

## Penggunaan Variable

Variable adalah tempat atau wadah untuk menyimpan suatu nilai.

- Variable diawali dengan tanda \$ dan diikuti dengan nama variabelnya.
- Nama variable sendiri harus diawali dengan huruf atau underscore
- Nama variable hanya boleh mengandung alfanumeric ( A-z, 0-9 ,dan \_ )
- Nama variable bersifat case sensitive

Contoh variable

```
<?php
```

```
    $tampil = "Sultan";  
    echo "$tampil mah bebas";
```

```
?>
```

Contoh di atas akan menampilkan output => Sultan mah bebas

Dan ini contoh variable yang akan menyebabkan error

```
<?php
```

```
    $-tampil = "Sultan";  
    echo "$tampil mah bebas";
```

```
?>
```

Kenapa error ? Karena variable **\$-tampil** terdapat simbol yang tidak boleh digunakan yaitu " - " .

Wal variable hanya boleh dengan huruf atau underscore.

## Tipe Data

Tipe data adalah jenis dari data yang digunakan pada sebuah variable .

Php mendukung tipe data sebagai berikut :

- String adalah susunan dari huruf, angka dan karakter lainnya.

string ditulis dengan diapit oleh tanda kutip (") atau tanda petik (').

Contoh

```
<?php
```

Pengulangan

```
$string = "miftah rahmat";  
var_dump($string);
```

```
?>
```

Contoh ini akan menampilkan output => **string(13)** "miftah rahmat"

Kenapa ada **string(13)** ? Itu karena perintah yang digunakan adalah **var\_dump** yang mana fungsinya untuk menampilkan nilai beserta tipe datanya ,

Dan nilai **13** itu diambil dari banyak nya huruf **miftah rahmat** ,termasuk space.

- Integer adalah tipe data angka non-desimal, dengan rentang 2,147,483,648 hingga 2,147,483,647

Contoh

```
<?php
```

```
$int = 1507;  
var_dump($int);
```

```
?>
```

Contoh ini akan menampilkan output => **int(1507)**

- Float adalah tipe data desimal,tanda desimal yang digunakan adalah tanda titik.

Contoh

```
<?php
```

```
$float = 1.507;  
var_dump($float);
```

```
?>
```

Contoh ini akan menampilkan output => **float(1.507)**

- Boolean adalah tipe data yang menyatakan dua kondisi yaitu **TRUE** dan **FALSE** .

Contoh

```
<?php
```

```
$bener =true;  
$salah =false;  
echo "benar = $bener ,salah = $salah"
```

```
?>
```

Contoh ini akan menampilkan output => **benar = 1 ,salah =**

## Pengulangan

- Array adalah tipe data yang digunakan untuk menyimpan lebih dari satu nilai data dalam satu variabel.

### Contoh

**<?php**

```
$nama = array("dadan","deden","didin","dodon");  
  
var_dump($nama);
```

**?>**

Contoh ini akan menampilkan output => array (4) { [0]=> string(5) "dadan" [1]=> string(5) "deden" [2]=> string(5) "didin" [3]=> string(5) "dodon" }

- Object adalah tipe data yang menyimpan nilai data beserta sejumlah operasi data,
- Null adalah tipe data dengan satu nilai saja

## Operator

Operator digunakan untuk melakukan operasi terhadap suatu nilai atau variabel.

Jenis-jenis operator sebagai berikut :

- Operator Assignment

**x = 5 y = 3**

Operator	Contoh	Sama dengan	Hasil
=	\$nama = "Cocok";		Cocok
+=	\$x += 5;	\$x = \$x + 5;	10
-=	\$x -= \$y;	\$x = \$x - \$y;	2
*=	\$x *= \$y;	\$x = \$x * \$y;	15
/=	\$x /= 2;	\$x = \$x / 2;	2.5
.=	\$x .= \$y;	\$x = \$x . \$y;	53
%=	\$x %= \$y;	\$x = \$x % \$y;	2

**Contoh <?php**

```
$x = 15;
```

```
$y = 7;
```

```
echo "Sebelum assignment : $x <br>";
```

```
$x += $y;
```

```
echo "Sesudah assignment : $x <br>";
```

**?>**Contoh ini akan menampilkan output => Sebelum assignment : 15 Sesudah assignment : 22

## • Operator Numerik

**x = 6**

Operator	Deskripsi	Contoh	Hasil
+	penjumlahan	\$x = \$x + 2;	8
-	pengurangan	\$x = \$x - 7;	1
*	perkalian	\$x = \$x * 3;	18
/	pembagian	\$x = \$x / 2;	3
%	modulus	\$x = \$x % 3;	0
++	increment	\$x++	7
--	decrement	\$x--	5

## • Operator String

Operator	Deskripsi	Contoh	Hasil
.	Menggabungkan 2 string (concat)	\$x = "Hallo"."World";	HalloWorld

