Pemrograman Web

Sirojul Munir | rojulman@nurulfikri.ac.id

Pengantar PHP Programming

Sirojul Munir | rojulman@nurulfikri.ac.id

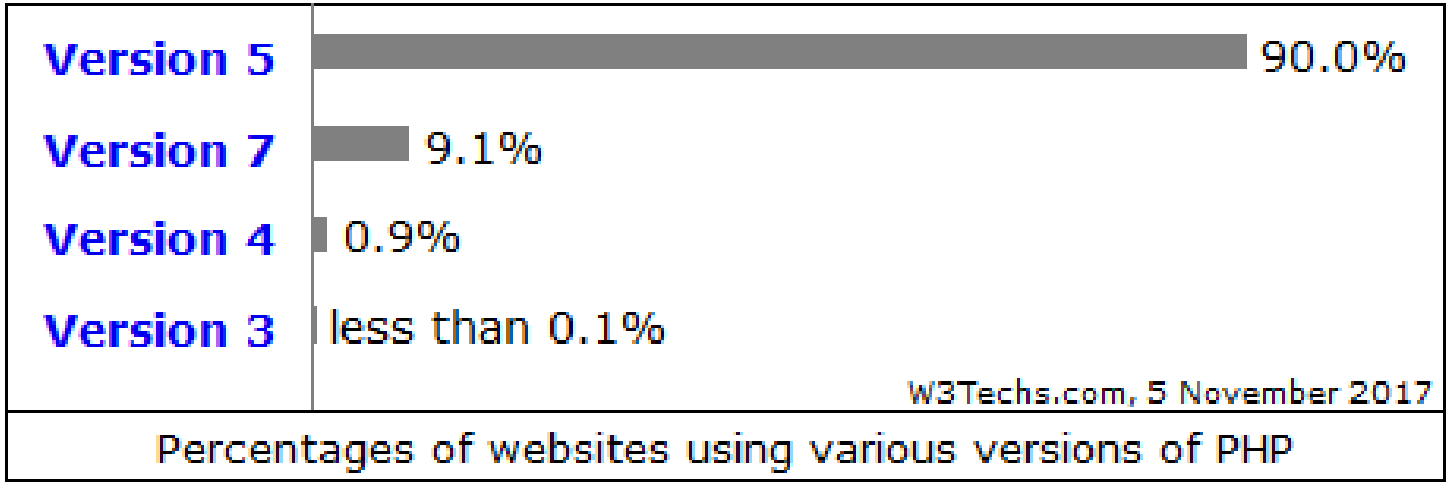
History PHP

- 1994: Created by Rasmus Lerdorf, software engineer (part of Apache Team)
- 1995: Called Personal Home Page Tool, then released as version 2 with name PHP/FI (Form Interpreter, to analyze SQL queries)
- Half 1997: used by 50,000 web sites
- October 1998: used by 100,000 websites
- End 1999: used by 1,000,000 websites
- Web : php.net

Alternative PHP

- Practical extraction and Report Language (Perl)
- Active Server Pages (ASP)
- Java server pages (JSP)
- Ruby
- Python (Django)

