



# Bab I

## Pengenalan dan Definisi PHP



## Dasar PHP



# Apa itu PHP ?

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>.





## Dasar PHP



# Sejarah PHP

❖ **PHP / FI, 1994 ( Personal Home Page / Form Interpreter )**

❖ **PHP Tools v.1, 1995**

❖ **PHP Tools v.2, 1997**

❖ **PHP 3, 1998**

- Zeev Suraski & Andi Gutmans
- Zend Technologies
- PHP : Hypertext Preprocessor

❖ **PHP 4, 2000**

- Zend Engine

❖ **PHP 5, 2004**

- Zend Engine 2
- OOP PHP
- PHP Data Object ( PDO )



## Dasar PHP



# Sejarah PHP

- ❖ **2008, PHP 4 berhenti dikembangkan**
  - Gerakan GoPHP5
- ❖ **PHP 6**
  - Ditunda, dan akhirnya diabaikan
  - Masalah di Unicode
  - PHP 5.4, sampai 2010
- ❖ **PHP 7, 2014**
  - Zend Engine 3



# Sekilas tentang PHP



## Dasar PHP

- ❖ Ekstensi file .php
- ❖ Ditulis didalam tag php
  - Delimiter
  - Diawali dengan `<?php`
  - Diakhiri dengan `?>`
- ❖ Bisa digunakan bersamaan dengan HTML
- ❖ Mengikuti kaidah Bahasa C



# 9 Alasan Mengapa Kita Belajar PHP ?

## 1. PHP Relatif Mudah bagi pemula

```
class HelloWorld {  
    public static void main( String args[] ) {  
        System.out.println("Hello World!");  
    }  
}
```

**Java**

**Vs**

```
<?php  
  
echo "Hello World !";  
  
?>
```



# Mengapa Kita Belajar PHP ?

## 2. Syarat belajar pemrograman PHP

**HTML**



**CSS**





# Mengapa Kita Belajar PHP ?

## 3. Open Source dan Gratis

### Current Stable PHP 7.3.6 (Changelog)

- [php-7.3.6.tar.bz2](#) ([sig](#)) [14,493Kb] 30 May 2019  
sha256: 1e5ac8700154835c0910e3a814517da9b87bb4a82cc7011fea1a82096b6f6f77
- [php-7.3.6.tar.gz](#) ([sig](#)) [18,993Kb] 30 May 2019  
sha256: 72fbf223ff8659a61eed08eebffb4ede0256e7a69d2151ae24affa5377b70bb8
- [php-7.3.6.tar.xz](#) ([sig](#)) [11,708Kb] 30 May 2019  
sha256: fefc8967daa30ebc375b2ab2857f97da94ca81921b722ddac86b29e15c54a164
- [Windows downloads](#)

Source : <https://www.php.net/downloads.php>





# Mengapa Kita Belajar PHP ?

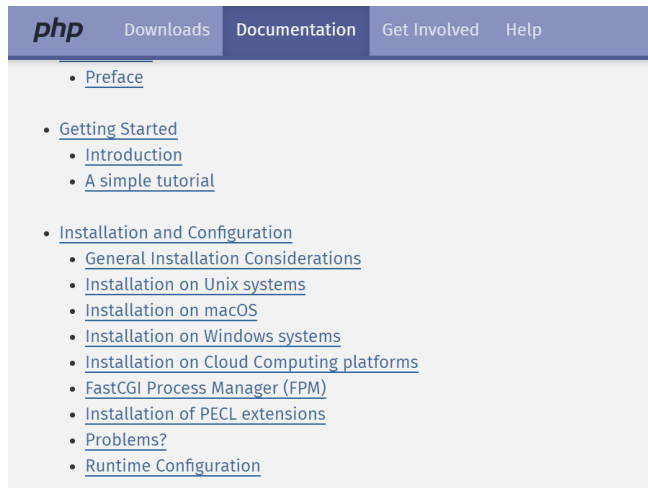
## 4. Dukungan Komunitas sangat besar





# Mengapa Kita Belajar PHP ?

## 5. Dokumentasi sangat lengkap



<https://www.php.net/manual/en/index.php>



# Mengapa Kita Belajar PHP ?

## 5. Dokumentasi sangat lengkap

**php** Downloads Documentation Get Involved Help

- [Preface](#)
- [Getting Started](#)
  - [Introduction](#)
  - [A simple tutorial](#)
- [Installation and Configuration](#)
  - [General Installation Considerations](#)
  - [Installation on Unix systems](#)
  - [Installation on macOS](#)
  - [Installation on Windows systems](#)
  - [Installation on Cloud Computing platforms](#)
  - [FastCGI Process Manager \(FPM\)](#)
  - [Installation of PECL extensions](#)
  - [Problems?](#)
  - [Runtime Configuration](#)

### Tutorial Pemrograman PHP untuk Pemula

PHP adalah bahasa pemrograman yang umum digunakan untuk membuat web

#### Tutorial PHP untuk Pemula

1. [Persiapan Perlengkapan untuk Pemrograman PHP di Linux \(recommend\)](#)
2. [Persiapan Pemrograman Web di Linux menggunakan XAMPP \(opsional\)](#)
3. [Cara Menggunakan Web Server dari PHP](#)
4. [Membuat Program PHP Pertama](#)
5. [Memahami Struktur Dasar Sintak Program PHP](#)
6. [Memahami Perbedaan Fungsi echo\(\), print\(\), dan printf\(\) pada PHP](#)
7. [Variabel dan Tipe Data di PHP](#)
8. [Apa itu Konstanta?](#)
9. [Menenal 9 Magic Constants di PHP](#)

