

# ***Pemrograman dasar PHP dan MYSQL***

**Ichwan Rizky Wahyudin**

**SMK Angkasa 1 Margahayu**

# Kata Pengantar

PHP merupakan bahasa pemrograman yang komplet. Olehkarena itu, butuh referensi yang lengkap pula untuk dapat mengetahui membuat aplikasi berbasis webiste.

Bahasan yang ada di dalam modul ini sangat mudah untuk dipahami dan dapat di praktikan menggunakan komputer biasa. Para programmer pemula akan mendapatkan manfaat lebih setelah membaca modul ini karena secara khusus, modul ini ditujukan bagi mereka yang ingin belajar.

Semoga dengan hadirnya modul ini, wawasan para pembaca akan bahasa pemrograman yang lebih baik serta cangguh dapat bertambah dan membantu membuat aplikasi website yang menarik.

Selamat membaca!

Bandung, 07 February 2021

“Ichwan Rizky Wahyudin”

# Modul I : Mengenal PHP

PHP Merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis website. Sebagai sebuah aplikasi, website tersebut hendaknya memiliki sifat dinamis dan interaktif. Memiliki sifat dinamis artinya, Website tersebut bisa berubah tampilan kontennya sesuai kondisi tertentu (misalnya, menampilkan produk yang berbeda-beda untuk setiap pengunjung). Interaktif artinya, website tersebut dapat memberi feedback bagi user (misalnya, menampilkan hasil pencarian produk).

Kode pemrograman standar untuk mendesain website sebenarnya adalah HTML, namun masalahnya tanpa memanfaatkan PHP, kode HTML tidak dapat digunakan untuk membuat website dengan dua sifat tampilan situs yang dilihat oleh para pengunjung satu dan lainnya relatif sama dan statics.

Celah inilah yang kemudian dapat diatasi oleh PHP. Dengan memanfaatkan PHP, Kamu bisa membuat sistem username login, shopping cart, database keanggotaan, portal, dan sebagainya.

## Instalasi Server untuk PHP

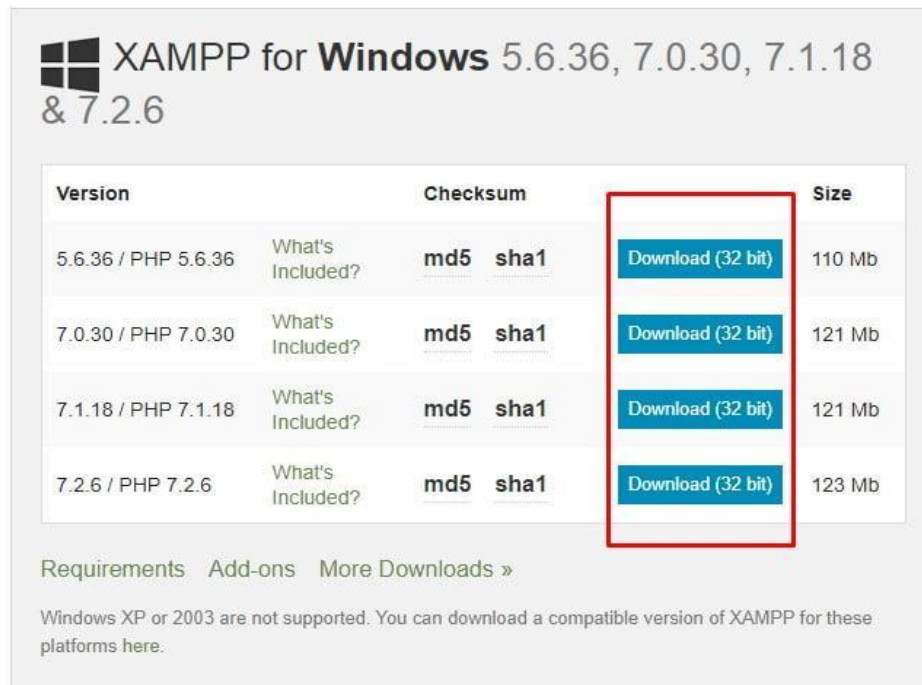
Dengan demikian, PHP akan diproses oleh server yang hasil olahannya akan dikirim kembali ke browser. Oleh karena itu, salah satu tool yang harus tersedia sebelum memulai programan PHP adalah server.

Berikut web server yang dapat digunakan, antara lain: *apache, IIS*

Salah satu cara membentuk web server di PC adalah menggunakan :

1. WAMP Server (<https://www.wampserver.com/en/download-wampserver-64bits/>).
2. XAMPP Server (<https://www.apachefriends.org/download.html>).

## Instalasi XAMPP untuk PHP



XAMPP for Windows 5.6.36, 7.0.30, 7.1.18 & 7.2.6

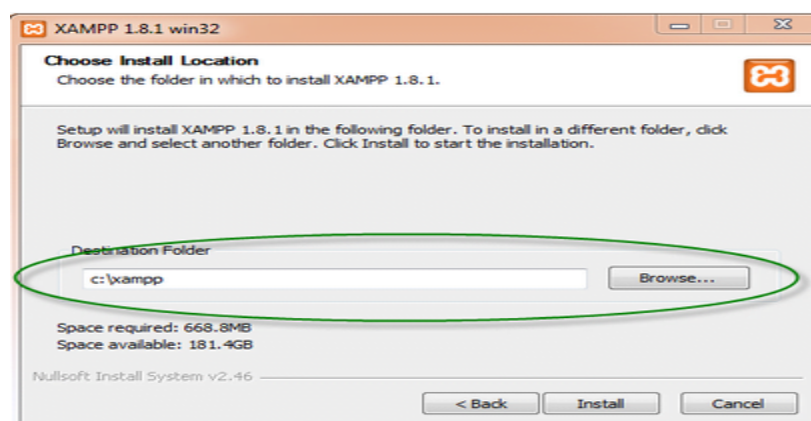
Version	Checksum	Size
5.6.36 / PHP 5.6.36	md5 sha1	110 Mb
7.0.30 / PHP 7.0.30	md5 sha1	121 Mb
7.1.18 / PHP 7.1.18	md5 sha1	121 Mb
7.2.6 / PHP 7.2.6	md5 sha1	123 Mb

