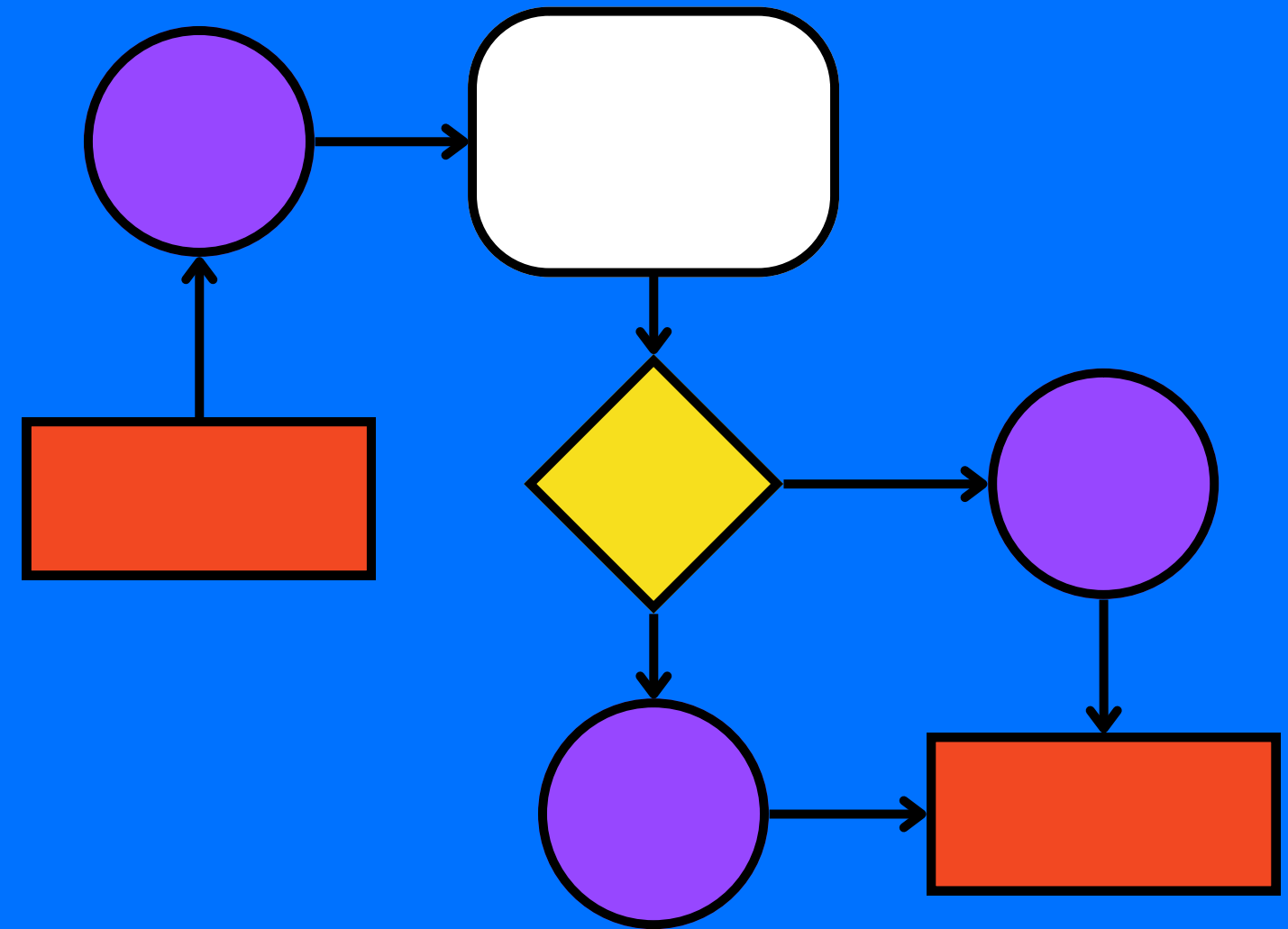


**Jefri Maruli**

# Back-End Development

## Materi 2 (Part 4)



# Agenda Belajar

- **Operator Aritmatika**
- **Operator Penugasan,**
- **Operator Increment dan Decrement**
- **Operator Perbandingan**
- **Operator Logika**



