

**MODUL PHP 2018**  
**PENGEMBANGAN APLIKASI BERBASIS WEB**

David Setiadi,S.Kom.,M.T.

Sumber : [www.w3schools.com](http://www.w3schools.com)



PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum (wikipedia). PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di <http://www.php.net>.

PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client).

Pada awalnya PHP merupakan singkatan dari Personal Home Page. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dll.

Saat ini PHP adalah singkatan dari PHP: Hypertext Preprocessor, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: PHP: Hypertext Preprocessor.

PHP dapat digunakan dengan gratis (free) dan bersifat Open Source. PHP dirilis dalam lisensi PHP License, sedikit berbeda dengan lisensi GNU General Public License (GPL) yang biasa digunakan untuk proyek Open Source.

Sebelum mempelajari PHP sebaiknya kuasai dahulu HTML, CSS, JavaScript.

What is PHP?

- Singkatan dari "PHP: Hypertext Preprocessor"
- PHP adalah bahasa pemrograman Opensource dan dangat luas
- PHP merupakan server side scripting atau di eksekusi diserver
- PHP dapat didownload gratis di web resminya

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: [www.php.net](http://www.php.net)
- PHP is easy to learn and runs efficiently on the server side

Editor :

- Notepad
- Dreamweaver
- Sublime

- Brackets
- Dll

Install :

- Install Wevserver (Xampp)
- Install PHP
- Install Database Server (Mysql)

## MEMULAI PHP

- Untuk memulai file php harus menggunakan tag pembuka dan penutup.
- Php dapat ditempatkan dimana saja didalam file HTML

```
<?php
// Ketik Kode PHP disini
?>
```

PHP didalam HTML

```
<html>
<body>
<?php
// Ketik Kode PHP disini
?>
</body>
</html>
```

HTML didalam PHP

```
<?php
echo "<html>
<body>
</body>
</html>
"
?>
```

## PHP Tidak Case Sensitif kecuali penulisan Variable

```
<html>
<body>

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

$color = "Merah";
echo "Mobil saya berwarna " . $color . "<br>";
echo "Rumah saya berwarna " . $COLOR . "<br>";
echo "Motor saya berwarna " . $coLOR . "<br>";

?>

</body>
</html>
```

## KOMENTAR

Penulisan Komentar dalam PHP menggunakan dobel slash //

komentar tidak akan di eksekusi sebagai program

```
<?php
//$txt = "Hello world!";
//$x = 5;
//$y = 10.5;

//$txt = "STMIK Sumedang";
//echo "I love $txt!";
?>
```

## VARIABEL

Variabel adalah tempat menyimpan data atau informasi

Rules for PHP variables:

- A variable starts with the \$ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_)
- Variable names are case-sensitive (\$age and \$AGE are two different variables)

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;

$txt = "STMIK Sumedang";
echo "I love $txt!";
?>
```

## Menyatukan 2 variabel

```
<?php
$txt1 = "I Love";
$txt2 = "STMIK Sumedang";
echo "Hallo ".$txt1.$txt2;
?>
```

```
<?php
$x = 5;
$y = 4;
echo $x + $y;
?>
```

## OPERATOR

### Operator Aritmetika

Operator	Keterangan	Contoh	Hasil
+	Penambahan	x=2 x+2	4
-	Pengurangan	x=2 5-x	3
*	Perkalian	x=4 x*5	20
/	Pembagian	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

### Operator Penugasan (Assignment Operators)

Operator	Contoh	Persamaannya
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

### Operator Perbandingan (Comparison Operators)

Operator	Keterangan	Contoh
==	sama dengan	5==8 returns false
!=	tidak sama dengan	5!=8 returns true
>	lebih besar dari	5>8 returns false
<	kurang dari	5<8 returns true
>=	lebih besar atau sama dengan	5>=8 returns false
<=	kurang dari atau sama dengan	5<=8 returns true

### Operator Logika (Logical Operators)

Operator	Keterangan	Contoh
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
	or	x=6 y=3 (x==5    y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

### TYPE DATA

PHP supports the following data types:

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

### STRING

Menghitung panjang karakter :

```
<?php
echo strlen("Hello world!"); // outputs 12
?>
```

Menghitung jumlah kata

```
<?php
echo str_word_count("Hello world!"); // outputs 2
?>
```

## Membalikan string

```
<?php
echo strrev("Hello world!"); // outputs !dlrow olleH
?>
```

## Replace String

```
<?php
echo str_replace("world", "Dolly", "Hello world!"); // outputs Hello Dolly!
?>
```

## Transform Huruf besar semua

```
<?php
echo strtoupper("Hello WORLD!");
?>
```

## Transform Huruf besar semua

```
<?php
echo strtoupper("Hello WORLD!");
?>
```

**Fungsi date() digunakan untuk memformat waktu dan tanggal.**

### Sintaks

