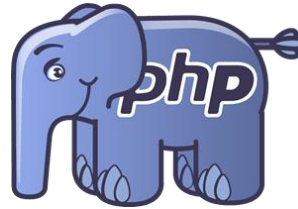


# Pemrograman PHP

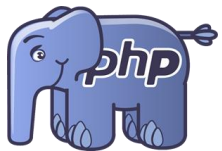


## Pesantren PeTIK II YBM PLN

Jl. KH. Bisri Syansuri RT/01 RW/05, Plosogeneng,  
Kec. Jombang, Kabupaten Jombang, Jawa Timur



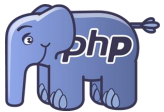
# Pertemuan Ke-2





# Materi

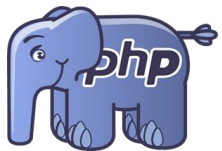
1. Pengantar Pemrograman PHP
2. Variabel dan Tipe Data
3. Array dan Struktur Data
4. PHP Form Processing
5. PHP Control Structure
6. PHP Function
7. Implementasi Template Web
9. PHP Object Oriented Programming
10. PHP Object Oriented Programming 2
11. PHP Database Connection
12. CRUD Database Operation 1
13. CRUD Database Operation 2
14. PHP Operasi file
15. Manajemen Session User



## 2. Variabel dan Tipe Data



# Apa itu Variable?





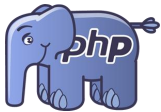
# Variable pada PHP

Sama seperti bahasa pemrograman pada umumnya yang memiliki variable sebagai tempat untuk menyimpan nilai (value) atau data yang dapat digunakan nanti dalam suatu program.

Variable bisa menyimpan data berupa integer, string, dan lainnya.

Nilai atau isi dari variable sendiri bisa diupdate atau diubah.

Variable akan ada selama program dijalankan. Namun kita juga bisa menghapusnya dari memori.





# Valid variable names

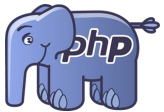
Dalam membuat nama variabel ada beberapa hal yang harus diperhatikan:

```
$!Ronaldo = "";  
$10Messi = "";
```

Contoh: ✗ Salah

```
$Ronaldo = "";  
$Messi10 = "";
```

Contoh: ✓ Benar





# Valid variable names

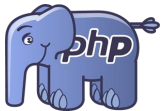
Nama variabel yang terdiri dari dua suku kata, bisa dipisah dengan underscore (\_) atau menggunakan style camelCase.

```
$nama_barang = "Laptop";
```

Underscore('\_')

```
$namaPembeli = "Usro";
```

StyleCamelCase







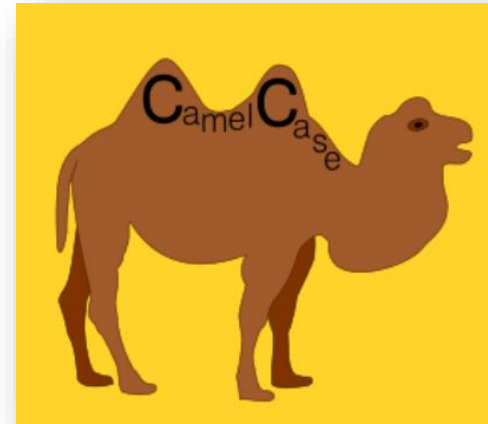
# Gaya penulisan case dalam kode program

Style snake\_case



\$nama\_variable  
\$nama\_barang  
\$nama\_class

Style CamelCase



\$NamaVariable  
\$NamaBarang  
\$NamaClass



# Gaya penulisan case dalam kode program

Style kebab-case



font-family  
text-align  
font-size

Style ALL CAPS

CAPS

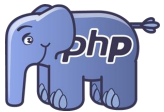
HOSTNAME  
DATABASE  
URL\_STRING



# Valid variable names

Nama variabel bersifat Case Sensitive, artinya huruf besar dan huruf kecil dibedakan.

```
$Belajar = "";  
$BELAJAR = "";  
$belajar = "";
```





# Bagaimana membuat Variable di PHP?

Pada PHP, kita membuat variabel dengan tanda dolar (\$), lalu diikuti dengan nama variabelnya serta nilai yang ingin kita simpan.

