement SWITCH

---

SWITCH termasuk statement yang dapat dipergunakan untuk menyatakan suatu pernyataan kondisional atau bersyarat seperti halnya IF.

Adapun sintaks secara umum statement SWITCH dalam PHP adalah sebagai berikut:

```
switch (var)
{
    case value1 : statement A
        .
        .
        break;
    case value2 : statement B
        .
        .
        break;
    case value3 : statement C
        .
        .
        break;
    .
    .
    default : statement D
        .
        .
}
```

Maksud dari sintaks di atas adalah: dalam hal ini nilai yang ada pada variabel ('var') akan dicek. Diantara value – value yang ada sebagai pilihan, dalam hal ini value1, value2, value3, ... dst manakah yang sesuai dengan value yang dimiliki oleh 'var'. Misalkan nilai var = value1, maka blok yang ada di statement A lah yang dijalankan. Bila nilai var = value2, maka blok statement B yang akan dijalankan. Namun bila tidak ada value yang cocok, maka barulah blok yang ada dibagian DEFAULT (statement D) yang akan dijalankan.

Catatan:

Penggunaan option DEFAULT pada SWITCH adalah optional. Kita bisa untuk tidak memberikan DEFAULT jika memang tidak diperlukan.

Bentuk sintaks SWITCH di atas ekuivalen dengan statement IF berikut ini

```
if (var == value1)
{
    statement A
    .
    .
}
else if (var == value2)
{
    statement B
    .
    .
}
else if (var == value3)
{
    statement C
    .
    .
}
else
{
    statement D
    .
    .
}
```

Apabila kita perhatikan dari konsep SWITCH, maka kita bisa menyimpulkan bahwa proses pengecekan atau pengujian persyaratan yang terjadi hanyalah proses perbandingan dalam bentuk 'sama dengan' atau '=='.

SWITCH tidak bisa digunakan untuk kondisional yang di dalamnya terdapat proses perbandingan seperti 'lebih besar dari', 'lebih kecil dari', 'lebih kecil atau sama dengan', 'lebih besar atau sama dengan' dan 'tidak sama dengan'. Melihat hal ini memang penggunaan statement IF untuk menyatakan kondisional lebih fleksibel daripada SWITCH.

Trus... berarti SWITCH tidak lebih efektif daripada IF donk. Tentu tidak. Ada kelebihan pada penggunaan SWITCH. Berikut ini kelebihanannya:

- Meskipun makna sintaks SWITCH dan IF yang diberikan di atas adalah sama, namun terdapat perbedaan dalam prosesnya.

Dalam SWITCH, persyaratan hanya diuji sekali saja. Diantara sekian pilihan value, secara otomatis PHP akan menjalankan blok dimana valuenya cocok. Proses pencarian value yang cocok ini tidak dimulai dari pilihan value paling atas namun secara otomatis langsung menuju value yang cocok.

Sedangkan pada IF.. ELSE syarat diuji beberapa kali. Proses pengujian dimulai dari syarat IF paling atas, jika tidak terpenuhi dicek dibagian IF bawahnya dst.

Akibatnya, apabila script Anda sangat kompleks atau banyak menggunakan pernyataan kondisional, penggunaan SWITCH akan menghasilkan proses yang lebih cepat daripada IF.

- Selain itu, statement yang akan dikerjakan dalam SWITCH bisa kosong atau tidak melakukan apa-apa. Contoh :

```
<?php
switch ($i) {
case 0:
case 1:
case 2:
    echo "i lebih kecil dari 3 tapi tidak negatif";
    break;
case 3:
    echo "i sama dengan 3";
}
?>
```

Nah... berikutnya, saya akan coba berikan contoh penggunaan SWITCH dalam studi kasus. Studi kasus yang saya ambil di sini adalah bagaimana menampilkan nama bulan sekarang (current month) dalam bahasa Indonesia.

Adapun ide penyelesaian dari studi kasus tersebut adalah sebagai berikut :

Langkah pertama kita harus tahu bulan sekarang. Untuk mengetahui nama bulan sekarang, kita bisa menggunakan function date("n") dalam PHP. Function tersebut akan menghasilkan bilangan bulat 1-12 yang mencerminkan angka bulan sekarang.

Trus... apakah Anda sudah punya ide berikutnya? Angka-angka tersebut mau diapain? ... Yes... Good idea.

Langkah kedua, dari angka bulan yang telah diperoleh tersebut selanjutnya kita nyatakan ke dalam bahasa Indonesia. Bagaimana menyatakannya? Ya... caranya kita gunakan logika seperti ini:

Jika angka bulan sama dengan 1, maka nama bulannya 'Januari'. Jika angka bulan sama dengan 2, maka bulannya 'Pebruari' dan seterusnya.

Untuk mengimplementasikan logika di atas, kita harus menggunakan pernyataan kondisional. Apakah kita bisa menggunakan SWITCH? Oh... tentu bisa, karena operasi perbandingan yang digunakan adalah 'sama dengan'. Sehingga bentuk script PHP nya untuk menampilkan nama bulan dalam bahasa Indonesia adalah sbb:

```
<?php
```

```
$angkaBln = date("n");
```

```
switch($angkaBln)
```

```
{
```

```
case 1 : $namaBln = "Januari";
```

```
    break;
```

```
case 2 : $namaBln = "Pebruari";
```

```
    break;
```

```
case 3 : $namaBln = "Maret";
```

```
    break;
```

```
case 4 : $namaBln = "April";
```

```
    break;
```

```
case 5 : $namaBln = "Mei";
```

```
    break;
```

```
case 6 : $namaBln = "Juni";
```

```
    break;
```

```
case 7 : $namaBln = "Juli";
```

```
    break;
```

```
case 8 : $namaBln = "Agustus";
```

```
    break;
```

```
case 9 : $namaBln = "September";
```

```
    break;
```

```
case 10: $namaBln = "Oktober";
```

```
    break;
```

```
case 11: $namaBln = "Nopember";
```

```
    break;
```

```
case 12: $namaBln = "Desember";
```

```
    break;
```

```
}
```

```
echo "Nama bulan sekarang adalah : ".$namaBln;
```

```
?>
```

Selain menggunakan `date("n");` untuk mengetahui bulan sekarang, kita juga bisa menggunakan `date("M")`. Penggunaan `date("M")` ini akan menghasilkan nama bulan akan tetapi dalam 3 huruf depan saja, misalnya 'Jan', 'Feb', 'Mar', 'Apr', ... , 'Dec'. Apabila kita menggunakan function `date("M")` ini, maka scriptnya menjadi

```
<?php
```

```
$bln = date("M");
```

```
switch($bln)
```

```
{
```

```
case "Jan" : $namaBln = "Januari";
```

```
break;
```

```
case "Feb" : $namaBln = "Pebruari";
```

```
break;
```

```
case "Mar" : $namaBln = "Maret";
```

```
break;
```

```
case "Apr" : $namaBln = "April";
```

```
break;
```

```
case "May" : $namaBln = "Mei";
```

```
break;
```

```
case "Jun" : $namaBln = "Juni";
```

```
break;
```

```
case "Jul" : $namaBln = "Juli";
```

```
break;
```

```
case "Aug" : $namaBln = "Agustus";
```

```
break;
```

```
case "Sep" : $namaBln = "September";
```

```
break;
```

```
case "Oct" : $namaBln = "Oktober";
```

```
break;
```

```
case "Nov" : $namaBln = "Nopember";
```

```
break;
```

```
case "Dec" : $namaBln = "Desember";
```

```
break;
```

```
}
```

```
echo "Nama bulan sekarang adalah : ".$namaBln;
```

```
?>
```



Perhatikan script di atas ! Karena value yang akan dicocokkan berupa string, maka setiap value pada option CASE diapit tanda petik

Catatan:

Perintah date() merupakan salah satu function. Apa itu function?, selain date() apakah ada function yang lain? Dan bagaimana cara membuat function, semuanya akan dijelaskan lebih lanjut di bab yang lain.

## Tugas Untuk Anda

1. Misalkan diberikan script berikut ini

```
<?php

$bil = 10;

switch ($bil)
{
case 5      : echo "Bilangan sama dengan 5 <br />";
              break;
case 8      : echo "Bilangan sama dengan 8 <br />";
              break;
case 10     : echo "Bilangan sama dengan 10 <br />";
case 12     : echo "Bilangan sama dengan 12 <br />";
default    : echo "Bilangan tidak sama dengan 5, 8, 10 dan 12 <br />";
              break;
}

?>
```

Dengan tanpa menjalankan script di browser, coba tebak kira-kira apa outputnya hayoo?? Bila sudah, coba bandingkan hasilnya bila dijalankan di browser. Selanjutnya simpulkan!

2. Kerjakan kembali soal no. 6 Modul 7. Gunakan konsep SWITCH!  
Keterangan:  
Gunakan komponen radiobutton dalam memilih golongan gaji.
3. Kerjakan kembali soal no. 2 Modul 7 menggunakan konsep SWITCH
4. Dengan menggunakan konsep SWITCH, buatlah script untuk membaca bulan saat ini dan tampilkan jumlah hari dalam bulan tersebut.

5. Buatlah aplikasi sederhana untuk menghitung luas beberapa bangun datar. Pertama buat halaman index.php yang isinya menu pilihan berbentuk radiobutton atau combobox yang itemnya sbb:

- Bujur sangkar
- Persegi panjang
- Lingkaran
- Segitiga

Dan sebuah tombol submit di bawah pilihan menu tersebut.

Misalkan apabila user memilih menu Bujur Sangkar dan ia mengklik submit, maka nantinya akan muncul form untuk mengisi panjang sisi bujur sangkar dan sebuah tombol submit. Jika tombol submit ini diklik maka selanjutnya akan muncul hasil luasnya. Selanjutnya bila user memilih Persegi Panjang dalam menunya dan ia mengklik submit maka akan muncul form untuk mengisi panjang dan lebar persegi panjang serta sebuah tombol submit. Bila tombol ini diklik maka akan memunculkan hasil luas persegi panjang. Begitu seterusnya....

Paham kan? Ya... sambil mengingat-ingat pelajaran di SD 😊

Catatan: Gunakan konsep SWITCH dalam menerapkan pemilihan menunya.

## BAB 9

# Statement FOR

---

Statement ini merupakan salah satu statement yang dapat digunakan untuk menyatakan proses yang diulang-ulang atau looping.

Tentu kita tahu, bahwa suatu saat looping itu harus berhenti karena tidak mungkin tidak pernah berhenti 😊 Nah.. berdasarkan hal tersebut, terdapat 2 kondisi yang dapat menghentikan looping tersebut. Coba perhatikan 2 pernyataan sehari-hari berikut ini:

*Hari ini saya makan bakso 10 kali.*

*Hari ini saya terus makan bakso sampai saya kenyang.*

Kedua pernyataan di atas sama-sama mengandung proses yang diulang-ulang yaitu 'makan bakso'. Namun proses 'makan bakso' dari kedua pernyataan tersebut berbeda cara berhentinya. Pernyataan pertama menunjukkan bahwa 'makan bakso' akan diulang-ulang sampai dengan 10 kali. Setelah sampai 10 kali, maka 'makan bakso' ini akan berhenti. Sedangkan pernyataan yang kedua, dalam hal ini jumlah berapa kali makan bakso tidak diketahui, namun yang diketahui hanyalah syarat kapan perulangan ini akan berhenti yaitu ketika 'sudah kenyang'.

Kedua bentuk looping berdasarkan kapan berhentinya di atas diadopsi ke dalam konsep pemrograman. Dalam pemrograman, khususnya PHP, nantinya juga akan dijumpai bentuk-bentuk perulangan tersebut.

**Untuk menyatakan looping yang telah diketahui jumlah perulangannya, kita bisa menggunakan statement FOR atau WHILE.**

**Sedangkan bila perulangannya tidak diketahui jumlah perulangannya atau hanya diketahui kondisi kapan dia harus berakhir maka bisa digunakan WHILE.**

Dalam bab ini kita akan bahas terlebih dahulu statement FOR.

Sintaks statement FOR adalah sebagai berikut

```
for (variabel = nilaiawal; variabel operatorlogika nilaiakhir; increment)
{
    Proses yang diulang
}
```

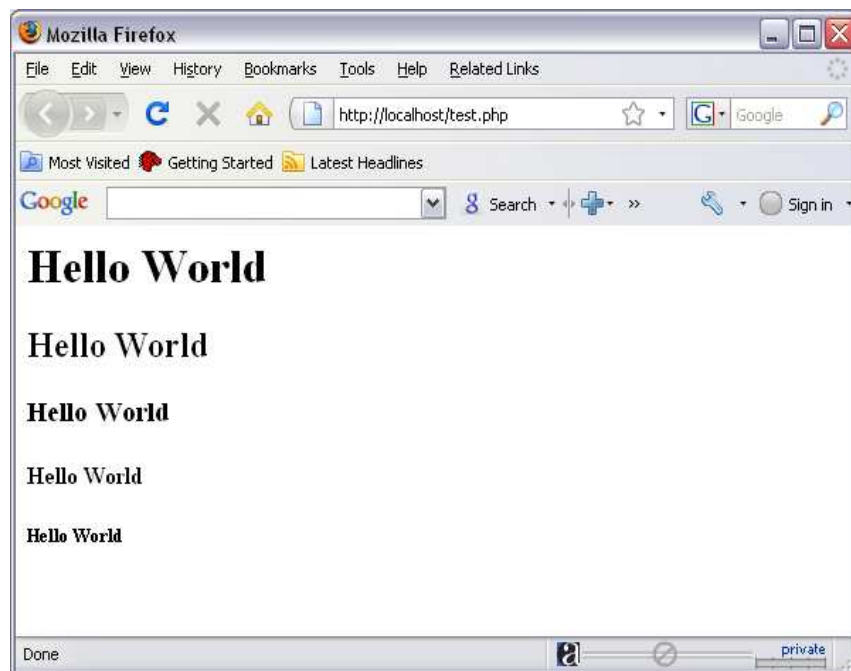
Variabel yang digunakan pada FOR di atas, dinamakan variabel counter atau pencacah.

Contoh:

Script berikut ini akan menghasilkan teks heading mulai dari heading 1, 2 s/d 5.

```
<?php  
  
$teks = "Hello World";  
  
for ($a = 1; $a <= 5; $a++)  
{  
    echo "<h".$a.">".$teks."</h".$a.">";  
}  
  
?>
```

Hasil output di atas adalah:



Pada for di atas, variabel counter-nya dalam hal ini adalah \$a. Nilai \$a ini memiliki nilai awal sama dengan 1. Selanjutnya operator increment (kenaikan) yang digunakan adalah \$a++ atau \$a = \$a + 1 (kenaikannya satu) dan nilai akhir untuk \$a adalah sama dengan 5.

Perhatikan proses loopingnya sebagai berikut:

- \$a = 1 dalam hal ini masih memenuhi kondisi (\$a <= 5), sehingga lakukan echo "<h".\$a.">".\$teks."</h".\$a.">"; atau mencetak "<h1>Hello World</h1>"
- \$a = 2 dalam hal ini masih memenuhi kondisi (\$a <= 5), sehingga lakukan echo "<h".\$a.">".\$teks."</h".\$a.">"; atau mencetak "<h2>Hello World</h2>"
- \$a = 3 dalam hal ini masih memenuhi kondisi (\$a <= 5), sehingga lakukan echo "<h".\$a.">".\$teks."</h".\$a.">"; atau mencetak "<h3>Hello World</h3>"
- \$a = 4 dalam hal ini masih memenuhi kondisi (\$a <= 5), sehingga lakukan echo "<h".\$a.">".\$teks."</h".\$a.">"; atau mencetak "<h4>Hello World</h4>"
- \$a = 5 dalam hal ini masih memenuhi kondisi (\$a <= 5), sehingga lakukan echo "<h".\$a.">".\$teks."</h".\$a.">"; atau mencetak "<h5>Hello World</h5>"
- \$a = 6 dalam hal ini **sudah tidak memenuhi** kondisi (\$a <= 5), sehingga perulangan berhenti.

Catatan:

Operator increment tidak harus menggunakan kenaikan 1, contoh

```
<?php
```

```
$teks = "Hello World";
```

```
for ($a = 1; $a <= 10; $a+=2)
{
    echo "<h".$a.">".$teks."</h".$a.">";
}
```

```
?>
```

Pada contoh di atas, increment yang digunakan adalah kenaikan 2 (\$a += 2). Adapun prosesnya adalah:

- \$a = 1 dalam hal ini masih memenuhi kondisi (\$a <= 10), sehingga lakukan echo "<h".\$a.">".\$teks."</h".\$a.">"; atau mencetak "<h1>Hello World</h1>"
- \$a = 3 dalam hal ini masih memenuhi kondisi (\$a <= 10), sehingga lakukan echo "<h".\$a.">".\$teks."</h".\$a.">"; atau mencetak "<h3>Hello World</h3>"
- \$a = 5 dalam hal ini masih memenuhi kondisi (\$a <= 10), sehingga lakukan echo "<h".\$a.">".\$teks."</h".\$a.">"; atau mencetak "<h5>Hello World</h5>"
- \$a = 7 dalam hal ini masih memenuhi kondisi (\$a <= 10), sehingga lakukan echo "<h".\$a.">".\$teks."</h".\$a.">"; atau mencetak "<h7>Hello World</h7>"
- \$a = 9 dalam hal ini masih memenuhi kondisi (\$a <= 10), sehingga lakukan echo "<h".\$a.">".\$teks."</h".\$a.">"; atau mencetak "<h9>Hello World</h9>"
- \$a = 11 dalam hal ini **sudah tidak memenuhi** kondisi (\$a <= 10), sehingga perulangan berhenti.

Catatan:

Nilai awal variabel counter tidak harus dimulai dari 1, tapi boleh dari bilangan yang lain.

Catatan:

Selain operator increment, kita juga bisa menggunakan operator decrement (turun)

Contohnya

```
<?php

$teks = "Hello World";

for ($a = 10; $a > 5; $a--)
{
    echo "<h\".$a.">".$teks."</h\".$a.">";
}

?>
```

Contoh di atas menggunakan decrement \$a-- (nilainya turun 1). Berikut ini prosesnya:

- \$a = 10 dalam hal ini masih memenuhi kondisi (\$a > 5), sehingga lakukan echo "<h\".\$a.">".\$teks."</h\".\$a.">"; atau mencetak "<h10>Hello World</h10>"
- \$a = 9 dalam hal ini masih memenuhi kondisi (\$a > 5), sehingga lakukan echo "<h\".\$a.">".\$teks."</h\".\$a.">"; atau mencetak "<h9>Hello World</h9>"
- \$a = 8 dalam hal ini masih memenuhi kondisi (\$a > 5), sehingga lakukan echo "<h\".\$a.">".\$teks."</h\".\$a.">"; atau mencetak "<h8>Hello World</h8>"
- \$a = 7 dalam hal ini masih memenuhi kondisi (\$a > 5), sehingga lakukan echo "<h\".\$a.">".\$teks."</h\".\$a.">"; atau mencetak "<h7>Hello World</h7>"
- \$a = 6 dalam hal ini masih memenuhi kondisi (\$a > 5), sehingga lakukan echo "<h\".\$a.">".\$teks."</h\".\$a.">"; atau mencetak "<h6>Hello World</h6>"
- \$a = 5 dalam hal ini **sudah tidak memenuhi** kondisi (\$a > 5), sehingga perulangan berhenti.

Anda juga bisa menggunakan operator decrement yang lain selain \$a--, misalkan \$a -= 2, \$a -= 3, dst.

### FOR di dalam FOR (*nested FOR*)

Statement FOR dapat terletak di dalam sebuah FOR, seperti contoh berikut ini

```
<?php

for ($a = 1; $a <= 3; $a++)
```

```
{
  for ($b = 1; $b <= 2; $b++)
  {
    echo "Nilai a = ".$a. " Nilai b = ".$b. "<br />";
  }
}

?>
```

Contoh di atas disebut juga “*nested* FOR tingkat 2”.

Proses yang terjadi pada script di atas adalah sbb:

- \$a = 1 (masih memenuhi \$a <= 3), maka lakukan for yang terletak di dalam
  - \$b = 1 (masih memenuhi \$b <= 2) maka cetak "Nilai a = 1 Nilai b = 1"
  - \$b = 2 (masih memenuhi \$b <= 2) maka cetak "Nilai a = 1 Nilai b = 2"
  - \$b = 3 (tidak memenuhi \$b <= 2), maka STOP
- \$a = 2 (masih memenuhi \$a <= 3), maka lakukan for yang terletak di dalam
  - \$b = 1 (masih memenuhi \$b <= 2) maka cetak "Nilai a = 2 Nilai b = 1"
  - \$b = 2 (masih memenuhi \$b <= 2) maka cetak "Nilai a = 2 Nilai b = 2"
  - \$b = 3 (tidak memenuhi \$b <= 2), maka STOP
- \$a = 3 (masih memenuhi \$a <= 3), maka lakukan for yang terletak di dalam
  - \$b = 1 (masih memenuhi \$b <= 2) maka cetak "Nilai a = 3 Nilai b = 1"
  - \$b = 2 (masih memenuhi \$b <= 2) maka cetak "Nilai a = 3 Nilai b = 2"
  - \$b = 3 (tidak memenuhi \$b <= 2), maka STOP
- \$a = 4 (tidak memenuhi \$a <= 3), maka STOP

Dengan demikian script di atas akan menghasilkan output

```
Nilai a = 1 Nilai b = 1
Nilai a = 1 Nilai b = 2
Nilai a = 2 Nilai b = 1
Nilai a = 2 Nilai b = 2
Nilai a = 3 Nilai b = 1
Nilai a = 3 Nilai b = 2
```

FOR juga dapat tersusun 3 tingkat atau lebih sesuai kebutuhan yang diinginkan programmer.

#### Contoh-contoh script menggunakan FOR:

- A. Script PHP untuk menampilkan bilangan bulat 5 s/d 100 yang berkelipatan 10

Konsep untuk menampilkan bilangan bulat tersebut kita bisa menggunakan looping. Looping dimulai dari bilangan 5, 6, 7, dst sampai dengan 100. Untuk

setiap bilangan nantinya kita cek apakah dia kelipatan 10 atau tidak (kita gunakan IF). Jika bilangan tsb merupakan kelipatan 10, maka kita tampilkan. Sedangkan jika tidak, maka abaikan. Berikut ini script PHP nya

```
<?php

for ($bil = 5; $bil <= 100; $bil++)
{
    if ($bil % 10 == 0) echo $bil. "<br />";
}

?>
```

#### B. Script PHP untuk menjumlahkan bilangan bulat 2 s/d 50

Yang dimaksud menjumlahkan di sini adalah kita mencari hasil dari  $2 + 3 + 4 + 5 + \dots + 50$

Wah... gimana konsepnya nih? Gampang... konsepnya adalah kita ambil suatu variabel sembarang terlebih dahulu, misalkan namanya 'jumlah'. Nah... pertama-tama 'jumlah' ini bernilai 0. Selanjutnya kita jumlahkan 'jumlah' dengan bilangan pertama yaitu 2, dan hasilnya kita simpan ke dalam 'jumlah' yang baru. Sehingga nilai 'jumlah' sekarang adalah  $0 + 2 = 2$ . Trus... kita jumlahkan lagi 'jumlah' dengan bilangan kedua yaitu 3. Dan lagi-lagi hasilnya kita simpan dalam 'jumlah' lagi. Sehingga hasil 'jumlah' yang baru adalah  $2 + 3 = 5$ . Berikutnya 'jumlah' kita jumlahkan lagi dengan bilangan ketiga yaitu 4, dan diperoleh 'jumlah' =  $5 + 4 = 9$ . Proses ini berlanjut sampai dengan 50.