```
date(format,timestamp)
```

Parameter	Keterangan
format	Required. Specifies the format of the timestamp
timestamp	Optional. Specifies a timestamp. Default is the current date and time (as a timestamp)

### TIMESTAMP/DATE/ WAKTU

Timestamp adalah jumlah detik sejak January 1, 1970 00:00:00 GMT. Juga dikenal sebagai Unix Timestamp.

### Format Tanggal

- d – format hari (01-31)
- m – format bulan (01-12)
- Y – format tahun

### Contoh Program :

```
<?php
echo date("Y/m/d");
echo "<br />";
echo date("Y.m.d");
echo "<br />";
echo date("Y-m-d");
?>
```

output:

```
2018/07/11
2018.07.11
2018-07-11
```

Fungsi **mktime()** akan memberikan nilai Unix timestamp untuk tanggal tertentu.

Sintaks

```
mktime(hour,minute,second,month,day,year,is_dst)
```

Program14-2.php

```
<?php
$besok = mktime(0,0,0,date("m"),date("d")+1,date("Y"));
echo "Besok Adalah ".date("Y/m/d/", $besok);
?>
```

output:

```
Besok Adalah 2018/07/12
```

Jika running menggunakan Xampp terjadi error date time kemungkinan ada perbedaan waktu antara xampp dan waktu lokal atau komputer anda, maka lakukan setting manual pada :  
C:\xampp\php\php.ini

```
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = "Asia/Jakarta"
```

Bisa juga menggunakan kode php berikut ini :

```
<?php
date_default_timezone_set('Asia/Jakarta');
date();
?>
```

## PHP Conditional Statements

In PHP we have the following conditional statements:

- if statement - executes some code if one condition is true
- if...else statement - executes some code if a condition is true and another code if that condition is false
- if...elseif....else statement - executes different codes for more than two conditions
- switch statement - selects one of many blocks of code to be executed

### Statement IF

```
<?php
$t = date("H");

if ($t < "20") {
    echo "Have a good day!";
}
?>
```



## Statement IF...Else

```
<?php
$t = date("H");
if ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

## Statement IF...Elseif...else

```
<?php
$t = date("H");

if ($t < "10") {
    echo "Have a good morning!";
} elseif ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

## Statement Switch

```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

## Looping /Perulangan (While)

```
<?php
$x = 1;

while($x <= 5) {
    echo "The number is: $x <br>";
    $x++;
}
?>
```

## Looping /Perulangan (For)

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

```
<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
    echo "$value <br>";
}
?>
```

## Function /Fungsi

```
<?php
function writeMsg() {
    echo "Hello world!";
}

writeMsg(); // call the function
?>
```

## Array

Array digunakan untuk menyimpan satu atau lebih nilai pada sebuah nama variabel.

Jenis-jenis Array

- Numeric array – Array dengan sebuah numeric ID key.
- Associative array - Array dimana setiap ID-nya berasosiasi dengan suatu nilai.
- Multidimensional array - Array yang terdiri dari satu atau lebih array.

### Numeric Array

#### Contoh 1

Pada contoh ini kunci ID secara otomatis di beri suatu nilai.

```
$names = array("Budi","Maman","Asep");
```

#### Contoh 2

Pada contoh ini kita memberikan nilai pada kunci ID secara manual.

```
<?php
$names[0] = "Budi";
$names[1] = "Maman";
$names[2] = "Asep";
echo $names[1] . " dan " . $names[2] .
" adalah teman ". $names[0];
?>
```

Output program:

```
Budi dan Maman adalah teman Asep
```

## Associative Arrays

### Contoh 1

Pada contoh ini kita menggunakan sebuah array untuk memberikan nilai umur pada beberapa orang yang berbeda.

```
$ages = array("Budi"=>32, "Maman"=>30, "Asep"=>34);
```

### Contoh 2

Pada contoh ini sama dengan diatas, hanya saja kita memperlihatkan cara yang lain dalam membuat array.

```
$ages['Budi'] = "32";  
$ages['Maman'] = "30";  
$ages['Asep'] = "34";
```

```
<?php  
$ages['Budi'] = "32";  
$ages['Maman'] = "30";  
$ages['Asep'] = "34";  
echo "Umur Budi adalah " . $ages['Budi'] . " Tahun."  
?>
```

Output program:

```
Umur Budi adalah 32 Tahun
```

## Multidimensional Arrays

### Contoh 1 cara inisialisasi multidimensional array

```
$families = array  
(  
    "Keluarga1"=>array  
    (  
        "Budi",  
        "Maman",  
        "Asep",  
    ),  
    "Keluarga2"=>array  
    (  

```

```

    "Glenn"
  ),
  "Keluarga3" => array
  (
    "Andi",
    "Rudi",
    "Zaenal"
  )
);

```

Array di atas akan terlihat seperti di bawah ini jika dituliskan ke output.

```

Array
(
    [Keluarga1] => Array
        (
            [0] => Budi
            [1] => Maman
            [2] => Asep
        )
    [Keluarga2] => Array
        (
            [0] => Glenn
        )
    [Keluarga3] => Array
        (
            [0] => Andi
            [1] => Rudi
            [2] => Zaenal
        )
)

```

## FORM HANDLING

Perintah \$\_GET dan \$\_POST digunakan untuk mengirim informasi dari form, seperti input user.

