

# MODUL PEMROGRAMAN WEB LANJUT



## DAFTAR ISI

1. Skema Aplikasi Web .....	1
2. Dasar PHP .....	2
2.1 Aturan Penulisan .....	2
2.2 Mencetak String .....	3
2.3 Karakter Escape .....	3
2.4 Menambahkan Komentar .....	5
3. Variabel .....	7
3.1 Deklarasi Variabel .....	7
3.2 Penugasan Variabel .....	8
3.3 Type Casting .....	9
4. Konstanta .....	10
5. Operator .....	12
5.1 Operator Aritmatika .....	12
5.2 Operator Relasional .....	13
5.3 Operator Logika .....	13
5.3.1 Logika AND ( Konjungsi ) .....	14
5.3.2 Logika OR ( Disjungsi ) .....	14
5.3.3 Logika XOR ( Exclusive OR ) .....	14
5.3.4 Logika NOT .....	15
5.4 Operator Penugasan .....	15
5.5 Operator Lain .....	17
5.5.1 Operator Penggabungan String .....	17
5.5.2 Operator Percabangan .....	18
5.5.3 Operator Error Suppression .....	18
Daftar Pustaka .....	20

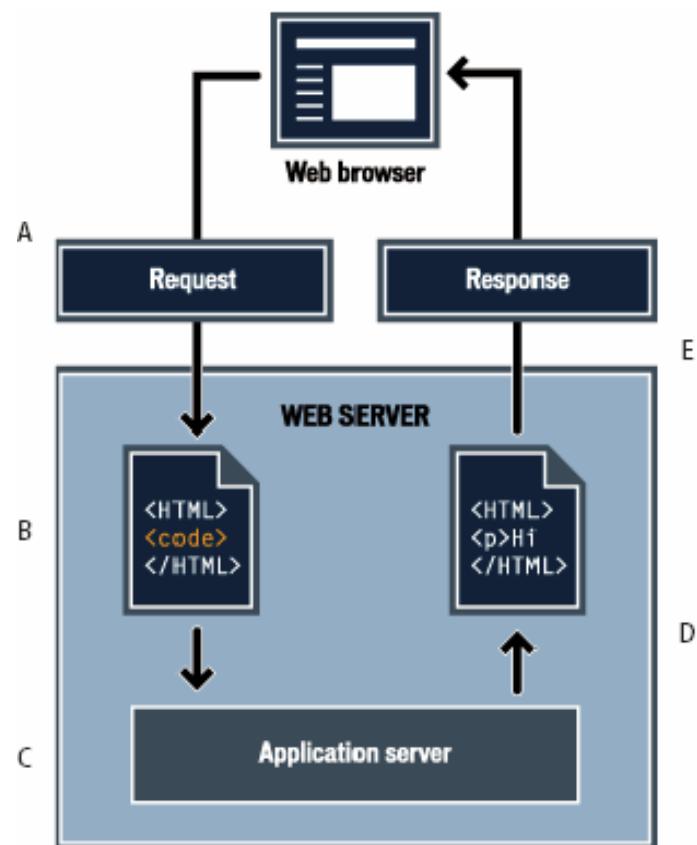
# PHP

PHP (*PHP : Hypertext Preprocessor*) adalah sebuah bahasa *scripting* yang paling popular saat ini. Berfungsi untuk membuat website dengan konten dinamis maupun aplikasi web lainnya. Berbeda dengan HTML yang hanya menampilkan konten statis, PHP dapat berinteraksi dengan database, file, folder bahkan dengan lingkungan sistem operasi. PHP termasuk kedalam golongan perangkat lunak interpreter. Yakni menerjemahkan baris demi baris program (tanpa mengcompilennya) kemudian mengeksekusinya.

PHP awal mula dikembangkan oleh Rasmus Lerdorf sebagai pengembangan dari Perl untuk menangani pengolahan form pada website pribadinya. Oleh karenanya, saat itu PHP disebut juga sebagai akronim dari *Personal Home Page*. Saat ini lebih dari 20 juta website menggunakan PHP *backend* aplikasi web.

## 1. Skema Aplikasi Web

Aplikasi web dinamis berbasiskan pada konsep arsitektur *client - server*, dimana server menangani apa yang dimininta oleh *client*, kemudian mengirimkan kembali hasilnya dalam bentuk HTML.



Skema aplikasi web pada gambar diatas dapat dijelaskan sebagai berikut:

- a. Client (web browser) meminta sebuah halaman pada sebuah website.
- b. Web server mencari halaman yang diminta, dan setelah ditemukan, memberikan halaman tersebut ke application server (PHP Interpreter).
- c. Application server mem-parse (membaca) dan memproses halaman tersebut.
- d. Application server mengirimkan halaman yang telah diproses kembali ke web server.
- e. Web server mengirimkan kembali halaman yang telah diproses kepada web browser dalam bentuk HTML.