Nah... konsep di atas bisa kita terapkan ke dalam suatu proses looping, karena di dalamnya terdapat proses yang diulang-ulang yaitu proses penjumlahan antara 'jumlah' dengan bilangan.

```
<?php

$jumlah = 0;
for ($bil = 2; $bil <= 50; $bil++)
{
    $jumlah = $jumlah + $bil;
}

echo "Hasilnya adalah ".$jumlah;

?>
```



- C. Script PHP untuk mencari banyaknya bilangan bulat mulai dari 3 s/d 127 yang merupakan kelipatan 6.

Konsep untuk membuat script ini, mirip dengan A dan B. Dalam hal ini kita harus mengecek untuk setiap bilangan apakah dia merupakan kelipatan 6 atau tidak. Jika ya, maka dia ikut dihitung. Jika tidak, maka dia diabaikan.

```
<?php
$hitung = 0;

for ($bil = 3; $bil <= 127; $bil++)
{
    if ($bil % 6 == 0) $hitung = $hitung + 1;
}

echo "Banyaknya bilangan bulat adalah ".$hitung;
?>
```

- D. Script PHP untuk membuat tabel dengan n baris dan m kolom secara otomatis

Nah... untuk kasus yang satu ini, kita harus tahu tag untuk membuat tabel terlebih dahulu. Tentu kita tahu bahwa tag untuk membuat tabel dengan 3 baris 2 kolom misalnya, adalah sebagai berikut:

```
<table border="1">
  <tr><td>...</td><td>... </td></tr>
  <tr><td>...</td><td>... </td></tr>
  <tr><td>...</td><td>... </td></tr>
</table>
```

Perhatikan baik-baik tag di atas. Untuk membuat tabel di atas, tag <td>..</td> dalam satu baris ditulis 2 kali (sesuai jumlah kolom yang diinginkan). Sedangkan <tr> dan </tr> ditulis 3 kali (sesuai jumlah baris yang diinginkan).

Konsep untuk membuat script PHP yang bisa mengenerate tabel dengan n baris dan m kolom secara otomatis adalah sebagai berikut.

Kita membuat baris pertama <tr> terlebih dahulu, kemudian dilanjutkan dengan membuat kolom <td>..</td> sebanyak jumlah kolom. Setelah selesai membuat kolom, ditutup dengan </tr>. Baris pertama selesai. Berikutnya buat baris ke dua. Pertama kita juga buat <tr>, lalu dilanjutkan dengan <td>...</td> sebanyak jumlah kolom dan diakhiri dengan </tr>. Nah.. proses ini diulang terus sampai dengan jumlah baris.

Apabila kita lihat konsep di atas, maka dapat kita lihat terdapat dua proses perulangan yaitu untuk membuat baris dan kolomnya. Sehingga dalam hal ini kita gunakan FOR di dalam FOR (*nested* FOR tingkat 2)

```
<?php

$jumlahBaris = 10;
$jumlahKolom = 6;

echo "<table border='1'>";

for ($baris = 1; $baris <= $jumlahBaris; $baris++)
{
    echo "<tr>";
    for ($kolom = 1; $kolom <= $jumlahKolom; $kolom++)
    {
        echo "<td>...</td>";
    }
    echo "</tr>";
}

echo "</table>";
?>
```

- E. Script PHP untuk menghasilkan list combo box berisi tahun mulai 1930 s/d 2008 secara otomatis

Anda mau membuat combo box berisi pilihan tahun 1930 s/d 2008? Jika secara manual, maka Anda harus membuat tag seperti di bawah ini

```
<select name="tahun">
    <option value="1930">1930</option>
    <option value="1931">1931</option>
    <option value="1932">1932</option>
    <option value="1933">1933</option>
    <option value="1934">1934</option>
    .
    .
</select>
```

Hwaduh... panjang banget yah... !!! Males dah... Eh, ngapain gak coba pakai looping saja? Kan bisa cepet?

```
<?php

echo "<select name='tahun'>";

for ($tahun = 1930; $tahun <= 2008; $tahun++)
{
    echo "<option value='". $tahun. "'>". $tahun. "</option>";
}
echo "</select>";
?>
```

## Tugas Untuk Anda

1. Tentu Anda mengenal syair lagu anak-anak berikut ini:

Anak ayam turun 10  
Anak ayam turun 10, mati satu tinggal 9  
Anak ayam turun 9, mati satu tinggal 8  
Anak ayam turun 8, mati satu tinggal 7  
Anak ayam turun 7, mati satu tinggal 6  
Anak ayam turun 6, mati satu tinggal 5  
Anak ayam turun 5, mati satu tinggal 4  
Anak ayam turun 4, mati satu tinggal 3  
Anak ayam turun 3, mati satu tinggal 2  
Anak ayam turun 2, mati satu tinggal 1  
Anak ayam turun 1, mati satu tinggal induknya

Nah.. coba buatlah script PHP untuk mengenerate syair lagu anak ayam tersebut untuk jumlah anak ayam mula-mula adalah N. Buatlah form untuk memasukkan sembarang nilai N ini, setelah itu tekan SUBMIT. Selanjutnya akan muncul baris syair seperti di atas.

Contoh di atas hanyalah untuk N = 10.

2. Seorang nasabah bank, mula-mula memiliki saldo awal Rp. 1.000.000,- di suatu bank X. Bank tersebut memiliki kebijakan memberikan bunga 3% jika besar saldo kurang dari Rp. 1.100.000,- dan 4% bila saldo lebih besar atau sama dengan Rp. 1.100.000,- . Perhitungan bunga dihitung dari besar saldo terakhir. Selain itu di setiap bulannya nasabah dikenai biaya administrasi Rp. 9.000,-. Buatlah script PHP untuk menentukan saldo akhir setelah N bulan.

Keterangan:

Buatlah form untuk mengisi saldo awalnya dan N (jangka waktu tabungan).

3. Dengan menggunakan script PHP, buatlah tampilan daftar perkalian berikut ini menggunakan looping

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
.
.
1 x 10 = 10
2 x 1 = 2
2 x 2 = 4
.
.
2 x 10 = 20
.
.
10 x 10 = 100
```

Petunjuk : Gunakan *nested* FOR 2 tingkat

4. Buatlah script PHP untuk menampilkan formasi bintang seperti berikut ini

```
*
* *
* * *
* * * *
```

Formasi bintang di atas untuk N = 4

Terlebih dahulu Anda harus membuat form untuk menerima input sembarang N dan script Anda harus menampilkan formasi bintang N tersebut.

Petunjuk:

- Coba kaitkan antara jumlah bintang untuk setiap barisnya dengan urutan barisnya.
- Gunakan *nested* FOR 2 tingkat

5. Dengan menggunakan script PHP, tampilkan semua pasangan nilai x, y dan z yang memenuhi persamaan  $x + y + z = 25$ , dengan x, y, dan z adalah bilangan asli.

Serta tampilkan pula banyaknya pasangan  $x$ ,  $y$ , dan  $z$  yang memenuhi persamaan tersebut

Contoh tampilan:

$x = 1, y = 1, z = 23$

$x = 1, y = 2, z = 22$

$x = 1, y = 3, z = 21$

.

.

.

Jumlah penyelesaian : ...

Petunjuk:

- Tentukan dahulu nilai-nilai yang mungkin dari  $x$ ,  $y$ , dan  $z$ . Berapa nilai minimum yang mungkin dari ketiganya, dan berapa maksimumnya.
- Gunakan *nested* FOR 3 tingkat

## BAB 10

# Statement WHILE

---

Pada bab 9, Anda sudah mempelajari bentuk perulangan FOR. Masihkah Anda ingat kapan kita harus menggunakan perulangan FOR? Yup... ketika kita tahu jumlah perulangannya.

Nah... pada bab 10 ini, kita akan membahas tentang perulangan WHILE. Seperti halnya FOR, statement WHILE ini juga dapat digunakan untuk menyatakan perulangan yang telah diketahui jumlah perulangan tersebut harus dilakukan. Namun... tidak hanya itu, WHILE juga dapat digunakan untuk menyatakan perulangan jika jumlah perulangannya tidak diketahui.

Dari pernyataan di atas, dapat kita simpulkan bahwa setiap bentuk perulangan FOR pasti dapat juga dinyatakan ke dalam bentuk WHILE. Namun... tidak berlaku kebalikannya, bahwa tidak semua bentuk WHILE dapat dinyatakan ke dalam FOR. Nanti akan kita lihat contohnya.

OK... bentuk umum statement WHILE adalah sbb:

```
while (syarat)
{
    // bagian yang diulang
}
```

Perhatikan bentuk umum WHILE di atas! Sebelum menuju bagian yang diulang, di atasnya terdapat syarat. Syarat inilah yang menentukan apakah bagian yang diulang akan dijalankan atau tidak. Selama syarat ini dipenuhi atau bernilai BENAR, maka bagian yang diulang akan terus dijalankan. Apabila suatu saat syarat sudah tidak terpenuhi lagi atau bernilai SALAH maka perulangan berhenti.

Contoh: Script berikut ini akan menampilkan bilangan 1 s/d 3

```
<?php
$bil = 1;
while ($bil <= 3)
{
    echo $bil. "<br />";
```

```
$bil++;  
}  
?>
```

Sekarang kita analisis prosesnya...

Nilai awal dari \$bil adalah 1. Selanjutnya terdapat while dengan syarat \$bil <= 3. Nah.. apabila dicek syaratnya, maka dalam hal ini syarat bernilai BENAR karena 1 <= 3, sehingga sekarang menuju ke bagian yang diulang. Di dalam bagian yang diulang terdapat echo \$bil, sehingga di browser akan tampil bilangan 1. Berikutnya nilai \$bil bertambah menjadi 2 (\$bil++).

Kembali lagi syaratnya dicek, dalam hal ini masih bernilai BENAR karena 2 <= 3. Akibatnya proses sekarang menuju kembali ke bagian yang diulang, yaitu tampilkan \$bil (muncul bilangan 2 dalam browser) dan \$bil bertambah satu menjadi 3.

Selanjutnya dicek kembali syaratnya apakah \$bil <= 3 dan ternyata BENAR karena 3 <= 3. Karena syaratnya BENAR maka tampilkan \$bil, sehingga muncul bilangan 3 di browser dan \$bil bertambah satu menjadi 4.

Lagi-lagi di cek syaratnya apakah \$bil <= 3. Nah.. sekarang syarat bernilai SALAH karena 4 <= 3 tidak berlaku. Dengan demikian perulangan berhenti. Sehingga bilangan yang tampil secara keseluruhan adalah

```
1  
2  
3
```

Sekarang.. coba kita bandingkan dengan FOR. Script berikut ini juga akan menampilkan bilangan 1 s/d 3.

```
<?php  
  
for ($bil = 1; $bil <= 3; $bil++)  
{  
    echo $bil. "<br />";  
}  
  
?>
```

Kedua script di atas akan ekuivalen.

Kembali lagi pada script PHP untuk menampilkan bilangan 1 s/d 3 menggunakan WHILE. Sekarang misalkan ditambahkan perintah berikut ini

```
<?php

$bil = 1;
while ($bil <= 3)
{
    echo $bil. "<br />";
    $bil++;
}
echo "Nilai bil sekarang adalah:". $bil;
?>
```

Dengan tanpa menjalankannya di browser, menurut Anda kira-kira perintah yang diberi warna biru tersebut akan menampilkan apa hayoo..?? apakah 3 atau berapa?

Bagi yang menjawab 3... Anda SALAH.

Bagi yang menjawab 4... OK Anda benar.

Lho kok bisa? Padahal pada syarat while tertulis \$bil <= 3. Ya.. Anda benar, tapi coba lihat kembali analisis yang pernah kita singgung di atas. Nilai terakhir \$bil ketika dalam perulangan WHILE adalah 4. Ketika \$bil = 4 inilah yang menyebabkan perulangan WHILE berhenti. Dengan demikian, nilai \$bil pada dasarnya sudah berubah menjadi 4 sehingga setelah WHILE selesai, dan ditampilkan nilai \$bil nya maka akan tampil 4.

Sekarang... mari kita tinjau contoh script WHILE berikutnya. Script berikut ini akan menampilkan bilangan random/acak 0 s/d 10 (menggunakan function rand()). Proses pengacakan ini akan terus dilakukan selama bilangan acak yang terpilih tidak sama dengan 0.

```
<?php

$bil = rand(0, 10);

while ($bil != 0)
{
    echo $bil."<br />";
    $bil = rand(0, 10);
}

?>
```

Nah... dari contoh kasus di atas, tentu kita tahu bahwa jumlah perulangan dalam hal ini tidak diketahui. Perulangan tersebut hanya diketahui kapan perulangan harus berhenti, yaitu ketika bilangan acak yang terpilih sama dengan 0.



Catatan: output yang tampil akan berubah-ubah setiap kali di REFRESH

Trus... kira-kira bisa tidak apabila kita terapkan perulangan WHILE di atas ke dalam bentuk FOR? Sebenarnya dalam PHP bisa dilakukan, namun hal ini tidak berlaku di semua bahasa pemrograman, misalnya dalam PASCAL (mungkin Anda pernah dengar namanya).

Oleh karena itu secara umum saya bisa katakan bahwa tidak semua bentuk WHILE dapat dinyatakan ke bentuk FOR. Apabila Anda menjumpai bentuk perulangan yang tidak diketahui jumlah perulangannya, maka saya sarankan gunakanlah WHILE.

### **WHILE di dalam WHILE (*nested* WHILE)**

Seperti halnya FOR, statement WHILE juga dapat terletak di dalam sebuah WHILE, seperti contoh berikut ini

```
<?php

$a = 1;
while ($a <= 3)
{
    $b = 1;
    while ($b <= 2)
    {
        echo "Nilai a = ".$a. " Nilai b = ".$b. "<br />";
        $b++;
    }
    $a++;
}

?>
```

Script di atas ekuivalen dengan script *nested* FOR berikut ini

```
<?php

for ($a = 1; $a <= 3; $a++)
{
    for ($b = 1; $b <= 2; $b++)
    {
        echo "Nilai a = ".$a. " Nilai b = ".$b. "<br />";
    }
}
```

?>

### Contoh-contoh script menggunakan WHILE:

Contoh-contoh script berikut ini mengadopsi kasus yang ada di modul 9 (tentang FOR)

- A. Script PHP untuk menampilkan bilangan bulat 5 s/d 100 yang berkelipatan 10

```
<?php

$bil = 5;

while ($bil <= 100)
{
    if ($bil % 10 == 0) echo $bil. "<br />";
    $bil++;
}

?>
```

- B. Script PHP untuk menjumlahkan bilangan bulat 2 s/d 50

```
<?php

$jumlah = 0;
$bil = 2;

while ($bil <= 50)
{
    $jumlah = $jumlah + $bil;
    $bil++;
}

echo "Hasilnya adalah ".$jumlah;

?>
```

- C. Script PHP untuk mencari banyaknya bilangan bulat mulai dari 3 s/d 127 yang merupakan kelipatan 6.

```
<?php
$hitung = 0;
$bil = 3;

while ($bil <= 127)
{
    if ($bil % 6 == 0) $hitung = $hitung + 1;
    $bil++;
}

echo "Banyaknya bilangan bulat adalah ".$hitung;
?>
```

- D. Script PHP untuk membuat tabel dengan n baris dan m kolom secara otomatis

```
<?php

$jumlahBaris = 10;
$jumlahKolom = 6;

echo "<table border='1'>";

$baris = 1;

while ($baris <= $jumlahBaris)
{
    echo "<tr>";
    $kolom = 1;
    while ($kolom <= $jumlahKolom)
    {
        echo "<td>...</td>";
        $kolom++;
    }
    echo "</tr>";
    $baris++;
}

echo "</table>";
?>
```

- E. Script PHP untuk menghasilkan list combo box berisi tahun mulai 1930 s/d 2008 secara otomatis

```
<?php

echo "<select name='tahun'>";

$tahun = 1930;

while ($tahun <= 2008)
{
    echo "<option value='" . $tahun . "'>" . $tahun . "</option>";
    $tahun++;
}
echo "</select>";
?>
```

## Tugas Untuk Anda

1. Pandang kembali contoh script B (untuk menjumlahkan bilangan bulat 2 s/d 50) dalam bab ini. Selanjutnya saya ubah sedikit scriptnya menjadi seperti di bawah ini.

```
<?php

$jumlah = 0;
$bil = 2;

while ($bil <= 50)
{
    $bil++;
    $jumlah = $jumlah + $bil;
}

echo "Hasilnya adalah " . $jumlah;

?>
```

Pada script di atas, saya hanya membalik urutan 2 perintah yang saya blok hitam (bold). Dengan tanpa menjalankan scriptnya, coba analisislah script di atas, kira-kira hasilnya sama tidak dengan script pada contoh B tersebut? Jika tidak sama, jelaskan alasannya!

2. Kerjakan kembali soal nomor 1, pada bab 9 menggunakan WHILE
3. Kerjakan kembali soal nomor 2, pada bab 9 menggunakan WHILE
4. Kerjakan kembali soal nomor 3, pada bab 9 menggunakan WHILE
5. Kerjakan kembali soal nomor 4, pada bab 9 menggunakan WHILE
6. Kerjakan kembali soal nomor 5, pada bab 9 menggunakan WHILE
7. Buatlah script PHP menggunakan WHILE untuk menampilkan formasi bintang seperti berikut ini

```
* * * *
* * *
* *
*
```

Formasi bintang di atas untuk N = 4

Terlebih dahulu Anda harus membuat form untuk menerima input sembarang N dan script Anda harus menampilkan formasi bintang N tersebut.

8. Buatlah script PHP menggunakan WHILE untuk menampilkan formasi bintang seperti berikut ini

```
*
* *
* * *
* * * *
* * *
* *
*
```

Formasi bintang di atas untuk N = 4

Terlebih dahulu Anda harus membuat form untuk menerima input sembarang N dan script Anda harus menampilkan formasi bintang N tersebut.

## BAB 11

# Teknik Modularitas

---

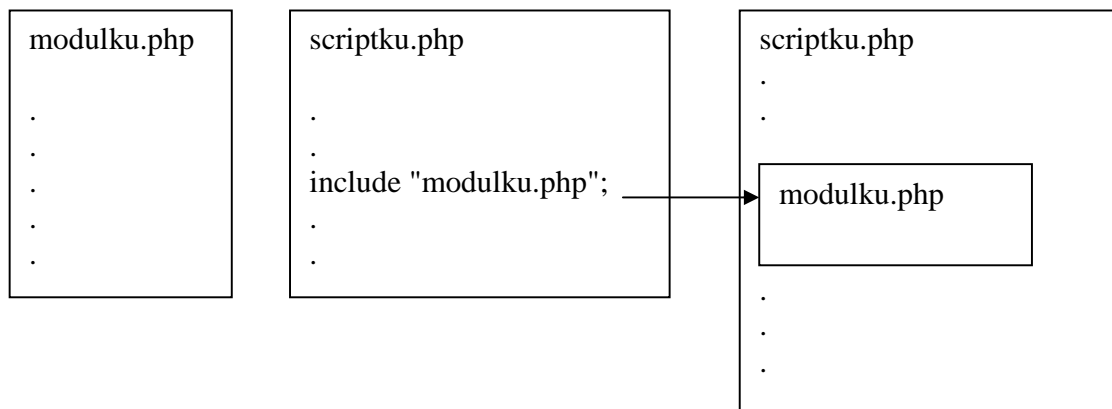
Konsep modularitas dalam programming diperlukan untuk mempermudah dalam pengorganisasian script/program. Adapun prinsip dari konsep ini adalah meletakkan beberapa perintah yang menjalankan suatu tugas khusus ke dalam modul atau file script tersendiri. Setiap kali modul tersebut dibutuhkan, kita hanya menyisipkan modul tersebut ke dalam script yang sedang dibuat dengan cara memanggilnya.

Adapun cara memanggil modul dengan menggunakan perintah

**include "namafilemodul";**

Perhatikan gambaran penggunaan konsep modularitas berikut ini:

Misalkan Anda membuat modul dan disimpan dalam script bernama **modulku.php**. Selanjutnya Anda sedang membuat script tertentu katakanlah diberikan nama file **scriptku.php**. Kebetulan dalam scriptku.php Anda membutuhkan modulku.php, maka skema yang terjadi adalah:



Dari skema di atas, jelas tampak bahwa bila Anda memberikan perintah `include "modulku.php";` dalam `scriptku.php`, maka akan identik dengan bila Anda menyisipkan script yang ada dalam `modulku.php` ke dalam `scriptku.php` secara manual.

Contoh:

Dalam contoh ini kita akan menggunakan konsep modularitas untuk membuat script operasi aritmatika penjumlahan dan pengurangan dari 2 buah bilangan. Untuk input bilangannya disimpan dalam modul tersendiri, misalkan dinamakan `bilangan.php`. Selanjutnya modul ini akan diincludekan ke dalam script penjumlahan dan pengurangan.

#### **bilangan.php**

```
<?php
```

```
$bil1 = 10;  
$bil2 = -5;
```

```
?>
```

#### **jumlah.php**

```
<?php
```

```
include "bilangan.php";
```

```
$hasil = $bil1 + $bil2;  
echo "Hasil penjumlahannya adalah : ".$hasil;
```

```
?>
```

Bentuk script `jumlahkan.php` di atas akan identik dengan script berikut ini

```
<?php
```

```
$bil1 = 10;  
$bil2 = -5;
```

```
$hasil = $bil1 + $bil2;  
echo "Hasil penjumlahannya adalah : ".$hasil;
```

```
?>
```

#### **kurang.php**

```
<?php
```

```
include "bilangan.php";
```

```
$hasil = $bil1 - $bil2;  
echo "Hasil pengurangannya adalah : ".$hasil;  
  
?>
```

Script kurang.php di atas akan identik dengan

```
<?php  
  
$bil1 = 10;  
$bil2 = -5;  
  
$hasil = $bil1 - $bil2;  
echo "Hasil pengurangannya adalah : ".$hasil;  
  
?>
```

## Penerapan Teknik Modularitas

Teknik modularitas dapat dipergunakan dalam pengaturan layout halaman web supaya lebih mudah. Sebelum Anda mengenal teknik modularitas, apa yang Anda lakukan bila diminta membuat halaman web sejumlah 20 buah dengan desain layout yang sama, namun kontennya berbeda? Tentu Anda akan membuat satu buah desain template untuk layout, lalu Anda tinggal mengganti kontennya saja.

OK.. itu mudah, namun bagaimana seandainya dari 20 buah halaman web tersebut Anda ingin ubah desainnya? Tentu Anda harus mengubah semua halaman satu persatu. Bagaimana bila terdapat 100 buah halaman web dan kesemuanya Anda ingin ubah desainnya? Wah repot ya kalau satu persatu.

Nah... di sinilah kita akan gunakan teknik modularitas. Dengan teknik modularitas ini, Anda dapat mengubah desain layout dari banyak halaman web dengan satu langkah saja secara cepat.

Perhatikan contoh berikut ini:

Misalkan kita buat desain layout untuk halaman web kita sebagai berikut:

```
<html>  
  <head>  
    <title>Halaman Webku</title>  
  </head>  
  <body background="yellow">
```



```
<h1><font face="arial">...</font></h1>
<p><font face="verdana"> ... </font></p>
</body>
</html>
```

Nah... misalkan kita akan membuat 3 halaman web menggunakan layout di atas. Maka caranya, kita cari blok dalam HTML layout yang nantinya **akan diisi dengan konten yang berbeda-beda** untuk masing-masing halaman.