Requirements Add-ons More Downloads »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

### Download XAMPP 7.2.6 yang telah mendukung PHP 7

Setelah proses pengunduhan selesai maka Kamu bisa memulai menginstall XAMPP. Tidak ada setting khusus yang perlu dilakukan pada saat instalasi kecuali pada pemilihan folder yang disarankan diletakan pada folder C:\xampp



Disarankan menginstall di folder C:\xampp

## Memilih Editor untuk PHP

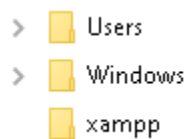
Php tidak membutuhkan editor khusus. Dengan begitu menulis kode-kode pemrograman PHP, yang Kamu butuhkan hanyalah Text Editor.

Berikut Text Editor Kamu bisa unduh salah satunya :

1. Visual Studio Code.
2. Notepad ++.
3. Sublime.
4. Ato.
5. Dan sebagainya.

## Membuat Folder untuk Project PHP

1. Buka Windows Explorer/File Explorer
2. Lalu, Buka folder C:\xampp\htdocs\.



### ***Membuka folder xampp dan htdocs***

3. Buat folder baru, misalnya **BelajarPHP**.

## Membuat Script PHP

Secara default, dokumen PHP memiliki ekstensi .php. Saat server web menemukan file dengan jenis ini, file tersebut secara otomatis dikirim untuk diproses oleh prosesor PHP.

Untuk menuliskan Script PHP :

*Untuk memulai pembuatan program berbasis PHP, Kamu bisa menggunakan tag berikut :*

```
<?php
```

*Seluruh isi dari PHP berada di dalam tag tersebut dan di tutup dengan tag di bawah ini.*

```
?>
```

Berikut ini adalah membuat program PHP “Hello World” sederhana.

1. Buka kode editor Kamu lalu *New File*
2. Simpan filenya di dalam folder **BelajarPHP** dengan nama ***index.php***.
3. Lalu ketik script PHP sederhana seperti ini.

```
<?php
```

```
    Echo “Hello World”;
```

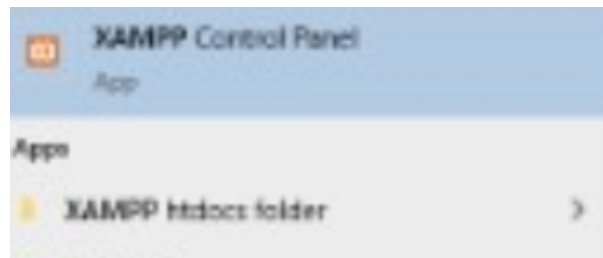
```
?>
```

4. Jika sudah jangan lupa untuk *Save* atau bisa dengan combine keyboard **CTRL + S**.

## Menjalankan Script PHP

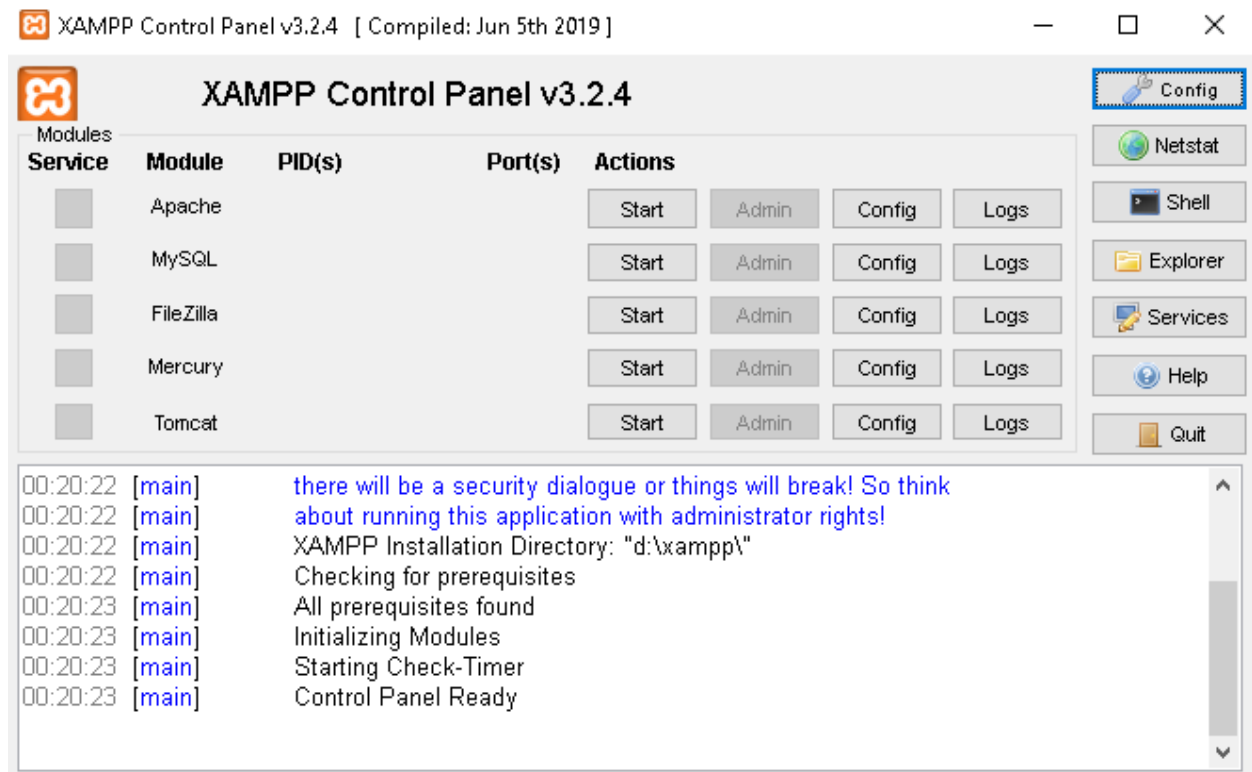
Berikut langkah-langkah untuk menjalankan script “Hello World” yang telah kita buat.