## 2. Dasar PHP

### 2.1 Aturan Penulisan

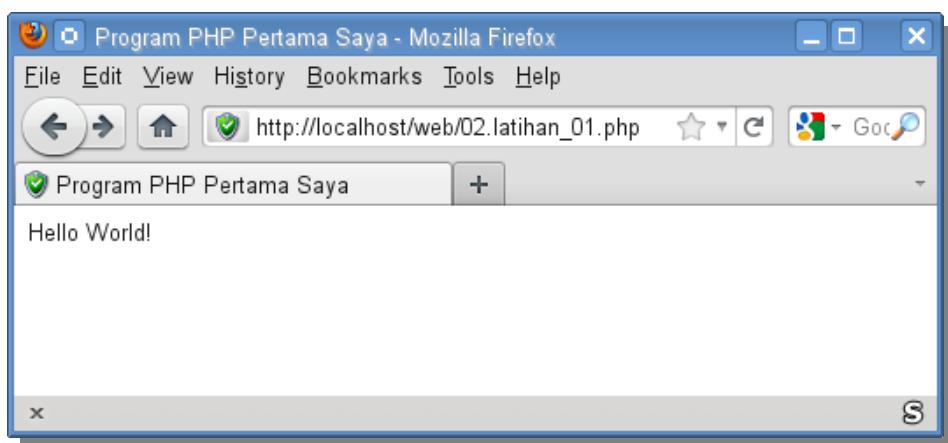
Sebagaimana umumnya bahasa pemrograman, PHP mempunyai aturan-aturan tertentu dalam penulisan skripnya. Diantaranya sebagai berikut:

- a. Program disimpan dengan ekstensi .php, .php3, .php4, atau .php5.
- b. Semua skrip harus diapit oleh tanda:
  - <?php dan ?>
  - <script language="php"> dan </script>
  - <? dan ?>
  - <% dan %>
- c. Setiap perintah harus diakhiri dengan tanda titik koma ( ; ).

Berbeda dengan bahasa banyak bahasa pemrograman web pada umumnya, skrip program PHP dapat langsung ditanamkan / disisipkan didalam HTML. Jika skrip PHP ditemukan dalam sebuah dokumen web, PHP Interpreter akan memproses skrip PHP dan hanya akan mengirimkan hasil berupa HTML ke browser client.

#### 02.latihan\_01.php

```
<html>
  <head>
    <title>Program PHP Pertama Saya</title>
  </head>
  <body>
    <?php echo "Hello World!"; ?>
  </body>
</html>
```



Dalam sebuah program, terdapat dua hal penting: *command* (perintah) dan *values* (nilai-nilai). *Command* memiliki tugas spesifik yang telah ditentukan. Sedangkan *values* adalah informasi yang berkaitan dengan tugas yang dilakukan oleh *command*. Dari contoh-contoh sebelumnya, kita telah banyak menggunakan *literal*. Yaitu *value* yang telah diterapkan secara eksplisit didalam program pada sebuah *command*. Sebagai contoh baris program berikut:

```
echo "Hello World!";
```

Pernyataan diatas melakukan *command echo* dan menggunakan *value literal* yang diberikan setelahnya, yaitu "Hello World!".

## 2.2 Mencetak String

Dalam PHP, string adalah rangkaian karakter, seperti "Once upon a time" atau "Doremifasolasido" bahkan "111211211". Ketika PHP membaca dan memproses sebuah kode program dan mengirimkan output ke web browser, setiap data yang ditampilkan akan dianggap sebagai sebuah string. Terdapat dua fungsi yang umum digunakan untuk mencetak sebuah string. Yaitu *print* dan *echo*.

Untuk mencetak sebuah string digunakan sepasang karakter *single quoted* ( ' ) atau *double quoted* ( " ) yang mengapit string tersebut.

## 2.3 Karakter Escape

Karakter escape adalah bagian dari syntax bahasa pemrograman. Biasa dimulai dengan karakter backslash ( \ ) dan diikuti oleh metakarakter tertentu yang disebut sebagai *escape sequences*.

*Escape sequences* memiliki 2 fungsi :

- a. Mengkodekan entitas syntax, yaitu perintah untuk menampilkan data khusus yang tidak dapat langsung diwakili oleh alfabet.
- b. Pengutip karakter, yaitu untuk mewakili karakter yang tidak dapat diketik dalam konteks saat ini, atau akan memiliki interpretasi yang tidak diinginkan.

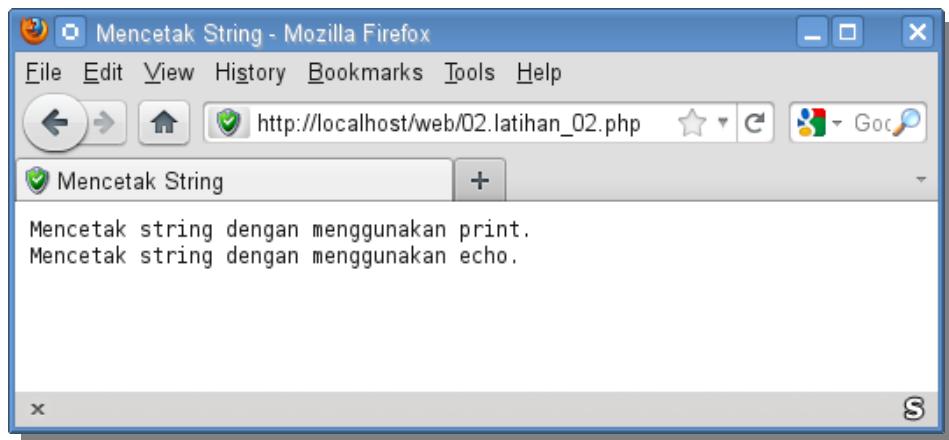
*Escape sequences* yang bekerja pada *double quoted* ( " ) adalah sebagai berikut:

<b>Escape sequences</b>	<b>Representasi Karakter</b>
\n	Ganti baris
\r	Carriage return
\t	Tab
\\"	Backslash
\\$	Tanda dolar
\"	Double quote
\{	Left brace
\}	Right brace
\[	Left bracket
\]	Right bracket
\0 atau \777	Octal value
\x0 atau \xff	Hexa value

Terdapat dua *escape sequences* yang dapat bekerja pada *single quoted* adalah '\, yang menempatkan tanda kutip tunggal dalam string yang dikutip tunggal, dan \\, yang menempatkan sebuah backslash dalam string *single quoted*.