Contoh :

```
<?php  
$harga = 1000;
```

```
<?php  
$nama_barang = "Kopi Robusta";  
$harga = 20000;  
$stok = 30;
```





# Variable Sistem PHP (Predefined Variables)

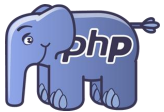
Predefined Variables atau yang dikenal dengan Variabel Sistem PHP, adalah beberapa variable yang telah didefinisikan secara oleh PHP, dan kita sebaiknya tidak membuat variable dengan nama yang sama.





## Beberapa contoh Predefined Variables dalam PHP adalah :

- ❖ \$GLOBALS
- ❖ \$\_SERVER
- ❖ \$\_GET
- ❖ \$\_POST
- ❖ \$\_FILES
- ❖ \$\_COOKIE
- ❖ \$\_SESSION
- ❖ \$\_REQUEST
- ❖ \$\_ENV
- ❖ \$php\_errormsg
- ❖ \$HTTP\_RAW\_POST\_DATA
- ❖ \$http\_response\_header
- ❖ \$argc
- ❖ \$argv
- ❖ \$this.





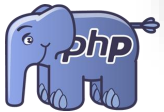
# Mengambil Nilai dari Variable

Mengambil nilai dari variabel bisa kita lakukan dengan menuliskan namanya dalam perintah 'echo'.

```
<?php

// Membuat Variable Baru
$harga1 = 1000;
$nama_barang = "Kopi Robusta";
$harga2 = 20000;
$stok = 30;

// Mengambil Nilai dari Variable
echo"Harga 1 Roti Gepeng adalah $harga1 <br>";
echo"Konsumen membeli $nama_barang dengan harga Rp$harga2 dengan jumlah $stok pcs";
```





# Tampilan di Browser







# Jenis-jenis Variable

## 1) Variable Local

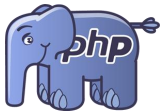
Variable Local adalah variable yang dideklarasikan dalam sebuah fungsi

## 2) Variable Global

Variable Global adalah variable yang dideklarasikan di luar sebuah fungsi

## 3) Variable Static

Variable Static adalah variable yang mempertahankan nilainya pada setiap pemanggilan fungsi.



# 1. Variable Local

```
1 <?php
2
3
4
5 function var_local()
6 {
7     $angka = 11;
8     echo "Local angka = $angka <br>";
9 }
10
11
12
13 ?>
```



localhost/



Maps



YouTube



Gmail



News

Berikut adalah Variable Local = 11

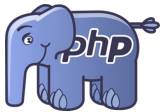
## 2. Variable Global

```
1 <?php
2
3     $angka = 12;
4
5     function var_local()
6     {
7         //
8     }
9
10    echo "Variable angka diluar var_local() adalah $angka";
11
12 ?>
```

← → ↻ ⓘ localhost

 Maps  YouTube  Gmail  News

Berikut adalah Variable Global = 12





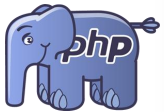
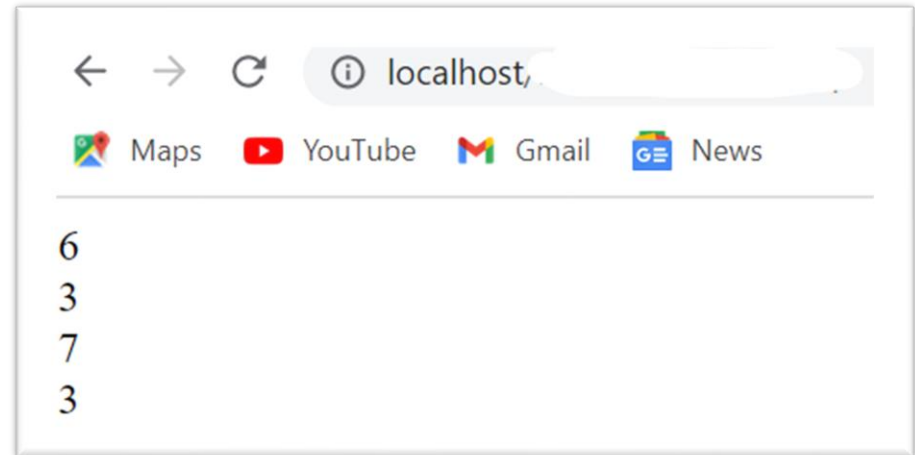
## 3. Variable Static

```
function var_static()
{
    // Definisikan Variable
    static $angka = 5;
    $nomor = 2;

    // penambahan pada variable
    $angka++;
    $nomor++;

    // Output Variable
    echo $angka, "<br>";
    echo $nomor, "<br>";
}

// panggil function var_static()
var_static();
var_static();
```