1. Klik Icon **Search** dan ketik **XAMPP**.
2. Lalu, pilih **XAMPP Control Panel**.



*Pilih XAMPP Control Panel*

3. Pada XAMPP Control Panel, tekan button Start pada pilihan Apache.



***Tekan button Start untuk memulai Apache***

4. Jika sudah, Buka Browser Kamu.
5. Lalu ketik di kotak Address <http://localhost/BelajarPHP/index.php>



# Modul II : Mengenal Variabel ,Tipe Data & Operator

## Tipe Data & Variabel

Variabel merupakan memori sementara untuk menyimpan data, baik String maupun numerik. Di dalam PHP sendiri.

### Contoh Variabel String

```
$nama ="Ichwan Rizky Wahyudin";
```

*Untuk melihat isi suatu variabel Kamu bisa melakukan seperti ini.*

```
echo "$nama";
```

### Contoh Variabel Numerik

Integer merupakan jenis tipe data bilangan bulat, dapat berupa bilangan bulat positif atau negatif, selain itu integer mendukung bilangan oktal, desimal dan hexadesimal.

```
$x = 5;
```

*Untuk melihat isi suatu variabel Kamu bisa melakukan seperti ini.*

```
echo "$x";
```

```
$y
```

```
$jumlah = $x + $y;
```

*Untuk melihat isi suatu variabel Kamu bisa melakukan seperti ini.*

```
Echo "$jumlah";
```

### Contoh Variabel Array

Array merupakan variabel yang dapat menyimpan beberapa value. Berikut contoh deklarasi variabel Array

```
$kelompok1 = array('Udin','Jajang','Somad');
```

```
echo $kelompok1[1]; // Menampilkan Jajang.
```

Statemen dalam kurung siku adalah 1 karena elemen pertama dari suatu array adalah elemen dengan indeks 0.

### Aturan Penamaan Variabel

Kamu bisa membuat variabel dengan nama apapun selama mengikuti aturan-aturan yang telah ditetapkan. Untuk membuat variabel dalam PHP, Kamu harus mengikuti aturan-aturan berikut:

1. Setiap variable php harus dimulai dengan \$ (tanda dollar).
2. setelah tanda \$, harus dimulai dengan garis bawah atau huruf.
3. Penamaan variable case Sensitive, \$data, \$Data, \$DATA adalah 3 variable yang berbeda.
4. Variable tidak boleh diawali dengan angka, contoh \$3data, \$1buah, ini contoh penamaan yang tidak valid.
5. variable hanya boleh mengandung karakter alpha-numeric dan underscores (A-z, 0-9, and \_).

## Operator Aritmatika

Berikut ini beberapa simbol-simbol matematika yang dapat di gunakan untuk melakukan perhitungan pada bahasa pemrograman php.

Simbol	Fungsi	Keterangan
+	Penjumlahan	Operator ini gunakan untuk penambahan dua operand, misalnya nilai $a + b$ ;
-	Pengurangan	Operator ini gunakan untuk pengurangan dua operand, misalnya nilai $a - b$ ;
*	Perkalian	Operator ini gunakan untuk melakukan perkalian dua operand, misalnya nilai $a * b$ ( $a \times b$ );
/	Pembagian	Operator ini gunakan untuk melakukan pembagian dua operand, misalnya nilai $a / b$ ;
%	Mod (Modulus)	Operator ini gunakan untuk menghasilkan nilai sisa pembagian, misalnya nilai $a \% b$ ;

## Operator Increment dan Decrement

**Penggunaan operator** jenis ini biasanya seringkali digunakan pada logika perulangan untuk dapat mengurutkan angka dan lain sebagainya, ada beberapa hal yang perlu dipahami dalam penggunaan masing-masing operator yaitu penempatan simbol pada bagian depan dan belakang adalah metode yang berbeda, misalnya  $\$a++$  dan  $++\$a$ ; untuk lebih jelasnya mari kita lihat tabel dibawah ini.

Operator	Nama	Keterangan
<code>\$a++</code>	Post-increment	Mengirimkan nilai <code>\$a</code> lalu ditambah 1
<code>++\$a</code>	Pre-increment	Dikurang 1 terlebih dahulu lalu mengirimnya ke variabel <code>\$a</code>
<code>\$a--</code>	Post-decrement	Mengirimkan nilai <code>\$a</code> lalu dikurang 1
<code>--\$a</code>	Pre-decrement	Dikurang 1 terlebih dahulu lalu mengirimnya ke variabel <code>\$a</code>

Dari tabel diatas terlihat perbedaan metode penambahan nilai atau pengurangan nilai terletak pada penempatan simbol operator tersebut, contohnya apabila pada variabel `$a` bernilai 1 dan akan ditampilkan di browser dengan menggunakan metode Post-increment maka nilai `$a` pertama yang muncul masih tetap angka 1, tetapi jika di echo dibaris kedua variabel `$a` telah berubah nilainya menjadi 2, itu karena post-increment mengirim nilai terlebih dahulu lalu di tambahkan, sama saja dengan metode-metode yang lain hanya prosesnya yang berbeda.