```
<html>
<head>
  <title>Halaman Webku</title>
</head>
<body bgcolor="yellow">
  <h1><font face="arial">...</font></h1>
  <p><font face="verdana"> ... </font></p>
</body>
</html>
```

Keterangan:

Bagian yang diblok di atas merupakan bagian tag yang nantinya akan diisi dengan konten yang berbeda-beda untuk setiap halamannya.

Langkah selanjutnya, kita pindahkan tag HTML mulai bagian paling atas dari layout sampai dengan sebelum tag yang dicetak tebal. Tag-tag tersebut kita pindahkan ke dalam modul tersendiri, misalkan dinamakan **header.php**

### header.php

```
<html>
<head>
  <title>Halaman Webku</title>
</head>
<body bgcolor="yellow">
```

Kemudian pindahkan tag setelah bagian yang dicetak tebal dalam layout sampai dengan tag terakhir ke dalam modul tersendiri dan simpan dengan nama **footer.php**.

### footer.php

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

Nah... selanjutnya untuk setiap halaman yang ingin Anda buat, berikan include "header.php" dan include "footer.php";

### **hal1.php**

```
<?php
include "header.php";
?>
```

```
<h1><font face="arial">Ini Halaman 1</font></h1>
<p><font face="verdana">Ini isi halaman 1. Ini isi halaman 1</font></p>
```

```
<?php
include "footer.php";
?>
```

### **hal2.php**

```
<?php
include "header.php";
?>
```

```
<h1><font face="arial">Ini Halaman 2</font></h1>
<p><font face="verdana">Ini isi halaman 2. Ini isi halaman 2</font></p>
```

```
<?php
include "footer.php";
?>
```

### **hal3.php**

```
<?php
include "header.php";
?>
```

```
<h1><font face="arial">Ini Halaman 3</font></h1>
<p><font face="verdana">Ini isi halaman 3. Ini isi halaman 3</font></p>
```

```
<?php
include "footer.php";
?>
```

Nah... dengan demikian, kita akan peroleh 3 halaman web yang pasti memiliki layout yang sama. Terus... bagaimana bila kita ingin mengubah layoutnya? Misal kita ubah

warna background menjadi hijau. Yang Anda lakukan hanyalah mengedit modul header.php karena pengaturan warna background terdapat pada modul tersebut.

#### **header.php**

```
<html>
  <head>
    <title>Halaman Webku</title>
  </head>
  <body bgcolor="green">
```

Secara ajaib ketiga halaman web tadi akan otomatis backgroundnya berubah menjadi hijau semuanya.

Selain cara di atas, Anda juga bisa menggunakan cara yang lain dalam pengaturan layout. Misalkan konfigurasi pengaturan layout seperti jenis font, dan warna background Anda letakkan dalam modul tersendiri.

#### **config.php**

```
<?php
$warnaBackground = "yellow";
$jenisFontHeading = "arial";
$jenisFontParagraf = "verdana";
?>
```

#### **header.php**

```
<?php
include "config.php";
?>
```

```
<html>
  <head>
    <title>Halaman Webku</title>
  </head>
  <body bgcolor="<?php echo $warnaBackground; ?>">
```

#### **footer.php**

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

**hal1.php**

```
<?php
include "header.php";
?>
```

```
<h1><font face="<?php echo $jenisFontHeading; ?>">Ini Halaman 1</font></h1>
<p><font face="<?php echo $jenisFontParagraf; ?>">Ini isi halaman 1. Ini isi halaman
1</font></p>
```

```
<?php
include "footer.php";
?>
```

**hal2.php**

```
<?php
include "header.php";
?>
```

```
<h1><font face="<?php echo $jenisFontHeading; ?>">Ini Halaman 2</font></h1>
<p><font face="<?php echo $jenisFontParagraf; ?>">Ini isi halaman 2. Ini isi halaman
2</font></p>
```

```
<?php
include "footer.php";
?>
```

**hal3.php**

```
<?php
include "header.php";
?>
```

```
<h1><font face="<?php echo $jenisFontHeading; ?>">Ini Halaman 3</font></h1>
<p><font face="<?php echo $jenisFontParagraf; ?>">Ini isi halaman 3. Ini isi halaman
3</font></p>
```

```
<?php
include "footer.php";
?>
```

Nah... dengan demikian, bila Anda ingin mengubah layout ketiga halaman web maka Anda cukup mengubah modul **config.php**

Sekarang kita lanjutkan, bagaimana bila kita mengintegrasikan konsep modularitas yang terkait dengan pengaturan layout di atas dengan form processing?

OK... untuk contoh sederhana, saya ambil contoh script menjumlahkan dua buah bilangan dengan inputnya melalui form.

Misalkan kita ambil desain layout halaman webnya seperti contoh sebelumnya, yaitu terdiri dari modul header.php dan footer.php nya sebagai berikut:

#### **config.php**

```
<?php
$warnaBackground = "yellow";
$jenisFontHeading = "arial";
$jenisFontParagraf = "verdana";
?>
```

#### **header.php**

```
<?php
include "config.php";
?>

<html>
  <head>
    <title>Penjumlahan Bilangan</title>
  </head>
  <body bgcolor="<?php echo $warnaBackground; ?>">
```

#### **footer.php**

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

Selanjutnya kita buat form yang terletak dalam halaman yang menggunakan layout tersebut

**form.php**

```
<?php
include "header.php";
?>

<h1><font face="<?php echo $jenisFontHeading; ?>">Input Bilangan</font></h1>
<p><font face="<?php echo $jenisFontParagraf; ?>">Masukkan dua buah bilangan pada
form berikut ini</font></p>

<form method="post" action="proses.php">
Bilangan 1 <input type="text" name="bil1" /><br />
Bilangan 2 <input type="text" name="bil2" /><br />
<input type="submit" name="submit" value="Jumlahkan" />
</form>

<?php
include "footer.php";
?>
```

**proses.php**

```
<?php
include "header.php";

$bil1 = $_POST['bil1'];
$bil2 = $_POST['bil2'];
$hasil = $bil1 + $bil2;
?>

<h1><font face="<?php echo $jenisFontHeading; ?>">Output</font></h1>
<p><font face="<?php echo $jenisFontParagraf; ?>">Hasil penjumlahannya adalah
<?php echo $hasil; ?> </font></p>

<?php
include "footer.php";
?>
```

## Tugas Untuk Anda

1. Coba buatlah desain layout sebuah halaman web (tanpa menggunakan tool apapun selain Notepad atau Notepad++) dengan tampilan berikut ini:

Nama Situs Anda	
Link Link Link	Heading(Judul Konten)  Konten/isi
copyright	

2. Dengan menggunakan teknik modularitas, coba buat modul config.php, header.php dan footer.php dari hasil soal nomor 1. Modul config.php digunakan untuk mengatur konfigurasi seperti jenis font, ukuran font, warna background dll.

Keterangan:

Untuk bagian “Nama Situs Anda”, “Link” dan “Copyright” bersifat fixed, artinya kelima halaman tersebut memiliki tampilan yang sama. Sehingga dalam hal ini bagian yang berubah-ubah untuk setiap halaman hanyalah bagian “Heading dan Konten” saja.

3. Dari hasil soal nomor 2, buat 5 halaman web berisi konten yang berbeda. Kelima halaman web tersebut harus terintegrasi dengan link yang ada pada bagian link (lihat desain layout dari soal no. 1)

Keterangan : hasil pekerjaan no.1 s/d 3 dijadikan dalam satu script yang sama

4. Kerjakan kembali tugas No. 2 bab 9 dengan mengintegrasikannya bersama layout yang Anda buat pada soal no. 1 di atas.

Keterangan: hasil pekerjaan no. 4 dijadikan dalam script tersendiri

5. Kerjakan kembali tugas No. 8 bab 10 dengan mengintegrasikannya bersama layout yang Anda buat pada soal no. 1 di atas.

Keterangan: hasil pekerjaan no. 5 dijadikan dalam script tersendiri

6. Coba cari dan download lah sebuah layout instant dari internet, disarankan berasal dari <http://freecsstemplates.org>, buatlah modul header.php dan footer.php. Lalu buatlah 5 halaman web menggunakan layout tersebut dengan teknik modularitas.

Keterangan:

Biasanya desain layout template terdiri dari 4 bagian, yaitu: bagian *header* (bagian yang terletak di atas konten), *footer* (bagian yang terletak di bawah konten), *sidebar* (bagian yang terletak di samping kiri atau kanan konten), dan bagian konten itu sendiri. Nah.. dari keempat bagian, hanya bagian konten saja yang berubah-ubah, sedangkan bagian yang lain adalah fixed.



## BAB 12

# Tentang ARRAY

---

Selama ini bentuk variabel yang kita buat untuk menyimpan suatu value atau nilai adalah berbentuk tunggal. Maksudnya, satu variabel hanya bisa untuk menyimpan sebuah value saja. Misalkan sebuah value telah disimpan atau diassign ke dalam sebuah variabel, selanjutnya variabel tersebut akan diassign kembali dengan sebuah value yang lain, maka value yang sebelumnya akan ditimpa oleh value yang baru.

Nah... kira-kira ada tidak ya, variabel yang dapat diassign oleh beberapa value yang lebih dari satu? Oya.. ada, namanya adalah variabel bertipe data array.

Misalkan suatu variabel bernama X bertipe data array, maka X ini dapat Anda bayangkan seperti gambar berikut

X

0	1	2	3	4	...	N-1

Wah... kok 'X' nya seperti memiliki beberapa ruang/kamar/bilik atau apalah terserah namanya ☺ yang jelas jumlah ruangnya adalah berhingga yaitu sejumlah N. Tiap-tiap kamar terdapat semacam nomor ruangan: 0, 1, 2, 3, ..., N-1. Nah.. di dalam ruangan-ruangan inilah kita bisa assign beberapa value sekaligus.

**Ingat ya... untuk PHP, nomor ruang array dimulai dari 0, bukan 1.**

Dalam PHP, untuk mendeklarasikan suatu variabel (misalkan \$x) bertipe data array adalah sebagai berikut:

```
$x = array();
```

Selanjutnya bagaimana perintah untuk mengassign suatu value ke dalam ruangan array?

Oya... sebelumnya, saya perlu sampaikan bahwa untuk PHP, variabel bertipe data array ini bisa diassign dengan berbagai tipe data. Maksudnya, untuk setiap ruang yang ada, boleh diassign dengan nilai berbagai tipe data yang berlainan, misalnya untuk ruang

nomor 1 akan diisi dengan nilai berupa bilangan bulat (integer), ruang no. 2 diisi dengan real, no. 3 diisi dengan string dsb. Itu bisa dilakukan.

Nah.. berikut ini contoh mengassign ruang variabel yang bertipe data array. Sebagai contoh misalkan kita akan mengassign ruang nomor 3 dengan value bilangan integer 10, dan ruang no. 5 dengan string 'I love PHP' perintahnya:

```
<?php
$х = array();
$х[3] = 10;
$х[5] = "I love PHP";

?>
```

Kita juga bisa mengisi value untuk setiap ruang array sekaligus secara berturut-turut seperti contoh berikut ini:

```
<?php
$х = array(10, 12, 3, 44, 50, "hallo");

?>
```

atau cara kedua bisa pula menggunakan seperti berikut ini:

```
<?php
$х = array(0 => 10, 1 => 12, 2 => 3, 3 => 44, 4 => 50, 5 => "hallo");

?>
```

Selanjutnya, bagaimana cara kita menampilkan value-value yang tersimpan dalam kamar array tersebut? Ya.. caranya mudah, tinggal panggil saja nomor kamarnya lantas di echo kan.

```
<?php
$х = array(10, 12, 3, 44, 50, "hallo");
echo $х[0]; // akan menampilkan value pada ruang ke-0, yaitu 10
echo $х[5]; // akan menampilkan value pada ruang ke-5, yaitu 'hallo'

$х[2] = -3; // mengganti value pada ruang ke-2 dengan value yang baru (-3)
echo $х[2]; // akan menghasilkan -3 (yang tampil bukan 3 karena sudah ditimpa -3)

?>
```

Kita juga bisa menampilkan semua value array sekaligus secara berturut-turut menggunakan looping

```
<?php

$x = array(10, 12, 3, 44, 50, "hallo");

for ($i = 0; $i <= 5; $i++)
{
    echo "Nilai kamar ke ".$i. " dari array = ".$x[$i]. "<br />";
}

?>
```

atau cara kedua, kita bisa menggunakan perintah `count()` untuk menentukan jumlah ruang array yang telah diisi value.

```
<?php

$x = array(10, 12, 3, 44, 50, "hallo");

for ($i = 0; $i <= count($x)-1; $i++)
{
    echo "Nilai kamar ke ".$i. " dari array = ".$x[$i]. "<br />";
}

?>
```

Dalam script di atas, perintah `count($x)` akan menghasilkan 6 karena dalam array `$x` tersebut terdapat 6 ruang yang terisi value. Penggunaan `count()` ini akan efektif bila kita malas menghitung jumlah ruang array yang telah diisi value secara manual, ibaratnya kita minta tolong si PHP untuk menghitungkannya. Bayangkan seandainya terdapat 100 lebih ruang dalam suatu array, masa' kita akan menghitungnya satu-persatu secara manual? Capee deh... ☺

## Array Asosiatif

Dari contoh-contoh di atas, kita lihat bahwa ruang atau kamar dari array diberikan nomor ruang sebagai penanda, yaitu mulai dari 0, 1, 2, dst... Nah... yang menjadi pertanyaan, apakah penanda setiap ruang ini harus berupa angka? Jawabnya tidak, Anda juga bisa memberi penanda setiap ruang dengan suatu kata atau karakter. Array yang demikian ini disebut array asosiatif.

Berikut ini contoh membuat array asosiatif dan cara menampilkan value pada suatu ruangnya

```
<?php

$x = array("kamar1" => 10, "kamar2" => 12, "kamar3" => 3, "kamar4" => "hallo");
echo $x['kamar2']; // akan menampilkan 12
echo $x['kamar4']; // akan menampilkan "hallo"

?>
```

Sekarang bagaimana cara menampilkan semua value yang tersimpan dalam array asosiatif tersebut sekaligus? Sama caranya dengan sebelumnya, namun kita khusus akan menggunakan looping foreach. Looping ini khusus untuk array asosiatif.

```
<?php

$x = array("kamar1" => 10, "kamar2" => 12, "kamar3" => 3, "kamar4" => "hallo");

foreach($x as $ruang => $value)
{
    echo "Isi dari ruang : ".$ruang. " adalah : ".$value. "<br />";
}

?>
```

## Menambah Elemen Array

Misalkan kita sebelumnya telah memiliki array dengan value-value tertentu dalam setiap ruangnya, perhatikan script berikut ini

```
<?php

$x = array(10, 12, 3, 44, 50, "hallo");

?>
```

Nah... bagaimana seandainya jika kita ingin menambahkan sebuah ruang lagi untuk diisi dengan suatu value? Dalam hal ini pada array di atas nomor ruangnya terdiri dari 0, 1, 2, ..., 5. Atau dengan kata lain kita akan menambah sebuah ruang lain yaitu ruang nomor 6 untuk diisi dengan suatu value, misalkan 100. OK.. caranya adalah sebagai berikut

Cara pertama, Anda bisa langsung memberikan perintah

```
$x[6] = 100;
```

atau

```
<?php
```

```
$x = array(10, 12, 3, 44, 50, "hallo");
```

```
$x[6] = 100;
```

```
?>
```

Atau dengan cara kedua menggunakan perintah

```
$x[] = 6;
```

Atau

```
<?php
```

```
$x = array(10, 12, 3, 44, 50, "hallo");
```

```
$x[] = 100;
```

```
?>
```

Sekarang kita lihat apakah ruangnya benar-benar bertambah.

```
<?php
```

```
$x = array(10, 12, 3, 44, 50, "hallo");
```

```
$x[] = 100;
```

```
for ($i = 0; $i <= count($x)-1; $i++)
```

```
{
```

```
    echo "Nilai kamar ke ".$i. " dari array = ".$x[$i]. "<br />";
```

```
}
```

```
?>
```

Yes... benar-benar bertambah ruang arraynya ☺ Anda dapat melihat hasilnya ketika dijalankan di browser.

## Contoh Kasus

Sekarang kita akan mencoba membahas beberapa contoh kasus yang menggunakan array.

### Kasus 1. Menghitung Rata-rata beberapa bilangan

Misalkan diberikan bilangan-bilangan berikut ini 9, 2, 10, 3, 7, 5, 2. Kemudian kita diminta membuat script untuk mencari rata-rata dari semua bilangan tersebut. Nah.. dalam hal ini, kita bisa menggunakan array untuk menghitungnya. Anda bisa saja sih menggunakan script berikut ini

```
<?php

$rataRata = (9 + 2 + 10 + 3 + 7 + 5 + 2)/7;
echo $rataRata;

?>
```

Namun... bagaimana bila jumlah data bilangannya sampai dengan 100 atau bahkan ribuan? He... 3x Anda bisa bayangkan sendiri alangkah repotnya dalam menulis formulanya. Nah.. untuk memudahkan perhitungan, kita akan menggunakan pendekatan array.

Langkah pertama: kita buat suatu array yang ruangnya diisi oleh bilangan-bilangan tersebut.

```
<?php

$bil = array(9, 2, 10, 3, 7, 5, 2);

?>
```

Selanjutnya.. langkah kedua: kita hitung dulu hasil penjumlahan setiap value dalam array tersebut. Untuk menghitungnya, kita manfaatkan looping. Konsepnya adalah seperti ini: mula-mula kita asumsikan hasil jumlahnya adalah 0 (karena kita belum mulai menjumlahkan), kemudian kita baca array ke-0, yaitu 9. Value ini kita jumlahkan dengan hasil penjumlahan sebelumnya yaitu 0 atau  $0 + 9 = 9$ . Dengan demikian hasil penjumlahan sekarang adalah 9. Selanjutnya kita baca array ke-1, yaitu 2. Value ini kembali kita jumlahkan dengan hasil penjumlahan sebelumnya yaitu 9 atau  $9 + 2 = 11$ . So.. hasil penjumlahan sekarang menjadi 11. Proses ini terus diulang sampai pada value terakhir dari array. Setelah selesai, maka akan kita peroleh hasil dari penjumlahan total semua value.

```
<?php

$bil = array(9, 2, 10, 3, 7, 5, 2);

$jumlah = 0;
for ($i = 0; $i <= count($bil)-1; $i++)
{
    $jumlah = $jumlah + $bil[$i]; // atau bisa dituliskan $jumlah += $bil[$i];
}
?>
```

Setelah kita peroleh hasil penjumlahan setiap value dari array, langkah ketiga kita bisa hitung deh rata-rata bilangannya.

```
<?php

$bil = array(9, 2, 10, 3, 7, 5, 2);

$jumlah = 0;
for ($i = 0; $i <= count($bil)-1; $i++)
{
    $jumlah = $jumlah + $bil[$i]; // atau bisa dituliskan $jumlah += $bil[$i];
}

$rataRata = $jumlah/count($bil);
echo "Rata-ratanya adalah : ".$rataRata;

?>
```

## Kasus 2. Mencari Nilai Terbesar dari Kumpulan Bilangan

Misalkan diberikan bilangan-bilangan berikut ini 9, 2, 10, 3, 7, 5, 2. Kita diminta membuat script untuk menampilkan bilangan terbesar dari bilangan-bilangan tersebut. Tentu kita tahu, dalam hal ini bilangan terbesarnya adalah 10. Nah... yang jadi masalah, bagaimana cara mendapatkan 10 tersebut sebagai bilangan terbesar bila dinyatakan dalam program.

Secara sekilas, kasusnya tampak sederhana namun ketika dinyatakan dalam program bisa jadi membuat pusing orang ☺

Untuk mencari nilai terbesar, kita gunakan konsep perbandingan.

Anda yang sudah berumur 17 tahun ke atas, mungkin tidak merasakan logika atau konsep ini karena Anda sudah banyak makan garam dalam masalah perhitungan sederhana seperti ini. Secara tidak sadar, ketika Anda diminta mencari bilangan terbesar, sebenarnya dalam pikiran Anda terjadi proses perbandingan tersebut beberapa kali. Namun, karena proses yang terjadi begitu cepat (seiring bertambahnya pengalaman dalam berpikir), maka proses ini seakan tidak terasa. Mungkin hal inilah yang kadang membuat para pembelajar program sering sulit mengekspresikan proses yang terjadi dalam pikirannya ke dalam bentuk program.

Tapi... sebenarnya tidak demikian, membuat program pada dasarnya mirip ketika Anda sedang mengajarkan konsep atau ide penyelesaian suatu kasus pada anak kecil yang sebelumnya belum tahu apa-apa. Kita harus bisa menyatakan setiap proses langkah demi langkah secara urut.

Nah... stop ber-intermezzo... Now.. back to the case.. ☺

Trus... idenya bagaimana untuk mencari bilangan terbesar?

OK.. idenya adalah, kita harus membuat asumsi terlebih dahulu. Kita asumsikan bilangan terbesar mula-mula adalah bilangan pertamanya. Selanjutnya bilangan terbesar asumsi ini kita bandingkan dengan setiap bilangan yang ada secara urut satu persatu. Jika terdapat bilangan yang lebih besar dari bilangan terbesar asumsi, maka bilangan tersebut akan dianggap sebagai bilangan terbesar asumsi berikutnya. Dengan demikian setelah proses perbandingan selesai semuanya, maka akan kita peroleh bilangan terbesar dari semua bilangan.

Contoh: dari kasus di atas, pertama kita asumsikan bilangan terbesarnya adalah bilangan pertama yaitu 9. Selanjutnya bilangan terbesar ini kita bandingkan dengan bilangan ke-1, yaitu 9. Karena 9 tidak lebih besar dari 9, maka bilangan terbesarnya tetap 9. Selanjutnya dibandingkan dengan bilangan ke-2, yaitu 2. Dalam hal ini 2 tidak lebih besar dari 9, maka bilangan terbesarnya tetap 9. Trus.. kita bandingkan dengan bilangan ke-3 yaitu 10. Nah... 10 lebih besar dari 9, maka bilangan terbesarnya sekarang menjadi 10. Proses ini dilanjutkan sampai dengan bilangan terakhir.

