

## PERTEMUAN 2

### Pokok Bahasan :

- ◆ Register Global On vs Register Global Off
- ◆ Variabel dan Konstanta
- ◆ Tipe Data
- ◆ Operator dan Aritmetika
- ◆ Parameter

#### 1.1 Register Global On vs Register Global Off

Pada awal perkembangannya, secara default register global di set On dengan tujuan untuk memberi kemudahan pengguna dalam mempelajari PHP, sehingga pengguna dapat mengambil/memanggil variable cukup dengan menyebutkan nama variabelnya, tidak peduli apakah variable tersebut berasal dari GET, POST, COOKIE, atau SESSION. Contoh skrip berikut :

Contoh penggunaan register global on :

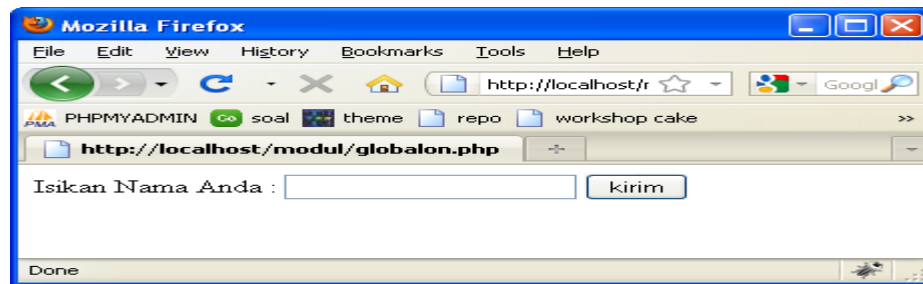
File : globalon.php

```
<?php
    echo "<form method=GET action=globalon.php>
    Isikan Nama Anda : <input type=text name='nama'>
    <input type=submit name=oke value=kirim></form>";
    if ($oke){
        echo "Nama Anda Adalah : <b>$nama</b>";
    }
?>
```

Ketentuan:

Jadi kalau kita menggunakan setting register global = ON, maka untuk mengambil data variable cukup menuliskan variable formnya saja tanpa perlu tahu berasal dari mana variable tersebut. Kelemahannya : misalkan script ini dijalankan di server di mana register global = OFF, maka script itu tidak akan berjalan.

Output :



Namun sejak versi PHP 4.3.X ke atas, maka default nilai register global = Off, sehingga apabila skrip globalon.php dijalankan, maka tidak akan menghasilkan apa-apa.

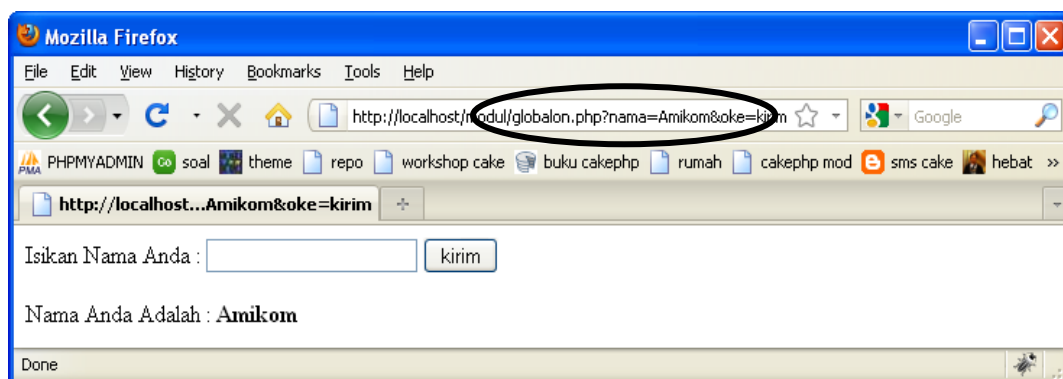
Sehingga agar skripnya dapat berjalan pada register globals = Off, maka perlu disesuaikan variabelnya menjadi seperti berikut ini :

Contoh penggunaan register global off :

File : globalon.php

```
<?php
echo "<form method=GET action=globalon.php>
Isikan Nama Anda : <input type=text name='nama'>
<input type=submit name=oke value=kirim></form>";
if (isset($_GET['oke'])) {
echo "Nama Anda Adalah : <b>$_GET[nama]</b>";
}
?>
```

Output :



Tips :

- Perhatikan sumber dari variable, apabila menggunakan method/parameter POST, maka pemanggilan variabelnya : `$_POST['nama_varibel']`.

- Apabila sumber dari variable diambil dari variable yang dibawa Address bar, misalnya : `http://localhost/modul/globalon.php?nama=Amikom` , maka penulisan variabelnya : `$_GET['nama_variabel']`.
- Sintaks variable seperti `$_ENV`, `$_GET`, `$_POST`, `$_COOKIE`, `$_SESSION`, `$_FILES` dan `$_SERVER` juga sering disebut dengan superglobal atau autoglobal, artinya variable tersebut dapat berjalan dimanapun, tidak peduli apakah register globalnya On atau Off.

