

▶ INTRO

PHP adalah bahasa pemrograman open source multi platforms yang sangat fleksibel, bahasa ini dapat disisipkan pada kode HTML ataupun CSS dengan syarat ekstensi file harus menggunakan ekstensi file PHP yaitu **.php** , file PHP dijalankan disisi server dan hasilnya dikembalikan pada browser dalam tampilan HTML. Sebuah file **.php** dapat berisikan teks, HTML, CSS, JavaScript dan kode PHP.

Tidak seperti HTML dan CSS yang tidak memerlukan host, untuk menjalankan PHP anda harus menginstal aplikasi web server dan PHP.

▶ SYNTAX

SYNTAX PHP dapat diletakkan dimanapun pada sebuah dokumen, dan syntax php dimulai dengan tag **<?php** dan diakhiri dengan tag **?>**

```
<?php
# Kode PHP diketikkan disini;
?>
```

Setiap dokumen yang terdapat syntax php harus memiliki ekstensi **.php** , setiap tag PHP selalu diakhiri dengan karakter semicolon (;) , dan variabel dalam PHP adalah case-sensitive. Berikut contoh penulisan tag PHP yang berfungsi memberikan output ke halaman web

```
<!DOCTYPE html>
<html>
  <body>
    <h1> Halaman PHP Pertamaku </h1><br>
    <?php
      echo "Hello World";

      // membuat komentar satu baris
      # juga membuat komentar satu baris
      /* ini contoh membuat komentar
         yang membutuhkan lebih dari satu baris */

      $warna="merah";
      echo "Mobil saya warna " . $warna . "<br>";
      echo "Rumah saya warna " . $WARNA . "<br>";
      echo "Kapal saya warna " . $waRNA . "<br>";

    ?>
  </body>
</html>
```

► VARIABLE

VARIABLE adalah kontainer (tempat) untuk menyimpan informasi seperti halnya pada ilmu matematik $x=5$, $y=6$ dan $z = x+y$. Pada PHP terdapat beberapa aturan dalam penamaan variabel :

- Variabel **dimulai dengan tanda \$** dan diikuti dengan nama variabel
- Nama variabel harus dimulai dengan huruf atau karakter underscore (A - z atau _) dan tidak boleh dimulai dengan angka
- Nama variabel berisikan karakter alpha-numeric dan underscore (A - z , 0 - 9 dan _)
- nama variabel adalah case sensitive (\$y dan \$Y adalah dua variabel berbeda)

PENDEKLARASIAN VARIABLE pada PHP tidak seperti pada bahasa pemrograman lain, kita tidak perlu mendefinisikan tipe data untuk variabel yang akan digunakan, secara otomatis PHP akan melakukan konversi variabel kepada tipe data yang sesuai tergantung nilai yang diberikan.

```
<?php
    $teks = "Mari Kita Belajar PHP";           # tipe data string / text
    $conf = true;                             # tipe data boolean
    $x     = 5;                               # tipe data integer / number
?>
```

CAKUPAN VARIABLE terbagi menjadi tiga yaitu :

- **LOCAL**, yaitu variabel yang dideklarasikan didalam sebuah fungsi dan hanya dapat digunakan didalam fungsi dimana dia dideklarasikan.
- **GLOBAL**, yaitu variabel yang dideklarasikan pada dokumen (diluar fungsi) dan hanya dapat digunakan diluar fungsi. Namun variabel ini dapat dipanggil didalam sebuah fungsi menggunakan syntax **global**.
- **STATIC**, merupakan variabel local, namun nilainya tidak hilang setelah fungsi selesai dijalankan, variabel static harus dideklarasikan dengan syntax **static**.

```
<?php
    $x     = 5;           # variable dengan cakupan global

    function Test(){
        $y     = 10; # variabel dengan cakupan local
        echo "<p>Tes variabel di dalam fungsi<p>";
        echo "Nilai variable x adalah : $x <br>"; # hasil kosong
        echo "Nilai variable y adalah : $y";      # hasil 10
        $y++;
    }
    Test();

    echo "<p>Tes variabel di luar fungsi<p>";
    echo "Nilai variable x adalah : $x <br>"; # hasil 5
    echo "Nilai variable y adalah : $y";      # hasil kosong
?>
```

Lalu sekarang sisipkan syntax global berikut sebelum baris **\$y =10;** dan refresh browser anda

```
global $x;
```

Berikutnya ubah baris **\$y=10;** menjadi **static \$y=10;** serta ubah baris **Test();** menjadi :

```
Test();
Test();
Test();
```

