

# **MODUL 5 - PHP**

## **Web Programming**



**Disusun oleh :**

**Willy Rafael F. Silalahi\_23083000168\_4G**

**JURUSAN SISTEM INFORMASI  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS MERDEKA  
MALANG**

## MODUL 5

### PHP (Hypertext Preprocessor)

---

#### TUJUAN

1. Mahasiswa Memahami penggunaan PHP
2. Mahasiswa Mampu mengaplikasikan dan membuat PHP
3. Mahasiswa mengetahui dan mampu menjelaskan cara kerja PHP.

#### CARA Pengerjaan

1. Setiap code diisikan dengan Screenshoot
2. Output diisikan dengan Screenshoot
3. Penjelasan dituliskan dalam text

#### OVERVIEW

PHP (PHP : Hypertext Preprocessor) adalah sebuah bahasa scripting yang paling populer saat ini. Berfungsi untuk membuat website dengan konten dinamis maupun aplikasi web lainnya. Berbeda dengan HTML yang hanya menampilkan konten statis, PHP dapat berinteraksi dengan database, file, folder bahkan dengan lingkungan sistem operasi. PHP termasuk kedalam golongan perangkat lunak interpreter. Yakni menerjemahkan baris demi baris program (tanpa mengcompilanya) kemudian mengeksekusinya. PHP awal mula dikembangkan oleh **Rasmus Lerdorf** sebagai pengembangan dari **Perl** untuk menangani **pengolahan form** pada website pribadinya. Oleh karenanya, saat itu PHP disebut juga sebagai akronim dari Personal Home Page. Saat ini lebih dari 20 juta website menggunakan PHP backend aplikasi web.

## MODUL 5

### PHP (Hypertext Preprocessor)

---

#### SINTAKS DASAR

Setiap bahasa pemrograman memiliki aturan coding sendiri. Begitu pula dengan PHP. Sintaks dasarnya dibuka dengan **<?php** dan ditutup dengan **?>** sebagai terlihat di contoh berikut:



Gambar 5. 1. Sintaks dasar PHP

Kotak warna merah pada Gambar 5.1 merupakan kode PHP pertama kali yang ada di halaman, kode tersebut berfungsi untuk mengeluarkan kata “belajar PHP pertamakali”. Mengapa bahasa pemrograman ini dibilang sebagai bahasa pemrograman backend? Hal ini dikarenakan setiap code yang anda tuliskan pada lembar kerja tidak akan ditampilkan semua.

## MODUL 5

### PHP (Hypertext Preprocessor)

---

Coba kode berikut :



Gambar 5. 2. Tag html didalam PHP

Ketikan kode diatas dan apabila terjadi eror kenapa bisa terjadi eror :

Code :

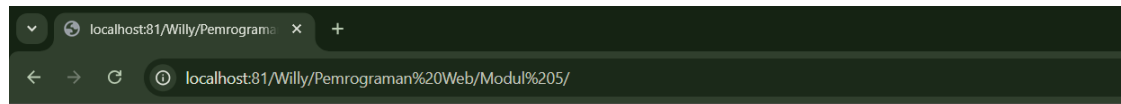


## MODUL 5

### PHP (Hypertext Preprocessor)

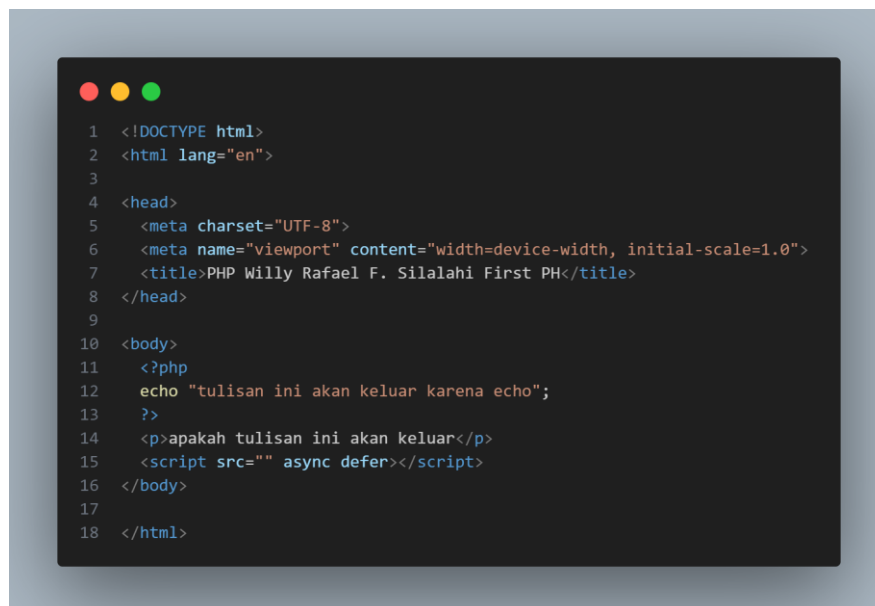
---

Output :



Kesalahan dalam kode PHP tersebut terjadi karena adanya tag HTML `<p>` yang dimasukkan di dalam blok PHP tanpa penanganan yang benar. Dalam PHP, semua teks atau elemen HTML yang ingin ditampilkan harus menggunakan perintah `echo` atau `print`. Jika sebuah tag HTML ditempatkan dalam blok PHP tanpa dibungkus dalam pernyataan `echo`, maka akan terjadi error karena PHP tidak mengenali tag tersebut sebagai bagian dari output yang perlu diproses.

