

# Pemrograman Web dengan PHP dan MySQL

**Akhmad Sofwan**

*sofwan@sofwan.net*

*http://www.sofwan.net*

## **Lisensi Dokumen:**

*Copyright © 2003-2016 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

Pemrograman adalah sebuah pekerjaan menulis kode program untuk menghasilkan sebuah aplikasi atau software. Salah satu platform aplikasi adalah Web, selain desktop dan Mobile. Pemrograman di dalam platform Web sering disebut Pemrograman Web. Di dalam pemrograman web, terdapat bahasa pemrograman dan script yang dapat dipergunakan. HTML dan CSS adalah script untuk membuat tampilan halaman web. JavaScript adalah bahasa pemrograman yang biasa dipergunakan untuk membuat halaman web lebih interaktif. JavaScript dengan node.js nya juga dipergunakan sebagai Back-end. Sebagai Back-End, PHP, Python dan Java juga dapat dipergunakan. Pada tutorial ini, kita akan membahas penggunaan PHP sebagai bahasa pemrograman web, dengan disinggung juga HTML dan CSS sebagai pembuat halaman webnya. Penulis juga menggunakan MySQL sebagai database untuk pegolahan data.

## **Pendahuluan**

Tutorial Pemrograman web ini, penulis gunakan untuk mengasuh kelas Web Programming di Perguruan Tinggi Raharja, Tangerang. Tutorial ini dibagi menjadi 14 bab dan pada bab 11 ke atas, terdapat sebuah proyek kecil PHP yang juga disertai source code. Source code pada tutorial ini disertai dengan nomor baris, untuk identitasnya.

Salah satu cara untuk menginstall PHP dengan Apache, MySQL dan Phpmyadmin, adalah dengan menggunakan Xampp ([www.apachefriends.org](http://www.apachefriends.org)). PHP versi terkini (Maret 2019) adalah PHP versi 7.3.2. Kita juga perlu menggunakan Text Editor untuk menulis kode PHP. Anda dapat menggunakan geany atau Notepad++ sebagai text editor PHP, disamping sejumlah Text Editor lainnya. Anda dapat juga menggunakan MariaDB yang memiliki kemiripan tinggi dengan MySQL dan pada versi terkini dari XAMPP, yang dulunya menggunakan MySQL, saat ini sudah menggunakan MariaDB. MariaDB sendiri adalah database open source yang dibuat oleh Developer MySQL. MySQL memiliki versi komersial dan community, namun MariaDB hanya memiliki lisensi open source.

# **BAB I**

## **Pengertian Dasar Pemrograman Web dan PHP**

### **I.1. Apa itu Pemrograman Web**

Pemrograman Web adalah pemrograman komputer yang berjalan dengan menggunakan browser. Pemrograman komputer dilakukan dengan penerapan ilmu-ilmu komputer tertentu, seperti Algoritma dan Struktur data yang diterapkan di dalam sebuah bahasa pemrograman tertentu. Untuk Web programming, ada beberapa bahasa pemrograman yang dapat digunakan, seperti : PHP, JSP, ASP dan Ruby. Di dalam kesempatan ini, kita akan menggunakan PHP untuk dalam Pemrograman Web.

### **I.2 Pengenalan PHP**

PHP (PHP:Hypertext Preprocessor) adalah sebuah bahasa pemrograman yang umum digunakan untuk pengembangan web yang dijalankan dalam sebuah browser dan diterjemahkan oleh Web Server. Web Server adalah sebuah perangkat keras atau perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP atau HTTPS atas file-file yang terdapat pada suatu situs web.

PHP di dalam penggunaannya untuk pembuatan web, juga bekerja sama dengan html,css dan javascript serta database dalam proses pembuatannya, yang peran PHP sendiri adalah untuk pemrograman di sisi server, yaitu pemrograman yang dieksekusi scriptnya oleh web server, sementara untuk design, digunakan html,css,java script dan grafis.

### **I.3 Menggunakan PHP**

Untuk menggunakan PHP, kita perlu menginstall Web Server dengan fitur dapat mengakses PHP. Beberapa Web Server yang digunakan oleh PHP, yaitu : Apache, IIS, Xitami dan Nginx. Untuk menggunakan database, kita dapat menggunakan sejumlah database yang dapat bekerja sama dengan PHP, beberapa di antaranya adalah : MySQL, PostgreSQL dan SQL Server. Kita juga dapat menggunakan software bundle instalasi PHP dan lingkungannya, yaitu Xampp, yang di dalamnya langsung terinstall dan terkonfigurasi apache, mysql, proftpd, phpmyadmin, dll.

### **I.4 Pengenalan MySQL**

MySQL adalah sebuah Database Open Source populer yang versi terbaru saat ini untuk versi Mysql Community Server adalah Mysql Community Server 5.1.49. Database ini dapat berjalan di beberapa platform atau Sistem Operasi, seperti Windows, Mac OSX, Linux dan OpenSolaris. Kita dapat mendownload MySQL di <http://dev.mysql.com/downloads>. Fungsi database sendiri adalah untuk menyimpan data.

### **I.5 Software dan script lainnya**

Selain PHP dan Mysql, terdapat beberapa aplikasi dan scripting yang dibutuhkan di dalam pemrograman PHP ini. Kita butuh Web Server dan yang akan kita gunakan adalah Web Server Apache. Web Server sendiri adalah sebuah program komputer untuk menjalankan sebuah konten seperti halaman web, menggunakan Hyper Text Transfer Protocol (HTTP) di dalam World Wide Web atau Internet<sup>1</sup>.

Di dalam pemrograman PHP, kita juga membutuhkan script lain selain PHP itu sendiri, yaitu HTML/XHTML untuk menampilkan halaman web, CSS untuk design dan Java Script untuk pemrograman berbasis client atau browser.

### **I.6 Penggunaan XAMPP**

Untuk menjalankan script PHP dengan Mysql, kita membutuhkan aplikasi PHP dan Mysql serta web server telah terinstall di komputer kita. Kita dapat menginstallnya masing-masing aplikasi tersebut satu persatu secara terpisah, namun ada sebuah paket aplikasi free yang memuat, PHP, Apache, Mysql, PHPMyadmin, dll yang dapat dengan mudah kita install. Aplikasi paket tersebut adalah Xampp, yang dapat di download gratis di <http://www.apachefriends.org>.

## I.7 PHP Editor

Salah satu yang dapat memudahkan kita di dalam pemrograman adalah pemilihan Editor atau Integrated Development Environment (IDE) yang baik dan tepat.

Beda editor dan IDE adalah, bahwa Editor adalah software untuk menulis dan mengedit kode kita, seperti Notepad di Windows dan Vim di Linux, sedangkan IDE adalah editor yang lebih canggih lagi, karena sudah di lengkapi berbagai feature, seperti Debugging, Code Highlighting dan pemunculan symbol class dan feature.

Beberapa Editor atau IDE yang bagus dan free untuk Windows adalah :

1. Notepad++ (<http://notepad-plus-plus.org>)
2. Crimson (<http://www.crimsoneditor.com>)
3. Html-Kit (<http://www.htmlkit.com>)

Beberapa Untuk Linux adalah :

1. gPHPEdit (<http://www.gphpedit.org>)
2. Quanta Plus (<http://quanta.kdewebdev.org>)
3. Kdevelop ([www.kdevelop.org](http://www.kdevelop.org))

Beberapa untuk Linux dan Windows adalah :

1. Bluefish (<http://bluefish.openoffice.nl>)
2. Eclipse ([www.eclipse.org](http://www.eclipse.org))
3. Netbeans IDE ([www.netbeans.org](http://www.netbeans.org))
4. Geany ([www. Geany.org](http://www.Geany.org))

Di samping itu juga terdapat php editor commercial.

## I.8 Web Browser

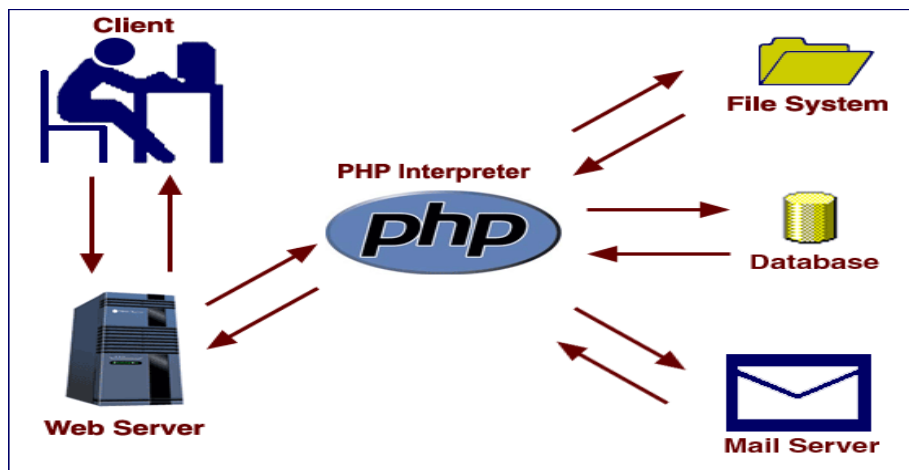
PHP berjalan di atas web server pada sebuah web browser. PHP dapat berjalan di semua web browser modern, seperti Internet Explorer, Mozilla Firefox, Opera, Konqueror dan Chrome. Jika Xampp kita telah berjalan dengan baik, dengan melakukan pengetikan <http://localhost> pada link address kita, maka browser dan xampp telah siap di gunakan.

Selanjut nya, kita cukup menaruh file-file php kita di dalam folder htdocs pada folder xampp. Untuk Windows pada folder [c:/xampp/htdocs](http://c:/xampp/htdocs) dan untuk Linux [/opt/lampp/htdocs](http://opt/lampp/htdocs).

## I.9 How PHP Works ?

Di dalam gambar tsb, client meminta kepada web server via browser sebuah page .php. Page tersebut di interpretasi kan oleh PHP Interpreter. PHP Interpreter, dapat merequest Database, File System dan Mail Server.

Untuk menggunakan PHP, anda harus mengenal HTML dan lebih utama, juga mengenal CSS. PHP di gunakan , sebagai embedded language (bahasa yang disisipkan) di dalam HTML atau kebalikannya, HTML adalah script embedded pada script PHP. Ekstension PHP adalah .php.



Gambar I.1 : How PHP works [5]

## Syntax dasar

Contoh :

### 1. Mencetak kata Hello World

1. <?php
2. echo "Hello World";
3. ?>

### 2. Mencetak kata Hello World di dalam halaman web

1. <!doctype html>
2. <html>
3. <head>
4. <title>PHP Pertama ku</title>
5. </head>
6. <body>
7. <?php echo "Hello World";
8. </body>
9. </html>

### 3. Mengetahui konfigurasi php environment

1. <!doctype html>
2. <html>
3. <head>
4. <title>PHP Info</title>
5. </head>
6. <body>
7. <?php echo phpinfo()
8. </body>
9. </html>

## BAB II

### Dasar- dasar PHP

#### II.1 Pendahuluan

Dasar-dasar PHP yang akan di bahas pada buku ini adalah berupa syntax dasar, tipe data dan algoritma, yang di butuhkan di dalam pemrograman database.

#### II.2 Hello World

Biasa nya, kita di minta menulis “Hello World” untuk awal mula belajar sebuah bahasa pemrograman. Demikian juga untuk kali ini, kita coba membuat kata “Hello World”

1. <?php
2.       echo “Hello World”;
3.       ?>

#### II.3 Tag awal dan penulisan PHP

Setelah kita berhasil menginstall XAMPP dan PHP Editor, maka kita sudah siap untuk mulai menulis Kode PHP kita. Kode PHP diawali dan di tutup dengan <?php ?> atau <? ?> atau <script language=”php”></script>.

Tetapi di sarankan untuk menggunakan tag <?php ?>. Tag <? ?> kurang di sarankan, karena dapat mirip pada tag script lain, sehingga dapat membuat web server kebingungan dalam mengakses script php, jika di gabung dengan script lain yang menggunakan tag sejenis, seperti XML.

