

## Modul Pemrograman Web

# PHP: Variabel, Tipe Data, Form Parameter

## Variabel PHP

**Variabel** adalah tempat untuk menyimpan data atau informasi yang akan diproses, yang berada di dalam memori komputer. **Nilai variable dapat berubah-ubah** dan data yang disimpan **bersifat sementara** dengan kata lain akan hilang setelah program selesai dieksekusi.

Pada penulisan variabel PHP terdapat beberapa aturan sebagai berikut.

- 1) Sebuah variable dimulai dengan tanda dollar (\$), diikuti dengan nama variable
- 2) Nama variable tidak boleh mengandung spasi
- 3) Nama variable tidak boleh diawali dengan angka
- 4) Selain tanda \$ yang terdapat di depan nama, variable tidak boleh menggunakan karakter simbol kecuali tanda \_ (Underscore)
- 5) Variabel di dalam PHP bersifat **case sensitive**, sehingga perbedaan huruf kapital dianggap sebagai dua variable yang berbeda (**\$age** ≠ **\$AGE**).

Contoh-contoh penulisan variabel:

Benar	Salah	Keterangan Salah
\$mahasiswa	\$maha siswa	Karena menggunakan spasi
\$_nim	-\$nim	Karena dimulai dengan karakter
\$kelas01	\$01kelas	Karena dimulai dengan angka
\$mata_kuliah	\$mata.kuliah	Karena dipisahkan dengan karakter titik
\$JenisKelamin	\$Jenis Kelamin	Karena dipisahkan dengan spasi

### Praktikum 9.1

Buat file dengan nama *praktikum9.php* di folder htdocs Anda.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Praktikum 9</title>
</head>
<body>
    <?php
    $txt = "Hello World";
    $x = 5;
    $y = 10.5;
    $Prodi = "Sistem Informasi";

    echo "Tampilkan variabel txt : $txt";
    echo "<br>";
    echo "$x <br>";
    echo "$y <br>";
    echo "Prodi di Fakultas Ilmu Komputer : $Prodi";
    ?>
</body>
</html>
```

## Konstanta

Konstanta adalah variabel yang **nilainya tetap**. Berbeda dengan variabel, Konstanta hanya diberi nilai pada awal program dan nilainya tidak pernah berubah selama program berjalan. Pendefinisian nilai konstanta dilakukan dengan menggunakan fungsi **define()**.

PHP telah mendefinisikan beberapa konstanta, misalnya : **PHP\_VERSION** yaitu konstanta yang memberikan informasi tentang versi PHP yang digunakan. Selain konstanta yang telah disediakan oleh PHP, kita juga dapat membuat konstanta sendiri. Aturan penulisan konstanta adalah sebagai berikut.

```
define ("NAMA_KONSTANTA","nilai_konstanta");
```

Perhatikan bahwa dalam pernyataan diatas, NAMA\_KONSTANTA tidak diawali dengan karakter (\$), karena karakter dollar hanya digunakan untuk sebuah variabel.

### Praktikum 9.2

Ganti code PHP di file *praktikum9.php* menggunakan code berikut.

```
<?php
//Mendefinisikan nilai konstanta
define("nilaimax", "100");
define("pi", "3.14");

//Mencetak nilai konstanta
print("Nilai maksimal : ".nilaimax);

//Menggunakan konstanta dalam perhitungan
$jarijari = 5;
$luaslingkaran = pi*$jarijari*$jarijari;
echo "<br> Luas lingkaran = ".$luaslingkaran;
?>
```

## Tipe Data di PHP

Pada PHP, tipe data variable tidak didefinisikan oleh programmer, akan tetapi secara otomatis ditentukan oleh interpreter PHP. PHP memiliki 5 tipe data diantaranya:

- 1) Integer (bilangan bulat)
- 2) Floating Point (bilangan pecahan atau decimal)
- 3) String (ditandai dengan adanya penggunaan tanda petik tunggal atau ganda)
- 4) Array (mengandung data dalam satu variabel index)
- 5) Object (ber-paradigma kepada object)

Contoh-contoh penulisan tipe data:

Tipe Data	Keterangan
Integer	\$tahun = 2009; \$umur = 20;
Floating-point	\$ipk = 3.75; \$rerata = 7.5;
String	\$nama = "Farida"; \$jurusan = "Sistem Informasi"
Boolean	Menghasilkan nilai <i>true</i> dan <i>false</i>

PHP mendukung *Object-Oriented Programming (OOP)*, yang dapat memudahkan dalam proses debugging, maintenance, dan penggunaan ulang kode (*reusable code*). Di dalam OOP, kelas merupakan unit terpenting. Kelas adalah suatu definisi dari struktur yang memiliki properti dan metode (fungsi). Dalam PHP, kelas didefinisikan menggunakan kata kunci `class`.

Contoh:

