



# PHP # 2

---

Anna Baita, M.kom



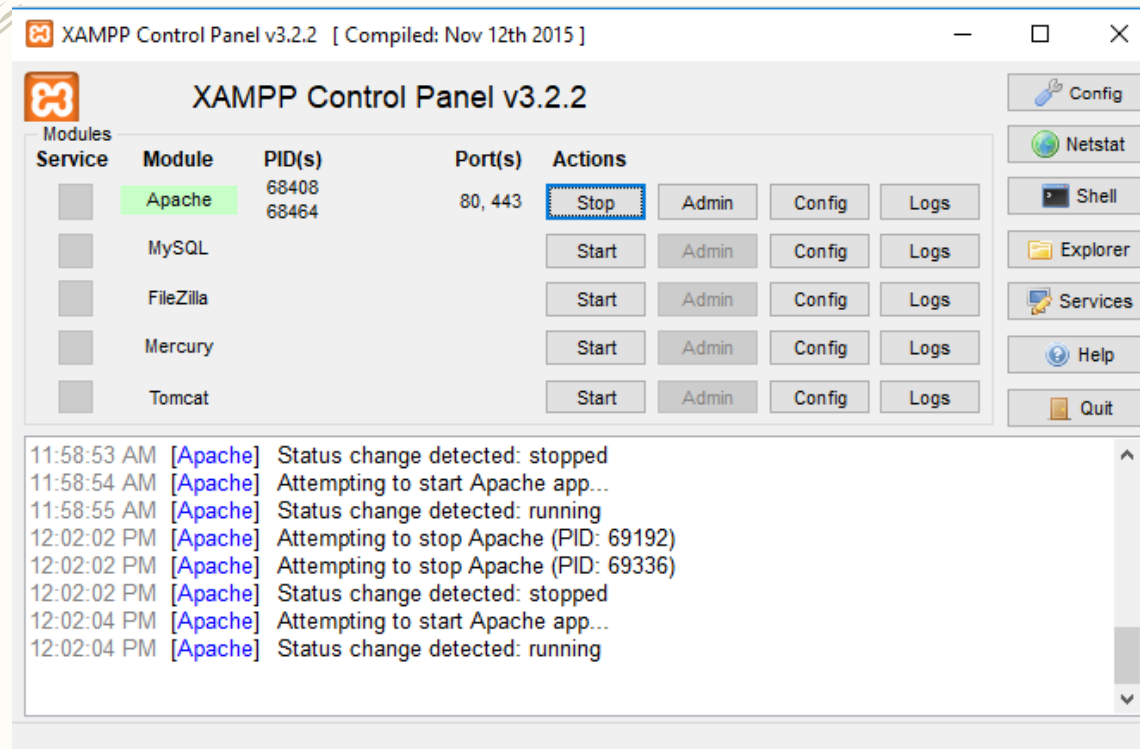
# Apa itu File php?

---

- File PHP dapat memuat Text, HTML, CSS, Java script dan kode php
- Kode php dieksekusi di server dan hasilnya ditampilkan di browser dalam bentuk plain HTML
- File Php dibuat dengan extensi file “.php”

# Menjalankan File PHP

- Sebelum menggunakan php, pastikan bahwa Web server telah aktif



# Menjalankan File PHP con't

---

- Buat File PHP
- Letakkan file php ke dalam directory documentroot. Default Lokasi documentroot, ada di directory:
  - ❑ xampp/htdocs (pada xampp)
  - ❑ apache/htdocs (pada apache)
- Jalankan File PHP dengan Mengetikkan url di browser:
  - ❑ **localhost:port/namafile.php** atau **127.0.0.1:port/namafile.php**

Jika port yang digunakan adalah default(port 80) maka tidak perlu dituliskan nomor portnya.
- Jika telah selesai, jangan lupa matikan web server