Code yang benar :



Untuk memperbaiki masalah ini, bisa memastikan bahwa tag `<p>` ditulis dengan benar menggunakan `echo`, seperti berikut: `echo "<p>apakah tulisan ini akan keluar</p>";`. Alternatif lainnya adalah memisahkan PHP dan HTML dengan cara menutup blok PHP sebelum menulis tag HTML, misalnya: `<?php echo "tulisan ini akan keluar karena echo"; ?> <p>apakah tulisan ini akan keluar</p>`. Dengan cara ini, PHP

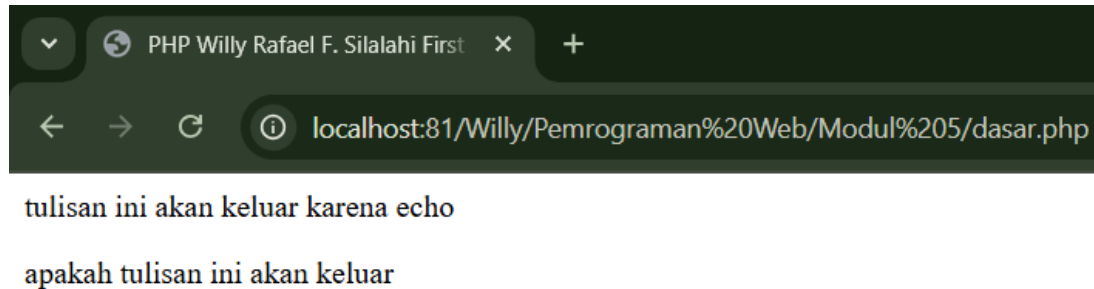
## MODUL 5

### PHP (Hypertext Preprocessor)

---

hanya akan menangani bagian yang perlu diproses, sementara HTML tetap dalam format yang benar sehingga tidak menyebabkan error.

Output :



### KEUNTUNGAN

Keuntungan dari bahasa pemrograman javascript adalah :

1. **Cenderung mudah dipelajari** — dibanding beberapa bahasa pemrograman populer lain, PHP lebih mudah dipelajari.
2. **Materi belajar yang melimpah** — umur PHP yang “cukup tua” menyebabkan banyak sekali dokumentasi, panduan, dan komunitas aktif bertebaran di jagat maya. Jadi, tak perlu takut jika Anda mengalami kesulitan.
3. **PHP bersifat open-source** — siapapun bisa menggunakan PHP tanpa mengeluarkan biaya sepeserpun.
4. **Banyaknya pilihan database** — PHP bisa digunakan di hampir semua jenis database. Mulai dari MySQL, hingga non-relational database seperti Redis.
5. **Kompatibilitas yang baik dengan HTML** — script PHP tidak mengganggu HTML sama sekali. Justru mereka berdua saling melengkapi.
6. **Fleksibilitas tinggi** — PHP bisa dikombinasikan dengan banyak sekali bahasa pemrograman lain. Sehingga bisa Anda gunakan sesuai kebutuhan.

## MODUL 5

### PHP (Hypertext Preprocessor)

---

7. **Multi-platform** — PHP bisa Anda gunakan di macam-macam operating system. Mulai dari Windows, Linux, hingga MacOS.
8. **Selalu diperbarui** — sejak pertama kali muncul tahun 1995, sekarang PHP sudah berada pada versi 8.0.
9. **Mendukung layanan cloud** — siapa sangka, walaupun umur PHP hampir dua dekade, tapi ia bisa mendukung layanan cloud dengan skalabilitas yang baik.

## MODUL 5

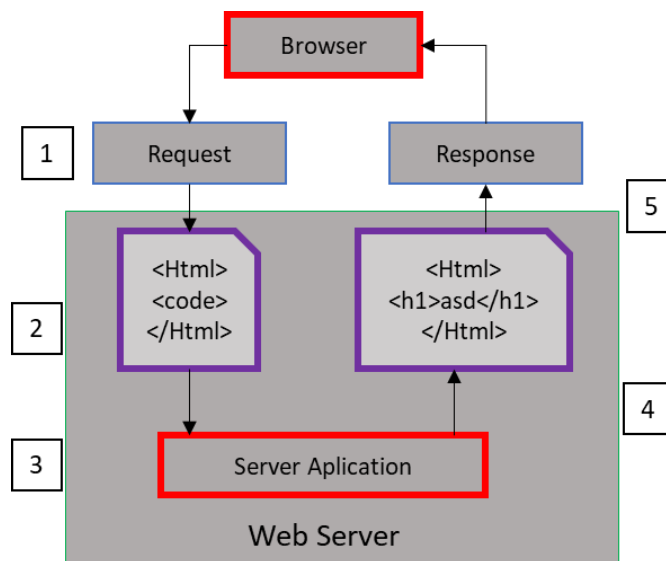
### PHP (Hypertext Preprocessor)