```
class Person{
    var $nama = "";
    function nama ($newname = NULL) {
        if(!is_null($newname)){
            $this->nama = $newname;
        }
        return $this->nama;
    }
}
```

### **Praktikum 9.3**

Ganti code PHP di file *praktikum9.php* menggunakan code berikut.

`<?php`

`//Mendefinisikan variabel dengan berbagai macam tipe data`

`$nim = "21.11.0000";`

`$nama = "Feby Rastanti";`

`$umur = 21;`

`$nilai = 92.2;`

`$status = TRUE;`

`//Menampilkan data`

`echo "NIM : ".$nim."<br>";`

`echo "Nama : $nama <br>";`

`print "Umur : ".$umur."<br>";`

`printf ("Nilai : %.3f<br>", $nilai);`

`if($status)`

`echo "Status : Aktif";`

`else`

`echo "Status : Tidak Aktif";`

`?>`

## Operator dan Aritmatika

Operator adalah symbol yang digunakan untuk memanipulasi data, seperti penambahan, pengurangan, perkalian, perbandingan, atau penugasan. Operator dikelompokkan dalam 4 kategori, yaitu:

- 1) Operator Aritmatika adalah operator yang berhubungan dengan fungsi matematika.
- 2) Operator Relasional adalah operator perbandingan yang digunakan untuk membandingkan dua buah nilai.
- 3) Operator logika adalah operator yang membandingkan TRUE dan FALSE.
- 4) Operator yang lainnya yang tidak termasuk dengan operator-operator di atas.

Contoh penulisan operator:

Jenis Operator	Operator	Contoh	Keterangan
Aritmatika	+	<b>\$a + \$b</b>	Penjumlahan
	-	<b>\$a - \$b</b>	Pengurangan
	*	<b>\$a * \$b</b>	Perkalian
	/	<b>\$a / \$b</b>	Pembagian
	%	<b>\$a % \$b</b>	Modulus, sisa habis bagi
	++	<b>\$a++</b>	<i>Increment</i> , menaikkan nilai \$a sebesar 1
	--	<b>\$a--</b>	<i>Decrement</i> , menurunkan nilai \$a sebesar 1
Relasional	==	<b>\$a == \$b</b>	Memeriksa apakah \$a sama dengan \$b
	!=	<b>\$a != \$b</b>	\$a Tidak sama dengan \$b
	<	<b>\$a &lt; \$b</b>	\$a kurang dari \$b
	>	<b>\$a &gt; \$b</b>	\$a lebih dari \$b
	<=	<b>\$a &lt;= \$b</b>	\$a kurang dari sama dengan \$b
	>=	<b>\$a &gt;= \$b</b>	\$a lebih dari sama dengan \$b
Logika	&&	<b>\$a &amp;&amp; \$b</b>	TRUE jika \$a and \$b TRUE
		<b>\$a    \$b</b>	TRUE jika \$a or \$b TRUE
	!	<b>!\$a</b>	TRUE jika \$a FALSE
String	.	<b>\$a.\$b</b>	Penggabungan string \$a dan \$b
Penugasan	=	<b>\$a = 5</b>	Variabel \$a bernilai 5

### Praktikum 9.4

Ganti code PHP di file *praktikum9.php* menggunakan code berikut.

```
<?php
$angka1=15;
$angka2=3;

$tambah = $angka1 + $angka2;
$kurang = $angka1 - $angka2;
$perkalian = $angka1 * $angka2;
$pembagian = $angka1 / $angka2;
$modulus = $angka1 % $angka2;

echo "<h2> Perhitungan Aritmatika </h2>";
echo "$angka1 + $angka2 = $tambah";
echo "<br>$angka1 - $angka2 = $kurang";
echo "<br>$angka1 * $angka2 = $perkalian";
echo "<br>$angka1 / $angka2 = $pembagian";
echo "<br>$angka1 % $angka2 = $modulus";
?>
```

### Praktikum 9.5

Ganti code PHP di file *praktikum9.php* menggunakan code berikut.

