

## Hari 2 - PHP Dasar 2 (Function, Looping, Control Flow)

Pada dokumentasi ini akan dibahas materi lanjutan PHP di antaranya: operator, boolean, conditional, function, dan looping.

### Boolean(true & false)

Boolean adalah tipe data untuk pengecekan nilai suatu kebenaran. nilai dari boolean ada dua yaitu `true` dan `false`. Di PHP, nilai `true` sering dinyatakan sebagai integer dengan nilai 1, sedangkan nilai `false` dinyatakan sebagai integer dengan nilai 0.

### Operator di PHP

Operator adalah karakter khusus yang merepresentasikan sebuah tindakan. Operator terbagi ke dalam beberapa jenis:

#### Operator Aritmatika

operator yang melibatkan operasi matematika seperti tambah, kurang, kali, bagi, dan sisa bagi.

- tambah (+)
- kurang (-)
- kali (\*)
- bagi (/)
- modulus atau sisa bagi (%)

#### Operator Assignment

Operator assignment merupakan operator yang berguna untuk memasukkan nilai kedalam suatu variable. Dan jangan lupa, setiap baris code harus di tutup dengan ';' atau script menjadi error.

```
<?php
// contoh operator assignment
$panjang = 10; // contoh assignment
$lebar = 8; // contoh assignment
$luas = $panjang * $lebar ; // contoh assignment juga
// tanda '=' mewakili proses assignment,
// ada beberapa tanda lagi yang dapat
// melakukan assignment,
// assignment penjumlahan diwakili tanda '+=' contohnya
$tambahdua += 2;
echo $tambahdua."<br>" ;
// hasilnya 2
// hal ini sama dengan pernyataan :
// $tambahdua = $tambahdua + 2;
// assignment pengurangan diwakili tanda '-=' contohnya
$clip = 22;
$clip -= 2; // sama dengan $clip = $clip - 2;
echo $clip;
?>
```

#### Operator Pemanding

### Equal Operator ( == )

membandingkan kedua nilai apakah bernilai sama atau tidak.

```
<?php
$angka = 10;
echo $angka == 10; // true atau 1
var_dump($angka == 9); // bool(false) atau 0
?>
```

### Not Equal Operator ( != atau <> )

membandingkan kedua nilai apakah bernilai tidak sama.

```
<?php
$sifat= "rajin";
var_dump($sifat != "malas"); // bool(true)
echo $sifat <> "bandel"; // 1
?>
```

### Strict Equal ( === )

membandingkan kesamaan kedua nilai dan tipe data nya.

```
<?php
$angka = 10;
var_dump($angka === "10");// bool(false)
var_dump($angka === 10);// bool(true)
?>
```

### kurang dari, lebih dari ( < , > , <= , >= )

```
<?php
$angka = 17;
var_dump($angka < 20); // bool(true)
var_dump($angka <= 17); // bool(true)
var_dump($angka >= 15); // bool(true)
var_dump($angka > 12); // bool(true)
?>
```

## Operator Kondisional

Operator yang mengkombinasikan dua nilai kebenaran . Terdapat operator dan (&&) dan atau (||). nilai kebenaran dari kondisional ini mengikuti [kaidah de Morgan](#).

```
<?php
var_dump(true && true); // bool(true)
var_dump(true && false); // bool(false)
var_dump(false && true); // bool(false)
var_dump(false && false); // bool(false)

var_dump(true || true); // bool(true)
var_dump(true || false); // bool(true)
var_dump(false || true); // bool(true)
var_dump(false || false); // bool(false)
?>
```

## Kondisional

Kondisional atau percabangan merupakan logika dasar untuk membuat sebuah program. Dengan kondisional, kita dapat mengarahkan suatu program dengan pengecekan suatu kondisi, oleh karena itu kondisional sering disebut juga *control flow*.

Sintaksnya adalah sebagai berikut:

```
// blok kode ini akan dieksekusi jika kondisi salah
}
```

Contoh nya seperti ini:

```
<?php
$hari_ini = "senin";
if($hari_ini == "senin") {
    echo "I Love Monday";
} else {
    echo "Ini bukan hari senin"
}
?>
```

## ternary operator untuk conditional

salah satu cara singkat untuk melakukan kondisional adalah dengan ternary operator. Cara ini lebih sederhana dan dapat ditulis dalam satu baris saja.