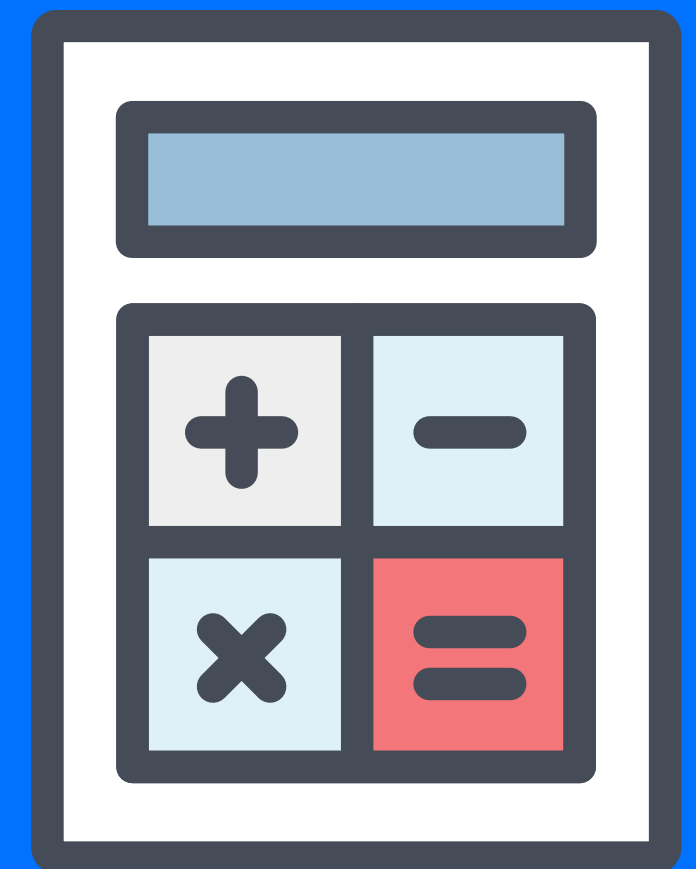
# Operator Aritmatika

Operator aritmatika atau di sebut juga operator matematika merupakan operator yang digunakan untuk pengolahan bilangan perhitungan secara matematis.



# Jenis Operator Aritmatika

- Penjumlahan (*addition*)
- Pengurangan (*subtraction*)
- Perkalian (*multiplication*)
- Pembagian (*division*)
- Sisa hasil bagi (*modulus*)
- Pangkat (*exponentiation*)
- Negasi (*negation*)



## Operator Penjumlahan (*Addition*)

Operator penjumlahan dalam bahasa pemrograman dituliskan dengan simbol "+" yang digunakan menambahkan satu operan dengan operan lainnya



```
1  <?php
2
3  $a = 1;
4  $b = 1;
5  $c = $a + $b;
```

# Implementasi Operator Penjumlahan

```
mardirabootcamp - addition.php
1  <?php
2
3  $number = 5 + 10 + 2 + 4;
4
5  $numberA = 2;
6  $numberB = 3;
7  $numberC = $numberA + $numberB;
8
9  $addition = $number + $numberC;
10
11 echo "Hasil Penjumlahan : {$addition}";
```

Hasil Penjumlahan : 26

## Operator Pengurangan (*Subtraction*)

Operator pengurangan dalam bahasa pemrograman dituliskan dengan simbol "-" yang digunakan mengurangi satu operan dengan operan lainnya



```
1  <?php
2
3  $a = 3;
4  $b = 1;
5  $c = $a - $b;
```

# Implementasi Operator Pengurangan


```
mardirabootcamp - Subtraction.php  
1  <?php  
2  
3  $number = 10 - 1 - 3;  
4  $numberA = 10;  
5  $numberB = 3;  
6  $numberC = $numberA - $numberB;  
7  $subtraction = $number - $numberB;  
8  
9  echo "Hasil Pengurangan : {$subtraction}";
```

Hasil Pengurangan : 3



## Operator Perkalian (*Multiplication*)

Operator perkalian dalam bahasa pemrograman dituliskan dengan simbol "\*" yang digunakan mengalikan satu operan dengan operan lainnya



```
1  <?php
2
3  $a = 10;
4  $b = 5;
5  $c = $a * $b;
```

# Implementasi Operator Perkalian

mardirabootcamp - Multiplication.php

```
1  <?php
2
3  $number = 5 * 10 * 2 * 4;
4  $numberA = 2;
5  $numberB = 3;
6  $numberC = $numberA * $numberB;
7  $multiplication = $number * $numberC;
8
9  echo "Hasil Perkalian : {$multiplication}";
```

Hasil Perkalian : 2400

## Operator Pembagian (*Division*)

Operator pembagian dalam bahasa pemrograman dituliskan dengan simbol "/" yang digunakan membagi satu operan dengan operan lainnya



```
1  <?php
2
3  $a = 8;
4  $b = 2;
5  $c = $a / $b;
```


# Implementasi Operator Pembagian

```
mardirabootcamp - Division.php  
1  <?php  
2  
3  $number = 10 / 2;  
4  $numberA = 2;  
5  $numberB = 3;  
6  $numberC = $numberA / $numberB;  
7  $division = $number / $numberC;  
8  
9  echo "Hasil Pembagian : {$division}";
```

Hasil Pembagian : 7.5

## Operator Sisa Hasil Bagi (*Modulus*)

Mod atau modulus merupakan sisa hasil bagi dalam bahasa pemrograman yang ditulis dengan simbol (%), biasanya digunakan untuk mengembalikan sisa angka setelah dibagi.



```
1  <?php
2
3  $a = 10;
4  $b = 3;
5  $c = $a % $b;
```

# Implementasi Operator Sisa Hasil Bagi

```
mardirabootcamp - Modulus.php  
1  <?php  
2  
3  $number = 5 * 20;  
4  $numberA = 10;  
5  $numberB = 2;  
6  $numberC = $numberA - $numberB;  
7  $modulus = $number % $numberC;  
8  
9  echo "Hasil Modulus : {$modulus}";
```

Hasil Modulus : 4

## Operator Pangkat (*Exponentiation*)

Operator pangkat dalam bahasa pemrograman dituliskan dengan simbol **\*\*** yang digunakan mencari perpangkatan pada suatu nilai.



```
1  <?php
2
3  $a = 3;
4  $b = 2;
5  $c = $a ** $b;
```