## • Operator Pembanding

**\$x = 10;**

Operator	Deskripsi	Contoh	Hasil
==	Sama dengan	\$x == 10	true
===	Sama persis dengan(tipe data dan nilai)	\$x === "10"	false
!=	Tidak sama dengan	\$x != "10"	false
!==	Tidak sama persis dengan	\$x !== "10"	true
>	Lebih besar	\$x > 10	false
<	Lebih kecil	\$x < 10	true
>=	Lebih besar dari	\$x >= 10	true
<=	Lebih kecil dari	\$x <= 10	true

## • Operator Logika

**\$x = 6 \$y = 3**

operator	Deskripsi	Contoh	Hasil
&&	Dan	(x < 7 && y > 2 )	true
	Atau	(x == 6    y == 3 )	true
!	Tidak	!(x==y)	true

## Standar Output

Adalah perintah yang digunakan untuk mengeluarkan hasil proses komputer sehingga dapat dibaca oleh si pemakai (user).

## Pengulangan

Diantaranya adalah :

- **Echo**

Data yang di cetak adalah string dan numerik.

Contoh

```
<?php
    echo "Hello World";

?>
```

- **Print**

Data yang di cetak adalah string,numerik atau object.

Contoh

```
<?php
    print "Hello World";

?>
```

- **Print\_r**

## Konstanta

- aturan penamaannya sama seperti variable
- nama konstanta tidak boleh sama dengan keyword php,seperti echo,break,empty dll
- hanya dapat dideklarisasi satu kali
- nilainya tidak dapat dirubah
- tipe data kecuali array dan object
- case sensitive
- pendepinisian konstanta

- `define("nama_konstanta","nilai");`

Contoh

```
<?php
    define("pas","Hello World");

    echo pas;
```

?>

## Struktur kontrol

- Pengkondisian IF

Cara penulisan syintak :

```
if (expression)
    Statement
```

Contoh

```
<?php
    If ($userid == "admin"){
        echo "benar";
    }
?>
```

- pengkondisian IF ELSE

```
<?php
    If ($userid == "admin"){
        echo "benar";
    } else {
        echo "salah";
    }
?>
```

- pengkondisian IF ELSE IF

```
$nama_hari = date("l");
if($nama_hari == "Sunday"){
    echo "Minggu";
} else if($nama_hari == "Monday"){
    echo "Senin";
} else{
```

## Pengulangan

```
        echo "Selasa";  
    }  
}
```

- pengkondisian SWITCH

### Contoh

```
$nama_hari = date("l");  
  
Switch ($nama_hari)  
{  
    case "Sunday" : { echo "Minggu"; }  
                    break;  
    Case "Monday" : { echo "Senin"; }  
                    break;  
    Case "Tuesday" : { echo "Sabtu"; }  
                    break;  
    Default : echo "Sabtu";  
}
```

## Pengulangan

- While

Statement while akan menjalankan blok code secara berulang-ulang selama kondisi masih bernilai TRUE.

### Contoh

```
<?php  
    $x = 1;  
    while ($x > 20) {  
        echo "nomor : $x <br>";  
    }  
    $x++;  
?  
>
```

Pengulangan

- **Do while**

Statement do..while mirip dengan statement while yakni menjalankan kode berulang-ulang.

Contoh

```
<?php
```

```
    $x = 1;
```

```
    Do {
```

```
        echo "nomor : $x <br>";
```

```
        $x++;
```

```
    while ($x > 10);
```

```
?>
```

Perbedaan do..while dengan while adalah pada do..while blok kode dijalankan terlebih dahulu baru kemudian dilakukan pengecekan kondisi, bila kondisi bernilai TRUE maka kode akan terus dijalankan.

- **For**

Pengulangan sebanyak jumlah yang ditentukan.

Contoh

```
<?php
```

```
    for($x = 1; $x > 20; $x++);
```

```
    echo "nomor : $x <br>";
```

```
?>
```

- **foreach**

Statement foreach hanya berlaku pada variable array, digunakan untuk mengulang membaca setiap element array.

Cara penulisan syntak :

```
foreach ($array as $value) {
```

```
// code yang akan dijalankan berulang }
```

Untuk setiap iterasi, nilai dari setiap element dapat diambil dari variable \$value.

Contoh

```
<?php
```

```
    $jurusan = array(
```



Pengulangan

```
'rpl' => 'Rekayasa Perangkat Lunak',  
  
'ak' => 'Analisis Kimia',  
  
'tkj' => 'Teknik Komputer Jaringan'  
  
);
```

```
echo "jurusan di sekolah pasundan: <br />";  
  
foreach ($jurusan as $value) {  
  
    echo "$value <br>";  
  
}
```

**TERIMA KASIH**