## 02.latihan\_02.php

```
<html>
  <head>
    <title>Mencetak String</title>
  </head>
  <body>
    <pre><?php
      print "Mencetak string dengan menggunakan print.\n";
      echo "Mencetak string dengan menggunakan echo." ;
    ?>
    </pre>
  </body>
</html>
```



## 2.4 Menambahkan Komentar

Selain digunakan untuk memberikan keterangan mengenai kode program kita, komentar juga memberikan informasi kepada orang-orang yang ingin memahami kode program kita. Baris yang diberi komentar akan diabaikan (tidak diproses) oleh PHP.

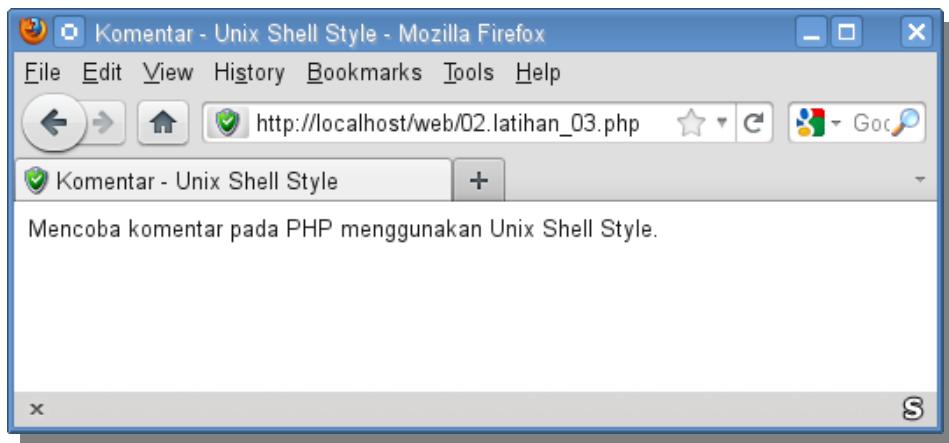
PHP menyediakan beberapa cara untuk mendefinisikan sebuah komentar. Dimana cara penulisan ini diinspirasi oleh bahasa program yang telah ada terlebih dahulu. Seperti C, C++, and the Unix shell.

### a. Unix Shell Style

Ketika PHP Interpreter menemukan tanda *hash* ( # ) pada kode program, segala sesuatu dari tanda *hash* ke akhir baris atau akhir dari bagian kode PHP akan dianggap sebagai komentar.

### 02.latihan\_03.php

```
<html>
<head>
    <title>Komentar - Unix Shell Style</title>
</head>
<body>
    <?php
        # ini adalah skrip latihan komentar
        # menggunakan hash ( # )
        echo "Mencoba komentar pada PHP menggunakan Unix Shell Style.";
    ?>
</body>
</html>
```



b. C++ Style

Ketika PHP Interpreter menemukan tanda *double slash* ( // ) pada kode program, segala sesuatu dari tanda *double slash* ke akhir baris atau akhir dari bagian kode PHP akan dianggap sebagai komentar. Hasil dari komentar C++ style ini akan sama dengan Unix Shell style.

02.latihan\_04.php

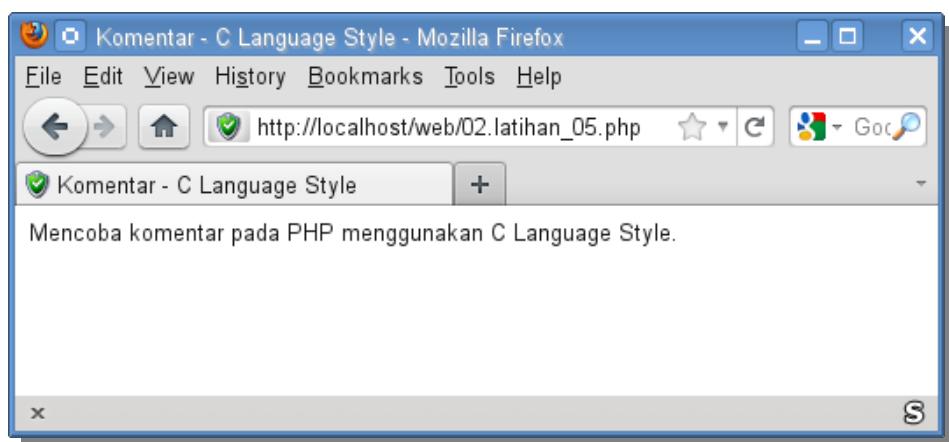
```
<html>
  <head>
    <title>Komentar - C++ Style</title>
  </head>
  <body>
    <?php
      // ini adalah skrip latihan komentar
      // menggunakan double slash ( // )
      echo "Mencoba komentar pada PHP menggunakan C++ Style.";
    ?>
  </body>
</html>
```

c. C Language Style

Penulisan komentar menggunakan Unix Shell style dan C++ style sangat berguna untuk membuat komentar singkat. PHP mendukung komentar blok, yang diinspirasi oleh bahasa C. Ketika PHP menemukan sebuah garis miring diikuti dengan tanda bintang ( /\* ), segala sesuatu sampai ditemukannya tanda bintang diikuti garis miring, akan dianggap sebagai sebuah komentar. Ini berarti C Language Style mendukung komentar dengan banyak baris sekaligus.