Di samping itu penggunaan tag <? ?> mengharuskan short\_open\_tag=On di php.ini.

#### II.4 Penggunaan echo dan print

Untuk menampilkan output, kita dapat menggunakan syntax **echo** dan **Print**.

- Contoh:
1. <?php
  2.       echo “Belajar Pemrograman PHP <br/>”;
  3.       echo (“Belajar Pemrograman PHP <br/>”);
  4.       print “Belajar Pemrograman PHP <br/>”;
  5.       print (“Belajar Pemrograman PHP <br/>”);
  6.       ?>

Kedua fungsi tsb dapat di gunakan dengan atau tanpa “()”. Syntax <br/> adalah syntax xhtml untuk turun ke bawah 1 baris.

#### II.5 Memberi Comment

Dengan : // atau /\*       \*/

Contoh:

- 1.<?php
2.   echo “Nama saya Sofwan <br/>”; // Baris pertama
3.   echo “Tinggal di Jakarta “;
4.   /\* Syntax di atas menggunakan
5.       echo dua-dua nya.
6.   \*/
7. ?>

## II.6 Insert Syntax PHP di dalam HTML

Seringkali di perlukan untuk insert code php di dalam html atau sebalik nya.

Contoh Insert PHP di dalam HTML:

```
1.<html>
2. <head>
3.     <title>Contoh Program PHP </title>
4. </head>
5. <body>
6.     <p>Nama <?php echo "Andi" ;?> </p>
7. </body>
8. </html>
```

Contoh insert html di dalam php.

```
1.<?php
2. // Membuat kata tebal pada sebuah kalimat
3. echo "<b>Belajar Database PHP</b>";
4. ?>
```

## II.7 Tipe Data

Di setiap bahasa pemrograman di kenal ada nya tipe data, termasuk juga di PHP. Tipe data adalah jenis data yang di gunakan oleh PHP untuk menyimpan nilai di dalam sebuah variabel. Untuk menggunakan tipe data, di dalam PHP tidak di perlukan statement tipe data tertentu di depan variabel.

Tipe data yang ada di PHP adalah :

### 1. **Booleans**

Tipe data yang bernilai **True** atau **False** dan penulisan nilai nya dapat berupa case-insensitive.

Contoh :

```
1.<?php
2. $nilai1 = True ;
3. $nilai2 = False
4. ?>
```

### 2. **Integer**

Tipe data yang dapat di tulis dalam decimal (base 10), hexadesimal (base 16) atau octal (base 8). Untuk nilai hexadesimal di awalai angka : 0x dan octal di awali angka : 0

Contoh :

```
1.<?php
2. $nilai1=15; // Memberi nilai 15
3. $nilai2= -20; // Memberi nilai -20
4. $nilai3= 0123; // angka octal (sama dengan 83)
5. $nilai4= 0x1A; // angka hexadecimal (sama dengan 26)
6. ?>
```

Ukuran maximal sekitar 2 milyar (32 bit )

### 3. **Float**

Tipe float adalah tipe yang dapat menampung nilai di belakang koma.

Contoh :

```
1.<?php
2. $nilai1= 2.56;
3. $nilai2=1.2e3;
4. ?>
```

#### 4. **String**

Kita dapat menggunakan koma satu dan koma dua :

##### - **Koma satu**

Contoh :

```
1. <?php
2. $kata1='Saya sedang belajar PHP';
3. $kata2='Nama saya Isma'il'; // Cara menulis kata yang ada koma
4. //di atas nya
5. echo "Kata 1 : $kata1";
6. echo "<br />"; // Baris ke bawah
7. echo "Kata 2 : $kata2";
8. ?>
```

##### - **Koma double**

dengan menggunakan koma double, kita dapat menggunakan karakter tertentu, yaitu :

Sequence	Arti
\n	Baris ke bawah
\r	Mengembalikan carriage
\t	Tab Horizontal
\f	1 baris form
\\	Backslash
\\$	Memberikan tanda dollar
\"	Koma double

Tabel : II.1 . Arti Sequence

Contoh :

```
1.<?php
2. $kata1 = "Saya memiliki uang \$ 15";
3. $kata2 = "Ibukota Indonesia adalah \" Jakarta \"";
4. echo "kata 1 : $kata1 <br />";
5. echo "kata 2 : $kata2";
6. ?>
```

Batas maksimal memory string, adalah penggunaan memory pada PHP.

#### 5. **Array**

Array adalah sebuah variabel yang menyimpan banyak nilai di dalam nya secara teratur, yang setiap nilai memiliki alamat masing-masing secara unique.

Syntax : array().

##### - **Array 1 dimensi**

Adalah array biasa yang terdiri dari 1 dimensi atau daftar nilai.

Untuk mencetak sebuah nilai pada array di gunakan :

\$nama\_array[alamat]

Contoh :

```
1.<?php
2. $hari = array ("Senin", "Selasa", "Rabu", "Kamis", "Jum\at",
3. "Sabtu", "Minggu");
4. echo "Hari ini adalah hari : $hari [0]"
5. ?>
```

Awal alamat array dapat di ubah, seperti :

1. <?php
2. \$hari = array (1=>"Senin", "Selasa", "Rabu", "Kamis", "Jum'at",
3. "Sabtu", "Minggu");
4. print\_r(\$hari); // Mencetak isi array \$hari
5. echo "Hari ini adalah hari : \$hari[1]"; // Hari senin
6. ?>

### - Array 2 dimensi

Adalah sebuah nama array yang dapat menyimpan array lain di bawah nya.

Contoh :

- 1.<?php
2. \$arr2=array("kelas\_5"=>array(1=>"Andi", "Umar", "Ihsan"),
3. "kelas\_6"=>array (1=>"Nia", "Siti", "Amir"));
4. \$data1=\$arr2["kelas\_5"][2]; // Umar
5. \$data2=\$arr2["kelas\_6"][3]; // Amir
6. echo "Data 1 : \$data1 <br/>";
7. echo "Data 2 : \$data2 <br/>";
8. ?>

### 6. Object

Untuk membuat object baru, di gunakan syntax new.

Contoh :

1. <?php
2. class coba
3. {function test()
4. {
5. echo "Testing ... testing";
6. }
7. }
8. \$akses= new coba;
9. \$akses → test();
10. ?>

### 7. NULL

NULL adalah type data yang berarti tidak mengandung nilai.

1. <?php
2. \$nilai1=0;
3. \$nilai2=null;
4. echo "Nilai 1 : \$nilai1";
5. echo "Nilai 2 : \$nilai2";
6. ?>

\$nilai1 mengandung nilai 0 dan \$nilai2 tidak ada nilai atau null.

Suatu variabel yang belum di beri nilai dapat di katakan, bahwa variabel tersebut tidak ada nilai atau null.

## II.8 Variabel

Variabel adalah sesuatu yang di pergunakan untuk menyimpan sebuah nilai. Pemberian nama variabel diawali dengan tanda \$, lalu dapat berupa huruf atau under score dan tidak dapat dengan angka.



Contoh :

1. <?php
2. \$nama="Anwar"; // Tipe variabel string
3. \$angka=4; // Tipe variabel numerik
4. \$benar=true;
5. ?>

Variabel dapat menentukan sendiri tipe nya masing-masing secara otomatis tatkala di beri sebuah nilai, apakah berbentuk string, numerik atau float.

## II.9 Operator

Operator di gunakan untuk menjalankan proses aritmatika, logika, perbandingan, string, array, type , incrementing dan decrementing dan lainnya.

### 1. Operator Aritmetika dan Sama dengan

Contoh	Deskripsi	Keterangan
-\$a	Negasi	Lawan dari \$a
\$a + \$b	Penjumlahan	Penjumlahan variabel \$a terhadap \$b
\$a - \$b	Pengurangan	Pengurangan \$a terhadap \$b
\$a * \$b	Perkalian	Perkalian \$a terhadap \$b
\$a / \$b	Pembagian	Pembagian \$a terhadap \$b
\$a % \$b	Modulus	Sisa pembagian \$a terhadap \$b
5 + 6 = 11 dan \$a = 5	Sama dengan	Hasil dari operasi Aritmetika atau pemberian nilai terhadap suatu variabel

Tabel II.2 : Operator Aritmetika dan Sama Dengan

### 2. Operator Incrementing dan Decrementing

Contoh	Deskripsi	Keterangan
++\$a	Pre-Increment	Menambah \$a dengan 1 dan mengembalikan nilai a
\$a++	Post-Increment	Mengembalikan \$a , kemudian menambah dengan 1
--\$a	Pre-decrement	Mengurangi \$a dengan 1 kemudian mengembalikan \$a
\$a--	Post-decrement	Mengembalikan \$a, kemudian mengurangi \$a dengan 1

Tabel II.3 : Operator Incrementing dan Decrementing

Contoh :

1. <?php
2. \$a=4;
3. echo \$a++ ;
4. echo \$a;
- 5.
6. \$a=4;
7. echo ++\$a;
8. echo \$a;

9. \$a=4;
10. echo \$a--;
11. echo \$a;
- 12.
13. \$a=4;
14. echo --\$a;
15. echo \$a;
16. ?>

### 3. Operator Logika

Contoh	Deskripsi	Hasil
\$a and \$b / \$a && \$b	And	Benar , jika \$a dan \$b benar
\$a or \$b / \$a    \$b	Or	Benar, jika \$a atau \$b benar
\$a xor \$b	Xor	Benar, jika \$a atau \$b benar, tetapi tidak kedua nya.
!\$a	Not	Benar, jika \$a tidak benar

Tabel II.4 : Operator Logika

Operator logika ini sering di gunakan bersamaan dengan feature **if .. endif**

Contoh :

1. <?php
2. \$a=5;
3. \$b=10;
4. if (\$a < 7 && \$b<7)
5. { echo "Kondisi Benar"; }
6. else
7. { echo "Kondisi salah";}
- 8.
9. if (\$a < 7 || \$a < 7)
10. { echo "Kondisi Benar"; }
11. else
12. { echo "Kondisi Salah" ; }
13. ?>

### 4. Operator String

Operator string memiliki 2 operator, yaitu . (titik) dan .= (titik sama dengan). Kedua nya berfungsi untuk menggabungkan 2 atau lebih string.

Contoh :

- 1.<?php
2. \$a = "Belajar";
3. \$a=\$a." PHP";
4. echo \$a; // Belajar PHP
5. echo "<br />";
6. \$a .= " PHP";
7. echo \$a; // Belajar PHP
8. ?>

## 5. Operator Perbandingan

Untuk membandingkan 2 buah nilai atau variabel. Kita dapat menggunakan operator perbandingan, yang terdiri dari :

Contoh	Deskripsi	Hasil
$\$a == \$b$	Sama dengan	Benar, jika \$a sama dengan \$b
$\$a === \$b$	Identik	Benar, jika \$a sama dengan \$b dan kedua nya bertipe sama
$\$a != \$b$ atau $\$a <> \$b$	Tidak sama dengan	Benar, jika \$a tidak sama dengan \$b
$\$a !== \$b$	Tidak identik	Benar, jika \$a tidak sama dengan \$b atau kedua nya tidak identik
$\$a < \$b$	Kurang dari	Benar, jika \$a kurang dari \$b
$\$a > \$b$	Lebih besar dari	Benar, jika \$a lebih besar dari \$b
$\$a <= \$b$	Lebih kecil atau sama dengan	Benar, jika \$a lebih kecil atau sama dengan \$b.
$\$a >= \$b$	Lebih besar atau sama dengan	Benar, jika \$a lebih besar atau sama dengan \$b.

Tabel II.5 : Operator Perbandingan

### Contoh :

```
1.<?php
2. $a=6;
3. $b=8;
4.
5. if ($a == $b)
6.     { echo "Kedua variabel sama dengan"; }
7. else
8.     { echo "Kedua variabel TIDAK sama dengan"; }
9.
10.    if ($a > $b)
11.        { echo " $a lebih besar dari $b"; }
12.    else
13.        { echo " $a tidak lebih besar dari $b"; }
14. ?>
```

## BAB III STRUCTURE CONTROL (1)

Struktur control sangat di perlukan di dalam pemrograman, termasuk web programming, untuk penanganan logika dan algoritma program.