## Penulisan Operator Increment dan Decrement

Dari pada bingung dengan teori ini itu, langsung saja kita praktekan agar lebih paham maksud dan cara kerja operator ini, ketikan kode dibawah ini lalu simpan dan buka browser.

```
<?php
echo '<h2>Post-increment</h2>';
$a1 = 3;
echo 'Variabel $a1 bernilai 3 dan dengan menggunakan metode ($a1++)<br /><br />';
echo '1. $a1 = ' . $a1++ . ' (echo $a1++) nilai awal<br />';
echo '2. $a1 = ' . $a1 . ' (echo $a1) ditambahkan 1<br />';
```

```

echo '<h2>Pre-increment</h2>';
$a2 = 3;
echo 'Variabel $a2 bernilai 3 dan dengan menggunakan metode (++$a2)<br /><br />';
echo '1. $a2 = ' . ++$a2 . ' (echo ++$a2) nilai awal ditambah 1 terlebih dahulu<br />';
echo '2. $a2 = ' . $a2 . ' (echo $a2)<br />';

echo '<h2>Post-decrement</h2>';
$b1 = 3;
echo 'Variabel $b1 bernilai 3 dan dengan menggunakan metode ($b1--)<br /><br />';
echo '1. $b1 = ' . $b1-- . ' (echo $b1--) nilai awal<br />';
echo '2. $b1 = ' . $b1 . ' (echo $b1) dikurang 1<br />';

echo '<h2>Pre-increment</h2>';
$b2 = 3;
echo 'Variabel $b2 bernilai 3 dan dengan menggunakan metode (--$b2)<br /><br />';
echo '1. $b2 = ' . --$b2 . ' (echo --$b2) nilai awal dikurang 1 terlebih dahulu<br />';
echo '2. $b2 = ' . $b2 . ' (echo $b2)<br />';

?>

```

Contoh diatas dapat hanya mengulang dua variabel, operator tersebut akan terus melakukan proses sesuai variabel yang ditulis. Manfaat yang bisa diambil dari **operator increment dan decrement php** ini adalah apabila kita ingin membuat urutan angka misalnya 1 – 100, dengan menggunakan operator ini Kamu tidak perlu mengetik angka 1 – 100, secara otomatis setiap variabel dengan menggunakan operator ini akan terus ditambah sesuai jumlah variabel.

## Operator Perbandingan

Ada beberapa jenis operator perbandingan php yang dapat digunakan, pada dasarnya pada setiap bahasa pemrograman operator perbandingan tidak jauh berbeda sehingga jika kamu dapat menggunakan operator perbandingan di php maka akan dapat menggunakannya di bahasa lain, berikut ini adalah daftar jenis operator perbandingan php.

Operator	Nama	Contoh	Keterangan
==	Sama dengan	\$a == \$b	Jika \$a sama dengan \$b maka BENAR
===	Identik dengan	\$a === \$b	Jika \$a sama dengan (tipe data harus sama) \$b maka BENAR
!=	Tidak sama dengan	\$a != \$b	Jika \$a tidak identik dengan \$b maka BENAR
<	Kurang dari	\$a < \$b	Jika \$a lebih kecil dari \$b maka BENAR
>	Lebih dari	\$a > \$b	Jika \$a lebih besar dari \$b maka BENAR
<=	Kurang dari atau sama dengan	\$a <= \$b	Jika \$a kurang dari/sama dengan \$b maka BENAR
>=	Lebih dari atau sama dengan	\$a >= \$b	Jika \$a lebih besar dari/sama dengan \$b maka BENAR

Ada beberapa poin yang perlu diperhatikan dari operator perbandingan diatas, bahwa beberapa operator php mengharuskan kesesuaian tipe data yang sama "===" untuk menentukan suatu

kondisi dimana masing-masing variabel yang dibandingkan tersebut memiliki tipe yang sama, selain itu php memberikan akses apabila sebuah variable yang dibandingkan dengan variabel lain dengan tipe data yang sama itu tidak masalah.

## Praktek Penggunaan Operator Perbandingan

Setelah kita pahami tabel mari kita praktekan penggunaan operator perbandingan, berikut ini contoh penggunaan operator perbandingan pada php:

### 1. Sama dengan (==)

```
1<?php
2. $a = 10;
3. $b = 10;
4.
5. if($a == $b)
6. {
7.     echo 'Benar';
8. }
9. else
10. {
11.     echo 'Salah';
12. }
13.
14. ?>
```

Dari contoh penggalan kode diatas terlihat pada proses kondisi yang terjadi bahwa variabel \$a memiliki nilai yang sama dengan variabel \$b sehingga menghasilkan pernyataan **Benar**.

### 2. Identik dengan (===)

```
1<?php
2. $a = 10;
3. $b = 10;
4.
5. if($a === $b)
6. {
7.     echo 'Benar';
8. }
9. else
10. {
11.     echo 'Salah';
12. }
13.
14. ?>
```

Dari contoh penggalan kode diatas pada proses kondisi yang terjadi bahwa variabel \$a identik dengan variabel \$b sehingga menghasilkan nilai **Benar**, lalu apa bedanya dengan Sama dengan (==), perhatikan tipe data \$a dan \$b dimana masing-masing



menggunakan tipe data integer, sehingga menghasilkan nilai Benar, silahkan untuk mengganti salah satu jenis tipe data variabel diatas dengan string, dan pisahkan dua kondisi Sama dengan (==) dan Identik dengan (===) dengan menggunakan nilai sama dan menggunakan tipe data yang berbeda

### 3. Tidak sama dengan (!=)

```
1. <?php
2. $a = 10;
3. $b = 10;
4. if($a != $b)
5. {
6. echo 'Benar';
7. }
8. else
9. {
10. echo 'Salah';
11. }
12. ?>
```

Dari contoh kondisi perbandingan diatas menghasilkan nilai Salah, karena variabel \$a dan \$b memiliki nilai yang sama.