## 02.latihan\_05.php

```
<html>
  <head>
    <title>Komentar - C Language Style</title>
  </head>
  <body>
    <?php
    /*
      ini adalah skrip latihan komentar
      menggunakan C style
      bisa banyak baris hanya dengan 1 buah komentar saja
    */
    echo "Mencoba komentar pada PHP menggunakan C Language Style.";
    ?>
  </body>
</html>
```



## 3. Variabel

Variable adalah sebuah pengenal yang digunakan untuk menyimpan suatu nilai yang nilai tersebut dapat berubah-ubah selama dijalankannya program.

### 3.1 Deklarasi Variabel

Untuk mendefinisikan sebuah variabel, diawali dengan karakter dolar ( \$ ) dan diikuti dengan **nama pengenal**. Adapun aturan-aturan yang berkaitan dengan variabel adalah sebagai berikut:

- a. Terdiri dari huruf, angka dan karakter garis bawah ( \_ ) atau kombinasi antara ketiganya.
- b. Tidak boleh mengandung spasi dan terdiri dari minimal satu karakter.
- c. Dimulai dengan huruf atau garis bawah, tidak boleh diawali dengan angka.
- d. Bersifat *case sensitive*, yaitu huruf besar dan kecil dibedakan.

Seperti yang telah kita ketahui, sebuah variabel akan bernilai valid jika telah memenuhi persyaratan yang telah ditentukan diatas. Namun, untuk memudahkan memahami alur program kita, cobalah untuk mengikuti cara penulisan variabel seperti dibawah ini:

- a. Nama variabel mencerminkan isinya. Sebagai contoh, jangan gunakan variable `$x` untuk menyimpan total harga, melainkan gunakan `$total_harga`.

Jika suatu saat kita ingin memperbaiki kode program beberapa bulan setelah kita pertama kali menulis program itu, kita bingung dengan variabel bernama `$x` karena ia tidak mencerminkan apapun dari isinya. Berbeda dengan nama variabel `$total_harga` yang dengan jelas mencerminkan isinya.

- b. Sebisa mungkin variabel harus singkat. Mendefinisikan variabel terlalu panjang kadang-kadang terlalu merepotkan. Cobalah untuk mempersingkat, namun tetap masuk akal.

## 3.2 Penugasan Variabel

Terdapat dua cara dalam penugasan variabel: dari skrip itu sendiri atau ditentukan oleh PHP Interpreter. Saat ini kita hanya fokus pada penugasan yang dilakukan dari skrip. Seperti yang telah kita pelajari, penugasan terjadi ketika sebuah nilai diberikan pada sebuah variabel. Perhatikan dua pernyataan sebagai berikut:

```
$int_tiga = 3;  
3 = $int_tiga;
```

Aturan penugasan variabel dapat dimulai dari kanan ke kiri, yang dapat diterjemahkan sebagai berikut. Untuk pernyataan pertama, dapat dibaca “bilangan 3 ditugaskan ke variabel `$int_tiga`”. Kemudian kita bandingkan dengan pernyataan ke dua: “variabel `$int_tiga` ditugaskan ke bilangan 3”. Pernyataan tidak masuk akal dan tidak akan bekerja, dan jika menerapkan peryataan tersebut pada program, PHP akan berhenti dengan pesan error.

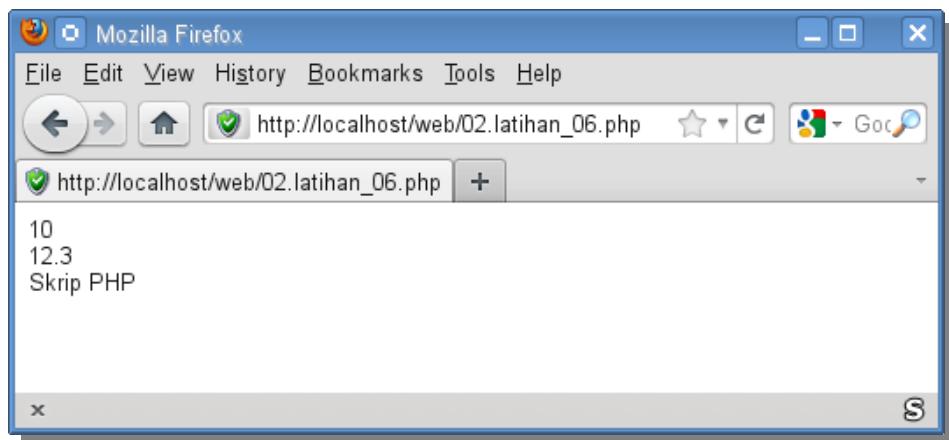
Tidak seperti bahasa pemrograman lainnya, tidak diperlukan deklarasi tipe data untuk sebuah variabel. Setiap variabel terbentuk dalam tipe data variant, yang artinya ia dapat menampung jenis data apapun.

