

Modul 2. Variabel, Tipe Data dan Operator

A. Tujuan

1. Memahami ruang lingkup variabel
2. Memahami tipe data pada variabel di PHP
3. Memahami penggunaan operator matematika

B. Dasar Teori

Ruang lingkup variabel

Variable memiliki tiga ruang lingkup yang berbeda:

- local
- global
- static

Sebuah variabel yang dideklarasikan di luar sebuah fungsi memiliki **ruang lingkup global** dan hanya bisa diakses di luar dari sebuah fungsi.

Sebuah variabel yang dideklarasikan di dalam sebuah fungsi memiliki **ruang lingkup local** dan hanya bisa diakses di dalam fungsi tersebut.

Kata kunci **global** digunakan untuk mengakses variabel global di dalam sebuah fungsi

Secara umum ketika sebuah fungsi sudah dieksekusi atau dikerjakan, semua variabel dihapus. Jika kita menginginkan variabel tersebut untuk digunakan maka menggunakan kata kunci **static**

Konstanta

Konstanta, menyatakan nilai yang tetap di dalam program. Syntaxnya adalah:

```
define (string $name , $value [, bool $case_insensitive = false ] )
```

Tipe Data

Pada PHP terdapat beberapa tipe data antara lain :

- Integer, menyatakan tipe data bilangan bulat
- Double , menyatakan tipe data bilangan real
- String, menyatakan tipe data teks/deret karakter

Operator

Operator adalah suatu simbol yang dipakai untuk memanipulasi nilai suatu variabel. Variabel yang nilainya dimodifikasi oleh operator disebut operand.

Misalnya 3 - 2, 3 dan 2 adalah operand dan - adalah operator

Berikut ini adalah tabel operator aritmatik, makna dan contohnya:

Operator	Makna	Contoh
+	Penjumlahan	2 + 4
-	Pengurangan	6 - 2
*	Perkalian	5 * 3
/	Pembagian	15 / 3
%	Modulus	43 % 10

Komparasi (Perbandingan)

Perbandingan digunakan untuk menguji hubungan antara nilai dan atau variabel. Operator ini digunakan dalam suatu statement bersyarat yang selalu menghasilkan nilai TRUE atau FALSE.

Misalkan:

\$x = 4; \$y = 5;

Maka operator perbandingan dan hasilnya adalah sebagai berikut

berikut ini adalah beberapa contoh penggunaan operator perbandingan dan hasilnya.

Operator	Makna	Contoh	Hasil
==	Sama dengan	\$x == \$y	FALSE
!=	Tidak sama dengan	\$x != \$y	TRUE
<	Lebih kecil dari	\$x < \$y	TRUE
>	Lebih besar dari	\$x > \$y	FALSE
<=	Lebih kecil atau sama dengan dari	\$x <= \$y	TRUE
>=	Lebih besar atau sama dengan dari	\$x >= \$y	FALSE

Increment dan Decrement

Dalam pemrograman seringkali dijumpai proses yang melibatkan proses increment.

Misalkan proses increment dengan tingkat kenaikan 1, maka perintah yang dituliskan dapat berupa

\$counter = \$counter + 1;

dalam PHP, perintah di atas dapat ditulis dalam satu perintah singkat sebagai

\$counter += 1;

Dari contoh di atas tampak bahwa operator yang digunakan (+=) merupakan gabungan dari operator aritmatik dan assignment. Berikut ini adalah bentuk-bentuk operator lain jenis ini.

Operator	Contoh	Operasi yang ekuivalen
+=	<code>\$x += 2;</code>	<code>\$x = \$x + 2;</code>
-=	<code>\$x -= 4;</code>	<code>\$x = \$x - 4;</code>
*=	<code>\$x *= 3;</code>	<code>\$x = \$x * 3;</code>
/=	<code>\$x /= 2;</code>	<code>\$x = \$x / 2;</code>
%=	<code>\$x %= 5;</code>	<code>\$x = \$x % 5;</code>
.=	<code>\$my_str.="hello";</code>	<code>\$my_str = \$my_str . "hello";</code>

Operator Pre/Post Increment dan Decrement

Operator jenis ini merupakan pengembangan dari operator jenis sebelumnya. Operator ini hanya digunakan pada proses increment maupun decrement dengan tingkat 1. Berikut ini adalah operator yang termasuk jenis ini:

- `$x++`; ekuivalen dengan `$x += 1`; atau `$x = $x + 1`;
- `$x--`; ekuivalen dengan `$x -= 1`; atau `$x = $x - 1`;

Contoh:

```
<?php
$x = 4;
$x++;
echo $x;
$x = 4;
$x--;
echo $x;
?>
```

C. Praktek

Praktek 1

1. Buat file **mod2prak1.php**
2. Ketikkan kode berikut:

```
<?php
$x = 5; // global scope

function cobaGlobal() {
    echo "<p>Variabel x: $x</p>";
    echo "</br>";
}
cobaGlobal();
?>
```