Di bawah ini adalah beberapa Struktur Control yang ada di dalam PHP.

### III.1. While

Syntax : While (expr)  
          statement

Syntax ini akan menjalankan statement selama expr benar.

Contoh :

```
1. <?php
2.   $angka=5;
3.   while ( $angka <=10)
4.     {
5.       echo "Angka : $angka <br/>";
6.       $angka++; // $angka=$angka+1;
7.     }
8. ?>
```

Statement mencetak akan di laksanakan selama variabel \$angka masih di bawah atau sama dengan 10.

### III.2. Do-while

Syntax : do  
          { statement ; }  
          while (expr);

Contoh :

```
1.   <?php
2.     $nilai=10;
3.     do
4.       { echo "Nilai : $nilai";
5.         $nilai++;
6.       }
7.     while ($nilai<=10)
8.     ?>
```

Statement untuk mencetak Nilai di jalankan terlebih dahulu, kemudian variabel \$nilai di tambah 1, baru setelah itu di lakukan cek, apakah variabel \$nilai sama dengan 10 atau tidak.

Statement akan terus di eksekusi selama kondisi \$nilai <= 10 benar.

### III.3. for

Syntax : for (expr1; expr2; expr3)  
          { statement; }  
expr1 : Statement awal  
expr2 : Cek Kondisi  
expr3 : Penjumlahan atau pengurangan

Contoh :

```
1. <?php
2.   for ($angka=1;$angka<=10;$angka++)
3.     { echo "Angka : $angka <br />"; }
4. ?>
```

Perintah di atas akan mencetak statement angka sebanyak 10 kali.

### III.4. foreach

Syntax : `foreach (array_expression as $value)`  
`{ statement }`

`array_expression` : nama array

`$value` : variabel untuk memuat nilai dari element array yang akan secara otomatis bergerak ke elemen array selanjut nya.

Contoh :

```
1. <?php
2.     $data=array("Budiman","Iman","Nani","Susi","Umar");
3.     foreach ($data as $nilai)
4.         { echo "Nama Siswa : $nilai <br/>"; }
5. ?>
```

Output :

```
Nama Siswa : Budiman
Nama Siswa : Iman
Nama Siswa : Nani
Nama Siswa : Susi
Nama siswa : Umar
```

Variabel `$nilai` akan memuat elemen array dari elemen array pertama hingga terakhir, pointer array akan bergerak secara otomatis dari element array pertama hingga terakhir.

- Syntax : `foreach (array_expression as $kunci=>$value)`  
`{ statement }`

Feature ini sama dengan `foreach` sebelum nya, beda nya hanya akan memuat nilai kunci array nya, yang secara default di mulai dari 0.

Contoh :

```
1. <?php
2.     $data=array("Budiman","Iman","Nani","Susi","Umar");
3.     foreach ($data as $kunci =>$nilai)
4.         { echo "Urut :$kunci. Nama Siswa : $nilai <br/>"; }
5. ?>
```

Output :

```
Urut : 0 Nama Siswa : Budiman
Urut : 1 Nama Siswa : Iman
Urut : 2 Nama Siswa : Nani
Urut : 3 Nama Siswa : Susi
Urut : 4 Nama siswa : Umar
```

Variabel `$kunci` memuat key dari element array dan variabel `$nilai` memuat element array nya.

### III.5. Switch

`Switch` berfungsi hampir sama dengan `if .. end if`. `Switch` cocok di gunakan saat kita hendak membandingkan sebuah variabel yang sama dengan nilai-nilai yang berbeda dan mengeksekusi statement yang berbeda, yang tergantung dari kondisi nilai nya.

Setiap bagian, kita harus menyertakan `break`, untuk menghentikan eksekusi yang telah di eksekusi.

Contoh :

```
1. <?php
2.     $nilai=5;
3.     switch ($nilai)
4.         { case 1 :
5.             echo "Nilai nya 1";
6.             break;
7.         case 2 :
```

```

8.     echo "Nilai nya 2";
9.     break;
10.    case 3:
11.        echo "Nilai nya 3";
12.        break;
13.    default :
14.        echo "Selain dari nilai di atas";
15.    }
16. ?>

```

### III.6. Break

Break mengakhiri eksekusi dari structure for,foreach,while, do-while atau switch. Break menerima sebuah nomor pilihan yang merupakan angka seberapa banyak structure yang ia keluar.

#### Contoh :

```

1.<?php
2.  for ($i=1;$i<=10;$i++)
3.      { echo $i . "<br />";
4.        if ($i==7)
5.            { break; } // atau break 1
6.        }
7.  echo "<br />";
8.  for ($i=1;$i<=10;$i++)
9.      {for ($j=1;$j<=5;$j++)
10.         {
11.             echo $i ;
12.             echo "<br />";
13.             echo $j;
14.             if ($i==6)
15.                 { break 2; } // Break untuk for $i dan $j
16.         }
17.     }
18. ?>

```

### III.7. Continue

Berfungsi untuk melewati/skip suatu looping

#### Contoh :

```

1.<?php
2.  for ($i=1;$i<=5;$i++)
3.      {
4.          echo $i;
5.          echo "<br />";
6.          if ($i==3)
7.              { continue; }
8.      }
9. ?>

```

## BAB IV

### STRUKTUR CONTROL (2), FUNGSI DAN PROCEDURE

#### IV.1. If .. End if

Feature **If .. End If** berfungsi untuk menjalankan suatu syntax tertentu jika kondisi tertentu terpenuhi.

Contoh :

```
1.<?php
2.     $nilai1=10;
3.     if($nilai1==10)
4.         { echo "Nilai sesuai "; }
5. ?>
```

#### IV.2. If ... Else ... End If

Feature **If .. Else .. End If** berfungsi untuk menjalankan suatu syntax tertentu jika kondisi tertentu terpenuhi dan ada kondisi lain.

Contoh :

```
1.<?php
2. $nilai1=10;
3. if ($nilai1==10)
4.     { echo "Nilai sesuai"; }
5. else
6.     { echo "Nilai tidak sesuai"; }
7. ?>
```

#### IV.3. If .. Elseif .. Endif

Contoh :

```
1.<?php
2. $nilai1=10;
3. $nilai2=15;
4. if ($nilai1 < $nilai2)
5.     { echo "Pernyataan benar"; }
6. elseif ($nilai1>$nilai2)
7.     { echo "Pernyataan salah"; }
8. else
9.     { echo "No Comment"; }
10. ?>
```

#### IV.4. Function

Jika ada sebuah baris coding yang akan di ulang pada beberapa bagian program atau kita hendak memisahkan suatu fungsi tertentu dari program utama, kita dapat menggunakan function dan procedure.

Syntax :

```
function nama_function(exp1,exp2,...)
{
    statement
}
exp1: variabel (optional)
```

Contoh :

```
1.<?php
2. function bilangan($nilai1,$nilai2)
3. {
4.     $hasil=$nilai1+$nilai2;
5.     echo "Hasil penjumlahan : $hasil";
6. }
7.
```

```

8. echo "Hasil Function nya nich <br />";
9. $bil1=4;
10. $bil2=17;
11. bilangan($bil1,$bil2); // Output adalah 21
12. ?>

```

### 1. Menggunakan return

Return dapat di gunakan di dalam sebuah function untuk mengembalikan sebuah nilai yang dapat bertipe apa saja.

Contoh :

```

1.<?php
2. function bilangan($nilai1,$nilai2)
3. {
4.     $hasil=$nilai1+$nilai2;
5.     return $hasil;
6. }
7.
8. echo "Hasil Function nya nich <br />";
9. $bil1=4;
10. $bil2=17;
11. $hasilnya=bilangan($bil1,$bil2);
12. echo $hasilnya; // Output : 21
13. ?>

```

## **IV.5 Procedure**

Procedure adalah fungsi yang tidak mengemballikan nilai.

Contoh :

```

1.<?php
2.function jumlah()
3. {
4.     $hasil=4+6;
5.     echo $hasil;
6. }
7. ?>

```

Contoh :

```

1.<?php
2.jumlah();
3.function jumlah()
4. { $hasil=4+6;
5.     echo $hasil;
6. }
7. ?>

```



## BAB V

### ARRAY DAN OBJECT ORIENTED (1)

Array adalah sebuah variabel yang menyimpan banyak nilai di dalam nya secara teratur, yang setiap nilai memiliki alamat masing-masing secara unique.

Syntax : array().

#### V.1 Array 1 dimensi

Adalah array biasa yang terdiri dari 1 dimensi atau daftar nilai. Untuk mencetak sebuah nilai pada array di gunakan :

\$nama\_array[alamat]

Contoh :

```
1.<?php
2. $hari = array ("Senin","Selasa","Rabu","Kamis","Jum'at","Sabtu","Minggu");
3. echo "Hari ini adalah hari : $hari [0]";
4. ?>
```

Awal alamat array dapat di ubah, seperti :

```
1. <?php
2. $hari = array (1=>"Senin","Selasa","Rabu","Kamis","Jum'at",
3. "Sabtu","Minggu");
4. print_r($hari); // Mencetak isi array $hari
5. echo "Hari ini adalah hari : $hari[1]"; // Hari senin
6. ?>
```

#### V.2 Array 2 dimensi

Adalah sebuah nama array yang dapat menyimpan array lain di bawah nya.

Contoh :

```
1. <?php
2. $arr2=array("kelas_5"=>array(1=>"Andi","Umar","Ihsan"),
3. "kelas_6"=>array (1=>"Nia","Siti","Amir"));
4. $data1=$arr2["kelas_5"][2]; // Umar
5. $data2=$arr2["kelas_6"][3]; // Amir
6. echo "Data 1 : $data1 <br/>";
7.echo "Data 2 : $data2 <br/>";
8. ?>
```

#### V.3 Object Oriented Programming (OOP) (1)

Selain menerapkan procedural programming, PHP sejak versi 5.0 juga sudah mensupport OOP Programming secara penuh. PBO sendiri dapat di artikan sebagai pengelompokkan modul-modul pemrograman di dalam class-class. Manfaat dari OOP yang terpenting adalah, agar seorang programmer terhindar dari mengulangi kode yang sama untuk tugas yang sama. Programmer cukup menggunakan kode yang sama untuk tugas yang sama dengan hanya memanggil method yang terdapat di dalam class. Di dalam procedural programming, method, dapat di samakan dengan function/procedure.Object Object adalah sebuah paket code yang berhubungan dengan state dan behaviour. Object sering di hubungkan dengan object di dalam kehidupan sehari-hari. Di dalam dunia nyata atau dalam sebuah program, Object memiliki 2 karakteristik, yaitu state dan behaviour. State adalah keadaan dari sebuah Object, sedang behaviour adalah kelakuan dari object tersebut.

Contoh :

Object : Mobil  
State : warna, merk, tahun pembuatan, model  
Behaviour : melaju, berbelok, membunyikan klakson

Sebuah Object yang di bentuk dari class biasa di sebut instance, arti nya Object tersebut adalah wujud nyata dari sebuah class. Variabel dan method dari instance ini di sebut variabel instance dan method instance. Setiap instance menyimpan variabel nya sendiri-sendiri, jadi nilai variabel untuk tiap instance bisa berbeda.

State di dalam OOP di sebut dengan properties dan Behaviour di sebut dengan method.

**- Membuat class**

```
1.<?php
2. class MyClass
3. {
4. // Di isi dengan property class dan method
5. }
6. ?>
```

Class yang baru di buat, dapat di instantiate (Memberi contoh) dan di simpan ke dalam sebuah variabel dengan menggunakan keyword new.

```
$obj = new MyClass;
```

**- Mendefinisikan property class**

Property di dalam sebuah class , dapat juga di sebut dengan variabel di luar OOP. Fungsi nya sama dengan variabel, yakni untuk menyimpan value, namun property terikat dengan dengan sebuah object, sehingga hanya dapat di akses lewat sebuah object.

```
1.<?php
2. class MyClass
3. {
4. public $nama="Akhmad Sofwan";
5. }
6. $obj = new MyClass;
7. ?>
```

Untuk dapat mengakses property \$nama, contoh nya, ingin mencetak di layar, nilai dari property \$nama, maka di gunakan kode : echo \$obj →nama;

**- Mendefinisikan method class**

Method di dalam OOP, sama dengan function / procedure di dalam procedural programming Hanyasaja, sama seperti dengan property, method di akses lewat object dengan menggunakan tanda panah (→).