```
<?php
```

```
$bil = array(9, 2, 10, 3, 7, 5, 2);
```

```
$max = $bil[0];
```

```
for ($i = 0; $i <= count($bil)-1; $i++)
```

```
{
```

```
    if ($bil[$i] > $max) $max = $bil[$i];
```

```
}
```



```
echo "Bilangan terbesarnya adalah : ".$max;
```

```
?>
```

**Kasus 3. Identik dengan kasus 1, namun bilangan-bilangan yang akan dicari rata-ratanya diinputkan melalui form.**

Adapun skenario untuk input bilangannya adalah: pertama kali akan muncul form untuk memasukkan jumlah bilangan yang akan dimasukkan. Setelah dimasukkan jumlah bilangannya, kemudian disubmit berikutnya akan muncul form baru yang berisi komponen textbox untuk memasukkan masing-masing bilangan. Jumlah komponen textbox ini sejumlah bilangan yang sebelumnya diinputkan. Misalkan pada form pertama tadi kita masukkan jumlah bilangannya 10, maka setelah disubmit akan muncul form baru berisi 10 buah textbox untuk memasukkan ke-10 bilangan. Setelah dimasukkan masing-masing bilangan, selanjutnya disubmit dan akhirnya akan muncul jumlah rata-ratanya.

Untuk itu pertama kita buat form untuk memasukkan jumlah bilangannya terlebih dahulu.

**form1.htm**

```
<h1>Cari Rata-rata Bilangan</h1>

<form method="post" action="form2.php">
Masukkan Banyaknya Bilangan <input type="text" name="n" />
<input type="submit" name="submit" value="Submit" />
</form>
```

Selanjutnya kita buat form kedua untuk menampilkan form dinamis guna memasukkan masing-masing bilangan. Karena sesuai skenario bahwa jumlah textboxnya sejumlah banyaknya bilangan yang dimasukkan pada form pertama, maka untuk mudahnya kita buat melalui perulangan FOR dengan jumlah perulangannya n kali.

**form2.php**

```
<h1>Cari Rata-rata Bilangan</h1>

<?php
$n = $_POST['n'];

echo "<form method='post' action='proses.php'>";
echo "<table>";
for ($i = 0; $i <= $n-1; $i++)
```

```
{
    echo "<tr><td>Bil ke-".($i+1)."</td>
        <td><input type='text' name='data".$i.'" /></td></tr>";
}
echo "<tr><td></td>
    <td><input type='submit' name='submit' value='Submit' /></td></tr>";
echo "</table>";
echo "</form>";

?>
```

Hasil script di atas akan menghasilkan tag HTML seperti di bawah ini (jika n = 4)

```
<h1>Cari Rata-rata Bilangan</h1>

<form method='post' action='proses.php'>
<table>
<tr>
    <td>Bil ke-1</td>
    <td><input type='text' name='data0' /></td>
</tr>
<tr>
    <td>Bil ke-2</td>
    <td><input type='text' name='data1' /></td>
</tr>
<tr>
    <td>Bil ke-3</td>
    <td><input type='text' name='data2' /></td>
</tr>
<tr>
    <td>Bil ke-4</td>
    <td><input type='text' name='data3' /></td>
</tr>
<tr>
    <td></td>
    <td><input type='submit' name='submit' value='Submit' /></td>
</tr>
</table>
</form>
```

Perhatikan tag HTML di atas! Untuk setiap komponen textbox yang dihasilkan tampak pada bagian atribut name="" secara otomatis akan membentuk nama: 'data0', 'data1', dst. Mengapa dimulai dari 'data0'? kok tidak mulai dari 'data1'? ya... ini masalah kesukaan saja. Kalau saya biasa mulai dari ke-0. Kalau Anda mulai dari 'data1' ya gpp 😊

Selanjutnya kita buat script untuk memprosesnya. Nah... pada bagian ini, mungkin yang sulit adalah bagaimana cara membaca value yang diisikan pada setiap komponen textbox mengingat komponen ini dihasilkan secara dinamis (dengan jumlah textbox yang berubah-ubah).

OK... trik untuk membaca value yang berasal dari komponen yang dinamis seperti ini adalah: **kita memanfaatkan perulangan saja**

```
for ($i=0; $i <= jumlahkomponen - 1; $i++)
{
    $bilangan[$i] = $_POST['data'.$i];
}
```

Perhatikan! Mengapa pada bagian \$\_POST berbentuk \$\_POST['data'.\$i] ?

Ya... karena nama komponen dari form kan bentuknya name='data0', name='data1', name='data2', dst. Maka dalam membaca valuenya harusnya juga menggunakan \$\_POST['data0'], \$\_POST['data1'], dst.

Nah... supaya mudah, maka kita gunakan \$\_POST['data'.\$i] dengan \$i nya diperoleh dari FOR. Simpel bukan? Dengan alasan inilah mengapa saya suka memberi nama komponen dalam formnya mulai dari 0 terlebih dahulu, seperti name='data0', name='data1', dst...

Kemudian... muncul masalah lagi yang lain. Untuk FOR nya harusnya \$i bergerak dari 0, 1, 2, sampai dengan 'jumlah komponen' dikurangi 1. Nah.. yang menjadi masalah, bagaimana mendapatkan 'jumlah komponen' pada script ini? Padahal jumlah komponen ini berada pada form2.php. Atau dengan kata lain, bagaimana caranya jumlah komponen yang berada pada form2.php ini juga bisa dikenali di script untuk memproses inputnya ini? Ya... solusinya mudah, kita tinggal buat saja komponen 'hidden textbox' yang berisi jumlah komponennya.

### form2.php

```
<h1>Cari Rata-rata Bilangan</h1>
```

```
<?php
$n = $_POST['n'];
```

```
echo "<form method='post' action='proses.php'>";
echo "<table>";
for ($i = 0; $i <= $n-1; $i++)
{
    echo "<tr><td>Bil ke-".($i+1)."</td>
        <td><input type='text' name='data'.".$i." /></td></tr>";
```

```
}
echo "<tr><td></td>
      <td><input type='submit' name='submit' value='Submit' /></td></tr>";

echo "<tr><td></td>
      <td><input type='hidden' name='n' value='\".$n.\"' /></td></tr>";

echo "</table>";
echo "</form>";

?>
```

Ingat... komponen hidden text ini harus terletak dalam <form></form>. Sesuai namanya, maka komponen ini tersembunyi (tidak tampak dalam browser). Namun begitu user mengklik submit, maka value ini ikut tersubmit untuk diproses pada script pemrosesnya.

OK... masalah teratasi, dan sekarang kita bisa buat scriptnya

### proses.php

```
<h1>Cari Rata-rata Bilangan</h1>

<?php

// membaca jumlah komponen dari hidden textbox

$n = $_POST['n'];

// membaca value dari masing-masing komponen textbox

for ($i=0; $i <= $n - 1; $i++)
{
    $bilangan[$i] = $_POST['data'.$i];
}

// menjumlahkan nilai semua bilangan

$jumlah = 0;

for ($i=0; $i <= $n - 1; $i++)
{
    $jumlah += $bilangan[$i];
}
```

```
// hitung rata-rata

$rataRata = $jumlah/$n;

echo "Rata-ratanya adalah : ".$rataRata;

?>
```

Atau cara lain Anda bisa menggabung proses menjumlahkan nilai semua bilangan bersama proses membaca bilangan dari form menjadi satu bagian

### **proses.php**

```
<h1>Cari Rata-rata Bilangan</h1>

<?php

// membaca jumlah komponen dari hidden textbox

$n = $_POST['n'];

// membaca value dari masing-masing komponen textbox dan menjumlahkannya

$jumlah = 0;

for ($i=0; $i <= $n - 1; $i++)
{
    $bilangan[$i] = $_POST['data'][$i];
    $jumlah += $bilangan[$i];
}

// hitung rata-rata

$rataRata = $jumlah/$n;

echo "Rata-ratanya adalah : ".$rataRata;

?>
```

## Tugas Untuk Anda

1. Perhatikan kembali kasus no. 2 pada bab ini. Apakah asumsi bilangan terbesar untuk pertama kali harus diberikan pada bilangan yang pertama? Jelaskan jawaban Anda
2. Identik dengan kasus no. 2 pada bab ini, sekarang coba Anda buat script PHP untuk mencari nilai terkecil dari bilangan-bilangan 283, 182, 381, 119, 391, 591, 123, 124, 284, 215, 312.

Keterangan: tanpa menggunakan form input

3. Identik dengan kasus no. 3 pada bab ini, coba Anda buat form input untuk memasukkan n buah bilangan, yang selanjutnya dicari nilai terbesar dan nilai terkecil, serta nilai jangkauan yang diperoleh dari nilai terbesar dikurangi nilai terkecil.
4. Diketahui sebuah data bilangan berikut ini:

273  
281  
384  
119  
392  
184  
105  
129  
204  
219  
274  
275  
263

Buatlah script untuk fasilitas pencarian data bilangan. Mula-mula buatlah form untuk memasukkan bilangan yang ingin dicari. Selanjutnya apabila ditekan submit, maka berikutnya muncul status 'ada atau tidaknya' bilangan yang dicari dalam keseluruhan data. Jika ada, maka tampilkan urutan data ke berapa dari keseluruhan data (urutan dimulai dari 1, 2, 3, ...)

Contoh:

**Masukkan Bilangan yang dicari = 281** [submit]

Selanjutnya muncul keterangan:

**“Bilangan yang Anda cari ada dalam data pada urutan ke-2”**

5. Sebuah toko swalayan ingin membuat aplikasi sederhana untuk menghitung total harga yang harus dibayarkan si pembeli. Misalkan dalam toko tersebut hanya terdapat 10 jenis barang saja. Berikut daftar barang yang ada dalam toko beserta harganya

Kode Barang	Nama Barang	Harga Satuan
1001	Sabun Lifebuoy	Rp. 1.500,-
1002	Permen Blaster	Rp. 5.600,-
1003	Pasta Gigi Pepsodent	Rp. 4.560,-
1004	Madu Arbain	Rp. 30.000,-
1005	Kecap ABC	Rp. 7.250,-
1006	Saus Tomat ABC	Rp. 6.700,-
1007	Gula Gulaku	Rp. 8.900,-
1008	Rinso	Rp. 7.100,-
1009	Super Pel	Rp. 6.450,-
1010	Permen Tango	Rp. 5.600,-

Aplikasi ini terletak pada kasir. Berikut ini tampilan aplikasinya

#### Form pertama

Masukkan banyaknya jenis barang yang dibeli [BANYAK JENIS BARANG]

[ SUBMIT ]

#### Form kedua

Masukkan kode barang yang dibeli dan banyaknya

Barang ke -1 [MASUKKAN KODE BARANG] [JUMLAH BRG]

Barang ke -2 [MASUKKAN KODE BARANG] [JUMLAH BRG]

Barang ke -3 [MASUKKAN KODE BARANG] [JUMLAH BRG]

.

.

[ SUBMIT ]

#### Tampilan Output

Daftar Barang Yang Dibeli

KODE BARANG	NAMA BARANG	JUM BRG	HARGA
XXX	XXX	XXX	Rp. XXX
XXX	XXX	XXX	Rp. XXX
XXX	XXX	XXX	Rp. XXX
.			
.			
TOTAL HARGA : Rp. XXX			

Contoh:

Misalkan ada konsumen yang membeli sabun Lifebuoy (10 buah), dan permen Blaster (20 pack), maka formnya:

#### Form Pertama:

Masukkan banyaknya jenis barang yang dibeli [ 2 ]

[SUBMIT]

#### Form Kedua

Masukkan kode barang yang dibeli dan banyaknya

Barang ke -1 [ 1001 ] [ 10 ]  
Barang ke -2 [ 1002 ] [ 20 ]

[ SUBMIT ]

#### Tampilan Output

Daftar Barang Yang Dibeli

KODE BARANG	NAMA BARANG	JUM BRG	HARGA
1001	Sabun Lifebuoy	10	Rp. 15000,-
1002	Permen Blaster	20	Rp. 112000,-
TOTAL HARGA : Rp. 127000,-			



## BAB 13

# Tentang SESSION

---

Untuk menjelaskan tentang SESSION dan kegunaannya, saya akan mulai dengan memberikan contoh visualisasi terlebih dahulu.

Misalkan kita memiliki sebuah form sebagai berikut

```
<form method="post" action="submit.php">
Bilangan ke-1 <input type="text" name="bil1" /><br />
Bilangan ke-2 <input type="text" name="bil2" /><br />
<input type="submit" name="submit" value="Submit" />
</form>
```

Form di atas digunakan untuk memasukkan dua buah bilangan. Selanjutnya kita buat script untuk membaca kedua bilangan yang diinputkan tersebut

### **submit.php**

```
<?php

$bil1 = $_POST['bil1'];
$bil2 = $_POST['bil2'];

echo "Anda memasukkan bilangan ke-1 : ".$bil1."<br />";
echo "Anda memasukkan bilangan ke-2 : ".$bil2."<br />";

echo "<a href='lanjut.php'>Klik di sini</a>";
?>
```

Apabila kita telah memasukkan dua buah bilangan melalui form, kemudian mensubmitnya maka pada script submit.php akan memunculkan kedua bilangan tersebut dengan sukses.

Pada submit.php ini terdapat pula link menuju script lanjut.php. Dalam script lanjut.php ini, kita juga akan menampilkan kedua bilangan yang diinputkan sebelumnya.

**lanjut.php**

```
<?php  
  
echo "Anda memasukkan bilangan ke-1 : ".$bil1."<br />";  
echo "Anda memasukkan bilangan ke-2 : ".$bil2."<br />";  
  
echo "<a href='lanjut.php'>Klik di sini</a>";  
?>
```

Nah... apa yang terjadi bila kita mengklik link menuju ke lanjut.php tersebut? Apakah kedua bilangan yang diinputkan tadi masih bisa terbaca nilainya? Jawabannya adalah TIDAK. Kedua bilangan hanya bisa terbaca pada script submit.php saja, karena submit.php adalah halaman pertama setelah pemrosesan form.

Mungkin Anda berpikiran bahwa perlu ditambahkan perintah

```
$bil1 = $_POST['bil1'];  
$bil2 = $_POST['bil2'];
```

pada lanjut.php. Cara ini pun juga tidak bisa untuk membaca kedua bilangan.

Trus.. bagaimana supaya dalam lanjut.php ini kedua bilangan yang diinputkan masih bisa dibaca nilainya?

Untuk mengatasi hal ini terdapat 2 solusi yang bisa digunakan, yaitu menggunakan GET method atau SESSION.

Apabila kita menggunakan GET method, maka caranya adalah menyisipkan parameter untuk kedua bilangan pada link menuju lanjut.php tersebut. Selanjutnya dalam lanjut.php, parameter kedua bilangan ini dibaca nilainya.

**submit.php**

```
<?php  
  
$bil1 = $_POST['bil1'];  
$bil2 = $_POST['bil2'];  
  
echo "Anda memasukkan bilangan ke-1 : ".$bil1."<br />";  
echo "Anda memasukkan bilangan ke-2 : ".$bil2."<br />";  
  
echo "<a href='lanjut.php?bil1= ".$bil1."&bil2=".$bil2."'>Klik di sini</a>";  
?>
```

**lanjut.php**

```
<?php

$bil1 = $_GET['bil1'];
$bil2 = $_GET['bil2'];

echo "Anda memasukkan bilangan ke-1 : ".$bil1."<br />";
echo "Anda memasukkan bilangan ke-2 : ".$bil2."<br />";

?>
```

Cara di atas akan dapat memunculkan kedua bilangan pada lanjut.php. Namun... cara ini terdapat kelemahan yaitu ditampilkannya nilai input sebagai parameter dalam URL link.

Trus... bagaimana dengan menggunakan SESSION? Ya... dengan menggunakan SESSION, kita tidak perlu menuliskan nilai input dalam URL link. Nilai input ini nantinya akan disimpan dalam suatu variabel SESSION. Selama browser kita masih aktif (browser belum ditutup) atau sengaja dinonaktifkan (dihapus) SESSION nya, maka variabel SESSION ini masih terus aktif.

Nilai yang disimpan dalam variabel SESSION ini bisa dibaca atau dikenali di beberapa halaman script. Beda dengan variabel biasa. Kalau variabel biasa itu hanya dikenali pada halaman script tersebut saja.

Selanjutnya bagaimana cara menyimpan nilai ke dalam suatu variabel SESSION? Ya... sebelum mulai menyimpan, kita harus mengaktifkan SESSION nya terlebih dahulu menggunakan perintah

```
session_start();
```

Setelah SESSION aktif, barulah kita bisa membuat variabel SESSION untuk menyimpan nilai. Berikut ini perintah untuk menyimpan sebuah nilai ke dalam variabel SESSION

```
$_SESSION['namaSession'] = value;
```

Nilai yang bisa disimpan dalam SESSION ini bisa berupa bilangan, array, karakter maupun string.

Sehingga apabila konsep ini diterapkan pada contoh di atas, maka berikut ini adalah script submit.php yang telah dimodifikasi

**submit.php**

```
<?php

session_start(); // mengaktifkan SESSION

$bil1 = $_POST['bil1'];
$bil2 = $_POST['bil2'];

// menyimpan kedua bilangan ke dalam SESSION

$_SESSION['bil1'] = $bil1;
$_SESSION['bil2'] = $bil2;

echo "Anda memasukkan bilangan ke-1 : ".$bil1."<br />";
echo "Anda memasukkan bilangan ke-2 : ".$bil2."<br />";

echo "<a href='lanjut.php'>Klik di sini</a>";
?>
```

Kemudian, bagaimana dengan script lanjut.php nya? Bagaimana cara menampilkan nilai kedua bilangan yang disimpan dalam SESSION? Ini dia scriptnya.

**lanjut.php**

```
<?php

session_start();

// membaca nilai dari variabel SESSION

$bil1 = $_SESSION['bil1'];
$bil2 = $_SESSION['bil2'];

echo "Anda memasukkan bilangan ke-1 : ".$bil1."<br />";
echo "Anda memasukkan bilangan ke-2 : ".$bil2."<br />";

?>
```

**Keterangan:**

Dalam script lanjut.php di atas, kita tetap memberikan perintah session\_start() karena dalam script lanjut.php ini kita masih memanfaatkan variabel SESSION.

## Menghapus Variabel SESSION

Kemudian, bagaimana cara menghapus variabel SESSION? Cara pertama, Anda akan langsung bisa menghapus SESSION bila Anda keluar dari browser (EXIT), buka meminimize lho! Cara kedua, gunakan perintah berikut ini pada script.

```
session_destroy();
```

Dengan perintah tersebut, maka semua variabel SESSION akan terhapus. Namun, bagaimana bila hanya ingin menghapus suatu variabel SESSION tertentu saja? Untuk hal ini, gunakan perintah unset()

```
unset($_SESSION['nama session']);
```

## Penerapan SESSION

Biasanya, SESSION ini digunakan untuk keperluan autentifikasi user melalui login. Adapun idenya adalah apabila user sukses melakukan login, maka username ini akan disimpan dalam sebuah SESSION.

Mengapa username ini harus disimpan ke dalam SESSION? Ya... dengan disimpannya username ke dalam SESSION sekaligus bisa digunakan untuk mencegah adanya penyusup yang masuk ke aplikasi tanpa melalui login.

Lho? Kok bisa, logikanya bagaimana? Ya.. untuk mengecek ada tidaknya penyusup yang masuk ke aplikasi tanpa login, bisa hanya dengan mengecek SESSION nya. Jika penyusup tadi masuk ke aplikasi tanpa login, otomatis SESSION nya belum ada, karena SESSION baru ada ketika user login dengan sukses.

Lebih lanjut mengenai bagaimana menerapkan SESSION dalam autentifikasi user ini, baca artikel saya di <http://blog.rosihanari.net/apa-sih-gunanya-session-di-php/>

## Tugas Untuk Anda

Buatlah autentifikasi user dari suatu aplikasi web menggunakan konsep SESSION. Adapun username-username yang boleh mengakses aplikasi ini adalah

USERNAME	PASSWORD
A	password1
B	password2
C	password3
D	password4
E	password5

Username dan password di atas harus disimpan dalam sebuah ARRAY. Adapun konsepnya sama seperti pada pengerjaan soal no. 5 Bab 11.

Berikut ini gambaran aplikasinya:

Terlebih dahulu buatlah sebuah desain halaman web dengan layout seperti di bawah ini:

TITLE
KONTEN
FOOTER

Selanjutnya, bagi-bagilah layout tersebut ke dalam modul header.php dan footer.php. Untuk halaman utama, tampilkan form login untuk user.

TITLE	
Silakan Login	
Username	<input type="text"/>
Password	<input type="password"/>
[ SUBMIT ]	
FOOTER	

Selanjutnya bila login berhasil tampilkan halaman berikut ini (munculkan link navigasi)

TITLE
Username : XXX
Link 1   Link 2   Link 3   Logout
Anda telah berhasil login
FOOTER

Keterangan:

- Login dikatakan berhasil bila password yang dimasukkan oleh seorang user melalui form login **sama dengan** password yang tersimpan dalam aplikasi (dalam hal ini password yang ada dalam array).
- XXX adalah username yang sukses login.

Untuk setiap halaman Link 1, 2, dan 3 berikut ini tampilannya

TITLE
Username : XXX
Link 1   Link 2   Link 3   Logout
Ini adalah isi dari halaman Link 1
FOOTER

**Catatan:**

- Aplikasi yang dibuat harus bisa menghandle apabila login salah, misalnya muncul keterangan “Password yang dimasukkan salah” bila username terdaftar namun passwordnya salah, atau “Username tidak terdaftar” bila usernamenya tidak terdaftar dalam aplikasi.

Bila password login salah atau user tidak terdaftar, maka menu link navigasi tidak ditampilkan

- Aplikasi yang dibuat harus bisa menghandle adanya penyusup yang ingin langsung masuk ke halaman Link 1, 2 dan 3 tanpa melalui login (autentifikasi).
- Aplikasi yang dibuat harus bisa menghandle proses logout.



## BAB 14

# Penggunaan Cookie

---

Pada Bab 13, Anda telah belajar bagaimana menggunakan SESSION untuk menyimpan data atau value supaya dikenali di beberapa script halaman web. Selain menggunakan SESSION untuk melakukan hal ini, Anda juga bisa menggunakan COOKIE.

Apa perbedaan antara SESSION dan COOKIE? Dari sisi fungsi atau kegunaan, keduanya adalah sama, namun perbedaannya adalah jika SESSION value disimpan dalam server web sedangkan value COOKIE disimpan dalam komputer client.

Memang keduanya ada plus dan minusnya. Kelebihan SESSION adalah bahwa user tidak bisa menghapus value yang tersimpan kecuali dirinya sendiri, misal dengan menutup browser atau ketika dalam proses logout. Namun kelemahannya adalah ketika tiba-tiba jaringan internetnya putus setelah proses login. Bila jaringannya putus, maka otomatis SESSION akan terhapus sehingga apabila hal ini terjadi ketika Anda mengakses halaman web yang terdapat autentifikasi maka Anda diharuskan login kembali.

Sedangkan COOKIE, value disimpan di komputer client. Dengan demikian value tersebut tetap tersimpan meskipun jaringan internetnya putus atau dalam kondisi offline. Namun kelemahannya adalah value COOKIE tersebut bisa dihapus oleh orang lain yang juga menggunakan komputer tersebut, atau mungkin user komputer tersebut telah mengeset bahwa browsernya tidak bisa menerima COOKIE sehingga COOKIE tidak bisa tersimpan dalam komputer client.

Sehingga apabila ditanya, lebih baik pakai yang mana? COOKIE atau SESSION? Ya... hal ini saya serahkan kepada Anda saja. Sebagai gambaran, bahwa sistem login Google Mail sepertinya menggunakan COOKIE, demikian pula dengan Wordpress. Namun banyak juga situs atau aplikasi yang menggunakan SESSION untuk membuat sistem login ini. Saya sendiri lebih senang menggunakan SESSION 😊

### Cara Menyimpan Value Ke Dalam COOKIE

Untuk menyimpan value ke dalam COOKIE caranya adalah dengan memberikan perintah berikut ini:

```
setcookie("nama cookie", value, expired time);
```

Pada perintah di atas, parameter 'nama cookie' bisa diisi dengan sebarang nama. Nama COOKIE ini seperti halnya Anda memberi nama pada SESSION. Parameter 'value'

nantinya diisi dengan value yang akan disimpan. Sedangkan parameter 'expired time' adalah lama waktu untuk mengatur masa berlaku COOKIE.

Catatan:

- Nama COOKIE tidak boleh mengandung spasi
- Expired Time dihitung dalam satuan detik

Contoh:

```
<?php
$kata = "Saya sedang belajar COOKIE";
setcookie("mycookie", $kata, time()+3600);
?>
```

Script di atas akan menyimpan value berupa teks 'Saya sedang belajar COOKIE' ke dalam COOKIE. Bila script di atas dijalankan, maka kita bisa lihat valuenya dalam COOKIE. Apabila Anda menggunakan Firefox, maka cara melihat COOKIE nya adalah:

1. Klik TOOLS > OPTIONS
2. Pilih menu tab PRIVACY
3. Klik tombol SHOW COOKIES
4. Kemudian dicari nama situsnya, karena Anda mencobanya melalui LOCALHOST maka LOCALHOST pada site name
5. Lalu akan tampil seperti pada gambar berikut ini



Oya... pada script di atas, parameter 'expired time' nya diberikan `time()+3600` itu maksudnya bahwa cookie tersebut akan disimpan selama 1 jam (3600 detik). Setelah 1 jam maka cookie akan terhapus secara otomatis. Bila parameter 'expired time' ini tidak dituliskan maka cookie akan terhapus begitu browser ditutup, seperti halnya SESSION.

Nah... konsep COOKIE yang bisa diatur jangka waktu atau umurnya ini sering dimanfaatkan oleh program-program bisnis afiliasi. Mungkin Anda pernah mendengar atau bahkan sering bahwa suatu program afiliasi X menggunakan sistem COOKIE selama 3 bulan, 9 bulan atau bahkan tahunan. COOKIE dalam bisnis afiliasi ini digunakan untuk menyimpan nomor ID dari referalnya. Bila terdapat konsumen yang membeli produk di bawah ID referral seseorang selama usia COOKIE ini masih aktif, maka orang yang memiliki ID referral tadi akan mendapatkan komisi.

## Membaca Value Cookie

Setelah COOKIE tersimpan, selanjutnya kita bisa membaca valuenya. Untuk membaca value COOKIE, caranya adalah menggunakan perintah

```
$_COOKIE['nama cookie'];
```

Contoh:

```
<?php
if isset($_COOKIE['mycookie'])
```

```
{
    echo $_COOKIE['mycookie'];
}
else echo "Cookie Tidak Ada";
?>
```

Script di atas digunakan untuk menampilkan value dari COOKIE bernama 'mycookie' bila COOKIE tersebut ada (masih tersimpan dalam komputer), sedangkan bila sudah tidak ada maka tampilkan 'Cookie Tidak Ada'.

Nah... mungkin Anda berpikir, bagaimana bila terdapat nama COOKIE yang sama lebih dari satu yang tersimpan dalam komputer? Bila nama COOKIE yang sama tersebut berasal dari situs yang sama, maka sifat timpa menimpa value seperti halnya dalam variabel akan terjadi.

Contoh:

#### **script1.php**

```
<?php
$kata = "COOKIE 1";
setcookie("mycookie", $kata, time()+3600);
?>
```

#### **script2.php**

```
<?php
$kata = "COOKIE 2";
setcookie("mycookie", $kata, time()+3600);
?>
```

Kedua script di atas akan melakukan proses penyimpanan COOKIE sebanyak 2 kali dengan value yang berbeda namun nama COOKIE nya sama. Jika Anda jalankan script di atas dalam LOCALHOST (sebagai nama situsnya), maka value COOKIE pertama akan ditimpa dengan COOKIE yang kedua.

Namun... misalkan Anda punya dua situs atau server yang berbeda katakanlah A dan B, kemudian script1.php Anda upload ke A sedangkan script2.php diupload ke B. Selanjutnya script1.php di A dijalankan dan script2.php di B juga dijalankan. Bila diperhatikan, maka proses ini akan menghasilkan 2 buah COOKIE yang berbeda, meskipun nama COOKIE nya sama. Di sini yang membedakan adalah nama server atau situsnya.

## Tugas Untuk Anda

1. Buatlah sebuah halaman web berisi sebarang konten yang bisa menginformasikan nama pengunjungnya, kapan kunjungan terakhir.

Bila seseorang baru pertama kali mengunjungi halaman web tersebut, maka akan muncul form untuk memasukkan nama pengunjung. Berikut ini contoh gambarannya:

Misalkan ada seorang seseorang bernama AGUS sedang membuka halaman web tersebut untuk pertama kalinya pada tanggal 20/4/2009 melalui laptop pribadinya dan mengisi namanya pada form nama pengunjung yang disediakan.

Selanjutnya bila si AGUS tadi membuka kembali halaman web tersebut pada tanggal 29/4/2009 dengan laptop yang sama, maka form untuk memasukkan nama pengunjung tidak akan muncul melainkan muncul keterangan: **“Terima kasih AGUS atas kunjungannya kembali ke halaman ini. Anda terakhir mengunjungi halaman ini pada tanggal 20/4/2009”**.

Gunakan COOKIE untuk menyelesaikan kasus di atas!

2. Buatlah sistem login dan autentifikasi halaman web seperti halnya tugas di Bab 13. Namun untuk tugas kali ini silakan dibuat dengan menggunakan COOKIE.

## BAB 15

# Tentang Function

---

Dalam suatu pengaturan organisasi, biasanya perlu dibentuk semacam bagian-bagian kerja atau divisi yang spesifik kerjanya. Sebagai contoh misalkan dalam suatu organisasi ada divisi humas yang tugasnya spesifik mengurus masalah kehumasan, divisi HRD yang khusus menangani masalah SDM dan pengembangannya, serta divisi-divisi yang lain.

Mengapa dalam pengaturan organisasi perlu dibentuk divisi atau suborganisasi yang lebih kecil dalam kinerjanya? Ya... tujuannya adalah efisiensi kerja. Dalam hal ini, seorang ketua atau manajer pastilah akan repot bila semua kegiatan harus dibebankan kepadanya. Selain itu apabila dilihat dari efektifitas, jelas akan efektif karena dengan dibentuknya suborganisasi maka seorang ketua organisasi lebih mudah memberikan komando. Sebagai contoh misalkan ketika sang ketua butuh informasi tentang kehumasan, maka dia tinggal menyuruh divisi humas untuk bekerja dan memberikan informasi. Informasi tersebut nantinya akan digunakan sang ketua untuk menentukan kebijakan organisasi dalam proses selanjutnya.

Nah.. mengadopsi hal di atas, konsep pembentukan suborganisasi ini juga bisa diterapkan ke dalam suatu script atau program. Dalam hal ini, sebuah program ibaratnya sebuah organisasi besar. Sedangkan Anda (programmer) sebagai ketua organisasinya. Dengan dibuatnya suatu subprogram yang memiliki tugas atau kerja yang spesifik maka dapat membawa ke efisiensi dan efektifitas kerja dari program atau script tersebut. Sekaligus, Anda pun akan lebih mudah dalam mengatur proses programnya.

Perhatikan contoh script berikut ini yang digunakan untuk menghitung operasi

$$\frac{3^4 10^3}{4^5}$$

Dalam PHP sebenarnya terdapat perintah untuk menghitung pangkat bilangan. Namun di sini diasumsikan kita tidak menggunakan perintah tersebut melainkan harus membuat program sendiri.

Untuk menghitung  $n$  pangkat  $m$ , kita bisa menggunakan program berikut ini

```
$hasil = 1;
for ($i = 1; $i <= m; $i++)
{
    $hasil = $hasil * n;
}
```

```
}
```

Konsep program di atas didasarkan pada ide  $n^m = n \times n \times n \times \dots \times n$  (sejumlah  $m$  kali perkalian)

Nah... bila program di atas kita terapkan ke dalam kasus yang kita hadapi maka script atau program PHP akan menjadi seperti ini:

```
<?php

// menghitung 3 pangkat 4

$hasil_a = 1;
for ($i = 1; $i <= 4; $i++)
{
    $hasil_a = $hasil_a * 3;
}

// menghitung 10 pangkat 3

$hasil_b = 1;
for ($i = 1; $i <= 3; $i++)
{
    $hasil_b = $hasil_b * 10;
}

// menghitung 4 pangkat 5

$hasil_c = 1;
for ($i = 1; $i <= 5; $i++)
{
    $hasil_c = $hasil_c * 4;
}

// hitung hasil terakhir

$hasil = $hasil_a * $hasil_b / $hasil_c;

echo $hasil;

?>
```

Wah... ternyata panjang juga ya. Bila kita perhatikan bahwa dalam script di atas ada beberapa pekerjaan yang dilakukan secara berulang-ulang yaitu 'menghitung pangkat'. Nah... mungkin untuk keperluan efisiensi dan efektifitas, bagaimana seandainya kita buat '**Divisi Perpangkatan**' dalam program yang tugasnya khusus menghitung pangkat bilangan ☺

Dengan kehadiran divisi tersebut, kita dapat dengan cepat menghitung pangkat bilangan karena urusan memangkatkan bilangan kita serahkan ke divisi tersebut. Dalam programming, divisi ini kita namakan **function** atau subprogram.

OK... sejenak kita tinggalkan kasus di atas. Sekarang kita lanjutkan dahulu tentang bagaimana membuat function.

Secara umum, bentuk function adalah seperti di bawah ini

```
function nama_function(parameter)
{
    ..
    ..
    return variabel;
}
```

Keterangan:

Setiap function pasti dan harus memiliki nama function. Nama function ini nantinya akan dipanggil oleh program utama bila akan digunakan. Parameter di sini sifatnya optional (boleh ada, boleh tidak). Parameter ini ibaratnya input yang akan diolah oleh function.

Sedangkan **return variabel** merupakan perintah untuk memberikan hasil setelah dikerjakan oleh function. Dalam hal ini perintah **return variabel** ini juga bersifat optional (boleh ada, boleh tidak).

Kapan **return variabel** ini digunakan? Dan kapan tidak perlu digunakan? **return variabel** ini perlu digunakan bila hasil dari pengolahan function ini akan digunakan untuk proses yang lain dalam program. Sedangkan bila hasil dari function tidak akan digunakan oleh program, maka tidak perlu diberikan perintah ini.

Contoh:

```
<?php
```

```
function jumlah($a, $b)
{
```



```
$c = $a + $b;  
return $c;  
}  
  
echo "Hasil penjumlahannya = ".jumlah(3, 5);  
  
?>
```

Script di atas menunjukkan penggunaan function untuk menjumlahkan dua buah bilangan. Apabila dikembangkan lagi untuk kasus menghitung (3 + 5) ditambah (7 + 6), maka scriptnya dapat berbentuk seperti di bawah ini

```
<?php  
  
function jumlah($a, $b)  
{  
    $c = $a + $b;  
    return $c;  
}  
  
$hasil = jumlah(3, 5) + jumlah(7, 6);  
  
echo "Hasil penjumlahannya = ". $hasil;  
  
?>
```

Atau

```
<?php  
  
function jumlah($a, $b)  
{  
    $c = $a + $b;  
    return $c;  
}  
  
$hasil = jumlah(jumlah(3, 5), jumlah(7, 6));  
  
echo "Hasil penjumlahannya = ". $jumlah;  
  
?>
```

Sedangkan berikut ini contoh function yang tidak perlu menggunakan **return**. Script berikut ini hanya sekedar menampilkan suatu string melalui sebuah function.

```
<?php

function tulis($x)
{
    echo "Anda menampilkan ". $x . "<br>";
}

tulis("Hello World..");
tulis("Apa kabar?");

?>
```

Nah... kembali lagi ke kasus semula, yaitu mencari hasil dari

$$\frac{3^4 10^3}{4^5}$$

Maka kita bisa membuat script yang lebih sederhana daripada sebelumnya menggunakan function

```
<?php

function pangkat($m, $n)
{
    $hasil = 1;
    for ($i = 1; $i <= $n; $i++)
    {
        $hasil = $hasil * $m;
    }
    return $hasil;
}

$jwb = pangkat(3, 4) * pangkat(10, 3) / pangkat(4, 5);
echo "Hasilnya adalah : ".$jwb;

?>
```

Anda juga bisa membuat beberapa function dalam file tersendiri, semacam daftar function. Selanjutnya setiap kali butuh suatu function dalam daftar tersebut, cukup meng-**include**-kan saja nama file nya, lalu panggil nama functionnya. Trik ini sering digunakan oleh para programmer handal dalam membuat aplikasi. Berikut ini contohnya:

**functions.php**

```
<?php

function jumlah($a, $b)
{
    $hasil = $a + $b;
    return $hasil;
}

function kurang($a, $b)
{
    $hasil = $a - $b;
    return $hasil;
}

function kali($a, $b)
{
    $hasil = $a * $b;
    return $hasil;
}

function bagi($a, $b)
{
    $hasil = $a / $b;
    return $hasil;
}

?>
```

**jumlah.php**

```
<?php

include "functions.php";

$bil1 = 103;
$bil2 = 192;

echo "Hasil ".$bil1." + ".$bil2." = " . jumlah($bil1, $bil2) ;

?>
```

**kurang.php**

```
<?php

include "functions.php";

$bil1 = 103;
$bil2 = 192;

echo "Hasil ".$bil1." - ".$bil2." = " . kurang($bil1, $bil2) ;

?>
```

**kali.php**

```
<?php

include "functions.php";

$bil1 = 103;
$bil2 = 192;

echo "Hasil ".$bil1." x ".$bil2." = " . kali($bil1, $bil2) ;

?>
```

**bagi.php**

```
<?php

include "functions.php";

$bil1 = 103;
$bil2 = 192;

echo "Hasil ".$bil1." / ".$bil2." = " . bagi($bil1, $bil2) ;

?>
```

**Built in Functions dalam PHP**

Sebuah function dapat kita create sendiri seperti halnya di atas, namun dapat pula kita langsung gunakan karena sudah disediakan oleh PHP. Function yang sudah disediakan oleh PHP ini selanjutnya disebut *built in functions*.

Berikut ini beberapa built in functions yang sudah ada dalam PHP berdasarkan kategori penggunaannya.

### Mathematics Built in Functions

#### **abs()**

Untuk menghitung nilai mutlak atau absolute

Contoh:

```
<?php
$bil = -10;
echo abs($bil); // menghasilkan 10
?>
```

#### **ceil()**

Untuk membulatkan ke atas suatu bilangan real

Contoh:

```
<?php
$bil = 19.1;
echo ceil($bil); // menghasilkan 20
?>
```

#### **floor()**

Untuk membulatkan ke bawah suatu bilangan real

Contoh:

```
<?php
$bil = 19.5;
echo floor($bil); // menghasilkan 19
?>
```

#### **max()**

Mencari nilai terbesar dari suatu data bertipe array

Contoh:

```
<?php

$data = array(19, 23, 11, 45);

$max = max($data);
echo "Nilai max = ".$max; // menghasilkan 45

?>
```

**min()**

Mencari nilai minimum dari suatu data bertipe array

**mt\_rand(x, y)**

Untuk menghasilkan bilangan bulat random antara x s/d y.

Contoh:

```
<?php

$random = mt_rand(4, 10); // menghasilkan bilangan random antara 4 s/d 10
echo $random;

?>
```

**pow(x, y)**

Digunakan untuk mencari hasil x pangkat y.

Contoh:

```
<?php

$hasil = pow(4, -5); // menghitung 4 pangkat -5
echo $hasil;

?>
```

**round()**

Membulatkan bilangan real ke bawah bila desimal di belakang komanya kurang dari 0.5, dan membulatkan ke atas bila desimal di belakang komanya lebih dari atau sama dengan 0.5.

Contoh:

```
<?php

echo round(10.23); // menghasilkan 10
echo round(10.6); // menghasilkan 11
echo round(-10.2); // menghasilkan -10

?>
```

**Array Built in Functions**

Berikut ini beberapa function yang bisa digunakan untuk mengolah data berbentuk array

**array\_multisort()**

Digunakan untuk mensorting beberapa array terkait sekaligus. Bisa juga digunakan untuk mensorting dalam sebuah array saja.

Sintaks dari penggunaan array\_multisort() adalah

```
array_multisort(x, metode, a, b, c, ...);
```

dengan 'x' adalah array yang digunakan sebagai acuan dalam sorting, 'metode' adalah metode yang digunakan sorting (ascending atau descending), dan a, b, c,... adalah array lain yang ikut disorting.

Contoh:

Script berikut ini akan mengurutkan data mahasiswa (NIM dan NAMA) berdasarkan NIM dari 2 buah array yang bersesuaian secara ascending.

```
<?php

$nim = array("M0197002","M0197004","M0197001","M0197008","M0197003");
```

```
$nama = array("Amir","Joko","Budi","Siti","Agus");
```

```
array_multisort($nim, SORT_ASC, $nama);
```

```
for ($i = 0; $i <= count($nim)-1; $i++)  
{  
    echo $nim[$i]. " " . $nama[$i]. "<br>";  
}
```

```
?>
```

Keterangan:

Untuk sorting secara descending, gunakan parameter SORT\_DESC

#### **array\_rand()**

Digunakan untuk mengambil salah satu elemen dari array secara random

Contoh:

```
<?php  
$bil = array(3, 1, 4, 5, 2, 6);  
$ambilAcak = array_rand($bil);  
echo $ambilAcak;  
?>
```

#### **array\_reverse()**

Digunakan untuk membalik urutan data dalam array

Contoh:

```
<?php  
  
$data = array("mangga","jambu","jeruk","apel");  
$balik = array_reverse($data);  
  
foreach($balik as $index => $buah)  
{  
    echo $buah. "<br>"; // menampilkan apel, jeruk, jambu, mangga  
}
```



?>

### **array\_search()**

Digunakan untuk mencari nomor urut elemen tertentu dalam array (dimulai dari 0)

Contoh:

Script berikut ini akan mencari nomor urut elemen dari data 'jambu' dalam array buah

```
<?php
```

```
$data = array("mangga","jambu","jeruk","apel");  
$no = array_search("jambu", $data);
```

```
echo $no; // akan menampilkan 1 (nomor urut elemen dalam array)
```

```
?>
```

### **count()**

Digunakan untuk menghitung jumlah elemen yang ada dalam suatu array.

## **String Built in Functions**

### **explode()**

Digunakan untuk memecah suatu string ke dalam array berdasarkan karakter tertentu.

Contoh:

```
<?php
```

```
$date = "20-01-2009";
```

```
$pecah = explode("-", $date); // memecah string berdasarkan karakter '-'  
echo "Tanggal : ". $pecah[0]; // menampilkan 20  
echo "Bulan : ". $pecah[1];   // menampilkan 01  
echo "Tahun : ". $pecah[2];   // menampilkan 2009
```

```
?>
```

**md5()**

Digunakan untuk mengenkripsi (*hashing*) suatu string. Biasanya function ini digunakan untuk keperluan enkripsi password login ke suatu aplikasi. Hasilnya adalah suatu string yang tidak dapat dibaca dengan panjang 32 karakter

```
<?php
$passwordAsli = "hello";
$passwordEnkrip = md5($passwordAsli);
echo $passwordEnkrip; // menampilkan '5d41402abc4b2a76b9719d911017c592'
?>
```

Keterangan:

Tidak ada function untuk membalik dari hasil md() ke string asli.

**number\_format()**

Memformat angka dengan mengelompokkan berdasarkan ribuan.

Contoh:

Berikut ini contoh script untuk menyajikan angka Rp. 1002372 menjadi bentuk Rp. 1.002.372,-

```
<?php

$harga = 1002372;
$formatted = number_format($harga, 0, "", ".");
echo "Harganya Rp. ".$formatted. "-";

?>
```

Keterangan:

Parameter pertama dari number\_format() menunjukkan bilangan yang akan diformat, parameter kedua menunjukkan jumlah digit desimal di belakang koma, parameter ketiga menunjukkan tanda pemisah untuk digit desimal di belakang koma, dan parameter keempat menunjukkan tanda pemisah untuk ribuan.

Contoh:

Berikut ini contoh script untuk menyajikan bilangan real 123456789.111111 menjadi bentuk 123.456.789,111 (tanda pemisah desimal menggunakan koma, jumlah digit desimal sebanyak 3 dan menggunakan tanda titik untuk pemisah ribuan)

```
<?php

$bil = 123456789.111111;
$formatted = number_format($bil, 3, ",", ".");
echo "Bilangannya ".$formatted;

?>
```

### str\_replace()

Digunakan untuk mereplace suatu substring dengan substring lain dalam string.

Contoh:

Script berikut ini akan mengganti substring 'be' menjadi 'bi'

```
<?php

$kalimat = "saya ingin bebas seperti burung beo";
$replace = str_replace("be", "bi", $kalimat);

echo $replace; // menampilkan 'saya ingin bibas seperti burung bio'

?>
```

### strip\_tags()

Digunakan untuk menghilangkan tag HTML dalam suatu string. Biasanya untuk keamanan dalam form guna mencegah disisipkannya tag-tag HTML yang bisa merusak halaman web.