Contoh : Buat halaman dengan nama input.php

```

<html>
<body>

<form action="welcome.php" method="post">
Nama: <input type="text" name="nama"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>

```

Kemudian buat halaman dengan nama welcome.php

```

<html>
<body>

Selamat Datang <?php echo $_POST["nama"]; ?><br>
Email anda adalah : <?php echo $_POST["email"]; ?>

</body>
</html>

```

Maka Outputnya adalah menampilkan text yang diinput pada halaman input.php

Untuk penggunaan GET sama saja tinggal mengganti POST dengan GET

Contoh : Buat halaman dengan nama input.php

```
<html>
<body>

<form action="welcome.php" method="get">
Nama: <input type="text" name="nama"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

Kemudian buat halaman dengan nama welcome.php

```
<html>
<body>

Selamat Datang <?php echo $_GET["nama"]; ?><br>
Email anda adalah : <?php echo $_GET["email"]; ?>

</body>
</html>
```

Maka Outputnya adalah menampilkan text yang diinput pada halaman input.php

Mempunyai pengertian bahwa metode ini digunakan untuk memarsing hasil inputan form untuk diolah lebih lanjut, tetapi tidak menampilkan query string pada address bar, sehingga lebih aman dibandingkan method GET yang akan terlihat pada address bar. Method POST akan sangat berguna penggunaannya untuk halaman-halaman yang membutuhkan keamanan lebih, karena data inputan kita tidak akan tampil pada address bar browser (disini kelebihan menggunakan Metode POST).

Nilai parsing tidak ditampilkan dalam address bar dan langsung di proses ketempat yang sudah dibuat. Sebenarnya nilai dari parshingan tetap ada tetapi tidak tampak didalam address bar karena valuenya bekerja dibelakang layar (backand). Disinilah keunggulan menggunakan Method POST.

Perbedaan yang lain dari method POST adalah:

- Method POST tidak pernah ter Cache dalam browser
- Method POST tidak dapat di Bookmark
- Method POST tidak mempunyai batasan jumlah panjang data

Sedangkan GET adalah metode pengiriman data menggunakan query string atau menampilkan nilainya pada address bar. Jadi semua nilai pada form akan dikirim kesisi server yang kemudian diterima oleh file tertentu dengan ketentuan nilai form tersebut akan tampil didalam address bar browser/URL browser. Untuk suatu keamanan method POST lebih aman dari method GET.

Perbedaan yang lain dari method GET adalah:

- Method GET dapat di cache dalam browser
- Method GET dapat di bookmark
- Method GET mempunyai batasan jumlah panjang data yaitu maksimal 2048 karakter

## FORM VALIDATION

Digunakan untuk memvalidasi inputan form yang bisa dilakukan di sisi client dan di sisi server. Disisi client biasanya dilakukan di browser dengan menambahkan kode validasi pada script html, misal required (inputan wajib diisi). Berbeda dengan validasi disisi Server, yaitu data divalidasi setelah sampai

```
<!DOCTYPE html>
<html>
<head>
<title>Membuat Form Validasi Dengan PHP</title>
</head>
<body>
<h1>Membuat Form Validasi Dengan PHP </h1>

    <?php
    if(isset($_GET['nama'])){
        if($_GET['nama'] == "kosong"){
            echo "<h4 style='color:red'>Nama Belum Di Masukkan !</h4>";
        }
    }
    ?>

    <h4>Masukkan Nama Anda :</h4>
    <form action="cek.php" method="post">
    Nama <input type="text" name="nama">
    <input type="submit" value="Cek">
    </form>

</body>
</html>
```

di server kemudian dieksekusi dan tentukan nilainya sesuai atau tidak.

Cek.php

```

<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>

<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
    }

    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    } else {
        $email = test_input($_POST["email"]);
    }
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>

<h2>PHP Form Validation Example</h2>
<p><span class="error">* required field.</span></p>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
    Name: <input type="text" name="name">
    <span class="error">* <?php echo $nameErr;?></span>
    <br><br>
    E-mail: <input type="text" name="email">
    <span class="error">* <?php echo $emailErr;?></span>
    <br><br>
    <input type="submit" name="submit" value="Submit">
</form>