```
<?php
$string1 = "Pemrograman";
$string2 = "Web Lanjut";
$string3 = "Semester Ganjil";

echo $string1." ".$string2." ".$string3;
?>
```

## Operator Assignment/Penugasan

Operator Assignment atau bisa disebut Operator penugasan adalah operator yang digunakan untuk memberi nilai ke suatu variabel ke variabel lain. Simbol operator ini adalah (=). Pada penggunaannya operator Assignment akan digabungkan dengan operator Aritmatik. Berikut ini daftar operator yang termasuk ke dalam operator assignment dalam PHP:

Operator	Operasi / Fungsi	Contoh
+=	Menambahkan nilai di kiri dengan nilai di kanan	a += 4 a = a+4
-=	Mengurangi nilai di kiri dengan nilai di kanan	a -= 3 a = a-3
.=	Melakukan concatenation / operasi gabungan antara nilai di kiri dengan nilai di kanan	a .= "tahun" a = a."tahun"
/=	Membagi nilai di kiri dengan nilai di kanan	a /= 3 a = a/3
%=	Sisa hasil bagi antara nilai di kiri dengan nilai di kanan	a %= 4 a = a%4
!=	Melakukan operasi OR antara nilai di kiri dengan di kanan	a != 5 a = a!5
^=	Melakukan operasi XOR antara nilai di kiri dengan nilai di kanan	a ^= 5 a = a^5

## Operator Increment dan Decrement

Operator Increment (++) dan Decrement (--) berfungsi untuk menambahkan dan mengurangi nilai dengan selisih 1 (satu). Tabel berikut ini adalah daftar operator yang termasuk ke dalam operator Increment dan Decrement dalam PHP:

Operator	Nama	Keterangan
++\$a	<i>Pre-increment</i>	\$a ditambah satu, kemudian operasi dijalankan
\$a++	<i>Post-increment</i>	Operasi dijalankan, kemudian \$a ditambah satu
--\$a	<i>Pre-decrement</i>	\$a dikurangi satu, kemudian operasi dijalankan
\$a--	<i>Post-decrement</i>	Operasi dijalankan, kemudian \$a ditambah satu

## Mengenal Struktur Dasar Form HTML

Mengenal Struktur dasar form HTML :

- 1) Pada struktur dasar sebuah Form HTML, terdapat 2 (dua) buah atribut yaitu:
  - a. Atribut **ACTION**. Atribut action ini diisi dengan nilai berupa alamat halaman PHP di mana anda akan memproses isi form tersebut.
  - b. Atribut **METHOD**. Atribut method menentukan bagaimana cara form "dikirim" ke dalam halaman action. Nilai dari atribut method hanya bisa

diisi dengan 1 dari 2 pilihan method, yaitu GET dan POST. Nilai dari atribut method ini juga akan mempengaruhi cara anda memproses nilai dari form.

- 2) Setelah membuat tag pembuka form dengan atribut action dan method, didalam sebuah form biasanya diisi dengan tag `<input type = "text">` yang akan menampilkan kotak isian form.
- 3) Hal yang paling penting untuk diperhatikan adalah atribut name dari masing-masing tag `<input>`. Nilai dari name inilah yang menjadi penanda masing-masing objek form agar dapat diproses dengan PHP.
- 4) Objek form lain yang harus tambahkan adalah tombol submit yang apabila diklik akan mengirimkan data dari form ke halaman action untuk diproses. Atribut penting di sini adalah atribut `type="submit"`, yang akan otomatis mengirim isian form ketika tombol ini diklik.
- 5) Berikut ini contoh form HTML yang berisi 2 tag `<input>` dan 1 tag `submit/button`:

## **Praktikum 9.6**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Praktikum 9</title>
</head>
<body>
  <form action="action.php" method="GET">
    User : <input type="text" name="user" id=""><br>
    Password: <input type="password" name="pass" id=""><br>
    <input type="submit" value="PROSES">
  </form>
</body>
```



`</html>`

## Method (POST dan GET)

PHP memiliki 2 macam method yang bisa digunakan untuk mengirim data dari client (browser) ke server, yaitu POST dan GET. **Method POST** digunakan untuk mengumpulkan nilai variabel dari sebuah form yang dikirimkan menggunakan metode POST. **Method GET** adalah sebuah fungsi yang digunakan untuk mengumpulkan nilai dari sebuah form yang dikirimkan menggunakan metode GET. Informasi yang dikirimkan dari sebuah form menggunakan metode GET dapat di lihat semua orang, maksudnya adalah nilai dari form tersebut dapat dengan jelas terlihat pada jendela address bar pada browser. Metode GET tidak cocok untuk nilai variabel yang sangat besar dan tidak bisa menyimpan variabel nilainya lebih dari 2000 karakter. Untuk dapat melihat perbedaannya, pada praktikum 9.7 dan action.php semua `method = 'POST'` ubahlah menjadi `method "GET"`.

