

PHP Dasar

PHP (Hypertext Preprocessor) merupakan salah satu bahasa web scripting yang sangat powerful. Ditemukan pertama kali oleh Rasmus Lerdorf tahun 1994. Bahasa ini dimaksudkan untuk menghasilkan halaman-halaman web yang dapat diintegrasikan dengan data dinamis. PHP dapat diakses jika server telah terinstall APACHE / NGINX sebagai Web Server dan URL untuk mengakses menggunakan IP server <http://192.168.254.1/> (sesuai setup server) atau <http://localhost/>. Untuk menginstall localhost di komputer dapat menggunakan [XAMPP](#) atau VMWare dengan OS yang mendukung APACHE dan PHP5.

1. Struktur Penulisan

Sama halnya dengan HTML dan CSS, PHP juga memiliki karakteristik penulisan pada file PHP. Adapun karakteristik tersebut adalah:

- File PHP berekstensi .php: **namafile.php**
- Ditulis diantara tag :

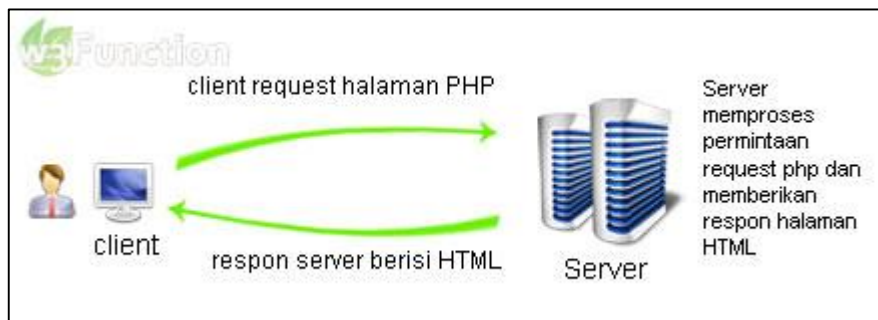
```
<?php
    $statemen_php = "Hello world";
?>

<?
    $statemen_php = "Hello world";
?>

<script language="php">
    $statemen_php = "Hello world";
</script>

<%
    $statemen_php = "Hello world";
%>
```

- Setiap statement / baris diakhiri dengan titik – koma (;)
- CASE SENSITIVE untuk nama identifier yang dibuat di file PHP (variable, konstanta, fungsi dan lain-lain), namun TIDAK CASE SENSITIVE untuk identifier built – in dari PHP. Contohnya adalah :
 - **\$nama** ≠ **\$Nama**
 - **hitungLuas()** ≠ **HitungLuas()**
 - **echo** = **ECHO**
 - **while** = **WHILE**
- Untuk menampilkan hasil dari sebuah script php harus menggunakan perintah **echo**. Untuk PHP dasar data yang di-**echo** dianggap sebagai kode HTML oleh browser.



```
<?php
    echo "HELLO WORLD" ;
?>
```

Source kode: 1.helloworld.php

2. Variable

Variable digunakan untuk menyimpan informasi / data yang dapat dimanipulasi sesuai kebutuhan. Penulisan variable adalah:

- Diawali dengan tanda \$ diikuti oleh nama variable
- Panjang tidak terbatas
- Setelah tanda \$, diawali oleh huruf / garis bawah
- Case sensitive
- Tidak boleh mengandung spasi

Benar	Salah
<code>\$_name</code>	<code>\$3name</code>
<code>\$first_name</code>	<code>\$name?</code>
<code>\$name</code>	<code>\$first+name</code>
<code>\$name_1</code>	<code>\$name.1</code>
<code>\$FirstName</code>	<code>\$first Name</code>

```
<?php
$variable = "Isi dari variable" ;
$nama     = "Mirza Ramadhany" ;
$salamat  = "Singosari" ;
$_nohp1   = "081945908363" ;
$_nohp2   = "-" ;

echo "NAMA SAYA " . $nama . " ALAMAT " . $salamat ;
echo "<br/>" ; /*echo akan menghasilkan bahasa html*/
echo "No HP 1 : " . $_nohp1 ;
echo "<br/>" ; /*echo akan menghasilkan bahasa html*/
echo "No HP 2 : " . $_nohp2 ;
?>
```

Source kode: 2.variable.php

```
NAMA SAYA Mirza Ramadhany ALAMAT Singosari
No HP 1 : 081945908363
No HP 2 : -
```

Hasil source kode: 2.variable.php

3. Tipe Data

- Integer

```
/*INTEGER*/  
$ndecimal = 1234 ; //decimal numbers  
$nnegative = -10 ; //negative number  
  
echo "<h3>Integer</h3>" ;  
echo var_dump($ndecimal) . " => " . $ndecimal . "<br />";  
echo var_dump($nnegative) . " => " . $nnegative . "<br />";
```