<https://www.php.net/manual/en/index.php>

<https://www.petanikode.com/tutorial/php/>



# Mengapa Kita Belajar PHP ?

## 6. PHP dibuat khusus untuk Web

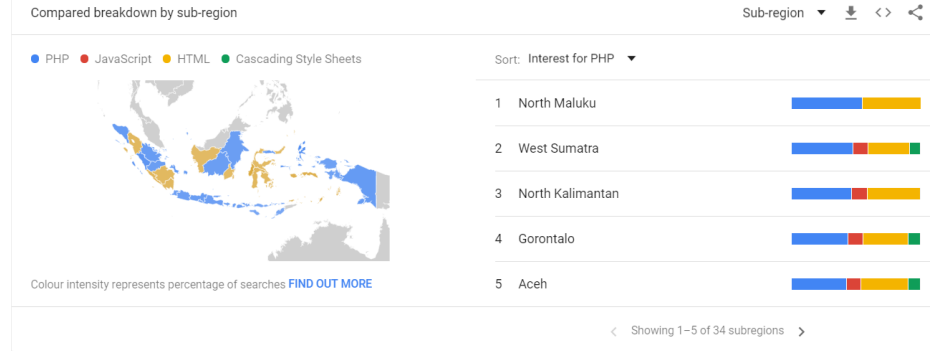
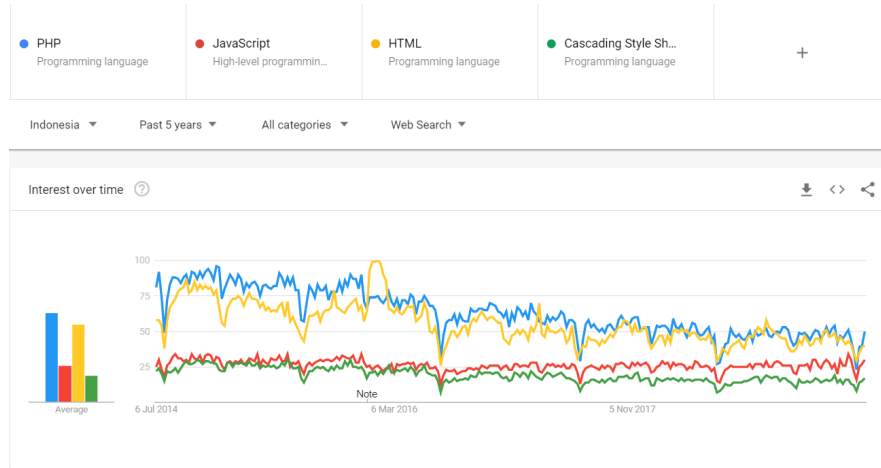


<https://www.arpatech.com/blog/wp-content/uploads/2017/02/choose-php-for-website.png>



# Mengapa Kita Belajar PHP ?

## 7. Mayoritas website yang ada menggunakan PHP



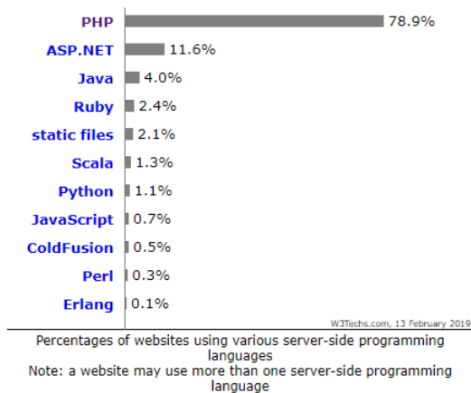
<https://trends.google.com/trends/explore?date=today%205-y&geo=ID&q=%2Fm%2F060kv,%2Fm%2F02p97,%2Fm%2F03g20,%2Fm%2F015tjh>



# Mengapa Kita Belajar PHP ?

## 7. Mayoritas website yang ada menggunakan PHP

It is the best server-side programming language according to a survey by [w3techs.com](http://w3techs.com).



<https://hackernoon.com/8-top-programming-languages-frameworks-of-2019-2f08d2d21a1>



