Tugas 2 Praktikum Pemrograman Web – Intro to PHP

By Dimas Rizky H.P. – 2110141011 – 3 D4 IT A

PHP Intro

PHP (PHP : Hypertext Preprocessor) merupakan bahasa *server script* yang digunakan untuk membuat halaman web yang dinamis dan interaktif. PHP bersifat gratis dan merupakan alternatif dari Microsoft ASP.

Script PHP dieksekusi di server, sehingga source code tidak akan ditampilkan pada source HTML, namun hasilnya akan dikirimkan dalam bentuk *plain-HTML*. PHP dapat berisi text, syntax HTML, CSS Javascript dan PHP code.

Kemampuan PHP antara lain :

* PHP dapat digunakan untuk membuat konten dinamis
* PHP dapat digunakan untuk *create, open, read, write, delete* dan *close* file yang ada pada server
* PHP dapat digunakan untuk mengambil data
* PHP dapat mengirim dan menerima *cookies*
* PHP dapat menambah, menghapus maupun mengubah data pada database
* PHP dapat digunakan untuk mengontrol *user-access*
* PHP dapat digunakan untuk mengenkripsi data.

Kelebihan PHP dibandingkan basa pemrograman lainnya :

* PHP bersifat multiplatform sehingga dapat dijalankan di banyak platform (Windoes, Linux, Unix, Mac OS, dll)
* PHP kompatibel dengan hampir semua server yang digunakan saat ini (Apache, IIS, dll)
* PHP mendukung sebagian besar sistem database yang ada
* PHP gratis
* PHP mudah dipelajari dan berjalan efisien pada sisi server

Untuk menggunakan PHP, user harus menggunakan web hosting yang sudah support dengan PHP dan MySQL. Untuk menjalankannya secara local, user perlu menginstall web server, menginstall PHP, dan menginstall database contohnya adalah MySQL.

PHP Syntax

Syntax PHP dapat ditempatkan dimanapun pada dokumen .php, selama masih ada dalam tag “<?php” dan “?>”. Syntax PHP harus diakhiri dengan semicolon (;)

*Contoh skrip PHP :*

OUTPUT :

<div class="card example1">

<div class="card-header card-inverse card-primary mb-3">

Basic PHP Syntax

</div>

<div class="card-block">

<blockquote class="card-blockquote">

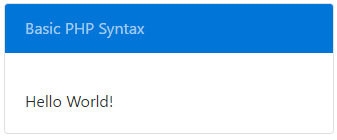
<?php **echo** "Hello World!" ?>

</blockquote>

</div>

</div>

<br>



* 1. **Komentar di PHP**

Pada PHP komentar merupakan baris kode yang tidak akan dibaca/dieksekusi oleh program yang ada di server. Tujuan adanya komentar ini hanyalah untuk memudahkan programmer untuk meninggalkan pesan yang dapat dibaca dengan mudah tanpa mengganggu jalannya skrip.

Ada beberapa cara untuk menambahkan comment pada PHP, yaitu dengan menambahkan character “//” atau “#” pada baris kode yang ingin dijadikan komentar, atau dengan cara menempatkan kode yang ingin dikomen di dalam blok “/\* ... \*/”.

*Contoh Comment di PHP :*

OUTPUT :

<div class="card comments">

<div class="card-header card-inverse card-primary mb-3">

Comments in PHP

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

//Ini comment baris tunggal

#Ini juga merupakan comment baris tunggal

/\*

Ini merupakan comment banyak baris

Berupa code block

Yang mengikuti lebar baris kode

\*/

//Comment juga dapat digunakan pada bagian spesifik dari baris kode

**$x** **=** 5 /\*+ 15 \*/ **+** 5;

**echo** **$x**;

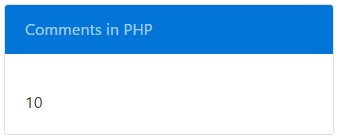
?>

</blockquote>

</div>

</div>

<br>



* 1. **PHP Case Sensitivity**

Pada PHP, semua keyword (if, else, while, echo, dll), class, fungsi bawaan, maupun fungsi buatan user tidak bersifat case-sensitive, artinya dapat dituliskan dalam kode dalam kondisi huruf besar atau kecil.

*Contoh sifat case-insensitive PHP :*

OUTPUT :

<div class="card casesensitive1">

<div class="card-header card-inverse card-primary mb-3">

PHP Case Sensitivity 1

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**ECHO** "Hello World!<br>";

**echo** "Hello World!<br>";

**Echo** "Hello World!<br>";

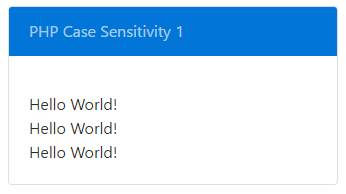
?>

</blockquote>

</div>

</div>

<br>

**

Namun, meskipun bersifat case-insensitive, variabel PHP bersifat case-sensitive. Artinya, variabel $color dan $COLOR merupakan dua variabel yang berbeda.

*Contoh sifat case-sensitive pada PHP :*

<div class="card casesensitive2">

<div class="card-header card-inverse card-primary mb-3">

PHP Case Sensitive 2

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$color** **=** "red";

**echo** "My car is " **.** **$color** **.** "<br>";

**echo** "My house is " **.** **$COLOR** **.** "<br>";

**echo** "My Boat is " **.** **$coLOR** **.** "<br>";

?>

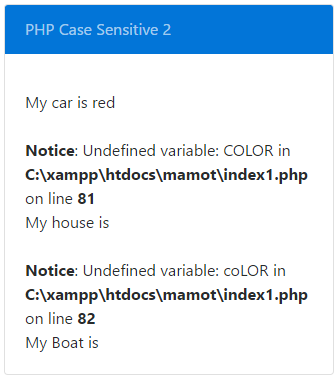
</blockquote>

</div>

</div>

<br>

OUTPUT :

**

PHP Variables

Variabel di PHP merupakan tempat untuk menyimpan suatu value atau suatu data. Variabel dapat memiliki nama yang pendek (seperti x ataupun y) atau nama yang lebih panjang (seperti umur, nama, alamat, total\_volume).