Source kode: 3.tipedaata.php /*INTEGER*/

```
Integer  
  
int(1234) => 1234  
int(-10) => -10
```

Hasil source kode: 3.tipedaata.php /*INTEGER*/

- Float

```
/*FLOAT*/  
$nfloat = 1.224 ;  
echo "<h3>FLOAT</h3>" ;  
echo var_dump($nfloat) . " => " . $nfloat . "<br />";
```

Source kode: 3.tipedaata.php /*FLOAT*/

```
Float  
  
float(1.224) => 1.224
```

Hasil source kode: 3.tipedaata.php /*FLOAT*/

- String

```
/*STRING*/  
$nstring = "Mirza Ramadhany" ;  
echo "<h3>STRING</h3>" ;  
echo var_dump($nstring) . " => " . $nstring . "<br />";
```

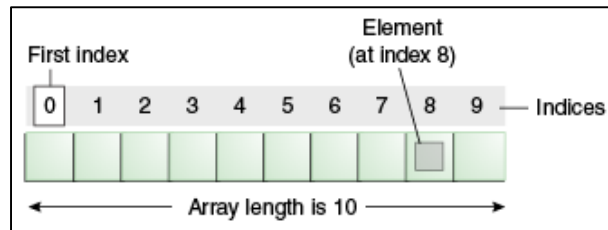
Source kode: 3.tipedaata.php /*STRING*/

```
String  
  
string(15) "Mirza Ramadhany" => Mirza Ramadhany
```

Hasil source kode: 3.tipedaata.php /*STRING*/

- Array

Array merupakan kumpulan data dalam satu variable, data-data dikumpulkan menggunakan inisial index. Index tidak hanya dalam bentuk angka (skalar) namun juga dapat diset menggunakan string (assosiatif). Array lebih baik di deklarasikan terlebih dahulu dengan kode **array()**, array dapat berisi data berupa string, integer ataupun float.



- **Array Dasar**

```
<?php
$vaskalar      = array() ;           //deklarasi array kosong
$vaskalar[0]   = "Index ke 0";      //index ke 0
$vaskalar[]    = "Index ke 1" ;     //index ke 1 otomatis mengikuti index sebelumnya

echo "<h3>Array Skalar</h3>" ;
echo "index ke-0 : " . $vaskalar[0] . "<br />" . "index ke-1 : " . $vaskalar[1] ;

$vaassosiatif  = array();           //deklarasi array kosong
$vaassosiatif["ke0"] = "data pertama" ; //index ke0
$vaassosiatif["oke"] = 1 ;          //index oke

echo "<h3>Array Assosiatif</h3>" ;
echo "index 'ke0' : " . $vaassosiatif["ke0"] . "<br />" . "index 'oke' : " . $vaassosiatif["oke"] ;

$vamultidimensi = array() ;         //deklarasi array kosong
$vamultidimensi[1] = array() ;      //deklarasi array kosong dalam index 1
$vamultidimensi[1][0] = "Isi Array 0" ; //index 1 array ke 0
$vamultidimensi[1][1] = "Isi Array 1" ; //index 1 array ke 1
$vamultidimensi[1]["ok"] = "Isi Array ok" ; //index 1 array "oke"

echo "<h3>Array Multidimensi</h3>" ;
echo '$vamultidimensi[1][0] : ' . $vamultidimensi[1][0] .
    "<br />" . '$vamultidimensi[1][1] : ' . $vamultidimensi[1][1] .
    "<br />" . '$vamultidimensi[1]["ok"] : ' . $vamultidimensi[1]["ok"] ;
?>
```

Source kode: 4.array_dasar.php

Array Skalar

index ke-0 : Index ke 0
index ke-1 : Index ke 1

Array Assosiatif

index 'ke0' : data pertama
index 'oke' : 1

Array Multidimensi

\$vamultidimensi[1][0] : Isi Array 0
\$vamultidimensi[1][1] : Isi Array 1
\$vamultidimensi[1]["ok"] : Isi Array ok