PHP Version



Percentages of websites using various versions of PHP

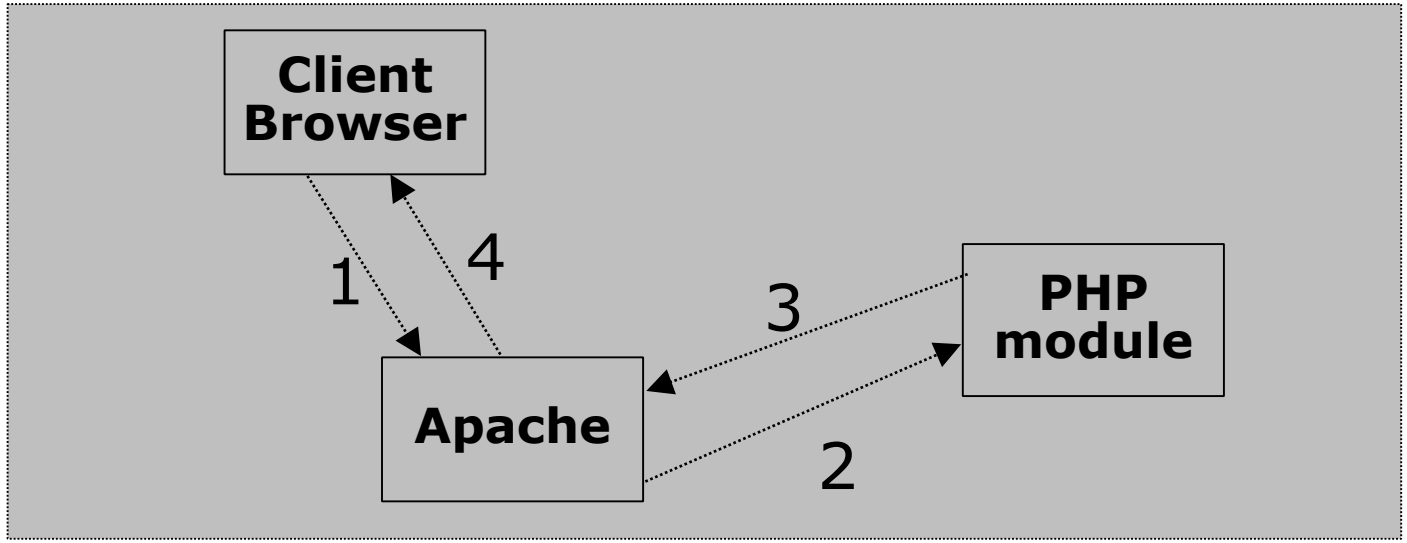
PHP Software :

1. Linux Distribution (Apache Web Server + PHP Module)
2. Windows / Mac : XAMPP, WAMP , LARAGON

Why PHP Popular ?

- Open-source & Free
- Easy to use (C-like and Perl-like syntax)
- Stable and fast
- Multiplatform
- Many databases support
- Many common built-in libraries
- Pre-installed in Linux distributions
- *PHP5* – better object-oriented and XML language support
- PHP7 – More Fast

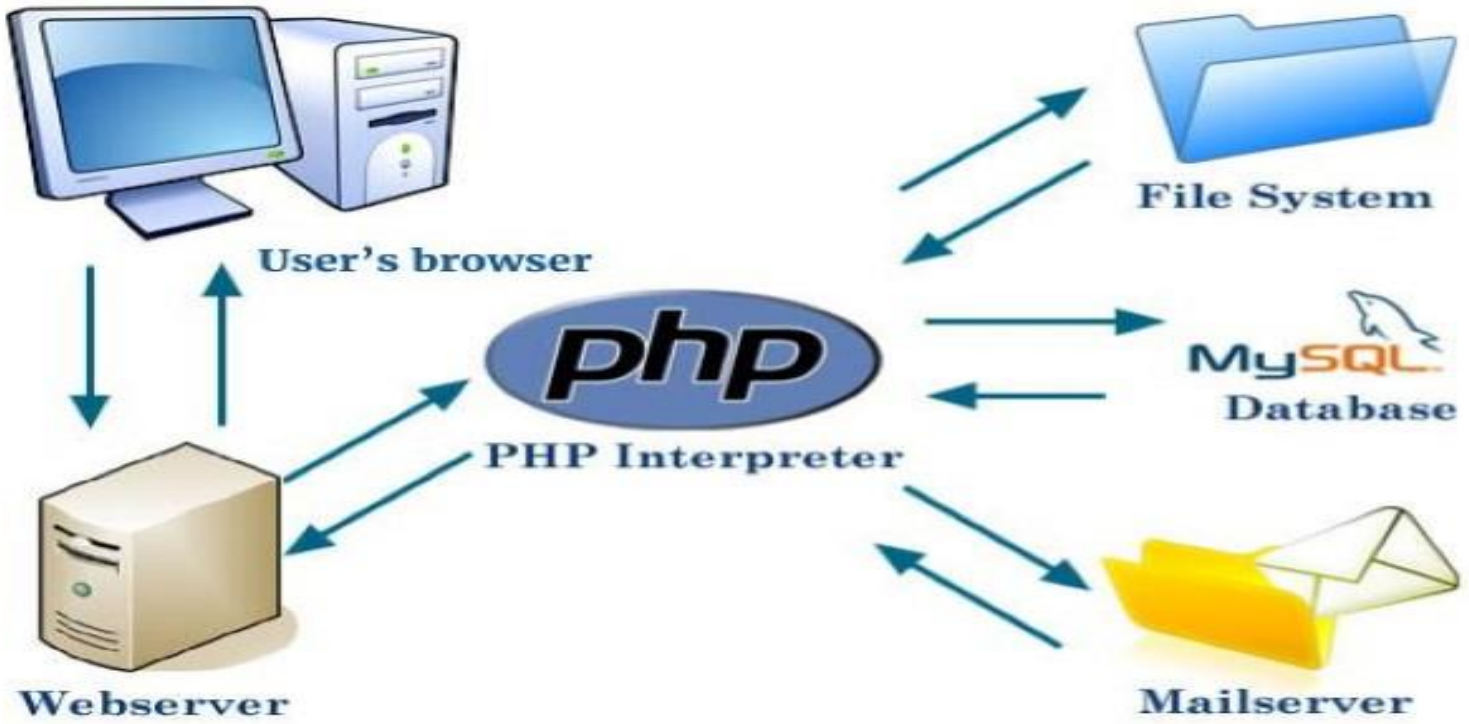
How PHP Work ?