# Mengapa Kita Belajar PHP ?

8. Banyak CMS yang menggunakan Bahasa PHP





# Mengapa Kita Belajar PHP ?

## 9. Banyak Framework PHP yang Sangat Handal





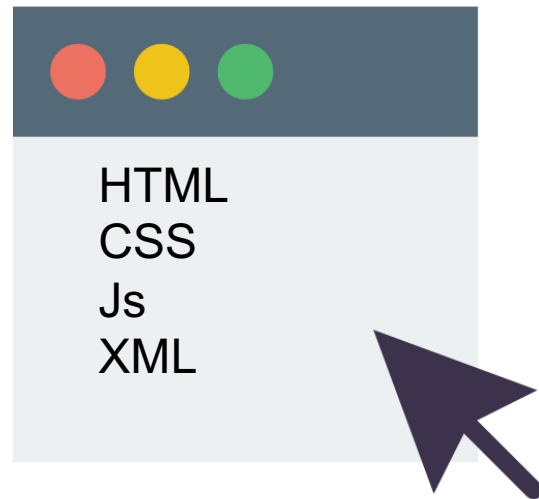
## Bab II



# Client-side vs Server-side Scripting Dan Environment Development



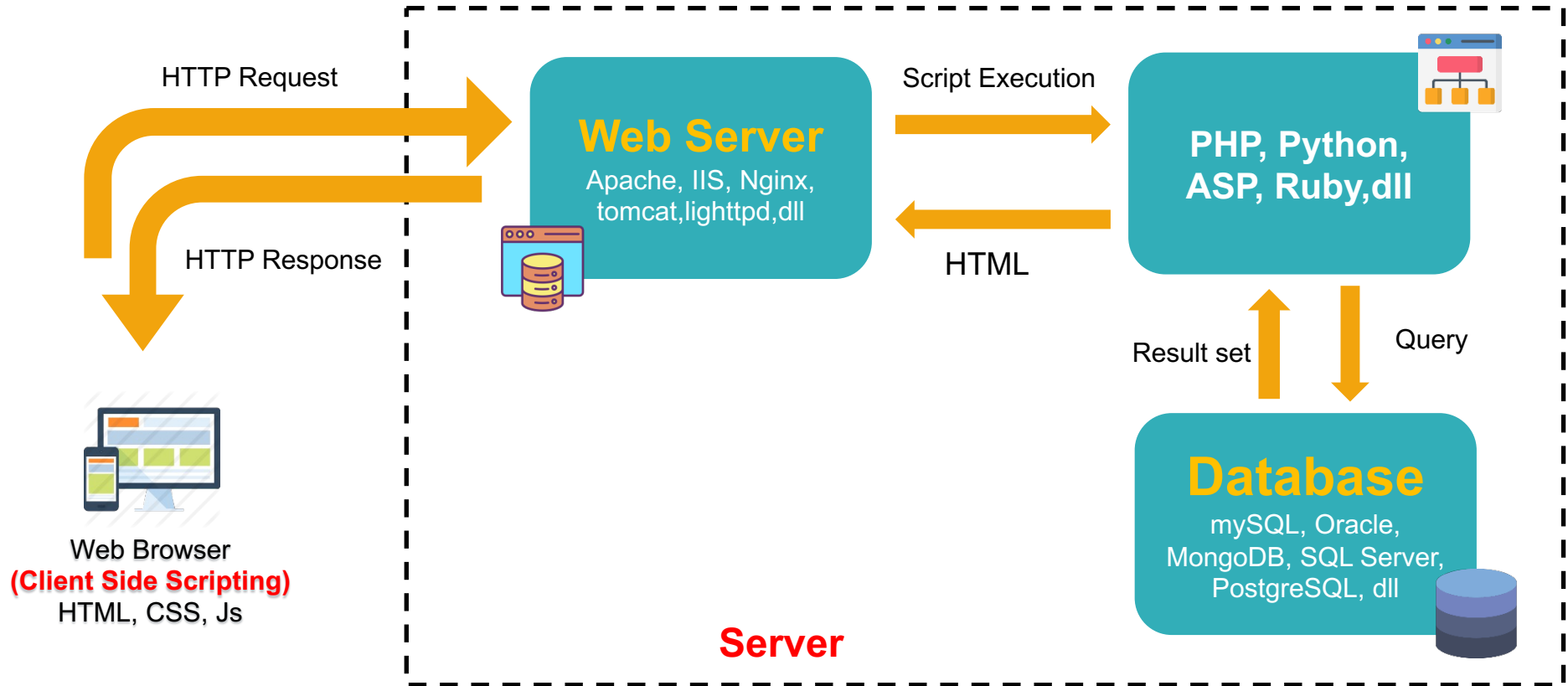
# Client-side Scripting



Web Browser



# Server-side Scripting





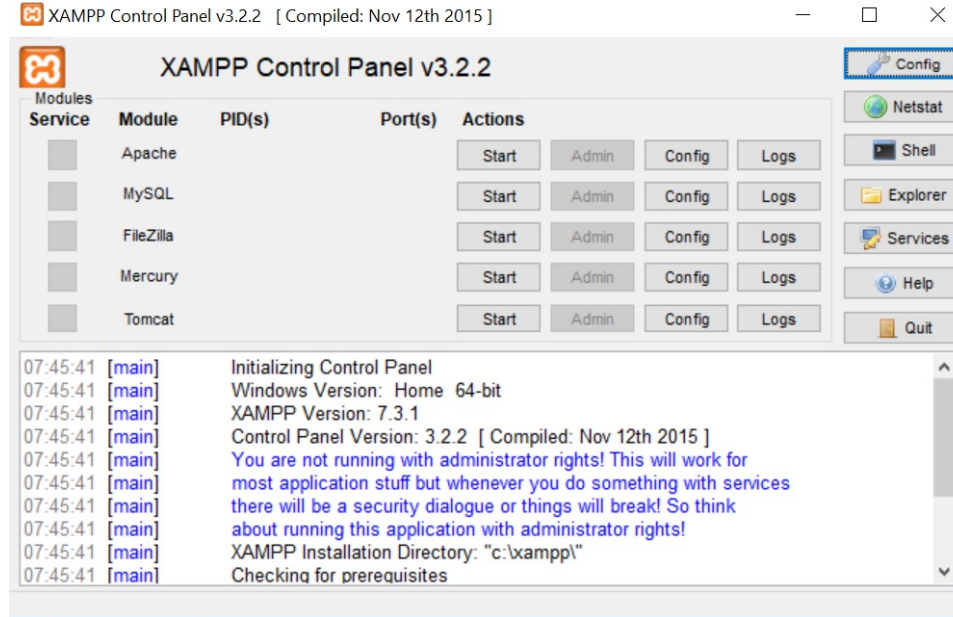
# Persiapan Aplikasi Website PHP

Tools yang diperlukan :

1. Code Editor ( **Visual Studio Code**, Sublime Text, Notepad ++ )
2. XAMPP
3. Browser ( Google chrome, Mozilla Firefox, dll )



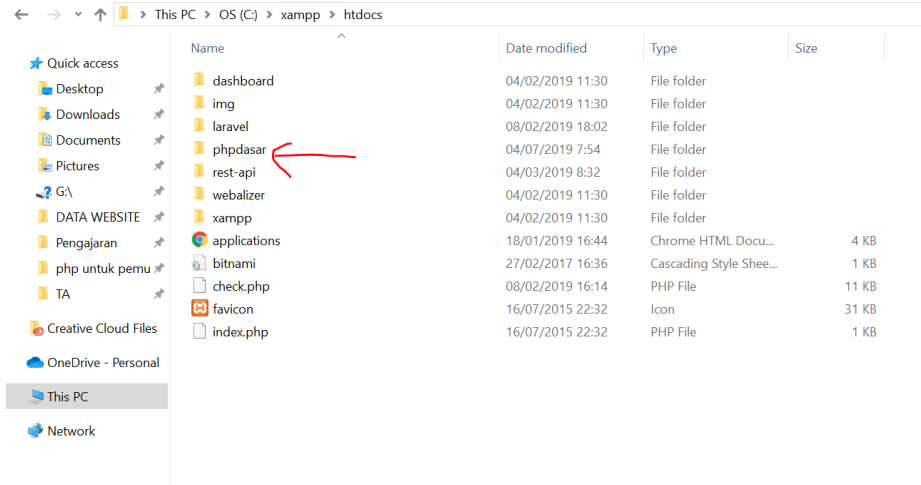
# Persiapan Aplikasi Website PHP



<https://www.apachefriends.org/download.html>



# Persiapan Aplikasi Website PHP



1. Buka Direktori C:xampp/htdocs
2. Bikin Folder dengan nama misalkan **Phpdasar**
3. Didalam folder **phpdasar** bikin folder **Pertemuan1**
4. Di dalam folder **pertemuan1** bikin file Dengan nama **index.php**



# LETS CODE

```
index.php x
pertemuan1 ▶ index.php
1  <?php
2      echo "Hallo World";
3      echo 'Rahmat Fauzi';
4      echo 123;
5      echo true;
6      echo false;
7  ?>
8
```

Coba Jalankan / Run  
Coddingan ini  
Pada browser anda.

Hallo WorldRahmat Fauzi1231



# LETS CODE

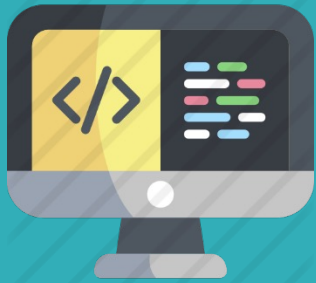
```
index2.php x
pertemuan2 > index2.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Percobaan1</title>
5  </head>
6  <body>
7
8  <p>hello world</p>
9  <p>Selamat Datang di Pemrograman Web
10     <?php
11         echo "Rahmat Fauzi :)";
12     ?>
13 </p>
14
15 </body>
16 </html>
```

Coba Jalankan / Run  
Coddingan ini  
Pada browser anda.

hello world

Selamat Datang di Pemrograman Web Rahmat Fauzi :)





# Bab 3

## Variabel dan Konstanta

# Variabel dalam PHP



## Dasar PHP

Apakah kamu pernah menemukan **x dan y** dalam pelajaran Matematika ?