# Implementasi Operator Pangkat

● ● ● mardirabootcamp - Exponentiation.php


```
1 $number = 2 ** 2;  
2 $numberA = 2;  
3 $numberB = 2;  
4 $numberC = $numberA ** $numberB;  
5 $exponent = $number ** $numberC;  
6  
7 echo "Hasil Perpangkatan : {$exponent}";
```

Hasil Perpangkatan : 256



## Operator Negasi (Negation)

Operator negasi atau minus yang ditulis dengan simbol "-" pada variable digunakan untuk mengkonversi nilai dari bilangan positif ke negatif dan sebaliknya.



```
1  <?php
2
3  $a = 10;
4  $b = -$a;
5  echo $b;
```

# Implementasi Operator Negasi

```
mardirabootcamp - Minus.php
1  <?php
2
3  $numberA = 2;
4  $numberB = 3;
5  $numberC = $numberA + $numberB;
6  $minus = -$numberC;
7
8  echo "Hasil Konversi Bilangan Negatif : {$minus}";
```

Hasil Konversi Bilangan Negatif : -5

# Alternatif Operator Negasi

```
mardirabootcamp - Minus.php  
1  <?php  
2  
3  $number = -20;  
4  $minus = -$number;  
5  
6  echo "Hasil Konversi : {$minus}";
```

Hasil Konversi : 20

Ketika nilai pada variable hasilnya negatif saat dikonversi menggunakan operator negasi maka akan menghasilkan bilangan positif.

## Operator Penugasan (*Assignment Operator*)

Operator penugasan atau assignment operator merupakan operator yang digunakan untuk mengisi nilai ke dalam variabel yang di tuliskan dengan simbol "=".



# Jenis Operator Penugasan

Operator	Implementasi	Penjabaran	Penjelasan
<b>=</b>	<b>\$a = nilai</b>	<b>\$a = nilai</b>	<b>Simpan nilai ke variable a</b>
<b>+=</b>	<b>\$a += angka</b>	<b>\$a = \$a + angka</b>	<b>jumlahkan \$a dan simpan</b>
<b>-=</b>	<b>\$a -= angka</b>	<b>\$a = \$a - angka</b>	<b>mengurangi \$a dan simpan</b>
<b>*=</b>	<b>\$a *= angka</b>	<b>\$a = \$a * angka</b>	<b>mengalikan \$a dan simpan</b>
<b>/=</b>	<b>\$a /= angka</b>	<b>\$a = \$a / angka</b>	<b>membagi \$a dan simpan</b>
<b>%=</b>	<b>\$a %= angka</b>	<b>\$a = \$a % angka</b>	<b>membagi \$a dan simpan hasil sisa baginya</b>
<b>**=</b>	<b>\$a **= angka</b>	<b>\$a = \$a ** angka</b>	<b>pangkatkan \$a dan simpan</b>

# Implementasi Operator Penugasan Penjumlahan

mardirabootcamp - AssignmentAddition.php

```
1  <?php
2
3  $number = 10;
4  // operator aritmatika hasil = 13
5  $addition = $number + 3;
6
7  // operator penugasan hasil = 15
8  $number += 5;
9
10 echo "Hasil : {$number}";
```

Hasil : 15

# Implementasi Operator Penugasan Pengurangan

mardirabootcamp - AssignmentSubstraction.php

```
1  <?php
2
3  $number = 10;
4  // operator aritmatika hasil = 7
5  $subtraction = $number - 3;
6
7  // operator penugasan hasil = 5
8  $number -= 5;
9
10 echo "Hasil : {$number}";
```

Hasil : 5

# Implementasi Operator Penugasan Perkalian

mardirabootcamp - AssignmentMultiplication.php

```
1  <?php
2
3  $number = 10;
4  // operator aritmatika hasil = 30
5  $multiplication = $number * 3;
6
7  // operator penugasan hasil = 30
8  $number *= 3;
9
10 echo "Hasil : {$number}";
```

Hasil : 30



# Implementasi Operator Penugasan Pembagian

mardirabootcamp - AssignmentDivision.php

```
1  <?php
2
3  $number = 10;
4  // operator aritmatika hasil = 2
5  $division = $number / 5;
6
7  // operator penugasan hasil = 2
8  $number /= 5;
9
10 echo "Hasil : {$number}";
```

Hasil : 2

# Implementasi Operator Penugasan Modulus

mardirabootcamp - AssignmentModulus.php

```
1  <?php
2
3  $number = 10;
4
5  // Operator aritmatika hasil = 1
6  $modulus = $number % 3;
7
8  // Operator penugasan hasil = 1
9  $number %= 3;
10
11 echo "Hasil : {$number}";
```

Hasil : 1

# Implementasi Operator Penugasan Perpangkatan

mardirabootcamp - AssignmentExponentiation.php

```
1  <?php
2
3  $number = 2;
4
5  // Operator aritmatika hasil = 8
6  $exponentiation = $number ** 3;
7
8  // Operator penugasan hasil = 8
9  $number **= 3;
10
11 echo "Hasil : {$number}";
```

A pixelated, multi-colored graphic of the text "Hasil : 8" on a white background. The characters are composed of small, multi-colored squares, giving it a retro, digital art appearance.