## 02.latihan\_06.php

```
<?php

$info = 10; // variabel $info menampung bilangan bulat
echo $info;
echo "<br />";
$info = 12.3; // variabel $info menampung bilangan pecahan
echo $info;
echo "<br />";
$info = "Skrip PHP"; // variabel $info menampung string
echo $info;

?>
```



Meskipun variabel pada PHP bersifat variant, kita dapat mengetahui tipe data yang dikandung pada sebuah variabel menggunakan fungsi `gettype()`.

## 02.latihan\_07.php

```
<?php

$bool = true; // sebuah boolean
$str = "foo"; // sebuah string
$int = 12; // sebuah integer

echo "Variabel \$bool bertipe : ";
echo gettype($bool);
echo "<br />";
echo "Variabel \$str bertipe : ";
echo gettype($str);

?>
```

### 3.3 Type Casting

Type casting adalah mekanisme pengubahan / konversi sebuah variabel kepada tipe data tertentu. Sebagai contoh, kadangkala kita hanya ingin menggunakan bagian integer dari

sebuah bilangan pecahan. Untuk melakukan konversi, gunakan format cara berikut:

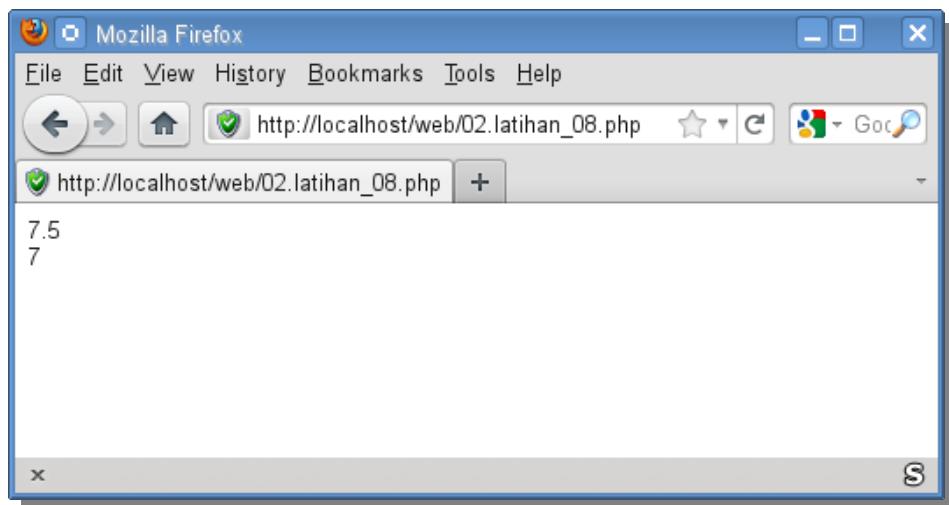
(tipe) \$variabel

Type casting dapat diterapkan pada sebuah variabel dengan tipe data sebagai berikut:

- (int) atau (integer)
- (real), (double) atau (float)
- (string)
- (array)
- (object)

### 02.latihan\_08.php

```
<?php  
  
$my_float = 7.5; // memberikan nilai pecahan ke variabel $my_float  
  
echo $my_float; // tampilkan data yang terkandung dalam $my_float  
echo "<br />";  
echo (int) $my_float; // hasil type casting dalam bentuk bilangan integer  
  
?>
```



## 4. Konstanta

Konstanta adalah sebuah tetapan nilai dalam sebuah program. Nilai konstanta tidak dapat diubah selama program dijalankan dan jika dilakukan hanya akan menyebabkan *error*. Deklarasi konstanta dapat dilakukan dengan cara sebagai berikut:

```
define("CONTOH_KONSTANTA", 5);
```

Perhatikan bahwa dalam pernyataan diatas, CONTOH\_KONSTANTA tidak diawali dengan

karakter dolar ( \$ ), karena karakter dolar hanya digunakan untuk sebuah variabel.

Sebagaimana variabel, konstanta juga yang memiliki aturan yang harus diikuti. Aturan berikut memastikan kita untuk mendeklarasikan konstanta dengan benar:

- a. Tidak boleh didahului oleh karakter dolar.
- b. Nama konstanta harus dimulai dengan huruf atau garis bawah ( \_ ).
- c. Konstanta tidak dapat dideklarasikan ulang dan nilainya tetap.

Sebagaimana variabel, dibutuhkan aturan-aturan khusus untuk memudahkan memahami alur program kita, cobalah untuk mengikuti cara penulisan konstanta seperti dibawah ini:

- a. Untuk kenyamanan, se bisa mungkin gunakan nama konstanta yang singkat. Karena pada umumnya konstanta digunakan untuk pilihan pengaturan.
- b. Selalu gunakan huruf besar. Ini membantu kita membedakan konstanta dengan pernyataan lain.
- c. Pisahkan setiap suku kata dalam konstanta dengan garis bawah ( \_ ). Garis bawah diperlukan untuk membantu kita membedakan satu konstanta dengan yang lainnya.