X dan Y adalah **Variabel** yang menyimpan sesuatu.

Di pemrograman juga memiliki arti yang sama seperti dalam Matematika.

Variabel adalah tempat menyimpan nilai sementara.

Variabel akan ada selama kita menjalankan program. Tapi bisa juga kita menghapusnya dari memori.

**Cara membuat Variabel** adalah dengan **tanda dollar (\$)**

# Konstanta dalam PHP

Konstanta seperti variabel. Ia bisa menyimpan nilai. Tapi tidak bisa diubah.

Pada PHP, kita dapat membuat konstanta dengan dua cara.

1. Menggunakan fungsi **define()**;
2. Menggunakan kata kunci **const**.

```
const PHI = 3.14;  
const API_KEY = "182939812739812478u12ehj1h2u3123h12";
```

```
<?php  
// membuat konstanta dengan fungsi define()  
define('DB_SERVER', 'localhost');  
define('DB_USER', 'Roniwijaya77');  
define('DB_PASS', 'R4Hasia');  
define('DB_NAME', 'belajar');  
  
// membuat konstanta dengan kata kunci const  
const API_KEY = "1234";  
?>
```

[ Tambahan Informasi / Pengayaan ]

<https://www.petanikode.com/php-magic-constant>



Dasar  
PHP

# LETS CODE Variabel PHP



## Dasar PHP

```
<?php
$nama = "Rahmat Fauzi";
?>

<!DOCTYPE html>
<html>
<head>
<title>Percobaan1</title>
</head>
<body>

<p>hello world</p>
<p>Selamat Datang di Pemrograman Web
    <?php
        echo"$nama :>";
    ?>
</p>

</body>
</html>
```

Contoh Variabel

```
<?php
$nama = "Fauzi";
$nama = "Rahmat";
?>

<!DOCTYPE html>
<html>
<head>
<title>Percobaan1</title>
</head>
<body>

<p>hello world</p>
<p>Selamat Datang di Pemrograman Web
    <?php
        echo"$nama :>";
    ?>
</p>

</body>
</html>
```

Oya, variabel juga dapat diisi ulang dengan nilai yang baru



## Dasar PHP

# Aturan membuat nama Variabel

1. Awal dari nama variabel tidak boleh menggunakan angka dan simbol, kecuali underscore.
2. Nama variable yang terdiri dari dua suku kata, bisa dipisah dengan underscore atau menggunakan style camelCase.
3. Variabel harus berisi saat pembuatannya. Jika tidak ingin mengisi, cukup diisi dengan nilai kosong.
4. Nama variable bersifat Case Sensitive, artinya huruf besar dan huruf kecil nilainya berbeda.

```
$!nama = "";  
$10nama = "";  
  
$_nama = "";  
$nama10 = "";
```

```
$nama_lengkap = "Anton Budi";  
$namaPanggilan = "Budi";
```

```
$nama_barang = "";  
$namaPembeli = "";  
$harga = 1000;
```

```
$Nama = "";  
$NAMA = "";  
$nama = "";
```



# Mengambil Nilai dari Variabel PHP

```
<?php
//membuat variable
$nama_barang = "Laptop ASUS";
$harga = 5000000;

?>

<!DOCTYPE html>
<html>
<head>
<title>Percobaan1</title>
</head>
<body>

<p>Amir
    <?php
        echo "membeli $nama_barang dengan harga $harga ";
    ?>
</p>

</body>
</html>
```

Amir membeli Laptop ASUS dengan harga 5000000



# Mengenal Tipe Data

Dalam PHP, Variabel yang sudah dibuat bisa kita simpan dengan berbagai jenis data. Jenis – jenis Data dalam PHP adalah :

1. Tipe data char (karakter)
2. Tipe data string (teks)
3. Tipe data integer (angka)
4. Tipe data float (pecahan)
5. Tipe data boolean
6. Tipe data objek
7. Tipe data Array
8. NULL
9. dll.

```
<?php
//membuat variable
// tipe data char (karakter)
$jenis_kelamin = 'L';
// tipe data string (teks)
$nama_lengkap = "Abdullah bin Salman";
$alamat = "Jl. Soekarno Hatta No.90 Bandung ";
// tipe data integer
$umur = 30;

// tipe data float
$berat = 48.3;

// tipe data float
$tinggi = 182.2;

// tipe data boolean
$menikah = false;

echo "Nama: $nama_lengkap<br>";
echo "Jenis Kelamin: $jenis_kelamin<br>";
echo " Alamat : $alamat <br>";
echo "Umur: $umur tahun<br>";
echo "berat badan: $berat kg<br>";
echo "tinggi badan: $tinggi cm<br>";
echo "menikah: $menikah";

?>
```

Nama: Abdullah bin Salman  
Jenis Kelamin: L  
Alamat : Jl. Soekarno Hatta No.90 Bandung  
Umur: 30 tahun  
berat badan: 48.3 kg  
tinggi badan: 182.2 cm  
menikah:



# Bab 4

## Operator dalam PHP





# Operator dalam PHP

Ada 6 +1 Jenis operator dalam pemrograman PHP yang harus kita ketahui:

1. Operator Aritmatika;
2. Operator Penugasan atau Assignment;
3. Operator Increment & Decrement;
4. Operator Relasi atau pembandingan;
5. Operator Logika;
6. Operator Bitwise;
7. dan Operator Ternary.



# Operator Aritmatika

Operator	Nama	Contoh	Hasil
+	Penjumlahan	$\$a + \$b$	Jumlah dari $\$a$ dan $\$b$
-	Pengurangan	$\$a - \$b$	Selisih dari $\$a$ dan $\$b$
*	Perkalian	$\$a * \$b$	Hasil kali dari $\$a$ dan $\$b$
/	Pembagian	$\$a / \$b$	Hasil bagi dari $\$a$ dan $\$b$
%	Modulus	$\$a \% \$b$	Sisa hasil bagi dari $\$a$ dan $\$b$
**	Eksponensiasi	$\$a ** \$b$	$\$a$ dipangkatkan sebesar $\$b$



# Operator Aritmatika

```
<?php
$a = 12;
$b = 5;
// penjumlahan
$c = $a + $b;
echo "$a + $b = $c";
echo "<hr>";
// pengurangan
$c = $a - $b;
echo "$a - $b = $c";
echo "<hr>";
// Perkalian
$c = $a * $b;
echo "$a * $b = $c";
echo "<hr>";
// Pembagian
$c = $a / $b;
echo "$a / $b = $c";
echo "<hr>";
// Sisa bagi
$c = $a % $b;
echo "$a % $b = $c";
echo "<hr>";
// Pangkat
$c = $a ** $b;
echo "$a ** $b = $c";
echo "<hr>";
?>
```

$$12 + 5 = 17$$

---

$$12 - 5 = 7$$

---

$$12 * 5 = 60$$

---

$$12 / 5 = 2.4$$

---

$$12 \% 5 = 2$$

---

$$12 ** 5 = 248832$$



# Operator Penugasan

Operator Penugasan digunakan dengan nilai numerik untuk menulis nilai pada variabel. Operator penugasan dasar di PHP adalah "=". Ini berarti bahwa operan kiri diatur ke nilai ekspresi tugas di sebelah kanan.