3. Simpan file dan jalankan pada browser. Amati hasilnya.
4. Ubah kode menjadi:

```
<?php
$x = 5; // global scope

function cobaGlobal() {
    echo "<p>Variabel x: $x</p>";
    echo "</br>";
}
```

```

}
cobaGlobal();

function cobaLocal() {
    $y = 10; //local scope
    echo "<p>Cetak variabel y di dalam fungsi: $y</p>";
    echo "</br>";
}
cobaLocal();
echo "<p> Cetak variabel y dari luar fungsi: $y</p>";
?>

```

5. Simpan file dan jalankan pada browser. Amati hasilnya.

6. Ubah kode menjadi

```

<?php
$x = 5; // global scope

function cobaGlobal() {
    echo "<p>Variabel x: $x</p>";
    echo "</br>";
}
cobaGlobal();

function cobaLocal() {
    $y = 10; //local scope
    echo "<p>Cetak variabel y di dalam fungsi: $y</p>";
    echo "</br>";
}
cobaLocal();
echo "<p> Cetak variabel y dari luar fungsi: $y</p>";

function cobaGlobal2() {
    global $x;
    echo "<p>Variabel x: $x</p>";
    echo "</br>";
}
cobaGlobal2();

?>

```

7. Simpan file dan jalankan pada browser. Amati hasilnya.

8. Ubah kode menjadi

```

<?php
$x = 5; // global scope

function cobaGlobal() {
    echo "<p>Variabel x: $x</p>";
    echo "</br>";
}
cobaGlobal();

function cobaLocal() {
    $y = 10; //local scope
    echo "<p>Cetak variabel y di dalam fungsi: $y</p>";
    echo "</br>";
}
cobaLocal();
echo "<p> Cetak variabel y dari luar fungsi: $y</p>";

```

```

function cobaGlobal2() {
    global $x;
    echo "<p>Variabel x: $x</p>";
    echo "</br>";
}
cobaGlobal2();

function cobaStatic() {
    static $z=0;
    echo $z;
    $z++;
    echo "</br>";
}
cobaStatic ();
cobaStatic ();
cobaStatic ();

?>

```

9. Simpan file dan jalankan pada browser. Amati hasilnya.

10. Ubah kode menjadi

```

<?php
$x = 5; // global scope

function cobaGlobal() {
    echo "<p>Variabel x: $x</p>";
    echo "</br>";
}
cobaGlobal();

function cobaLocal() {
    $y = 10; //local scope
    echo "<p>Cetak variabel y di dalam fungsi: $y</p>";
    echo "</br>";
}
cobaLocal();
echo "<p> Cetak variabel y dari luar fungsi: $y</p>";

function cobaGlobal2() {
    global $x;
    echo "<p>Variabel x: $x</p>";
    echo "</br>";
}
cobaGlobal2();

function cobaStatic() {
    static $z=0;
    echo $z;
    $z++;
    echo "</br>";
}
cobaStatic ();
cobaStatic ();
cobaStatic ();

define("CONSTANT", "Hello world.");
echo CONSTANT; // "Hello world."
echo "</br>";

```

```

echo Constant; // Menampilkan tulisan "Constant".
echo "</br>";

define("PHI", 3.14, true);
echo PHI; // 3.14
echo "</br>";
echo pHI; // 3.14
echo "</br>";
echo "Keliling lingkaran dengan jari-jari 5 adalah ".(2*PHI*5);
echo "</br>";
?>

```

11. Simpan file dan jalankan pada browser. Amati hasilnya.

Praktek 2

1. Buat file **mod2prak2.php**

```

<html>
<head>
<title>Contoh Konversi dengan cast</title>
</head>
<body>

<?php
$berat = "40.5 kg";
echo"Type String : $berat <br>";
echo"Type Double :"; echo (double) $berat;
echo"<br>";
echo"Type Integer:"; echo (int) $berat; echo"<br>";
echo"Type String : $berat <br>";
$bilangan = 28;
$berat = (string) $bilangan; //konversi ke string
echo "Bil ke string : $berat";
?>

</body>
</html>

```

2. Simpan file dan jalankan pada browser. Amati hasilnya.

Praktek 3

1. Buat file **mod2prak3.php**

```

<?php
$x = 10;

echo "Nilai x = ".$x;
echo "</br>";
echo "Nilai x setelah increment ++x = ".$x;
echo "</br>";
echo "Nilai x = ".$x++;
echo "</br>";
echo "Nilai x setelah increment x++ = ".$x;
echo "</br>";

?>

```

2. Simpan file dan jalankan pada browser. Amati hasilnya.

D. Tugas

Buatlah file php yang menampilkan pada browser sebagai berikut:

Nilai:

x = 5

y = 7

Operator	Kondisi	Hasil
x < y	5 < 7	BENAR
x > y	5 > 7	SALAH
x == y	5 == 7	SALAH
x = y	5 != 7	BENAR

Catatan:

Nilai 5 dan 7 adalah sebagai nilai dari variabel x dan y. Angka pada tabel kondisi dapat berganti jika nilai dari variabel diganti. Kolom hasil menyatakan BENAR atau SALAH sesuai dengan tabel kondisi

Simpan dengan nama file *kelas_nim_nama.php*