Hasil source kode: 4.array_dasar.php

- **Array Lanjutan**

```

<?php
$vaskalar      = array("Index ke 0","Index ke 1");    //mengisi variable dengan array

echo "<h3>Array Skalar</h3>";
echo "index ke-0 : " . $vaskalar[0] . "<br />" . "index ke-1 : " . $vaskalar[1] ;

$vaassosiatif  = array("ke0"=>"data pertama","oke"=>1); //deklarasi variable dengan array

echo "<h3>Array Assosiatif</h3>";
echo "index 'ke0' : " . $vaassosiatif["ke0"] . "<br />" . "index 'oke' : " . $vaassosiatif["oke"] ;

$vamultidimensi = array(
    1=>array("Isi Array 0","Isi Array 1","ok"=>"Isi Array ok")
);

echo "<h3>Array Multidimensi</h3>";
echo '$vamultidimensi[1][0] : ' . $vamultidimensi[1][0] .
    "<br />" . '$vamultidimensi[1][1] : ' . $vamultidimensi[1][1] .
    "<br />" . '$vamultidimensi[1]["ok"] : ' . $vamultidimensi[1]["ok"] ;
?>

```

Source kode: 5.array_lanjutan.php

Array Skalar

index ke-0 : Index ke 0
index ke-1 : Index ke 1

Array Assosiatif

index 'ke0' : data pertama
index 'oke' : 1

Array Multidimensi

\$vamultidimensi[1][0] : Isi Array 0
\$vamultidimensi[1][1] : Isi Array 1
\$vamultidimensi[1]["ok"] : Isi Array ok

Hasil source kode: 5.array_lanjutan.php

- Object

Merupakan tipe data lanjutan yang digunakan untuk PHP OOP (Object-Oriented PHP) yang lebih spesifik dan lebih terstruktur dalam kode pembuatan aplikasi / website.

```

1  <?php
2      class foo{
3          function do_foo(){
4              echo "Doing foo.." ;
5          }
6      }
7      $object = new foo ;
8      $object->do_foo() ;
9  ?>

```

Source kode: 6.object.php

Doing foo..

Hasil source kode: 6.object.php

4. Konstanta

Konstanta digunakan untuk mendeklarasi variable yang tidak dapat dirubah.

```
1 <?php
2 define("konstanta_pi",3.14) ;           //deklarasi konstanta
3 $njari2 = 10 ;
4 $nluas = konstanta_pi * $njari2 * $njari2 ;
5 echo "Jari-jari Lingkaran = " . $njari2 .
6     "<br />Luas Lingkaran = " . $nluas ;
7 ?>
```

Source kode: 7.konstanta.php

Jari-jari Lingkaran = 10
Luas Lingkaran = 314

Hasil source kode: 7.konstanta.php

5. Operator

- Aritmatika

Digunakan untuk perhitungan matematika

Operator	Operasi	Contoh
+	Penjumlahan	$a + b$
-	Pengurangan	$a - b$
*	Perkalian	$a * b$
/	Pembagian	a / b
%	Modulus (sisia pembagian)	$a \% b$

- Bitwise

Untuk membandingkan operan sebagai bilangan biner (0 atau 1)

Operator	Operasi	Contoh
&	And	$a \& b$
	Or	$a b$
^	XOR	$a \wedge b$
~	Not	$\sim a$
<<	Geser bit ke kiri	$a \ll b$
>>	Geser bit ke kanan	$a \gg b$

- Pembandingan

Untuk menbembalikan nilai dalam bentuk true atau false

Operator	Operasi	Contoh
==	Sama	$a == b$
===	Identik	$a === b$
!=	Tidak sama	$a != b$
!==	Tidak Identik	$a !== b$
<	Lebih kecil	$a < b$
>	Lebih besar	$a > b$
<=	Lebih kecil atau sama dengan	$a \leq b$
>=	Lebih besar atau sama dengan	$a \geq b$

- Penambahan/Pengurangan

Untuk penambahan/pengurangan secara teratur

Operator	Operasi
++\$a	Pre Increment
\$a++	Post Increment
--\$a	Pre Decrement
\$a--	Post Decrement

- Logika

Untuk melakukan perbandingan logic

Operator	Operasi	Contoh
and &&	Benar, jika keduanya bernilai benar	\$a and \$b \$a && \$b
or 	Benar, jika salah satu bernilai benar	\$a or \$b \$a \$b
xor	Benar, jika salah satu bernilai benar, tapi bernilai salah jika keduanya bernilai benar	\$a xor \$b
!	Benar jika pernyataan salah Salah jika pernyataan benar	!\$a