Contoh:

```
<?php
class MyClass
{
    public $nama="Akhmad Sofwan";
    public function hello()
    {
        return $this→nama;
    }
}
```

```
$obj = new MyClass;
echo $obj → hello();
?>
```

Output :  
Akhmad Sofwan

### - Constructor dan Destructor

Ketika sebuah object di instantiasi, seringkali di perlukan untuk menset sejumlah hal secara otomatis pada awal proses. Dalam hal ini, PHP menyediakan magic method `__construct()`, yang secara otomatis di panggil kapanpun ketika sebuah object baru di buat.

#### Contoh:

```
1.<?php
2.class MyClass
3. {
4.   public $nama="Akhmad Sofwan";
5.   public function __construct()
6.   {
7.     echo 'The class "',__CLASS__,'" was initiated !<br/>';
8.   }
9.
10. public function hello()
11.   {
12.     return $this->nama;
13.   }
14. $obj = new MyClass;
15. echo $obj -> hello();
16. ?>
```

Output :  
The class MyClass was initiated  
Akhmad Sofwan

Pada source code di atas, method `__construct` di jalankan secara otomatis tatkala class `MyClass` di inisiasi, baru kemudian di cetak method `hello`, yang berupa pencetakan property `$nama` dengan value "Akhmad Sofwan".

Untuk memanggil sebuah function ketika object di destroy, PHP menyediakan magic method `__destruct()`. Fungsi nya untuk class cleanup (Misal : Menutup koneksi database)

#### Contoh:

```
1.<?php
2.class MyClass
3.{
4. public $nama="Akhmad Sofwan";
5. public function __construct()
6. {
7.   echo 'The class "',__CLASS__,'" was initiated !<br/>';
8. }
9.
10. public function __destruct()
11. {
12.   echo 'The class "',__CLASS__,'" was destroyed !<br/>';
13. }
14.
15. public function hello()
16. {
17.   return $this->nama;
18. }
19. $obj = new MyClass;
20. echo $obj -> hello();
21. ?>
```

Output :

The class MyClass was initiated

Akhmad Sofwan

The class MyClass was destroyed

Magic method `__destruct` secara otomatis di jalankan, saat aplikasi berakhir. Untuk menjalankan Magic method `__destruct` secara eksplisit, dapat di nyatakan dengan `unset(nama_object)` [3].

## PERTEMUAN VI OBJECT ORIENTED PROGRAMMING (Cont)

### VI.1. Menggunakan class inheritance

Sebuah class dapat mewariskan method dan property dari kelas yang lain, dengan menggunakan keyword extends.

Contoh:

```
1.<?php
2.class oop1
3.{
4.public function penjumlahan($a,$b)
5. {
6. $c=$a+$b;
7. return $c;
8. }
9.}
10.
11.class MyanotherClass extends oop1
12. {
13. public function perkalian($a,$b)
14. {
15. $c=$a*$b;
16. return $c;
17. }
18. }
19.
20. $obj = new oop1;
21. echo $obj->penjumlahan(2,6)."<br/>";
22.
23. $obj1=new MyanotherClass;
24. echo $obj1->penjumlahan(4,10)."<br/>";
25. echo $obj1->perkalian(4,10)."<br/>";
26. ?>
```

Output :

```
8
14
40
```

### VI.2. Memberikan hak akses dari property dan method

Untuk mengakses property dan method terdapat beberapa keyword yang berbeda-beda, yakni public, protected dan private. Terdapat juga keyword static, yang mana property dan method dengan hak akses ini dapat di akses langsung tanpa menginstansiasi class.

#### - Public property dan Method

Public : Arti nya property dan method dengan keyword ini dapat di akses di mana saja, di dalam maupun di luar class nya. Protected property dan Method

Protected : Arti nya property dan method dengan keyword ini dapat di akses hanya di dalam class atau class yang meng extends class yang memiliki property dan method ini.

```
1.<?php
2. class oop2
3. {
4. protected function penjumlahan($a,$b)
```

```

5.     {
6.         $c=$a+$b;
7.         return $c;
8.     }
9. }
10.
11.class MyanotherClass extends oop2
12. {
13.     public function perkalian($a,$b)
14.     {
15.         $c=$a*$b;
16.         return $c;
17.     }
18.
19.public function callprotected()
20. {
21.     return $this->penjumlahan(4,5);
22. }
23. }
24.
25. $obj = new oop2;
26.
27. $obj1=new MyanotherClass;
28. echo $obj1->callprotected()."<br/>";
29. echo $obj1->perkalian(4,10)."<br/>";
30. ?>

```

Output :

9  
40

Pada contoh source code di atas, function penjumlahan(\$a,\$b) adalah protected (baris : 4) dan tidak dapat di panggil langsung oleh objek \$obj1. Function penjumlahan(\$a,\$b) dapat di panggil oleh method callprotected yang bersifat public pada class MyanotherClass (baris : 19 - 22). MyanotherClass sendiri mewarisi class oop2. Baru, method callprotected yang public dapat di panggil oleh \$obj1 (baris : 28).

### - Private properties dan methods

Private : Method dan property hanya dapat di akses oleh class itu sendiri. Class lain dan class yang mengextends nya tidak dapat mengakses. Pada contoh source code di atas, jika baris 4 di ganti menjadi: private function penjumlahan(\$a,\$b), maka saat program di eksekusi, output nya akan keluar pesan error, sbb :

Fatal error: Call to private method oop2::penjumlahan() from context 'MyanotherClass' in /opt/lampp/htdocs/belajar-raharja/oop2.php on line 21karena method penjumlahan di panggil oleh class MyanotherClass, yang merupakan class eksternal, walaupun class tersebut mewarisi class MyClass, class nya method penjumlahan.

### - Static properties dan method

Sebuah method atau property yang di deklarasikan static, dapat langsung di akses tanpa melakukan instantiate class. Anda hanya memberikan nama class, scope operator resolusi dan nama property atau method.

Contoh:

```

1.<?php
2.class oop2
3. {
4.     public static function penjumlahan($a,$b)
5.     {
6.         $c=$a+$b;

```

```

7.     return $c;
8.     }
9.}
10.
11. echo oop2::penjumlahan(10,7);
12. ?>

```

Output :  
17

### VI.3. Memberikan komentar dengan DocBlocks

Seringkali kita menjumpai komentar yang ada di atas sebuah class. Komentar di perlukan untuk memberikan informasi mengenai sejumlah informasi terhadap class atau file tersebut, seperti : author, copyright dan lisensi. Komentar di dalam PHP diawali dengan tag /\* dan diakhiri dengan \*/ atau jika perbaris, dapat diawali dengan // . Komentar tidak di proses oleh komputer.

Cara memberikan komentar dengan DocBlocks, sudah di terima secara meluas.

Tag yang paling umum di gunakan oleh Docblock adalah :

@author : Pencipta dari class atau file . Format nya adalah nama pencipta dan e-mail.

Contoh :

@author Akhmad Sofwan (sofwan@sofwan.net)

@Copyright : Berisi tahun copyright dan nama pemegang copyright.

Contoh :

@Copyright 2015 Akhmad Sofwan

@license : Link license ke file tersebut dan nama license.

Contoh :

@license <https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html> GPL 2

### VI.4. Membandingkan Procedural dengan Object Oriented Programming (OOP)

Procedural bagus diterapkan di dalam pemrograman, namun di sisi lain OOP menawarkan kelebihan mudahan dalam berhubungan dengan data. Sebagai contoh, anda memiliki informasi mengenai orang yang perlu diproses, informasi tersebut adalah nama, pekerjaan dan usia. Terdapat beberapa

- **keunggulan OOP dibanding procedural, yaitu :**

#### 1. Kemudahan implementasi

Object di dalam OOP dapat menyimpan data secara internal, variabel tidak perlu dilewatkan dari function ke function. Contoh source code procedural dan OOP dibawah ini, menjelaskan hal tersebut.

```

1.<?php
2. function changeJob($person, $newjob)
3. {
4.     $person['job'] = $newjob; // Change the person's job
5.     return $person;
6. }
7.
8. function happyBirthday($person)
9. {
10.     ++$person['age']; // Add 1 to the person's age
11.     return $person;
12. }
13.
14. $person1 = array(
15.     'name' => 'Tom',
16.     'job' => 'Button-Pusher',

```

```

17.'age' => 34
18. );
19.
20. $person2 = array(
21. 'name' => 'John',
22. 'job' => 'Lever-Puller',
23.'age' => 41
24. );
25.
26. // Output the starting values for the people
27. echo "<pre>Person 1: ", print_r($person1, TRUE), "</pre>";
28. echo "<pre>Person 2: ", print_r($person2, TRUE), "</pre>";
29.
30. // Tom got a promotion and had a birthday
31. $person1 = changeJob($person1, 'Box-Mover');
32. $person1 = happyBirthday($person1);
33.
34. // John just had a birthday
35. $person2 = happyBirthday($person2);
36.
37. // Output the new values for the people
38. echo "<pre>Person 1: ", print_r($person1, TRUE), "</pre>";
39. echo "<pre>Person 2: ", print_r($person2, TRUE), "</pre>";
40. ?>

```

Output :

```

Person 1: Array
(
[name] => Tom
[job] => Button-Pusher
[age] => 34
Person 2: Array
(
[name] => John
[job] => Lever-Puller
[age] => 41
)
Person 1: Array
(
[name] => Tom
[job] => Box-Mover
[age] => 35
)
Person 2: Array
(
[name] => John
[job] => Lever-Puller
[age] => 42
)

```

Pada kode procedural di atas, array yang menyimpan atribut person harus di lewatkan dan dikembalikan dari masing-masing pemanggilan function yang dapat meninggalkan margin untuk error.

Untuk mengatasi hal ini, hendaknya hanya informasi yang penting saja untuk operasi saat ini yang di lewatkan ke functions atau metode. Hal ini di akomodir oleh OOP, dengan kode sebagai berikut :



```

1.<?php
2.class Person
3. {
4. private $_name;
5. private $_job;
6. private $_age;
7. public function __construct($name, $job, $age)
8. {
9.     $this->_name = $name;
10.    $this->_job = $job;
11.    $this->_age = $age;
12. }
13.
14. public function changeJob($newjob)
15. {
16.     $this->_job = $newjob;
17. }
18.
19. public function happyBirthday()
20. {
21.     ++$this->_age;}
22. }
23.
24. // Create two new people
25. $person1 = new Person("Tom", "Button-Pusher", 34);
26. $person2 = new Person("John", "Lever Puller", 41);
27.
28. // Output their starting point
29. echo "<pre>Person 1: ", print_r($person1, TRUE), "</pre>";
30. echo "<pre>Person 2: ", print_r($person2, TRUE), "</pre>";
31.
32. // Give Tom a promotion and a birthday
33. $person1->changeJob("Box-Mover");
34. $person1->happyBirthday();
35.
36. // John just gets a year older
37. $person2->happyBirthday();
38.
39. // Output the ending values
40. echo "<pre>Person 1: ", print_r($person1, TRUE), "</pre>";
41. echo "<pre>Person 2: ", print_r($person2, TRUE), "</pre>";
42. ?>

```

Output :

Person 1: Person Object

```

(
[_name:Person:private] => Tom
[_job:Person:private] => Button-Pusher
[_age:Person:private] => 34
)

```

Person 2: Person Object

```

(
[_name:Person:private] => John
[_job:Person:private] => Lever Puller
[_age:Person:private] => 41
)

```

Person 1: Person Object

```
(  
[_name:Person:private] => Tom  
[_job:Person:private] => Box-Mover  
[_age:Person:private] => 35  
)  
Person 2: Person Object  
(  
[_name:Person:private] => John  
[_job:Person:private] => Lever Puller  
[_age:Person:private] => 42 )
```

OOP membutuhkan waktu untuk memahaminya dan menerapkannya, namun saat sudah implementasi, kemudahan dalam pengorganisasian kode dan data akan kita dapatkan, terutama di dalam aplikasi yang cukup besar. Walau begitu, tidak semua hal di dalam pemrograman ideal untuk diterapkannya OOP, hal hal kecil di dalam program, seringkali justru lebih baik diterapkan di dalam procedural.