Nama Operator	Symbol
Pengisian Nilai	=
Pengisian dan Penambahan	+=
Pengisian dan Pengurangan	-=
Pengisian dan Perkalian	*=
Pengisian dan Pemangkatan	**=
Pengisian dan Pembagian	/=
Pengisian dan Sisa bagi	%=
Pengisian dan Peggabungan ( string)	.=



# Operator Penugasan

```
<?php
$speed = 83;

// ini opertor aritmatika
$speed = $speed + 10;

// maka nilai speed akan samadengan 83 + 10 = 93

// ini operator penugasan
$speed += 10;

// sekarang nilai speed akan menjadi 93 + 10 = 103
echo "$speed";

?>
```

103

```
<?php
$speed = 83;

// ini opertor aritmatika
$speed = $speed + 10;

// maka nilai speed akan samadengan 83 + 10 = 93

// ini operator penugasan
$speed += 10;

// sekarang nilai speed akan menjadi 93 + 10 = 103

$speed *=2;
//sekarang nilai speed dikalikan 2, menjadi 103*2 = 206

echo "$speed";

?>
```

206



# Operator Increment & Decrement

Operator increment dan decrement merupakan operator yang digunakan untuk menambah +1 (tambah satu) dan mengurangi -1 (kurangi dengan satu).

Operator increment menggunakan **simbol ++**, sedangkan decrement menggunakan **simbol --**.

Penugasan	Nama	Deskripsi
++\$x	Pre-increment	Nilai \$x ditambah 1 terlebih dahulu, kemudian nilai \$x dikembalikan
\$x++	Post-increment	Mengembalikan nilai \$x terlebih dahulu, kemudian nilai \$x ditambah 1
--\$x	Pre-decrement	Nilai \$x dikurangi 1 terlebih dahulu, kemudian nilai \$x dikembalikan
\$x--	Post-decrement	Mengembalikan nilai \$x terlebih dahulu, kemudian nilai \$x dikurangi 1



# Operator Increment & Decrement

```
<?php  
  
$score = 100;  
$score++;  
  
echo "$score" ;  
?>
```

```
<?php  
  
$score = 100;  
--$score;  
  
echo "$score" ;  
?>
```



# Operator Relasi

Operator relasi adalah operator untuk membandingkan dua buah nilai.

Hasil operasi dari operator relasi akan menghasilkan nilai dengan tipe data boolean, yaitu true (benar) dan false (salah).

Operator	Nama	Contoh	Hasil
==	Sama dengan	$\$x == \$y$	<i>true</i> , jika nilai $\$x$ sama dengan $\$y$
===	Identik	$\$x === \$y$	<i>true</i> , jika nilai dan tipe data $\$x$ sama dengan $\$y$
!=	Tidak sama dengan	$\$x != \$y$	<i>true</i> , jika nilai $\$x$ tidak sama dengan $\$y$
<>	Tidak sama dengan	$\$x <> \$y$	<i>true</i> , jika nilai $\$x$ tidak sama dengan $\$y$
!==	Tidak identik	$\$x !== \$y$	<i>true</i> , jika nilai dan tipe data $\$x$ tidak sama dengan $\$y$
>	Lebih dari	$\$x > \$y$	<i>true</i> , jika nilai $\$x$ lebih besar dari $\$y$
<	Kurang dari	$\$x < \$y$	<i>true</i> , jika nilai $\$x$ lebih kecil dari $\$y$
>=	Lebih dari sama dengan	$\$x >= \$y$	<i>true</i> , jika nilai $\$x$ lebih besar atau sama dengan $\$y$
<=	Kurang dari sama dengan	$\$x <= \$y$	<i>true</i> , jika nilai $\$x$ lebih kecil atau sama dengan $\$y$
??	Null coalescing	$\$x ?? \$y$	$\$x$ , jika $\$x$ tidak null $\$y$ , jika $\$x$ null
<=>	Pesawat luar angkasa / spaceship	$\$x <=> \$y$	1, jika $\$x > \$y$ 0, jika $\$x = \$y$ -1, jika $\$x < \$y$





# Operator Relasi

```
<?php
$a = 20;
$b = 10;
// menggunakan operator relasi lebih besar
$c = $a > $b;
echo "$a > $b: $c";
echo "<hr>";
// menggunakan operator relasi lebih kecil
$c = $a < $b;
echo "$a < $b: $c";
echo "<hr>";
// menggunakan operator relasi lebih sama dengan
$c = $a == $b;
echo "$a == $b: $c";
echo "<hr>";
// menggunakan operator relasi lebih tidak sama dengan
$c = $a != $b;
echo "$a != $b: $c";
echo "<hr>";
// menggunakan operator relasi lebih besar sama dengan
$c = $a >= $b;
echo "$a >= $b: $c";
echo "<hr>";
// menggunakan operator relasi lebih kecil sama dengan
$c = $a <= $b;
echo "$a <= $b: $c";
echo "<hr>";
?>
```

20 > 10: 1

---

20 < 10:

---

20 == 10:

---

20 != 10: 1

---

20 >= 10: 1

---

20 <= 10:

---



# Operator Logika

Operator Logika digunakan untuk menggabungkan pernyataan kondisional.

Jika kamu pernah belajar logika matematika, kamu pasti tidak akan asing dengan operator ini.

Operator logika adalah operator untuk melakukan operasi logika seperti **AND, OR, dan NOT**.