---

#### CODE PADA PHP

**JavaScript** termasuk jenis bahasa script, yang digunakan di dalam file HTML. Untuk menginput, atau memasukkan kode JavaScript ke dalam HTML, JavaScript menyediakan 4 alternatif, yaitu :

#### 5.1 Sekema App berbasis WEB



Gambar 5. 3. Skema Web

Skema aplikasi web pada gambar diatas dapat dijelaskan sebagai berikut:

1. Client (web browser) meminta sebuah halaman pada sebuah website.
2. Web server mencari halaman yang diminta, dan setelah ditemukan, memberikan halaman tersebut ke application server (PHP Interpreter).
3. Application server mem-parse (membaca) dan memproses halaman tersebut.
4. Application server mengirimkan halaman yang telah diproses kembali ke web server.
5. Web server mengirimkan kembali halaman yang telah diproses kepada web browser dalam bentuk HTML.



## MODUL 5

### PHP (Hypertext Preprocessor)

---

#### 5.2 DasarPHP

##### 5.2.1 Penulisan

Berbeda dengan bahasa banyak bahasa pemrograman web pada umumnya, skrip program PHP dapat langsung ditanamkan / disisipkan didalam HTML. Jika skrip PHP ditemukan dalam sebuah dokumen web, PHP Interpreter akan memproses skrip PHP dan hanya akan mengirimkan hasil berupa HTML ke browser client.

Dalam sebuah program, terdapat dua hal penting: command (perintah) dan values (nilai-nilai). Command memiliki tugas spesifik yang telah ditentukan. Sedangkan values adalah informasi yang berkaitan dengan tugas yang dilakukan oleh command. Dari contoh-contoh sebelumnya, kita telah banyak menggunakan literal. Yaitu value yang telah diterapkan secara eksplisit didalam program pada sebuah command. Sebagai contoh baris program berikut:

**echo "Hello World!";**

Pernyataan diatas melakukan command echo dan menggunakan value literal yang diberikan setelahnya, yaitu "Hello World!".

Cobalah menampilkan Hello World pertama anda menggunakan PHP:

Code :

A code editor window with a dark background and light blue text. It shows two lines of PHP code: line 1 is `<?php` and line 2 is `echo "Hello World!";`. The code is displayed on a light blue background with a dark blue border.

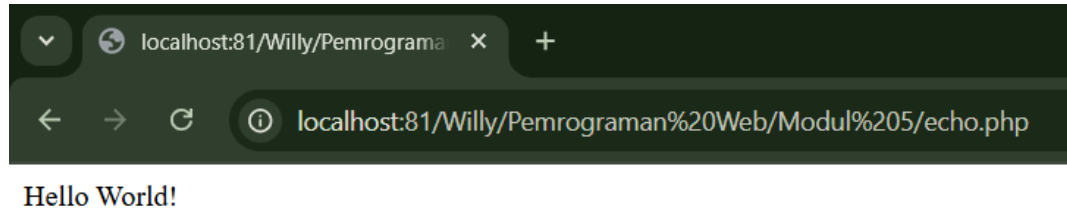
```
1 <?php
2 echo "Hello World!";
```

## MODUL 5

### PHP (Hypertext Preprocessor)

---

Output :



#### 5.2.2 Cetak String

Dalam PHP, string adalah rangkaian karakter, seperti “Saya telah belajar PHP” atau “Mydreamisprogrammer” atau “123123123123123”. Ketika PHP membaca dan memproses sebuah kode program dan mengirimkan output ke web browser, setiap data yang ditampilkan akan dianggap sebagai sebuah string. Terdapat dua fungsi yang umum digunakan untuk mencetak sebuah string. Yaitu print dan echo. Untuk mencetak sebuah string digunakan sepasang karakter single quoted ( ' ) atau double quoted ( " ) yang mengapit string tersebut.