## **2. Organisasi struktur program yang lebih baik**

Class-class di dalam OOP dapat dibuat menjadi sebuah file yang di dalam file tersebut dapat dibuat menjadi 1 fitur atau lebih namun yang berdekatan. Hal itu memudahkan jika kita ingin menggunakan kembali fitur-fitur tersebut, di tempat lain dalam sebuah program, yakni hanya tinggal memanggilnya kembali.

## **3. Mudah dalam pemeliharaan**

Bayangkan, jika di tengah-tengah pemrograman, kita ingin menambahkan parameter di dalam sebuah kode. Di dalam procedural, kita harus menambahkan parameter tersebut secara manual dari function ke function. Di dalam OOP, kita tinggal membuat parameter yang bersifat publik dan menambahkan metode untuk menangani fitur baru yang ingin ditambahkan [3].

## BAB VII

### PEMBUATAN PROGRAM SEDERHANA BERBASIS PHP

Pada Pertemuan ke VII akan di bahas pembuatan program sederhana Berbasis PHP. Pembuatan program dalam bentuk soal dan jawab.

#### 1. Program 1.

Buatlah program dengan PHP untuk mencari sisi miring dan keliling pada segitiga siku-siku. Entri alas (a) dan tinggi (b) dengan Keyboard.

Rumus sisi miring :  $c = \sqrt{a^2 + b^2}$

Rumus Keliling :  $k = a + b + c$

Jika nilai keliling  $> 100$  , maka keterangan adalah “Segitiga Besar”

Jika nilai keliling  $\geq 50$  dan  $< 100$ , maka keterangan adalah “Segitiga Sedang”

Jika nilai keliling  $> 0$  dan  $< 50$ , maka keterangan adalah “Segitiga Kecil”

Contoh :

Alas (a) : 4

Tinggi (b) : 6

(Enter)

Sisi Miring : 8.94

Keliling : 18.94

Keterangan : Segitiga kecil

#### Nama file : keliling.php

```
1.<!DOCTYPE html>
2.<html>
3.<head>
4. <title>Segitiga</title>
5.</head>
6.<body>
7. <table>
8. <form method="post" action="hasil.php">
9. <tr>
10. <td>Alas(a)</td><td>:</td><td><input type="text" name="valas" size="5" /></td>
11.</tr>
12.<tr>
13. <td>Tinggi(b)</td><td>:</td><td><input type="text" name="vtinggi" size="5" /></td>
14. </tr>
15. <tr>
16. <td colspan="3"><input type="submit" value="OK" /></td>
17.</tr>
18.</form>
19.</table>
20.</body>
21.</html>
```

### **Nama File : hasil.php**

```
1.<?php
2. $xalas=$_POST["valas"];
3. $xtinggi=$_POST["vtinggi"];
4. $xsisi_miring=sqrt(($xalas*$xalas)+($xtinggi*$xtinggi));
5. $xkeliling=$xalas+$xtinggi+$xsisi_miring;
6.
7. // Cari keterangan //
8. if ($xkeliling > 100)
9. {
10. $keterangan="Segitiga Besar";
11. }
12. else if ($xkeliling >=50 and $xkeliling <100)
13. {
14. $keterangan="Segitiga Sedang";
15. }
16.else if ($xkeliling >0 and $xkeliling < 50)
17. {
18. $keterangan="Segitiga kecil";
19. }
20.
21. echo "Nilai Alas : $xalas <br/>";
22. echo "Nilai Tinggi: $xtinggi <br/>";
23. echo "----- <br/>";
24. echo "Sisi Miring : ".round($xsisi_miring,2)."<br/>";
25. echo "Keliling : ".round($xkeliling,2)."<br/>";
26. echo "Keterangan : $keterangan <br/>";
27. echo "<input type='button' value='Kembali' onClick='history.back(1)' />";
28.??>
```

### **2. Program 2.**

Buatlah program untuk mencetak deret dari X ke max Y dengan lompatan Z. Jika X lebih kecil dari Y, maka merupakan deret naik, dengan pencetakan di awali karakter : ">". Jika X lebih besar dari Y, maka merupakan deret turun, dengan pencetakan di awali karakter : "<". Dan jika tidak ada deret yang dibuat, maka tidak mencetak apapun. X, Y dan Z diinput melalui keyboard

#### **Contoh 1 :**

X : 4

Y : 11

Z : 2

Output :

> 4 6 8 10

#### **Contoh 2 :**

X : 13

Y : 3

Z : 4

Output :

< 13 9 5

### **Jawab :**

```
1.<!DOCTYPE html>
2.<html>
3.<head>
```

```

4. <title>Pertemuan 7-2</title>
5.</head>
6.<body>
7.<table>
8.<form method="post" action="<?php echo $_SERVER["PHP_SELF"];?>" >
9.<tr>
10. <td>X</td><td>:</td><td><input type="text" name="vx" size="5" /></td>
11.</tr>
12.<tr>
13. <td>Y</td><td>:</td><td><input type="text" name="vy" size="5" /></td>
14.</tr>
15.<tr>
16. <td>Z</td><td>:</td><td><input type="text" name="vz" size="5" /></td>
17.</tr>
18.<tr>
19. <td><input type="submit" value="Proses" /></td>
20.</tr>
21.</form>
22.</table>
23.<?php
24. if (isset($_POST["vx"]) && isset($_POST["vy"]) && isset($_POST["vz"]))
25. {
26.     $zx=$_POST["vx"];
27.     $zy=$_POST["vy"];
28.     $zz=$_POST["vz"];
29.
30.     if ($zx<$zy)
31.     {
32.         echo "> ";
33.         for ($i=$zx;$i<=$zy;$i=$i+$zz)
34.         {
35.             echo $i." ";
36.         }
37.     }
38.     else if ($zx>$zy)
39.     {
40.         echo "< ";
41.         for ($j=$zx;$j>=$zy;$j=$j-$zz)
42.         {
43.             echo $j." ";
44.         }
45.     }
46. }
47.?>
48.</body>
49.</html>

```

## Bab IX

### Operasi String dan File

#### IX.I. Operasi String

Berikut akan diuraikan beberapa fungsi yang berkaitan dengan operasi String.

##### 1. substr

Syntax : string substr (string \$string, int \$start [,int \$length])

Fungsi : Untuk mengambil bagian dari sebuah string.

Contoh :

1. <?php
2. \$nama = "Akhmad Sofwan";
3. \$ambil1 = substr (\$nama,6); //Mengambil 6 karakter dari kanan
4. \$ambil2 = substr (\$nama,0,6); // Mengambil 6 karakter dari posisi 0 (awal)
5. echo "Part 1 : \$ambil1 <br/>";
6. echo "Part 2 : \$ambil2";
7. ?>

Output :

Part 1 : Sofwan

Part2 : Akhmad

##### 2. strtolower

Syntax : string strtolower (string \$str)

Fungsi : Merubah string menjadi huruf kecil

Contoh:

1. <?php
2. \$kata = "Bandung Lautan Api";
3. echo strtolower(\$kata);
4. ?>

Output :

bandung lautan api

##### 3. strtoupper

Syntax : string strtoupper (string \$str)

Fungsi : Merubah string menjadi huruf besar

Contoh :

1. <?php
2. \$kata = "Bandung Lautan Api";
3. echo strtoupper(\$kata);
4. ?>

Output :

BANDUNG LAUTAN API

##### 4.strlen

Syntax : int strlen (string \$string)

Fungsi : Menghitung jumlah karakter pada sebuah string

Contoh :

1. <?php
2. \$kata = "Bandung Lautan Api";
3. echo strlen(\$kata);
4. ?>

Output :

18

## 5. **strpos**

Syntax : int strpos (string \$string, mixed \$needle[, int \$offset = 0])

Keterangan

\$string : String untuk pencarian

mixed \$needle : Elemen yang di cari

\$offset : Awal pencarian. Namun hasil pencarian tetap dari posisi awal.

Fungsi : Mencari posisi awal dari string yang di cari.

Contoh :

1. <?php
2. \$kata = "Bandung Lautan Api";
3. \$cari = "u";
4. \$dicari=strpos(\$kata,\$cari);
5. echo "Hasil : \$dicari";
6. ?>

Output :

Hasil : 4

## 6. **implode**

Syntax : string implode (string \$glue, array \$pieces)

Keterangan

\$glue : Berisi string yang akan di gabungkan dengan elemen array

array \$pieces : Array , yang elemen nya akan di gabung kan.

Fungsi : Untuk menggabungkan string dengan elemen array

Contoh :

1. <?php
2. \$array = array('lastname','email','phone');
3. \$comma\_separated = implode(" ", \$array);
4. echo \$comma\_separated; // lastname,email,phone
5. // Empty string when using an empty array:
6. var\_dump(implode('hello', array())); // string(0) ""
7. ?>

Output :

lastname\*email\*phone

string(0) ""

## 7. **explode**

Syntax : array explode (string \$delimiter, string \$string [,int \$limit])

Keterangan :

\$delimiter : string yang akan menjadi pemisah

\$string : inputan string

\$limit : batas

Fungsi : Untuk memisahkan string dengan string

1. <?php
2. //Example 1
3. \$pizza="piece1 piece2 piece3 piece4 piece5 piece6";
4. \$pieces=explode(" ",\$pizza);
5. echo \$pieces[0];//piece1
6. echo \$pieces[1];//piece2
- 7.

8. //Example 2
9. \$data="foo:\*:1023:1000:~/home/foo:/bin/sh";
10. list(\$user,\$pass,\$uid,\$gid,\$gecos,\$home,\$shell)=explode(":",\$data);
11. echo \$user; //foo
12. echo \$pass; /\*
13. ?>

Output :

piece1 piece2  
foo\*

**Operasi file**

Berikut akan diuraikan beberapa syntax di dalam operasi file .

**1. fopen**

Syntax : resource fopen (string \$filename, string \$mode);

Keterangan :

\$filename : Berupa nama file

\$mode : Tipe akses file

Beberapa mode yang dapat di gunakan :

'r' : Hanya membuka, meletakkan file pointer di awal file

'r+' : Membuka untuk membaca dan menulis. Meletakkan pointer di awal file

'w' : Di buka untuk di tulis. Meletakkan pointer file di awal file dan memotong panjang file menjadi 0.

Jika file tidak ada, maka file akan di buat.

'w+' : Membuka untuk di baca dan di tulis. Meletakkan pointer file di awal file dan memotong panjang file

menjadi 0. Jika file tidak ada, maka file akan di buat.

Contoh :

1. <?php
2. \$buka = fopen ("~/home/sofwan/test.txt","r");
3. ?>

**2. fread**

Syntax : string fread (resource \$handle, int \$length);

Keterangan :

\$handle : Pointer file , biasa nya di buat dengan menggunakan fopen

\$length : Sampai panjang jumlah byte yang di baca

Contoh :

1. <?php
2. //get contents of a file into a string
3. \$filename="/usr/local/something.txt";
4. \$handle=fopen(\$filename,"r");
5. \$contents=fread(\$handle,filesize(\$filename));
6. fclose(\$handle);
7. ?>

**3. fwrite**

Syntax : int fwrite (resource \$handle, string \$string [, int \$length])

Keterangan :

\$handle : File system pointer, biasa nya di bentuk oleh fopen.

\$string : String yang akan di tulis.

Length : panjang byte max yang akan di tulis.