### 4. Tidak identik dengan (!==)

```
1. <?php
2. $a = 10;
3. $b = 10;
4. if($a !== $b)
5. {
6. echo 'Benar';
7. }
8. else
9. {
10. echo 'Salah';
11. }
12. ?>
```

Dari contoh operator perbandingan diatas menghasilkan nilai salah, karena variabel \$a dan \$b memiliki tipe data yang sama.

## 5. Kurang dari (<)

```
1. <?php
2. $a = 5;
3. $b = 10;
4. if($a < $b)
5. {
6. echo 'Benar';
7. }
8. else
9. {
10. echo 'Salah';
11. }
12. ?>
```

Dari kondisi di atas menghasilkan nilai benar, karena nilai yang terdapat pada variabel \$a lebih kecil dari yang terdapat pada variabel \$b.

## 6. Lebih Dari(>)

```
1. <?php
2. $a = 10;
3. $b = 5;
4. if($a > $b)
5. {
6. echo 'Benar';
7. }
8. else
9. {
10. echo 'Salah';
11. }
12. ?>
```

Dari contoh kondisi diatas menghasilkan nilai Benar, karena nilai pada variabel \$a lebih besar jumlahnya dari nilai yang terdapat pada variabel \$b.

## 7. Kurang dari atau sama dengan (<=)

```
1. <?php
2. $a = 5;
3. $b = 10;
4. if($a <= $b)
5. {
6. echo 'Benar';
7. }
8. else
9. {
10. echo 'Salah';
11. }
12. ?>
```

Dari kondisi diatas akan menghasilkan nilai Benar karena nilai pada variabel \$a lebih kecil dari nilai pada variabel \$b, meski jumlah variabel \$a sama dengan variabel \$b akan tetap menampilkan kondisi Benar, kecuali jika \$a sudah lebih dari variabel \$b

## 8. Lebih dari atau sama dengan (>=)

```
1. <?php
2. $a = 10;
3. $b = 5;
4. if($a >= $b)
5. {
6. echo 'Benar';
7. }
8. else
9. {
10. echo 'Salah';
11. }
12. ?>
```

Sama kasusnya pada poin nomor 7, hanya bedanya operator ini menyatakan nilai Lebih pada lawan kondisinya.

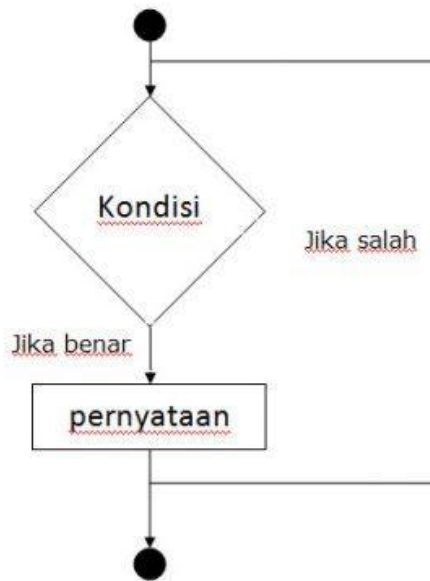
## Pengertian logika pada PHP

Dalam bahasa pemrograman manapun, logika IF ELSE ini tentu pasti selalu digunakan, lantas yang menjadi pertanyaan adalah apa bedanya dengan bahasa pemrograman lain, “tidak ada bedanya” hanya penulisannya saja berbeda. Lalu apa itu IF ELSE? IF ELSE adalah suatu struktur logika untuk membuat pencabangan alur program.

Contoh sederhanya adalah pada suatu pernyataan dalam dunia nyata, berikut ini:

Jika kamu bekerja keras, maka kamu akan berhasil!

Dari kutipan diatas termasuk logika pencabangan dalam bahasa pemrograman, bagaimana itu bisa terjadi? mari kita bahas lebih dalam.



### *Ilustrasi if else*

Dalam kondisi, “Jika kamu bekerja keras” maka “kamu akan berhasil!”, dalam kondisi pertama jika terpenuhi “bekerja keras” maka tentu akan menghasilkan pernyataan “berhasil!”, lalu bagaimana jika “tidak bekerja keras?” tentu kamu akan “gagal!”, kurang lebih sederhanya seperti itu, mari kita tuangkan dalam bahasa pemrograman sederhana.

*Contoh penulisan Logika IF ELSE pada PHP.*

```
1. $antrian = 3;
2. <?php if($antrian == 3){
3. // akan terpanggil
4. Echo "Terpanggil";
5. }else{
6. //belum terpanggil
7. Echo "belum terpanggil";
8. }
9. ?>
```

Dari penggalan kode diatas terlihat bahwa yang menjadi pertanyaan apakah nomor antrian yang kamu miliki nomor "3" atau nomor yang lain, parameter tentuk terletak pada variabel \$antrian, karna isi dari variabel \$antrian telah di ubah menjadi "4", maka kondisi pertama tidak terpenuhi, lalu akan melakukan lompatan ke proses selanjutnya dan menghiraukan proses sebelumnya karna tidak terpenuhi sehingga kan menampilkan pesan "Belum terpanggil"

## Penulisan Logika ELSE IF pada PHP

Setelah kita pahami materi diatas selanjutnya adalah pembahasan mengenai **ELSE IF pada PHP**. Loh, lalu apabedanya dengan IF ELSE? baiklah, IF ELSE hanya menampilkan satu kondisi dengan dua pernyataan yang berbeda, sedangkan ELSE IF mampu menampung lebih dari satu kondisi.

Maksudnya adalah apabila kondisi pertama tidak terpenuhi lalu program akan melakukan proses kondisi berikutnya, maka jika kondisi kedua juga tidak terpenuhi barulah melompat ke pernyataan "else" dimana program akan mengeksekusi yang paling akhir.