## 1.2 Variabel PHP

Variabel adalah sebuah tempat di memori untuk menyimpan data yang nilainya dapat berubah-ubah selama program dijalankan. Penulisan variabel didalam pemrograman PHP, ada aturan tertentu yang harus kita ketahui, diantaranya :

1. Variabel dimulai dengan tanda dollar ( \$ ).
2. Harus dimulai dengan huruf atau underscore ( \_ ).
3. Tidak boleh mengandung spasi
4. Case sensitive atau huruf besar dan huruf kecil berbeda.
5. Karakter pertama setelah simbol dollar, tidak boleh menggunakan angka (harus huruf)
6. Setelah simbol dollar dan huruf, maka karakter selanjutnya boleh menggunakan angka

Contoh penulisan variabel :

Benar	Salah
<code>\$variabel</code>	<code>\$var!abel</code>
<code>\$_nim</code>	<code>\$-nim</code>
<code>\$kelas01</code>	<code>\$01kelas</code>
<code>\$tepat_waktu</code>	<code>\$tepat-waktu</code>

Contoh penggunaan variabel :

File : lat01.php

```
<html>
<head>
  <title>Menggunakan Variabel</title>
</head>
<body>
  <?php
    $kampus ="STMIK AMIKOM Yogyakarta";
    echo "Kampus Terpadu : $kampus";
  ?>
</body>
</html>
```

Contoh variabel yang bernilai dinamis :

File : lat02.php

```
<html>
<head>
  <title>Menggunakan Variabel Dinamis </title>
</head>
<body>
  <form action="lat02.php" method="post">
    Masukkan Nama Anda = <input type="text" name="nama" /><br>
    <input type="submit" name="submit" value="simpan" />
  </form>
  <?php
    If(isset($_POST['sumbit'])) {
      echo "Nama Anda : $_POST[nama]";
    }
  ?>
</body>
</html>
```

### 1.3 Konstanta

Konstanta adalah variabel yang nilainya tetap. Konstanta hanya diberi nilai pada awal program dan nilainya tidak pernah berubah selama program berjalan. 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")
```

Contoh penggunaan konstanta :

File : lat03.php

```
<html>
<head>
  <title>Menggunakan konstanta</title>
</head>
<body>
  <?php
    define("KAMPUS", "STMIK AMIKOM YOGYAKARTA");
    define("SLOGAN", "TEMPAT KULIAH ORANG BERDASI");
    print "KAMPUS : ".KAMPUS;
    echo "<br>SLOGAN : ".SLOGAN;
  ?>
</body>
</html>
```

## 1.4 Tipe Data di PHP

Pada PHP, tipe data variabel 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 desimal)
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 penulisan tipe data :

Tipe Data	Contoh	keterangan
Integer	\$jumlah = 23; \$nilai = -3;	Bilangan bulat
Double	\$ipk = 3.85; \$total = 13.00;	Bilangan real
String	\$jurusan = "Sistem Informasi"; \$jenjang = "Strata Satu";	Karakter, kalimat

Contoh penggunaan tipe data :

File : lat04.php

