

# BAB 1

## PENGENALAN

### PEMROGRAMAN WEB

Dalam memilih atau membuat sebuah program, sering kali Anda dihadapkan pada pilihan, apakah itu berbasis web atau desktop. Tepat sekali jika Anda memilih yang berbasis web! Aplikasi yang berbasis web merupakan aplikasi yang mudah diakses oleh pengguna dengan perangkat apa pun yang dipunyai, lalu yang penting terdapat browser di dalamnya. Lewat browser itulah suatu aplikasi web dapat dengan leluasa dapat diakses. Berbeda dengan aplikasi berbasis desktop, karena Anda harus melakukan instalasi program secara lokal dengan berbagai setting yang terlihat rumit. Belum lagi untuk masalah sistem operasinya, Windows-kah? Linux-kah?.

Sedangkan untuk membuat aplikasi berbasis web sangat mudah, cukup dengan mengerti bahasa pemrograman dasar web seperti HTML, PHP, dan sedikit pengetahuan database MySQL.

HTML adalah kependekan dari *Hyper Text Markup Language* yang biasanya digunakan untuk menentukan tata tampilan web ataupun informasi statis. HTML dapat dikombinasikan dengan bahasa pemrograman lain seperti PHP ataupun Javascript. HTML dapat diakses secara langsung pada browser tanpa menggunakan server. Berbeda dengan PHP, bahasa pemrograman ini memerlukan server yang dapat menerjemahkan ke dalam browser. Dalam buku ini, kita akan belajar tentang PHP dan database karena kebutuhan yang diperlukan adalah untuk membuat aplikasi atau web dinamis.

PHP adalah kependekan dari *PHP-Hypertext Pre-processor*. PHP merupakan suatu bahasa pemrograman berbasis web yang menyatu dengan HTML dan dijalankan oleh server side.

Mengapa dikatakan server side?

Karena PHP tidak bisa langsung dibaca oleh browser, tapi membutuhkan sebuah web server yang mengolah permintaan browser terhadap suatu halaman.

Teknisnya adalah sebuah server bertugas mengolah file PHP, lalu melakukan konversi menjadi HTML untuk dibaca sebuah browser. Dari hasil konversi tersebut barulah browser dapat membaca atau menampilkan hasil.

Apakah HTML saja tidak cukup?

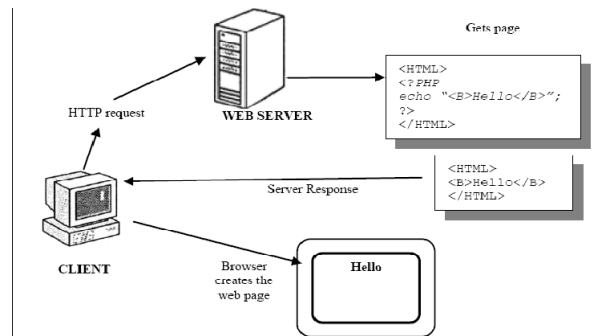
Ya, tentu saja HTML tidak cukup, salah satunya adalah karena sifat HTML yang statis. HTML hanya berguna untuk desain saat kita browsing, maka yang kita lihat sebenarnya adalah interpretasi dari HTML. Sebagai contoh, jika kita mengisi form di dalam web, maka untuk tampilan isian adalah menggunakan HTML, tetapi jika data yang diolah kemudian disimpan dalam database misalnya, tidaklah cukup menggunakan HTML.

Ketika ingin menjalankan web berbasis php, maka terlebih dahulu harus mengaktifkan web server. Tidak perlu langsung terkoneksi internet, cukup menguji dengan komputer kita sendiri yang dijadikan server. Web server yang dapat digunakan misalnya Apache, IIS, Xitami, Nginx, ataupun GWS.

Demi kepraktisan, Anda bisa juga menggunakan sebuah software yang menggabungkan paket program-program lain, termasuk Apache dan MySQL. Sehingga dengan sekali install, kita sudah mendapatkan semuanya. Praktis kan? Contohnya:

- XAMPP ([www.apachefriends.org](http://www.apachefriends.org)).
- EasyPhp ([www.easyphp.org](http://www.easyphp.org)).
- Appserv ([www.appservnetwork.com](http://www.appservnetwork.com)).
- WampServer ([www.wampserver.com](http://www.wampserver.com)).
- Bitnami ([www.bitnami.org](http://www.bitnami.org)).

Selain yang telah disebutkan, masih banyak lagi paket lain yang menyediakan web server. Dalam hal ini, php akan menjadi server side embedded script language.



**Gambar 1.1 Skema hubungan Web Browser – Web Server – PHP**

Seperti Gambar 1.1, webserver akan bekerja bila ada permintaan dari client. Client memakai kode-kode php untuk mengirimkan permintaan ke server. Ketika menggunakan PHP sebagai server side embedded script language, maka server akan melakukan hal-hal sebagai berikut:

- Membaca permintaan dari client/browser.
- Mencari halaman/page di server.
- Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/page.