```
<?php

$string = "<font face='verdana' size='2'>Hello world</font>";
$hilangkanTag = strip_tags($string);

echo $hilangkanTag; // akan menghasilkan 'Hello World'
```

?>

**strlen()**

Digunakan untuk menghitung jumlah karakter dari suatu string

Contoh:

```
<?php
```

```
$kata = "Hello World";  
$jumKarakter = strlen($kata);
```

```
echo $jumKarakter; // akan menghasilkan 11 (spasi ikut dihitung)
```

```
?>
```

**strtolower()**

Digunakan untuk mengubah semua huruf penyusun string menjadi huruf kecil

Contoh:

```
<?php
```

```
$kata = "Rosihan Ari Yuana";  
$kecil = strtolower($kata);
```

```
echo $kecil; // menghasilkan 'rosihan ari yuana'
```

```
?>
```

**strtoupper()**

Merupakan kebalikan strtolower()

**substr\_count()**

Menghitung jumlah substring dalam suatu string

Contoh:

```
<?php
```

```
$lagu = "topi saya bundar, bundar topi saya, kalau tidak bundar, bukan topi saya";  
$hitungTopi = substr_count($lagu, "topi");
```

```
echo $hitungTopi; // akan menghasilkan 3.
```

```
?>
```

<b>substr()</b>
-----------------

Mengambil sejumlah karakter substring dari suatu string

Contoh:

Script berikut ini akan mengambil tahun angkatan mahasiswa dari NIM 'M0197001', dimana tahun angkatan terletak pada digit ke 4 dan 5 dari NIM (dalam hal ini tahun angkatannya adalah 97)

```
<?php
```

```
$nim = "M0197001";  
$angkatan = substr($nim, 3, 2);
```

```
echo $angkatan; // menghasilkan 97
```

```
?>
```

Keterangan:

Parameter pertama dari substr() menunjukkan string aslinya. Parameter kedua menunjukkan posisi awal substring yang akan diambil (dihitung mulai 0 dari karakter paling kiri string), dan parameter ketiga menunjukkan jumlah karakter substring yang akan diambil.

Sehingga dari contoh di atas, karena posisi awal tahun angkatan itu terletak pada karakter ke-3 dari string NIM, dan tahun angkatan ini terdiri dari 2 karakter maka perintahnya adalah

```
substr($nim, 3, 2)
```

## Tugas Untuk Anda

1. Di SMA tentu Anda telah mengenal rumus kombinasi  $C(m, n)$  kan?  $C(m, n)$  dirumuskan dengan

$$C(m, n) = \frac{m!}{n!(m-n)!}$$

Simbol ! menunjukkan faktorial. Sebagai contoh,  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ .

Nah.. buatlah script PHP untuk menghitung nilai  $C(m, n)$  dengan  $m$  dan  $n$  nya suatu input, dimana  $m \geq n$ .

Dalam hal ini buatlah sebuah function yang khusus untuk menghitung nilai faktorial suatu bilangan. Selanjutnya gunakanlah function tersebut untuk menghitung  $C(m, n)$ .

2. Buatlah script untuk menghitung jumlah bilangan ganjil antara 10 dan 123456. Gunakan function untuk mengecek apakah suatu bilangan termasuk bilangan ganjil atau tidak.
3. Buatlah script untuk mencari selisih jumlah detik dari dua waktu yang berbeda. Format input waktu harus dalam bentuk hh:mm:ss

Contoh:

Waktu 1 = 12:03:10

Waktu 2 = 13:03:20

Selisih kedua waktu adalah 3610 detik.

Petunjuk: Anda dapat gunakan function explode() untuk memecah waktu ke dalam satuan jam, menit dan detik untuk perhitungan.

4. Misalkan dalam suatu kelas terdapat 5 orang siswa dengan nama dan NIS sebagai berikut

NIS	Nama Siswa
-----	------------

A001	Agus
------	------

A002	Budi
------	------

A003 Amir  
A004 Acong  
A005 Siti

Buatlah sebuah form untuk memasukkan nilai ujian 1 dan ujian 2 pelajaran Matematika kelima siswa tersebut. Selanjutnya tampilkan data NIS, Nama Siswa dan Rata-rata Ujiannya, dan urutkan berdasarkan Rata-rata Ujiannya mulai dari yang paling tinggi sampai yang paling rendah.

Contoh Tampilan Form Input Nilai

NIS	Nama Siswa	Nilai Ujian 1	Nilai Ujian 2
A001	Agus	<input type="text"/>	<input type="text"/>
A002	Budi	<input type="text"/>	<input type="text"/>
A003	Amir	<input type="text"/>	<input type="text"/>
A004	Acong	<input type="text"/>	<input type="text"/>
A005	Siti	<input type="text"/>	<input type="text"/>

Petunjuk:

Data NIS dan Nama Siswa disimpan dalam array, begitu pula dengan nilai rata-ratanya. Selanjutnya sort berdasarkan nilai rata-rata (gunakan `array_multisort()`) lalu tampilkan.

5. Suatu perusahaan X, terdapat 10 orang karyawan dengan NIK dan Nama Karyawannya sebagaimana tampak pada tabel berikut ini

NIK	Nama Karyawan	Masa Kerja (tahun)
K01001	A	20
K01002	B	18
K03001	C	12
K03002	D	12
K03003	E	10
K04001	F	8
K04002	G	11
K04003	H	9
K04004	I	7
K04005	J	14

Perhatikan kode NIK di atas. Digit ke 2 dan 3 menunjukkan posisi karyawan dalam perusahaan. '01' berarti di bagian keuangan, '03' di bagian HRD dan '04' di bagian maintenance.

Buatlah script PHP untuk menampilkan gaji setiap karyawan dengan ketentuan sbb:

Jika karyawan di bagian keuangan, maka gaji pokoknya: Rp. 2.000.000,-. Di bagian HRD gaji pokoknya Rp. 1.900.000,- dan di maintenance gaji pokoknya Rp. 1.950.000,-.

Selain itu masih ditambah tunjangan sesuai masa kerjanya.

Jika masa kerja > 15 tahun maka tunjangannya Rp. 500.000,- Sedangkan jika masa kerja 10 s/d 15 tahun, tunjangannya Rp. 300.000,-. Dan jika masa kerjanya di bawah 10 tahun tunjangannya Rp. 100.000,-

Untuk format tampilan, gunakan tabel berikut ini

NIK	Nama Karyawan	Masa Kerja (tahun)	Gaji Kotor
K01001	A	20	Rp. XXX,-
K01002	B	18	Rp. XXX,-
K03001	C	12	Rp. XXX,-
K03002	D	12	Rp. XXX,-
K03003	E	10	Rp. XXX,-
K04001	F	8	Rp. XXX,-
K04002	G	11	Rp. XXX,-
K04003	H	9	Rp. XXX,-
K04004	I	7	Rp. XXX,-
K04005	J	14	Rp. XXX,-
		<b>Total</b>	Rp. XXX,-

Catatan: Gunakan format angka gaji sesuai format Rp. XXX.XXX.XXX,-

6. Berdasarkan data karyawan yang ada pada soal no. 5, buatlah script untuk menampilkan data karyawan (NIK, nama dan masa kerja) dan jumlah karyawannya dari ketentuan sbb:
  - a. Karyawan di bagian 'HRD' yang masa kerjanya di atas 15 tahun
  - b. Karyawan di bagian 'maintenance' atau 'keuangan' yang masa kerjanya 10 tahun s/d 15 tahun



Gunakan Radiobutton untuk proses switching tampilan sebagaimana contoh form berikut ini.

**Data Karyawan**

- ☒ Karyawan di bagian 'HRD' dengan masa kerja > 15 tahun
- ☐ Karyawan di bagian 'Maintenance' atau 'Keuangan' dengan masa kerja 10 – 15 th

SUBMIT

## BAB 16

# Pengenalan SQL

---

Pada bab ini Anda akan diperkenalkan dengan SQL atau *structured query language*. SQL adalah suatu statement atau pernyataan atau perintah standart yang digunakan untuk keperluan administrasi database.

Ada banyak software database yang mendukung SQL ini. Software database ini disebut juga *SQL based DBMS* (Database Management System). Beberapa di antaranya adalah MySQL, Oracle, MS. Access, MS. SQL, SQL Server dll.

Bab pengenalan SQL ini diberikan karena mulai bab selanjutnya materi kursus terkait dengan penggunaan database, dalam hal ini software database yang digunakan adalah MySQL. Mengapa MySQL yang digunakan? Ya... karena MySQL adalah DBMS yang bersifat free alias gratis namun kehandalannya tidak kalah dengan DBMS yang tidak free.

### Tentang Database

Database ibaratnya suatu wadah yang digunakan untuk menyimpan data. Supaya mudah diatur data-data itu disimpan dalam sebuah tabel.

Database ibaratnya seperti sebuah lemari kabinet. Dalam lemari kabinet terdapat beberapa laci yang di dalamnya terdapat beberapa dokumen file atau data. Laci inilah yang diumpamakan tabel dari database dan dokumen file atau data yang tersimpan dalam laci ibaratnya merupakan data atau record dari database. Bisa membayangkan ya? 😊

Di dalam sebuah database **bisa terdapat lebih dari satu tabel**.

Pada sebuah tabel, terdapat unsur yaitu kolom dan baris, seperti halnya tabel-tabel umum yang Anda kenal. Kolom pada database disebut juga ***field***, dan barisnya disebut ***record***.

## Tentang PhpMyAdmin

Terkadang dalam mengatur atau pengadministrasian database sangatlah repot. Misalnya membuat database, membuat tabel, mengubah struktur tabel, menyimpan data ke database, mengedit data, menghapus data dsb.

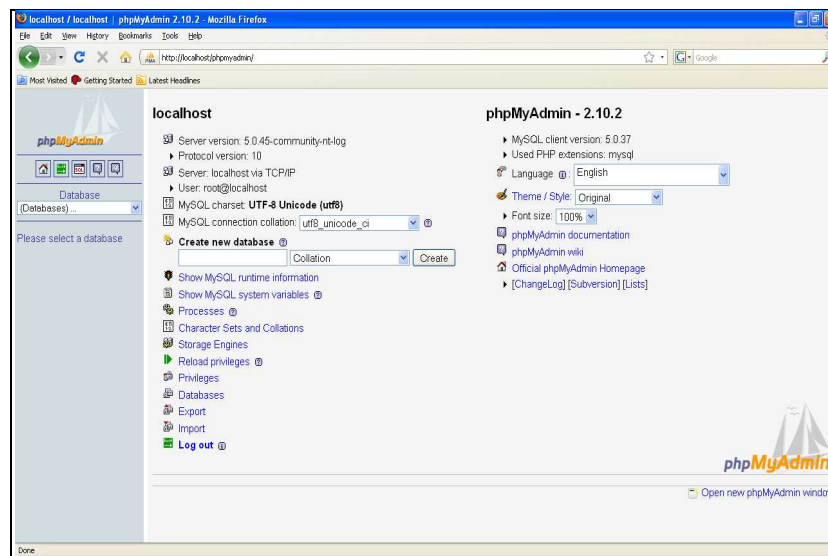
Nah... khusus MySQL terdapat software khusus yang dapat dipergunakan untuk pengadministrasian database supaya mudah dilakukan. Software ini bernama PhpMyAdmin. Software ini merupakan web based software yang dijalankan melalui URL:

<http://localhost/phpmyadmin>

Apabila Anda menggunakan AppServ atau XAMPP maka ketika dipanggil URL di atas akan langsung masuk ke PhpMyAdmin.

Catatan:

Sebelum masuk ke PhpMyAdmin biasanya kita disuruh memasukkan username dan password. Username dan password ini sesuai dengan yang pernah kita masukkan sewaktu proses instalasi AppServ.



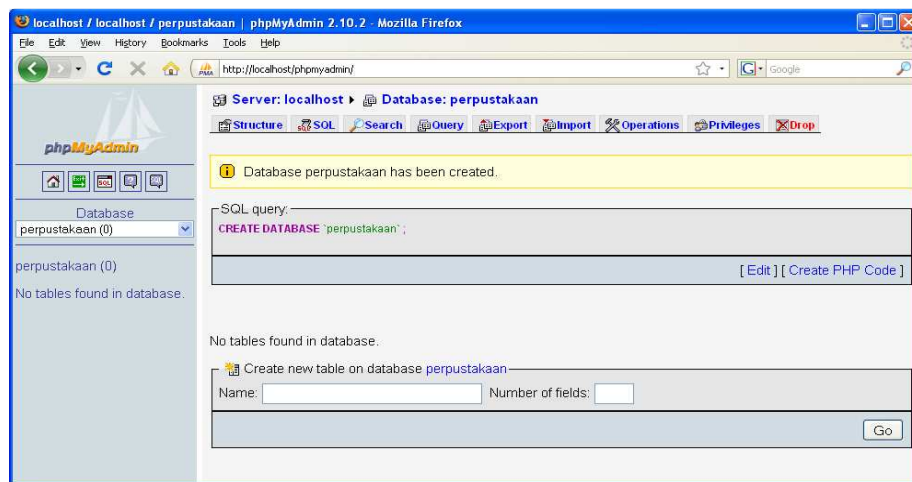
Tampilan PhpMyAdmin

## Membuat Database

Untuk membuat database baru dengan PhpMyAdmin, caranya adalah isikan nama database yang akan Anda buat ke dalam bagian kotak CREATE NEW DATABASE lalu klik tombol CREATE.

Sebagai contoh silakan Anda buat database dengan nama: 'perpustakaan'

**Tips:** sebaiknya nama database jangan mengandung spasi.



## Membuat Tabel

Setelah database dibuat, selanjutnya kita bisa membuat tabel. Untuk membuat tabel baru dalam database yang telah kita buat caranya ketikkan nama tabel yang akan dibuat pada bagian CREATE NEW TABLE ON DATABASE XXX. 'XXX' di sini merupakan nama databasenya.

Selain itu masukkan pula jumlah fieldnya. Oleh karena itu sebaiknya sebelum proses pembuatan tabel hendaknya Anda rancang terlebih dahulu nama-nama fieldnya sesuai yang dibutuhkan.

Dalam contoh ini andaikan kita akan membuat tabel 'kategori\_buku'

**Tips:** seperti halnya membuat nama database, untuk nama tabel juga sebaiknya jangan gunakan spasi.

Untuk tabel 'kategori\_buku' misalnya kita desain seperti di bawah ini:

Tabel : kategori\_buku

Field	: kode_kat	varchar(5)	PRIMARY KEY
	nama_kat	varchar(20)	

Keterangan:

Dalam tabel 'kategori buku' terdapat 2 buah field yaitu 'kode\_kat' dan 'nama\_kat'. Field 'kode\_kat' nanti berisi kode kategori dari buku-buku yang tersimpan dalam perpustakaan dan 'nama\_kat' merupakan penjelasan yang berisi nama kategori terkait dengan kode kategorinya. Misalnya kode kategori buku 'X10' itu merupakan kode kategori untuk buku tentang 'MATEMATIKA'.

Varchar merupakan tipe data dari fieldnya. Tipe data ini harus dimiliki setiap field yang akan dibuat. Tipe data ini menentukan jenis atau sifat data yang akan disimpan dalam field tersebut. Terdapat beberapa tipe data yang bisa dipilih, yaitu

Varchar	bila datanya berbentuk karakter (maks 255 karakter)
Int	bila datanya berbentuk bilangan bulat
Float	bila datanya berbentuk bilangan real
Text	bila datanya berbentuk karakter (size nya lebih besar dari varchar)
Date	bila datanya berbentuk tanggal (th-bln-tgl)
Time	bila datanya berbentuk waktu (jam:menit:detik)
Blob	bila datanya berupa file

Catatan:

Khusus untuk varchar harus diberikan lebar size yang diinginkan, misalnya suatu field bertipe varchar(5). Maksud 5 di sini adalah banyak karakter yang bisa disimpan dalam field tersebut adalah 5 buah.

PRIMARY KEY adalah fitur yang ada dalam DBMS yang dapat digunakan untuk menjamin bahwa sebuah data itu tunggal. Dalam contoh di atas PRIMARY KEY diberikan pada field 'Kode Kategori'. Tentu kita tahu bahwa suatu kode kategori buku itu bersifat tunggal atau tidak boleh ada kode yang sama, seperti halnya kode karyawan atau kode mahasiswa. Bila suatu field sudah dijadikan PRIMARY KEY maka tidak mungkin ada data yang sama pada field tersebut. Apabila kita masukkan kode yang sama, maka akan ditolak oleh sistem.

OK... setelah kita rancang tabelnya, kita bisa membuatnya. Berikut ini caranya:

1. Isikan nama tabel di kotak CREATE NEW TABLE ON DATABASE ...  
Dalam kasus ini masukkan 'kategori\_buku'

2. Masukkan jumlah field dari tabel yang akan dibuat pada NUMBER OF FIELDS  
Dalam kasus ini masukkan '2'
3. Klik GO

Setelah itu masukkan nama-nama field dan tipe datanya nya pada form yang muncul berikutnya

Field	Type ?	Length/Values <sup>1</sup>	
kode_kat	VARCHAR	5	
nama_kat	VARCHAR	20	

Menunjukkan primary key telah dipilih pada field 'kode\_kat'

Sebagai latihan coba Anda buat lagi tabel 'tabel\_buku' dengan field sebagai berikut

```

kode_buku    varchar(10)    PRIMARY KEY
judul_buku   text
kode_kat     varchar(5)
pengarang    varchar(30)
penerbit     varchar(30)
thn_terbit   int
  
```

Tabel di atas dibuat dalam database 'perpustakaan'. Dengan demikian terdapat 2 tabel dalam database 'perpustakaan'.

## Memasukkan Data/Record ke Tabel

Untuk memasukkan sebuah record ke tabel, perintah SQL nya adalah berikut ini

```
INSERT INTO nama_tabel (field1, field2, ...) VALUES (data1, data2, ..);
```

Sebagai contoh misalkan kita akan memasukkan record kode kategori: 'X001' dan nama kategori: 'MAJALAH' ke tabel 'kategori\_buku', maka perintahnya

```
INSERT INTO kategori_buku (kode_kat, nama_kat) VALUES ('X001', 'MAJALAH');
```

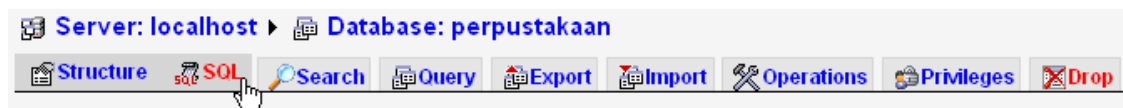
**Catatan Penting:**

Khusus field yang bertipe VARCHAR, TEXT, DATE, TIME, dan BLOB, value yang akan dimasukkan ke tabel harus diapit dengan tanda petik tunggal, misal 'X001'. Namun untuk field bertipe data bilangan (INT, FLOAT) tanpa menggunakan tanda petik.

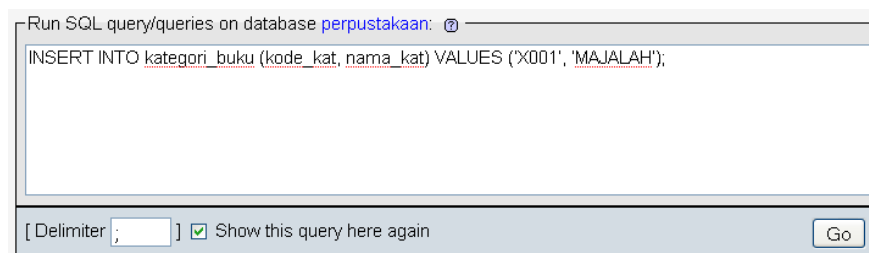
Dimanakah kita menuliskan perintah SQLnya? Ya... caranya adalah klik terlebih dahulu nama database yang kita gunakan pada kolom sebelah kiri dari PhpMyAdmin, dalam hal ini klik pada nama database 'perpustakaan'



Lalu klik tombol SQL pada menu atas PhpMyAdmin



Kemudian tulis perintah SQL pada kotak 'RUN SQL QUERY/QUERIES ON DATABASE ...'



Selanjutnya klik GO.

Anda dapat menuliskan perintah atau query SQL lebih dari satu sekaligus.

Untuk melihat record yang telah tersimpan ke dalam tabel, caranya klik nama tabel yang mau dilihat recordnya pada sisi kiri PhpMyAdmin



Lalu klik menu BROWSE pada bagian atas PhpMyAdmin



Selanjutnya akan muncul semua record yang telah disimpan pada tabel tersebut

	kode_kat	nama_kat
<input type="checkbox"/>	X001	MAJALAH

Sebagai latihan coba masukkan record-record berikut ini ke dalam tabel masing-masing menggunakan query SQL.

Tabel 'kategori\_buku'

Kode Kategori	Nama Kategori
X002	REFERENSI
X003	SURAT KABAR
X004	JURNAL ILMIAH

Tabel 'tabel\_buku'



Kode Buku	Judul Buku	Kode Kategori	Pengarang	Penerbit	Th. Terbit
B001	Pemrograman Pascal	X002	Mr. X	PT. A	2003
B002	Pemrograman PHP	X002	Mr. Y	PT. A	2004
B003	Kompas (Edisi X 2006)	X003	-	-	2006
B004	Sains Matematika (Edisi XI 2008)	X004	-	LIPI	2008
B005	Tempo (Edisi XI 2008)	X001	-	PT. B	2008

## Mengedit Record

Apabila Anda ingin mengedit record, perintah SQL nya adalah:

```
UPDATE nama_tabel SET field1 = value, field2 = value, ... [WHERE syarat]
```

Dalam hal ini WHERE merupakan syarat yang bersifat optional. Bila WHERE tidak diberikan, maka proses update data diberikan ke semua record.

Contoh:

Perintah SQL untuk mengedit tahun terbit buku berkode 'B001' menjadi 2010.

```
UPDATE tabel_buku SET thn_terbit = 2010 WHERE kode_buku = 'B001';
```

Perintah SQL untuk mengedit tahun terbit buku berkode 'B002' menjadi 2004 sekaligus mengedit kode kategorinya menjadi 'X002'.

```
UPDATE tabel_buku SET thn_terbit = 2004, kode_kat = 'X002' WHERE kode_buku = 'B002';
```

Perintah SQL untuk mengubah judul buku yang diterbitkan oleh PT. A pada tahun 2003 menjadi 'XXX'.

```
UPDATE tabel_buku  
SET judul_buku = 'XXX'  
WHERE thn_terbit = 2003 AND penerbit = 'PT. A';
```

Catatan:

Selain operator logika AND, Anda juga bisa menggunakan OR sebagai operator logika 'atau'.

Perintah SQL untuk mengubah nama penerbit menjadi PT. C dari semua buku yang tahun terbitnya di atas 2004.

```
UPDATE tabel_buku  
SET penerbit = 'PT. C'  
WHERE thn_terbit > 2004;
```

Catatan:

Untuk operator relasional, Anda bisa menggunakan tanda =, >, >=, <=, <> (tidak sama dengan)

Perintah SQL untuk mengubah tahun terbit buku menjadi 2005 semuanya

```
UPDATE tabel_buku SET thn_terbit = 2005;
```

## Menghapus Record

Sebuah record atau beberapa record dapat dihapus dari tabel dengan perintah SQL

```
DELETE FROM nama_tabel [WHERE syarat];
```

Contoh:

Perintah SQL untuk menghapus data buku yang berkode 'B001'

```
DELETE FROM tabel_buku WHERE kode_buku = 'B001';
```

Perintah SQL untuk menghapus data buku dari penerbit PT. A yang tahun terbitnya 2004

```
DELETE FROM tabel_buku WHERE penerbit = 'PT. A' AND thn_terbit = 2004;
```

## Tentang SQL SELECT

Perintah SELECT digunakan untuk mencari data atau menampilkan data pada field-field tertentu. Sintaksnya adalah:

```
SELECT field1, field2, ... FROM nama_tabel [WHERE syarat ] [ORDER BY field DESC|ASC];
```

Keterangan:

ORDER BY digunakan untuk mensorting hasil data yang ditampilkan berdasarkan field tertentu baik secara DESCENDING atau ASCENDING.

Contoh:

Perintah SQL untuk menampilkan data (kode buku dan judul buku) dari buku yang minimal terbit pada tahun 2003.

```
SELECT kode_buku, judul_buku
FROM tabel_buku
WHERE thn_terbit >= 2003;
```

Perintah SQL untuk menampilkan data (untuk semua field) dari buku yang diterbitkan oleh PT. A

```
SELECT *
FROM tabel_buku
WHERE penerbit = 'PT. A';
```

Perintah SQL untuk menampilkan data (kode buku, judul buku, tahun terbit) yang diurutkan berdasarkan tahun terbit mulai dari yang paling terbaru sampai dengan terlama.

```
SELECT kode_buku, judul_buku, thn_terbit
FROM tabel_buku
ORDER BY thn_terbit DESC;
```

Perintah SQL untuk menampilkan data (kode buku, judul buku dan nama kategori buku) dari buku yang diterbitkan oleh PT. A pada tahun 2003.

Untuk hal ini, kita harus bekerja di dalam 2 buah tabel sekaligus. Kode buku dan judul buku diperoleh dari tabel 'tabel\_buku', sedangkan nama kategori diperoleh dari tabel 'kategori\_buku'.

Kedua tabel saling berhubungan karena masing-masing terdapat field bernama 'kode kategori' atau 'kode\_kat'.