Buatlah biodata anda dengan PHP (bisa menggunakan echo atau print ) :

Code :

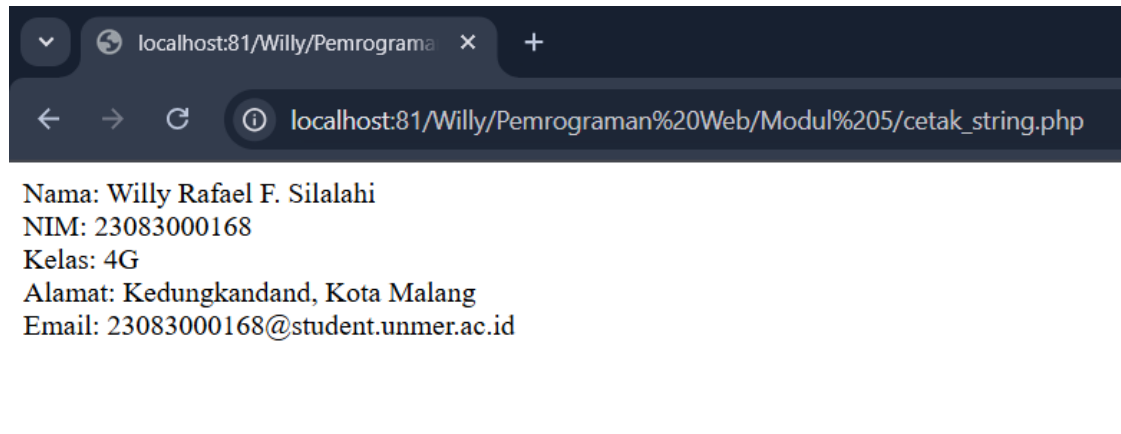
```
1 <?php
2 echo "Nama: Willy Rafael F. Silalahi<br>";
3 echo "NIM: 23083000168<br>";
4 echo "Kelas: 4G<br>";
5 echo "Alamat: Kedungkandand, Kota Malang<br>";
6 echo "Email: 23083000168@student.unmer.ac.id<br>";
7 >>
```

## MODUL 5

### PHP (Hypertext Preprocessor)

---

Output :



### 5.3 Variabel

Variable adalah sebuah pengenal yang digunakan untuk menyimpan suatu nilai yang nilai tersebut dapat berubah-ubah selama dijelankannya program.

#### 5.3.1 Deklarasi Variabel

Untuk mendefinisikan sebuah variabel, diawali dengan karakter dolar (\$) dan diikuti dengan nama pengenal. Adapun aturan-aturan yang berkaitan dengan variabel adalah sebagai berikut:

1. Terdiri dari huruf, angka dan karakter garis bawah ( \_ ) atau kombinasi antara ketiganya.
2. Tidak boleh mengandung spasi dan terdiri dari minimal satu karakter.
3. Dimulai dengan huruf atau garis bawah, tidak boleh diawali dengan angka.
4. Bersifat case sensitive, yaitu huruf besar dan kecil dibedakan.

Seperti yang telah kita ketahui, sebuah variabel akan bernilai valid jika telah memenuhi persyaratan yang telah ditentukan diatas. Namun, untuk memudahkan memahami alur program kita, cobalah untuk mengikuti cara penulisan variabel seperti di bawah ini:

1. Nama variabel mencerminkan isinya. Sebagai contoh, jangan gunakan

## MODUL 5

### PHP (Hypertext Preprocessor)


---

variable \$h untuk menyimpan total harga, melainkan gunakan

2. \$total\_harga. Jika suatu saat kita ingin memperbaiki kode program beberapa bulan setelah kita pertama kali menulis program itu, kita bingung dengan variabel bernama \$h karena ia tidak mencerminkan apapun dari isinya. Berbeda dengan nama variabel \$total\_harga yang dengan jelas mencerminkan isinya.

Sebisa mungkin variabel harus singkat. Mendefinisikan variabel terlalu panjang kadang-kadang terlalu merepotkan. Cobalah untuk mempersingkat, namun tetap masuk akal.

Untuk lebih jelasnya lihatlah code berikut :



```
<html>
  <head>
    <?php
      $tinggi = 5;
      $panjang = 10;
      $lebar = 5;
      $volume = $tinggi * $panjang * $lebar;
      echo $volume;
      echo ('<br>');
      print $volume;
    ?>
  </head>
</html>
```

Gambar 5. 4. Contoh Variabel

Buatlah contoh Variabel degan menghitung volume tabung menggunakan PHP (r dan t terserah anda) :

## MODUL 5

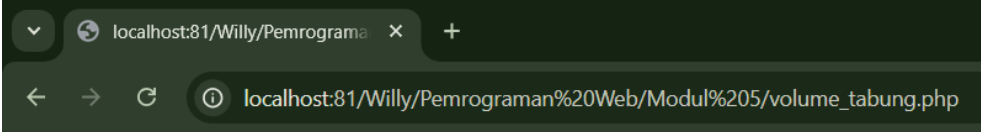
### PHP (Hypertext Preprocessor)

---

Code :

```
1 <?php
2 $pi = 3.14;
3 $r = 5;
4 $h = 10;
5
6 $volume = $pi * $r * $h;
7
8 echo "Volume tabung dengan jari-jari $r dan tinggi $h adalah $volume";
9
```

Output :