# Konstanta pada PHP

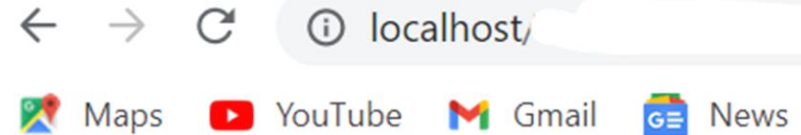
Dalam bahasa pemrograman, Konstanta (constant) adalah suatu lokasi penyimpanan (dalam memory) yang berisikan nilai yang sifatnya tetap dan tidak bisa diubah sepanjang program berjalan.









Berikut adalah contoh penulisan konstanta pada php.

```
<?php  
  
const gaji = "Rp10.000.000,00";  
echo gaji;  
  
?>
```



← → ↻ ⓘ localhost

 Maps  YouTube  Gmail  News

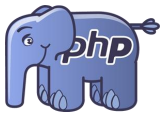
Rp10.000.000,00





# Penamaan Konstanta pada PHP

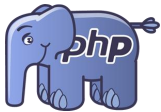
Aturan penamaan konstanta sama seperti variable, yakni untuk karakter pertama hanya boleh menggunakan huruf dan underscore ('\_'), dan untuk huruf kedua dan seterusnya, boleh menggunakan huruf, angka dan underscore. Sehingga sebuah konstanta juga tidak boleh diawali dengan angka atau mengandung karakter khusus seperti #, \*, atau &.





# Penamaan Konstanta pada PHP

Pembuatan konstanta dengan keyword `const` hanya dapat digunakan pada top-level scope, yakni harus dalam lingkungan global PHP. Sehingga kita tidak bisa menggunakan `const` di dalam function, loop, atau kondisi if. Apabila kita membutuhkan konstanta di dalam fungsi, maka harus menggunakan keyword **`define`**.







# Konstanta pada PHP bersifat Case Sensitif

Sama seperti variable, konstanta dalam PHP bersifat Case Sensitif, sehingga perbedaan huruf besar dan kecil dianggap berbeda. GAJI, GaJi, dan gaji merupakan 3 konstanta yang berbeda.

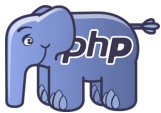
Walaupun kita boleh menggunakan huruf kecil dalam penulisan konstanta, kesepakatan programmer PHP menganjurkan menggunakan HURUF BESAR untuk penulisan konstanta.





## Beberapa contoh Predefined Constant dalam PHP adalah :

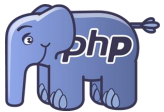
- |                       |                        |               |
|-----------------------|------------------------|---------------|
| ❖ PHP_VERSION         | ❖ PHP_SAPI             | ❖ PHP_MANDIR  |
| ❖ PHP_MAJOR_VERSION   | ❖ PHP_EOL              | ❖ PHP_LIBDIR  |
| ❖ PHP_MINOR_VERSION   | ❖ PHP_INT_MAX          | ❖ PHP_DATADIR |
| ❖ PHP_RELEASE_VERSION | ❖ PHP_INT_SIZE         | ❖ _LINE_      |
| ❖ PHP_VERSION_ID      | ❖ DEFAULT_INCLUDE_PATH | ❖ _FILE_      |
| ❖ PHP_EXTRA_VERSION   | ❖ PEAR_INSTALL_DIR     | ❖ _DIR_       |
| ❖ PHP_ZTS             | ❖ PEAR_EXTENSION_DIR   | ❖ _FUNCTION_  |
| ❖ PHP_DEBUG           | ❖ PHP_EXTENSION_DIR    | ❖ _CLASS_     |
| ❖ PHP_MAXPATHLEN      | ❖ PHP_PREFIX           | ❖ _TRAIT_     |
| ❖ PHP_OS              | ❖ PHP_BINDIR           | ❖ _METHOD_    |
|                       | ❖ PHP_BINARY           | ❖ _NAMESPACE_ |