Operator	Nama	Contoh	Hasil
and	And	\$x and \$y	<b>true</b> , jika \$x dan \$y sama - sama bernilai <b>true</b> <b>false</b> , jika diantara \$x atau \$y ada yang bernilai <b>false</b>
or	Or	\$x or \$y	<b>true</b> , jika diantara \$x atau \$y ada yang bernilai <b>true</b> <b>true</b> , jika \$x dan \$y sama - sama bernilai <b>true</b>
xor	Xor	\$x xor \$y	<b>true</b> , jika diantara \$x atau \$y ada yang bernilai <b>true</b> <b>false</b> , jika \$x dan \$y sama - sama bernilai <b>true</b>
&&	And	\$x && \$y	<b>true</b> , jika \$x dan \$y sama - sama bernilai <b>true</b> <b>false</b> , jika diantara \$x atau \$y ada yang bernilai <b>false</b>
	Or	\$x    \$y	<b>true</b> , jika diantara \$x atau \$y ada yang bernilai <b>true</b> <b>true</b> , jika \$x dan \$y sama - sama bernilai <b>true</b>
!	Not	!\$x	<b>true</b> , jika \$x bernilai <b>false</b>



# Operator Logika

```
<?php
$a = true;
$b = false;

// variabel $c akan bernilai false
$c = $a && $b;
printf("%b && %b = %b", $a,$b,$c);
echo "<hr>";

// variabel $c akan bernilai true
$c = $a || $b;
printf("%b || %b = %b", $a,$b,$c);
echo "<hr>";

// variabel $c akan bernilai false
$c = !$a;
printf("!%b = %b", $a, $c);
echo "<hr>";
?>
```

$$1 \ \&\& \ 0 = 0$$

---

$$1 \ || \ 0 = 1$$

---

$$!1 = 0$$

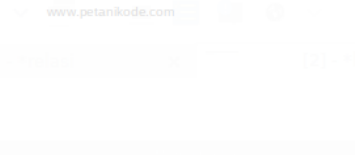
---

Coba cek kembali hukum logika AND, OR, dan NOT.

AND		Hasil
true	true	true
true	false	false
false	true	false
false	false	false

OR		Hasil
true	true	true
true	false	true
false	true	true
false	false	false

NOT	Hasil
true	false
false	true





# Operator Logika

```
<!DOCTYPE html>
<html>
<body>
<?php
    $x = 100;
    $y = 50;
    if ( $x == 100 && $y==50)
    {
        echo"Hello World";
    }
?>
</body>
</html>
```

Hello World

Coba cek kembali hukum logika AND, OR, dan NOT.

AND		Hasil
true	true	true
true	false	false
false	true	false
false	false	false

OR		Hasil
true	true	true
true	false	true
false	true	true
false	false	false

NOT	Hasil
true	false
false	true

[www.petanikode.com](http://www.petanikode.com)

petanikode

100

100



# Operator Bitwise

Operator bitwise merupakan operator yang digunakan untuk operasi bit (biner).

Operator ini berlaku untuk tipe data int, long, short, char, dan byte.

Operator ini akan menghitung dari bit-ke-bit.

Misalnya, kita punya variabel a = 60 dan b = 13.

Bila dibuat dalam bentuk biner, akan menjadi seperti ini:

a = 00111100 ( **60** )

b = 00001101 ( **13** )

Nama	Simbol di <del>Java</del> PHP
AND	&
OR	
XOR	^
Negasi/kebalikan	~
Left Shift	<<
Right Shift	>>



# Operator Bitwise

## Operasi AND

```
a = 00111100  
b = 00001101  
a & b = 00001100
```

## Operasi OR

```
a = 00111100  
b = 00001101  
a | b = 00111101
```

## Operasi XOR

```
a = 00111100  
b = 00001101  
a ^ b = 00110001
```

## Operasi NOT (Negasi/kebalikan)

```
a = 00111100  
~a = 11000011
```



# Operator Bitwise

```
<?php
$a = 60;
$b = 15;
// bitwise AND
$c = $a & $b;
echo "$a & $b = $c";
echo "<br>";

// bitwise OR
$c = $a | $b;
echo "$a | $b = $c";
echo "<br>";

// bitwise XOR
$c = $a ^ $b;
echo "$a ^ $b = $c";
echo "<br>";

// Shift Left
$c = $a << $b;
echo "$a << $b = $c";
echo "<br>";

// Shift Right
$c = $a >> $b;
echo "$a >> $b = $c";
echo "<br>";
?>
```

$60 \& 15 = 12$

$60 | 15 = 63$

$60 \wedge 15 = 51$

$60 \ll 15 = 1966080$

$60 \gg 15 = 0$



# Operator Ternary

Operator ternary adalah operator untuk membuat sebuah kondisi. Simbol yang digunakan adalah **tanda tanya (?)** dan **titik dua (:)**.

Operator Ternary  
↓  
kamu suka aku ? ya : tidak;  
↑                    ↑  
jawaban benar    jawaban salah





# Operator Itenary

```
<?php

$status = true;

// menggunakan operator ternary
$jawab = $status ? "Jomblo": "tidak";

// menampilkan jawaban
echo $jawab;

?>
```

Jomblo



# Operator String

Operator String memiliki dua operator yang dirancang khusus untuk string, yaitu :

Operator	Nama	Contoh	Hasil
.	Rangkaian (Concatenation)	\$text1 . \$text2	Rangkaian dari \$text1 dan \$text2
.=	Pembubuhan (Appends)	\$text1 .= \$text2	Pembubuhan \$text2 kepada \$text1



# Operator String

```
<?php

$nama_depan = "Ahmad";
$nama_belakang = "Setiawan";

echo $nama_depan.$nama_belakang;
echo "<br>";
echo $nama_depan." ".$nama_belakang;
echo "<br>";
echo $nama_depan.$nama_belakang;
echo "<br>";
echo $nama_depan." ".$nama_belakang;

?>
```

AhmadSetiawan  
Ahmad Setiawan  
AhmadSetiawan  
AhmadSetiawan Setiawan



# Operator Array

Operator Array digunakan untuk membandingkan array. Macam macam operator

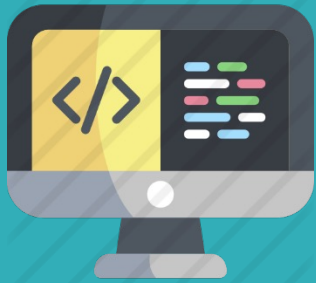
Operator	Nama	Contoh	Hasil
+	Union	$\$x + \$y$	Gabungan dari $\$x$ dan $\$y$
==	Equality	$\$x == \$y$	<i>True</i> , jika $\$x$ dan $\$y$ memiliki pasangan kunci / nilai yang sama
===	Identity	$\$x === \$y$	<i>True</i> , jika $\$x$ dan $\$y$ memiliki pasangan kunci / nilai yang sama dengan urutan yang sama dan tipe data yang sama
!=	Inequality	$\$x != \$y$	<i>True</i> , jika $\$x$ tidak sama dengan $\$y$
<>	Inequality	$\$x <> \$y$	<i>True</i> , jika $\$x$ tidak sama dengan $\$y$
!==	Non-identity	$\$x !== \$y$	<i>True</i> , jika $\$x$ tidak identik dengan $\$y$



# Operator Array

```
<!DOCTYPE html>
<html>
<body>
    <?php
        $x = array("a"=>"anggrek","b"=>"Bougenville");
        $y = array("c"=>"camelia","d"=>"dahlia");
        var_dump($x !== $y);
    ?>
</body>
</html>
```

bool(true)



# Percabangan dan perulangan dalam Program PHP



# Percabangan



# Percabangan If

Bentuk yang paling sederhana dari percabangan adalah “If” saja.

```
<?php
```

```
if (<kondisi>){  
    // eksekusi kode ini  
}
```





# Percabangan If

```
<?php

$total_belanja = 150000;

if($total_belanja > 100000){
    echo "Anda dapat hadiah!";
}

?>
```

Anda dapat hadiah!