# Jenis Operator Penugasan Lainnya

Operator	Implementasi	Penjabaran	Penjelasan
<b>.=</b>	<b>\$a .= \$b</b>	<b>\$a = \$a . \$b</b>	<b>Menggabungkan kedua variable bernilai string (String Concatenation)</b>
<b>??</b>	<b>\$a ?? \$b</b>	<b>\$a = \$a ?? \$b</b>	<b>untuk mengecek suatu variabel tertentu ada atau tidak dan jika ada apakah bernilai null atau tidak (Null Coalescing)</b>

# Operator Penugasan String Concatenation

mardirabootcamp - AssignmentConcatenation.php

```
1  <?php
2
3  $hello = "Hello";
4  $world = "World";
5
6  // String Concatenation = "Hello World"
7  $concat = $hello . $world;
8
9  // Operator penugasan = "Hello World"
10 $hello .= $world;
11
12 echo "Hasil : {$hello}";
```

Hasil : HelloWorld

# Operator Penugasan Null Coalescing

```
mardirabootcamp - AssignmentCoalescing.php

1  <?php
2
3  $string = null;
4
5  // Cara biasa mengecek nilai null dengan statement
6  if (isset($string)) {
7      $string = 'hi';
8  } else {
9      $string = 'Hello World';
10 }
11
12 // Null Coalescing standar
13 $string = $string ?? 'Hello World';
14
15 // Null Coalescing dengan operator penugasan
16 $string ??= 'Hello World';
17
18 echo "Hasil : {$string}";
```

Hasil : Hello World

# Operator Increment dan Decrement

Operator increment dan decrement berfungsi untuk menaikkan dan menurunkan nilai. Untuk menggunakan operator ini wajib menggunakan variabel yang telah di deklarasikan.

```
1  <?php
2
3  $value++;
4  $value--;
5  ++$value;
6  --$value;
```



# Jenis Operator Increment dan Decrement

- Pre-(Increment & Decrement)
- Post-(Increment & Decrement)





# Jenis Operator Increment dan Decrement

Operator	Nama	Penjelasan
<b>++\$a</b>	<b>Pre-Increment</b>	<b>menaikkan \$a sejumlah 1, lalu kembalikan</b>
<b>\$a++</b>	<b>Post-Increment</b>	<b>kembalikan \$a, lalu naikkan sejumlah 1</b>
<b>--\$a</b>	<b>Pre-Decrement</b>	<b>menurunkan \$a sejumlah 1, lalu kembalikan</b>
<b>\$a--</b>	<b>Post-Decrement</b>	<b>kembalikan \$a, lalu turunkan sejumlah 1</b>

- Pre-(Increment & Decrement) akan melakukan kenaikan dan penurunan nilai terlebih dahulu baru setelahnya nilai tersebut dikembalikan
- Post-(Increment & Decrement) akan mengembalikan nilai terlebih dahulu sebelum dilakukan operasi kenaikan dan penurunan nilai.

# Implementasi Operator Pre-Increment

● ● ● mardirabootcamp - PreIncrement.php

```
1  <?php
2
3  $number = 0;
4  ++$number;
5  ++$number;
6
7  echo "Hasil : {$number}";
```

hasil : 2

# Implementasi Operator Post-Increment

● ● ● mardirabootcamp - PostIncrement.php

```
1  <?php
2
3  $number = 0;
4  $number++;
5  $number++;
6
7  echo "Hasil : {$number}";
```

hasil : 2

# Implementasi Operator Pre-Decrement

● ● ● mardirabootcamp - PreDecrement.php

```
1  <?php
2
3  $number = 10;
4  --$number;
5  --$number;
6
7  echo "Hasil {$number}";
```

Hasil 8

# Implementasi Operator Post-Decrement

● ● ● mardirabootcamp - PostDecrement.php

```
1  <?php
2
3  $number = 10;
4  $number--;
5  $number--;
6
7  echo "Hasil : {$number}";
```

Hasil : 8

# Perbedaan Pre dan Post

Jika diterapkan kedua cara tersebut maka outputnya akan sama karena hanya disimpan dalam satu variable, perbedaan output akan muncul ketika membuat variable lebih dari satu dan saat menggunakan perulangan (looping).

mardirabootcamp - PreDecrementAlt.php

```
1 <?php
2
3 $number = 0;
4 $preIncrement = ++$number;
5
6 var_dump($number); # int(1)
7
8 var_dump($preIncrement); # int(1)
```

mardirabootcamp - PostIncrementAlt.php

```
1 <?php
2
3 $number = 0;
4 $postIncrement = $number++;
5
6 var_dump($number); # int(1)
7
8 var_dump($postIncrement); # int(0)
```

# Penjabaran Pre Increment

Penjelasan cara kerja dari pre increment berikut ini:

- variable \$number akan ditambah nilai satu terlebih dahulu
- kemudian disimpan ke dalam variable \$preIncrement

mardirabootcamp - PreDecrementAlt.php

```
1 <?php
2
3 $number = 0;
4 $preIncrement = ++$number;
5
6 var_dump($number); # int(1)
7
8 var_dump($preIncrement); # int(1)
```

mardirabootcamp - ExplanationPreIncrement.php

```
1 <?php
2
3 $x = 0;
4
5 $number = 0;
6 $number = $number + 1;
7 $preIncrement = $number;
8
9 var_dump($number); # int(1)
10
11 var_dump($preIncrement); # int(1)
```