Fungsi : Untuk menulis string ke dalam sebuah file.



```
1. <?php
2. $fp=fopen('/home/sofwan/test.txt','w+');
3. fwrite($fp,'1');
4. fwrite($fp,'23');
5. fclose($fp);
6.
7. // the content of 'data.txt' is now 123 and not 23!
8. ?>
```

## **BAB X**

### **Koneksi PHP dengan Database menggunakan PDO**

PHP Data Object (PDO) extension adalah interface yang ringan dan konsisten untuk mengakses berbagai macam database di PHP. Setiap Driver database yang menerapkan interface PDO dapat mengekspose fitur khusus database sebagai fungsi ekstensi regular. Dengan menggunakan PDO, anda dapat membuat code yang portabel/sama untuk Database dan Platform yang berbeda.

Perbedaan hanya pada driver dan koneksi saja. Untuk retrieve, delete, update dan insert, anda dapat menggunakan script yang sama atau hampir sama untuk database dan platform yang berbeda.

PDO sudah tersedia sejak PHP 5.1 dan di kenal sebagai PECL Extension untuk PHP 5.0. PDO membutuhkan fitur OO yang ada di PHP 5.x, sehingga PDO tidak kompatibel pada PHP 4.x dan versi di bawah nya.

PDO mendukung Database sbb :

- DBLIB : FreeTDS / Microsoft SQL Server / Sybase
- Firebird : Firebird / Interbase 6
- IBM (IBM DB2)
- Informix – IBM Informix Dynamic Server
- Mysql : Mysql 3.x / 4.0
- OCI : Oracle Call Interface
- ODBC : ODBC v3 (IBM DB2 dan UnixODBC)
- PGSQL : PotgreSql
- SqlLite : SQLite 3.x

Untuk mengakses setiap Database di atas, di butuhkan driver yang di kenal oleh PHP. Anda dapat melihat di phpinfo driver-driver apa saja yang sudah terinstall dan di kenal oleh PHP di system anda, atau anda dapat menjalankan kode berikut [1]:

```
1.<?php
2. foreach(PDO::getAvailableDrivers() as $driver)
3. {
4.     echo $driver.'<br />';
5. }
6. ?>
```

Untuk dapat menjalankan driver database pada PHP, anda dapat menggunakan parameter sbb (Untuk Mysql, PostgreSQL dan Sqlite) :

- Mysql : --with-pdo-mysql=mysqlnd
- PostgreSQL : --with-pdo-pgsql=/opt/lampp/postgresql
- Sqlite : --with-pdo-sqlite

Untuk mengaktifkan driver-driver database lain, pada pengguna Windows, dapat meng enable dll file database-database tsb pada php.ini, seperti :

```
extension=php_pdo.dll
extension=php_pdo_firebird.dll
extension=php_pdo_informix.dll
extension=php_pdo_mssql.dll
```

## X.1 Menghubungkan ke Database

### Mysql

Mysql adalah database yang paling banyak di gunakan oleh Programmer PHP dan mendapat dukungan penuh dari PHP. Contoh menghubungkan Mysql dari PHP dengan PDO adalah sbb :

```
1.<?php
2.$dsn = 'mysql:host=localhost;dbname=test';
3.$username = 'opan';
4.$password = 'opansan';
5.
6. try {
7. $dbh = new PDO($dsn, $username, $password);
8. echo "Database connected";
9. }
10.catch (PDOException $e)
11. {
12. echo $e->getMessage();
13. }
14. ?>
```

### PostgreSQL

```
1.<?php
2. try
3. {
4. $dbh = new PDO("pgsql:dbname=pdo:host=localhost","username","password");
5. echo "PDO Connection object created";
6. }
7. catch (PDOException $e)
8. {
9. echo $e->getMessage();
10. }
11. ?>
```

## X.2 Insert, Select dan Update

Seperti yang sudah di sebutkan di atas, bahwa keunggulan PDO adalah, bahwa sekali database terkoneksi, maka untuk masalah query, itu sama, antara 1 database dengan database lain, sehingga anda tidak perlu di reportkan saat ingin mengganti database. Tentu saja selama database tersebut di akses dengan bahasa SQL. Walaupun ada perbedaan, namun hanya sedikit, menyesuaikan gaya SQL pada masing-masing database. Untuk penggunaan query, kita menggunakan database Mysql.

### Insert

Sebelum anda mempraktekan query, buatlah tabel sbb :

Nama kolom	Tipe	Keterangan
id	int (4)	Primary Key, Auto Increment
nim	varchar (30)	
nama	varchar (30)	
alamat	varchar (30)	

Pada script di atas, tambahkanlah script untuk menginsert, yakni :

```
$dbh->exec ("insert into mahasiswa (nim,nama,alamat) values ('1211500100','Budiman','Jl.Mawar No.14
```

Jakarta Selatan')");

Sehingga script akhir menjadi sebagai berikut :

```
1.<?php
2. $dsn = 'mysql:host=localhost;dbname=test';
3. $username = 'opan';
4. $password = 'opansan';
5.
6. try {
7.   $dbh = new PDO($dsn, $username, $password);
8.   echo "Database connected";
9. }
10. catch (PDOException $e)
11. {
12.   echo $e->getMessage();
13. }
14.
15. // Insert Database
16.$query=$dbh->exec ("insert into mahasiswa (nim,nama,alamat) values
17.('1211500100','Budiman','Jl.Mawar No.14 Jakarta Selatan)");
18.
19.if ($query)
20. {
21.   echo "Berhasil di insert";
22. }
23.else
24. {
25.   echo "Tidak berhasil di insert";
26. }
27.??>
```

### **Select**

Untuk menampilkan isi database, anda menggunakan perintah 'select' pada query. Query lengkap nya adalah sbb :

```
<?php
1.$dsn = 'mysql:host=localhost;dbname=test';
2.$username = 'opan';
3.$password = 'opansan';
4.
5. try {
6.   $dbh = new PDO($dsn, $username, $password);
7.   echo "Database connected <br/>";
8.   }
9. catch (PDOException $e)
10. {
11.   echo $e->getMessage();
12. }
13.
14. // Select database //
15. $querynya="select * from mahasiswa";
16.
17. foreach ($dbh->query($querynya) as $row)
18. {
19.   echo "NIM :".$row['nim']."<br/>";
20.   echo "Nama :".$row['nama']."<br/>";
```

```
21. echo "Alamat :".$row['alamat']."<br/>";  
22. }  
23. ?>
```

## BAB XI

### Membuat aplikasi CRUD sederhana

CRUD (Create, Read, Update dan Delete) sering digunakan dalam banyak aplikasi, termasuk juga di dalam pemrograman web. Create, yakni melakukan input data ke dalam tabel di sebuah database, Read, yakni membaca data pada tabel, Update, melakukan edit data dan Delete, menghapus data.

Untuk dapat menjalankan aplikasi sederhana ini, anda perlu menginstalasi Apache, PHP dan Mysql. Cara mudah untuk menginstall aplikasi-aplikasi tersebut adalah dengan mendownload dan menginstall Xampp, yang dapat anda download di [www.apachefriends.org](http://www.apachefriends.org). Setelah melakukan instalasi, script aplikasi CRUD ini dapat anda taruh di direktori `c:/xampp/htdocs` jika anda menggunakan Sistem Operasi Windows atau di direktori `/opt/lampp/htdocs`, jika anda menggunakan sistem operasi Linux.

#### IX.1. Contoh Aplikasi

Pada Contoh Aplikasi kali ini, kita akan membuat aplikasi CRUD sederhana, yakni, melakukan entri sebuah bilangan ( $n$ ), yang bilangan tersebut akan melakukan pencetakan karakter "\*" sebanyak  $n$  kali dan memunculkan keterangan, yakni. Jika nilai lebih besar atau sama dengan 100, Keterangan = "Besar", jika nilai lebih besar atau sama dengan 50 dan lebih kecil dari 100, maka keterangan = "Sedang" dan selain itu, keterangan = "Kecil". Banyak karakter yang dicetak dan keterangan disimpan di dalam sebuah tabel.

##### Langkah-langkah:

##### 1. Membuat form entri.

Form entri hanya memuat sebuah Text Field dan Button.



Gambar XI.1 : Form Entri

Pada form entri tersebut, terdapat menu Entri dan Lihat. Menu Entri dan Lihat selalu ada di setiap halaman aplikasi, oleh karena itu, kita perlu memisahkan file menu dan dipanggil di setiap file lain, dengan menggunakan include.

##### 2. Membuat proses untuk pengolahan data

Setelah data  $n$  di entri, maka data dipergunakan untuk mencetak karakter \* sebanyak  $n$  kali dan dicari keterangan dengan menggunakan aturan yang ditentukan. Pencetakan karakter menggunakan looping dan pencarian keterangan dengan condition.

### 3. Membuat Tampilan data

Tampilan data menggunakan tabel, berisi data-data yang sudah disimpan di dalam database. Tampilan data memiliki opsi edit dan delete. Edit, data yang terpilih dapat di edit dan Delete, data yang terpilih dapat



No	Nilai	KETERANGAN	Opsi
1	40	*****	<a href="#">Edit</a>   <a href="#">Hapus</a>
2	40	*****	<a href="#">Edit</a>   <a href="#">Hapus</a>
3	20	*****	<a href="#">Edit</a>   <a href="#">Hapus</a>
4	5	*****	<a href="#">Edit</a>   <a href="#">Hapus</a>
5	7	*****	<a href="#">Edit</a>   <a href="#">Hapus</a>
6	2	**	<a href="#">Edit</a>   <a href="#">Hapus</a>
7	2	**	<a href="#">Edit</a>   <a href="#">Hapus</a>
8	2	**	<a href="#">Edit</a>   <a href="#">Hapus</a>
9	2	**	<a href="#">Edit</a>   <a href="#">Hapus</a>
10	2	**	<a href="#">Edit</a>   <a href="#">Hapus</a>
11	2	**	<a href="#">Edit</a>   <a href="#">Hapus</a>
12	2	**	<a href="#">Edit</a>   <a href="#">Hapus</a>
13	2	**	<a href="#">Edit</a>   <a href="#">Hapus</a>
14	2	**	<a href="#">Edit</a>   <a href="#">Hapus</a>
15	4	****	<a href="#">Edit</a>   <a href="#">Hapus</a>
16	2	**	<a href="#">Edit</a>   <a href="#">Hapus</a>
17	2	**	<a href="#">Edit</a>   <a href="#">Hapus</a>

didelete.

Gambar XI.2 : Tampil Data

### 4. Informasi Aksi

Setiap Entri dan Delete yang sukses, diinformasikan kepada user. Halaman informasi ini perlu juga kita buat.



Gambar XI.3 : Informasi Aksi

## Bab XII

### Validasi dan Verifikasi

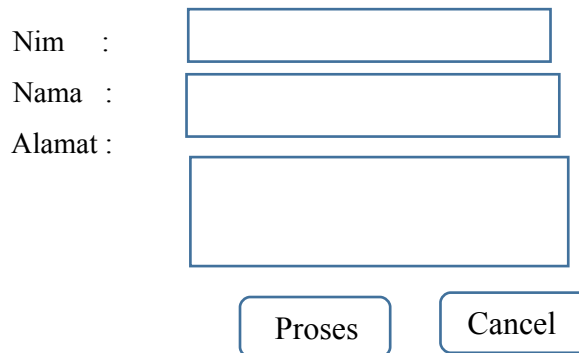
#### XII.1. Validasi

Di dalam sebuah form, kita perlu melakukan validasi terhadap isian form, seperti sebuah field yang tidak boleh kosong dan harus terisi atau input e-mail yang harus sesuai format e-mail. Validasi di dalam Ilmu Komputer adalah meyakinkan data dimasukkan ke dalam aplikasi agar sesuai format dan input kriteria lainnya (en.wikipedia.org).

Validasi di dalam web dapat menggunakan php dan javascript. Untuk saat ini, kita mencoba melakukan validasi dengan menggunakan JavaScript dan PHP, yaitu validasi inputan, yang harus di isi pada field nim pada form inpuan data mahasiswa.