### Praktikum 9.7

Buat 2 file untuk praktikum 9.7 dengan nama prak9.php dan action.php . Hasil output akan ditampilkan pada halaman baru.

File prak9.php:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Praktikum 9</title>
</head>
<body>
  <form method="POST" action="action.php">
```

```
Nama : <input type="text" name="nama"> <br>
Email : <input type="text" name="email"> <br>
Pesan : <textarea name="pesan"> </textarea> <br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

*File action.php:*

```
<?php
$nama = $_POST['nama'];
$email = $_POST['email'];
$pesan = $_POST['pesan'];
echo "Nama : <b>$nama</b> <br>";
echo "E-mail : <b>$email</b> <br>";
echo "Pesan : <b>$pesan</b> <br>";
?>
```

## **Praktikum 9.8**

Buat file untuk praktikum 9.8 dengan nama prak9-2.php. Hasil output akan berada didalam satu halaman yang sama.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Praktikum 9</title>
</head>
<body>
<form method="POST" action="prak9-2.php" >
  Nama Anda = <input type="text" name="namaku"><br>
  Email Anda = <input type="email" name="emailku"><br>
  Alamat Anda = <textarea name="alamatku">
  </textarea><br>
```

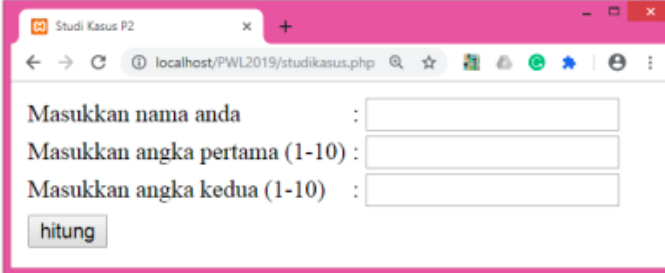
```
<input type="submit" name="kirim" value="sent">
</form>
<?php
    if (isset($_POST ['kirim'])) {
        echo "Nama Anda : $_POST[namaku]<br>";
        echo "Email Anda : $_POST[emailku]<br>";
        echo "Alamat Anda : $_POST[alamatku]<br>";
    }
?>
</body>
</html>
```

## CHALLENGE pertemuan 9

---

Pada praktikum pertemuan 9 kali ini, kita akan membuat website yang menggunakan dasar PHP:

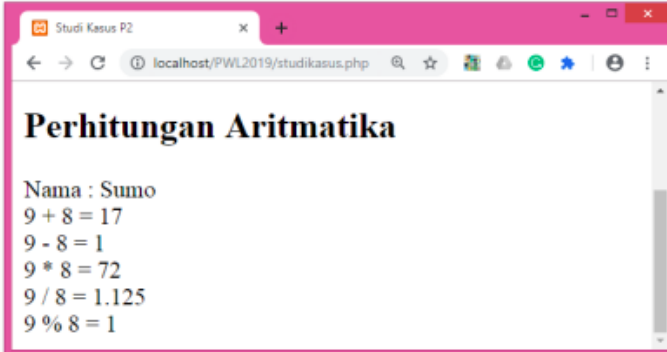
1. Buka text editor VSCode.
2. Simpan terlebih dahulu file tersebut dengan nama project dengan nama Challenge9\_NIM Lengkap. Contoh: Challenge9\_20112020.php  
(Note. Gunakan ketentuan ini untuk semua file praktikum)
3. Buatlah sebuah form dinamis yang dapat menjalankan operator aritmatika, seperti gambar di bawah ini.



The screenshot shows a web browser window with the address bar displaying 'localhost/PWL2019/studikasukas.php'. The page contains a form with three input fields and a button. The labels for the input fields are 'Masukkan nama anda', 'Masukkan angka pertama (1-10)', and 'Masukkan angka kedua (1-10)'. The button is labeled 'hitung'.



The screenshot shows the same web browser window, but the form is now filled with data. The 'Masukkan nama anda' field contains 'Sumo', the 'Masukkan angka pertama (1-10)' field contains '9', and the 'Masukkan angka kedua (1-10)' field contains '8'. The 'hitung' button is still visible.



The screenshot shows the web browser window displaying the results of the arithmetic calculations. The title of the page is 'Perhitungan Aritmatika'. The results are listed as follows:  
Nama : Sumo  
 $9 + 8 = 17$   
 $9 - 8 = 1$   
 $9 * 8 = 72$   
 $9 / 8 = 1.125$   
 $9 \% 8 = 1$