# Penjabaran Post Increment

Penjelasan cara kerja dari post increment berikut ini:

- nilai dalam variable \$number akan disimpan ke variable \$postIncrement terlebih dahulu.
- Setelah itu nilai dari variable \$number ditambahkan 1

mardirabootcamp - PostIncrementAlt.php

```
1 <?php
2
3 $number = 0;
4 $postIncrement = $number++;
5
6 var_dump($number); # int(1)
7
8 var_dump($postIncrement); # int(0)
```

mardirabootcamp - ExplanationPostIncrement.php

```
1 <?php
2
3 $number = 0;
4 $postIncrement = $number;
5 $number = $number + 1;
6
7 var_dump($number); # int(1)
8
9 var_dump($postIncrement); # int(0)
```



# Implementasi Pre-Increment Looping

● ● ● mardirabootcamp - PreIncrementLooping.php

```
1  <?php
2
3  $number = 0;
4
5  for ($i = 0; $i < 10; ++$i) {
6      echo ++$number;
7  }
```

12345678910

# Implementasi Post-Increment Looping

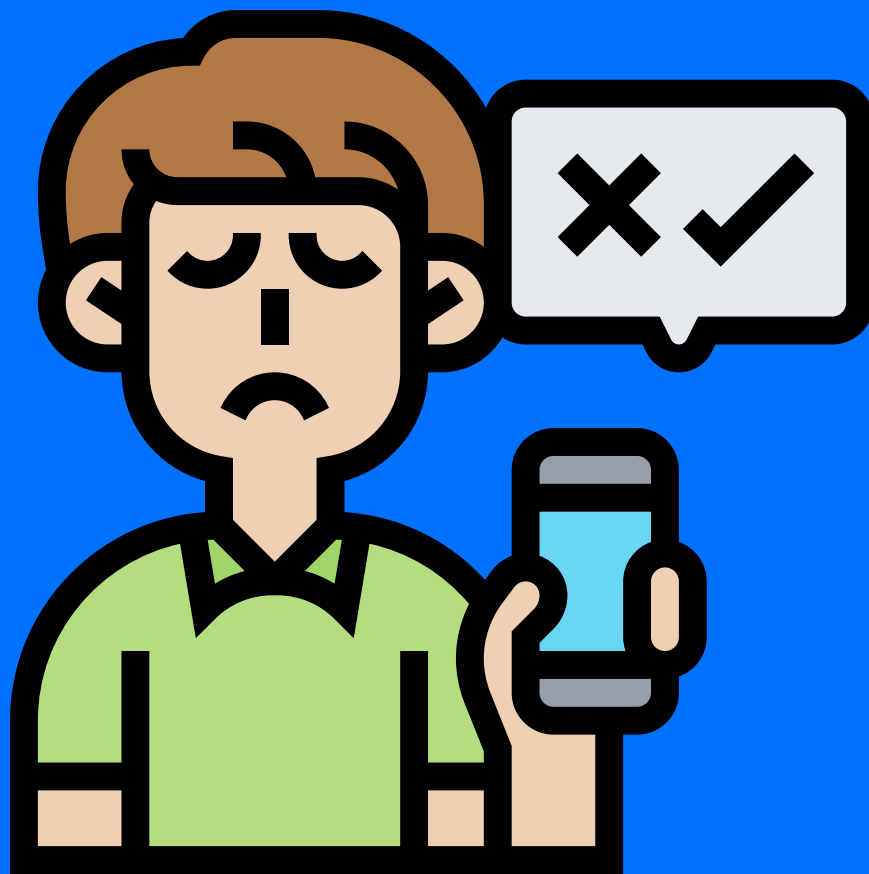
● ● ● mardirabootcamp - PostIncrementLooping.php

```
1  <?php
2
3  $number = 0;
4
5  for ($i = 0; $i < 10; $i++) {
6      echo $number++;
7  }
```

0123456789

## Operator Perbandingan (*Comparison Operator*)

Operator perbandingan digunakan untuk membandingkan dua buah nilai. Hasil yang dikembalikan berupa nilai bertipe boolean. Jika hasilnya true maka pernyataan perbandingan benar dan sebaliknya.



# Jenis Operator Perbandingan

Operator	Nama	Implementasi	Nilai True Jika
<	Kurang dari	$\$a < \$b$	$\$a$ kurang dari $\$b$
>	Lebih dari	$\$a > \$b$	$\$a$ lebih dari $\$b$
<=	Kurang dari sama dengan	$\$a \leq \$b$	$\$a$ kurang dari sama dengan $\$b$
>=	Lebih dari sama dengan	$\$a \geq \$b$	$\$a$ lebih dari sama dengan $\$b$

# Operator Kurang dari (*Less than*)

● ● ● mardirabootcamp - ComparisonLessThan.php

```
1  <?php
2
3  // benar
4  var_dump(0 < 1); # bool(true)
5
6  // salah
7  var_dump(1 < 0); # bool(false)
8  var_dump(1 < 1); # bool(false)
```

bool(true) bool(false) bool(false)

# Operator Lebih dari (*More than*)

● ● ● mardirabootcamp - ComparisonMoreThan.php

```
1  <?php
2
3  // benar
4  var_dump(1 > 0); # bool(true)
5
6  // salah
7  var_dump(0 > 1); # bool(false)
8  var_dump(1 > 1); # bool(false)
```

bool(true) bool(false) bool(false)

