

**MODUL**

# **PEMOGRAMAN WEB II**

Oleh:

CHALIFA CHAZAR

## MODUL 3

# Type Data, Variable, dan Konstanta

### Tujuan:

Mahasiswa mengenal tipe data dan mendeklarasikan variable di dalam pemrograman PHP sebagai pondasi dalam mempelajari materi-materi berikutnya.

### Pustaka:

Raharjo, B. (2015): Mudah Belajar PHP Teknik Penggunaan Fitur-Fitur Baru Dalam PHP 5. Informatika. Bandung

Buku Pintar Webmaster: Adhi Prasetyo, 2015

w3schools.com

### 1. Tipe Data

Dalam PHP, terdapat beberapa pengelompokan tipe-tipe data antara lain sebagai berikut.

- Integer
- Floating-point
- String
- Karakter
- Boolean
- Array
- Object

#### 1.1 Tipe Integer

Tipe integer adalah tipe data yang merepresentasikan bilangan bulat, dengan rentang nilai dari -2,147,483,648 sampai dengan 2,147,483,648.

Chalifa Chazar – edu.script.id – chalifa.chazar@gmail.com

Integer dapat ditulis dalam bentuk desimal (basis 10), oktal (basis 8), maupun heksadesimal (basis 16).

Contoh:

```
<?php
//basis 10
$rupiah=10000;

//basis 8 menampilkan desimal 8
$x=+010;

//basis 16 menampilkan desimal 10
$y=0x10;
?>
```

## 1.2. Tipe Floating-point

Tipe floating-point adalah tipe data yang merepresentasikan nilai-nilai numerik dalam bentuk pecahan atau mengandung angka desimal di belakang koma.

Contoh:

```
<?php
$nilai=3.14;
?>
```

## 1.3. Tipe String

Tipe string merepresentasikan data berupa teks. Penulisan string harus diapit diantara tanda petik ganda (") atau tanda petik tunggal (').

Contoh:

```
<?php
$x = 'hello world'
$y = "hello world"
echo $x;
echo "<br>";
echo $y;
?>
```

## 1.4. Tipe Karakter

Tipe data jenis ini digunakan untuk merepresentasikan data dengan nilai karakter tunggal. Penulisannya diapit dengan menggunakan tanda petik tunggal (').

Contoh:

```
<?php
  $x = 'A'
  $y = 'Z'
  echo $x;
  echo "<br>";
  echo $y;
?>
```

Sama seperti pada bahasa C, PHP juga mendukung *escape sequence*, yaitu karakter yang diawali dengan karakter backslash (\). *Escape sequence* memiliki aturan khusus dan kegunaan yang berbeda-beda seperti yang ditunjukkan pada tabel dibawah ini.

<i>Escape sequence</i>	Karakter yang direpresentasikan
\"	Double quotes
\n	Newline
\r	Carriage Return
\t	Tab
\\	Backslash
\\$	Dollar sign
\{	Left brace
\}	Right brace
\[	Left bracket
\]	Right bracket
\0 sampai \777	Karakter ASCII yang direpresentasikan dengan nilai oktal
\x0 sampai \xFF	Karakter ASCII yang direpresentasikan dengan nilai heksadesimal

## 1.5. Tipe Boolean

Tipe ini digunakan untuk menyatakan nilai **true** atau **false**. Tipe ini banyak digunakan dalam proses pemilihan atau pengulangan.

Contoh:

```
<?php
  if (!$connection) {
    echo "koneksi ke server error...";
  }
?>
```

### 1.6. Tipe Array

Tipe array adalah variable yang menyimpan sekelompok nilai yang dapat diidentifikasi atau diakses berdasarkan posisinya atau dengan nama yang telah didefinisikan sebelumnya. Dalam PHP, *index array* dimulai dari 0 bukan 1.

Contoh:

```
<?php
  $siswa[0]="budi";
  $siswa[1]="ajeng";
  $siswa[2]="danu";
?>
```

### 1.7. Tipe Objek

PHP merupakan pemograman berorientasi objek, oleh sebab itu PHP mendukung adanya tipe objek.

Melalui paradigma dan konsep OO (*object-oriented*), suatu permasalahan dapat disederhanakan menjadi kumpulan objek yang saling berkaitan. Dalam OOP (*object oriented programming*), kelas merupakan unit terpenting. Kelas adalah suatu definisi dari struktur yang memiliki properti dan metode (fungsi). Dalam PHP, kelas didefinisikan menggunakan kata kunci **class**.