Saat ini, PHP menjadi bahasa pemrograman favorit di bidang web. Selain kemudahan, juga karena kepraktisan implementasinya. Dalam pemrograman PHP dapat pula menggunakan database, sehingga akses penyimpanan data bisa dengan mudah dilakukan. Web bisa diakses juga lewat media internet maupun intranet asal ada terdapat browser, maka Anda dapat mengakses programnya.

## 1.1 Perbedaan Web Statis dan Dinamis

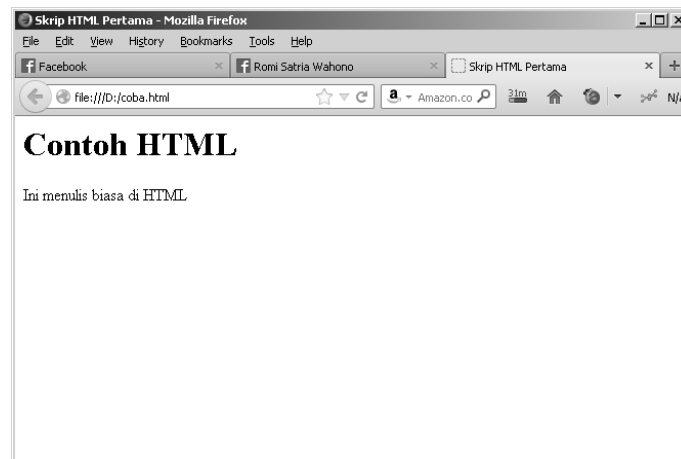
Sebuah Web, menurut cara kerjanya dapat dibedakan menjadi 2 bagian, yaitu bekerja secara statis dan dinamis. Web dikatakan statis jika perubahan tampilan halaman yang diakses harus dilakukan dengan mengubah script langsung pada halaman tersebut. Suatu halaman web dikatakan dinamis jika perubahan tampilan halaman yang diakses dilakukan tanpa mengubah script, tetapi hanya menambah atau mengubah inputan pada halaman tersebut. Oya, statis di sini bukan diartikan sebagai halaman web yang selalu bergerak-gerak seperti layaknya animasi dan sebagainya.

## 1.2 Contoh Script HTML

Penulisan HTML untuk keperluan web statis seperti script berikut:

```
<html>
<head>
<title>Skrip HTML Pertama</title>
</head>
<body>
<h1>Contoh HTML</h1>
Ini menulis biasa di HTML
</body>
</html>
```

Hasil program setelah diakses adalah seperti pada Gambar 1.2.



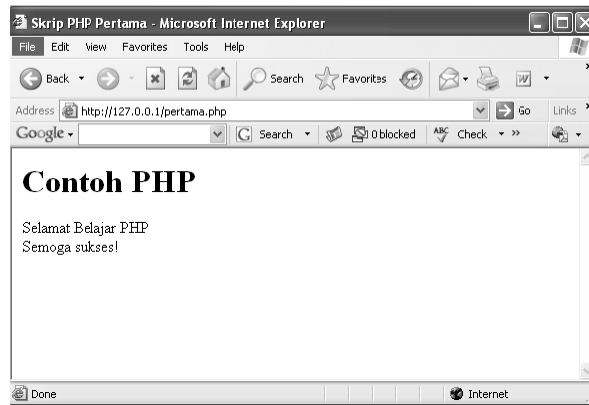
*Gambar 1.2 Contoh tampilan program HTML lewat browser*

## 1.3 Contoh Script PHP

Contoh penulisan script php seperti berikut:

```
<html>
<head>
<title>Skrip PHP Pertama</title>
</head>
<body>
<h1>Contoh PHP</h1>
<?php
    print("Selamat Belajar PHP<br>");
    print("Semoga sukses!");
?>
</body>
</html>
```

Hasilnya seperti pada Gambar 1.3.



*Gambar 1.3 Contoh tampilan program PHP lewat browser*

Penyimpanan kode PHP menggunakan format ASCII, sehingga kode PHP dapat ditulis di hampir semua plain-text yang artinya dapat menggunakan Notepad, Wordpad, Microsoft Word, Editplus, Notepad++, dan lain sebagainya untuk menulis script. Kode PHP adalah kode yang disertakan di sebuah halaman HTML dan kode tersebut dijalankan oleh server sebelum dikirim ke browser.

## 1.4 Tag di PHP

Tag dapat diartikan sebagai tanda untuk memulai dan mengakhiri penulisan sebuah script PHP atau dapat juga digunakan untuk membatasi daerah script mana yang menggunakan script PHP.

### 1.4.1 Cara Pertama

```
<?php //menandai awal tag
...
?> //menandai akhir tag
```

### 1.4.2 Cara Kedua

```
<? //menandai awal tag
...
?> //menandai akhir tag
```

Option short tag di web server sudah diaktifkan atau bernilai on.

### 1.4.3 Cara Ketiga

```
<% //menandai awal tag  
...  
%> //menandai akhir tag
```

Cara ini sama dengan tag pada ASP. Opsi ini bisa dilakukan jika nilai `asp_tags` pada `php.ini` bernilai `on`.