Terdapat beberapa aturan dalam penulisan variabel di PHP. Yaitu,

* Sebuah variable harus dimulai dengan tanda dollar (‘$’), diikuti dengan nama variabelnya.
* Sebuah nama variabel harus dimulai dengan huruf ataupun karakter underscore (‘\_’)
* Nama variabel tidak dapat diawali dengan nomor
* Nama variabel hanya boleh terdiri dari karakter alpha-numeric dan underscores (A-z, 0-9, dan \_)
* Nama variabel bersifat case-sensitive

PHP merupakan bahasa skrip yang ‘luwes’, artinya, programmer tidak perlu mendefinisikan tipe data terlebih dahulu pada saat mendeklarasikan sebuah variabel. PHP secara otomatis mengkonversi variabel-variabel yang telah dibuat ke tipe data bergantung dengan value nya. Jika dibandingkan dengan bahasa yang lain (C++, Java, C, dll) tipe data harus didefinisikan terlebih dahulu sebelum mendeklarasikan variabel.

*Contoh deklarasi variabel PHP :*

OUTPUT :

<div class="card declarevariable">

<div class="card-header card-inverse card-success mb-3">

Creating (Declaring) PHP Variables

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$txt** **=** "Hello World!";

**$x** **=** 5;

**$y** **=** 10.5;

**echo** "Nilai variabel \$x : "**.$x.**"<br>";

**echo** "Nilai variabel \$txt : "**.$txt.**"<br>";

**echo** "Nilai variabel \$y : "**.$y.**"<br>";

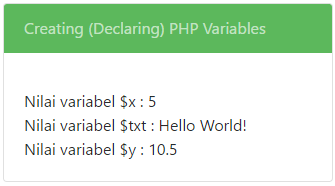
?>

</blockquote>

</div>

</div>

<br>

**

**2.1 Output Variables**

Untuk menampilkan variabel ke layar, statement “echo” pada PHP seringkali digunakan. Echo akan menampilkan value dari variabel (bergantung pada tipe datanya) ke layar dengan datanya dikirimkan dalam bentuk HTML.

*Contoh penggunaan echo untuk melakukan output text dan variabel :*

OUTPUT :

<div class="card outputvariable1">

<div class="card-header card-inverse card-success mb-3">

Output Variables 1

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$txtw** **=** "W3Schools.com";

**echo** "Echo Model 1 : "**.**"I love **$txtw**!"**.**"<br>";

**echo** "Echo model 2 : "**.**"I Love "**.$txtw.**"!"**.**"<br>";

**echo** "<bold>Kedua baris kata di atas akan menghasilkan output yang sama meskipun penggunaan codingnya berbeda</bold><br>"

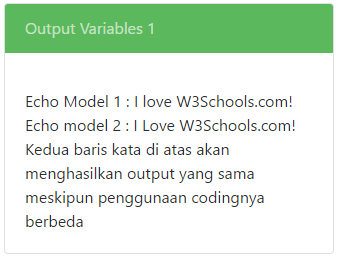
?>

</blockquote>

</div>

</div>

<br>

**

Dengan statement echo, programmer juga dapat mencetak hasil dari sebuah operasi, misalkan penambahan antara 2 variabel dengan menuliskan expresinya menggunakan statement “echo”.

*Contoh penggunaan echo untuk output hasil penjumlahan antara 2 variabel :*

OUTPUT :

<div class="card outputvariable2">

<div class="card-header card-inverse card-success mb-3">

Output Variables 2

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** 5;

**$y** **=** 4;

**echo** **$x** **+** **$y**;

?>

</blockquote>

</div>

</div>

<br>

**

**2.2 PHP Variables Scope**

Dalam PHP, variabel dapat dideklarasikan di bagian script manapun (di dalam fungsi, di luar fungsi, dibagian atas tag body ataupun dimanapun. Yang membedakan antara penempatan itu adalah scope dari variabel tersebut. Scope maksudnya adalah, sampai seberapa atau dimana saja variabel tersebut dapat dipanggil/digunakan.

PHP memiliki tiga variabel scope, yaitu local, global dan static. Yang dimaksud variabel dengan GLOBAL SCOPE adalah variabel yang dideklarasikan **diluar** fungsi, dan hanya dapat diakses di luar fungsi sehingga jika variabel tersebut diakses di dalam suatu fungsi, PHP akan mengembalikan error jika variabel tersebut digunakan di dalam fungsi namun belum dideklarasikan secara global.

*Contoh penggunaan global variabel di dalam suatu fungsi :*

OUTPUT :

<div class="card globallocalscope1">

<div class="card-header card-inverse card-success mb-3">

Global and Local Scope1

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** 5; //Ini merupakan variabel global scope

function myTest(){

//Menggunakan variabel x di dalam fungsi ini akan menghasilkan error

**echo** "<p>Variabel X di dalam fungsi ini bernilai : **$x** </p>";

}

myTest();

**echo** "<p>Variabel x di luar fungsi myTest() adalah : **$x** </p>";

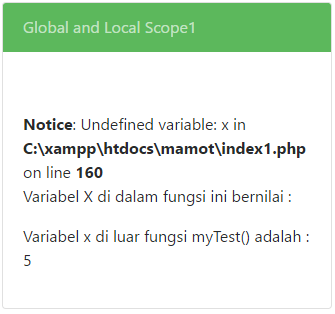
?>

</blockquote>

</div>

</div>

<br>

**

Sedangkan, variabel yang dideklarasikan di dalam fungsi memiliki scope local. Artinya, variabel itu hanya dapat dipanggil/diakses di dalam fungsi tersebut saja. Jika dipanggil di luar fungsi itu, variabel tersebut tidak dapat dipanggil/digunakan.