# Percabangan If/Else

Percabangan If/Else memiliki dua pilihan. Jika <kondisi> bernilai **false**, maka blok else akan dikerjakan.

```
<?php  
  
$umur = 19;  
  
if ($umur < 18 ){  
    echo "Kamu tidak boleh membuka situs ini!";  
} else {  
    echo "Selamat datang di website kami!";  
}  
  
?>
```

Selamat datang di website kami!



# Percabangan If/Elseif/Else

Percabangan If/Elseif/Else memiliki lebih dari dua pilihan kondisi.

Nilai anda: 78  
Grade: B+

```
<?php

$nilai = 78;

if ($nilai > 90) {
    $grade = "A+";
} elseif($nilai > 80){
    $grade = "A";
} elseif($nilai > 70){
    $grade = "B+";
} elseif($nilai > 60){
    $grade = "B";
} elseif($nilai > 50){
    $grade = "C+";
} elseif($nilai > 40){
    $grade = "C";
} elseif($nilai > 30){
    $grade = "D";
} elseif($nilai > 20){
    $grade = "E";
} else {
    $grade = "F";
}

echo "Nilai anda: $nilai<br>";
echo "Grade: $grade";

?>
```



# Percabangan Switch/Case

Percabangan Switch/Case adalah bentuk lain dari percabangan If/Elseif/Else. Format penulisannya seperti ini:

```
<?php

switch($variabel){
    case <konidisi>:
        // eksekusi kode ini
        break;
    case <kondisi2>:
        // eksekusi kode ini
        break;
    default:
        // eksekusi kode ini
}

?>
```



# Percabangan Switch/Case

```
<?php

$lulus_tingkat = 3;

switch($lulus_tingkat){
    case 1:
        echo "Lulus TPB";
        break;
    case 2:
        echo "Lulus tingkat dasar";
        break;
    case 3:
        echo "Lulus tingkat lanjut";
        break;
    case 4:
        echo "lulus kuliah";
        break;
    default:
        echo "Kamu bukan mahasiswa telkom bojongsoang!";
}

?>
```

Lulus tingkat lanjut



# Percabangan dengan Operator Ternary

Percabangan menggunakan operator ternary adalah bentuk sederhana dari percabangan If/Else..

```
<?php  
  
<kondisi> ? benar : salah;  
  
?>
```



# Percabangan dengan Operator Ternary

```
<?php  
  
$anaktel_u = true;  
  
echo $anaktel_u ? "Aku juga anak tel-u" : "Baiklah!";  
  
?>
```

Aku juga anak tel-u



# Percabangan Bersarang

Percabangan bersarang artinya ada percabangan di dalam percabangan (*nested*).

```
<?php
|
$umur = 20;
$menikah = false;

if($umur > 25){
    if($menikah){
        echo "Selamat datang pak!";
    } else {
        echo "Selamat datang wahai pemuda!";
    }
} else {
    echo "Maaf website ini hanya untuk yang sudah berumur 20+";
}
?>
```

Maaf website ini hanya untuk yang sudah berumur 20+





# Perulangan



# Perulangan

Fungsi Perulangan adalah untuk mencetak data yang berulang kali, misalkan 100 x.

**Perulangan**

For (counted loop)	VS	While (uncounted loop)
		
Push up 10x		Push up Sampai bosan
<pre>for(i=0; i&lt;10; i++){   pushUp(); }</pre>		<pre>while(not bosan){   pushUp(); }</pre>

Petanikode.com @petanikode

Ada dua jenis perulangan dalam pemrograman:

- 1.Counted loop;
- 2.Uncounted loop.



# Perulangan

**Perulangan**

**For**  
(counted loop)

**While**  
(uncounted loop)

VS

Push up 10x

```
for(i=0; i<10; i++){  
    pushUp();  
}
```

Push up  
Sampai bosan

```
while(not bosan){  
    pushUp();  
}
```

Petanikode.com @petanikode

Apa perbedaannya?

*Counted loop* adalah perulangan yang sudah jelas banyak pengulangannya. Sedangkan *Uncounted loop* tidak pasti berapa kali dia akan mengulang.

Pada PHP ada 4 jenis perulangan yang bisa kita gunakan:

1. Perulangan For
2. Perulangan While
3. Perulangan Do/While
4. Perulangan Foreach



# 1.Perulangan For

Perulangan *For* adalah perulangan yang termasuk dalam ***counted loop***, karena kita bisa menentukan jumlah perulangannya. Bentuk dasar perulangan for:

```
<?php

for ($i = 0; $i < 10; $i++){
    // blok kode yang akan diulang di sini!
}

?>
```



# 1.Perulangan For

```
<?php

for($i = 0; $i < 10; $i++){
    echo "<h2>Ini perulangan ke-$i</h2>";
}

?>
```

**Ini perulangan ke-0**

**Ini perulangan ke-1**

**Ini perulangan ke-2**

**Ini perulangan ke-3**

**Ini perulangan ke-4**

**Ini perulangan ke-5**

**Ini perulangan ke-6**

**Ini perulangan ke-7**

**Ini perulangan ke-8**

**Ini perulangan ke-9**



## 2. Perulangan While

Perulangan *while* adalah perulangan yang termasuk dalam ***uncounted loop***. Karena biasanya digunakan untuk mengulang sesuatu yang belum jelas jumlah pengulangannya. Namun, perulangan *while* juga bisa digunakan seperti perulangan *for* sebagai ***counted loop***.

```
<?php

while (<kondisi>){
    // blok kode yang akan diulang di sini
}

?>
```



## 2.Perulangan While

```
<?php  
  
$ulangi = 0;  
  
while($ulangi < 10){  
    echo "<p>Ini adalah perulangan ke-$ulangi</p>";  
    $ulangi++;  
}  
  
?>
```

Ini adalah perulangan ke-0

Ini adalah perulangan ke-1

Ini adalah perulangan ke-2

Ini adalah perulangan ke-3

Ini adalah perulangan ke-4

Ini adalah perulangan ke-5

Ini adalah perulangan ke-6

Ini adalah perulangan ke-7

Ini adalah perulangan ke-8

Ini adalah perulangan ke-9



## 3. Perulangan Do/While

Perulangan *Do/While* sama seperti perulangan *while*. Ia juga tergolong dalam **uncounted loop**.