► CONSTANT

CONSTANT seperti variabel hanya saja sekali nilai konstanta didefinisikan maka nilainya akan tetap (tidak bisa dirubah ataupun dibatalkan pendefinisian), konstanta dimulai dengan huruf atau karakter underscore (_) dan tidak boleh dimulai dengan tanda \$

```
<?php
function Test(){
    static $i=1;
    $i++;
    define("GREETING", "Welcome to W3Schools.com!" . $i);
    echo GREETING . "<b> | $i </b><br>"; }
Test();    Test();    Test();
?>
```

► OPERATOR

OPERATOR ARITHMETIC (PERHITUNGAN)

Operator	Nama	Contoh	Hasil
		\$x = 10; \$y = 4;	
+	Penambahan	\$x + \$y	14
-	Pengurangan	\$x - \$y	6
*	Perkalian	\$x * \$y	40
/	Pembagian	\$x / \$y	2.5
%	Sisa Pembagian (Modulus)	\$x % \$y	2

```
<?php
$x=10;
$y=4;
echo ($x + $y); # output 14
echo "<br>";
echo ($x - $y); # output 6
echo "<br>";
echo ($x * $y); # output 40
echo "<br>";
echo ($x / $y); # output 2.5
echo "<br>";
echo ($x % $y); # output 2
?>
```

OPERATOR ASSIGNMENT (PEMBERIAN NILAI)

Operator	Sama Dengan	Contoh	Nilai \$x
		\$x = 10; \$y = 4;	
=	x = y	\$x = \$y	4
+=	x = x + y	\$x += \$y	14
-=	x = x - y	\$x -= \$y	6
*=	x = x * y	\$x *= \$y	40
/=	x = x / y	\$x /= \$y	2.5
%=	x = x % y	\$x %= \$y	2

```
<?php
$x=10; $y=4;
$x = $y; echo $x; # output 4
echo "<br>";
$x += $y; echo $x; # output 8
echo "<br>";
$x *= $y; echo $x; # output 32
echo "<br>";
$x -= $y; echo $x; # output 28
echo "<br>";
$x /= $y; echo $x; # output 7
echo "<br>";
$x %= $y; echo $x; # output 3
?>
```

OPERATOR STRING

Operator	Nama	
.	Penyambungan	<pre><?php \$a="nama"; \$b=" saya" \$sambung = \$a . \$b; echo \$sambung . "
"; \$sambung .= "dani"; echo \$sambung; ?></pre>
.=	Penyambungan	

OPERATOR INCREMENT / DECREMENT (PENAMBAHAN / PENGURANGAN NILAI)

Operator	Nama	Keterangan
++\$x	Pre-increment	Nilai \$x + 1, lalu return (kembalikan) \$x
\$x++	Post-increment	Return (kembalikan) \$x , lalu nilai \$x + 1
--\$x	Pre-decrement	Nilai \$x - 1, lalu return (kembalikan) \$x
\$x--	Post-decrement	Return (kembalikan) \$x , lalu nilai \$x - 1

```
<?php
    $x = 10;
    echo $x++ . "<br>"; # output 10
    echo ++$x . "<br>"; # output 12
    echo --$x . "<br>"; # output 11
    echo $x-- . "<br>"; # output 11
?>
```

OPERATOR COMPARISON (PEMBANDING)

Operator	Nama	Contoh	Hasil
		\$x =100 \$y ="100"	
==	Sama dengan	\$x == \$y	True , nilai \$x dan \$y sama
===	Identik	\$x === \$y	False , tipe \$x dan \$y beda
!=	Tidak sama dengan	\$x != \$y	False , nilai \$x dan \$y sama
<>	Tidak sama dengan	\$x <> \$y	False , nilai \$x dan \$y sama
!==	Tidak Identik	\$x !== \$y	True , tipe \$x dan \$y beda
		\$x = 40 \$y = 70	
>	Lebih besar dari	\$x > \$y	False , nilai \$x lebih kecil
<	Lebih kecil dari	\$x < \$y	True , nilai \$x lebih kecil
>=	Lebih besar atau sama	\$x >= \$y	False , nilai \$x lebih kecil
<=	Lebih kecil atau sama	\$x <= \$y	True , nilai \$x lebih kecil

OPERATOR LOGICAL

Operator	Nama	Contoh	Hasil
and	And	\$x and \$y	True , jika nilai \$x dan \$y keduanya benar
or	Or	\$x or \$y	True , jika \$x atau \$y salah satunya bernilai benar
xor	XOR	\$x xor \$y	True , jika \$x atau \$y benar, tapi tidak benar keduanya
&&	And	\$x && \$y	Sama dengan and
 	Or	\$x \$y	Sama dengan or
!	Not	!\$x	True , jika \$x bernilai salah

► IF CONDITIONAL STATEMENTS

CONDITIONAL STATEMENTS digunakan untuk memberikan instruksi berbeda pada kondisi yang berbeda pula.