*Contoh variabel dengan local scope pada fungsi PHP :*

OUTPUT :

<div class="card globallocalscope2">

<div class="card-header card-inverse card-success mb-3">

Global and Local Scope 2

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

function myTest2(){

**$x** **=** 5; //Variabel dengan scope local

**echo** "<p>Variabel x didalam fungsi ini bernilai : **$x** </p>";

}

myTest2();

//Menggunakan variabel x di luar fungsi ini akan menghasilkan error

**echo** "<p>Variabel x diluar fungsi myTest2() bernilai : **$x** </p>";

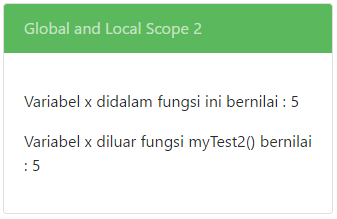
?>

</blockquote>

</div>

</div>

<br>

**

**2.3 PHP Global Keyword**

Keyword global digunakan untuk mengakses global variabel dari dalam fungsi. Untuk menerapkan keyword global bisa dengan menambahkan keyword ‘global’ sebelum nama variabel yang ada di dalam fungsi.

*Contoh penggunaan global keyword di dalam fungsi PHP :*

OUTPUT :

<div class="card globalkey1">

<div class="card-header card-inverse card-success mb-3">

Global Keyword 1

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** 5;

**$y** **=** 10;

function myTest3(){

global **$x**, **$y**;

**$y** **=** **$x** **+** **$y**;

}

myTest3();

**echo** **$y**; //Output nya harusnya bernilai 15

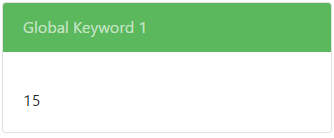
?>

</blockquote>

</div>

</div>

<br>

**

Selain itu, PHP juga menyimpan semua global variabel dalam array yang bernama $GLOBALS[index]. Setiap index-nya menyimpan nama dari variabelnya. Array ini juga dapat diakses secara langsung dari dalam fungsi dan dapat digunakan untuk mengupdate nilai dari global variabel secara langsung.

*Contoh akses global variabel menggunakan array $GLOBALS :*

OUTPUT :

<div class="card globalkey2">

<div class="card-header card-inverse card-success mb-3">

Global keyword 2 - Indexed

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** 5;

**$y** **=** 10;

function myTest4(){

**$GLOBALS**['y'] **=** **$GLOBALS**['x'] **+** **$GLOBALS**['y'];

}

myTest4();

**echo** **$y**;

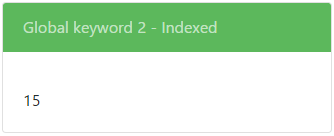
?>

</blockquote>

</div>

</div>

<br>

**

**2.4 PHP Static Keyword**

Ketika sebuah fungsi selesai dieksekusi, semua variabel yang dideklarasikan di dalam fungsi tersebut akan dihapus. Namun, kadangkala programmer membutuhkan variabel lokal itu dengan nilai yang sama setelah fungsi selesai dieksekusi. Hal ini dapat dilakukan dengan menggunakan static keyword ketika mendeklarasikan variabel di dalam fungsi. Sehingga, setiap kali fungsi tersebut dipanggil, variabel static tersebut masih menyimpan informasi yang tersimpan sebelum fungsi tersebut dipanggil. Namun tetap, variabel static masih bersifat local variabel untuk fungsi tersebut.

*Contoh pendeklarasian static variabel di dalam fungsi PHP :*

OUTPUT :

<div class="card statickey">

<div class="card-header card-inverse card-success mb-3">

PHP Static Keyword

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

function myTest5(){

static **$x** **=** 0;

**echo** **$x**;

**$x++**;

}

myTest5();

myTest5();

myTest5();

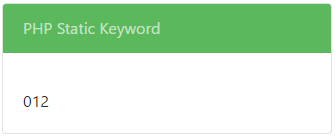
?>

</blockquote>

</div>

</div>

<br>



PHP Echo/Print

Pada PHP terdapat dua cara mendasar untuk menampilkan output, yaitu dengan statement echo, ataupun print. Statement echo dan print kuranglebih sama fungsinya. Keduanya digunakan untuk menampilkan output ke layar. Perbedaannya sangat kecil, yaitu echo tidak mempunyai return value, sedangkan print memiliki return value 1, sehingga dapat digunakan dalam expression. Echo dapat menangani lebih dari satu argumen dan dalam perihal kecepatan, echo lebih cepat dibandingkan dengan print.

**3.1 PHP Echo Statement**

Echo statement dapat digunakan dengan atau tanpa tanda kurung (echo atau echo()). Statement echo dapat digunakan untuk mencetak output berupa text maupun variabel dan operasi variabel. Pada pencetakan text, text yang di cetak dapat memiliki unsur tag HTML didalamnya yang nantinya pada saat ditampilkan akan berlaku sebagai tag HTML seperti biasa.

*Contoh penggunaan echo untuk mencetak text :*

OUTPUT :

<div class="card echostatement">

<div class="card-header card-inverse card-warning mb-3">

PHP Echo Statement

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**echo** "<h2>PHP is Fun!</h2>";

**echo** "Hello World!<br>";

**echo** "I'm about to learn PHP!<br>";

**echo** "This ", "string ", "was ", "made ", "with multiple parameters.";

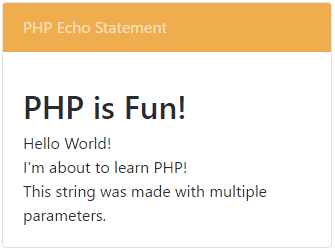
?>

</blockquote>

</div>

</div>

<br>

**

Selain menampilkan text variabel, statement echo juga untuk menampilkan variabel ataupun operasi variabel.