Untuk menggunakan kelas tersebut kita perlu menggunakan kata kunci **new**; sedangkan properti (data) dan metode (fungsi) dalam suatu kelas dapat diakses melalui operator **->**.

Contoh:

```
<?php
    class manusia {
        var $nama="";
        //untuk memberikan nilai pada variabel nama
        function setnama($namasaya){
            $this->nama=$namasaya;
        }

        //untuk mendapatkan nilai dari variabel nama
        function getnama(){
            return $this->nama;
        }
    }

    $fae= new manusia();
    $ari= new manusia();

    $fae->setnama("chalifa");
    echo $fae->getnama();

?>
```

Catatan:

**\$this** adalah sebuah variabel khusus dalam OOP PHP yang mungkin digunakan sebagai petunjuk kepada objek, ketika kita mengaksesnya dari dalam class.

## 2. Variabel

Variabel adalah suatu pengenal di dalam program yang merepresentasikan nilai atau data tertentu. Nilai variabel dapat diubah sesuai kebutuhan program.

### 2.1. Mendefinisikan Variable

Variabel dalam PHP didefinisikan dengan menggunakan tanda dollar (\$) didepannya.

Contoh:

```
<?php
    $nama="budi";
    $umur=17;
?>
```

Penggunaan variabel PHP sedikit berbeda dengan bahasa pemograman lain pada umumnya. Dalam PHP suatu variabel tidak perlu dideklarasikan terlebih dahulu. Selain itu, satu variabel PHP dapat digunakan untuk menampung nilai yang berasal dari tipe data yang berbeda.

Contoh:

```
<?php
    $x="budi";
    $x=17;
    $x=1.5;
    $x=true;
    $x=array("budi", 17, 1.3, true)
?>
```

## 2.2. Variabel Lokal, Variabel Global, dan Variabel sebagai parameter fungsi

Variabel lokal adalah variabel yang hanya dideklarasikan pada suatu fungsi tertentu. Variable global adalah variabel yang akan dikenal oleh seluruh bagian program.

Variabel juga dapat digunakan sebagai parameter fungsi. Artinya, parameter hanya dikenal di lingkungan fungsi saja dan tidak dapat diakses dari luar fungsi.

Contoh:

```
<?php
    class manusia {
        var $nama="";
        //untuk memberikan nilai pada variabel nama
        function setnama($namasaya){
            $this->nama=$namasaya;
        }

        //untuk mendapatkan nilai dari variabel nama
        function getnama(){
            return $this->nama;
        }
    }

    $fae= new manusia();
    $ari= new manusia();

    $fae->setnama("chalifa");
    echo $fae->getnama();

?>
```

### Catatan.

Variabel "nama" (ditandai dengan warna kuning) adalah variabel global. Dan variabel "namasaya" (ditandai dengan warna biru) adalah variabel lokal sekaligus variabel sebagai parameter fungsi.

### 3. Konstanta

Konstanta adalah suatu pengenal yang sifatnya konstan (tetap/tidak dapat diubah). Dalam PHP, konstanta didefinisikan menggunakan fungsi `define()`.

Contoh:

```
<?php
    define('NAMA', "budi wibowo");
    print NAMA;

?>
```

## 4. Operator

Operator dalam PHP dapat dikategorikan menjadi beberapa kelompok, yaitu:

- Operator Penugasan
- Operator Aritmetika
- Operator Relasional
- Operator Logika
- Operator *Bitwise*
- Operator *String*
- Operator *Array*
- Operator Tipe
- Operator Kondisional
- Operator Eksekusi

### 4.1. Operator Penugasan

Operator penugasan adalah operator yang digunakan untuk mengisi nilai ke dalam suatu variabel. Notasi yang digunakan adalah tanda sama dengan (=).

### 4.2. Operator Aritmetika

Operator aritmetika adalah operator yang digunakan untuk melakukan operasi perhitungan standar.

Operator	Keterangan
+	Operator untuk penjumlahan
-	Operator untuk pengurangan
*	Operator untuk perkalian
/	Operator untuk pembagian
%	Operator untuk mengetahui sisa dari pembagian (modulus)
++	Operator untuk menaikkan nilai variabel sebesar 1 ( <i>increment</i> )
--	Operator untuk menurunkan nilai variabel sebesar 1 ( <i>decrement</i> )

### 4.3. Operator Relasional

Operator relasional digunakan untuk membandingkan dua buah nilai. Hasil dari operasi perbandingan akan menghasilkan nilai `true` atau `false`.