# PHP SYNTAX

---

- Script php harus diawali dengan **<?php** dan diakhiri dengan **?>**
- Statement php diakhiri dengan semicolon **“;”**
- ```
<?php  
    //isi code php  
?>
```
- Script PHP bisa digabungkan dengan script html



# variabel

---

- Sebuah variabel di buat menggunakan simbol **\$** kemudian diikuti nama variabel nya
- Berbeda dengan beberapa bahasa pemrograman lain, variabel dalam php tidak perlu dideklarasikan terlebih dahulu, langsung diisikan value nya
- Tipe data variabel akan menyesuaikan dari konten value nya
- Contoh:

```
$string="ini contoh string";
```

```
$angka=17;
```

```
$desimal=17.8;
```





# Aturan penulisan variabel

---

- Case sensitive
- Harus diawali dengan huruf ataupun karakter underscore
- Tidak boleh diawali dengan angka
- Nama variabel dapat memuat alpha-numeric (huruf & angka) serta underscore



# Php output(echo, print, printf)

---

- Php memiliki 3 fungsi untuk menampilkan output/mencetak teks ke layar:
  - ❑ Fungsi echo
  - ❑ Fungsi print
  - ❑ Fungsi printf
- Tidak jauh berbeda antara print & echo
- Echo tidak memiliki return value, print memiliki return value
- Echo lebih cepat dieksekusi daripada print





# Fungsi Echo

---

- Digunakan untuk menampilkan teks di layar
- Dapat digunakan dengan tanda kurung maupun tanda kurung
- Boleh digunakan untuk mengoutputkan 2 parameter atau lebih

Contoh:

```
<?php
```

```
echo "<h2>Html di dalam PHP</h2><br>";
```

```
echo("Penggunaan Tanda Kurung <br>");
```

```
echo "Aku sedang belajar PHP!<br>";
```

```
echo "Ini ", "contoh ", "penggunaan ", "parameter ", "lebih  
dari satu";
```

```
?>
```



# Fungsi Echo

---

- Fungsi Echo tidak memiliki return value

Contoh:

```
<?php
    $cetak=echo"perintah echo tidak memiliki return value";
    echo $cetak;

?>
```

# Fungsi Print

- Digunakan untuk menampilkan teks di layar
- Dapat digunakan dengan tanda kurung maupun tanda kurung
- **Tidak bisa** digunakan untuk mengoutputkan 2 parameter atau lebih

Contoh:

```
<?php  
    print "<h2>Html di dalam PHP</h2><br>";  
    print("Penggunaan Tanda Kurung <br>");  
    print "Aku sedang belajar PHP!<br>";  
    print "Ini ", "akan ", "dianggap ", "error ";  
  
?>
```



# Fungsi Print

---

- Fungsi Print memiliki return value

Contoh:

```
<?php
    $cetak=print"perintah echo tidak memiliki return value";
    print $cetak;

?>
```



# Menggabungkan dua string/ teks

---

- Dalam fungsi echo untuk menggabungkan dua string dengan memberikannya sebagai argument (dipisah dengan tanda koma (,))
- Bisa dilakukan dengan menambahkan tanda titik
- Contoh:

```
echo "menggabungkan ","dua ","string <br>";  
echo "menggabungkan "."pake "."titik <br>";  
print "menggabungkan "."pake "."titik <br>";
```



# Fungsi Printf

---

- Fungsi printf merupakan fungsi untuk mengoutputkan string, serta memformat string
- Fungsi ini akan memberikan return value berupa panjang teks saat dieksekusi
- fungsi ini dituliskan dengan menggunakan tanda kurung

Contoh:

```
<?php  
    printf (" Menulis dengan Printf<br>");  
    $cetak=printf ("berapa karakter??");  
    printf ($cetak);  
?>
```

# Fungsi Printf

---

- Simbol %s merupakan sebuah placeholder untuk teks/string
- Simbol %d merupakan sebuah placeholder untuk bilangan decimal
- Simbol %f merupakan sebuah placeholder untuk tipe float
- Simbol %b merupakan sebuah placeholder untuk tipe boolean