Sintaks nya adalah seperti berikut

```
(Kondisi) ? (Statement1) : (Statemen2);
```

Statemen1 akan diisi jika Kondisi bernilai benar atau true, Sedangkan Statemen2 akan diisi jika Kondisi bernilai salah atau false.

Contohnya sebagai berikut:

```
<?php
$materi= "PHP";
echo ($materi == "PHP")?"Hari ini belajar PHP": "Hari ini bukan materi PHP";
?>
```

## Function

Function atau Fungsi adalah sebuah blok kode yang dibuat untuk menyederhanakan pemrograman agar proses-proses logika yang sering diperlukan dapat dipanggil kembali dengan hanya memanggil nama fungsi tersebut.

Sintaks untuk deklarasi function seperti berikut :

```
function <nama_function> (<parameter>) {
    <blok kode yang dijalankan>
}
```

Contohnya sebuah function untuk mengeprint <br>

```
// Deklarasi Function
function printBr() {
    echo "<br>";
}

// Menjalankan function atau pemanggilan function
printBr()
```

Sebuah function dapat menerima parameter sebagai input yang akan diolah sebelum memberikan output.

```
function kenalkan($nama) {
    echo "kenalkan nama saya $nama";
}

// panggil function kenalkan
kenalkan("abduh");
```

## return dalam function

contohnya sebagai berikut

```
function buat_nama_kapital($nama) {  
    return ucwords($nama);  
}  
  
$nama_lengkap = buat_nama_kapital("muhamad abduh");  
echo $nama_lengkap; // "Muhamad Abduh"  
  
// memanggil function di dalam function lain  
function perkenalan($nama, $asal) {  
    echo "perkenalkan nama saya buat_nama_kapital($nama), asal saya dari $asal";  
}  
  
perkenalan("muhamad abduh", "bandung");
```

## Looping

looping atau perulangan adalah melakukan suatu proses secara berulang sampai dengan suatu kondisi tertentu. Terdapat beberapa cara di PHP untuk melakukan looping yaitu for, while, do while.

### for loop

sintaks for seperti berikut:

```
for(nilai awal; batas nilai; operator increment/decrement)  
{  
    pernyataan yang akan di proses  
}
```

contohnya :

```
<?php  
for($i = 1; $i < 5 ; $i++){  
    echo "ini adalah looping ke $i <br>";  
}  
?>
```

### While Loop

sintaks while seperti berikut:

```
while (Kondisi) {  
    pertanyaan yang akan di proses;  
}
```

contohnya :

```
<?php  
$x = 1;  
  
while($x <= 5) {  
    echo "$x - satu Sampai lima<br>";  
    $x++;  
}  
?>
```

### Do While Loop

sintaks while do seperti berikut:

```
do {  
    pertanyaan yang akan diproses;  
} while (kondisi);
```

contohnya :

```
do {  
    echo "$x - satu Sampai lima<br>";  
    $x++;  
} while ($x <= 5);  
>>
```

### Foreach Loop

sintaks foreach do seperti berikut:

```
foreach ($array as $value) {  
    pertanyaan yang akan diproses;  
}
```

contohnya :

```
<?php  
$animals = array("cat", "dog", "snake", "ant", "lion");  
  
foreach ($animals as $animal) {  
    echo "$animal <br>";  
}  
>>
```

contoh lainnya:

```
<?php  
$age = array("Rezky"=>"25", "Abduh"=>"29", "Iqbal"=>"33");  
  
foreach($age as $x => $val) {  
    echo "$x = $val<br>";  
}  
>>
```

PKS Digital School : [https://www.youtube.com/playlist?list=PLdYMHqs00p5uu3fmPYEev\\_GXSHPfc9NSd](https://www.youtube.com/playlist?list=PLdYMHqs00p5uu3fmPYEev_GXSHPfc9NSd)

Referensi lain :

Sekolah koding : <https://www.youtube.com/playlist?list=PLdMopq3Lfhpe4IsVGITNqHjnupPXvxKrp>

## Rating - Feedback

Berikan Rating pada posting ini:



Berikan kritik dan saran..

Submit