<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
echo $email;
echo "<br>";
echo $website;
echo "<br>";
echo $comment;
echo "<br>";
echo $gender;
?>

</body>
</html>

```

## INCLUDE

Server Side Includes (SSI) digunakan untuk menyimpan fungsi, header, footer, atau elemen-elemen yang dapat digunakan pada halaman yang berlainan. Server Side Includes ,

### Fungsi include()

Fungsi include() akan mengambil semua teks pada file include dan mengkopinya ke file tujuan.

### Fungsi require()

Fungsi require() sama dengan include(), tetapi berbeda dalam cara penanganan kesalahan.

Fungsi include() akan menghasilkan peringatan (dan program akan melanjutkan eksekusinya) sedangkan fungsi require() akan menghasilkan fatal error dan menghentikan program.

Contoh :

Buat footer.php

```
<?php
echo "<p>Copyright &copy; 1999-" . date("Y") . " stmik-sumedang.ac.id</p>";
?>
```

Buat welcome.php

```
<html>
<body>

<h1>Selamat Datang DiWeb Kami</h1>
<p>Some text.</p>
<p>Some more text.</p>
<?php include 'footer.php';?>

</body>
</html>
```

## FILE HANDLING

Digunakan untuk untuk read, add, edit file menggunakan php.

Mode	Keterangan
r	Read only. Starts at the beginning of the file
r+	Read/Write. Starts at the beginning of the file
w	Write only. Opens and clears the contents of file; or creates a new file if it doesn't exist
w+	Read/Write. Opens and clears the contents of file; or creates a new file if it doesn't exist
a	Append. Opens and writes to the end of the file or creates a new file if it doesn't exist
a+	Read/Append. Preserves file content by writing to the end of the file
x	Write only. Creates a new file. Returns FALSE and an error if file already exists
x+	Read/Write. Creates a new file. Returns FALSE and an error if file already exists

Contoh :



Buatlah file kamus.txt dengan notepad

```
AJAX = Asynchronous JavaScript and XML
CSS = Cascading Style Sheets
HTML = Hyper Text Markup Language
PHP = PHP Hypertext Preprocessor
SQL = Structured Query Language
SVG = Scalable Vector Graphics
XML = EXTensible Markup Language
```

```
<?php
$myfile = fopen("kamus.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("webdictionary.txt"));
fclose($myfile);
?>
```

### Memeriksa EOF (End Of File)

**Catatan:** Kita tidak dapat membaca file yang terbuka dalam mode w, a, dan x!

```
if (feof($file)) echo "End of file";
```

### Membaca file baris per baris (fgets())

```
<?php
$file = fopen("welcome.txt", "r") or exit("Unable to open file!");
//Output a line of the file until the end is reached
while(!feof($file))
{
    echo fgets($file). "<br />";
}
fclose($file);
?>
```

### Membaca file karakter per karakter (fgetc())

```
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
while (!feof($file))
{
    echo fgetc($file);
}
fclose($file);
?>
```

## FILE UPLOAD

Digunakan untuk mengupload file ke server menggunakan PHP, tetapi ini akan sangat berbahaya jika scriptnya terbuka tanpa batasan session atau hak akses karena berpotensi mudah di Hack.

Sebelumnya pastikan setting pada php.ini sudah di On kan : `file_uploads = On`

```
<html>
<body>

<form action="upload.php" method="post" enctype="multipart/form-data">
    Select image to upload:
    <input type="file" name="fileToUpload" id="fileToUpload">
    <input type="submit" value="Upload Image" name="submit">
</form>

</body>
</html>
```

```
?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
    if($check !== false) {
        echo "File is an image - " . $check["mime"] . ".";
        $uploadOk = 1;
    } else {
        echo "File is not an image.";
        $uploadOk = 0;
    }
}
?>
```

Pastikan **folder** untuk menampung file upload sudah sesuai

Berikut skrip lengkap untuk memfilter duplikat file, size dan type file.

## COOKIES

## SESSION

```

<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
    if($check !== false) {
        echo "File is an image - " . $check["mime"] . ".";
        $uploadOk = 1;
    } else {
        echo "File is not an image.";
        $uploadOk = 0;
    }
}
// Check if file already exists
if (file_exists($target_file)) {
    echo "Sorry, file already exists.";
    $uploadOk = 0;
}
// Check file size
if ($_FILES["fileToUpload"]["size"] > 500000) {
    echo "Sorry, your file is too large.";
    $uploadOk = 0;
}
// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
&& $imageFileType != "gif" ) {
    echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
    $uploadOk = 0;
}
// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded.";
// if everything is ok, try to upload file
} else {
    if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
        echo "The file " . basename( $_FILES["fileToUpload"]["name"]). " has been uploaded.";
    } else {
        echo "Sorry, there was an error uploading your file.";
    }
}
?>

```