*Contoh penggunaan echo untuk mencetak variabel :*

OUTPUT :

<div class="card echostatement2">

<div class="card-header card-inverse card-warning mb-3">

PHP Echo Statement 2

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$txt1** **=** "Learn PHP";

**$txt2** **=** "W3Schools.com";

**$x** **=** 5;

**$y** **=** 4;

**echo** "<h2>**$txt1**</h2>";

**echo** "Study PHP at **$txt2**<br>";

**echo** **$x** **+** **$y**;

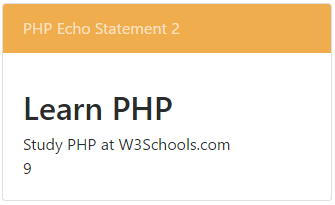
?>

</blockquote>

</div>

</div>

<br>

**

**3.2 PHP Print Statement** Sama seperti statement echo, statement print dapat digunakan dengan atau tanpa tanda kurung. Statement print juga dapat digunakan untuk menampilkan output hasil operasi variabel atau nilai variabel disamping mencetak text saja.

*Contoh penggunaan print untuk mencetak text :*

<div class="card printstatement">

<div class="card-header card-inverse card-warning mb-3">

PHP Print Statement

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**print** "<h2>PHP is Fun!</h2>";

**print** "Hello World!<br>";

**print** "I'm about to learn PHP!";

?>

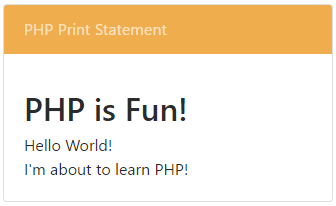
</blockquote>

</div>

</div>

<br>

OUTPUT :

**

Statement print juga dapat digunakan untuk mencetak hasil operasi variabel dan mencetak nilai variabel.

*Contoh penggunaan print untuk mencetak hasil operasi atau nilai dari variabel :*

OUTPUT :

<div class="card printstatement2">

<div class="card-header card-inverse card-warning mb-3">

PHP Print Statement 2

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$txt1** **=** "Learn PHP";

**$txt2** **=** "W3Schools.com";

**$x** **=** 5;

**$y** **=** 4;

**print** "<h2>**$txt1**</h2>";

**print** "Study PHP at **$txt2**<br>";

**print** **$x** **+** **$y**;

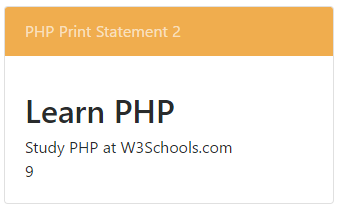
?>

</blockquote>

</div>

</div>

<br>

**

PHP Data Types

Variabel digunakan untuk menyimpan sebuah data dengan tipe yang bermacam macam, dan setiap data tersebut dapat melakukan hal-hal yang berbeda. PHP mendukung tipe data **string, integer, float, boolean, array, object, NULL, dan resource**.

**4.1 PHP String**

String merupakan rangkaian dari karakter, dan dapat direpresentasikan atau dideklarasikan dengan mengapit kata/kalimat/karakter dalam tanda petik satu (‘) atau petik dua (“). Pada PHP data bertipe string biasa digunakan untuk melakukan manipulasi text, mencetak text, dan melakukan manajemen data yang memiliki text.

*Contoh variabel dengan tipe data string :*

OUTPUT :

<div class="card datatypestring">

<div class="card-header card-inverse card-primary mb-3">

PHP String

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** "Hello world!";

**$y** **=** 'Hello world!';

**echo** **$x**;

**echo** "<br>";

**echo** **$y**;

**echo** "<br> Tipe Data variabel \$x : "; **var\_dump**(**$x**);

**echo** "<br> Tipe Data variabel \$y : "; **var\_dump**(**$y**);

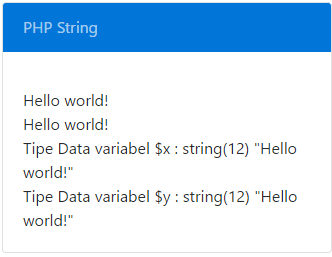
?>

</blockquote>

</div>

</div>

<br>

**

**4.2 PHP Integer**

Integer merupakan tipe data bilangan bulat antara -2,147,483,648 dan 2,147,483,637. Adapun beberapa aturan integer, integer harus memiliki sekurangkurangnya satu digit. Integer tidak boleh memiliki nilai dibelakang koma. Integer bisa bernilai positif atau negatif. Integer dapat dispesifikan dalam tiga bentuk, yaitu dalam decimal, hexadecimal, ataupun octa.

Pada PHP terdapat fungsi untuk menampilkan jenis tipe data apa yang dimiliki oleh suatu variabel. Fungsi itu bernama var\_dump. Fungsi var\_dump mengembalikan tipe data dan nilai dari variabelnya.

*Contoh penggunaan tipe data integer pada PHP :*

OUTPUT :

<div class="card datatypeinteger">

<div class="card-header card-inverse card-primary mb-3">

PHP Integer

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** 5958;

**echo** "<br> Tipe Data variabel **$x** : "; **var\_dump**(**$x**);

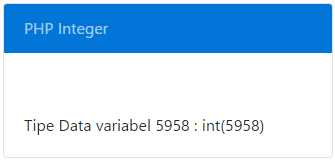
?>

</blockquote>

</div>

</div>

<br>

**

**4.3 PHP Float**

Float merupakan angka yang memiliki nilai decimal atau angka dibelakang koma.