1: Client from browser send HTTP request (with POST/GET variables)
2: Apache recognizes that a PHP script is requested and sends the request to PHP module
3: PHP interpreter executes PHP script, collects script output and sends it back
4: Apache replies to client using the PHP script output as HTML output

PHP Interpreter

- PHP is an interpreted language
  - Scripts are interpreted by PHP's Zend parsing engine
  - Output is HTML



PHP Configuration

- Konfigurasi ada pada file php.ini
  - Linux Ubuntu 16.04 : /etc/php/7.0/apache/php.ini
- Development Environment
  - display_errors = ON
  - error_reporting = E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED & ~E_WARNING

PHP Basic Syntax

```
<?php
   // Your script is here
?>
```

- File PHP berekstensi .php
- Setiap statement (pernyataan program) diakhir ; (titik koma)

PHP Basic Syntax - Comment

```
<?php
  // satu baris komentar
  # ini juga satu baris komentar

  /*
    banyak baris komentar
   */

  // komentar php juga bisa diantara baris kode
  $x = 5 /* +15 */ + 5 ;
  echo $x;
 ?>
```

PHP Basic Syntax - Variables

- Variabel : Tempat menyimpan nilai selama program berjalan
- Variable diawali dengan tanda $ setelah nama variable, contoh *$var_name*
- Valid variable names
  - Diawali dengan huruf atau garis bawah ($nama, $_alamat)
  - Dapat mengandung huruf, angka atau garis bawah
  - Tidak boleh menggunakan kata kunci (keyword) PHP (contoh:. “class”)
  - Tidak boleh menggunakan spasi, tanda (“.” , “,” , “[“, “]”)

PHP Basic Syntax - Variables

- Memberi nilai variable :
  - By Value:: $myVar = 25;
  - By Reference:: $myRef = &$myVar;
  - Invalid:: $myRef = &(7 * 52) //Expression is unnamed
- Variable tidak perlu dideklarasikan terlebih dahulu
- PHP variable adalah CASE-SENSITIVE

PHP Basic Syntax – print vs echo

- Echo dan print adalah memiliki fungsi yang sama, mencetak output data ke layar display user
- Perbedaannya:
  - Echo tidak mengembalikan suatu nilai, sedang print mengembalikan nilai 1 karenanya bisa digunakan sebagai ekspresi program
  - Echo dapat menerima banyak parameter , print hanya 1 parameter
  - Echo lebih cepat disbanding print

```
<?php
    $nama = 'Putri Ramadhani' ;
    $_umur = 8;
    echo 'Nama Siswa : ' . $nama . ' umurnya ' . $_umur . ' tahun ';
?>
 <br/>
<?php print($nama) ; ? >
```

PHP Data Type

Type	Description
`int`, `integer`	Whole numbers (i.e., numbers without a decimal point).
`float`, `double`, `real`	Real numbers (i.e., numbers containing a decimal point).
`string`	Text enclosed in either single (' ') or double (" ") quotes. [*Note:* Using double quotes allows PHP to recognize more escape sequences.]
`bool`, `boolean`	True or false.
`array`	Group of elements.
`object`	Group of associated data and methods.
`resource`	An external source—usually information from a database.
`NULL`	No value.