# PHP – Multiple Code Block

```
<html>
  <head>
    <title><?php echo "Welcome to my site!";?></title>
  </head>
  <body>
    <?php
      $date = "May 18, 2003";
    ?>
    <h3>Today's date is <?=$date;?></h3>
  </body>
</html>
```





# PHP Basic Syntax - Comment

```
<?php
```

```
// satu baris komentar
```

```
# ini juga satu baris komentar, tapi jarang digunakan
```

```
/*
```

```
    banyak baris komentar
```

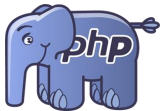
```
*/
```

```
// komentar php juga bisa diantara baris kode
```

```
$x = 5 /* +15 */ + 5 ;
```

```
echo $x;
```

```
?>
```

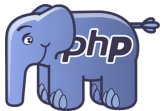




# PHP Cetak Standard Output

```
1  <?php
2      echo "Selamat Belajar PHP";
3  ?>
4  <?="Selamat Belajar PHP"?>
5  <hr/>
6  <?php
7      $_nama = "Ahmad Budiman";
8      echo $_nama; // cetak ke output
9  ?>
10 <br/>
11 Nama Lengkap : <?=$_nama?>
12 |
```

- Gunakan echo fungsi untuk cetak ke output
- atau tag <?=\$var ?> untuk variable tunggal

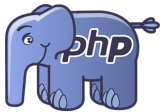




# PHP Basic Syntax – print vs echo

- Echo dan print adalah memiliki fungsi yang sama, mencetak output data ke layar display user
- Perbedaannya:
  - Echo tidak mengembalikan suatu nilai, sedang print mengembalikan nilai 1 karenanya bisa digunakan sebagai ekspresi program
  - Echo dapat menerima banyak parameter , print hanya 1 parameter
  - Echo lebih cepat dibanding print

```
<?php
    $nama = 'Putri Ramadhani' ;
    $_umur = 8;
    echo 'Nama Siswa : ' . $nama . ' umurnya ' . $_umur . ' tahun ' ;
?>
<br/>
<?php print($nama) ; ? >
```





## print()

boolean print (*argument*)

```
<?php  
    print("<p>I love the summertime.</p>");  
?>
```

```
<?php  
    $season = "summertime";  
    print "<p>I love the $season.</p>";  
?>
```

```
<?php  
    print "<p>I love the  
    summertime.</p>";  
?>
```

---

I love the summertime.

---

```
<?php  
    $season = "summertime";  
    print "<p>I love the ".$season."</p>";  
?>
```

# PHP print()



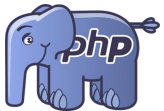


# PHP echo()

## echo()

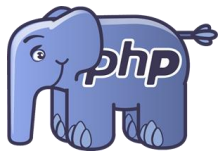
`void echo (string argument1 [, ...string argumentN])`

```
1 <?php
2     $_nama="Faiz Fikri";
3     $_umur= 8;
4     echo $_nama . " umurnya " . $_umur . " Tahun";
5 ?>
```





# Apa itu Tipe Data?

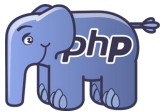




# Tipe Data pada PHP

Tipe bisa diartikan dengan jenis, dan data maksudnya seperti value pada PHP yang mempunyai jenis dan klasifikasi yang berbeda. Berikut adalah tipe data yang digunakan pada Pemrograman PHP:

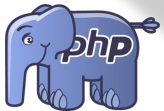
- ❖ String
- ❖ Integer
- ❖ Double/Float
- ❖ Boolean
- ❖ NULL
- ❖ Array
- ❖ Object
- ❖ Resource
- ❖ dll





# PHP Data Type

| Type                | Description   |
|---------------------|---|
| int, integer        | Whole numbers (i.e., numbers without a decimal point).  |
| float, double, real | Real numbers (i.e., numbers containing a decimal point).  |
| string              | Text enclosed in either single ( ' ') or double ( " " ) quotes.<br>[Note: Using double quotes allows PHP to recognize more escape sequences.] |
| bool, boolean       | True or false.  |
| array               | Group of elements.  |
| object              | Group of associated data and methods.   |
| resource            | An external source—usually information from a database.   |
| NULL                | No value.   |

