

# MODUL 1

## PHP

### Tugas Pendahuluan

1. php adalah server side programming, apa yang dimaksud dengan server side programming?
2. Sebutkan beberapa contoh bahasa server side programming dan client side programming!.
3. Komponen apa sajakah yang diperlukan untuk menjalankan php ? sebutkan dan jelaskan !
4. jelaskan apa yang dimaksud dengan variable, tipe data, dan fungsi dalam PHP !
5. Jelaskan apa yang anda ketahui tentang struktu control dan perulangan!
6. Buatlah contoh kasus penggunaan struktur control dan perulangan dalam kehidupan sehari-hari.
7. Buatlah sebuah contoh studi kasus yang menggunakan gabungan struktur control dan perulangan!.

---

### 1. PHP Introduction

PHP banyak digunakan oleh programmer berlatar belakang C/C++ karena kemiripan syntaxnya. Open source, karenanya gratis dan bebas. Database pasangannya biasanya MySQL, dijalankan bersama webserver Apache di atas operating system Linux. Semuanya gratis dan bebas.

#### Yang perlu diketahui?

Kemampuan yang harus dimiliki sebelum belajar PHP:

HTML

CSS

JavaScript

#### Apa itu PHP?

PHP adalah akronim untuk "PHP Hypertext Preprocessor"

PHP adalah bahasa scripting open source yang digunakan secara luas

PHP scripts dijalankan pada server

PHP gratis untuk didownload dan digunakan

### **Apa itu PHP File?**

PHP files dapat berupa teks, HTML, CSS, JavaScript, PHP code

PHP code dijalankan pada server, dan hasilnya berupa return ke browser sebagai plain HTML

PHP files memiliki ekstensi ".php"

### **Apa yang dapat dilakukan PHP?**

PHP dapat menghasilkan konten halaman dinamis

PHP dapat membuat, membuka, menulis, menghapus dan menutup file di server

PHP dapat mengumpulkan data formulir

PHP dapat mengirim dan menerima cookie

PHP dapat menambah, menghapus, mengubah data dalam database

PHP dapat digunakan untuk mengontrol akses pengguna

PHP dapat mengenkripsi data

PHP dapat menghasilkan output berupa images, file PDF, dan bahkan Flash film

### **Mengapa PHP?**

PHP berjalan pada berbagai platform (Windows, Linux, Unix, Mac OS X, dll.)

PHP kompatibel dengan hampir semua server yang digunakan sekarang (Apache, IIS, dll.)

PHP mendukung berbagai database

PHP gratis. Download dari sumber PHP resmi: [www.php.net](http://www.php.net)

PHP mudah dipelajari dan berjalan efisien pada sisi server

## **2. PHP Install**

Komputer anda belum dapat menjalankan PHP, sehingga Anda Harus melakukan Instalasi:

install a web server

install PHP

install a database, such as MySQL

### 3. PHP syntax

#### PHP Syntax sederhana

PHP script dapat ditempatkan di manapun dalam dokumen. PHP script dimulai dengan `<? php` dan berakhir dengan `? >`:

```
<?
    echo ("Selamat Belajar PHP para peserta peraktikum");
?>
```

Ekstensi file default untuk file PHP adalah ".php". PHP file biasanya berisi tag HTML, dan PHP beberapa skrip kode.

#### Menuliskan Comments pada PHP

Sebuah komentar dalam kode PHP adalah garis yang tidak dibaca/dijalankan sebagai bagian dari program. Satu-satunya tujuan adalah untuk dibaca oleh seseorang yang mencari kode.

```
<?
//single-line comment
# single-line comment
/*
multiple-lines comment block
*/

echo ("Selamat Belajar PHP di Lab CC ");
$x = 5 /* + 15 */ + 5;
echo $x;
?>
```

#### PHP Case Sensitivity

Dalam PHP, semua keywords (misal: if, else, while, echo, dll), classes, functions, dan user-defined functions tidak case-sensitive.

```
<?php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>
```

Berbeda halnya dengan penamaan sebuah variable yang bersifat case-sensitive.

```
<?php
```

```
$color = "red";  
echo "My car is " . $color . "<br>";  
echo "My house is " . $COLOR . "<br>";  
echo "My boat is " . $coLOR . "<br>";  
?>
```

#### 4. PHP Variables

Variabel digunakan sebagai tempat penyimpanan data sementara. Data yang disimpan dalam variabel akan hilang setelah program selesai dieksekusi. Untuk penyimpanan data yang permanen, kita dapat menyimpan data di database atau di disk. Silahkan mengacu pada akses Database untuk mendalami penggunaan database, dan Akses File dan Folder untuk penyimpanan data di filesystem.