Penggunaan JavaScript sebagai validasi pada pemrograman web di Client memiliki keuntungan, yakni, validasi lebih user interaktif dibanding dengan PHP, namun PHP lebih aman, karena scriptnya tidak dapat dilihat dan tidak dapat dibatalkan atau di By Pass dengan menggunakan opsi “disabled” di browser, seperti halnya JavaScript.

#### XII.2 Validasi di form dengan menggunakan JavaScript.



Nim :

Nama :

Alamat :

Gambar XII.1. Form Entri

Pada form di atas, kita buat validasi dengan mewajibkan user untuk mengisi NIM. Untuk itu, kita insert kode Java Script di dalam html. Nama file nya adalah : form\_mhs\_validasi\_php.php

```
1.<!DOCTYPE html>
2.<html>
3. <head>
4. <title>Form Mahasiswa</title>
5. <script>
6.   function cek()
7.     {
8.       var nim=myform.vnim.value;
9.       if (nim.length==0)
10.        {
11.          alert ("NIM tidak boleh kosong");
12.          return false;
13.        }
14.     }
15. </script>
16. </head>
17. <body>
18.   <form name="myform" method="post" onSubmit="return cek()" >
20.   <table>
21.     <caption>Form Mahasiswa</caption>
```



```

22.     <tr>
23.         <td>NIM</td><td>:</td><td><input type="text" name="vnim" size="20" /></td>
24.     </tr>
25. <tr>
26.     <td>Nama</td><td>:</td><td><input type="text" name="vnama" size="20" /></td>
27. </tr>
28. <tr>
29.     <td>Alamat</td><td>:</td><td><textarea rows="3" cols="40" 30.
30.         name="valamat"></textarea></td>
31. </tr>
32. <tr>
33.     <td colspan="3"><input type="submit" value="Proses" /><input type="button"
34.         value="Cancel" /></td>
35. </tr>
36. </table>
37. </form>
38. </body>
39. </html>

```

Pada Script baris 5 – 15, kita insert function Java Script untuk validasi.

```

5. <script>
6.     function cek()
7.     {
8.         var nim=myform.vnim.value;
9.         if (nim.length==0)
10.            {
11.                alert ("NIM tidak boleh kosong");
12.                return false;
13.            }
14.     }
15. </script>

```

Pada script HTML, kita insert event **onSubmit**, yang berfungsi untuk menjalankan function ketika form di submit. Saat field NIM tidak diisi dan kita tekan tombol, maka akan keluar pop up dengan kalimat : “NIM tidak boleh kosong”.

## XII.3 Validasi pada form dengan menggunakan PHP

Kita akan mencoba membuat validasi pada form di Gambar XII.1, yakni dengan mem validasi NIM agar harus melakukan inputan dengan tidak melanjutkan proses pengolahan data dan memberikan pesan, agar field NIM diisi, jika kosong.

Script lengkapnya adalah sebagai berikut :

```

1.<!DOCTYPE html>
2.<html>
3. <head>
4.     <title>Form Mahasiswa</title>
5. </head>
6. <body>
7.     <form name="myform" method="post"
8.         action="<?php echo $_SERVER['PHP_SELF'];?>"
9.     <table>
10.         <caption>Form Mahasiswa</caption>
11.     <tr>
12.         <td>NIM</td><td>:</td><td><input type="text" name="vnim" size="20" /></td>

```

```

13. </tr>
14. </tr>
15. <td>Nama</td><td>:</td><td><input type="text" name="vnama" size="20" /></td>
16. </tr>
17. <tr>
18. <td>Alamat</td><td>:</td><td><textarea rows="3" cols="40"
19. name="valamat"></textarea></td>
20. </tr>
21. <tr>
22. <td colspan="3"><input type="submit" value="Proses" /><input type="button"
23. value="Cancel" /></td>
24. </tr>
25. </table>
26. </form>
27. </body>
28. <?php
29. if (isset ($_POST["vnim"]))
30. {
31.     $xnim=$_POST["vnim"];
32.     if (strlen($xnim)==0)
33.     {
34.         echo "NIM tidak boleh kosong";
35.     }
36.     else
37.     {
38.         echo "Proses Lanjut";
39.         if (isset ($_POST["vnama"]))
40.         {
41.             $xnama=$_POST["vnama"];
42.         }
43.         if (isset ($_POST["valamat"]))
44.         {
45.             $xalamat=$_POST["valamat"];
46.         }
47.     }
48. }
49. }
50. ?>
51. </html>

```

Pada script no.29 s/d 50, kita menuliskan kode PHP untuk melakukan validasi, yakni dengan menghitung jumlah karakter pada field NIM, jika tidak diisi atau jika panjang karakter field NIM adalah 0, maka muncul pesan : “NIM tidak boleh kosong” dan jika NIM ada isinya, maka proses aplikasi lanjut, dengan dicetak pesan : ”Proses Lanjut”.

## XII.4 Verifikasi

Data verification is a process in which different types of data are checked for accuracy and inconsistency after data migration is done ([https://en.wikipedia.org/wiki/Data\\_verification](https://en.wikipedia.org/wiki/Data_verification)).

Berdasarkan definisi tersebut, di dalam input data, verifikasi adalah memastikan bahwa data yang diinput adalah memang data yang benar, seperti inputan pada form login, yang meminta username dan password. Setelah form menerima inputan username dan password, maka form memproses data tersebut untuk dibandingkan, data inputan username dan password dengan data username dan password di database. Jika kedua data sama, maka user dapat masuk ke aplikasi.

Berikut adalah contoh aplikasi verifikasi Login, dengan verifikasi menggunakan PHP. Sehubungan dalam hal ini, membandingkan data inputan dengan data di database, maka kita tidak dapat menggunakan

JavaScript, karena JavaScript tidak dapat mengakses langsung database, kecuali JavaScript “versi” lain, yakni node.js.

Source code nya adalah sebagai berikut :

```
1.<!DOCTYPE html>
2.<html>
3.<head>
4. <title>Form Login</title>
5.</head>
6.<body>
7. <table align="center">
8.   <form method="POST" action="<?php echo $_SERVER['PHP_SELF'];?>">
9.     <caption>Login</caption>
10.    <tr>
11.      <td>Username</td><td>:</td>
12.      <td><input type="text" name="vusername" size="15" /></td>
13.    </tr>
14.    <tr>
15.      <td>Password</td><td>:</td>
16.      <td><input type="password" name="vpassword" size="15" /></td>
17.    </tr>
18.    <tr>
19.      <td colspan="3"><input type="submit" value="Proses" />
20.        <input type="button" value="Batal" /></td>
21.    </tr>
22.  </form>
23.</table>
24.<?php
25. $dsn='mysql:host=localhost;dbname=test';
26. $username='root';
27. $password='opansan';
28.
29. try {
30.     $dbh=new PDO($dsn,$username,$password);
31.   }
32.   catch (PDOException $e)
33.   {
34.     echo $e->getMessage();
35.   }
36.
37. if (isset($_POST["vusername"]))
38.   {
39.     $username=$_POST["vusername"];
40.   }
41.   else
42.   {
43.     $username="";
44.   }
45.
46. if (isset($_POST["vpassword"]))
47.   {
48.     $password=$_POST["vpassword"];
49.     $zpassword=md5($password);
50.   }
51.   else
52.   {
```

```

53.         $xpassword="";
54.         $zpassword="";
55.     }
56.
57.     $querynya="select count(*) from user where username='$username' and
58.         password='$zpassword'";
59.     $jumlah=$dbh->query($querynya)->fetchColumn();
60.     echo "<br/>";
61.     if ($jumlah>0)
62.     {
63.         echo "<center>Lanjut</center>";
64.     }
65.     else
66.     {
67.         echo "<center>Gagal</center>";
68.     }
69. ?>
70. </body>
71. </html>

```

Kita membutuhkan tabel “user” dengan struktur sebagai berikut :

No	Nama field	Tipe Data
1	username	varchar (30)
2	password	Varchar (100)

Alur program di atas, adalah, jika username dan password yang dimasukkan ada di salah satu record pada field username dan password di atas, maka tercetak kata “Lanjut”. Sebaliknya, jika tidak ada datanya, maka tercetak kata “Gagal”. Password yang dibandingkan adalah password yang sudah dengan format enkripsi md5.

Source code dapat anda peroleh pada Git di url, dengan keterangan awal “Bab 12” :  
[https://github.com/sofwanbl/modul\\_web\\_programming](https://github.com/sofwanbl/modul_web_programming)

## Bab XIII Pencarian

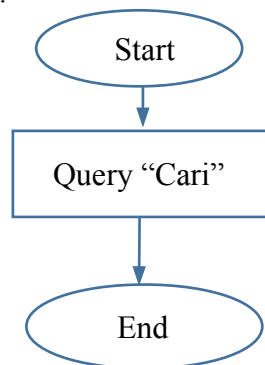
Pencarian atau Searching adalah proses menemukan sebuah nilai yang diberikan di dalam sebuah list nilai [1]. Di dalam pencarian data di database, nilai yang diberikan, diinput user pada sebuah text field, lalu dikirimkan nilainya untuk menjadi nilai untuk syntax "where" di dalam SQL. Contoh, pada program penyimpanan nilai di Bab XI (Membuat aplikasi CRUD sederhana), dicari nilai: 15, yang disimpan didalam sebuah variabel, misalkan "\$xnilai". Lalu, pada SQL, nilai 15 dimasukkan menjadi nilai yang dicari di field "nilai" pada tabel "contoh\_program". SQL nya akan menjadi seperti ini : "select \* from contoh\_program where nilai ='\$xnilai'", \$xnilai = 15.

Form yang kita gunakan untuk implementasi Searching adalah form Lihat Data pada program CRUD di Bab XI. Pertama-tama, kita akan buat text field searching dan tombol "Cari" nya. Setelah dimodifikasi, maka tampilan akan seperti ini :

No	Nilai	KARAKTER	KETERANGAN	Opsi
1	15	.....	Kecil	<a href="#">Edit</a>   <a href="#">Hapus</a>
2	4	....	Kecil	<a href="#">Edit</a>   <a href="#">Hapus</a>
3	0		Kecil	<a href="#">Edit</a>   <a href="#">Hapus</a>

Gambar XIII.1 Tabel Pencarian

Alur programnya adalah sebagai berikut :



Script lengkapnya adalah sebagai berikut:

```

1.<?php
2.include ("hubung_db.php");
3.include ("header.php");
4.??>
5. <form method="post" action="<?php $_SERVER['PHP_SELF'];?>" >
6. Cari : <input type="text" name="vcari" size="5" /><input type="submit" value="Cari" />
7. </form>
8. <br/>
9. <table border='1' class="tabel_tampil">
  
```

```

10. <th>No</th><th>Nilai</th><th>KARAKTER</th><th>KETERANGAN</th><th>Opsi</th>
11. <?php
12.     if (isset($_POST["vcari"]))
13.     {
14.         $xcari=$_POST["vcari"];
15.     }
16.     else
17.     {
18.         $xcari="";
19.     }
20.
21.     if (strlen($xcari)>0)
22.     {
23.         $katawhere=" where nilai='$xcari'";
24.     }
25.     else
26.     {
27.         $katawhere="";
28.     }
29. $query="select * from contoh_program". $katawhere;
30. $result=$dbh->query($query);
31. $no=1;
32. foreach($result as $row)
33.     {
34.         echo "<tr>";
35.         echo "<td>$no</td><td>".$row['nilai'].</td><td>".$row['karakter'].</td>";
36.         echo "<td>".$row['keterangan'].</td>";
37.         echo "<td><a href='edit.php?id=".$row['id']."'>Edit</a> |";
38.         echo "<a href='proses_hapus.php?id=".$row['id']."'>Hapus</a></td>";
39.         echo "</tr>";
40.         $no++;
41.     }
42.     ?>
43.</table>
44.</form>
45.</div>
46. <?php include ("footer.php"); ?>

```

### **Penjelasan :**

Pada source code 12- 29, adalah script untuk melakukan pencarian. Diawali dengan mengambil nilai “Cari” dari text field “Cari”.

```

12. if (isset($_POST["vcari"]))
13.     {
14.         $xcari=$_POST["vcari"];
15.     }
16.     else
17.     {
18.         $xcari="";
19.     }

```

Lalu memberi query Where, jika karakter “cari” diisi.

```

20. if (strlen($xcari)>0)
22.     {
23.         $katawhere=" where nilai='$xcari'";
24.     }

```

```
25. else
26. {
27.     $katawhere="";
28. }
```

Dan menambahkan query where ke dalam query pencarian.

```
29. $query="select * from contoh_program". $katawhere;
```

File lengkap beserta SQL create table nya, dapat anda peroleh di link :  
<https://goo.gl/BaPLhp>

## Bab XIV Pagination

Saat tampilan data sudah banyak, maka halaman webnya akan semakin panjang menampilkan data dan program akan bekerja keras untuk menampilkan data. User pun akan melihat tampilan data dengan menscroll ke bawah layar dengan cukup lama.