IF STATEMENT untuk menjalankan syntax tertentu jika kondisi terpenuhi, berikut format if :

```
if (kondisi) {  
    syntax yang akan dijalankan jika kondisi benar;  
}
```

```
<?php  
    $login = true;  
    If ($login == true) {  
        echo "selamat datang admin, anda telah berhasil login";  
    }  
?>
```

IF ... ELSE STATEMENT untuk menjalankan syntax tertentu pada kondisi yang benar, atau menjalankan syntax lainnya pada kondisi yang salah. Berikut format syntax IF ... ELSE

```
if (kondisi) {  
    syntax yang akan dijalankan jika kondisi benar;  
}else{  
    syntax yang akan dijalankan jika kondisi bernilai salah;  
}
```

```
<?php  
date_default_timezone_set("Asia/Jakarta");  
$jam = date("H");  
if ($jam < 20) {  
    echo "selamat siang";  
}else{  
    echo "selamat malam";  
}  
?>
```

IF ... ELSEIF ... ELSE STATEMENT untuk menjalankan syntax tertentu pada kondisi yang benar, pada statement ini kondisi yang digunakan bisa lebih dari dua kondisi. Berikut format syntax

```
if (kondisi) {
    syntax yang akan dijalankan jika kondisi benar;
}elseif (kondisi) {
    syntax yang akan dijalankan jika kondisi benar;
}else{
    syntax yang akan dijalankan jika kondisi bernilai salah;
}
```

```
<?php
    $nilai = 70;
    if ($nilai <= 100 AND $nilai >= 85) {
        echo "SELAMAT , NILAI ANDA A";
    } elseif ($nilai < 85 AND $nilai >= 71) {
        echo "SELAMAT , NILAI ANDA B";
    } elseif ($nilai < 71 AND $nilai >= 55) {
        echo "NILAI ANDA C , KASIHAN ...";
    } elseif ($nilai < 55 AND $nilai >= 40) {
        echo "NILAI ANDA D , SELAMAT MENGULANG KEMBALI";
    } else {
        echo "NILAI ANDA E , SELAMAT MENGULANG KEMBALI";
    }
?>
```

► SWITCH STATEMENTS

SWITCH STATEMENTS fungsinya sama dengan perintah if, berikut format syntax nya.

```
switch (variabel) {
case nilai1 :
    syntax yang akan dijalankan jika variabel = nilai1;
    break;
case nilai2 :
    syntax yang akan dijalankan jika variabel = nilai2;
    break;
...
default :
    syntax yang akan dijalankan jika semua nilai diatas tidak sesuai;
}
```

```
<?php
    $warna = "merah";
    switch ($warna) {
    case "merah" :
        echo "warna kesukaan anda adalah merah";break;
    case "biru" :
        echo "warna kesukaan anda adalah biru";break;
    default :
        echo "warna $warna adalah warna yang tidak terdaftar";
    }
?>
```

► LOOPS STATEMENTS

LOOPS adalah perintah untuk melakukan pengulangan terhadap suatu blok syntax selama kondisi yang ditentukan bernilai benar, Terdapat beberapa jenis pengulangan pada PHP.

WHILE adalah perintah untuk mengulang blok kode selama kondisi bernilai benar, berikut adalah format syntax

```
while (kondisi bernilai benar) {  
    syntax yang akan dijalankan;  
}
```

```
<?php  
    $x = 1;  
    while ($x <= 5) {  
        echo "nilai x sekarang adalah $x <br>";  
        $x++;  
    }  
?>
```

DO ... WHILE sama seperti perintah while, hanya saja blok kode akan dijalankan terlebih dahulu kali, setelah itu baru melakukan pengecekan, apabila kondisi bernilai benar maka pengulangan akan dilakukan.

```
<?php  
    $x = 6;  
    do {  
        echo "nilai x sekarang adalah $x <br>";  
        $x++;  
    } while ($x <= 5)  
?>
```

FOR juga merupakan perintah untuk melakukan pengulangan dengan jumlah yang telah ditentukan. Berikut format syntax FOR

```
for(nilai awal; nilai akhir; peningkatan nilai) {  
    syntax yang akan dijalankan;  
}
```

```
<select name="tgl">  
    <?php    for($x = 1; $x <= 31; $x++) {    ?>  
        <option value="<?php echo $x; ?>">  
            <?php echo $x; ?>  
        </option>  
    <?php    }    ?>  
</select>
```