*Contoh penggunaan variabel dengan tipe data float :*

OUTPUT :

<div class="card datatypefloat">

<div class="card-header card-inverse card-primary mb-3">

PHP Float

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** 10.365;

**echo** "<br> Tipe Data variabel **$x** : "; **var\_dump**(**$x**);

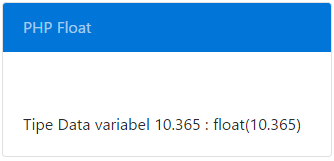
?>

</blockquote>

</div>

</div>

<br>

**

**4.4 PHP Boolean**

Boolean merupakan representasi dua state atu kondisi, yaitu TRUE atau FALSE. Tipe data boolean seringkali digunakan dalam conditional testing.

*Contoh penggunaan variebel dengan tipe data boolean :*

OUTPUT :

<div class="card datatypeboolean">

<div class="card-header card-inverse card-primary mb-3">

PHP Boolean

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** true;

**echo** "Tipe Data variabel **$x** : "; **var\_dump**(**$x**);

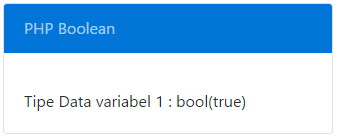
?>

</blockquote>

</div>

</div>

<br>

**

**4.5 PHP Array**

Tipe data array dapat menyimpan lebih dari satu variabel. Variabel yang berada di dalam array dapat diakses berdasarkan index masing-masing variabelnya. Index dalam array dimulai dari indeks ke 0, dan untuk mengakses indexks selanjutnya bisa dilakukan inkremen atau langsung menspesifikasikan indeks yang diinginkan.

*Contoh penggunaan varibel bertipe data array :*

OUTPUT :

<div class="card datatypearray">

<div class="card-header card-inverse card-primary mb-3">

PHP Array

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$cars=** **array**("Volvo","BMW","Toyota");

**echo** "Tipe Data variabel **$x** : "; **var\_dump**(**$cars**);

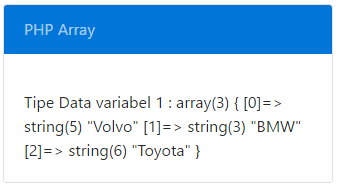
?>

</blockquote>

</div>

</div>

<br>

**

**4.6 PHP Object**

Object merupakan tipe data yang dapat menyimpan data dan informasi mengenai cara memproses data tersebut. Pada PHP, sebuah object harus dideklarasikan secara eksplisit. Object dideklarasikan mulai dari nama class nya, lalu diikuti dengan variabel dan function yang ada di dalamnya.

*Contoh penggunaan object pada PHP :*

OUTPUT :

<div class="card datatypearray">

<div class="card-header card-inverse card-primary mb-3">

PHP Array

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$cars=** **array**("Volvo","BMW","Toyota");

**echo** "Tipe Data variabel **$x** : "; **var\_dump**(**$cars**);

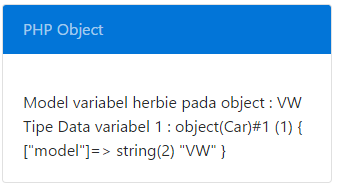
?>

</blockquote>

</div>

</div>

<br>

**

**4.7 PHP NULL Value**

Null merupakan tipe data spesial yang mana hanya memiliki satu nilai yang memungkinkan, yaitu NULL. Variabel bertipe data null merupakan variabel yang tidak mempunyai value. Jadi jika variabel dideklarasikan tanpa melakukan assignment terhadap variabel tersebut, variabel itu akan dianggap null variabel.

*Contoh penggunaan variabel bertipe data null :*

<div class="card datatypeNULL">

<div class="card-header card-inverse card-primary mb-3">

PHP NULL

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** "Hello world!";

**$x** **=** null;

**echo** "<br> Tipe Data variabel **$x** : "**.var\_dump**(**$x**);

?>

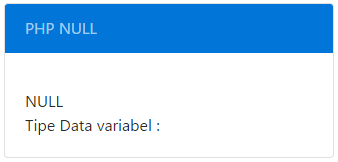
</blockquote>

</div>

</div>

<br>

OUTPUT :



**4.8 PHP Resource**

Tipe data resource merupakan tipe data spesial yang sebenarnya bukan tipe data. Tipe ini merupakan penampung dari referensi fungsi dan resource external milik sistem PHP. Contoh umum pada penggunaan tipe data resource adalah pada saat pemanggilan database.

PHP Strings

Seperti yang sudah dijelaskan sebelumnya, string merupakan tipe data yang terdiri dari kumpulan karakter karatker yang digabungkan. String pada PHP memiliki banyak fungsi khusus yang dibuat khusus untuk memodifikasi maupun memanipulasi variabel string.

**5.1 Panjang String**

PHP menyediakan fungsi untuk menghitung berapa banyak karakter yang ada pada sebuah teks string. Fungsi ini dinamai strlen() yang akan mengembalikan panjang string jika diberikan parameter sebuah data string.

*Contoh penggunaan fungsi strlen() :*

OUTPUT :

<div class="card stringfunction1">

<div class="card-header card-inverse card-success mb-3">

PHP String Character Length

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**echo** **strlen**("Hello world!"); //Output nya adalah 12

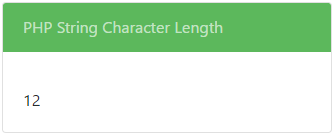
?>

</blockquote>

</div>

</div>

<br>

**

**5.2 Jumlah Kata Pada String**

Fungsi lainnya untuk memanipulasi sebuah variabel string adalah fungsi str\_word\_count(). Fungsi ini merupakan fungsi bawaan dari PHP yang berfungsi untuk mengembalikan jumlah kata dari data teks yang dijadikan sebagai parameter.