### 1.4.4 Cara Keempat

```
<script language="php"> //menandai awal script  
...  
</script> //menandai akhir script
```

## 1.5 Komentar di PHP

Komentar digunakan untuk menandai program atau membuat catatan di dalam program. Penulisan komentar di dalam program tidak akan ikut dieksekusi sebagai perintah. Komentar dalam php ada 3 macam cara penulisannya:

### 1.5.1 /\* komentar \*/

Tulisan apa pun yang berada di antara `/*` dan `*/` akan dianggap sebagai komentar. Cara seperti ini sangat berguna dan efisien untuk pemberian komentar yang memakan banyak baris.

### 1.5.2 // komentar

Tulisan di baris yang sama setelah `//` akan dianggap sebagai komentar. Cara ini berguna untuk pemberian komentar singkat yang tak lebih dari 1 baris saja.

### 1.5.3 # komentar

Sama seperti `//`, tulisan di baris yang sama setelah `#` akan dianggap sebagai komentar. Cara ini berguna untuk pemberian komentar singkat yang tak lebih dari 1 baris saja.

## 1.6 Penamaan File

File PHP harus disimpan dengan ekstensi `.php`. Jika menemukan file dengan ekstensi `.php3` atau `.phtml`, maka kemungkinan besar file-file tersebut ditulis menggunakan PHP versi 3 ke bawah. Seperti dijelaskan sebelumnya, file-file tersebut akan disimpan sebagai file teks biasa. Artinya, tidak butuh editor khusus bahkan cukup dengan notepad. Namun, perlu diingat bahwa notepad tidak memunculkan nomor baris (line) yang akan kita perlukan saat melakukan pengecekan (debugging) kode jika suatu saat ada masalah dengan script.

## 1.7 Aturan yang Lain

Tanda kurung akan banyak Anda gunakan dalam kode PHP. Salah satu penggunaan yang sering dilakukan adalah dalam memanggil fungsi. Secara sederhana, setiap fungsi PHP akan berbentuk seperti berikut:

```
print ( );
```

"print" adalah nama fungsi, lalu informasi yang perlu ditambahkan pada fungsi tersebut akan Anda tuliskan dalam tanda kurung. Jangan lupa untuk mengakhiri dengan tanda titik koma, kesalahan tidak mengakhiri dengan titik koma ini paling sering terjadi pada pemula. `Echo()` juga memiliki kegunaan yang sama dengan `print()`. Spasi, pergantian baris, dan lainnya tidak akan mempengaruhi output. Sehingga, bagian kode berikut ini:

```
<?php  
print ("Anda berada di situs MIPA_UNSI!");  
?>
```

akan menghasilkan hal yang sama dengan kode berikut ini:

```
<?php print ("Anda berada di situs MIPA_UNSI!"); ?>
```

Aturan mendasar tentang penulisan variabel di PHP, yaitu variabel ditulis dengan diawali oleh tanda '\$', sehingga dapat dibedakan secara lebih mudah mana yang merupakan perintah variabel dan bukan variabel. Contoh variabel `$nama`, `$luas segi3`, dan lainnya.

## 1.8 Mengapa Menggunakan PHP

Banyak sekali kelebihan yang dimiliki PHP bila dibandingkan dengan bahasa pemrograman yang lain, di antaranya:

- 1 Bisa membuat Web menjadi Dinamis.
- 2 PHP bersifat Open Source yang berarti dapat digunakan oleh siapa saja secara gratis.
- 3 Program yang dibuat dengan PHP bisa dijalankan oleh semua sistem operasi karena PHP berjalan secara *web base* yang artinya semua sistem operasi bahkan gadget mobile apa pun asalkan mempunyai browser dapat menikmati program PHP.
- 4 Aplikasi PHP lebih cepat dibanding dengan ASP maupun Java.
- 5 Mendukung banyak paket Database seperti MySQL, Oracle, PostgreSQL, dan lain-lain.
- 6 Bahasa pemrograman PHP tidak memerlukan *compile* dalam penggunaannya.
- 7 Banyak Web Server yang mendukung PHP seperti Apache, Lighttpd, IIS, dan lain-lain.
- 8 Pengembangan aplikasi PHP sangat cepat karena banyaknya dokumentasi, referensi, dan antar sesama *developer* yang turut membantu dalam pengembangannya.
- 9 Banyak bertebaran aplikasi dan program PHP yang free, serta siap digunakan atau dikembangkan sendiri seperti WordPress, Joomla, Opencart, PrestaShop, Drupal, Magento, dan lain-lain.

Selain kelebihannya, sampai detik ini ternyata PHP juga mempunyai kekurangan. Di antaranya:

1. PHP tidak mengenal Package. Jika tidak di-encoding (dikunci), maka kode PHP dapat dibaca semua orang. Hal ini sudah bisa diatasi dengan tool-tool encoding untuk mengamankan script.
2. PHP memiliki celah kelemahan keamanan. Jadi sang programmer harus jeli dan berhati-hati dalam melakukan setting dan konfigurasi PHP.



## 1.9 Paket Software yang Digunakan

Seperti yang sudah dijelaskan sebelumnya, sebelum bekerja dengan PHP kita perlu mengunduh paket php-mysql, misalnya Easy-PHP, XAMPP, Appserv yang di dalamnya sudah terdapat 2 server, yaitu web server apache dan database server MySQL yang biasanya juga ditambah tool untuk mengakses database, yaitu phpmyadmin.