Contoh: `<?php`

```
$buku="pemrograman";  
printf("Dia membeli buku %s ",$buku."<br>");  
$harga=10000;  
printf("harga buku tersebut adalah Rp %.2f",$harga);  
$diskon=0.025;  
printf("mendapat diskon %d",$diskon);  
printf("mendapat diskon %f",$diskon);  
?>
```






# PHP Variables Scope

---


- Variabel php dapat di deklarasikan dibagian mana saja dalam sebuah script
- Ruang lingkup variable adalah cakupan dimana sebuah variabel dapat digunakan/dijadikan referensi
- Berdasarkan lingkupnya ada 3 macam variable:
  - Local
  - Global
  - Static

# Global scope



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <?php
6 $x = 5; // var global
7
8 function myFunction() {
9     // x diluar fuction, sehingga tidak bisa diakses di function
10     echo "<p>Variable x di dalam function adalah: $x</p>";
11 }
12 myFunction();
13
14 echo "<p>Variable x di luar function adalah: $x</p>";
15 ?>
16
17 </body>
18 </html>
```

# Local scope



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <?php
6 function myFunction() {
7     $x = 5; // local scope
8     echo "<p>Variable x di dalam function adalah: $x</p>";
9 }
10 myFunction();
11
12 echo "<p>Variable x di luar function adalah: $x</p>";
13 ?>
14
15 </body>
16 </html>
```


# Penggunaan global keyword

Global Keyword:  
untuk mengakses  
variable global di  
dalam function

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <?php
6 $x = 5;
7 $y = 10;
8
9 function myFunction() {
10     global $x, $y;
11     $y = $x + $y;
12
13 }
14
15 myFunction();
16 echo $y; // outputs 15
17 ?>
18
19 </body>
20 </html>
```


# Meng-set variable global

---



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <?php
6 $x = 15;
7 $y = 100;
8
9 function myFunction() {
10     $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
11 }
12
13 myFunction();
14 echo $y; // outputs 15
15 ?>
16
17 </body>
18 </html>
```

# Keyword static



Global static: agar  
variable local tidak  
dihapus meski  
fungsi selesai  
dijalankan

```
4
5 <?php
6 function myFunction() {
7
8     static $x = 0;
9     echo $x;
10    $x++;
11 }
12
13 myFunction();
14 echo "<br>";
15 myFunction();
16 echo "<br>";
17 myFunction();
18 ?>
19
```



# konstanta


---

- Merupakan pengenalan sebuah nilai sederhana
- Nilai tidak bisa diubah di dalam script
- Konstanta tidak diawali dengan symbol \$
- Konstanta secara otomatis menjadi variable global
- Syntax:
  - `define(name, value, case-insensitive)`
- Defaultnya case-sensitive



# contoh

---



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <?php
6 define("Hello", "Hello world!");
7 echo Hello;
8 ?>
9
10
11 </body>
12 </html>
```



# PHP Global Variables - Superglobals

---

- `$GLOBALS`
- `$_SERVER`
- `$_REQUEST`
- `$_POST`
- `$_GET`
- `$_FILES`
- `$_COOKIE`
- `$_SESSION`

# PHP \$\_REQUEST

\$\_REQUEST  
digunakan untuk  
mengambil data  
dari form html

```
7 <form method="get" action="<?php echo $_SERVER['PHP_SELF'];?>">
8   Name: <input type="text" name="fname">
9   <input type="submit">
10 </form>
11
12 <?php
13 if ($_SERVER["REQUEST_METHOD"] == "GET") {
14     // collect value of input field
15     $name = $_GET['fname'];
16     if (empty($name)) {
17         echo "Name is empty";
18     } else {
19         echo $name;
20     }
21 }
22 ?>
23
```