Contoh kode program **ELSE IF pada PHP**:

```
1. <?php
2. $rudi = 'wanita';

3. if($rudi == 'pria')
4. {
5. echo 'Rudi adalah seorang pria';
6. }
7. else if($rudi == 'wanita')
8. {
9. echo 'Rudi adalah seorang wanita';
10. }
11. else
```

```
12. {  
13. echo 'Hah? Lalu apa jenis kelamin rudi?';  
14. }  
15. ?>
```

## Penulisan Logika Switch Case Pada PHP

Seperti dikatakan bahwa switch case pada php mirip seperti penggunaan if else, yaitu untuk pencabangan dimana aplikasi memeriksa suatu kondisi variable atau hasil perhitungan ekspresi dan mengambil tindakan keputusan dengan perintah yang diberikan apabila memenuhi kondisi yang sesuai.

Berikut ini contoh penggunaan logika switch case:

```
1. <?php  
2. switch (variable)  
3. {  
4. case 'nilai1':  
5. # pernyataan...  
6. break;  
7. case 'nilai2':  
8. # pernyataan...  
9. break;  
10. default:  
11. # pernyataan...  
12. break;  
13. }  
14. ?>
```

Bagaimana cara switch case pada php bekerja? berikut ini penjelasan lebih lanjut mengenai bagaimana proses switch dapat mengeksekusi perintah yang dijalankan.

1. Pertama program akan memeriksa nilai dari variabel pada switch.
2. Selanjutnya program akan mengeksekusi masing-masing case apakah sesuai dengan nilai variabel yang diberikan.
3. Apabila nilai variabel terdapat pada salah satu case maka proses eksekusi akan dihentikan dengan perintah break.
4. Apabila nilai variabel tidak terdapat pada salah satu case maka proses akan berakhir pada perintah default.

Berikut ini contoh penulisan logika switch case pada PHP:

```
1. <?php  
2. $rudi = 'pria';  
3. switch ($rudi)
```

```
4. {
5. case 'pria':
6. echo 'Rudi adalah seorang pria';
7. break;
8. case 'wanita':
9. echo 'Oh ternyata rudi adalah seorang wanita';
10. break;
11. default:
12. echo 'Perkenalkan nama saya rudi';
13. break;
14. } ?>
```

Jika dilihat dari kode diatas maka seperti pada penjelasan sebelumnya, program akan memeriksa nilai yang diberikan pada variabel \$rudi, lalu program akan mengeksekusi masing-masing case, apakah memenuhi nilai yang sudah ditentukan, jika ya maka pernyataan akan di proses, sedangkan jika variabel tidak memiliki nilai yang sesuai pada masing-masing case maka program akan mengeksekusi perintah default, atau lebih mirip else pada if else.

## Struktur Logika Perulangan Pada PHP

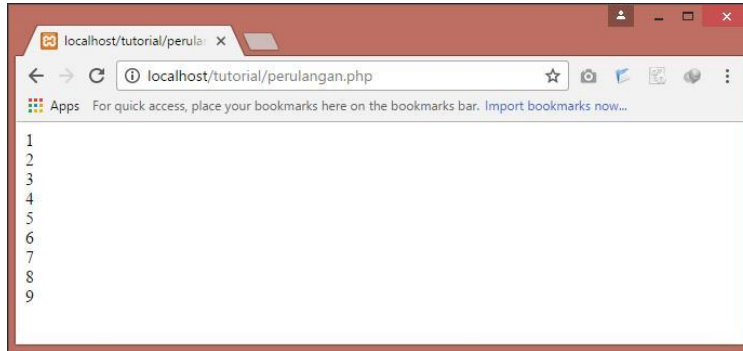
Seperti namanya perulangan atau looping yaitu digunakan untuk mempermudah melakukan proses yang berulang-ulang sesuai dengan perintahnya. Contoh sederhananya adalah, apabila kita mengurutkan angka 1 sampai 100, maka hal yang paling mudah dalam pemrograman ialah dengan menggunakan perulangan, yaitu dengan Algoritma menambahkan nilai 1 pada setiap angka yang diulang. Misalnya dimulai dari angka 1, ditambah dengan 1 maka menghasilkan angka 2, dan seterusnya hingga mencapai 100. Bagaimana jika tidak dengan perulangan? tentunya menyita waktu, tenaga dan kerepotan lainnya :p

Seperti awal pembahasan bahwa ada beberapa jenis metode perulangan atau looping pada php yang bisa digunakan, mari kita kupas satu persatu dengan menggunakan bahasa pemrograman PHP.

## Struktur Logika Perulangan Pada PHP

Perulangan for digunakan untuk mengulang suatu pernyataan sebanyak yang kita mau. Penggunaan perulangan for pada php yaitu dengan menentukan kondisi jumlah atau nilai yang ingin diulang “variabel = nilai; variabel < batas; variabel++”, contoh sederhananya:

```
1. <?php
2. for($i = 1; $i <= 9; $i++)
3. {
4. echo $i . '<br />';
5. }?>
```



***Hasil perulangan for jika di jalankan***



## Penulisan Perulangan While Pada PHP

Perulangan while mirip dengan for, namun untuk menentukan kondisinya hanya dengan menggunakan operator perbandingan saja, misalnya `$i <= batas`, contoh sederhananya:

```
1. <?php
2. $i = 1;
3. while($i <= 9)
4. {
5. echo $i . '<br />';
6. $i++;
7. }
8. ?>
```

Untuk outputnya kurang lebih sama dengan penjelasan perulangan sebelumnya, yaitu perulangan dengan menggunakan for. Dimana browser akan menampilkan angka 1 – 9, perbedaan yang mencolok dengan menggunakan for adalah kondisi tidak ditulis didalam fungsi perulangannya, dan proses penambahan angka dengan menggunakan metode post increment ditulis di dalam proses perulangan.