Volume tabung dengan jari-jari 5 dan tinggi 10 = 157

#### 5.3.2 Penugasan Variabel

Terdapat dua cara dalam penugasan variabel: dari skrip itu sendiri atau ditentukan oleh PHP Interpreter. Saat ini kita hanya fokus ada penugasan yang dilakukan dari skrip. Seperti yang telah kita pelajari, penugasan terjadi ketika sebuah nilai diberikan pada sebuah variabel. Perhatikan dua pernyataan sebagai berikut:

```
$int_tujuh = 7;
```

```
7 = $int_tujuh;
```

Aturan penugasan variabel dapat dimulai dari kanan ke kiri, yang dapat diterjemahkan sebagai berikut :

1. Untuk pernyataan pertama, dapat dibaca “bilangan 7 ditugaskan ke variabel \$int\_tujuh”.

## MODUL 5

### PHP (Hypertext Preprocessor)

---

2. Kemudian kita bandingkan dengan pernyataan ke dua: “variabel \$int\_tujuh ditugaskan ke bilangan 7”.
3. Pernyataan tidak masuk akal dan tidak akan bekerja, dan jika menerapkan pernyataan tersebut pada program, PHP akan berhenti dengan pesan error.

Tidak seperti bahasa pemrograman lainnya, tidak diperlukan deklarasi tipe data untuk sebuah variabel. Setiap variabel terbentuk dalam tipe data variant, yang artinya ia **dapat menampung jenis data apapun**. Meskipun variabel pada PHP bersifat variant, kita dapat mengetahui tipe data yang dikandung pada sebuah variabel menggunakan fungsi `gettype()`

Code :

```
1 <?php
2 // Penugasan variabel
3 $int_tujuh = 7; // Bilangan 7 ditugaskan ke variabel $int_tujuh
4
5 // Menampilkan nilai variabel
6 echo "Nilai dari \$int_tujuh: " . $int_tujuh . "<br>";
7
8 // Menampilkan tipe data variabel menggunakan gettype()
9 echo "Tipe data dari \$int_tujuh: " . gettype($int_tujuh) . "<br>";
10
11 // Mengubah nilai dan tipe data variabel
12 $int_tujuh = "Tujuh"; // Variabel sekarang berisi string
13
14 // Menampilkan nilai dan tipe data setelah perubahan
15 echo "Nilai baru dari \$int_tujuh: " . $int_tujuh . "<br>";
16 echo "Tipe data baru dari \$int_tujuh: " . gettype($int_tujuh) . "<br>";
17 ?>
```

Penjelasan kode:

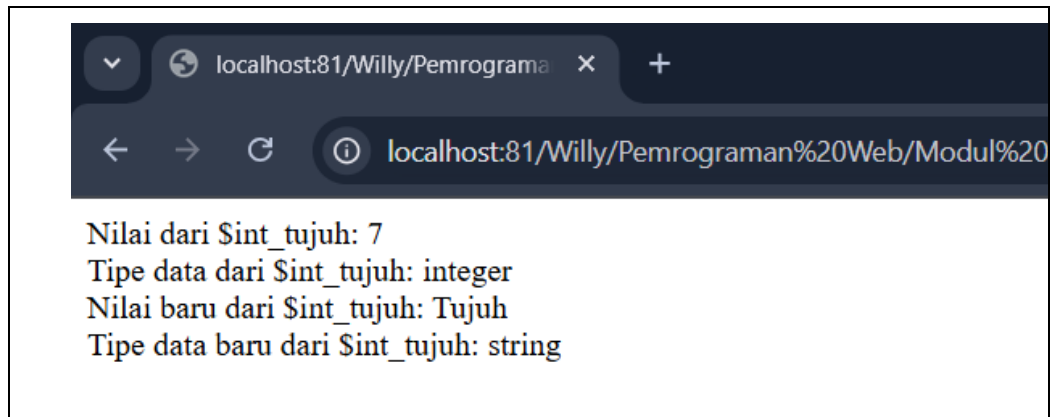
- Variabel \$int\_tujuh pertama kali diberi nilai 7, sehingga bertipe integer.
- Kemudian, nilai dari \$int\_tujuh diubah menjadi "Tujuh", yang merupakan string.
- Fungsi `gettype()` digunakan untuk mengetahui tipe data variabel sebelum dan sesudah perubahan.