# Operator Kurang dari Sama Dengan (*Less than or equal to*)

● ● ● mardirabootcamp - ComparisonLessThanEqual.php

```
1  <?php
2
3  // true
4  var_dump(0 ≤ 1); # bool(true)
5  var_dump(1 ≤ 1); # bool(true)
6
7  // false
8  var_dump(1 ≤ 0); # bool(false)
```

bool(true) bool(true) bool(false)

# Operator Lebih dari Sama Dengan (*Greater than or equal to*)

● ● ● mardirabootcamp - ComparisonMoreThanEqual.php

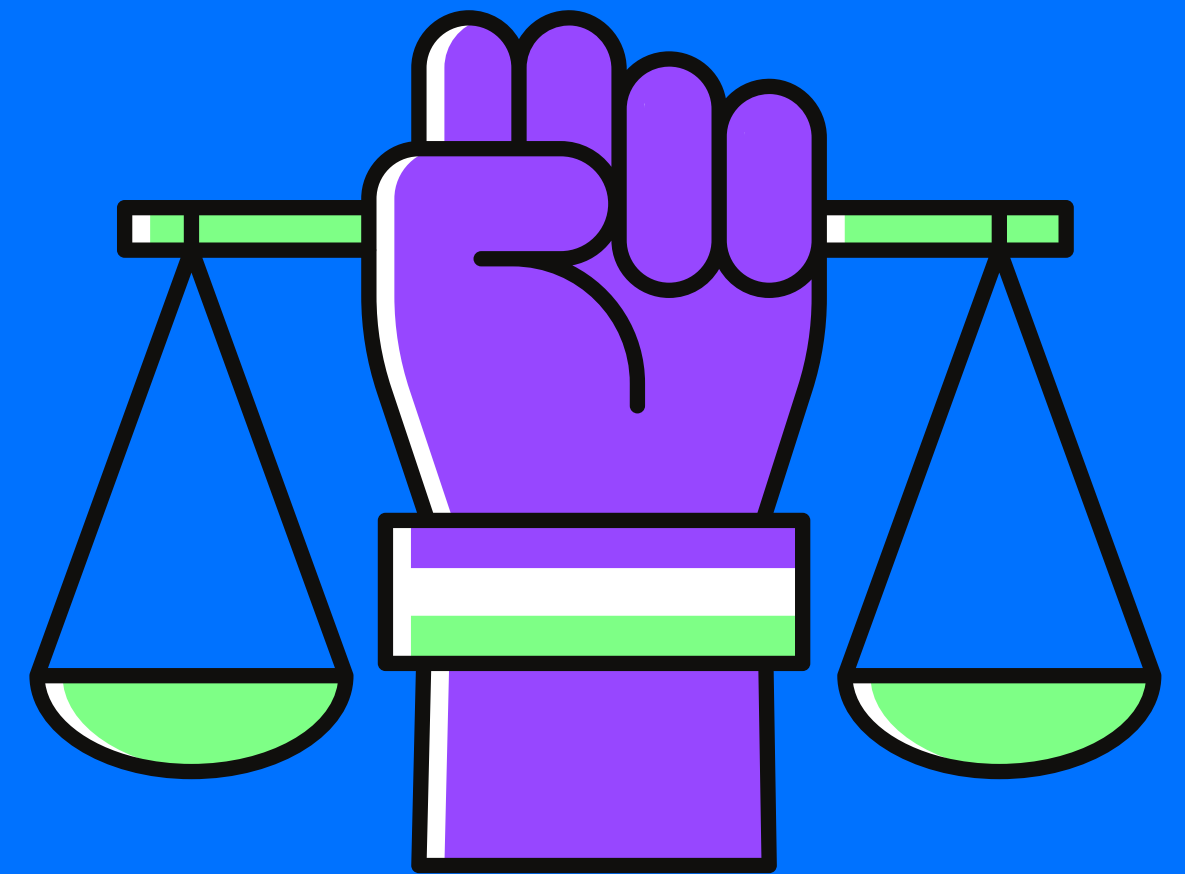
```
1  <?php
2
3  // benar
4  var_dump(1 ≥ 0); # bool(true)
5  var_dump(1 ≥ 1); # bool(true)
6
7  // salah
8  var_dump(0 ≥ 1); # bool(false)
```

bool(true) bool(true) bool(false)



# Perbandingan Equal dan Identical

Perbandingan equal dan identical akan membandingkan kesamaan dua buah nilai atau lebih. Perbedaan dari keduanya terletak pada pengecekan tipe datanya.



# Perbedaan Equal dan Identical

Perbandingan equal akan mengecek nilai dan mengabaikan tipe datanya, jadi ia akan menganggap bahwa nilai 123 dan "123" itu sama.

Sedangkan perbandingan identical akan mengecek nilai beserta tipe datanya, sehingga ia akan menganggap 123 dan "123" itu berbeda.