PHP Data Type

- PHP supports the following:
  - Boolean: $isActive = TRUE;
  - Integer:: $zip = 15213;
  - Float:: $myBalance = 567.89;
  - String:: $isActive = 'TRUE';
  - Arrays:: $myInfo = array("name" => "Bob","age"=>8);
  - Objects:: $myBook = new Book( );
  - Resources :: $a=fopen('tes.txt');
  - NULL :: null
  - Constants:: define("PHI", 3.14);

PHP Keyword

PHP keywords				
abstract	die	exit	interface	require
and	do	extends	isset	require_once
array	echo	__FILE__	__LINE__	return
as	else	file	line	static
break	elseif	final	list	switch
case	empty	for	__METHOD__	throw
catch	enddeclare	foreach	method	try
__CLASS__	endfor	__FUNCTION__	new	unset
class	endforeach	function	or	use
clone	endif	global	php_user_filter	var
const	endswitch	if	print	while
continue	endwhile	implements	private	xor
declare	eval	include	protected	
default	exception	include_once	public	

Fig. 23.5 | PHP keywords.

Tipe Variable

- Define By User (programmer)

  $_nama = ‘Naurah Husna’;

  $_ipk = 3.87;

- Define By System (PHP Global Variable)

  $_SERVER[‘DOCUMENT_ROOT’];

  $_SERVER[‘PHP_SELF’];

String

- Single-quoted strings
  - Ex.:: $quote = ‘I\’ll be back’;
  - Use \’ to specify a single quote character.
  - Does not expand newlines or variables.
- Double-quote strings
  - Does expand newlines & variables.
  - Use \ to escape characters.
  - *Example::* $q = “Arnold said, ‘$quote’\n”;
  - *Result::* Arnold said, ‘I’ll be back’ (followed by a linefeed.)
  - Double-quote Use in SQL Statement :
    $sql=“SELECT * FROM mahasiswa WHERE nim=‘011011’”;
- Use . to concatenate strings
  - Example $q = ‘Arnold said,’ . $quote;

PHP Basic - Array

- Indexed Array – array dengan index numeric
- Associative Array – array dengan menggunakan keys
- Multidimensi Array – array mengandung satu atau lebih array didalamnya

- Mendefinisikan array :
  - $_array = [element array] ;
  - $_array = array (element array) ;
- Panjang array : function count($arrayname)

PHP Basic - Indexed Array

- Indexed Array – array dengan index numeric

  $_fruits = [‘apple’,’orange’,’manggo’]; // index mulai dari 0

- Index array dapat di set secara manual

  $_fruits[0] = ‘apple’;

  $_fruits[1] = ‘orange’;

  $_fruits[2] = ‘manggo’;

- Hapus array

  unset($_fruits) ; // hapus seluruh elemen array

  unset($_fruits[1]); // hapus elemen array index ke-1

PHP Basic - Indexed Array

- Loop Indexed Array

```
$jml_data = count($_fruits);
for ($i = 0 ; $i < $jml_data ; $i++) {
   echo $_fruits[$i];
   echo '<br/>';
}
```

PHP Basic – Array Associative

- Two Ways creation

```
$umur = [ ‘ahmad’=>20, ‘ali’=>21, ‘dewi’=>19 ];

$umur[‘ahmad’] = 20;
$umur[‘ali’] = 21;
$umur[‘dewi’] = 19;
```

PHP Basic - Associative Array

- Loop Associative Array

```
foreach ($umur as $key => $val) {
    echo ‘ Key : ‘ . $key . ‘ , value ‘ . $val ;
    echo ‘<br/>’;
}
```

PHP Basic – Array Multidimensi

- $ar_jus = [
  ['buah'=>'Mangga','harga'=>8000],
  ['buah'=>'Alpukat','harga'=>10000],
  ['buah'=>'Durian','harga'=>14000],
  ] ;

  foreach($ar_jus as $jus){
  echo ' Jus ' . $jus['buah'] . ' harganya : ' . $jus['harga'] . '
';
  }

Superglobal Array

Variable name	Description
$_SERVER	Data about the currently running server.
$_ENV	Data about the client's environment.
$_GET	Data sent to the server by a `get` request.
$_POST	Data sent to the server by a `post` request.
$_COOKIE	Data contained in cookies on the client's computer.
$GLOBALS	Array containing all global variables.

```
<?php
    // info.php
    phpinfo();
?>
```