Output:

## MODUL 5

### PHP (Hypertext Preprocessor)

---



#### 5.3.3 Type Casting

Type casting adalah mekanisme pengubahan / konversi sebuah variabel kepada tipe data tertentu. Sebagai contoh, kadangkala kita hanya ingin menggunakan bagian integer dari sebuah bilangan pecahan. Untuk melakukan konversi, gunakan format cara berikut:

(tipe) \$variabel

Type casting dapat diterapkan pada sebuah variabel dengan tipe data sebagai berikut:

- a. (int) atau (integer)
- b. (real), (double) atau (float)
- c. (string)
- d. (array)
- e. (object)

## MODUL 5

### PHP (Hypertext Preprocessor)

---

Buatlah contoh code type casting pada PHP (Pilih salah satu dari 5) :

Code :

```
1 <?php
2 // Type casting dari float ke integer
3 $float_var = 12.8;
4 $int_var = (int) $float_var;
5
6 echo "Nilai float: $float_var<br>";
7 echo "Nilai setelah di-cast ke integer: $int_var<br><br>";
8
9 // Type casting dari integer ke string
10 $int_var2 = 100;
11 $string_var = (string) $int_var2;
12
13 echo "Nilai integer: $int_var2<br>";
14 echo "Nilai setelah di-cast ke string: $string_var<br><br>";
15
16 // Type casting dari string ke array
17 $string_var2 = "Hello, world!";
18 $array_var = (array) $string_var2;
19
20 echo "Nilai string: $string_var2<br>";
21 echo "Nilai setelah di-cast ke array:<br>";
22 print_r($array_var);
23 echo "<br><br>";
24
25 // Type casting dari array ke object
26 $array_var2 = ["nama" => "Willy", "usia" => 21];
27 $object_var = (object) $array_var2;
28
29 echo "Nilai array:<br>";
30 print_r($array_var2);
31 echo "<br>Nilai setelah di-cast ke object:<br>";
32 print_r($object_var);
33 echo "<br>";
34 ?>
```

Penjelasan:

- Float ke Integer: Hanya mengambil bagian bilangan bulat, tanpa pembulatan.
- Integer ke String: Mengubah angka menjadi tipe teks.
- String ke Array: Mengubah teks menjadi array, di mana teks akan berada dalam indeks pertama.
- Array ke Object: Konversi array asosiatif menjadi object.

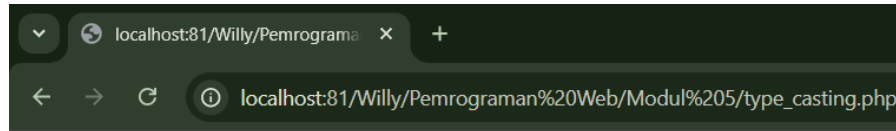


## MODUL 5

### PHP (Hypertext Preprocessor)

---

Output :



```
Nilai float: 12.8
Nilai setelah di-cast ke integer: 12

Nilai integer: 100
Nilai setelah di-cast ke string: 100

Nilai string: Hello, world!
Nilai setelah di-cast ke array:
Array ( [0] => Hello, world! )

Nilai array:
Array ( [nama] => Willy [usia] => 21 )
Nilai setelah di-cast ke object:
stdClass Object ( [nama] => Willy [usia] => 21 )
```

#### 5.4 Konstanta

Konstanta adalah sebuah tetapan nilai dalam sebuah program. Nilai konstanta tidak dapat diubah selama program dijalankan dan jika dilakukan hanya akan menyebabkan error. Deklarasi konstanta dapat dilakukan dengan cara sebagai berikut:

```
define("CONTOH_KONSTANTA", 5);
```

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

Sebagaimana variabel, konstanta juga yang memiliki aturan yang harus diikuti. Aturan berikut memastikan kita untuk mendeklarasikan konstanta dengan benar:

1. Tidak boleh didahului oleh karakter dolar.
2. Nama konstanta harus dimulai dengan huruf atau garis bawah ( \_ ).
3. Konstanta tidak dapat dideklarasikan ulang dan nilainya tetap.

Sebagaimana variabel, dibutuhkan aturan-aturan khusus untuk memudahkan memahami alur program kita, cobalah untuk mengikuti cara penulisan konstanta

## MODUL 5

### PHP (Hypertext Preprocessor)

---

seperti dibawah ini:

1. Untuk kenyamanan, sebisa mungkin gunakan nama konstanta yang singkat. Karena pada umumnya konstanta digunakan untuk pilihan pengaturan.
2. Selalu gunakan huruf besar. Ini membantu kita membedakan konstanta dengan pernyataan lain.
3. Pisahkan setiap suku kata dalam konstanta dengan garis bawah ( \_ ). Garis bawah diperlukan untuk membantu kita membedakan satu konstanta dengan yang lainnya.