# Jenis Operator Equal dan Identical

Operator	Nama	Implementasi	Nilai True Jika
<b>==</b>	<b>Equal</b>	<b>\$a == \$b</b>	<b>\$a sama dengan \$b</b>
<b>===</b>	<b>Identical</b>	<b>\$a === \$b</b>	<b>\$a sama dengan \$b dan keduanya bertipe sama</b>
<b>!=</b>	<b>Not Equal</b>	<b>\$a != \$b</b>	<b>\$a tidak sama dengan \$b</b>
<b>&lt;&gt;</b>	<b>Not Equal</b>	<b>\$a &lt; &gt;</b>	<b>\$a tidak sama dengan \$b</b>
<b>!==</b>	<b>Not Identical</b>	<b>\$a !== \$b</b>	<b>\$a tidak sama dengan \$b atau keduanya berbeda tipe</b>

# Operator Sama Dengan (*Equal*)

```
mardirabootcamp - ComparisonEqual.php  
1  <?php  
2  
3  // benar  
4  var_dump(123 == 123); # bool(true)  
5  var_dump(123 == "123"); # bool(true)  
6  
7  // salah  
8  var_dump(123 == 456); # bool(false)
```

bool(true) bool(true) bool(false)

# Operator Sama Dengan Tipenya (*Identical*)

```
mardirabootcamp - ComparisonIdentical.php  
1  <?php  
2  
3  // benar  
4  var_dump(123 === 123); # bool(true)  
5  
6  // salah  
7  var_dump(123 === 456); # bool(false)  
8  var_dump(123 === "123"); # bool(false)
```

bool(true) bool(false) bool(false)

# Operator Tidak Sama Dengan (*Not Equal*) "!="

● ● ● mardirabootcamp - ComparisonNotEqual.php

```
1  <?php
2
3  // benar
4  var_dump(123 != 456); # bool(true)
5
6  // salah
7  var_dump(123 != 123); # bool(false)
8  var_dump(123 != "123"); # bool(false)
```

bool(true) bool(false) bool(false)

# Operator Tidak Sama Dengan (*Not Equal*) "<>"

mardirabootcamp - ComparisonNotEqualCross.php

```
1 <?php
2
3 // benar
4 var_dump(123 <> 456); # bool(true)
5
6 // salah
7 var_dump(123 <> 123); # bool(false)
8 var_dump(123 <> "123"); # bool(false)
```

bool(true) bool(false) bool(false)

# Operator Tidak Sama Dengan Tipenya (*Not Identical*)

● ● ● mardirabootcamp - ComparisonNotIdentical.php

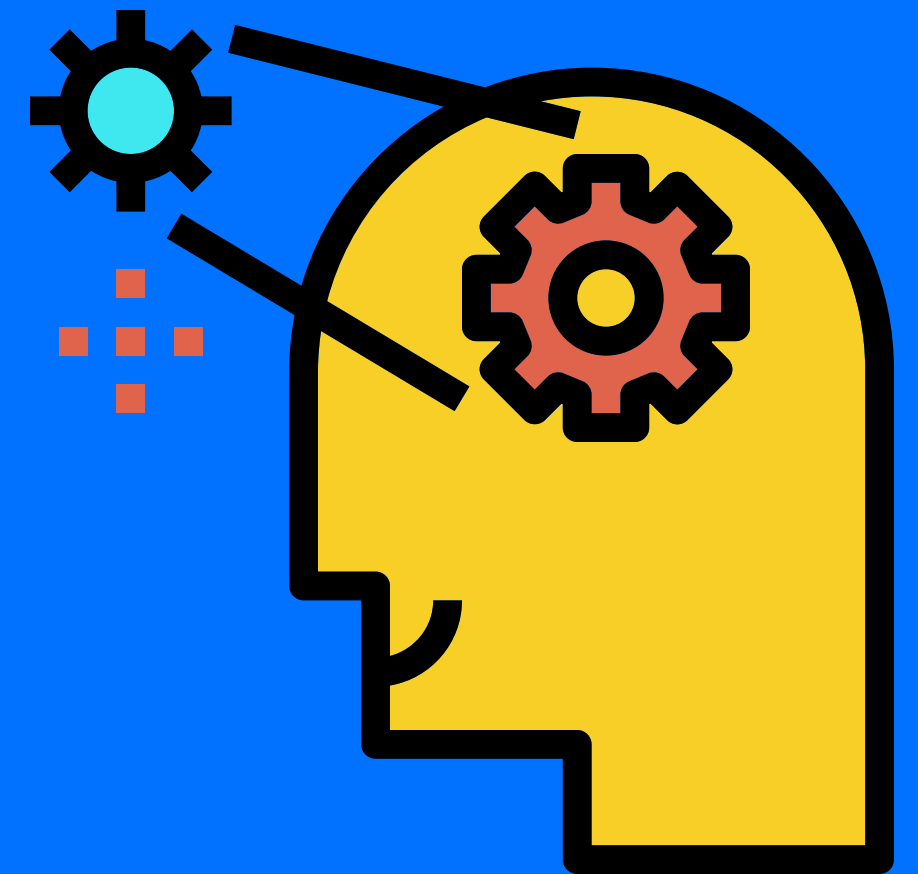
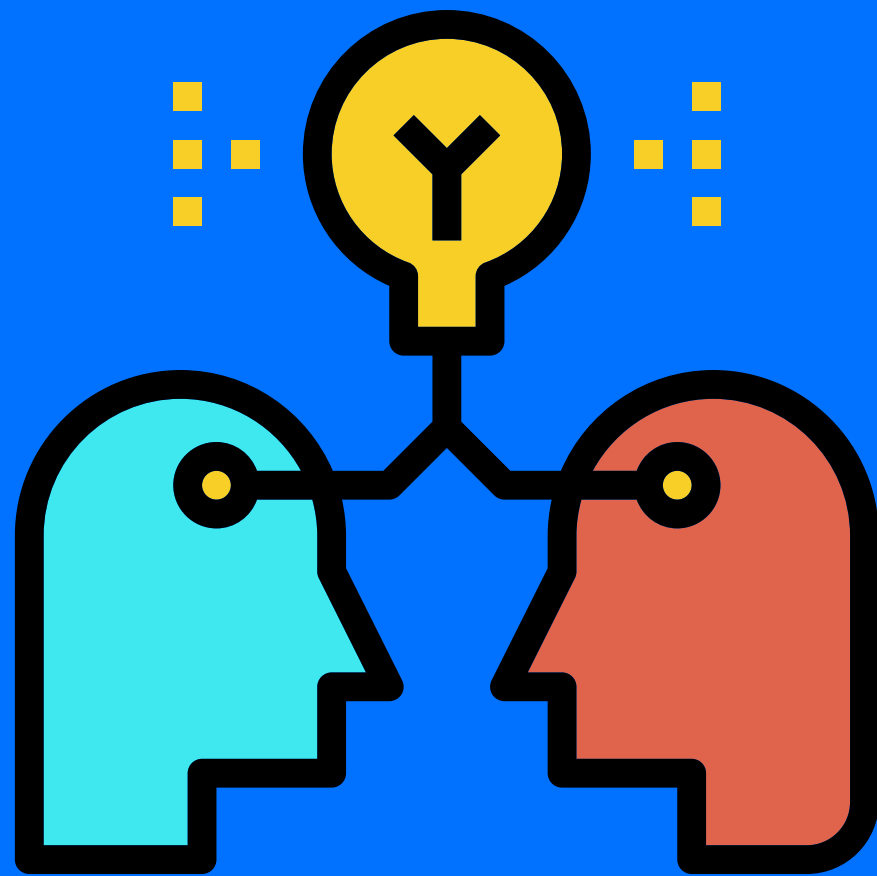
```
1  <?php
2
3  // benar
4  var_dump(123 !== 456); # bool(true)
5  var_dump(123 !== "123"); # bool(true)
6
7  // salah
8  var_dump(123 !== 123); # bool(false)
```