Operator	Keterangan
<code>\$a == \$b</code>	Memeriksa apakah \$a sama dengan \$b
<code>\$a === \$b</code>	Menghasilkan nilai <code>true</code> jika \$a sama dengan \$b, dan keduanya bertipe sama
<code>\$a != \$b</code>	Memeriksa apakah \$a tidak sama dengan \$b
<code>\$a !== \$b</code>	Menghasilkan nilai <code>true</code> jika \$a tidak sama dengan \$b, atau tipe keduanya berbeda
<code>\$a &lt; \$b</code>	Memeriksa apakah \$a lebih kecil dari \$b
<code>\$a &gt; \$b</code>	Memeriksa apakah \$a lebih besar dari \$b
<code>\$a &lt;= \$b</code>	Memeriksa apakah \$a lebih kecil atau sama dengan \$b
<code>\$a &gt;= \$b</code>	Memeriksa apakah \$a lebih besar atau sama dengan \$b

### 4.4. Operator Logika

Operator logika digunakan untuk mengoperasikan dua buah *operand* yang bertipe *boolean*.

Operator	Keterangan
<code>&amp;&amp;</code>	Logika AND
<code>  </code>	Logika OR
<code>!</code>	Logika NOT
<code>and</code>	Logika AND
<code>or</code>	Logika OR
<code>xor</code>	Logika XOR (exclusive OR)

### 4.5. Operator Bitwise

Operator *bitwise* memiliki cara kerja yang sama dengan operator logika, tapi operasinya dilakukan bit-demi-bit (*bit= binary digit*). Dalam operasi ini bilangan bulat akan dijadikan *operand* dan akan dikonversi ke bentuk biner terlebih dahulu selanjutnya dibandingkan dengan menggunakan operator logika.

Operator	Keterangan
&	<i>Bitwise AND</i>
	<i>Bitwise OR</i>
^	<i>Bitwise XOR</i>
~	<i>Bitwise NOT</i>
<<	<i>Bitwise shift Left</i> (menggeser n bit ke arah kiri)
>>	<i>Shift right</i> (menggeser n bit ke arah kanan)

#### 4.6. Operator *String*

Operator *string* terdiri dari dua jenis. Pertama adalah operator titik (.) yang berguna menyambung *string* dan kedua adalah operator titik diikuti oleh tanda sama dengan (.=) untuk proses penyambungan *string* yang diisikan ke dalam variabel.

#### 4.7. Operator *Array*

Berikut ini adalah operasi yang dapat diterapkan dalam *array*.

Operator	Keterangan
+	Membandingkan <i>array</i>
==	Membandingkan <i>array</i> . Bernilai true jika <i>array1</i> dan <i>array2</i> memiliki nilai yang sama
===	Membandingkan <i>array</i> . Bernilai true jika <i>array1</i> dan <i>array2</i> memiliki nilai yang sama serta urutan dan tipe datanya sama
!=	Membandingkan <i>array</i> . Bernilai true jika <i>array1</i> tidak sama dengan <i>array2</i>
<>	Membandingkan <i>array</i> . Bernilai true jika <i>array1</i> tidak sama dengan <i>array2</i>
!==	Membandingkan <i>array</i> . Bernilai true jika <i>array1</i> tidak identik dengan <i>array2</i>

#### 4.8. Operator *Tipe*

PHP menyediakan operator **instanceof** untuk memeriksa tipe dari suatu objek, apakah merupakan *instance* dari kelas tertentu atau tidak.

Contoh:

```
<?php
class A {
    # badan kelas A
}
class B {
    # badan kelas B
}

$a= new A();
$b= new B();

$a instanceof A? print "true\n" : print "false\n";
$b instanceof B? print "true\n" : print "false\n";

?>
```

#### 4.9. Operator Kondisional

PHP juga mendukung operator kondisional (sering disebut juga sebagai operator *ternary*). Operator ini merupakan bentuk singkat dari struktur pemilihan if-else.

Contoh:

```
$b instanceof B? print "true\n" : print "false\n";
```

Perintah diatas juga dapat ditulis sebagai berikut.

```
If ($b instanceof B) {
    print "true\n";
} else {
    print "false\n";
}
```

#### 4.10. Operator Eksekusi

PHP mendukung operator eksekusi, disebut *backtick* (```). PHP akan mengeksekusi perintah baris (*shell command*) yang ditulis di dalam *backtick* dan menampung hasilnya ke dalam variabel.

Tugas

Buat sebuah program untuk menghitung penjumlahan, pengurangan, perkalian, dan pembagian menggunakan PHP.