6. Statement

- If

Pernyataan if terdiri dari suatu ekspresi dan sebuah statemen atau blok statemen yang dieksekusi apabila ekspresi bernilai true.

```
<?php
$n1    = 10 ;
$n2    = 20 ;

//1. IF
echo "<h5>IF</h5>";
echo "10 > 20 ? " ;
if($n1 > $n2){
    echo "Nilai pertama lebih besar dari nilai kedua" ;
}

//1.1 BENTUK IF 1 BARIS
//(STATEMENT) ? HASIL BENAR : HASIL SALAH ;
echo "<h5>IF 1 BARIS</h5>";
echo "10 > 20 ? " ;
echo ($n1 > $n2) ? "Nilai pertama lebih besar dari nilai kedua" : "Nilai kedua lebih besar dari nilai pertama" ;

//2. IF ELSE
echo "<h5>IF ELSE</h5>";
echo "10 > 20 ? " ;
if($n1 > $n2){
    echo "Nilai pertama lebih besar dari nilai kedua " ;
}else{
    echo "Nilai pertama lebih kecil dari nilai kedua" ;
}

//3. IF ELSE
echo "<h5>IF ELSE</h5>";
echo "10 < 20 ? " ;
if($n1 > $n2){
    echo "Nilai pertama lebih besar dari nilai kedua " ;
}else if($n1 < $n2){
    echo "Nilai pertama lebih kecil dari nilai kedua" ;
}else{
    echo "Nilai pertama lebih sama dengan nilai kedua" ;
}
?>
```

Source kode: 8.if.php

```
IF

10 > 20 ?

IF 1 BARIS

10 > 20 ? Nilai kedua lebih besar dari nilai pertama

IF ELSE

10 > 20 ? Nilai pertama lebih kecil dari nilai kedua

IF ELSE

10 < 20 ? Nilai pertama lebih kecil dari nilai kedua
```

Hasil source kode: 8.if.php

- Switch

Sebuah pernyataan control flow yang dimulai dengan suatu ekspresi dan mentransfer kontrol ke satu kasus berdasarkan nilai ekspresi.

```
1  <?php
2  $n = 1 ;
3  echo "<h5>Switch</h5>" ;
4  echo '$n ' ;
5  switch ($n) {
6      case 0:
7          echo "equals 0" ;
8          break;
9      case 1 :
10         echo "equals 1" ;
11         break;
12     default:
13         echo "Tidak ditemukan di dalam case" ;
14         break;
15     }
16  ?>
```

Source kode: 9.switch.php

```
Switch

$1 equals 1
```

Hasil source kode: 9.switch.php

- While

Bentuk pengulangan dengan kondisi tertentu sampai kondisi tersebut menjadi false.


```
<?php
    $n = 0 ;
    while ($n < 10) {           //while
        echo 'Loop dengan nilai $n = ' . $n . '<br />';
        $n++ ;                 //increment yang digunakan agar $n menambah 1 setiap loop
    }
?>
```

Source kode: 10.while.php

```
Loop dengan nilai $n = 0
Loop dengan nilai $n = 1
Loop dengan nilai $n = 2
Loop dengan nilai $n = 3
Loop dengan nilai $n = 4
Loop dengan nilai $n = 5
Loop dengan nilai $n = 6
Loop dengan nilai $n = 7
Loop dengan nilai $n = 8
Loop dengan nilai $n = 9
```

Hasil source kode: 10.while.php

- Do

Bentuk pengulangan ini mirip dengan while, kecuali bahwa ekspresi pengontrol pengulangan dilakukan di akhir blok. Ini juga berarti bahwa blok pengulangan akan dieksekusi sedikitnya satu kali, meskipun bernilai false.

```
<?php
    $n = 0 ;
    do{
        echo 'Do WHILE dengan nilai $n = ' . $n . '<br />';
        $n++ ;                 //increment yang digunakan agar $n menambah 1 setiap loop
    }while($n < 10) ;
?>
```

Source kode: 11.dowhile.php

```
Do WHILE dengan nilai $n = 0
Do WHILE dengan nilai $n = 1
Do WHILE dengan nilai $n = 2
Do WHILE dengan nilai $n = 3
Do WHILE dengan nilai $n = 4
Do WHILE dengan nilai $n = 5
Do WHILE dengan nilai $n = 6
Do WHILE dengan nilai $n = 7
Do WHILE dengan nilai $n = 8
Do WHILE dengan nilai $n = 9
```

Hasil source kode: 11.dowhile.php

- For

Perulangan yang memiliki insialisasi, kondisi, dan inkrimen pada penulisan struktur for.

```
<?php
for($n = 0 ; $n < 5 ; $n++){
    echo 'For dengan nilai $n : ' . $n . '<br />';
}
?>
```

Source code: 12.for.php

```
For dengan nilai $n : 0
For dengan nilai $n : 1
For dengan nilai $n : 2
For dengan nilai $n : 3
For dengan nilai $n : 4
```