Apache digunakan untuk webserver-nya, sedangkan MySQL digunakan untuk server database-nya. Selebihnya hanya tools saja yang dapat kita gunakan.

Aplikasi yang lain seperti untuk mengolah database kita bisa menggunakan Navicat, untuk editornya kita bisa menggunakan Dreamweaver atau Notepad++.



*Gambar 1.3 Paket Software yang dapat digunakan untuk instalasi program*

Paket yang akan kita gunakan adalah paket XAMPP (lambang seperti pada Gambar 1.3) dibandingkan dengan paket yang lain paket ini yang paling komplit dikarenakan terdapat tool-tool yang memudahkan dalam pembangunan aplikasi berbasis web. Kita dapat memperoleh XAMPP secara gratis dengan mengunduhnya di halaman resmi mereka di [www.apachefriends.org](http://www.apachefriends.org) atau di dalam CD/DVD yang disertakan dalam buku ini.



# BAB 2

## INSTALASI SERVER

Seperti yang sudah dijelaskan pada bab sebelumnya bahwa untuk menjalankan aplikasi web dinamis yang berbasis PHP diperlukan webserver. Di mana tugas web server tersebut adalah menerjemahkan kode-kode untuk dapat dibaca browser. Selain server web, diperlukan juga server database karena aplikasi yang kita buat memang menggunakan database.

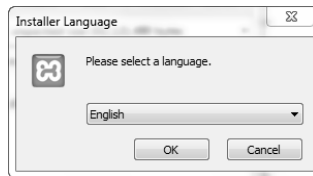
### 2.1 Paket XAMPP

XAMPP merupakan program paket PHP dan MySQL berbasis *opensource* yang saat ini merupakan andalan para programmer PHP dalam melakukan programming dan melakukan testing hasil programnya.

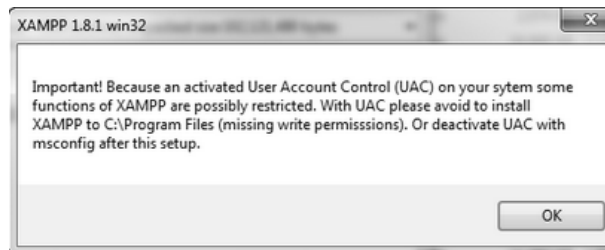
Berikut adalah langkah-langkah melakukan instalasi:

File installer bisa ditemukan pada CD buku ini. Kemudian, klik ganda dari file master *xampp-win32-1.8.1a-installer.exe*. Atau Anda bisa download dari website <http://www.apachefriends.org>. Dari lokasi download, kita bisa memilih sesuai sistem operasi yang kita pergunakan.

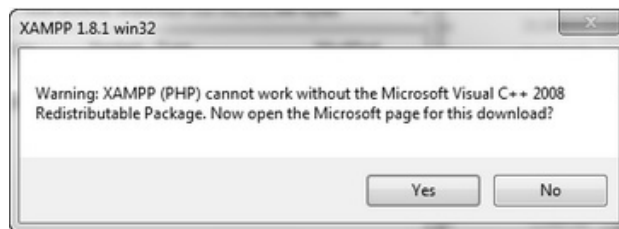
Pilihlah bahasa yang ingin digunakan.



*Gambar 2.1 pemilihan bahasa program*



*Gambar 2.2 Authority untuk UAC (User Account Control), klik OK*

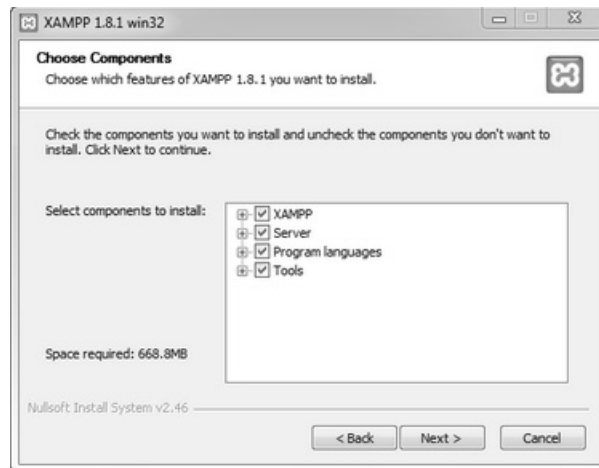


*Gambar 2.3 Open Microsoft page, klik No*



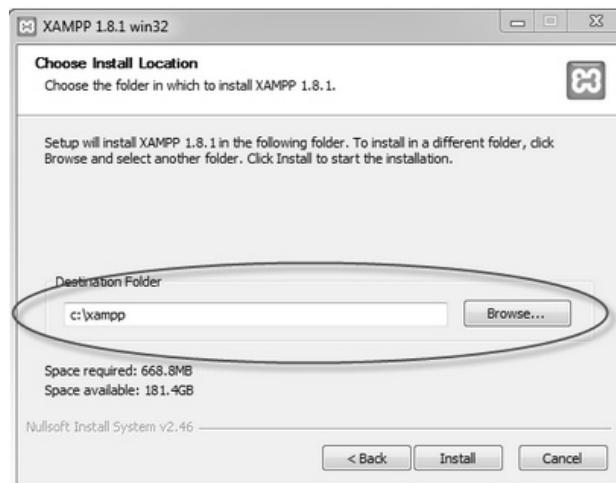
*Gambar 2.4 Welcome Screen, klik "Next"*

Pilih komponen-komponen yang akan diinstal dalam XAMPP. Jika sudah, kita tinggal klik tombol menu “Next”.