Hal ini dapat di atasi dengan membatasi jumlah tampilan data yang tampil pada setiap halamannya misalkan . Hal ini dikenal dengan nama **Pagination** atau **Paging**. Konsep Pagination, yaitu menampilkan data untuk jumlah tertentu pada setiap halaman dengan menggunakan limit di query. Kita akan menggunakan tampilan pada program CRUD yang sudah pernah kita buat sebelumnya. Hasil akhir dari halaman yang sudah diberi pagination adalah seperti di bawah ini :



Gambar XIV.1 : Pagination

Pada gambar di atas, kita memberikan link perpindahan halaman di bawah tabel, yang saat ini, jumlah total data di atas adalah 4 buah dan per halamannya, kita buat tampil per 2 record. Pembuatan session ini mengambil referensi dari halaman “Membuat Paging dengan PHP dan MySQL [2]”.

Alur program pagination adalah berikut :

1. Menentukan jumlah record yang muncul disetiap halaman. Pada contoh tersebut adalah, 2.
2. Menentukan nilai mulai halaman, dengan cara, jika halaman pertama yang tampil, maka nilai mulai adalah 0 dan jika halaman kedua atau selanjutnya yang tampil, maka nilai mulai adalah halaman  $-1 \times 2$  . 2 adalah jumlah record per halaman yang muncul.
3. Menghitung jumlah total record. Contoh di atas, jumlah total record = 4.
4. Menghitung jumlah total record dibagi jumlah record per halaman. Cotnoh di atas,  $4/3$ . Kita menggunakan fungsi `ceil()` untuk pembulatan ke atas, sehingga hasilnya adalah 2.  
Nilai 2 ini kita pergunakan untuk mencetak jumlah halaman di bawah tabel, karena ada 2, maka kita mencetak nilai 1 dan 2. Nilai ini merupakan link yang dapat diklik.

Program lengkapnya adalah sebagai berikut :

```
1.<?php
2.include ("hubung_db.php");
3.include ("header.php");
4.??>
5. <form method="post" action="<?php $_SERVER['PHP_SELF'];?>" >
6. Cari : <input type="text" name="vcari" size="5" /><input type="submit" value="Cari" />
7. </form>
8. <br/>
9. <table border='1' class="tabel_tampil">
10. <th>No</th><th>Nilai</th><th>KARAKTER</th><th>KETERANGAN</th><th>Opsi</th>
11. <?php
```



```

12.
13. if (isset($_POST["vcari"]))
14.     {
15.         $xcari=$_POST["vcari"];
16.     }
17. else
18.     {
19.         $xcari="";
20.     }
21.
22. if (strlen($xcari)>0)
23.     {
24.         $katawhere=" where nilai='$xcari'";
25.     }
26. else
27.     {
28.         $katawhere="";
29.     }
30.
31. // Cari Total //
32. $shalaman=2;
33. if (isset($_GET['halaman']))
34.     {
35.         $shal=(int)$_GET['halaman'];
36.     }
37. else
38.     {
39.         $shal=1;
40.     }
41.
42. if ($shal>1)
43.     {
44.         $mulai=($shal-1)*$shalaman;
45.     }
46. else
47.     {
48.         $mulai=0;
49.     }
50.
51. $query_total="Select count(*) from contoh_program". $katawhere;
52. $query_totalnya=$dbh->query($query_total);
53. $total=$query_totalnya->fetchColumn();
54. $shalamanya=ceil($total/$shalaman);
55. $query="select * from contoh_program". $katawhere ." limit $mulai, $shalaman";
56. $result=$dbh->query($query);
57.
58. $no=$mulai+1;
59. foreach($result as $row)
60.     {
61.         echo "<tr>";
62.         echo "<td>$no</td><td>". $row['nilai']. "</td><td>". $row['karakter']. "</td>";
63.         echo "<td>". $row['keterangan']. "</td>";
64.         echo "<td><a href='edit.php?id=" . $row['id']. "'>Edit</a> | <a href='proses_hapus.php?";
65.         echo "id=" . $row['id']. "'>Hapus</a></td>";
66.         echo "</tr>";
67.         $no++;
68.     }

```

```
69.     ?>
70. </table>
71. </form>
72. <br/>
73. <center>
74. <?php
75. for ($i=1;$i<=$halamannya;$i++)
76.   { ?>
77.     <a href="?halaman=?php echo $i;?"><?php echo $i;?></a>
78. <?php
79.     }
80. ?>
81. </center>
82. <br/>
83.</div>
84.<?php include ("footer.php"); ?>
```

Source code pada Bab ini tidak tersedia dan akan tersedia secara penuh setelah digabungkan dengan topik : session, yang merupakan topik pada Bab selanjutnya, Bab XV.

## Bab XV

### Session dan program CRUD

Pada bab XV ini, kita akan membahas mengenai Session di dalam PHP. Session adalah cara sederhana untuk menyimpan data untuk user individu dengan menggunakan nama session. Nama session dikirim ke dalam browser menggunakan session cookies. ID session ini juga dapat diambil. Untuk menggunakan session, pertama kali kita mengaktifkan session dengan perintah: `session_start()`. Selanjutnya kita dapat mengirim session dengan perintah `$_SESSION['nama_session']`. Session dapat dihapus dengan menggunakan perintah: `unset(session)`.

#### Contoh :

```
1.<?php
2. session_start();
3. $_SESSION['nama']='Opan';
4. echo $_SESSION['nama'];
5. unset($_SESSION['nama'];
6. echo $_SESSION['nama'];
7. ?>
```

Pada source code di atas, kita membuat session 'nama', mencetak session tersebut dan kemudian menghilangkan session tersebut, lalu dicetak kembali. Output program di atas, di komputer saya adalah sebagai berikut :

Opan

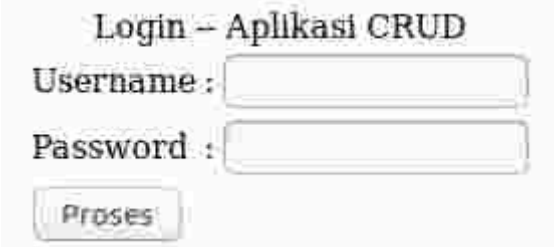
**Notice:** Undefined index: nama in `/var/www/html/contoh_session.php` on line 6

Kata "Opan", adalah nilai session "nama". Lalu session tersebut dihilangkan dan menjadi error, ketika dicetak kembali.

Session sering digunakan untuk Login dan Logout. Pada Login, biasanya kita diminta untuk memasukkan username dan password, yang jika username dan password tersebut valid, maka kita boleh memasuki halaman aplikasi. Session menyimpan login ini dan akan dihapus atau didestroy ketika user melakukan logout. Keberlakuan session memiliki waktu tertentu, yang jika waktu tersebut sudah habis, maka session secara otomatis akan dihapus.

Manfaat session di dalam Login adalah, ketika kita keluar program tanpa melakukan logout atau menghapus session, maka ketika masuk kembali ke program, maka akan bisa, tidak ditanyai username dan password di halaman Login. Dan ketika kita sudah melakukan Logout, maka kita tidak bisa masuk ke halaman aplikasi tanpa melakukan Login, termasuk mem by pass URL aplikasi.

Untuk menerapkan session di dalam aplikasi CRUD yang pernah kita buat sebelumnya, kita tambahkan halaman login, seperti di bawah ini :



The image shows a web form for logging into a CRUD application. The title is "Login - Aplikasi CRUD". There are two input fields: "Username :" and "Password :". Below the input fields is a button labeled "PROSES".

Gambar XV.1 Login

Source code Login tersebut adalah sebagai berikut :

```
1.<!DOCTYPE html>
2.<html>
3.<table id="tabel_entri" align="center">
4.<form method="post" action="proses.php">
5.<caption>Login -- Aplikasi CRUD</caption>
6.<tr>
7.<td>Username</td><td>:</td><td><input type="text" name="vusername" size="10" /></td>
8.</tr>
9.<tr>
10.<td>Password</td><td>:</td>
11.<td><input type="password" name="vpassword" size="10" /></td>
12.</tr>
13.<tr>
14.<td colspan="3"><input type="submit" value="Proses" /></td>
15.</tr>
16.</form>
17.</table>
18.</div>
19.<?php include("footer.php"); ?>
```

Setelah Login terbentuk, maka kita dapat menggunakan session untuk menyimpan nilai saat Login dan menghapus atau destroy session saat Logout. Pada awal file, kita perlu mengecek apakah user sudah login atau belum, untuk menghindari bypass ke dalam aplikasi.

Session dimulai dengan perintah `session_start()`. Untuk mengecek apakah session sudah dimulai atau belum, kita perlu mengecek dengan perintah :

```
if(session_status() == PHP_SESSION_NONE)
```

Jika session belum dimulai, kita perlu memberikan perintah `session_start()`.

Source code lengkap aplikasi CRUD dengan Login dan Logout dengan session, dapat anda peroleh dengan mendownload file : contoh\_aplikasi\_pdo\_full.zip di : [http://kuliah.sofwan.net/?page\\_id=264](http://kuliah.sofwan.net/?page_id=264) atau di [https://github.com/sofwanbl/modul\\_web\\_programming](https://github.com/sofwanbl/modul_web_programming)

## Penutup

Demikian lah tutorial pemrograman web dengan menggunakan PHP dan MySQL. Untuk penjelasan mengenai PHP lebih lanjut, anda dapat melihat dokumentasi dari situs resmi PHP, yakni : [www.php.net](http://www.php.net), dan untuk mengenal Mysql, anda dapat melihat ke [www.mysql.com](http://www.mysql.com). Untuk mengetahui lebih lanjut mengenai MariaDB, silahkan anda lihat ke [www.mariadb.org](http://www.mariadb.org).

## Referensi

- [1] PDO Functions, <http://www.nusphere.com/kb/phpmanual/ref.pdo.htm>
- [2] Muzanni, "Membuat Paging dengan PHP dan MySQL", <https://www.malasngoding.com/membuat-paging-dengan-php-dan-mysql/>, waktu akses : 18 Desember 2018
- [3] "Object Oriented PHP for Beginners", <https://code.tutsplus.com/tutorials/object-oriented-php-for-beginners—net-12762>
- [4] "php website", php.net
- [5] How does PHP work ?. <https://www.quora.com/How-does-PHP-work>, waktu akses : 09 Maret 2019

## Biografi Penulis

**Akhmad Sofwan** - Menyelesaikan pendidikan Sarjana di Teknik Informatika Universitas Budi Luhur ,tahun 2001 dan Magister Ilmu Komputer, Fakultas Ilmu Komputer ,Universitas Indonesia, tahun 2017. Penulis saat ini adalah seorang Freelance Software Developer. Selain itu, penulis juga mengajar mata kuliah ilmu komputer di Perguruan Tinggi Raharja, Tangerang dan di Program Vokasi, Universitas Indonesia. Penulis juga aktif di Pusat Kajian Biostatistik dan Informatika Kesehatan (PKBIK) Fakultas Kesehatan Masyarakat, Universitas Indonesia sebagai Software Developer dan Peneliti. Bidang penelitian beliau adalah Machine

Learning.