Perbedaan *Do/While* dengan *while* terletak pada **cara iya memulai pengulangan**.

Perulangan *Do/While* akan selalu melakukan pengulangan **sebanyak 1 kali**, kemudian melakukan pengecekan kondisi.

Sedangkan perulangan *while* akan **mengecek kondisi terlebih dahulu**, baru melakukan pengulangan.

```
<?php

do {
    // blok kode yang akan diulang
} while (<kondisi>);

?>
```





## 3. Perulangan Do/While

```
<?php
$ulangi = 10;

do {
    echo "<p>ini adalah perulangan ke-$ulangi</p>";
    $ulangi--;
} while ($ulangi > 0);

?>
```

ini adalah perulangan ke-10

ini adalah perulangan ke-9

ini adalah perulangan ke-8

ini adalah perulangan ke-7

ini adalah perulangan ke-6

ini adalah perulangan ke-5

ini adalah perulangan ke-4

ini adalah perulangan ke-3

ini adalah perulangan ke-2

ini adalah perulangan ke-1



## 4. Perulangan Foreach

Perulangan *foreach* sama seperti perulangan *for*. Namun, ia lebih khusus digunakan untuk mencetak array. Rumus umum Foreach :

```
<?php  
  
foreach($array as $data){  
    echo $data;  
}
```



## 4. Perulangan Foreach

```
<?php

$books = [
    "Panduan Belajar PHP untuk Pemula",
    "Membangun Aplikasi Web dengan PHP",
    "Tutorial PHP dan MySQL",
    "Membangun Peradaban dari Kampus"
];

echo "<h5>Judul Buku Karya Prof. Adi Sasongko:</h5>";
echo "<ul>";
foreach($books as $buku){
    echo "<li>$buku</li>";
}
echo "</ul>";

?>
```

**Judul Buku Karya Prof. Adi Sasongko:**

- Panduan Belajar PHP untuk Pemula
- Membangun Aplikasi Web dengan PHP
- Tutorial PHP dan MySQL
- Membangun Peradaban dari Kampus



# Bonus: Perulangan Bersarang

Perulangan bersarang adalah istilah untuk menyebut perulangan di dalam perulangan. Dalam bahasa inggris, perulangan bersarang disebut *nested loop*.

```
<?php

for($i = 0; $i < 5; $i++){
    for($j = 0; $j < 10; $j++){
        echo "Ini perulangan ke ($i, $j)<br>";
    }
}

?>
```

Ini perulangan ke (0, 0)  
Ini perulangan ke (0, 1)  
Ini perulangan ke (0, 2)  
Ini perulangan ke (0, 3)  
Ini perulangan ke (0, 4)  
Ini perulangan ke (0, 5)  
Ini perulangan ke (0, 6)  
Ini perulangan ke (0, 7)  
Ini perulangan ke (0, 8)  
Ini perulangan ke (0, 9)  
Ini perulangan ke (1, 0)  
Ini perulangan ke (1, 1)  
Ini perulangan ke (1, 2)  
Ini perulangan ke (1, 3)  
Ini perulangan ke (1, 4)  
Ini perulangan ke (1, 5)  
Ini perulangan ke (1, 6)  
Ini perulangan ke (1, 7)  
Ini perulangan ke (1, 8)  
Ini perulangan ke (1, 9)  
Ini perulangan ke (2, 0)  
Ini perulangan ke (2, 1)  
Ini perulangan ke (2, 2)  
Ini perulangan ke (2, 3)  
Ini perulangan ke (2, 4)  
Ini perulangan ke (2, 5)  
Ini perulangan ke (2, 6)  
Ini perulangan ke (2, 7)  
Ini perulangan ke (2, 8)  
Ini perulangan ke (2, 9)  
Ini perulangan ke (3, 0)



# Bonus: Perulangan Bersarang

Perulangan bersarang adalah istilah untuk menyebut perulangan di dalam perulangan. Dalam bahasa inggris, perulangan bersarang disebut *nested loop*.

```
$i = 0;
while($i < 10){
    for($j = 0; $j < 10; $j++){
        echo "Ini perulangan ke ($i, $j)<br>";
    }

    $i++;
}
```

Ini perulangan ke (0, 0)  
Ini perulangan ke (0, 1)  
Ini perulangan ke (0, 2)  
Ini perulangan ke (0, 3)  
Ini perulangan ke (0, 4)  
Ini perulangan ke (0, 5)  
Ini perulangan ke (0, 6)  
Ini perulangan ke (0, 7)  
Ini perulangan ke (0, 8)  
Ini perulangan ke (0, 9)  
Ini perulangan ke (1, 0)  
Ini perulangan ke (1, 1)  
Ini perulangan ke (1, 2)  
Ini perulangan ke (1, 3)  
Ini perulangan ke (1, 4)  
Ini perulangan ke (1, 5)  
Ini perulangan ke (1, 6)  
Ini perulangan ke (1, 7)  
Ini perulangan ke (1, 8)  
Ini perulangan ke (1, 9)  
Ini perulangan ke (2, 0)  
Ini perulangan ke (2, 1)  
Ini perulangan ke (2, 2)  
Ini perulangan ke (2, 3)  
Ini perulangan ke (2, 4)  
Ini perulangan ke (2, 5)  
Ini perulangan ke (2, 6)



# Latihan 1

Bikinkan Tabel dengan Program HTML dan PHP dengan Fungsi **pengulangan**. Dengan ketentuan 5 baris 5 kolom.

1,1	1,2	1,3	1,4	1,5
2,1	2,2	2,3	2,4	2,5
3,1	3,2	3,3	3,4	3,5
4,1	4,2	4,3	4,4	4,5
5,1	5,2	5,3	5,4	5,5



Thank you



## PHP Dasar

### Link Referensi Tambahan

[www.Youtube.com/webprogrammingUNPAS](http://www.Youtube.com/webprogrammingUNPAS)

Modul Praktikum WAD SI FRI

Modul Praktikum WAD Teknik Informatika FIF

<https://www.petanikode.com/tutorial/php/>