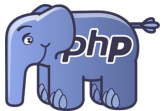




# Tipe Data String pada PHP

String adalah tipe data angka yang memuat segala karakter huruf, angka, symbol, dan tipe data ini memiliki ciri pada penulisan yang di apit oleh kutip 2. Berikut merupakan contoh penggunaan Tipe Data String di PHP:

```
<?php
    $lembaga = "Pesantren Teknologi Informasi dan Komunikasi (PeTIK II Jombang)";
    $alamat = "Jl. Denanyar, Jombang";
?>
```

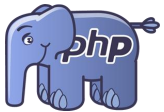




# Tipe Data Integer pada PHP

Integer adalah tipe data angka yang berbentuk bilangan bulat, tanpa titik decimal dan penulisannya tidak menggunakan tanda kutip 2. Berikut merupakan contoh penggunaan Tipe Data Integer di PHP:

```
<?php  
    $angka = 12345;  
    $nomor = 81905;  
?>
```

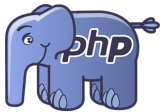




# Tipe Data Float/Double pada PHP

Tipe data float atau disebut juga tipe data double adalah tipe data yang berisi bilangan desimal, cara penulisannya hampir sama dengan tipe data integer karena tidak memerlukan tanda kutip 2. Berikut merupakan contoh penggunaan Tipe Data Float/Double di PHP:

```
<?php  
    $decimal = 123.45;  
    $phi = 3.14;  
?>
```

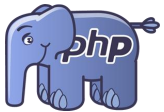




# Tipe Data Boolean pada PHP

Boolean adalah tipe data yang hanya memiliki dua buah value atau isi, yaitu true dan false. True valuenya adalah 1, False valuenya adalah 0. Berikut merupakan contoh penggunaan Tipe Data Boolean di PHP:

```
<?php
    if(TRUE){
        echo "Fulan selalu Cool";
    } else {
        echo "Fulan seorang Programmer";
    }
?>
```

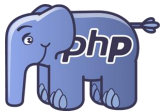




# Tipe Data NULL pada PHP

NULL adalah tipe data yang hanya memiliki satu nilai yaitu NULL atau kosong. Jika kita mengisi variable dengan nilai 'NULL', maka variable tersebut akan dianggap kosong atau tidak punya nilai. Berikut merupakan contoh penggunaan Tipe Data NULL di PHP:

```
<?php  
    $nama = NULL;  
?>
```



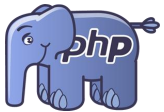




# Tipe Data Array pada PHP

Array adalah tipe data yang dapat menyimpan beberapa nilai atau value dari tipe data yang sama dalam sebuah variable. Berikut merupakan contoh penggunaan Tipe Data Array di PHP:

```
<?php
    $buah = array("Anggur", "Mangga", "Jeruk");
    $makanan = ["Kebab", "Sate", "Bakso"];
?>
```





# Tipe Data Object pada PHP

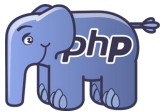
Object adalah tipe data yang tidak sekedar menyimpan data tetapi berisikan juga informasi bagaimana untuk mengolah data tersebut. Tipe data object sering digunakan dalam konsep Object Oriented Programming (OOP). Berikut merupakan contoh penggunaan Tipe Data Object di PHP:

```
.php
class Salam
{
    var $hai;

    // membuat object hello, ini disebut instansiasi
    $hello = new Salam();

    // memberikan value untuk properti hai pada objek hello
    $hello->hai="Hello World!";

    // mencetak nilainya (Pengolahan data)
    echo $hello->hai;
```

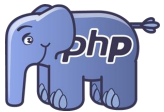




# Tipe Data Resource pada PHP

Resource adalah tipe data yang mereferensikan keterkaitan luar melalui fungsi. Contoh umum penggunaan tipe data ini adalah pemanggilan database. Berikut merupakan contoh penggunaan Tipe Data Resource di PHP:

- ❖ `mysqli_connect()`, digunakan untuk menghubungkan ke database SQL.
- ❖ `ftp_connect()`, digunakan untuk menghubungkan dan membuka protocol FTP
- ❖ `imagerotate()`, digunakan untuk memutar suatu gambar.