```
SELECT tabel_buku.kode_buku, tabel_buku.judul_buku, kategori_buku.nama_kat
FROM tabel_buku, kategori_buku
WHERE tabel_buku.kode_kat = kategori_buku.kode_kat AND
      tabel_buku.penerbit = 'PT. A' AND
      tabel_buku.thn_terbit = 2003;
```

Perintah SQL untuk menampilkan kode buku dan judul buku, yang judul bukunya mengandung kata 'MATEMATIKA'

```
SELECT kode_buku, judul_buku
FROM tabel_buku
WHERE judul_buku LIKE '%MATEMATIKA%';
```

Perintah SQL untuk menampilkan kode buku dan judul buku, yang kata depan judul bukunya adalah 'MATEMATIKA'

```
SELECT kode_buku, judul_buku
FROM tabel_buku
WHERE judul_buku LIKE 'MATEMATIKA%';
```

Perintah SQL untuk menampilkan kode buku dan judul buku, yang judul bukunya berakhiran dengan kata 'MATEMATIKA'

```
SELECT kode_buku, judul_buku
FROM tabel_buku
WHERE judul_buku LIKE '%MATEMATIKA';
```

## Penggunaan Aggregate Function

Anda juga bisa menggunakan function yang sudah disediakan oleh MySQL, seperti AVG() untuk menghitung rata-rata nilai, MAX() untuk menghitung nilai maksimum, MIN() untuk mencari nilai minimum, COUNT() untuk menghitung cacah data, SUM() untuk menjumlahkan nilai dari beberapa data.

Contoh:

Perintah SQL untuk mencari jumlah buku yang ada di dalam database:

```
SELECT count(kode_buku) FROM tabel_buku;
```

Perintah SQL untuk mencari jumlah buku yang memiliki kode kategori 'X002':

```
SELECT count(kode_buku) FROM tabel_buku WHERE kode_kat = 'X002';
```

## Tugas Untuk Anda

1. Buatlah database dengan nama 'dbmhs'. Selanjutnya buatlah tabel untuk keperluan penyimpanan data perkuliahan mahasiswa dalam database tersebut.

Berikut ini tabel-tabelnya:

Nama Tabel : 'mhs'

Nama Field :

- nim
- namaMhs
- tglLahir
- alamat

Nama Tabel : 'mk'

Nama Field :

- kodeMK
- namaMK
- sks
- smt

Nama Tabel : 'ambilMK'

Nama Field :

- nim
- kodeMK
- nilai

Tentukan sendiri tipe data yang cocok untuk setiap fieldnya, dan tentukan PRIMARY KEY nya. Sebutkan alasan pemilihan tipe data dan PRIMARY KEY nya.

2. Selanjutnya sisipkan ke dalam masing-masing tabel 5 buah record menggunakan perintah SQL
3. Tuliskan perintah SQL untuk proses berikut ini
  - a. Hapus data mahasiswa yang bernim 'M0197001'
  - b. Hapus matakuliah yang memiliki jumlah SKS 4 yang ada di semester 2.
  - c. Hapus matakuliah yang memiliki jumlah SKS 2 s/d 4
  - d. Hapus matakuliah yang memiliki jumlah SKS 4 atau di semester 3
  - e. Ubahlah nilai matakuliah berkode 'K001' yang diambil mahasiswa bernim 'M0197001' menjadi 3.
  - f. Ubahlah semua nilai mahasiswa yang mengambil matakuliah berkode 'K001' menjadi satu lebihnya dari nilai semula. Contoh nilai sebelumnya 2, maka diubah menjadi  $2+1 = 3$ .

- g. Tampilkan data matakuliah (kode matakuliah, nama matakuliah) yang berada di semester 2
- h. Tampilkan jumlah mahasiswa yang beralamat di 'SOLO'.
- i. Tampilkan data mahasiswa (nim, nama mahasiswa) yang namanya mengandung kata 'JOKO'.
- j. Tampilkan data mahasiswa (nim, nama mahasiswa) yang mengambil matakuliah berkode 'K001'
- k. Tampilkan nilai tertinggi dari matakuliah berkode 'K001' yang diambil mahasiswa.
- l. Tampilkan data mahasiswa (nim, nama mahasiswa, nilai matakuliah) yang mengambil matakuliah berkode 'K001' diurutkan berdasarkan nilai mulai dari tertinggi sampai terendah
- m. Tampilkan rata-rata nilai matakuliah mahasiswa yang mengambil matakuliah berkode 'K001'
- n. Tampilkan daftar matakuliah (kode matakuliah, nama matakuliah) yang pernah diambil oleh mahasiswa bernim 'M0197001'
- o. Tampilkan nilai IPK (index prestasi kumulatif) dari mahasiswa bernim 'M0197001'.

## BAB 17

# Konsep INSERT Data dengan PHP

---

Setelah Anda mengenal beberapa bentuk perintah SQL untuk memanipulasi atau mengolah data di MySQL, pada bab ini Anda akan diperkenalkan bagaimana menerapkan konsep tersebut yang terintegrasi dengan script PHP.

Pada bab ini terlebih dahulu akan diperkenalkan bagaimana konsep inserting data ke MySQL melalui PHP.

### Koneksi PHP ke MySQL

Sebelum kita membahasnya lebih lanjut tentang inserting data, terlebih dahulu akan disinggung tentang bagaimana caranya melakukan koneksi dari PHP ke MySQL.

Perintah PHP untuk koneksi ke MySQL adalah:

```
mysql_connect('nama host', 'nama user', 'password');  
mysql_select_db('nama db');
```

Keterangan:

- Nama Host : nama komputer di mana MySQL nya berada, kalo berada di komputer itu sendiri maka nama hostnya: 'localhost', bisa juga diberi nomor IP dari komputernya.
- Nama User : nama user yang memiliki hak akses ke MySQL nya
- Password : password dari nama user yang punya hak akses ke MySQL
- Nama DB : nama database yang akan dipilih

Contoh:

Berikut ini adalah script untuk melakukan koneksi ke MySQL dengan nama host: localhost, user : root, password: root, dan nama database: contoh

```
<?php  
mysql_connect('localhost', 'root', 'root');  
mysql_select_db('contoh');  
?>
```

Sebaiknya script di atas kita simpan dalam file script tersendiri, misalkan **koneksi.php**. Selanjutnya script tersebut kita includekan ke dalam script lain yang di dalamnya terdapat query-query SQL.

## Menjalankan Query SQL di PHP

Setelah kita melakukan koneksi dari PHP ke MySQL, selanjutnya kita bisa menjalankan query melalui script PHP.

Untuk menjalankan query SQL di PHP, kita gunakan perintah

```
mysql_query(query);
```

Contoh:

Berikut ini contoh script PHP untuk menjalankan query SQL untuk insert data ke tabel MySQL

```
<?php
```

```
include "koneksi.php";
```

```
$query = "INSERT INTO namaTabel(field1, field2) VALUES(value1, value2) ";  
mysql_query($query);
```

```
?>
```

Selanjutnya akan dibahas bagaimana mengimplementasikan konsep inserting data ke MySQL melalui script PHP yang diintegrasikan dengan form input

## Studi Kasus

Dalam studi kasus ini akan dicontohkan bagaimana cara membuat form registrasi yang di dalamnya terdapat beberapa jenis komponen input seperti textbox, radiobutton, combo box, checkbox dan textarea.

Form tersebut akan diterapkan dalam kasus registrasi peserta suatu seminar secara online.

Untuk keperluan tersebut terlebih dahulu kita buat tabelnya.

Nama Tabel : peserta  
Field



id	integer	auto increment	primary key
namaPeserta	varchar(30)		
tglLahir	date		
tmptLahir	varchar(30)		
sex	varchar(1)		
alamat	text		
tglDaftar	date		
informasi	text		

Keterangan:

Field 'informasi' digunakan untuk menyimpan keterangan darimana peserta seminar mendapatkan informasi tentang seminar (dari surat kabar, televisi, atau dari teman).

Field 'id' diset bertipe data integer dan berupa auto increment. Maksud dari auto increment adalah secara otomatis akan muncul nomor urut begitu data baru dimasukkan ke dalam tabel.

Field dengan tipe data DATE, harus berformat YYYY-MM-DD (tahun-bulan-tgl)

Selanjutnya kita siapkan form untuk input data peserta seminarnya

### registrasi.php

```
<html>
<head>
  <title>Registrasi Peserta Seminar</title>
</head>
<body>
  <h1>Registrasi Peserta Seminar</h1>
  <form method="post" action="submit.php">
    <table>
      <tr><td>Nama    Lengkap</td><td>:</td><td><input type="text" name="nama"
/></td></tr>
      <tr><td>Alamat</td><td>:</td><td><textarea
name="alamat"></textarea></td></tr>
      <tr><td>Tempat Lahir</td><td>:</td><td><input type="text" name="tmptLhr"
/></td></tr>
      <tr><td>Tanggal Lahir</td><td>:</td>
      <td>
        <select name="tgl">
          <?php
            for($i=1; $i<=31; $i++)
            {
```

```

        echo "<option value='".$.i.">".$.i."</option>";
    }
    ?>
</select>
<select name="bln">
<?php
for($i=1; $i<=12; $i++)
{
    echo "<option value='".$.i.">".$.i."</option>";
}
?>
</select>
<select name="thn">
<?php
for($i=1950; $i<=1990; $i++)
{
    echo "<option value='".$.i.">".$.i."</option>";
}
?>
</select>
</td>
</tr>
<tr><td>Jenis Kelamin</td><td>:</td><td><input type="radio" name="sex"
value="L" /> Laki-laki <br/><input type="radio" name="sex" value="P" />
Perempuan</td></tr>
<tr><td>Tahu informasi seminar dari mana?</td><td>:</td><td><input
type="checkbox" name="info1" value="TV" /> Televisi<br /><input type="checkbox"
name="info2" value="Koran" /> Surat Kabar<br /><input type="checkbox"
name="info3" value="Teman" /> Teman</td></tr>
<tr><td></td><td></td><td><input type="submit" name="submit"
value="Submit" /></td></tr>
</table>
</form>
</body>
</html>

```

Keterangan:

Pusing melihat script diatas? ☺ Jangan khawatir, script nantinya akan disertakan melalui email untuk dipelajari.

Form di atas akan tampak seperti di bawah ini begitu dibuka di browser

## Registrasi Peserta Seminar

Nama Lengkap :

Alamat :

Tempat Lahir :

Tanggal Lahir :

Jenis Kelamin : ☐ Laki-laki  
☐ Perempuan

Tahu informasi seminar dari mana? : ☐ Televisi  
☐ Surat Kabar  
☐ Teman

Setelah kita buat form registrasinya, selanjutnya kita buat form untuk proses penyimpanan data pesertanya.

### submit.php

```
<?php
```

```
include "koneksi.php";
```

```
$nama = $_POST['nama'];
```

```
$alamat = $_POST['alamat'];
```

```
$tmptLhr = $_POST['tmptLhr'];
```

```
$tglLhr = $_POST['thn']. "-" . $_POST['bln']. "-" . $_POST['tgl']; // membentuk format thn-  
bln-tgl
```

```
$sex = $_POST['sex'];
```

```
$tglDaftar = date("Y-m-d"); // membaca tanggal sekarang (recent date) format thn-  
bln-tgl
```

```
// menggabungkan informasi darimana peserta mendapatkan info tentang seminar
```

```
$info = $_POST['info1']. " " . $_POST['info2']. " " . $_POST['info3'];
```

```
$query = "INSERT INTO peserta (namaPeserta, tglLahir, tmptLahir, sex, alamat, tglDaftar,  
informasi) VALUES ('$nama', '$tglLhr', '$tmptLhr', '$sex', '$alamat', 'tglDaftar',  
'$info' )";
```

```
$hasilQuery = mysql_query($query);
```

```
if ($hasilQuery) echo "Data sudah tersimpan ke database";  
else echo "Penyimpanan data gagal";
```

```
?>
```

## Tugas Untuk Anda

1. Buatlah form untuk keperluan pengisian buku tamu (guestbook) dengan data yang diisikan sbb:

Nama pengisi  
Email  
Situs Web  
Komentar

Buat pula sebuah tabel dalam database untuk menyimpan data isian buku tamu tersebut dengan field sbb:

Id	integer	auto increment	Primary Key
Nama	varchar(30)		
Email	varchar(30)		
Website	varchar(30)		
Tanggal	date		
Komentar	text		

Keterangan:

Field 'tanggal' digunakan untuk menyimpan informasi tanggal kapan komentar tersebut dikirim (tanggal ini secara otomatis membaca tanggal ketika komentar buku tamu disubmit)

2. Buatlah form untuk registrasi mahasiswa baru angkatan tahun 2009 di sebuah universitas X yang di dalamnya terdapat 3 buah fakultas, yaitu KEDOKTERAN (kode: A), TEKNIK (kode: B), dan MIPA (kode: C)

Berikut ini gambaran form untuk registrasi mahasiswa adalah:

## Form Registrasi Mahasiswa

Nama Mahasiswa :

Alamat Mahasiswa :

Pilih Fakultas :

Tempat Lahir :

Tanggal Lahir :

No. Telp :

Setiap kali data mahasiswa baru diinputkan, maka secara otomatis dibuatkan nomor induk mahasiswa (NIM) dengan format X-XX-XXXX sebelum disimpan ke database dengan keterangan:

Digit pertama menunjukkan kode fakultas si mahasiswa, Digit kedua dan ketiga menunjukkan tahun angkatan, dan 4 digit sisanya menunjukkan nomor urut mahasiswa sesuai kode fakultas dan tahun angkatannya

Berikut ini contoh cara pemberian NIM, misalkan terdapat 6 data mahasiswa yang akan diinputkan

Mhs 1 (KEDOKTERAN )	Nomor induk: A-09-0001
Mhs 2 (TEKNIK)	Nomor induk: B-09-0001
Mhs 3 (KEDOKTERAN)	Nomor induk: A-09-0002
Mhs 4 (TEKNIK)	Nomor induk: B-09-0002
Mhs 5 (MIPA)	Nomor induk: C-09-0001
Mhs 6 (KEDOKTERAN)	Nomor induk: A-09-0003

Sedangkan field tabel database yang diperlukan untuk registrasi ini adalah

nim	varchar(9)	primary key
namaMhs	varchar(30)	
alamatMhs	text	
tmptLahir	varchar(30)	
tglLahir	date	
noTelp	varchar(30)	

**Petunjuk Pengerjaan:**

Silakan baca artikel saya di blog tentang membuat kode unik incremental otomatis terkait dengan membuat NIM untuk kasus di atas.

<http://blog.rosihanari.net/ide-membuat-generator-kode-unik-incremental-otomatis/>

## BAB 18

# Konsep Menampilkan Data MySQL dengan PHP

---

Pada bab sebelumnya Anda telah belajar bagaimana konsep menyisipkan data atau record ke dalam tabel database MySQL. Sekarang kita lanjutkan ke konsep bagaimana menampilkan data yang ada di database MySQL dengan script PHP.

Konsep menampilkan data dengan script PHP ini adalah menggabungkan query SELECT di SQL yang dijalankan dalam script PHP.

Konsep dasar menampilkan data dengan script PHP ini adalah

1. Buat query SQL untuk menampilkan data (gunakan SELECT)
2. Jalankan query tersebut dengan menggunakan `mysql_query()` di PHP
3. Baca data hasil query tersebut dengan proses *fetching*. Gunakan perintah `mysql_fetch_array()` di PHP. Hasil dari `mysql_fetch_array()` ini adalah suatu nilai dalam bentuk array asosiatif dengan nama field tabel sebagai indeksinya.
4. Tampilkan data dari proses fetching sesuai nama field yang ada dalam array.

Langsung saja, dalam pembahasan bab ini akan saya berikan contohnya untuk menampilkan data semua peserta yang telah registrasi sesuai kasus yang diberikan pada Bab 17. Sehingga dalam hal ini, struktur tabelnya sama dengan kasus yang dibahas pada Bab 17.

Dalam contoh ini, kita akan mencoba menampilkan data peserta khususnya pada field Nama Peserta, Tanggal Lahir, Jenis Kelamin, Alamat, Tanggal Daftar, dan Tempat Lahir.

Langkah pertama, kita siapkan terlebih dahulu script untuk koneksi ke MySQL nya dan ke databasenya.

### **koneksi.php**

```
<?php
mysql_connect("localhost", "root", "root");
mysql_select_db("contoh");
?>
```

Catatan: sesuaikan nama username, password dan nama database yang Anda gunakan.

Langkah kedua, kita bisa mulai membuat script untuk menampilkan datanya.

#### view.php

```
<html>
<head>
  <title>Data Peserta</title>
</head>
<body>
  <h1>Data Peserta</h1>
  <table border="1">
    <tr><th>Nama                Peserta</th><th>Tgl                Lahir</th><th>Tempat
Lahir</th><th>Alamat</th>
    <th>Jenis Kelamin</th><th>Tgl Daftar</th></tr>

    // letak script menampilkan data //

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

Script di atas baru membuat judul tabel untuk menampilkan datanya. Sekarang barulah kita buat script untuk menampilkan datanya.

#### view.php

```
<html>
<head>
  <title>Data Peserta</title>
</head>
<body>
  <h1>Data Peserta</h1>
  <table border="1">
    <tr><th>Nama                Peserta</th><th>Tgl                Lahir</th><th>Tempat
Lahir</th><th>Alamat</th>
    <th>Jenis Kelamin</th><th>Tgl Daftar</th></tr>

    <?php

    include "koneksi.php";

    $query = "SELECT * FROM peserta";
    $hasil = mysql_query($query);
```



```
while ($data = mysql_fetch_array($hasil))
{
    echo "<tr><td>".$data['namaPeserta']. "</td><td>".$data['tglLahir']. "</td>
        <td>".$data['tmptLahir']. "</td><td>".$data['alamat']. "</td>
        <td>".$data['sex']. "</td><td>".$data['tglDaftar']. "</td></tr>";
}
?>

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

**Keterangan:**

Dalam melakukan proses fetching data menggunakan `mysql_fetch_array()` dilakukan per baris data dari hasil query. Sehingga untuk memfetching semua data hasil query, kita gunakan looping WHILE.

Selama masih ada baris data hasil query yang bisa difetch, maka looping ini akan terus berjalan dan baru akan berhenti bila sudah tidak ada baris lagi yang bisa difetch (akhir dari baris data).

Setiap kali data difetch, selanjutnya langsung ditampilkan dengan echo.

**Catatan:**

Bila data hasil query SELECT hanya terdiri dari sebuah baris data, maka Anda tidak perlu menggunakan looping WHILE.

Array `$data['namaPeserta']` digunakan untuk menampilkan data terkait dengan field `'namaPeserta'` yang ada dalam tabel.

**PENTING!!**

Besar kecilnya huruf dalam indeks array harus sama dengan nama fieldnya. Jika tulisannya berbeda, maka data tidak akan muncul.

Hasil dari script `view.php` di atas adalah sebagai berikut



## Tugas Untuk Anda

1. Dengan menggunakan tabel dan data yang ada dalam pembahasan modul ini di atas, buatlah script untuk menampilkan data peserta berdasarkan jenis kelaminnya.

Adapun tampilan datanya diharapkan seperti di bawah ini

No	Nama Peserta	Tgl Lahir	Tmpt Lahir	Alamat	Jenis Kelamin
1					
2					

Jumlah Data : ...

Dengan 'No' adalah nomor urut 1, 2, 3, ... (bukan nomor ID data dalam tabel database)

Terlebih dahulu buatlah form berisi pilihan menggunakan Radio Button untuk memilih jenis kelamin dari peserta yang akan ditampilkan. Selanjutnya setelah dipilih jenis kelaminnya, dan kemudian disubmit, maka akan muncul daftar peserta berdasarkan jenis kelamin yang dipilih.

2. Dengan menggunakan tabel dan data yang ada dalam pembahasan modul ini di atas, buatlah script untuk menampilkan data peserta berdasarkan nama yang ingin dicari.

Terlebih dahulu buatlah form berisi text box untuk memasukkan nama peserta yang akan dicari. Selanjutnya setelah dimasukkan nama, dan kemudian disubmit, maka akan muncul daftar peserta berdasarkan nama yang dicari. Misal jika ingin dicari nama 'JOKO' maka akan menampilkan semua data peserta yang namanya memuat nama 'JOKO'.

Bantuan:

Gunakan klausa WHERE nama LIKE '%...%';

Tampilkan data hasil pencariannya menggunakan format tampilan seperti no. 1 di atas

3. Dengan menggunakan tabel dan data yang ada dalam pembahasan modul ini di atas, buatlah script untuk menampilkan semua data peserta yang diurutkan berdasarkan TANGGAL PENDAFTARAN mulai dari yang baru saja mendaftar sampai dengan yang paling dahulu mendaftar

Adapun tampilan datanya diharapkan seperti di bawah ini

No	Nama Peserta	Tgl Lahir	Alamat	Jenis Kelamin	Tgl Daftar
1					
2					

Jumlah Data Semua : ...

Jumlah Peserta Laki-Laki : ...

Jumlah Peserta Perempuan : ...

Catatan:

Untuk mempercantik tampilan data dalam tabel di atas, gunakan warna selang-seling dalam setiap barisnya.

Tutorial tentang bagaimana cara membuat warna baris selang-seling, dapat dibaca di

<http://blog.rosihanari.net/script-php-untuk-menampilkan-data-dalam-tabel-dengan-warna-baris-selang-seling/>

## BAB 19

# Update Data dengan PHP - MySQL

---

Proses update suatu data dengan menggunakan script PHP dan MySQL biasanya memiliki mekanisme sebagai berikut:

1. Tampilkan list data terlebih dahulu
2. Buat sebuah link pada setiap baris data yang menuju proses update. Link tersebut memuat sebuah parameter yang di dalamnya terdapat value yang nantinya digunakan sebagai acuan dalam proses update datanya.
3. Selanjutnya bila link tersebut diklik akan muncul form berisi data yang akan diedit.

Dalam melakukan proses update data, pada prinsipnya adalah menjalankan query update pada SQL nya.

UPDATE namaTabel SET namafield = value WHERE ...

Baiklah... kita langsung saja mempraktekkan proses update data. Sebagai contoh, kita akan mengupdate data mahasiswa, dimana tabel untuk menyimpan data mahasiswanya berbentuk seperti di bawah ini

nim	varchar(5)	primary key
nama	varchar(30)	
alamat	text	
sex	varchar(1)	
tgllahir	date	

Sebagai contoh misalkan kita telah mempunyai data awal sebagai berikut

NIM	NAMA	ALAMAT	SEX	TGLLAHIR
M0001	A	SOLO	L	1979-10-20
M0002	B	JAKARTA	L	1982-09-10
M0003	C	BANDUNG	P	1980-11-07

Catatan: Untuk membuat data awal Anda bisa mengisikannya langsung ke MySQL dengan phpMyAdmin atau membuat script PHP untuk memasukkan datanya.

Sesuai dengan mekanisme di atas, maka langkah pertama kita buat terlebih dahulu script untuk menampilkan list datanya. Dalam script list ini, data akan disajikan dalam bentuk tabel dengan 6 buah kolom. Lima kolom pertama untuk menampilkan data semua fieldnya, sedangkan kolom ke-6 untuk meletakkan link untuk editnya.

### list.php

```
<?php

mysql_connect("localhost", "root", "root");
mysql_select_db("data");

// query SQL untuk memunculkan semua mahasiswa

$query = "SELECT * FROM mhs";
$hasil = mysql_query($query);

echo "<table border='1'>";
echo "<tr><td>NIM</td>
      <td>Nama Mahasiswa</td>
      <td>Alamat</td>
      <td>Sex</td>
      <td>Tgl Lahir</td>
      <td>Action</td></tr>";
while ($data = mysql_fetch_array($hasil))
{
    echo "<tr><td>".$data['nim']. "</td>
          <td>".$data['nama']. "</td>
          <td>".$data['alamat']. "</td>
          <td>".$data['sex']. "</td>
          <td>".$data['tgllahir']. "</td>
          <td>Edit</td></tr>";
}
echo "</table>";
?>
```

Hasil dari script di atas akan menghasilkan tampilan sebagai berikut:

NIM	Nama Mahasiswa	Alamat	Sex	Tgl Lahir	Action
M0001	A	SOLO	L	1979-10-20	Edit
M0002	B	JAKARTA	L	1982-09-10	Edit
M0003	C	BANDUNG	P	1980-11-07	Edit

Selanjutnya, bagaimana bentuk link editnya? Link edit misalkan kita buat sebagai berikut

<http://.../edit.php?nim=...>

Dalam link edit di atas, link akan diarahkan ke script edit.php yang nantinya akan memunculkan form untuk keperluan edit data. Di dalam link tersebut terdapat parameter nim yang nantinya digunakan sebagai acuan mahasiswa manakah yang datanya akan diedit. Sebagai contoh, misalkan akan diedit mahasiswa bernim 'M0001' maka linknya akan berbentuk

<http://.../edit.php?nim=M0001>

OK.. sekarang kita modifikasi sedikit script list.php di atas untuk memunculkan link editnya.

### list.php

```
<?php

mysql_connect("localhost", "root", "root");
mysql_select_db("data");

// query SQL untuk memunculkan semua mahasiswa

$query = "SELECT * FROM mhs";
$hasil = mysql_query($query);

echo "<table border='1'>";
echo "<tr><td>NIM</td>";
    <td>Nama Mahasiswa</td>
    <td>Alamat</td>
    <td>Sex</td>
    <td>Tgl Lahir</td>
    <td>Action</td></tr>";
while ($data = mysql_fetch_array($hasil))
{
    echo "<tr><td>".$data['nim'].</td>
```

```

        <td>".$data['nama']. "</td>
        <td>".$data['alamat']. "</td>
        <td>".$data['sex']. "</td>
        <td>".$data['tgllahir']. "</td>
        <td><a href='edit.php?nim=".$data['nim']."'>Edit</a></td></tr>";
    }
    echo "</table>";
    ?>

```

Langkah selanjutnya kita buat script edit.php yang nantinya akan memunculkan form edit sekaligus data mahasiswa yang akan diedit. Mungkin yang menjadi pertanyaan adalah bagaimana cara memunculkan data mahasiswa yang akan diedit? Ya... jawabnya adalah dengan menggunakan query SELECT dengan nim sebagai syaratnya. Dalam hal ini nim yang ada dalam link edit lah yang nantinya akan digunakan sebagai syarat.