##### a. Deklarasi

Dalam PHP, variabel diawali dengan tanda \$, diikuti oleh nama variabel.

```
<?php  
$x=5;  
$y=6;  
$z=$x+$y;  
echo $z; //nilai Z akan di tampilkan  
?>
```

##### b. Peraturan penamaan variabel

Variabel diawali tanda \$, diikuti oleh nama variabel

Nama variabel diawali dengan huruf atau karakter garis bawah

Nama variabel tidak dapat diawali dengan angka

Nama variabel hanya dapat mengandung karakter alfa-numerik dan garis bawah (AZ, 0-9 dan \_)

Nama variabel bersifat case-sensitive (\$age dan \$AGE adalah dua variabel yang berbeda)

##### c. PHP Variables Scope

Dalam PHP, variabel dapat dideklarasikan di mana saja dalam script.

Lingkup variabel adalah bagian dari script dimana variabel dapat dirujuk/digunakan.

PHP memiliki tiga cakupan variabel yang berbeda, yaitu: local, global, dan static

**Local Variables**

```
<?php
function myTest() {
    $x = 5;
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();
echo "<p>Variable x outside function is: $x</p>";
?>
```

**Global Variables**

```
<?php
$x = 5;
$y = 10;
function myTest() {
    global $x, $y;
    $y = $x + $y;
}
myTest();
echo $y;
?>
```

PHP menyimpan semua variabel global dalam array yang disebut `$GLOBALS` [indeks]. Indeks merujuk pada nama variabel. Array dapat diakses dari dalam fungsi dan dapat digunakan untuk memperbarui variabel global secara langsung.

```
<?php
$x = 5;
$y = 10;
function myTest() {
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
}
myTest();
echo $y;
?>
```

**Static Variables**

```
<?php
```

```
function myTest() {  
    static $x = 0;  
    echo $x;  
    $x++;  
}  
myTest();  
myTest();  
myTest();  
?>
```

## 5. Tipe Data pada PHP

Dalam bahasa pemrograman yang lain, ada bermacam-macam tipe data, misalnya integer(bilangan bulat), float (bilangan pecahan), char(karakter angka dan huruf), string(kumpulan huruf atau kata), dan berbagai tipe lainnya.

PHP mengenal dua tipe data sederhana; numerik dan literal. Ditambah dengan dua tipe data yang tidak sederhana, yaitu array dan object. Tipe Numerik dapat menyimpan bilangan bulat. PHP mampu menyimpan data bilangan bulat dengan jangkauan dari -2 milyar sampai +2 milyar. Contoh bilangan bulat: 3, 7, 20.

Selain itu, tipe numerik juga digunakan untuk menyimpan bilangan pecahan. Tipe literal digunakan untuk menyimpan data berupa kumpulan huruf, kata, dan angka. Tipe boolean, yang dikenal dalam bahasa program yang lainnya, tidak ada dalam PHP.

Untuk menguji benar salah (true false), kita menggunakan tipe data yang tersedia. FALSE dapat digantikan oleh integer 0, double 0.0 atau string kosong, yaitu "". Selain nilai itu, semua dianggap TRUE.

Variabel dapat digunakan untuk menyimpan berbagai jenis data. Misalnya data numerik yang dapat dioperasikan secara matematika. Contoh :

```
$jumlahBarang = 3;  
$harga = 1000;  
$pembayaran = $jumlahBarang * $harga;
```

pada contoh di atas, variabel pembayaran akan menyimpan nilai 3000.

Sedangkan data non numerik (disebut juga data literal) tidak dapat dioperasikan secara matematika. Contoh :

```
$nama = $namaDepan + $namaBelakang;
```

variabel nama akan menyimpan gabungan dari dua variabel, yaitu “Endy Muhardin”.