*Gambar 2.5 Memilih komponen yang akan diinstal*

Masukkan lokasi dari file dan folder XAMPP beserta file-file datanya.



*Gambar 2.6 Menentukan lokasi XAMPP*

Proses installasi sudah selesai dilakukan.



*Gambar 2.7 Proses instalasi telah selesai*

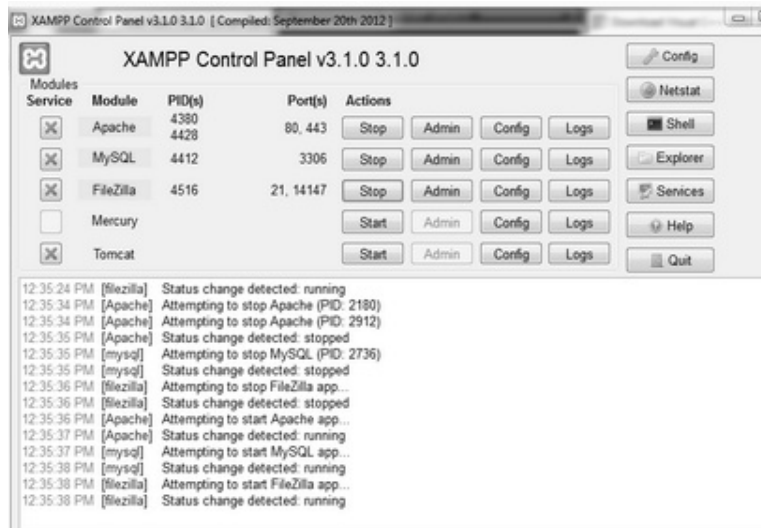
Buka control panel dari XAMPP untuk menjalankan server.



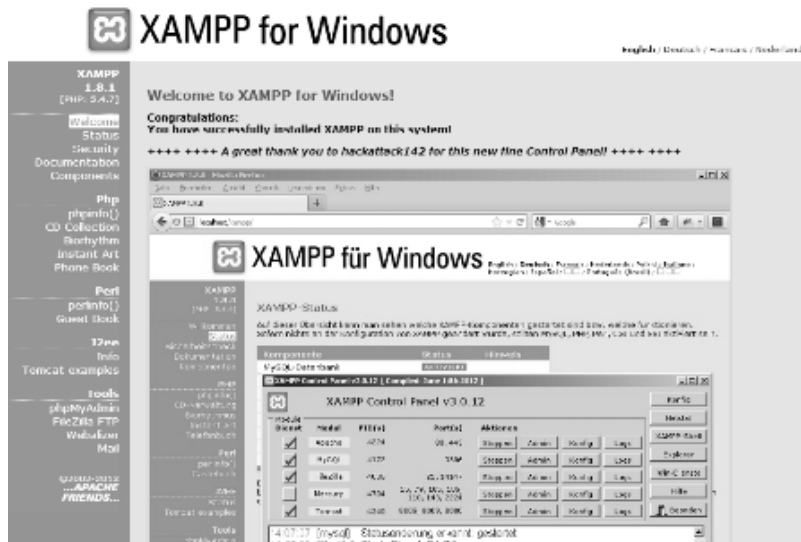
*Gambar 2.8 Membuka control panel XAMPP*



*Gambar 2.9 Tampilan control panel XAMPP*



Gambar 2.10 Konfigurasi control panel XAMPP



Gambar 2.11 Server XAMPP siap digunakan





# BAB 3

## VARIABEL DAN OPERATOR DI PHP

Siapa pun yang akan membuat program aplikasi dengan menggunakan PHP harus mengetahui dan memahami variable, operator, aturan, serta cara kerja pemrogramannya.

Variable berfungsi menyimpan nilai yang dapat diubah sewaktu-waktu. Variable ini yang menyebabkan web menjadi dinamis karena perubahan nilai pada variabelnya.

Variable digunakan untuk menyimpan nilai seperti text, number, atau array. Sebuah variable yang kita deklarasikan bisa digunakan berkali-kali dalam sebuah script.

Semua variable dalam php dimulai dengan tanda string (\$) untuk semua tipe data. Selain itu, kita tidak perlu mendeklarasikan tipe variable yang kita buat, PHP akan mendeteksi otomatis tipe variable yang kita buat.

Aturan pemberian nama variable dalam PHP:

1. Karakter yang dapat digunakan, yaitu huruf, angka, dan garis bawah (\_).
2. Karakter pertama harus huruf atau garis bawah.
3. Sebelum karakter pertama diawali dengan tanda \$.
4. Panjang pengenalan bisa bebas berapa saja.
5. Huruf besar dan kecil dibedakan.

