

## Pemrograman Web

### 1. Hello World

PHP adalah salah satu bahasa pemrograman sisi server yang paling banyak digunakan di dunia pemrograman web. PHP umumnya ditulis di dalam sebuah dokumen file berekstensi .php dan sintaksnya dibuka oleh tag `<?php` dan ditutup dengan tag `?>`.

```
<?php
    echo 'Hello World';
?>
```

### 2. Variabel

Variabel adalah wadah untuk menyimpan suatu nilai. Pada PHP, syarat variabel diantaranya:

- variabel diawali oleh tanda `$` diikuti oleh nama variabelnya
- nama variabel hanya boleh diawali oleh huruf dan underscore
- nama variabel hanya boleh mengandung alfanumerik (A-z, 0-9, dan `_`)
- nama variabel bersifat case sensitive (huruf kapital dan non-kapital dianggap berbeda)

```
<?php
$nama = "Student";
echo "Halo $nama";
?>
```

- Coba ganti baris kedua dengan kode `$1nama = "Student";` dan lihat pesan error apa yang muncul. Mengapa muncul error?
- Coba ganti baris kedua dengan kode `$-nama = "Student";` dan lihat pesan error apa yang muncul. Mengapa muncul error?
- Coba ganti baris kedua dengan kode `$_nama = "Coders";` dan lihat output apa yang muncul. Apakah menampilkan pesan error juga? Mengapa demikian?

### 3. Echo

Echo adalah perintah untuk menampilkan output nilai ke layar. Kita dapat menampilkan output berupa nilai langsung ataupun nilai dari sebuah variabel. Perhatikan dan jalankan contoh kode di samping dan lihat masing-masing outputnya. Perhatikan juga keterangan komentar dari masing-masing barisnya.

```
<?php
// mencetak nilai langsung
echo "Ini string <br>";

// mencetak nilai variabel
$nama = "POLNEP";
$tahun = 2020;
echo "$nama berdiri pada tahun $tahun <br>";
?>
```

- Coba tambahkan kode `echo '$nama berdiri pada tahun $tahun <br>';` dan jalankan. Output apa yang muncul? Mengapa demikian?

#### 4. Tipe Data

Tipe data adalah jenis dari data yang digunakan pada sebuah variabel. PHP mendukung tipe data berikut:

- **String** adalah susunan dari huruf, angka dan karakter lainnya, seperti `"Halo Student!"`. String ditulis dengan diapit oleh tanda kutip (") atau tanda petik (').
- **Integer** adalah tipe data angka non-desimal, dengan rentang -2,147,483,648 hingga 2,147,483,647.
- **Float** adalah tipe data angka desimal. Tanda desimal yang digunakan adalah tanda titik.
- **Boolean** adalah tipe data yang menyatakan dua kondisi, `TRUE` dan `FALSE`.
- **Array** adalah tipe data yang digunakan untuk menyimpan lebih dari satu nilai data dalam satu variabel.
- **Object** adalah tipe data yang menyimpan nilai data beserta sejumlah operasi data. Objek akan dibahas detail di pembahasan OOP.
- **NULL** adalah tipe data dengan satu nilai saja, yakni `NULL`. Suatu variabel yang tidak berisi nilai dari tipe data lain, maka variabel tersebut bernilai `NULL`.
- Umumnya kita menampilkan nilai dari suatu variabel dengan menggunakan perintah `echo`. Akan tetapi `echo` tidak akan menampilkan nilai dengan tipe data kompleks (yang tidak memiliki nilai tunggal).
- Untuk tipe data kompleks, kita dapat menggunakan perintah `print_r()` atau `var_dump()`. Fungsi `var_dump()` akan menampilkan output nilai berikut tipe datanya.

```
<?php
$message = "Halo Student!";
$lightspeed = 299792458;
$pi = 3.14159265;
$jomblo = TRUE;
$goldarah = array('A', 'B', 'AB', 'O');
$kosong;

var_dump($message);
?>
```

#### 5. Konstanta

Konstanta adalah sebuah nama (seperti variabel) untuk menyimpan suatu nilai. Berbeda dengan variabel, nilai dari konstanta tidak dapat diubah setelah dia didefinisikan. Selain itu, konstanta tidak menggunakan karakter `$` di awalnya. Membuat sebuah konstanta tidak menggunakan operator assignment (`=`) seperti pada saat membuat variabel, melainkan menggunakan fungsi

```
define(name, value, case-insensitive)
```

`name` adalah nama konstanta. `value` adalah nilai dari konstanta. `case-insensitive` adalah penentuan apakah konstanta akan bersifat *case-insensitive* (tidak memperhitungkan huruf kapital dan non-kapital), yang defaultnya bernilai `false` (artinya secara default bersifat *case-sensitive*).

```
<?php

define('PI', 3.14159265);
echo PI;
?>
```

- Pada baris 4, ganti kode menjadi

```
echo pi;
```

Output apa yang muncul?