## Penulisan Perulangan Foreach Pada PHP

Jenis perulangan Foreach pada PHP ini adalah yang paling berbeda dari perulangan sebelumnya, yaitu perulangan Foreach pada PHP ini berguna untuk memecah isi array, atau lebih sederhananya ialah untuk menyederhanakan nilai array agar dapat dibaca dengan mudah, berikut ini adalah contoh pseudo-code nya:

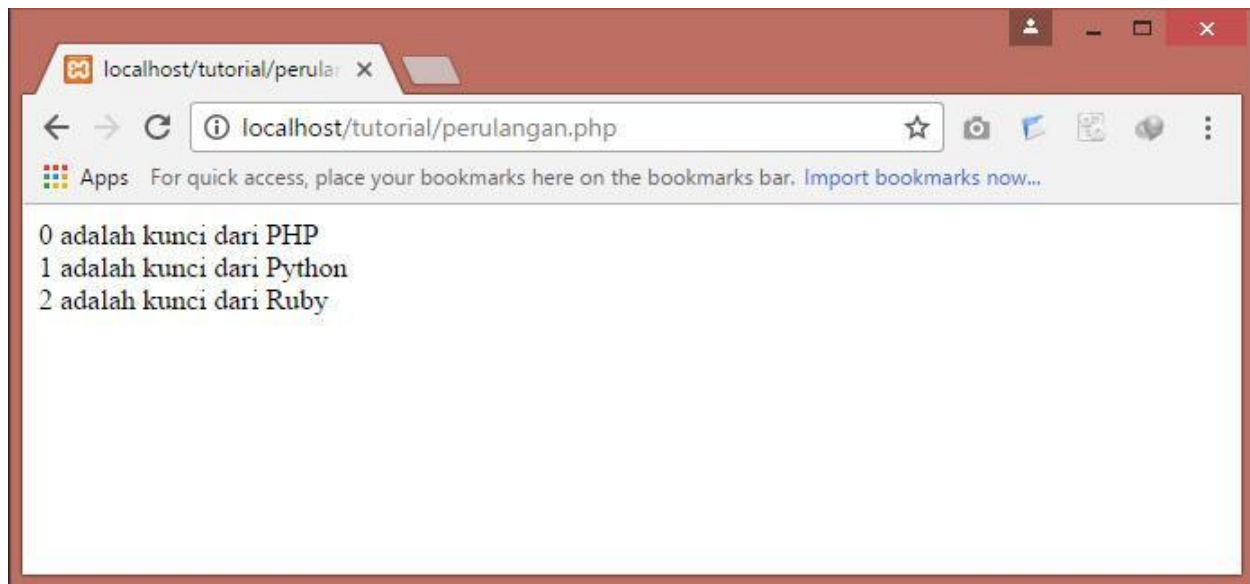
```
foreach(Array as Key => Value)
{
    # Code...
}
```

Berikut ini untuk lebih jelasnya langsung saja menulis program dengan menggunakan Foreach pada PHP.

Contoh Praktek Perulangan Foreach:

```
1. <?php
2. $array = array('PHP', 'Python', 'Ruby');
3. foreach($array as $key => $value) {
4. echo $key . ' adalah kunci dari '. $value . '<br />';
5. }
6. ?>
```

Terlihat bahwa variabel \$key digunakan untuk menampilkan kunci dari array tersebut, berbeda dengan yang sebelumnya yaitu hanya dengan menampilkan nilainya saja. Contoh sederhana diatas maka akan menampilkan output seperti gambar dibawah ini:



***Hasil perulangan foreach jika di jalankan***

## Modul III : Dasar-Dasar MYSQL.

Dalam bahasa SQL pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi terdiri dari baris (row atau record) dan kolom(column atau field). Sedangkan dalam sebuah database dapat terdiri dari beberapa table.

Beberapa tipe data dalam MySQL yang sering dipakai:

Tipe Data	Keterangan
INT(M) [UNSIGNED]	Angka -2147483648 s/d 2147483647
FLOAT(M,D)	Angka pecahan
DATE	Tanggal Tanggal Format : YYYY-MM-DD
DATETIME	Tanggal dan Waktu Format : YYYY-MM-DD HH:MM:SS
CHAR(M)	String dengan panjang tetap sesuai dengan yang ditentukan. Panjangnya 1-255 karakter
VARCHAR(M)	String dengan panjang yang berubah-ubah sesuai dengan yang disimpan saat itu. Panjangnya 1 – 255 karakter
BLOB	Teks dengan panjang maksimum 65535 karakter
LOB	Teks dengan panjang maksimum 4294967295 karakter

### Membuat Database & Table

Untuk membuat database dan table langkah yang harus kamu praktekan dengan menggunakan Command Prompt :

1. Buka Command Prompt dan buka folder xampp/mysql di command prompt jalankan perintah berikut ini :

**cd c:\xampp\mysql** // lalu tekan enter.

2. Jalankan perintah agar masuk ke MYSQL berikut perintahnya :

**mysql.exe -u root** //lalu tekan enter

3. Berikut untuk membuat sebuah database baru adalah dengan perintah :

**Create database namadatabase;** // harus di akhiri dengan titik koma(;) untuk mengakhiri perintah tersebut

Contoh :

**Create database db\_ichwan;**

4. Untuk membuka sebuah database dapat menggunakan perintah berikut ini:

**use namadatabase;**

Berikut Contoh untuk membuka database yang kita buat tadi :

**use db\_ichwan;**

5. Buat table dengan nama users, Berikut perintah untuk membuat Table :

**Create table namatable(struktur);**

Kolom/Field	Tipe data	Keterangan
id	Int, primary key,AUTO INCREMENT	angka, sebagai primary key,Field bisa di kosongkan karena akan di isi dengan sendirinya
username	Char(36) not null	teks dengan panjang maksimal 36 karakter, tidak boleh kosong
password	Varchar(255) not null	teks dengan panjang maksimal 255 karakter, tidak boleh kosong

nama	Varchar (60) not null	teks dengan panjang maksimal 40 karakter, tidak boleh kosong
email	Char(255) not null	teks dengan panjang maksimal 255 karakter, tidak boleh kosong