Secara umum, data literal ditandai dengan pasangan “ dan “. Data numerik tidak dikelilingi oleh “ dan “. Tetapi biasanya PHP akan secara otomatis mengubah tipe data sesuai kebutuhan. Contoh :

```
$jalan = “Gubeng Kertajaya”;
```

```
$noRumah = 29;
```

```
$blok = 4c;
```

```
$jumlahPenghuni = 3;
```

```
$alamat = $jalan + $noRumah;
```

```
$hasil = $noRumah + $jumlahPenghuni;
```

```
$hasilAneh = $blok + $noRumah;
```

Pada sampel kode di atas, variabel alamat akan menyimpan nilai Gubeng Kertajaya 29. PHP secara otomatis mengubah tipe data variabel noRumah (numerik) menjadi literal. Variabel alamat akan bertipe literal.

Variabel hasil akan menyimpan nilai 32, yaitu penjumlahan dari 29 dan 3. Perhatikan, konversi otomatis ini kadang berjalan secara tidak semestinya. Ini dapat dilihat dari variabel hasilAneh yang akan menyimpan nilai 7. PHP mengambil nilai numerik dari variabel blok, yaitu 4, kemudian menambahkannya dengan isi variabel jumlah Penghuni. Hasil akhirnya adalah 4 + 3, yaitu 7.

## 6. PHP if..else ... elseif

### a. Pernyataan if

Pernyataan akan dijalankan hanya kalau bagian ekspresi bernilai benar.

Contoh :

```
<?php
$total_beli = 200000;
$keterangan = "Tak Dapat Diskon";

if($total_beli >= 100000)
    $keterangan = "Dapat Diskon";
print("$keterangan<br>\n");
?>
```

### b. Pernyataan If-else

Menjalankan suatu tindakan tertentu bila kondisi benar dan menjalankan tindakan yang lain jika kondisi bernilai salah.

Contoh :

Hari ini :<br>

```
<?php
$nama_hari = date("l");
if ($nama_hari == "Sunday")
    print("Minggu");
elseif ($nama_hari == "Monday")
    print("Senin");
elseif ($nama_hari == "Tuesday")
    print("Selasa");
elseif ($nama_hari == "Wednesday")
    print("Rabu");
elseif ($nama_hari == "Thursday")
    print("Kamis");
elseif ($nama_hari == "Friday")
    print("Jum'at");
else
    print("Sabtu");
?>
```



## 7. PHP Switch

Switch merupakan bentuk struktur kontrol yang dapat lebih menyederhanakan bentuk struktur if...else ataupun if ...elseif. jika pada struktur if program akan memilih beberapa kondisi, dalam switch program hanya akan mencocokkan. Jadi perintah dalam program akan dilaksanakan apabila perintah tersebut sesuai dengan kondisi.

sintaks dasar struktur switch dalam php :

```
<?php
switch ($variabel)
{
    case <nilai1> :
        perintah 1;
        perintah ...n;
        break;
    case <nilai2>:
        perintah 1;
        perintah ...n;
        break;
    case <nilai...n>:
        perintah 1;
        perintah ...n;
        break;
    default :
        perintah 1;
        perintah ...n;
        break;
} ?>
```

## 8. PHP While Loops

Loop mengeksekusi blok kode jumlah tertentu kali, atau saat kondisi tertentu adalah benar.

Seringkali ketika Anda menulis kode, Anda ingin blok kode yang sama untuk menjalankan lagi dan lagi dalam satu baris. Alih-alih menambahkan baris hampir sama beberapa script kita dapat menggunakan loop untuk melakukan tugas seperti ini.

Dalam PHP, kita memiliki pernyataan perulangan berikut:

**While – loop** melalui blok kode sementara kondisi yang ditentukan benar

Lakukan ... **while – loop** melalui blok kode satu kali, kemudian mengulangi loop selama kondisi yang ditentukan benar

**Foreach – loop** melalui blok kode untuk setiap elemen dalam array

Contoh :

```
<?php
$i=1;
while($i<=5)
{
    echo "The number is " . $i . "<br />";
    $i++;
}
?>
```

## 9. PHP For Loops

Pengulangan atau looping dalam jumlah tertentu

```
<?php
for ($x=0; $x<=10; $x++)
{
    echo "The number is: $x <br>";
}
?>
```