Tipe data dalam PHP:

1. Integer, menyatakan tipe data bilangan bulat.
2. Double, menyatakan tipe data bilangan real.
3. String, menyatakan tipe data teks/deretan karakter.
4. Konstanta, menyatakan nilai yang tetap di dalam program.

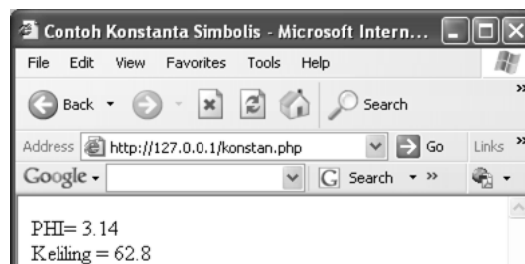
Penulisannya selalu diapit tanda petik tunggal atau petik ganda.

Contoh:

### Halaman konstan.php

```
<html>
<head>
<title>Contoh Konstanta Simbolis</title>
</head>
<body>
  <?php
    define("PHI", 3.14);
    $jari_jari = 10;
    $keliling = 2 * PHI * $jari_jari;
    printf("PHI = %s<br>\n", PHI);
    printf("Keliling = %s <br>\n", $keliling);
  ?>
</body>
</html>
```

Tampilan browsernya sebagai berikut.



*Gambar 3.1 Contoh hasil dari konstan.php*

## Halaman cast.php

```
<html>
<head>
<title>Contoh Konversi dengan cast</title>
</head>
<body>
<?php
$berat = "40.5 kg";
printf("Tipe String : $berat <br>\n");
    printf("Tipe Double : %s <br>\n", (double) $berat);
    printf("Tipe Integer: %s <br>\n", (int) $berat);
    printf("Tipe String : %s <br>\n", $berat);
$bilangan = 28;
$berat = (string) $bilangan; //konversi ke string
printf ("Bil.ke string : %s <br>\n", $berat);
?>
</body>
</html>
```

Tampilan browsernya sebagai berikut.



*Gambar 3.2 Tampilan Web Penulisan Variabel*

**Operator** adalah suatu symbol yang dipakai untuk memanipulasi nilai suatu variable. Kemudian, variable yang nilainya dimodifikasi oleh moderator disebut **operand**.

Misalnya 3-2, 3 dan 2 adalah operand dan - adalah operator.

Macam-macam operator adalah:

1. Aritmatik.
2. Perbandingan.
3. Kombinasi Aritmatik dan Asigment.
4. Pre/Post Increment dan Decrement.
5. Logika.

### 3.1 Operator Aritmatik

Berikut ini adalah tabel operator aritmatik, makna dan contohnya:

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

*Gambar 3.3 Tabel operator Aritmatik*

Contoh:

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

## 3.2 Operator Perbandingan

Perbandingan digunakan untuk menguji hubungan antara nilai dan atau variabel. Operator ini digunakan dalam suatu statement bersyarat yang selalu menghasilkan nilai TRUE atau FALSE.

Misalkan:

```
$x = 4; $y = 5;
```

Berikut ini adalah beberapa contoh penggunaan operator perbandingan dan hasilnya.

Operator	Makna	Contoh	Hasil
==	Sama dengan	<code>\$x == \$y</code>	FALSE
!=	Tidak sama dengan	<code>\$x != \$y</code>	TRUE
<	Lebih kecil dari	<code>\$x &lt; \$y</code>	TRUE
>	Lebih besar dari	<code>\$x &gt; \$y</code>	FALSE
<=	Lebih kecil atau sama dengan dari	<code>\$x &lt;= \$y</code>	TRUE
>=	Lebih besar atau sama dengan dari	<code>\$x &gt;= \$y</code>	FALSE

*Gambar 3.4 Operator di PHP dan hasilnya*

## 3.3 Kombinasi Operator Aritmatik dan Assignment

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

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

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

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

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

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

*Gambar 3.5 Operasi Aritmatik dan Assignment*

### 3.4 Operator Pre/Post Increment dan Decrement

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

Berikut adalah operator yang termasuk jenis ini:

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

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

Contoh:

```
<?php
$x = 4;
$x++;
echo "$x;
$x = 4;
$x--;
echo "$x;
?>
```

### 3.5 Operator Logika

Operator ini terdiri atas *and* dan *or* yang merupakan operator untuk mengombinasikan logika.

Contohnya, syarat diterima kerja adalah  $IPK > 3.0$  dan  $TEOFL > 400$ . Pernyataan “dan” atau *and* di sini merupakan kombinasi dari dua syarat ketika “dan” jika diganti *or* maka akan bermakna lain.