# PHP Keyword

## PHP keywords

|           |            |              |                 |              |
|-----------|------------|--------------|-----------------|--------------|
| abstract  | die        | exit         | interface       | require      |
| and       | do         | extends      | isset           | require_once |
| array     | echo       | __FILE__     | __LINE__        | return       |
| as        | else       | file         | line            | static       |
| break     | elseif     | final        | list            | switch       |
| case      | empty      | for          | __METHOD__      | throw        |
| catch     | enddeclare | foreach      | method          | try          |
| __CLASS__ | endfor     | __FUNCTION__ | new             | unset        |
| class     | endforeach | function     | or              | use          |
| clone     | endif      | global       | php_user_filter | var          |
| const     | endswitch  | if           | print           | while        |
| continue  | endwhile   | implements   | private         | xor          |
| declare   | eval       | include      | protected       |              |
| default   | exception  | include_once | public          |              |

Fig. 23.5 | PHP keywords.





# PHP Global Variabel Environment

- Define By System (PHP Global Variable ) – phpinfo
- Informasi Lingkungan Kerja Server

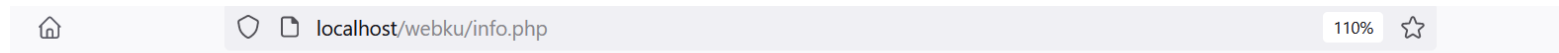
| localhost/webku/info.php 110% |                                     |
|-------------------------------|-------------------------------------|
| Module Name                   |                                     |
| Environment                   |                                     |
| Variable                      | Value                               |
| <i>no value</i>               | ::=:\                               |
| ALLUSERSPROFILE               | C:\ProgramData                      |
| APPDATA                       | C:\Users\62859\AppData\Roaming      |
| CMDER                         | %CMDER_HOME%\bin                    |
| CMDER_HOME                    | D:\devtools\cmdr                    |
| CommonProgramFiles            | C:\Program Files\Common Files       |
| CommonProgramFiles(x86)       | C:\Program Files (x86)\Common Files |
| CommonProgramW6432            | C:\Program Files\Common Files       |
| COMPUTERNAME                  | THINKBOOK-ROJUL                     |





# PHP Global Variabel Environment

- Define By System (PHP Global Variable ) – phpinfo
- PHP Variabel



## PHP Variables

| Variable   | Value  |
|--|--|
| <code>\$_SERVER['MIBDIRS']</code>                        | /xampp/php/extras/mibs   |
| <code>\$_SERVER['MYSQL_HOME']</code>                     | \xampp\mysql\bin   |
| <code>\$_SERVER['OPENSSL_CONF']</code>                   | /xampp/apache/bin/openssl.cnf  |
| <code>\$_SERVER['PHP_PEAR_SYSCONF_DIR']</code>           | \xampp\php   |
| <code>\$_SERVER['PHPRC']</code>                          | \xampp\php   |
| <code>\$_SERVER['TMP']</code>                            | \xampp\tmp   |
| <code>\$_SERVER['HTTP_HOST']</code>                      | localhost  |
| <code>\$_SERVER['HTTP_USER_AGENT']</code>                | Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0 |
| <code>\$_SERVER['HTTP_ACCEPT']</code>                    | text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8     |
| <code>\$_SERVER['HTTP_ACCEPT_LANGUAGE']</code>           | en-US,en;q=0.5   |
| <code>\$_SERVER['HTTP_ACCEPT_ENCODING']</code>           | gzip, deflate  |
| <code>\$_SERVER['HTTP_CONNECTION']</code>                | keep-alive   |
| <code>\$_SERVER['HTTP_UPGRADE_INSECURE_REQUESTS']</code> | 1  |





# Operator Aritmatika

- Operator digunakan untuk operasi matematika

| Example      | Label          | Outcome                             |
|--------------|----------------|-------------------------------------|
| $\$a + \$b$  | Addition       | Sum of $\$a$ and $\$b$              |
| $\$a - \$b$  | Subtraction    | Difference of $\$a$ and $\$b$       |
| $\$a * \$b$  | Multiplication | Product of $\$a$ and $\$b$          |
| $\$a / \$b$  | Division       | Quotient of $\$a$ and $\$b$         |
| $\$a \% \$b$ | Modulus        | Remainder of $\$a$ divided by $\$b$ |