*Contoh penggunaan function str\_word\_count() :*

OUTPUT :

<div class="card stringfunction2">

<div class="card-header card-inverse card-success mb-3">

PHP String Word Length

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**echo** **str\_word\_count**("Hello world!"); //Output nya adalah 2

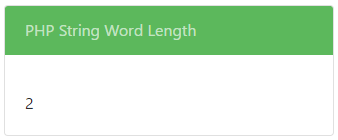
?>

</blockquote>

</div>

</div>

<br>

**

**5.3 Membalik String**

PHP menyediakan fungsi untuk membalik susunan huruf dari data bertipe string. Fungsi ini dinamakan strrev(). Strrev akan mengembalikan string bernilai kebalikan dali string yang digunakan sebagai parameter.

*Contoh penggunaan fungsi strrev() pada sebuah string :*

OUTPUT :

<div class="card stringfunction3">

<div class="card-header card-inverse card-success mb-3">

PHP String Reverse

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**echo** **strrev**("Hello world!"); //Output nya adalah !dlrow olleH

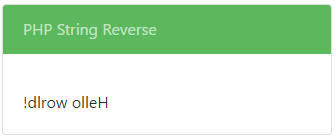
?>

</blockquote>

</div>

</div>

<br>

**

**5.4 Mencari Text Spesifik di Dalam String**

Fungsi strpos() milik PHP digunakan untuk mencari text di dalam string. Jika ada kata yang sama ditemukan, fungsi ini akan mengembalikan posisi karakter pertama pada kata yang sama. Jika tidak ada yang sam maka fungsi akan mengembalikan false. Posisi yang dimaksud adalah indeks dari karakter pada sebuah string. Indeksnya dimulai dari 0.

*Contoh penggunaan fungsi strpos() dalam string di PHP :*

OUTPUT :

<div class="card stringfunction4">

<div class="card-header card-inverse card-success mb-3">

PHP String Search Text

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**echo** **strpos**("Hello world!", "world"); //Output nya adalah 6

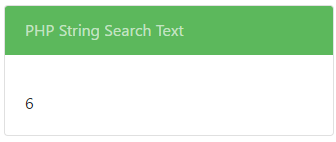
?>

</blockquote>

</div>

</div>

<br>

**

**5.5 Melakukan Replace Text di Dalam String**

Fungsi PHP str\_replace() melakukan replace beberapa karakter dengan karakter lain dalam data string. Fungsi ini menerima tiga parameter paling tidak (str\_replace(txt\_yang\_akan\_direplace, txt\_pengganti, string\_yang\_digunakan)).

*Contoh penggunaan fungsi str\_replace() :*

OUTPUT :

<div class="card stringfunction5">

<div class="card-header card-inverse card-success mb-3">

PHP String Replace

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**echo** **str\_replace**("world","Dolly","Hello world!"); //Output nya adalah Hello Dolly!

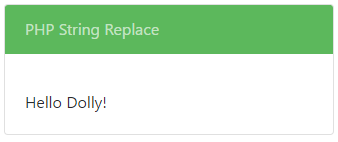
?>

</blockquote>

</div>

</div>

<br>

**

PHP Constants

Constant mirip seperti variabel, namun sekali saja constant didefinisikan, constant tidak dapat diganti atau dihapus.

Constant merupakan sebutan untuk simple value. Value yang tidak dapat diganti nilainya selama skrip berjalan. Constant yang valid dimulai dengan huruf atau tanda underscore. Tidak seperti variabel, constant sudah bersifat global disemua skrip yang berjalan.

**6.1 Membuat Constant PHP**

Untuk membuat constant bisa menggunakan fungsi define(). Fungsi define memiliki beberapa parameter, yaitu define(name, value, case-insensitive). Parameternya adalah sebagai berikut :

* name : merupakan nama dari constant,
* value : merupakan nilai dari constant,
* case insensitive : merupakan nilai boolean yang menentukan antara pemanggilan dan penggunaan nama konstant harus menggunakan kaidah case-sensitive atau tidak

*Contoh penggunaan constant dengan nama case-sensitive :*

<div class="card createconstant1">

<div class="card-header card-inverse card-danger mb-3">

PHP Create Constant

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**define**("GREETING", "Welcome to W3Schools.com!");

**echo** GREETING;

?>

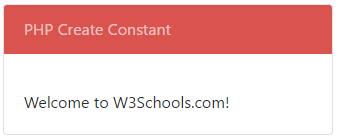
</blockquote>

</div>

</div>

<br>

OUTPUT :

**

Penggunaan constant atau pemanggilan variabel constant juga dapat di set menjadi case-insensitive, yang berarti jika ada constant bernama GREETING, constant tersebut dapat juga diakses menggunakan keyword Greeting.

*Contoh penggunaan constant dengan nama case-insensitive :*

OUTPUT :

<div class="card createconstant2">

<div class="card-header card-inverse card-danger mb-3">

PHP Create Constant 2

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**define**("GREETING", "Welcome to W3Schools.com!", true);

**echo** greeting;

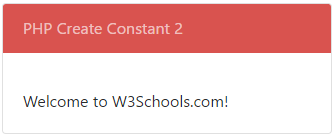
?>

</blockquote>

</div>

</div>

<br>

**

**6.2 Constant Bersifat Global**

Constant secara otomatis bersifat global dan dapat digunakan dimanapun di skrip. Constant dapat digunakan di dalam fungsi tanpa harus menggunakan global keyword

*Contoh penggunaan constant di dalam fungsi :*

<div class="card createconstant3">

<div class="card-header card-inverse card-danger mb-3">

PHP Constant Scope

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**define**("GREETING2", "Welcome to W3Schools.com!");

function myTest6(){

**echo** GREETING2;

}

myTest6();

?>

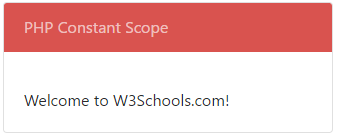
</blockquote>

</div>

</div>

<br>

OUTPUT :