### Contoh:

```
<?php
$IPK= 3.6;
$TEOFL=390;
If (($IPK>=3.0) and ($TEOFL>=400))
{echo "LULUS";}else{echo"TIDAK LULUS";}
If (($IPK>=3.0) or ($TEOFL>=400))
{echo "LULUS";}else{echo"TIDAK LULUS";}
?>
```

## 3.6 Struktur Control

Dalam bahasa pemrograman PHP, secara mendasar struktur program dapat memiliki 3 kombinasi struktur kontrol, yakni:

1. Urutan (Sequence).
2. Percabangan (Barnc).
3. Pengulangan (Loop).

### 3.6.1 Struktur IF

Struktur *if* merupakan struktur kontrol pemilihan yang digunakan untuk percabangan. Apakah perintah-perintah di dalam blok dikerjakan atau tidak. Perintah dalam blok *if* akan di kerjakan jika nilai dari ekspresi di dalam *if* bernilai benar (*true*).

Sintaks dari statement tersebut dalam PHP adalah:

```
if (syarat)
{
    statement1;
    statement2;
}
```

Untuk menyatakan sebuah syarat, silakan menggunakan operator perbandingan seperti yang telah dibahas sebelumnya. Apabila syarat bernilai TRUE, maka statement yang diapit dengan tanda kurung kurawal akan dijalankan. Contoh sintak dengan struktur *if*:

```

<html>
<body>
<h1>if</h1>
<p> Contoh Struktur Kontrol if </p>
<br> $a = 5;
<br> $b = 7;
<hr>
<?
$a = 5;
$b = 7;
echo "\$a = $a <br>";
echo "\$b = $b <br>";
if ($a>$b){
echo "\$a > \$b";}
if ($a<$b){
echo "\$a < \$b";}
if ($b==$a){
echo "\$b = \$a";}
?>
</body>
</html>

```

### 3.6.2 Struktur ELSE

Digunakan untuk memberikan alternatif urutan perintah apabila ada proses yang memberikan dua alternatif benar atau salah. Else merupakan bagian seurutan perintah yang harus dikerjakan apabila hasil evaluasi dari ekspresi pada if bernilai salah.

```

if (syarat)
{
statement1;
statement2;
}
else
{
statement3;
statement4;
}

```

Untuk sintaks kedua di atas, statement3, statement4, dan seterusnya akan dijalankan apabila syarat bernilai FALSE.

Contoh:

```

<html>
<body>
<h1>if</h1>
<p> Contoh Srtuktur Kontrol if dan else</p>
<br> $a = 5;
<br> $b = 7;
<br> Hitung selisih

```



```

<br> $a = 7;
<br> $b = 5;
<br> Hitung selisih
<hr>
<?php
$a=5;
$b=7;
echo "\$a = $a <br>";
echo "\$b = $b <br>";
if ($a>$b)
    {$selisih=$b-$a;
    echo "Selisih \$a > \$b adalah $selisih
    "."<br>";}
if ($a<$b)
    {$selisih=$b-$a;
    echo "\$b < \$a adalah $selisih"."<br>";}
if ($b==$a)
    {echo "\$b = \$a"."<br>";}

$a=5;
$b=7;
echo "\$a = $a <br>";
echo "\$b = $b <br>";
if ($a>$b)
    {$selisih=$a-$b;
    echo "Selisih \$a > \$b adalah $selisih "."
    <br>";}

if ($a<$b)
    {$selisih=$b-$a;
    echo "\$a < \$b adalah $selisih "."
<br>";}
if ($b==$a)
    {echo "\$b = \$a"."<br>";}
?>
</body>
</html>

```

### 3.6.3 Struktur ELSE IF

Nilai suatu ekspresi bisa jadi bukan dua nilai benar atau salah, tetapi bisa jadi banyak nilai. Struktur `if...elseif` menyederhanakan model struktur kontrol `if...else`.

Sintaknya:

```

if (syarat1)
{
statement11;
statement12;
}
elseif (syarat2)
{
statement21;

```

```

statement22;
}
else
{
statement1;
statement2;
}

```

Jika syarat1 bernilai TRUE, maka statement11, statement12, dan seterusnya akan terus dijalankan. Sedangkan jika syarat1 FALSE, maka selanjutnya akan dicek untuk syarat2. Jika syarat2 TRUE, maka statemen21, statement22, dan seterusnya akan dijalankan, sedangkan jika syarat2 FALSE akan dicek syarat berikutnya sampai dengan selesai. Statement1, statement2, dan seterusnya baru akan dijalankan apabila semua syarat sebelumnya bernilai FALSE.

Contoh:

```

<html>
<body>
<h1> Demo elseif </h1>
<?php
    $bil=28;
    if ($bil>0)
        {echo $bil." Adalah positif";}
    else if ($bil<0)
        {echo $bil." Adalah negatif";}
    else
        {echo $bil." Adalah nol";}
?>
</body>
</html>

```

### 3.6.4 Struktur SWITCH

Merupakan bentuk struktur kontrol yang lebih sederhana dari pada if...else ataupun bentuk else if. Kontrol switch digunakan untuk mengevaluasi suatu ekspresi dengan kemungkinan banyak nilai dan banyak perintah yang harus dieksekusi berdasarkan ekspresi dan nilainya.