```
SELECT * FROM mhs WHERE nim = ...;
```

OK... berikut ini adalah script edit.php untuk memunculkan form dan data yang akan diedit.

### edit.php

```

<?php

mysql_connect("localhost", "root", "root");
mysql_select_db("data");

$nim = $_GET['nim'];

$query = "SELECT * FROM mhs WHERE nim = '$nim'";
$hasil = mysql_query($query);
$data = mysql_fetch_array($hasil);

echo "<form method='post' action='update.php'>";
echo "<table>";
echo "<tr><td>NIM</td><td>:</td>
        <td><input type='text' name='nim' value='".$data['nim']."'></td></tr>";
echo "<tr><td>Nama Mahasiswa</td><td>:</td>
        <td><input type='text' name='nama' value='".$data['nama']."'></td></tr>";
echo "<tr><td>Alamat</td><td>:</td>
        <td><textarea name='alamat'>".$data['alamat']. "</textarea></td></tr>";
echo "<tr><td>Sex</td><td>:</td>
        <td><input type='radio' name='sex' value='L'> Laki-Laki

```

```
        <input type='radio' name='sex' value='P'> Perempuan</td></tr>";
echo "<tr><td>Tgl Lahir</td><td>:</td><td>
        <select name='tgl'>";
for ($i=1; $i<=31; $i++)
{
    echo "<option value='".$i.">".$i."</option>";
}
echo "</select>
        <select name='bln'>";
for ($i=1; $i<=12; $i++)
{
    echo "<option value='".$i.">".$i."</option>";
}
echo "</select>
        <select name='thn'>";
for ($i=1980; $i<=1997; $i++)
{
    echo "<option value='".$i.">".$i."</option>";
}
echo "</select>
        </td></tr>";
echo "</table>";
echo "<input type='submit' name='submit' value='Submit'>";
echo "</form>";

?>
```

Catatan:

- Untuk keperluan edit data nim, script di atas menggunakan komponen textbox. Demikian pula edit data nama mahasiswa.
- Sedangkan untuk alamat, script di atas menggunakan komponen textarea.
- Untuk edit jenis kelamin, kita menggunakan komponen radiobutton
- Untuk edit tanggal lahir mahasiswa, kita menggunakan 3 komponen combobox yang masing-masing digunakan untuk tanggal, bulan dan tahun. Option dari masing-masing combobox digenerate otomatis dengan looping FOR.

Hasil tampilan dari script di atas ketika digunakan untuk mengedit data mahasiswa bernim 'M0001' adalah



NIM	:	<input type="text" value="M0001"/>
Nama Mahasiswa	:	<input type="text" value="A"/>
Alamat	:	<input type="text" value="SOLO"/>
Sex	:	<input type="radio"/> Laki-Laki <input type="radio"/> Perempuan
Tgl Lahir	:	<input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1980"/>
<input type="button" value="Submit"/>		

Bila kita perhatikan tampilan script edit di atas, maka untuk edit jenis kelamin masih belum berfungsi. Seharusnya karena data awal mahasiswa bernim 'M0001' adalah berjenis kelamin 'Laki-laki' maka radiobutton Laki-laki terpilih (checked). Demikian pula bila jenis kelaminnya adalah Perempuan. Lantas.. bagaimana caranya mewujudkan hal ini? OK.. caranya adalah dengan menggunakan konsep IF.

Perhatikan bagian script dari edit.php di bawah ini:

```
echo "<tr><td>Sex</td><td>:</td>
      <td><input type='radio' name='sex' value='L'> Laki-Laki
      <input type='radio' name='sex' value='P'> Perempuan</td></tr>";
```

Nah... kita akan memodifikasi bagian script di atas menjadi seperti di bawah ini:

```
echo "<tr><td>Sex</td><td>:</td><td>";

if ($data['sex'] == "L")
    echo "<input type='radio' name='sex' value='L' checked> Laki-Laki
        <input type='radio' name='sex' value='P'> Perempuan";
else if ($data['sex'] == "P")
    echo "<input type='radio' name='sex' value='L'> Laki-Laki
        <input type='radio' name='sex' value='P' checked> Perempuan";

echo "</td></tr>";
```

Konsep dari ide script modifikasi di atas adalah jika data jenis kelamin mahasiswa adalah laki-laki (L) maka komponen radiobutton laki-laki akan dichecked. Demikian pula jika perempuan.

Setelah bagian script edit jenis kelamin diedit maka hasilnya adalah

NIM	:	<input type="text" value="M0001"/>
Nama Mahasiswa	:	<input type="text" value="A"/>
Alamat	:	<input type="text" value="SOLO"/>
Sex	:	<input checked="" type="radio"/> L Laki-Laki <input type="radio"/> Perempuan
Tgl Lahir	:	<input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1980"/>
<input type="button" value="Submit"/>		

Simpel bukan idenya? ☺

OK... next... kita perhatikan bagian edit tanggal lahir. Dalam tampilan script yang telah kita buat di atas, seharusnya tanggal lahir menyesuaikan data tanggal lahir awal sebelum diedit.

Untuk mewujudkan hal ini, pada dasarnya konsep atau idenya sama dengan jenis kelamin di atas yaitu menggunakan IF. Namun... yang menjadi masalah bagaimana caranya memecah tanggal supaya ketiga komponen combobox yang ada dapat membaca tanggal, bulan dan tahun.

OK solusi untuk memecah tanggal, pernah kita bahas sebelumnya yaitu dengan memanfaatkan function explode().

Berikut ini script edit.php yang telah dimodifikasi untuk edit tanggal lahirnya

#### **edit.php**

```
<?php
```

```
mysql_connect("localhost", "root", "root");  
mysql_select_db("data");
```

```
$nim = $_GET['nim'];
```

```
$query = "SELECT * FROM mhs WHERE nim = '$nim'";  
$hasil = mysql_query($query);  
$data = mysql_fetch_array($hasil);
```

```
// memecah tanggal lahir ke dalam masing-masing unsur (tgl, bln, thn)
```

```
$pecahTgl = explode("-", $data['tgllahir']);  
$tgl = $pecahTgl[2];
```

```
$bln = $pecahTgl[1];
$thn = $pecahTgl[0];

echo "<form method='post' action='update.php'>";
echo "<table>";
echo "<tr><td>NIM</td><td>:</td>";
    <td><input type='text' name='nim' value='".$data['nim']."'></td></tr>";
echo "<tr><td>Nama Mahasiswa</td><td>:</td>";
    <td><input type='text' name='nama' value='".$data['nama']."'></td></tr>";
echo "<tr><td>Alamat</td><td>:</td>";
    <td><textarea name='alamat'>".$data['alamat']."</textarea></td></tr>";
echo "<tr><td>Sex</td><td>:</td>";
    <td><input type='radio' name='sex' value='L'> Laki-Laki
        <input type='radio' name='sex' value='P'> Perempuan</td></tr>";
echo "<tr><td>Tgl Lahir</td><td>:</td><td>";
    <select name='tgl'>";
for ($i=1; $i<=31; $i++)
{
    if ($i == $tgl) echo "<option value='".$i."' selected>".$i."</option>";
    else echo "<option value='".$i."'>".$i."</option>";
}
echo "</select>";
    <select name='bln'>";
for ($i=1; $i<=12; $i++)
{
    if ($i == $bln) echo "<option value='".$i."' selected>".$i."</option>";
    else echo "<option value='".$i."'>".$i."</option>";
}
echo "</select>";
    <select name='thn'>";
for ($i=1980; $i<=1997; $i++)
{
    if ($i == $thn) echo "<option value='".$i."' selected>".$i."</option>";
    else echo "<option value='".$i."'>".$i."</option>";
}
echo "</select>";
    </td></tr>";
echo "</table>";
echo "<input type='submit' name='submit' value='Submit'>";
echo "</form>";

?>
```

Nah... mudah dan simpel bukan idenya?

Sekarang yang terakhir, kita buat script untuk updatenya. Tapi... eiiiit... wait a minute.. bagaimana bila NIM mahasiswanya yang diubah? Lho.. emangnya ada apa? Mungkin di pikiran Anda konsepnya sama dengan update data yang lain. Iya sih... but... coba kita lihat pemisalannya. Jika data seorang mahasiswa sebelum diubah memiliki NIM 'M0001'. Kemudian data NIM nya diubah menjadi 'M0002'. Bagaimana SQL nya? SQL nya adalah sebagai berikut:

```
UPDATE mhs SET nim = 'M0002' WHERE nim = 'M0001';
```

Dari SQL di atas, kita bisa melihat bahwa kita perlu NIM lama mahasiswa guna proses update tersebut. Nah... bila komponen textbox pada form edit untuk update NIM di atas kita ubah isinya, maka dengan kata lain kita akan kehilangan NIM yang lama. Nah... gimana tuh?

OK... ide untuk menanggulangi hal tersebut adalah kita buat komponen hidden input untuk menyimpan NIM yang lama (sebelum diedit). NIM lama ini nantinya kita gunakan sebagai acuan proses updatenya.

Nah.. dengan demikian kita modifikasi kembali script edit.php yang telah kita buat sebelumnya.

#### **edit.php**

```
<?php

mysql_connect("localhost", "root", "root");
mysql_select_db("data");

$nim = $_GET['nim'];

$query = "SELECT * FROM mhs WHERE nim = '$nim'";
$hasil = mysql_query($query);
$data = mysql_fetch_array($hasil);

$pecahTgl = explode("-", $data['tgllahir']);
$tgl = $pecahTgl[2];
$bln = $pecahTgl[1];
$thn = $pecahTgl[0];

echo "<form method='post' action='update.php'>";
echo "<table>";
echo "<tr><td>NIM</td><td></td>";
    <td><input type='text' name='nimBaru' value='". $data['nim'] ."'></td></tr>";
```

```
echo "<tr><td>Nama Mahasiswa</td><td>:</td>
      <td><input type='text' name='nama' value='".\$data['nama']."'></td></tr>";
echo "<tr><td>Alamat</td><td>:</td>
      <td><textarea name='alamat'>".\$data['alamat']."'</textarea></td></tr>";
echo "<tr><td>Sex</td><td>:</td>
      <td><input type='radio' name='sex' value='L'> Laki-Laki
      <input type='radio' name='sex' value='P'> Perempuan</td></tr>";
echo "<tr><td>Tgl Lahir</td><td>:</td><td>
      <select name='tgl'>";
for ($i=1; $i<=31; $i++)
{
    if ($i == $tgl) echo "<option value='".\$i."' selected>".\$i."</option>";
    else echo "<option value='".\$i."'>".\$i."</option>";
}
echo "</select>
      <select name='bln'>";
for ($i=1; $i<=12; $i++)
{
    if ($i == $bln) echo "<option value='".\$i."' selected>".\$i."</option>";
    else echo "<option value='".\$i."'>".\$i."</option>";
}
echo "</select>
      <select name='thn'>";
for ($i=1980; $i<=1997; $i++)
{
    if ($i == $thn) echo "<option value='".\$i."' selected>".\$i."</option>";
    else echo "<option value='".\$i."'>".\$i."</option>";
}
echo "</select>
      </td></tr>";
echo "</table>";
echo "<input type='hidden' name='nimLama' value='".\$data['nim']."'>";
echo "<input type='submit' name='submit' value='Submit'>";
echo "</form>";

?>
```

Catatan:

Letak komponen hidden input ini bisa diletakkan di sebarang tempat di dalam form, mengingat komponen ini tidak terlihat oleh mata ☺

Nah.. dengan demikian kita sekarang bisa konsentrasi ke script updatenya. Konsepnya hanyalah menjalankan query SQL UPDATE.

**update.php**

```
<?php

mysql_connect("localhost", "root", "root");
mysql_select_db("data");

$nimLama = $_POST['nimLama'];
$nimBaru = $_POST['nimBaru'];
$nama = $_POST['nama'];
$alamat = $_POST['alamat'];
$sex = $_POST['sex'];

// menggabung unsur tgl dalam combobox menjadi satu dalam format yyyy-mm-dd

$tgllahir = $_POST['thn']. "-". $_POST['bln']. "-". $_POST['tgl'];

$query = "UPDATE mhs
        SET nim = '$nimBaru', nama = '$nama', alamat = '$alamat', sex = '$sex',
            tgllahir = '$tgllahir'
        WHERE nim = '$nimLama'";
$hasil = mysql_query($query);
if ($hasil) echo "Data sudah diupdate";
else echo "Data gagal diupdate";
?>
```

OK deh selesai proses pembuatan script updatenya.

## Tugas Untuk Anda

Misalkan Anda memiliki data dan tabel sebagai berikut:

Tabel: Jenis Buku

kodeJenisBuku	namaJenisBuku
M	Majalah
R	Referensi
N	Koran
J	Journal

Tabel : Buku

kodeBuku	judulBuku	Pengarang	jenisBuku
B001	A	Mr. X	M
B002	B	Mr. Y	R
B003	C	Mr. X	R
B004	D	Mr. A	J

Tabel : Member

kodeMember	namaMember
M01	XXX
M02	YYY
M03	ZZZ

Tabel : Pinjam Buku

kodePinjam	kodeBuku	kodeMember	tglPinjam	tglHrsKembali
1	B001	M01	2009-06-10	2009-06-17
2	B002	M02	2009-06-10	2009-06-17
3	B001	M02	2009-06-18	2009-06-25
4	B003	M03	2009-06-20	2009-06-27
5	B004	M02	2009-06-21	2009-06-28

## Ketentuan Tugas:

1. Buatlah tabel-tabel di atas di MySQL dengan menentukan sendiri tipe data masing-masing fieldnya. Tentukan pula manakah field yang merupakan primary key untuk setiap tabelnya. Untuk tabel 'Pinjam Buku', field **kodePinjam** menggunakan auto increment.
2. Buatlah fasilitas untuk input data untuk masing-masing tabel. Khusus input data untuk tabel buku, jenis buku harus dipilih user melalui sebuah combo box. Isi list combobox tersebut harus membaca data dari tabel 'Jenis Buku'. Demikian pula untuk input data peminjaman buku. Isian untuk kode buku harus menggunakan combo box yang membaca data dari tabel buku, dan isian kode member harus menggunakan combo box yang membaca data dari tabel member.
3. Pada input data peminjaman, setelah mengisi tanggal peminjaman secara otomatis tanggal harus kembali akan disimpan dalam tabel peminjaman. Selisih tanggal pinjam dengan tanggal harus kembali adalah 7 hari.

4. Buatlah fasilitas untuk update data untuk keempat tabel di atas. Bentuk form editnya sama dengan bentuk form untuk input data.



## BAB 20

# Hapus Data dengan PHP - MySQL

---

Pada modul terakhir ini, kita akan membahas bagaimana cara melakukan hapus data pada database MySQL dengan PHP.

Konsep utama dari proses penghapusan data MySQL dengan PHP adalah menjalankan query SQL DELETE FROM ... pada script PHP tersebut.

Untuk lebih jelasnya konsep tersebut, kita langsung terapkan pada studi kasus.

Sebagai studi kasus, kita ambil sampel seperti halnya pada Bab 19, yaitu tentang data mahasiswa.

Misalkan kita memiliki tabel mahasiswa sebagai berikut:

nim	varchar(5)	primary key
nama	varchar(30)	
alamat	text	
sex	varchar(1)	
tgllahir	date	

dan andaikan kita telah mempunyai data awal sebagai berikut

NIM	NAMA	ALAMAT	SEX	TGLLAHIR
M0001	A	SOLO	L	1979-10-20
M0002	B	JAKARTA	L	1982-09-10
M0003	C	BANDUNG	P	1980-11-07

Selanjutnya kita akan membuat script PHP untuk menghapus data mahasiswa, baik melalui POST maupun GET method.

### Studi Kasus 1

Pada studi kasus pertama ini, kita akan membuat script hapus mahasiswa menggunakan POST method.

Adapun skenarionya adalah sebagai berikut. User diminta memasukkan NIM mahasiswa yang akan dihapus melalui sebuah form. Selanjutnya script akan menghapus data

mahasiswa sesuai NIM yang ingin dihapus tadi. Namun... bila NIM yang akan dihapus tidak ada dalam database, maka akan muncul pesan bahwa data tersebut tidak ada.

Berikut ini adalah tampilan script yang akan kita buat dalam studi kasus 1 ini

## Hapus Data Mahasiswa

Masukkan NIM

NIM	Nama Mhs	Alamat	Sex	Tgl Lahir
M0001	A	SOLO	L	1979-10-20
M0002	B	JAKARTA	L	1982-09-10
M0003	C	BANDUNG	P	1980-11-07

Adapun scriptnya berbentuk seperti di bawah ini:

### hapus01.php

```
<h1>Hapus Data Mahasiswa</h1>
```

```
<!-- Form untuk memasukkan nim mahasiswa yang mau dihapus -->
```

```
<form method="post" action="<?php $_SERVER['PHP_SELF'] ?>?proses=hapus">
Masukkan NIM <input type="text" name="nim">
<input type="submit" name="submit" value="Submit">
</form>
```

```
<?php
```

```
// koneksi ke mysql
mysql_connect("localhost", "...", "...");
mysql_select_db("...");
```

```
if ($_GET['proses'] == "hapus")
{
```

```
    // bagian ini akan dijalankan ketika proses SUBMIT dari form
```

```
    $nim = $_POST['nim']; // membaca NIM yang dimasukkan dalam form
```

```
    // mengecek keberadaan data mahasiswa dengan NIM sesuai yang
    // dimasukkan dalam form
```

```
    $query = "SELECT count(*) as jum FROM mahasiswa WHERE nim = '$nim'";
```

```
$hasil = mysql_query($query);
$data = mysql_fetch_array($hasil);

// jika jumlah mahasiswa ber NIM tersebut tidak ada, maka muncul keterangan

if ($data['jum'] == 0) echo "<p>Data yang akan dihapus tidak ada</p>";
else
{
    // jika ada, maka lakukan penghapusan
    $query = "DELETE FROM mahasiswa WHERE nim = '$nim'";
    $hasil = mysql_query($query);
}
}

// menampilkan data semua mahasiswa

$query = "SELECT * FROM mahasiswa";
$hasil = mysql_query($query);

echo "<table border='1'>";
echo "<tr><th>NIM</th><th>Nama Mhs</th><th>Alamat</th><th>Sex</th><th>Tgl
Lahir</th></tr>";
while ($data = mysql_fetch_array($hasil))
{
    echo "<tr><td>". $data['nim']. "</td><td>". $data['nama']. "</td>
        <td>". $data['alamat']. "</td><td>". $data['sex']. "</td>
        <td>". $data['tglahir']. "</td></tr>";
}
echo "</table>";
?>
```

## Studi Kasus 2

Untuk studi kasus yang kedua ini, kita akan lakukan penghapusan data mahasiswa melalui GET method atau melalui link berparameter. Link ini nanti akan diletakkan di setiap baris data mahasiswa.

Bentuk link berparameter yang nanti akan digunakan untuk menghapus data adalah

<http://.../hapus02.php?proses=hapus&nim=xxx>

dengan 'xxx' adalah NIM mahasiswa yang akan dihapusnya.

Berikut ini screen shot tampilan studi kasus 2 (perhatikan link berparameternya yang tampak di status bar)



Untuk script studi kasus 2 ini adalah sebagai berikut:

### hapus02.php

```
<h1>Hapus Data Mahasiswa</h1>
```

```
<?php
```

```
// koneksi ke mysql
```

```
mysql_connect("localhost", "...", "...");
```

```
mysql_select_db("...");
```

```
if ($_GET['proses'] == "hapus")
```

```
{
```

```
    // bagian ini akan dijalankan ketika link hapus diklik (parameter proses=hapus)
```

```
    // membaca NIM dari URL berparameter
```

```
    $nim = $_GET['nim'];
```

```
    // mengecek keberadaan data mahasiswa dengan NIM
```

```
    // sesuai yang dimasukkan dalam form
```

```
    $query = "SELECT count(*) as jum FROM mahasiswa WHERE nim = '$nim'";
```

```
    $hasil = mysql_query($query);
```

```
    $data = mysql_fetch_array($hasil);
```

```
    // jika jumlah mahasiswa ber NIM tersebut tidak ada, maka muncul keterangan
```

```
    if ($data['jum'] == 0) echo "<p>Data yang akan dihapus tidak ada</p>";
```

```
    else
```

```

{
    // jika ada, maka lakukan penghapusan
    $query = "DELETE FROM mahasiswa WHERE nim = '$nim'";
    $hasil = mysql_query($query);
}
}

// menampilkan data semua mahasiswa & link untuk menghapus

$query = "SELECT * FROM mahasiswa";
$hasil = mysql_query($query);

echo "<table border='1'>";
echo "<tr><th>NIM</th><th>Nama Mhs</th><th>Alamat</th><th>Sex</th><th>Tgl
Lahir</th><th>Proses</th></tr>";
while ($data = mysql_fetch_array($hasil))
{
    echo "<tr><td>".$data['nim']. "</td><td>".$data['nama']. "</td>
        <td>".$data['alamat']. "</td><td>".$data['sex']. "</td>
        <td>".$data['tgllahir']. "</td>
        <td><a href='".$_SERVER['PHP_SELF']."'?proses=hapus&nim=".$data['nim']."'>
            Hapus</a></td></tr>";
}
echo "</table>";
?>

```

## Tugas untuk Anda

Menggunakan kasus yang ada Bab 19, buatlah script untuk menghapus data buku, data member dan data peminjaman buku menggunakan GET method. Untuk menghapus data peminjaman buku, gunakan kode peminjaman buku sebagai parameter nya.

Tambahkan pula informasi jumlah data yang ada di masing-masing tabel, di bawah list data yang tersaji dalam bentuk tabel. Contoh tampilan:

NIM	Nama Mhs	Alamat	Sex	Tgl Lahir	Proses
M0001	A	SOLO	L	1979-10-20	<a href="#">Hapus</a>
M0003	C	BANDUNG	P	1980-11-07	<a href="#">Hapus</a>

Jumlah data : 2