Superglobal Array – Form Processing

- Array Superglobal adalah array associative yang didefinisikan oleh PHP untuk menyimpan variable dari inputan user, variable lingkungan web server dan dapat diakses dalam skup variable apapun (global)
- Array $_GET dan $_POST mengembalikan informasi data yang dikirim oleh server menggunakan request HTTP get dan post
- Pada form gunakan nilai argument method = “POST” atau “GET”

PHP Basic – Form Processing

- Salah satu fungsi “original” PHP adalah memproses inputan dari FORM HTML
  - Simpan data ke database
  - Email inputan untuk ke customer
  - Validasi input (security)
- *Setiap elemen form dapat diproses oleh* PHP.
- PHP5 menyimpan informasi data dari form di $_GET, $_POST, and $_REQUEST yaitu variabel superglobal berbentuk array associative

Method Form = GET

- GET
  - Data Request ditampilkan pada URL
  - Digunakan untuk Pencarian data
  - Tidak mengubah data di sisi server
  - Di Cache oleh Client (Browser)
  - Tools : Form method GET, HyperlLink, Header Redirection
  - Query SQL : SELECT
  - Menangkap request : $_GET , $_REQUEST
  - Client : <input type=“text” name=“nama” size=“20”/>
  - Contoh : $nama_siswa = $_GET[‘nama’]

Method Form = POST

- POST
  - Data Request TIDAK ditampilkan pada URL
  - Digunakan untuk Manipulasi data
  - Dapat digunakan untuk mengubah data di sisi server
  - Tools : Form method POST
  - Query SQL : INSERT, UPDATE, DELETE
  - Menangkap request : $_POST , $_REQUEST
  - Client : <input type=“text” name=“nama” size=“20”/>
  - Contoh : $nama_siswa = $_POST[‘nama’]

Contoh : Form GET

```
<form name="f1" method="get" action="thanks.php">
   <input type="hidden" name="fType" value="GET" />
   <input type="text" name="fullName" /><br />
   <select name="program" size="1">
      <option>Undergraduate</option>
      <option>Masters</option>
      <option>Doctoral</option>
   </select>
   <input type="submit" value="Submit" />
</form>
```

REQUEST : GET

- URL: http://*some_domain*/thanks.php?fullName=John&program=Masters

```
<?php
    $formType = $_GET['fType'];
    $fullName  = $_GET['fullName'];
    $program   = $_GET['program'];

    echo ('<p>');
    echo ('Hello,'.$fullName. "<br>\n");
    echo ('You are in the '.$program.' program."<br>\n");
    echo ('Form method was '.$formType. "</p>\n");
?>
```

Contoh : Form POST

```
<form name="f2" method="post" action="thanks.php">
   <input type="hidden" name="toEmail" value="xyz@pitt.edu" />
   <input type="text" name="firstName" /><br />
   <input type="text" name="lastName" /><br />
   <textarea name="comment"></textarea>
   <input type="submit" value="Submit" />
</form>
```

REQUEST : POST

- URL: http://*some_domain*/thanks.php

```
<?php
    $to = $_POST[‘toEmail’];
    $firstName = $_POST[‘firstName’];
    $lastName = $_POST[‘lastName’];
    $comment = addslashes($_POST[‘comment’]);

    echo (‘<p>’);
    echo (‘Thank you,’.$firstName. ‘ for your feedback\.’);
    echo ( “</p>\n”);
?>
```

PHP Basic Operator

- PHP has most of the operators one would find in other scripting languages.
- These include
  - Arithmetic:: + - * / %
  - Assignment:: +=, etc
  - Increment/decrement:: ++ --
  - Comparison:: <, >, >=, <=, ==, != , < >, ===
  - Logical:: &&, ||
  - String:: ., .=
  - Type:: instanceof *someClass*;

PHP Control Structures

- if (*expression*) {*block*} elseif {*block*} else {*block*}
- while (*expression*) {*block*} OR
- do {*block*} while (*expression*)
- for (expression) {block}
- foreach var (list) {block}
- switch (*variable*) //variable can be int or string

```
case 0:
    // do something;
    break;
case 1:
    // do something else;
    break;
```

- break;
- continue;
- return;