# Operator Assignment

- Operator untuk memberi nilai variabel

| Example               | Label                     | Outcome  |
|-----------------------|---------------------------|--|
| <code>\$a = 5</code>  | Assignment                | <code>\$a</code> equals 5                                    |
| <code>\$a += 5</code> | Addition-assignment       | <code>\$a</code> equals <code>\$a</code> plus 5              |
| <code>\$a *= 5</code> | Multiplication-assignment | <code>\$a</code> equals <code>\$a</code> multiplied by 5     |
| <code>\$a /= 5</code> | Division-assignment       | <code>\$a</code> equals <code>\$a</code> divided by 5        |
| <code>\$a .= 5</code> | Concatenation-assignment  | <code>\$a</code> equals <code>\$a</code> concatenated with 5 |





# String Operator

- Operator digunakan untuk operasi tipe data string
- Gunakan titik ( . ) untuk operasi string

| Example                         | Label                    | Outcome  |
|---------------------------------|--------------------------|--|
| <code>\$a = "abc"."def";</code> | Concatenation            | <code>\$a</code> is assigned the string "abcdef"                     |
| <code>\$a .= "ghijkl";</code>   | Concatenation-assignment | <code>\$a</code> equals its current value concatenated with "ghijkl" |

```
// $a contains the string value "Spaghetti & Meatballs";  
$a = "Spaghetti" . "& Meatballs";
```

```
$a .= " are delicious";  
// $a contains the value "Spaghetti & Meatballs are delicious."
```



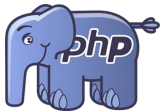


# Operator Incremental/Decremental

- Operator untuk menambah data secara incremental atau decremental

| Example      | Label     | Outcome            |
|--------------|-----------|--------------------|
| ++\$a, \$a++ | Increment | Increment \$a by 1 |
| --\$a, \$a-- | Decrement | Decrement \$a by 1 |

```
$inv = 15;          /* Assign integer value 15 to $inv. */  
$oldInv = $inv--;   /* Assign $oldInv the value of $inv, then decrement $inv.*/  
$origInv = ++$inv; /*Increment $inv, then assign the new $inv value to $origInv.*/
```





# Operator Logika

- Operator digunakan untuk logika perbandingan

| Example                         | Label        | Outcome  |
|---------------------------------|--------------|--|
| <code>\$a &amp;&amp; \$b</code> | And          | True if both <code>\$a</code> and <code>\$b</code> are true    |
| <code>\$a AND \$b</code>        | And          | True if both <code>\$a</code> and <code>\$b</code> are true    |
| <code>\$a    \$b</code>         | Or           | True if either <code>\$a</code> or <code>\$b</code> is true    |
| <code>\$a OR \$b</code>         | Or           | True if either <code>\$a</code> or <code>\$b</code> is true    |
| <code>!\$a</code>               | Not          | True if <code>\$a</code> is not true                           |
| <code>NOT \$a</code>            | Not          | True if <code>\$a</code> is not true                           |
| <code>\$a XOR \$b</code>        | Exclusive Or | True if only <code>\$a</code> or only <code>\$b</code> is true |

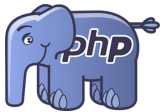




# Operator Equality

- Operator digunakan untuk mengecek kesamaan nilai

| Example   | Label           | Outcome  |
|-----------|-----------------|--|
| $a == b$  | Is equal to     | True if $a$ and $b$ are equivalent                                     |
| $a != b$  | Is not equal to | True if $a$ is not equal to $b$  |
| $a === b$ | Is identical to | True if $a$ and $b$ are equivalent, and $a$ and $b$ have the same type |

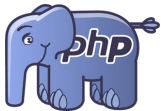




# Operator Equality

- Operator digunakan perbandingan nilai

| Example                | Label                    | Outcome   |
|------------------------|--------------------------|---|
| $\$a < \$b$            | Less than                | True if $\$a$ is less than $\$b$  |
| $\$a > \$b$            | Greater than             | True if $\$a$ is greater than $\$b$                                     |
| $\$a \leq \$b$         | Less than or equal to    | True if $\$a$ is less than or equal to $\$b$                            |
| $\$a \geq \$b$         | Greater than or equal to | True if $\$a$ is greater than or equal to $\$b$                         |
| $(\$a == 12) ? 5 : -1$ | Ternary                  | If $\$a$ equals 12, return value is 5;<br>otherwise, return value is -1 |



**TERIMA KASIH  
ATAS SEGALA PERHATIAN  
SEMOGA BERMANFAAT...**