bool(true) bool(true) bool(false)



# Operator Logika (*Logical Operator*)

Operator logika umumnya digunakan untuk menggabungkan suatu pernyataan kondisional.



# Jenis Operator Logika

Operator	Nama	Implementasi	Nilai True Jika
AND	And	$\$a \text{ AND } \$b$	$\$a$ dan $\$b$ bernilai true
OR	OR	$\$a \text{ OR } \$b$	minimal salah satu dari $\$a$ atau $\$b$ bernilai true
XOR	xor (eXclusive OR)	$\$a \text{ XOR } \$b$	hanya salah satu dari $\$a$ atau $\$b$ bernilai true, tidak keduanya
&&	And	$\$a \text{ \&\& } \$b$	$\$a$ dan $\$b$ bernilai true
	OR	$\$a \text{    } \$b$	minimal salah satu dari $\$a$ atau $\$b$ bernilai true
!	Not	$!\$a$	nilai $\$a$ bernilai <i>false</i>

# Operator AND

```
● ● ● mardirabootcamp - LogicalAnd.php  
1  <?php  
2  
3  // benar  
4  var_dump(true and true); # bool(true)  
5  
6  // salah  
7  var_dump(true and false); # bool(false)
```

bool(true) bool(false)

# Operator OR

● ● ● mardirabootcamp - LogicalOR.php

```
1  <?php
2
3  // benar
4  var_dump(true or true); # bool(true)
5  var_dump(true or false); # bool(true)
6
7  // salah
8  var_dump(false or false); # bool(false)
```

bool(true) bool(true) bool(false)

# Operator XOR

● ● ● mardirabootcamp - LogicalXOR.php

```
1  <?php
2
3  // benar
4  var_dump(true xor false); # bool(true)
5
6  // salah
7  var_dump(true xor true); # bool(false)
8  var_dump(false xor false); # bool(false)
```

bool(true) bool(false) bool(false)

# Operator &&

```
● ● ● mardirabootcamp - LogicalAndSymbol.php
1  <?php
2
3  // benar
4  var_dump(true && true); # bool(true)
5
6  // salah
7  var_dump(true && false); # bool(false)
```

bool(true) bool(false)

# Operator ||

● ● ● mardirabootcamp - LogicalORSymbol.php

```
1  `<?php
2
3  // benar
4  var_dump(true || true); # bool(true)
5  var_dump(true || false); # bool(true)
6
7  // salah
8  var_dump(false || false); # bool(false)`
```

bool(true) bool(true) bool(false)

# Operator Not (!)

● ● ● mardirabootcamp - LogicalNot.php

```
1  <?php
2
3  // benar
4  var_dump(!false); # bool(true)
5
6  // salah
7  var_dump(!true); # bool(false)
```

Operator not (!) bisa dikatakan bentuk lain dari operator negasi (-) versi logika. Operator ini akan membalikkan nilai true menjadi false dan sebaliknya false menjadi true.

`bool(true)` `bool(false)`



**Terimakasih**