

**A T I B**

## Dynamic Programming – PHP Basic

Robertus Setiawan Aji Nugroho  
References: [w3schools.com](http://w3schools.com)

# Dynamic Programming

- HTML is static
  - To change the content, user needs to edit the html file
  - But its good for presentation/visualization
- Dynamic Programming
  - User is able to modify the content without editing the file
  - HTML content changed dynamically
  - Database driven

# Dynamic Programming

- Dynamic Programming
  - Must be in the server
  - Example: Blog, facebook, etc.
  - User don't need to edit the html file, but manipulating the content via forms given by the system
  - It's not in the client computer, but in the server. We just need to write the URL in the address bar and click enter

# Web Server

- Web Server
  - A server that provides service for the web
  - Internet web server needs to be:
    - Connected to the internet
    - Accessed by client (have a global IP)
    - Easy to remember: Domain name (www.unika.ac.id, www.facebook.com, etc)
    - For learning purposes, we will use our local computer as web server

# Web Server

- XAMPP
  - Complete web server
  - Easy to install
  - Supports PHP and MySQL
  - Multi Operating System
  - Free
  - [www.xampp.org](http://www.xampp.org)

# Web Server

- XAMPP
  - XAMPP/ LAMPP has been installed in lab computer
  - To install it to your own computer, follow the tutorial at [www.xampp.org](http://www.xampp.org)

# PHP

- PHP
  - Server side programming
  - Only works on a computer with web server
  - File extension: “.php”
  - Can be mixed with HTML, but the extension must be “.php” not “.html”
  - HTML handle visualitation, PHP handle data
  - Put all documents at:
    - For Linux: /opt/lampp/htdocs/**your folder**
    - For windows: C://xampp/htdocs/**yourfolder**

# PHP

- Write this code, save it with “.php” extension on your folder at htdocs!

```
<html>  
  <head>  
    <title>My first PHP</title>  
  </head>  
  <body>  
    <?php  
      echo “Hello World”;  
    ?>  
  </body>  
</html>
```



- Variable
  - A container for data
  - Can be container for number, character, string, boolean, etc
  - But it must be consistent!
  - Once it is used as number, it must be for number all the time.
  - Format:
    - Started with “\$” sign and followed by variable name
    - Example: \$myVariable

# PHP

- Common rules:
  - PHP Code is started/nested by:
    - **<?php**
    - And closed by:
    - **?>**
  - Every statement must be ended with semicolon (“;”)
    - **echo “hello world”;**
  - One line one statement
  - Not necessarily, but it will help you much
  - **Case sensitive**

- Variable
  - A container for data
  - Can be container for number, character, string, Boolean, etc
  - But it must be consistent!
  - Once it is used as number, it must be for number all the time.
  - Once you declare it, you can use it inside the scope (page, function ,etc)

# PHP Variable

```
<html>
  <head>
    <title>My first PHP</title>
  </head>
  <body>
    <?php
      $myVar1 = 15;
      $myVar2 = 10;
      $myVar3 = $myVar1 * $myVar2;
      echo "Multiplication of $myVar1 and $myVar2 is $myVar3";
    ?>
    <br />
    <br />
    <?php
      $myVar4 = "This is a container for String";
      echo $myVar4;
    ?>
  </body>
</html>
```

# PHP Operator

- Php Operator:

Operator	Description	Example(s)
*	Multiplication	<code>\$c = \$a * \$b;</code>
+	Addition	<code>\$c = \$a + \$b;</code>
-	Substraction	<code>\$c = (\$a * \$b) - \$d;</code>
/	Division	<code>\$c = \$a / \$b;</code>
%	Modulo	<code>\$c = \$a % \$b;</code>
.	Concatenation (concatenate one string with other)	<code>\$c = \$a.\$b;</code>