## 02.latihan\_09.php

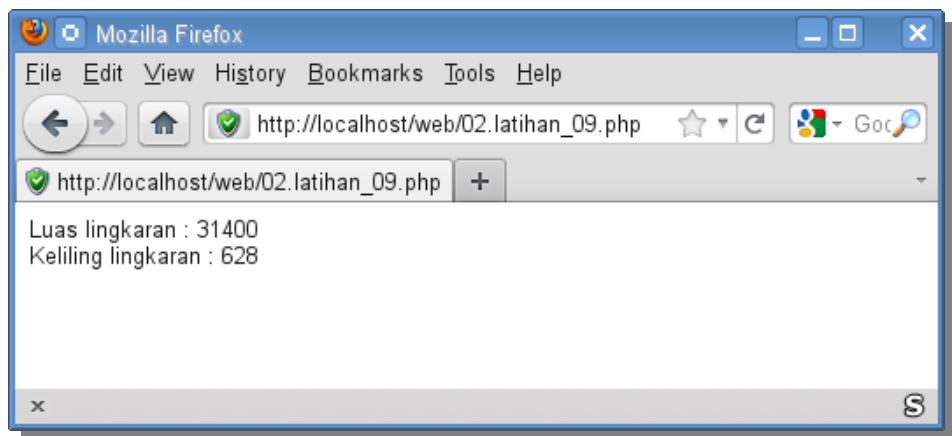
```
<?php

define("PI", 3.14);

$radius = 100;
$luas_lingkaran = PI * ($radius * $radius);
$keliling_lingkaran = 2 * PI * $radius;

echo "Luas lingkaran : ".$luas_lingkaran;
echo "<br />";
echo "Keliling lingkaran : ".$keliling_lingkaran;

?>
```



## 5. Operator

Operator digunakan untuk memanipulasi nilai sebuah variabel. Sedangkan nilai yang dimodifikasi oleh *operator* disebut *operand*. Sedangkan kombinasi dari keduanya disebut *ekspresi*. Dalam pemrograman operator dapat dikelompokkan menjadi:

- Operator Aritmatika
- Operator Penugasan
- Operator Perbandingan
- Operator Logika
- Lain-lain

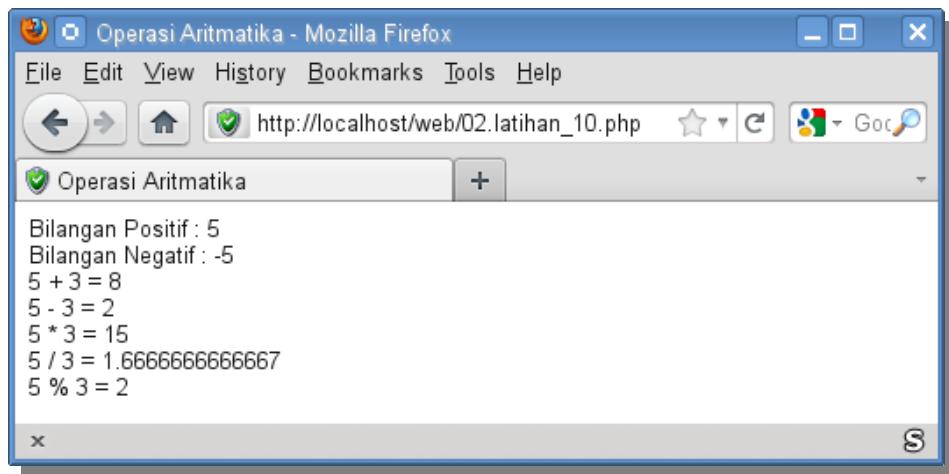
### 5.1 Operator Aritmatika

Operator aritmatika digunakan untuk melakukan perhitungan matematika.

Operator	Nama Operasi	Contoh
+ ( Unary )	Positif	+5
- ( Unary )	Negatif	-5
+	Penjumlahan	5 + 5
-	Pengurangan	5 - 5
*	Perkalian	5 * 5
/	Pembagian	5 / 5
%	Hasil Bagi	5 % 5

02.latihan\_10.php

```
<html>
  <head>
    <title>Operasi Aritmatika</title>
  </head>
  <body>
    Bilangan Positif : <?=> +5 ?>
    <br />
    Bilangan Negatif : <?=> -5 ?>
    <br />
    5 + 3 = <?=> 5 + 3 ?>
    <br />
    5 - 3 = <?=> 5 - 3 ?>
    <br />
    5 * 3 = <?=> 5 * 3 ?>
    <br />
    5 / 3 = <?=> 5 / 3 ?>
    <br />
    5 % 3 = <?=> 5 % 3 ?>
  </body>
</html>
```



Perhatikan pada contoh kode program diatas. Pernyataan <?= 5 + 3 ?> secara singkat dapat menggantikan pernyataan lengkap yang lebih panjang:

```
$bil_1 = 5;
$bil_2 = 3;
$hasil = $bil_1 + $bil_2;
echo $hasil;
```

## 5.2 Operator Relasional

Merupakan operator yang digunakan untuk membandingkan nilai dari dua operand. Hasil perbandingannya dinyatakan dalam nilai boolean. TRUE untuk pernyataan bernilai benar, dan FALSE untuk operator bernilai salah.

Operator	Keterangan
==	Memeriksa apakah operand kiri bernilai sama dengan operand kanan
>	Memeriksa apakah operand kiri lebih besar dari operand kanan
<	Memeriksa apakah operand kiri lebih kecil dari operand kanan
>=	Memeriksa apakah operand kiri lebih besar atau sama dengan operand kiri
<=	Memeriksa apakah operand kiri lebih kecil atau sama dengan operand kiri
!=	Memeriksa apakah operand kiri tidak bernilai sama dengan operand kanan

## 5.3 Operator Logika

Operator logika digunakan untuk membandingkan dua atau lebih nilai variabel yang bertipe boolean. Hasil yang didapat dari penggunaan operator logika adalah boolean.