Buatlah contoh code Konstanta pada PHP :

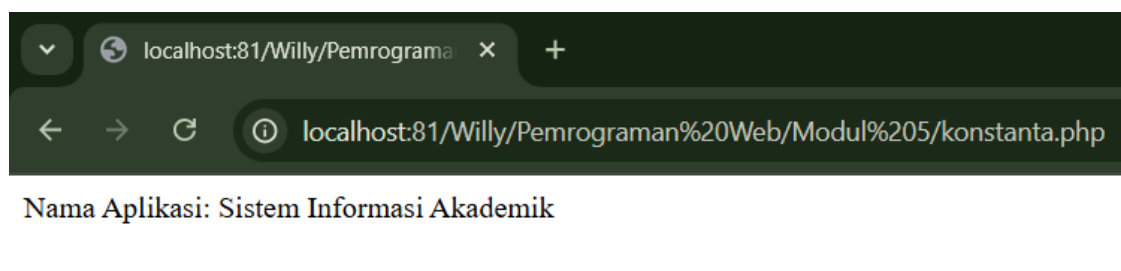
Code :

```
1 <?php
2 define("NAMA_APLIKASI", "Sistem Informasi Akademik");
3 echo "Nama Aplikasi: " . NAMA_APLIKASI . "<br>";
4 ?>
```

Penjelasan:

1. **Menggunakan** define(): Konstanta seperti NAMA\_APLIKASI, dideklarasikan menggunakan define(), lalu ditampilkan di layar.
2. **Konstanta Tidak Bisa Diubah:** Setelah dideklarasikan, mencoba mengubahnya akan menyebabkan error.

Output :



## MODUL 5

### PHP (Hypertext Preprocessor)

---

#### 5.5 Operator

Operator digunakan untuk memanipulasi nilai sebuah variabel. Sedangkan nilai yang dimodifikasi oleh operator disebut operand. Sedangkan kombinasi dari keduanya disebut ekspresi. Dalam pemrograman operator dapat dikelompokkan menjadi:

1. Operator Aritmatika
2. Operator Relasional
3. Operator Logika
4. Lain-lain

##### 5.5.1 Operator Aritmatika

Operator aritmatika digunakan untuk melakukan perhitungan matematika.

Operator	Nama Operasi	Contoh
+ ( Unary )	Positif	+5
- ( Unary )	Negatif	-5
+	Penjumlahan	5 + 5
-	Pengurangan	5 - 5
*	Perkalian	5 * 5
/	Pembagian	5 / 5
%	Hasil Bagi	5 % 5

##### 5.5.2 Operator Relasional

Merupakan operator yang digunakan untuk membandingkan nilai dari dua operand. Hasil perbandingannya dinyatakan dalam nilai boolean. TRUE untuk pernyataan bernilai benar, dan FALSE untuk operator bernilai salah.

## MODUL 5

### PHP (Hypertext Preprocessor)

---

Operator	Keterangan
==	Memeriksa apakah operand kiri bernilai sama dengan operand kanan
>	Memeriksa apakah operand kiri lebih besar dari operand kanan
<	Memeriksa apakah operand kiri lebih kecil dari operand kanan
>=	Memeriksa apakah operand kiri lebih besar atau sama dengan operand kiri
<=	Memeriksa apakah operand kiri lebih kecil atau sama dengan operand kiri
!=	Memeriksa apakah operand kiri tidak bernilai sama dengan operand kanan

#### 5.5.3 Operator Logika

Operator logika digunakan untuk membandingkan dua atau lebih nilai variabel yang bertipe boolean. Hasil yang didapat dari penggunaan operator logika adalah boolean.

Terdapat beberapa jenis operator logika, diantaranya sebagai berikut:

1. Logika AND ( Disjungsi )
2. Logika OR ( Konjungsi )
3. Logika XOR
4. Logika NOT ( Negasi )

Buatlah contoh code logika pada PHP Pilih 2 dari 4 operator logika diatas :

Code :

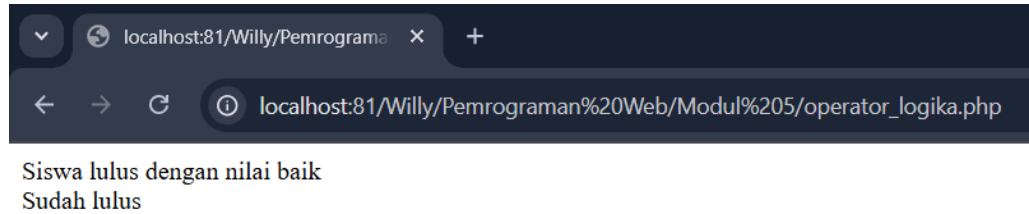
```
1 <?php
2 $nilai = 85;
3 $lulus = true;
4
5 // Operator Logika AND (&&)
6 echo ($nilai > 80 && $lulus) ? "Siswa lulus dengan nilai baik" : "Belum memenuhi syarat";
7 echo "<br>";
8 // Operator Logika NOT (!)
9 echo !$lulus ? "Belum lulus" : "Sudah lulus";
10 >>
```

## MODUL 5

### PHP (Hypertext Preprocessor)

---

Output :



#### 5.6 Operator Percabangan

Pada umumnya, operator percabangan dilakukan dengan struktur if-else, seperti pada contoh berikut:

```
<html>
  <head>
    <?php
      $user = "asd";
      if($user == "asd"){
        echo "Selamat datang Firnanda.";
      }
      else {
        echo "Username salah.";
      }
    ?>
  </head>
</html>
```