Sintaks dari statement ini adalah:

```

switch (variabel)
{
case option1:
statement11;
statement12;
break;

```

```

    case option2:
        statement21;
        statement22;
        break;
    default:
        statementdefault1;
        statementdefault2;
        break;
}

```

Pada sintaks di atas, nilai dari variabel akan dicek pada setiap option yang ada (lihat di bagian case). Jika ada option yang sama dengan nilai variabel, maka statement-statement di bawah option tersebutlah yang akan dijalankan. Sedangkan pada bagian default adalah optional (boleh ada dan boleh juga tidak).

```

<html>
<body>
<h1> Demo switch</h1>
<?
$nohari=2;
echo "No. hari : $nohari adalah hari :";
switch ($nohari) {
case 1:
echo "minggu";
break;
case 2:
echo "senin";
break;
case 3:
echo "selasa";
break;
case 4:
echo "rabu";
break;
case 5:
echo "kamis";
break;
case 6:
echo "jumat";
break;
case 7:
echo "sabtu";
break;
}
?>
</body>
</html>

```

Contoh berikutnya:

```

<html>
<body>
<h1> Demo switch</h1>

```

```

<?
$nohari=2;
echo "No. hari : $nohari adalah hari :";
switch ($nohari){
case 1:
echo "minggu";
case 2:
echo "senin";
case 3:
echo "selasa";
case 4:
echo "rabu";
case 5:
echo "kamis";
case 6:
echo "jumat";
case 7:
echo "sabtu";
}
?>
</body>
</html>

```

### 3.6.5 Struktur WHILE

Merupakan bentuk perulangan. Struktur kontrol ini merupakan seurutan perintah yang dieksekusi berulang-ulang. Jumlah perulangan yang dilakukan harus ditentukan oleh suatu nilai ekspresi. Statement dikerjakan secara berulang-ulang sampai suatu syarat dipenuhi. Sintaksnya adalah:

```

while (syarat)
{
    statement;
    statement;
}

```

Pada sintaks di atas, selama syarat bernilai **TRUE**, maka statement di dalam *while* akan terus dijalankan secara berulang-ulang. Perulangan baru akan berhenti apabila syarat bernilai **FALSE**. Sebelum statement yang diulang-ulang dilakukan, terlebih dahulu akan dicek syaratnya apakah bernilai **TRUE** atau **FALSE**. Apabila **TRUE**, maka statement akan dijalankan. Sedangkan apabila **FALSE**, perulangan akan langsung berhenti. Dengan kata lain, statement dalam **WHILE** bisa jadi tidak akan pernah dilakukan, yaitu apabila syaratnya langsung bernilai **FALSE**.

Contoh:

```
<html>
<body>
<h1> Demo while</h1>
<?
$bil =3;
while ($bil<10) {
echo "$bil";
echo " ";
$bil=$bil+3;
}
?>
</body>
</html>
```

### 3.6.6 Struktur DO WHILE

Membuat satu blok perintah di dalamnya untuk diulang-ulang perintah eksekusi perintahnya. Perbedaannya dengan do...while yaitu pemeriksaan ekspresi dilakukan pada bagian akhir dari blok perulangan. Perintah dalam blok akan dikerjakan selama kondisinya masih benar.

```
<html>
<body>
<h1> Demo do while</h1>
<?
$bil =3;
do {
echo "$bil";
echo " ";
$bil=$bil+3;
} while ($bil<10)
?>
</body>
</html>
```

### 3.6.7 Struktur FOR

Merupakan struktur kontrol perulangan dengan jumlah perulangan dapat ditentukan beberapa kali. Harus dilakukan perulangan dengan menggunakan bilangan sebagai penghitung. Statement FOR mirip dengan WHILE yang memiliki sintaks berikut:

```
for (inisialisasi counter; syarat; increment/decrement
counter)
{
statement;
}
```

Untuk memperjelas pemahaman tentang FOR, berikut ini contoh kode dengan for untuk menghasilkan tampilan yang sama dengan contoh while sebelumnya (tentang jumlah sikat dan harganya). Coba bandingkan dengan kode contoh while sebelumnya.

Contoh:

```
<?
$harga_sikat = 1500;
echo "<table border=\"1\" align=\"center\">";
echo "<tr><td><b>Jumlah Sikat</b></td>";
echo "<td><b>Harga</b></td></tr>";
for ($jumlah_sikat = 10; $jumlah_sikat <= 100;
$jumlah_sikat+=10)
{
echo "<tr><td>";
echo $jumlah_sikat;
echo "</td><td>";
echo "Rp. ".$harga_sikat * $jumlah_sikat;
echo "</td></tr>";
}
echo "</table>";
?>
```

### 3.6.8 Struktur FOREACH

Merupakan struktur kontrol khusus yang digunakan untuk melakukan pengulangan pada *array*. Dengan cara ini, kita tidak perlu mengetahui berapa jumlah *array* melakukan pengulangan.

Sintaksnya adalah:

```
foreach(variabelarray as kunci => value)
{
    statement;
}
```

Contohnya:

```
<?
$UsiaKaryawan["Lisa"] = "28";
$UsiaKaryawan["Jack"] = "16";
$UsiaKaryawan["Ryan"] = "35";
$UsiaKaryawan["Rachel"] = "46";
$UsiaKaryawan["Grace"] = "34";

foreach($UsiaKaryawan as $Nama => $umur)
{
echo "Nama Karyawan: $Nama, Usia: $umur." th <br>";
}
?>
```