**

PHP Operators

Operator digunakan untuk melakukan beberapa operation pada variabel dan valuenya. PHP membagi beberapa operator dalam grup, yaitu

* Arithmetic operators, merupakan operator untuk melakukan operasi aritmtika, penambahan, pengurangan, perkalian, pembagian, dll
* Assignment operators, merupakan operator untuk melakukan assign nilai ke variabel
* Compariosn operators, merupakan operator untuk melakukan komparasi / pembandingkan antara dua variabel atau value. Return valuenya berupa data boolean
* Increment/Decrement operators, operator ini digunakan untuk melakukan operasi inkremen atau dekremen pada sebuah variabel atau value
* Logical operators, operator ini digunakan untuk melakukan operasi logika seperti and, or, xor, not ataupun equal
* String operators, operator ini digunakan untuk melakukan operasi pada tipe data khusus string
* Array operators, khusus digunakan untuk tipe data array, yang digunakan untuk melakukan berbagai operasi untuk array\

**7.1 PHP Arithmetic Operators**

PHP arithmetic operators digunakan untuk melakukan operasi aritmatika pada data numerik. Operasi yang dapat dilakukan antara lain penambahan, pengurangan, perkalian, dll.

*Contoh penggunaan arithmetic operators :*

OUTPUT :

<div class="card arithmaticsOP">

<div class="card-header card-inverse card-warning mb-3">

PHP Arithmetic Operators

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** 6;

**$y** **=** 3;

**echo** "Addition : "**.**(**$x** **+** **$y**)**.**"<br>";

**echo** "Subtraction : "**.**(**$x** **-** **$y**)**.**"<br>";

**echo** "Multiplication : "**.**(**$x** **\*** **$y**)**.**"<br>";

**echo** "Division : "**.**(**$x** **/** **$y**)**.**"<br>";

**echo** "Modulus : "**.**(**$x** **%** **$y**)**.**"<br>";

**echo** "Exponentiation : "**.**(**$x** **\*\*** **$y**)**.**"<br>";

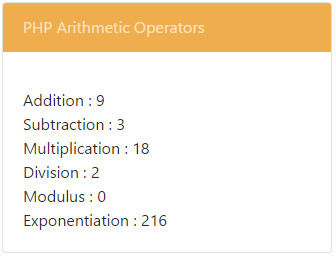
?>

</blockquote>

</div>

</div>

<br>

**

**7.2 PHP Assignment Operators**

Assignment operator digunakan untuk menuliskan atau melakukan assign nilai numerik ke sebuah variabel. Umumnya, penggunaan operator ini cukup menggunakan tanda “=”. Yang berarti operan sebelah kiri mendapat nilai dari operan sebelah kanan.

*Contoh penggunaan assignment operator :*

OUTPUT :

<div class="card assignmentOP">

<div class="card-header card-inverse card-warning mb-3">

PHP Assignment Operators

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** 6;

**$add** **=** 1; **$sub** **=** 2; **$mult** **=** 3; **$div** **=** 4; **$mod** **=** 5;

**$add** **+=** 3; **$sub** **-=** 3; **$mult** **\*=** 3; **$div** **/=** 3; **$mod** **%=** 3;

**echo** "Assignment normal(x = y) : "**.$x.**"<br>";

**echo** "Assignment-Addition (x += y) : "**.$add.**"<br>";

**echo** "Assignment-Subtraction (x -= y) : "**.$sub.**"<br>";

**echo** "Assignment-Multiplication (x \*= y) : "**.$mult.**"<br>";

**echo** "Assignment-Division (x /= y) : "**.$div.**"<br>";

**echo** "Assignment-Modulus (x %= y) : "**.$mod.**"<br>";

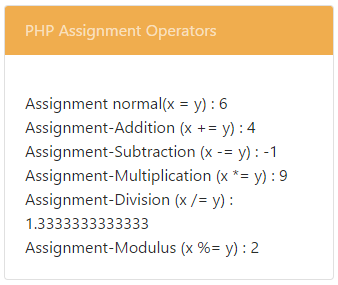
?>

</blockquote>

</div>

</div>

<br>

**

**7.3 PHP Comparison Operators**

Comparison operator milik PHP digunakan untuk membandingan dua nilai (bisa bertipe angka ataupun string)

*Contoh penggunaan comparison operators pada PHP :*

OUTPUT :

<div class="card comparisonOP">

<div class="card-header card-inverse card-warning mb-3">

PHP Comparison Operators

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** 100;

**$y** **=** "100";

**$z** **=** 283;

**echo** "Equal(x == y) : "; **var\_dump**(**$x** **==** **$y**); **echo** "<br>";

**echo** "Identical(x === y) : "; **var\_dump**(**$x** **===** **$y**); **echo** "<br>";

**echo** "Not Equal(x != y) : "; **var\_dump**(**$x** **!=** **$z**); **echo** "<br>";

**echo** "Not Equal(x <> y) : "; **var\_dump**(**$x** **<>** **$z**); **echo** "<br>";

**echo** "Not Identical(x !== y) : "; **var\_dump**(**$x** **!==** **$y**); **echo** "<br>";

**echo** "Greater Than(x > y) : "; **var\_dump**(**$x** **>** **$z**); **echo**"<br>";

**echo** "Less Than(x < y) : "; **var\_dump**(**$x** **<** **$z**); **echo** "<br>";

**echo** "Greater Than or Equal(x >= y) : "; **var\_dump**(**$x** **>=** **$z**); **echo** "<br>";

**echo** "Greater Than or Equal(x <= y) : "; **var\_dump**(**$x** **<=** **$z**); **echo** "<br>";

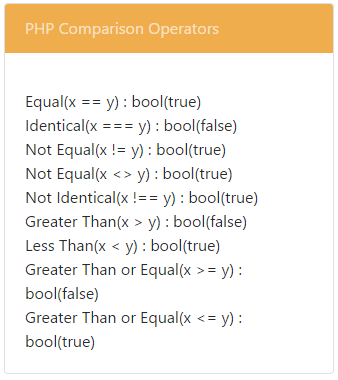
?>

</blockquote>

</div>

</div>

<br>

**

**7.4 PHP Increment/Decrement Operators**

Operator increment digunakan untuk melakukan operasi increment nilai pada sebuah variabel, sedangkan operator decrement digunakan untuk melakukan operasi decrement nilai pada sebuah variabel.