Tuliskan code diatas dengan nama anda dan screenshoot outputnya :

## MODUL 5

### PHP (Hypertext Preprocessor)

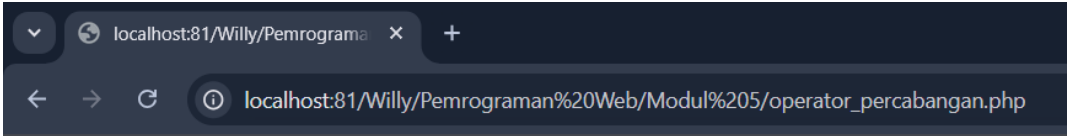
---

Code :



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <?php
5     $user = "Willy";
6
7     if ($user == "Willy") {
8       echo "Selamat datang Willy.";
9     } else {
10      echo "Username salah.";
11    }
12  ?>
13 </head>
14 </html>
```

Output :



localhost:81/Willy/Pemrograma x +

localhost:81/Willy/Pemrograman%20Web/Modul%205/operator\_percabangan.php

Selamat datang Willy.

Hal yang sama dapat dilakukan dengan cara:

**echo (\$user == "asd" ? "Selamat datang Firnanda." : "Username salah.");**

Kondisi

Pernyataan 1

Pernyataan 2

Struktur diatas dapat dijelaskan sebagai berikut:

1. Pertama kali, PHP akan memeriksa apakah kondisi yang menjadi prasyarat bernilai benar. Yaitu apakah variabel \$user bernilai "asd".
2. Jika benar, pernyataan 1 yang dipilih, dan pernyataan 2 akan dipilih jika kondisi prasyarat bernilai salah.
3. Kemudian program akan menampilkan (echo) pernyataan yang telah dipilih tadi.

## MODUL 5

### PHP (Hypertext Preprocessor)

---

Tuliskan code daftar nilai dan screenshot outputnya menggunakan PHP :

**Soal Percabangan :**

Jika nilai  $> 80$  = A

Jika  $70 < \text{nilai} < 79$  dan mendapat nilai = B

Jika  $60 < \text{nilai} < 69$  dan mendapat nilai = C

Jika  $50 < \text{nilai} < 59$  dan mendapat nilai = D

Jika nilai  $< 49$  dan mendapat nilai = Tidak Lulus

**Catatan** : User menginputkan nama dan nilai yang mana akan terklasifikasi secara otomatis ppada hasil.

Code :

```
1  <!DOCTYPE html>
2  <html lang="id">
3  <head>
4      <meta charset="UTF-8">
5      <title>Form Nilai Mahasiswa</title>
6  </head>
7  <body>
8      <h2>Input Nilai Mahasiswa</h2>
9      <form method="post">
10         <label>Nama:</label><br>
11         <input type="text" name="nama" required><br><br>
12
13         <label>Nilai:</label><br>
14         <input type="number" name="nilai" required><br><br>
15
16         <input type="submit" value="Kirim">
17     </form>
18
19     <?php
20     if ($_SERVER["REQUEST_METHOD"] == "POST") {
21         $nama = $_POST["nama"];
22         $nilai = $_POST["nilai"];
23
24         echo "<hr>";
25         echo "<h3>Hasil Klasifikasi:</h3>";
26         echo "Nama: $nama<br>";
27         echo "Nilai: $nilai<br>";
28         echo "Keterangan: ";
29
30         if ($nilai > 80) {
31             echo "A";
32         } elseif ($nilai >= 70 && $nilai <= 79) {
33             echo "B";
34         } elseif ($nilai >= 60 && $nilai <= 69) {
35             echo "C";
36         } elseif ($nilai >= 50 && $nilai <= 59) {
37             echo "D";
38         } else {
39             echo "Tidak Lulus";
40         }
41     }
42     ?>
43 </body>
44 </html>
```

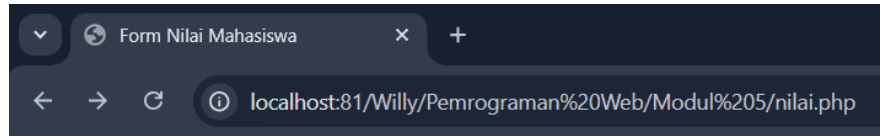


## MODUL 5

### PHP (Hypertext Preprocessor)

---

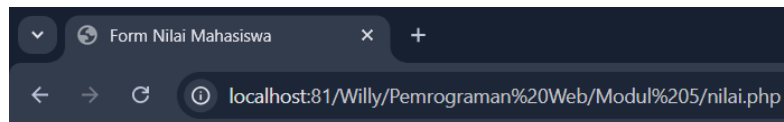
Output :



#### Input Nilai Mahasiswa

Nama:

Nilai:



#### Input Nilai Mahasiswa

Nama:

Nilai:

#### Hasil Klasifikasi:

Nama: Willy

Nilai: 95

Keterangan: A