Terdapat beberapa jenis operator logika, diantaranya sebagai berikut:

- Logika AND ( Disjungsi )
- Logika OR ( Konjungsi )
- Logika XOR
- Logika NOT ( Negasi )

### 5.3.1 Logika AND ( Konjungsi )

Logika AND akan bernilai TRUE jika pernyataan-pernyataan yang menjadi prasyaratnya bernilai TRUE. Dalam PHP, operator AND disimbolkan dengan tanda **&&**. Dibawah ini adalah tabel kebenaran untuk logika AND.

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

### 5.3.2 Logika OR ( Disjungsi )

Logika OR akan bernilai TRUE jika salah satu pernyataan yang menjadi prasyarat bernilai TRUE. Dalam PHP, operator OR disimbolkan dengan tanda **||**. Dibawah ini adalah tabel kebenaran untuk logika OR.

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

### 5.3.3 Logika XOR ( Exclusive OR )

Logika XOR akan bernilai TRUE jika pernyataan yang menjadi prasyaratnya mempunyai nilai kebenaran yang berbeda. Dalam PHP, operator XOR disimbolkan dengan tanda **XOR**. Dibawah ini adalah tabel kebenaran untuk logika XOR.

P	Q	$P \vee Q$
T	T	F
T	F	T
F	T	T
F	F	F

### 5.3.4 Logika NOT ( Negasi )

Logika NOT adalah nilai kebalikan dari pernyataan sebenarnya. Dalam PHP, operator NOT disimbolkan dengan tanda seru ( ! ). Dibawah ini adalah tabel kebenaran untuk logika NOT.

P	$\neg P$
T	F
F	T

## 5.4 Operator Penugasan

Operator penugasan digunakan untuk memberikan nilai pada sebuah variabel. Contoh:

```
$nama = "Haryo" ;
```

Pada contoh diatas, operator penugasan “=” digunakan untuk memasukkan string “Haryo” kedalam variabel \$nama. Selain operator penugasan “=”, juga dikenal beberapa operator penugasan lain. Diataranya adalah sebagai berikut:

- Operator +=

Digunakan untuk menambah variabel dengan bilangan tertentu. Contoh:

```
$a += 5; akan sama artinya dengan $a = $a + 5;
```

- Operator -=

Digunakan untuk mengurangi variabel dengan bilangan tertentu . Contoh:

```
$a -= 5; akan sama artinya dengan $a = $a - 5;
```

- Operator \*=

Digunakan untuk mengalikan variabel dengan bilangan tertentu. Contoh:

```
$a *= 5; akan sama artinya dengan $a = $a * 5;
```

- Operator /=

Digunakan untuk membagi variabel dengan dengan bilangan. Contoh:

```
$a /= 5; akan sama artinya dengan $a = $a / 5;
```

- Operator %=

Digunakan untuk mencari sisa hasil bagi dari sebuah variabel dengan bilangan tertentu. Contoh:

`$a %= 5;` akan sama artinya dengan `$a = $a % 5;`

- Operator &=

Digunakan untuk melakukan operasi logika AND pada sebuah variabel. Contoh:

`$a &= 5;` akan sama artinya dengan `$a = $a & true;`

- Operator |=

Digunakan untuk melakukan operasi logika OR pada sebuah variabel. Contoh:

`$a |= 5;` akan sama artinya dengan `$a = $a | false;`

- Operator ^=

Digunakan untuk melakukan operasi logika XOR pada sebuah variabel. Contoh:

`$a ^= 5;` akan sama artinya dengan `$a = $a ^ false;`

- Operator .=

Digunakan untuk menambahkan string pada sebuah variabel. Contoh:

`$a .= " Agung Nugroho ";` akan sama artinya dengan `$a = $a . " Agung Nugroho ";`

- Operator ++

Digunakan untuk menambahkan nilai satu (increment) pada sebuah variabel.

Contoh:

`$a++;` akan sama artinya dengan `$a = $a + 1;`

- Operator --

Digunakan untuk mengurangi nilai satu (decrement) pada sebuah variabel. Contoh:

`$a-- ;` akan sama artinya dengan `$a = $a - 1;`

## 02.latihan\_11.php

```
<html>
  <head>
    <title> Operator Penugasan </title>
  </head>
  <body>
    <?php
      $a = 10; // mula mula nilai variabel $a = 10

      echo "1. Penugasan +=      ";
      echo "<br />      ";

      $a += 8;
      echo "Variabel \$a saat ini bernilai : " . $a;
    ?>
```