*Contoh penggunaan operator incremen/decrement :*

OUTPUT :

<div class="card arithmaticsOP">

<div class="card-header card-inverse card-warning mb-3">

PHP Arithmetic Operators

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** 6;

**$y** **=** 3;

**echo** "Addition : "**.**(**$x** **+** **$y**)**.**"<br>";

**echo** "Subtraction : "**.**(**$x** **-** **$y**)**.**"<br>";

**echo** "Multiplication : "**.**(**$x** **\*** **$y**)**.**"<br>";

**echo** "Division : "**.**(**$x** **/** **$y**)**.**"<br>";

**echo** "Modulus : "**.**(**$x** **%** **$y**)**.**"<br>";

**echo** "Exponentiation : "**.**(**$x** **\*\*** **$y**)**.**"<br>";

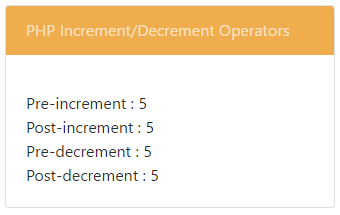
?>

</blockquote>

</div>

</div>

<br>

**

**7.5 PHP Logical Operators**

Operator logika digunakan untuk mengombinasikan conditional statement, terdapat beberapa opertor logika pada PHP, antara lain yaitu and, or, xor, not, dll

*Contoh penggunaan logical operators pada PHP :*

OUTPUT :

<div class="card logicalOP">

<div class="card-header card-inverse card-warning mb-3">

PHP Logical Operators

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$x** **=** true;

**$y** **=** false;

**echo** "And Operator(And/&&) : "; **var\_dump**(**$x** **and** **$y**); **echo** "<br>";

**echo** "Or Operator(Or/||) : "; **var\_dump**(**$x** **or** **$y**); **echo** "<br>";

**echo** "Xor Operator(Xor) : "; **var\_dump**(**$x** **xor** **$y**); **echo** "<br>";

**echo** "Not Operator(!) : "; **var\_dump**(**!$x**); **echo** "<br>";

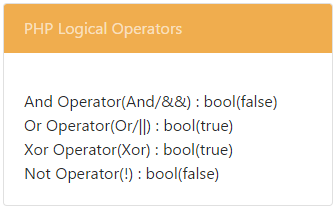
?>

</blockquote>

</div>

</div>

<br>

****

**7.6 PHP String Operators**

PHP mempunyai dua operator yang khusus dibuat untuk string. Fungsi utamanya adalah untuk menggabungkan string yang ada. Dua operator itu adalah concatenation dan concatenation assignment.

*Contoh strings operator pada PHP :*

OUTPUT :

<**div** class="card stringOP">

<**div** class="card-header card-inverse card-warning mb-3">

PHP **String** Operators

</**div**>

<**div** class="card-block">

<blockquote class="card-blockquote">

<?php

$x = "Politeknik Elektronika ";

$y = "Negeri Surabaya";

echo "Concatenation(.) : ".$x.$y."<br>";

echo "Concatenation Assignment(.=) : ".$x.=$y."<br>";

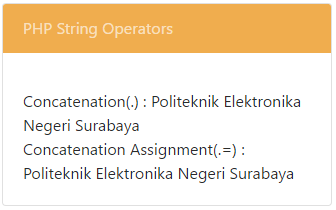
?>

</blockquote>

</**div**>

</**div**>

<br>



**7.7 PHP Array Operators**

Array operators pada PHP digunakan untuk operasi terkait dengan array. Pada umumnya digunakan untuk membandingkan dua buah array.

*Contoh penggunaan operator array pada PHP :*

OUTPUT :

<div class="card arrayOP">

<div class="card-header card-inverse card-warning mb-3">

PHP Array Operators

</div>

<div class="card-block">

<blockquote class="card-blockquote">

<?php

**$array1** **=** **array**(1, 1, 2, 3, 5, 8, 11, 19, 30, 49, 79);

**$array2** **=** **array**(1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21);

**echo** "Union(+) : "; **print\_r**(**$array1** **+** **$array2**); **echo** "<br>";

**echo** "Equality(==) : "; **var\_dump**(**$array1** **==** **$array2**); **echo** "<br>";

**echo** "Identity(===) : "; **var\_dump**(**$array1** **===** **$array2**); **echo** "<br>";

**echo** "Inequality(!=) : "; **var\_dump**(**$array1** **!=** **$array2**); **echo** "<br>";

**echo** "Inequality(<>) : "; **var\_dump**(**$array1** **<>** **$array2**); **echo** "<br>";

**echo** "Non-identity(!==) : "; **var\_dump**(**$array1** **!==** **$array2**); **echo** "<br>";

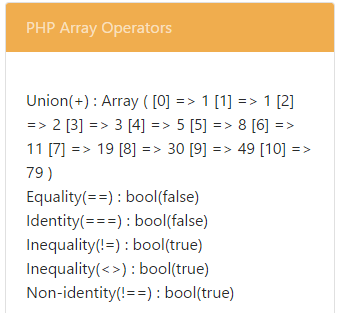
?>

</blockquote>

</div>

</div>

<br>

**