Berikut perintah MYSQL untuk membuat table seperti diatas :

<p><b>Create table users</b></p> <p><b>(id int primary key AUTO_INCREMENT,</b></p> <p><b>Username char(36) not null,</b></p> <p><b>Password varchar(255) not null,</b></p> <p><b>Nama varchar(60) not null,</b></p> <p><b>Email char(255) not null);</b></p>
--

Sedangkan data yang akan diisikan dalam tabel users adalah sebagai berikut:

Id	Username	Password	Nama	Email
1	Irizkyw	123456	Ichwan Rizky	Ichwanrizkywahyudin3@gmail.com
2	Ujang	654321	Ujang Jejer	UjangJejer@gmail.com

Untuk memasukan sebuah record kedalam table MySQL adalah sebagai berikut :

**Insert into namatable values(kolom1,kolom2,kolom3,..);**

Contoh :

```
Insert into users values('1','Irizkyw','123456','Ichwan  
Rizky','ichwanrizkywahyudin3@gmail.com');
```

## Menampilkan ISI TABLE

Isi table dapat ditampilkan dengan menggunakan perintah **SELECT**, cara penulisan perintah **SELECT** adalah :

```
SELECT * FROM namatable;
```

Contoh menampilkan seluruh isi table **users** :

```
SELECT * FROM users;
```

Untuk menampilkan username & nama saja , Berikut perintah nya:

```
SELECT username,nama FROM users;
```

# Modul IV : Menghubungkan PHP dengan MySQL

Agar script PHP yang kita buat, mari buat system login yang dapat berhubungan dengan database dari MySQL Berikut langkah-langkahnya:

1. Buat folder di **xampp/htdocs** dengan nama **Login**
2. Lalu buat file-file seperti di bawah ini :

**File koneksi.php:**

```
<?php

$koneksi = mysqli_connect('localhost','root','','users') or die ("Database tidak dapat
dihubungkan");

?>
```

**File index.php:**

```
<?php
require 'koneksi.php';
if(isset($_POST['login'])){
$username = $_POST['username'];
$password = $_POST['password'];

$query=mysqli_query($koneksi,"SELECT * FROM users WHERE username='$username'");
if(mysqli_num_rows($query) > 0){

$result = mysqli_fetch_assoc($query);

if(password_verify($password,$result['password'])){
    $_SESSION['login'] = true;
    header('location: home.php');
} else{
    echo"gagal Login";
}
}
?>
<!DOCTYPE html>
<html lang="en">
```

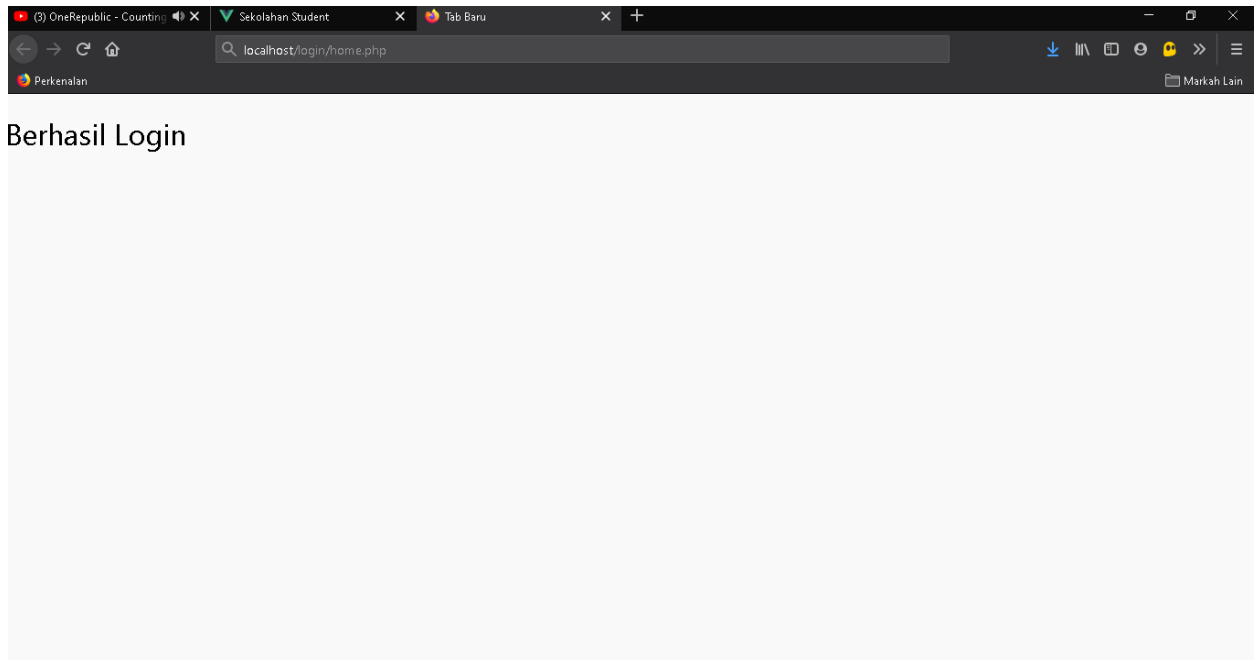
```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login</title>
</head>
<body>
  <form action="" method="post">
    Username
    <input type="text" name="username"> <br>
    Password
    <input type="text" name="password">
    <button type="submit" name="login">Login</button>
  </form>
</body>
</html>
```

**File home.php:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Berhasil Login</h1>
</body>
</html>
```



Jika berhasil login Tampilan nya seperti ini:



***Berhasil login***