FOREACH hanya bekerja dengan array, dan digunakan untuk melakukan pengulangan pada setiap item dalam array. Berikut format syntax foreach

```
foreach($array as $value) {  
    syntax yang akan dijalankan;  
}
```

```
<?php
    $warna = array("merah","hijau","biru","kuning");
    foreach($warna as $value) {
        echo $value . "<br>";
    }
?>
```

► FUNCTION

FUNCTION adalah sebuah blok syntax yang dapat anda panggil berulang-ulang pada aplikasi, function biasanya berisi blok syntax dengan tujuan khusus, dan fungsi tidak dijalankan secara otomatis ketika suatu halaman ditampilkan, tetapi untuk menjalankan sebuah fungsi kita harus memanggilnya. Berikut format syntax function

```
function nama_fungsi([$argumen1],[$argumen2]) {
    syntax yang akan dijalankan;
}
```

```
<?php
    function hari_ini() { # fungsi tanpa argumen
        date_default_timezone_set("Asia/Jakarta");
        echo date('l, d-m-Y H:i:s');
    }
    hari_ini();
?>
```

```
<?php
    function nama_keluarga($fnama,$thnlahir) { # fungsi dengan argumen
        echo "$fnama ganteng, lahir pada tahun $thnlahir";
    }
    nama_keluarga("dani","1984");
?>
```

```
<?php
    # fungsi dengan argumen yang memiliki nilai default
    function tinggi_badan($tinggi=165) {
        echo "tinggi badan saya $tinggi cm <br>";
    }
    tinggi_badan(150);
    tinggi_badan(175);
    tinggi_badan();
?>
```

```
<?php
    # fungsi dengan pengembalian nilai
    function hitung_umur($thnlahir) {
        return date('Y') - $thnlahir;
    }
    $umur = hitung_umur(1984);
    echo "Dani ganteng umurnya sekarang udah $umur tahun";
?>
```


► ARRAY

ARRAY adalah variabel khusus yang dapat menampung banyak nilai, ada beberapa cara untuk mendefinisikan array.

```
<?php
# cara 1
$mobil = array("Volvo","BMW","Toyota");

# cara 2
$makanan[0] = "Nasi Goreng";
$makanan[1] = "Nasi Bakar";
$makanan[2] = "Nasi Uduk";

# cara 3 (Associative array)
$ukuran = array("Dani"=>"41","Yuni"=>"28","Daffa"=>"21")

# cara 4
$no_urut = array();
for($x = 1; $x <= 12; $x++) {
    $no_urut[] = $x;
}

?>
```

Berikut adalah cara untuk menampilkan nilai dari sebuah array

```
<?php
# cara 1
echo "saya suka mobil $mobil[0] , $mobil[1] , $mobil[2] <br>";

# cara 2
asort($makanan);      # mengurutkan array
$suka_makan = "saya suka makan ";
for($x = 0; $x <= count($makanan)-1; $x++) {
    $suka_makan .= $makanan[$x] . ", ";
}
echo $suka_makan . "<br>";

# cara 3 (Associative array)
echo "ukuran sepatu dani adalah nomor " . $ukuran['Dani'];

?>
```

► COLLECTING FORMS VALUE

Ketika tombol submit form diklik, maka data dari form tersebut akan dikirimkan ke file yang telah didefinisikan pada bagian atribut action, terdapat tiga jenis perintah untuk mengambil data yang dikirimkan tersebut yaitu **\$_POST** , **\$_GET** , **\$_REQUEST**

Berikut kode untuk mengambil data pada form yang telah dibuat pada pembahasan HTML, letakkanlah kode berikut pada bagian paling atas pada dokumen HTML

```

<?php
    if(isset($_POST['submit'])){
        echo "Nama : " . $_POST['nama'] . "<br>";
        echo "Jenis Kelamin : " . $_POST['sex'] . "<br>";
        echo "Tanggal Lahir : " . $_POST['tgl'] . "/" .
            $_POST['bln'] . "/" . $_POST['thn'] . "<br>";
        echo "Foto : "; # sempurnakan dengan $_FILES["foto"]["size"]
        move_uploaded_file($_FILES["foto"]["tmp_name"],
            "upload/" . $_FILES["foto"]["name"])
    ?>

    "
        width="200px" /><br>

<?php
    $kegemaran = ""; # sempurnakan dengan count
    foreach($_POST['suka'] as $value){
        $kegemaran .= $value . ", ";
    }
    echo "Kegemaran : " . $kegemaran . "<br>";
    echo "Deskripsi : " . $_POST['ds'] . "<br>";
?>

    <a href="index.php">Kembali ke formulir</a>

<?php
    exit;
}
?>

```

► **SELENGKAPNYA**

<http://www.w3schools.com/php/default.asp>