Hasil source code: 12.for.php

- Foreach

Konstruksi foreach yang dapat digunakan untuk melakukan iterasi di array atau koleksi.

```
<?php
$adata = array(1,2,3,4,"ini ada") ;
foreach ($adata as $key => $value) {
    echo 'Hasil dari $adata dengan index ' . $key . " adalah " |. $value . "<br />" ;
}
?>
```

Source code: 13.foreach.php

```
Hasil dari $adata dengan index 0 adalah 1
Hasil dari $adata dengan index 1 adalah 2
Hasil dari $adata dengan index 2 adalah 3
Hasil dari $adata dengan index 3 adalah 4
Hasil dari $adata dengan index 4 adalah ini ada
```

Hasil source code: 13.foreach.php

7. Menyisipkan File PHP

Penyisipan file digunakan untuk menyisipkan file php di file php lainnya, hal ini akan selalu digunakan untuk menghubungkan beberapa file konfigurasi dengan file lainnya.

Jika file yang disisipkan terdapat **ERROR**,
Include akan menghasilkan **WARNING**, sedangkan require akan menghasilkan **FATAL ERROR**.

```
<?php
echo "<h3>INCLUDE ONCE</h3>" ;
include_once "../13.foreach.php" ;

echo "<h3>REQUIRE</h3>" ;
require_once "../12.for.php" ;
?>
```

Source code: 14.penyisipanfile.php

INCLUDE ONCE

Hasil dari \$vadata dengan index 0 adalah 1
Hasil dari \$vadata dengan index 1 adalah 2
Hasil dari \$vadata dengan index 2 adalah 3
Hasil dari \$vadata dengan index 3 adalah 4
Hasil dari \$vadata dengan index 4 adalah ini ada

REQUIRE

For dengan nilai \$n : 0
For dengan nilai \$n : 1
For dengan nilai \$n : 2
For dengan nilai \$n : 3
For dengan nilai \$n : 4

Hasil source kode: 14.penyisipanfile.php

8. Function

Function adalah kode program yang dirancang untuk menyelesaikan sebuah tugas tertentu, dan merupakan bagian dari program utama. Kita dapat membuat fungsi sesuai kebutuhan atau menggunakan fungsi yang dibuat oleh programmer lain. Dalam sebuah program utama fungsi hanya boleh dideklarasikan 1kali dan nama fungsi tidak boleh sama dengan fungsi lainnya.

```
<?php
function nama_fungsi(parameter1,parameter2,...){
    statemen_fungsi;
    ....
    return $nilai_return;
}
?>
```

Keterangan	Kegunaan
nama_fungsi	Merupakan nama dari fungsi yang dibuat
parameter	Paramenter yang harus diisi jika fungsi dipanggil, sebuah fungsi dapat memiliki beberapa parameter atau tidak ada parameter apapun
statemen_fungsi	Kode-kode yang dirancang sesuai dengan kegunaan fungsi
return	Return / mengembalikan hasil dari fungsi, sebuah fungsi hanya bisa mengembalikan 1 variable atau tidak mengembalikan fungsi tersebut

- Function passing by value

```
<?php
/*Function passing by value*/
function fvhitung($nharga){
    $nharga = $nharga - (0.1*$nharga) ;
    return $nharga ;
}

$nharga = 2500 ;
echo "<h3>Fuction passing by value</h3>" ;
echo fvhitung($nharga) ;
echo "<br />";
echo $nharga ;
?>
```

Source code: 15.function.php /*Function passing by value*/

Fuction passing by value
2250
2500

Hasil source code: 15.function.php /*Function passing by value*/

- Function passing by reference

Dalam fungsi ini terdapat parameter yang diawali dengan & sebagai tanda penggantian nilai variable.

```
/*Function passing by reference*/  
function frhitung(&$nharga){  
    $nharga = $nharga - (0.1*$nharga) ;  
    return $nharga ;  
}  
  
$nharga = 9000 ;  
echo "<h3>Fuction passing by reference</h3>" ;  
echo frhitung($nharga) ;  
echo "<br />" ;  
echo $nharga ;
```

Source code: 15.function.php /* Function passing by reference*/

Fuction passing by reference
8100
8100

Hasil source code: 15.function.php /* Function passing by reference*/

- Function passing by default

```
/*Function passing by default*/  
function fdhitung($nharga=10000){  
    $nharga = $nharga - (0.1*$nharga) ;  
    return $nharga ;  
}  
  
$nharga = 23000 ;  
echo "<h3>Fuction passing by default</h3>" ;  
echo fdhitung() ;  
echo "<br />" ;  
echo fdhitung($nharga) ;
```

Source code: 15.function.php /* Function passing by default */

Fuction passing by default
9000
20700

Hasil source code: 15.function.php /* Function passing by default */