```
<?php
    $nim = "09.12.3508";
    $nama = "M.Rezza Ardava";
    $umur = 21;
    $nilai = 90.25;
    $status = TRUE;
    echo "NIM : " . $nim. "<br>";
    echo "Nama : $nama<br>";
    print "Umur : " . $umur;
    print "<br>";
    printf ("Nilai : %.3f<br>", $nilai);
    if ($status)
    echo "Status : Aktif";
    else
    echo "Status : Tidak Aktif";
?>
```

## 1.5 Operator dan Aritmetika

Operator adalah simbol yang digunakan untuk memanipulasi data, seperti penambahan, pengurangan, perkalian, perbandingan, atau penugasan.

Operator dapat dikelompokkan dalam 4 kategori, yaitu :

1. Operator Aritmetika adalah operator yang berhubungan dengan fungsi matematika.
2. Operator Logika adalah operator yang membandingkan TRUE dan FALSE.
3. Operator Bitwise adalah operator yang membandingkan binary.
4. Ada juga operator yang sering digunakan namun tidak termasuk dalam kelompok di atas, kita dapat mempelajarinya dalam kelompok lain.

Contoh penulisan Operator :

Jenis Operator	Operator	Contoh	Keterangan
<b>Aritmatika</b>	+	\$a + \$b	Pertambahan
	-	\$a - \$b	Pengurangan
	*	\$a * \$b	Perkalian
	/	\$a / \$b	Pembagian
	%	\$a % \$b	Modulus, sisa pembagian
<b>Penugasan</b>	=	\$a = 7;	\$a diisi dengan 4
<b>Bitwise</b>	&	\$a & \$b	Bitwise AND
		\$a   \$b	Bitwise OR
	^	\$a ^ \$b	Bitwise XOR (Exclusive OR)
	~	\$a ~ \$b	Bitwise NOT
	<<	\$a << \$b	Pergeseran bit ke kiri
	>>	\$a >> \$b	Pergeseran bit ke kanan
<b>Perbandingan</b>	==	\$a == \$b	Sama dengan
	===	\$a === \$b	Identik
	!=	\$a != \$b	Tidak sama dengan
	<>	\$a <> \$b	Tidak sama dengan
	!==	\$a !== \$b	Tidak identik
	<	\$a < \$b	Kurang dari
	>	\$a > \$b	Lebih dari
	<=	\$a <= \$b	Kurang dari sama dengan
	>=	\$a >= \$b	Lebih dari sama dengan

<b>Logika</b>	and	\$a and \$b	TRUE jika \$a dan \$b TRUE
	&&	\$a && \$b	TRUE jika \$a dan \$b TRUE
	or	\$a or \$b	TRUE jika \$a atau \$b TRUE
		\$a    \$b	TRUE jika \$a dan/atau \$b TRUE
	xor	\$a xor \$b	TRUE jika \$a atau \$b TRUE, tapi tidak keduanya
	!	!\$a	TRUE jika \$a FALSE
<b>String</b>	.	\$a . \$b	Penggabungan string \$a dan \$b

Contoh penggunaan operator :

File : lat05.php

```
<?php
$angka1 = 12;
$angka2 = 3;

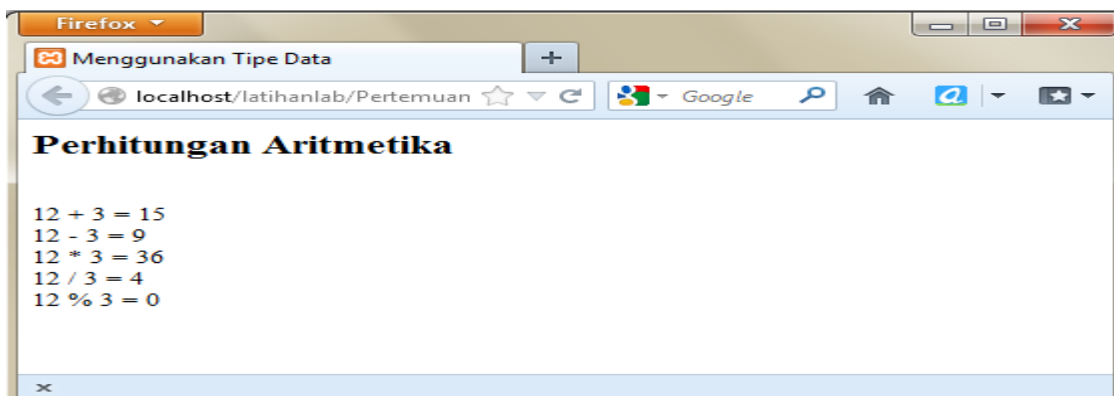
$tambah = $angka1 + $angka2;
$kurang = $angka1 - $angka2;
$perkalian = $angka1 * $angka2;
$bagi = $angka1 / $angka2;
$modulus = $angka1 % $angka2;

echo "<h2>Perhitungan Aritmetika</h2>";

echo "$angka1 + $angka2 = $tambah";
echo "<br>$angka1 - $angka2 = $kurang";
echo "<br>$angka1 * $angka2 = $perkalian";
echo "<br>$angka1 / $angka2 = $bagi";
echo "<br>$angka1 % $angka2 = $modulus";

?>
```

Output dari lat05.php :



## 1.6 Parameter (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.

Contoh penggunaan parameter method POST :

File : lat06.php

```
<html>
<head>
    <title> Form Input </title>
</head>
<body>

<form method="POST" action="lat07.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 : lat07.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>";
?>
```

Method GET adalah sebuah fungsi yang digunakan untuk mengumpulkan nilai dari sebuah form yang dikirimkan menggunakan metode GET. Informasi yang di kirimkan dari sebuah form menggunakan metode GET dapat di lihat semua orang, maksudnya adalah nilai dari form tersebut dapat terlihat dengan jelas 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 melihat perbedaanya, pada lat06.php dan lat07.php semua method="POST" menjadi method="GET".



## Referensi

Arief, M. R. 2011. *Pemrograman Web Dinamis menggunakan PHP dan MySQL*. Yogyakarta: Andi Offset.

Solichin, Achmad. *Modul Pemrograman Web dengan PHP dan MySQL*.