- pada baris 3, tambahkan parameter ketiga pada fungsi **define()** dengan nilai **true** sehingga kodenya menjadi

```
define('PI', 3.14159265, true);
```

Jalankan lagi program dan lihat output apa yang muncul?

- Pada baris 5 - 7, tambahkan kode berikut untuk mengeset ulang nilai konstanta:

```
echo '<br />';
define('PI', 3.14, true);
echo PI;
```

Apakah nilai konstanta berubah dari nilai sebelumnya?

6. Operator Aritmatika

Operator digunakan untuk melakukan operasi terhadap suatu nilai dan variabel. Operator Aritmatika digunakan untuk melakukan operasi aritmatik terhadap nilai numerik, seperti penjumlahan, pengurangan, perkalian, pembagian, dan sebagainya.

Op	Nama	Contoh	Keterangan
+	Penjumlahan	\$x + \$y	Penjumlahan nilai \$x dan \$y
-	Pengurangan	\$x - \$y	Pengurangan nilai \$x oleh \$y
*	Perkalian	\$x * \$y	Perkalian nilai \$x dan \$y
/	Pembagian	\$x / \$y	Pembagian nilai \$x oleh \$y
%	Modulus	\$x % \$y	Sisa dari pembagian \$x oleh \$y
**	Perpangkatan	\$x ** \$y	Hasil dari \$x pangkat \$y (mulai PHP 5.6)

```
<?php

$x = 20;
$y = 3;
$z = $x + $y;
echo $z;
?>
```

Gantilah operator yang digunakan pada kode baris ke-5 dengan operator lainnya, dan lihatlah hasilnya!

7. Operator Penugasan (Assignment)

Operator penugasan (assignment) digunakan untuk menuliskan nilai pada suatu variabel. Operator penugasan yang paling umum digunakan adalah operator 'sama dengan' =. Operator ini digunakan untuk mengisi variabel yang ada di sebelah kiri dengan nilai yang ada di sebelah kanan. Misalnya `$x = 2` berarti kita mengisi variabel `$x` dengan nilai 2. Atau `$x = $y` yang berarti kita mengisi variabel `$x` dengan nilai yang ada di dalam variabel `$y`.

Selain itu, kita dapat menyisipkan operator aritmatika sebelum tanda =, yang berarti mengisi variabel di sebelah kiri operator dengan hasil operasi dari variabel sebelah kiri sebelum diisi dengan nilai di sebelah kanan. Agar lebih paham, perhatikan tabel berikut:

Contoh	Serupa	Keterangan
<code>\$x += \$y</code>	<code>\$x = \$x + \$y</code>	<code>\$x</code> diisi dengan nilai dari <code>\$x</code> sebelumnya ditambah nilai <code>\$y</code>
<code>\$x -= \$y</code>	<code>\$x = \$x - \$y</code>	<code>\$x</code> diisi dengan nilai dari <code>\$x</code> sebelumnya dikurangi nilai <code>\$y</code>
<code>\$x *= \$y</code>	<code>\$x = \$x * \$y</code>	<code>\$x</code> diisi dengan nilai dari <code>\$x</code> sebelumnya dikali nilai <code>\$y</code>
<code>\$x /= \$y</code>	<code>\$x = \$x / \$y</code>	<code>\$x</code> diisi dengan nilai dari <code>\$x</code> sebelumnya dibagi nilai <code>\$y</code>
<code>\$x %= \$y</code>	<code>\$x = \$x % \$y</code>	<code>\$x</code> diisi dengan sisa nilai dari <code>\$x</code> sebelumnya dibagi nilai <code>\$y</code>
<code>\$x **= \$y</code>	<code>\$x = \$x ** \$y</code>	<code>\$x</code> diisi dengan nilai dari <code>\$x</code> sebelumnya dipangkatkan nilai <code>\$y</code>

```
<?php
    $x = 20;
    $y = 2;
    echo "nilai x sebelum assignment: $x <br>";
    $x += $y;
    echo "nilai x setelah assignment: $x <br>";
?>
```

Coba ganti operator penugasan pada kode baris ke-6 dengan operator lain dan lihat bagaimana outputnya!

8. Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan dua buah nilai numerik atau string. Output dari operasi ini adalah nilai **TRUE** atau **FALSE**.

Op	Nama	Contoh	Hasil
==	Sama dengan	\$x == \$y	TRUE bila nilai \$x sama dengan \$y
===	Identik	\$x === \$y	TRUE bila nilai \$x sama dengan \$y, DAN tipe data keduanya sama
!=	Tidak sama dengan	\$x != \$y	TRUE bila nilai \$x tidak sama dengan \$y
<>	Tidak sama dengan	\$x <> \$y	TRUE bila nilai \$x tidak sama dengan \$y
!==	Tidak identik	\$x !== \$y	TRUE bila nilai \$x tidak sama dengan \$y, ATAU tipe data keduanya berbeda
>	Lebih dari	\$x > \$y	TRUE bila nilai \$x lebih dari \$y
<	Kurang dari	\$x < \$y	TRUE bila nilai \$x kurang dari \$y
>=	Lebih dari atau sama dengan	\$x >= \$y	TRUE bila nilai \$x lebih dari atau sama dengan \$y
<=	Kurang dari atau sama dengan	\$x <= \$y	TRUE bila nilai \$x kurang dari atau sama dengan \$y

```
<?php
  
$x = 20;
$y = 20;
var_dump($x == $y);
?>
```

- Gantilah operator pada kode di baris 5 dengan **<**, **>**, **<=**, **>=** dan **!=** lalu amati output masing-masing operasinya.
- Ganti nilai **\$x** dengan string **'20'** dan ganti operator pada kode di baris 5 dengan **==** dan **===** lalu amati outputnya. Apakah outputnya berbeda? Mengapa berbeda?
- Ekplorasi lebih lanjut dengan mengganti nilai variabel **\$x** dan **\$y** dan juga operator yang digunakan dan bandingkan masing-masing outputnya.

9. Operator Increment-Decrement

Operator increment-decrement adalah operator yang digunakan untuk menaikkan atau menurunkan sebanyak satu dari nilai variabel yang dikenai operasi. Operator

increment adalah tanda ++ yang disimpan sebelum atau sesudah variabel, dan operator decrement adalah tanda -- yang disimpan sebelum atau sesudah operasi.  
Agar lebih jelas lihat tabel berikut:

Operator	Nama	Keterangan
++\$x	pre-increment	tambah nilai \$x sebanyak 1, kemudian kembalikan nilai \$x
\$x++	post-increment	kembalikan nilai \$x, lalu tambah nilai \$x sebanyak 1
--\$x	pre-decrement	kurangi nilai \$x sebanyak 1, kemudian kembalikan nilai \$x
\$x--	post-decrement	kembalikan nilai \$x, lalu kurangi nilai \$x sebanyak 1

<?php

```
$x = 10;
echo ++$x;
?>
```

- Ganti kode di baris 4 menjadi `echo $x++;` dan jalankan. Bagaimana hasil outputnya? Mengapa?
- Agar lebih jelas, coba tampilkan lagi nilai `$x` di baris selanjutnya dengan kode `echo $x;`, lalu jalankan. Dapatkah Kamu menyimpulkan kondisi tersebut?
- Eksplorasi dengan mengganti operator pada baris 4 dengan operator increment-decrement lainnya dan perhatikan outputnya.

10. Operator Logika

Operator logika digunakan untuk membandingkan dua pernyataan kondisi. Kondisi yang dimaksud adalah nilai variabel atau hasil perbandingan variabel dalam tipe data boolean (TRUE dan FALSE). Operator logika yang tersedia diantaranya `and`, `or`, `xor` dan `not`.

| Op  | Nama | Contoh  | Output  |
|---|------|---|---|
| <code>and</code><br><code>&amp;&amp;</code> | And  | <code>\$x and \$y</code><br><code>\$x &amp;&amp; \$y</code> | TRUE hanya bila \$x and \$y keduanya bernilai TRUE                              |
| <code>or</code><br><code>  </code>          | Or   | <code>\$x or \$y</code><br><code>\$x    \$y</code>          | TRUE bila salahsatu atau keduanya dari \$x atau \$y bernilai TRUE               |
| <code>xor</code>                            | Xor  | <code>\$x xor \$y</code>                                    | TRUE bila hanya salahsatu saja (tidak keduanya) dari \$x atau \$y bernilai TRUE |
| <code>!</code>                              | Not  | <code>!\$x</code>   | TRUE bila \$x bernilai FALSE  |

```

<?php
$x = true;
$y = true;
?>

<table class="table table-bordered table-striped">
  <tr><th>$x</th><th><?php var_dump($x); ?></th></tr>
  <tr><th>$y</th><th><?php var_dump($y); ?></th></tr>
  <tr><th>!$x</th><td><?php var_dump(!$x); ?></td></tr>
  <tr><th>!$y</th><td><?php var_dump(!$y); ?></td></tr>
  <tr><th>$x and $y</th><td><?php var_dump($x and $y); ?></td></tr>
  <tr><th>$x or $y</th><td><?php var_dump($x or $y); ?></td></tr>
  <tr><th>$x xor $y</th><td><?php var_dump($x xor $y); ?></td></tr>
</table>