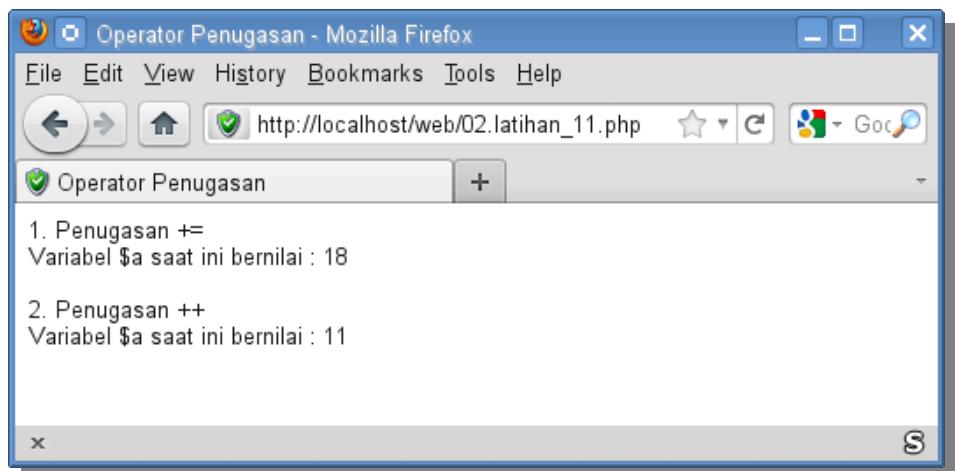
```

<br />
<br />
<?php
$a = 10; // nilai variabel $a dikembalikan menjadi 10

echo "2. Penugasan ++";
echo "<br />";

$a++;
echo "Variabel \$a saat ini bernilai : ". $a;
?>
</body>
</html>

```



## 5.5 Operator Lain

### 5.5.1 Operator penggabungan string

Pada bahasa pemrograman PHP, string digabungkan dengan karakter . ( titik ).

#### 02.latihan\_12.php

```

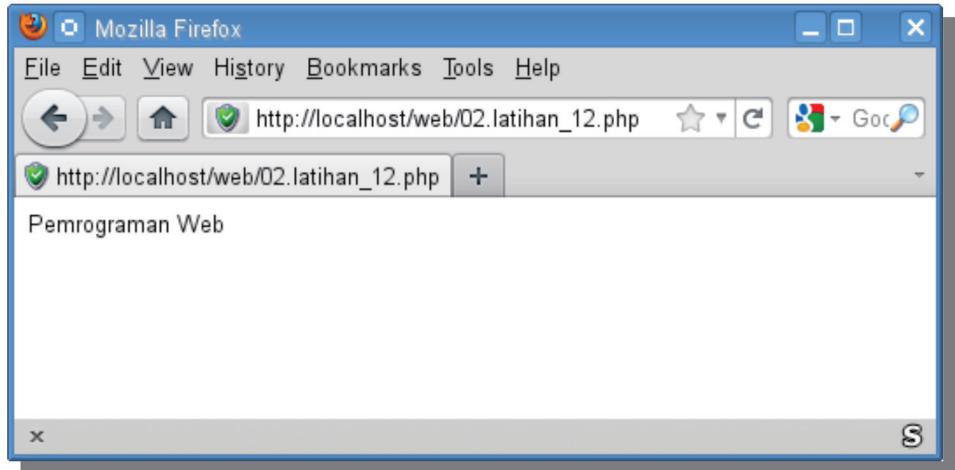
<?php

$string_1 = "Pemrograman";
$string_2 = "Web";

echo $string_1 . ".$string_2;

?>

```



### 5.5.2 Operator percabangan

Pada umumnya, operator percabangan dilakukan dengan struktur if-else, seperti pada contoh berikut:

```
if ( $user == "Haryo" ) {
    echo "Selamat datang Haryo.";
} else {
    echo "Username salah.";
}
```

Hal yang sama dapat dilakukan dengan cara:

```
echo ( $user == "Haryo" ? "Selamat datang Haryo." : "Username salah." );
```

Kondisi                      Pernyataan 1                      Pernyataan 2

Struktur diatas dapat dijelaskan sebagai berikut:

1. Pertama kali, PHP akan memeriksa apakah kondisi yang menjadi prasyarat bernilai benar. Yaitu apakah variabel \$user bernilai "Haryo".
2. Jika benar, pernyataan 1 yang dipilih, dan pernyataan 2 akan dipilih jika kondisi prasyarat bernilai salah.
3. Kemudian program akan menampilkan (echo) pernyataan yang telah dipilih tadi.

### 5.5.3 Operator Error Suppression

PHP akan menampilkan pesan kesalahan bilamana fungsi-fungsi *built-in* PHP mengalami sebuah kegagalan. Misalnya: tidak dapat membuka folder, file atau koneksi dengan database.

Pada saat pembuatan program, pesan kesalahan sangat berguna dalam membantu mencari dan memperbaiki kesalahan. Tetapi pada saat aplikasi yang dibuat sudah *dipublish*, hal ini akan membuat user yang menggunakan program menjadi terganggu. Untuk itu, diperlukan mekanisme khusus untuk menghilangkan pesan kesalahan tersebut. Untuk hal tersebut PHP menyediakan operator @.

```
@chdir("temp");
```

Pada kondisi normal, fungsi `chdir()` akan menampilkan pesan kesalahan apabila direktori `temp` tidak ditemukan atau tidak dapat diakses. Dengan operator @, PHP akan menghilangkan pesan kesalahan yang diakibatkan dari tidak ditemukan atau tidak dapat diaksesnya direktori `temp`.

## **DAFTAR PUSTAKA**

1. Buzton, Toby. 2002. *PHP By Example*. Indianapolis, Indiana: Que.
2. Lerdorf, Rasmus., Tatroe, K. 2002. *Programing PHP*. O'Reilly.
3. Sklar, David., Trachtenberg, A. 2002. *PHP Cookbook*. O'Reilly.
4. Yuliano,Triswansyah. 2007. *Pengenalan PHP*. IlmuKomputer.Com.
5. Muhardin, Endy. 2003. *PHP Programming Fundamental dan MySQL Fundamental*. IlmuKomputer.Com