```

Ganti nilai variabel `$x` dan `$y` pada editor menjadi `true` atau `false`, lalu jalankan program. Amati dan pelajari tiap-tiap outputnya.

## 11. Operator String

Operator string sangat membantu dan akan sering digunakan pada operasi yang berkaitan dengan data string. Operasi yang dimaksud adalah konkatenansi atau penggabungan dua buah string, dan operator yang digunakan adalah tanda titik `..`.

Selain itu, operator konkatenansi ini juga dapat digunakan pada operasi assignment, seperti halnya operator aritmatika.

```
$x .= $y;
```

Contoh kode di atas artinya kita mengisi variabel `$x` dengan nilai string hasil penggabungan antara nilai variabel `$x` sebelumnya dengan nilai variabel `$y`.

```

<?php

$x = "Halo ";
$y = "Coders!";
echo $x . 'Indonesia! <br>';
echo $x .= $y;
?>

```

coba mengeksplorasi dengan string lainnya dan lihat bagaimana hasilnya.

## 12. Statemen IF

Seringkali kita harus melakukan suatu operasi hanya pada kondisi tertentu saja. Artinya kita mengecek terlebih dahulu sebuah kondisi, yang apabila kondisi tersebut terpenuhi (atau bernilai true), maka kita lakukan serangkaian operasi.

Statemen if digunakan untuk mengecek kondisi dan bila kondisi bernilai true, maka program tertentu akan dijalankan. Penulisan sintaksnya adalah:

```
if(kondisi) {  
    // kode untuk dieksekusi bila kondisi bernilai true  
}
```

Pada bagian kondisi, kita isi dengan suatu variabel yang nantinya akan dicek nilainya. Variabel bernilai false, null, string kosong atau 0 akan menghasilkan output false, dan selain nilai tersebut akan memberikan output true. Selain itu kita juga dapat membandingkan variabel menggunakan operator perbandingan dan operator logika, atau hasil keluaran dari sebuah fungsi.

```
<?php
```

```
$nama = "Toni";
```

```
$message = "Selamat datang $nama!";
```

```
if($nama == "Jobs"){
```

```
    $message = "Hi, it's an honor to greet you, $nama!";
```

```
}
```

```
echo $message;
```

```
?>
```

- Apa output yang ditampilkan bila Kamu mengisi variabel **\$nama** dengan namamu? Apa pula output yang ditampilkan bila variabel tersebut diisi dengan string "Jobs"?

## 13. Statemen if..else

Adakalanya kita harus mengeksekusi satu blok kode program saat suatu kondisi terpenuhi, dan mengeksekusi blok kode program lain yang berbeda bila kondisi tidak terpenuhi. Untuk itu kita gunakan statemen if..else.

```
if (kondisi) {  
    // kode untuk dieksekusi bila kondisi bernilai TRUE  
} else {  
    // kode untuk dieksekusi bila kondisi bernilai FALSE  
}
```

```
<?php
```

```
$panjang = 14; // atau diameter lingkaran
```

```
$lebar = 5;
```

```
$tinggi = 4; // tinggi bangun ruang
```

```
if(isset($tinggi) && $tinggi > 0) {
```

```
    echo "Luas selimut balok: ". $panjang * $lebar * $tinggi . "<br>";
```

```
    echo "Luas selimut tabung: " . 2 * 22/7 * $panjang/2 * ( $panjang/2 + $tinggi );
```

```
} else {
```



```
echo "Luas persegi: ". $panjang * $lebar . "<br>";  
echo "Luas lingkaran: " . 1/4 * 22/7 * $panjang**2;  
}  
?>
```

Coba hapus variabel `$tinggi` atau isi variabel dengan nilai 0, dan lihat output apa yang muncul.