

# Pemrograman Web

2 SKS | Semester 5 | UNIKOM

*Nizar Rabbi Radliya*  
*nizar@email.unikom.ac.id*

Lesson 4

PHP Loops





## PHP Increment / Decrement Operators

The PHP increment operators are used to increment a variable's value.  
The PHP decrement operators are used to decrement a variable's value.

| Operator           | Name           | Description  |
|--------------------|----------------|--|
| <code>++\$x</code> | Pre-increment  | Increments <code>\$x</code> by one, then returns <code>\$x</code>  |
| <code>\$x++</code> | Post-increment | Returns <code>\$x</code> , then increments <code>\$x</code> by one |
| <code>--\$x</code> | Pre-decrement  | Decrements <code>\$x</code> by one, then returns <code>\$x</code>  |
| <code>\$x--</code> | Post-decrement | Returns <code>\$x</code> , then decrements <code>\$x</code> by one |



## Syntax Increment / Decrement Operators

```
<?php
$x = 10;
echo ++$x; //result 11
echo "<br/>";
$x = 10;
echo $x++; //result 10
?>
```





Often when you write code, you want the same block of code to run over and over again in a row. Instead of adding several almost equal code-lines in a script, we can use loops to perform a task like this.

In PHP, we have the following looping statements:

- for - loops through a block of code a specified number of times
- while - loops through a block of code as long as the specified condition is true
- do...while - loops through a block of code once, and then repeats the loop as long as the specified condition is true





## PHP for Loop

- PHP for loops execute a block of code a specified number of times.
- The for loop is used when you know in advance how many times the script should run.
- *Syntax:*

```
for (init counter; test counter; increment counter) {  
    code to be executed;  
}
```

Parameters:

- **init counter:** Initialize the loop counter value
- **test counter:** Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.
- **increment counter:** Increases the loop counter value



# Syntax for Loop

```
<?php  
for ($x = 1; $x <= 5; $x++) {  
    echo "The number is: $x <br>";  
}  
?>
```



The while loop executes a block of code as long as the specified condition is true.

Syntax:

```
while (condition is true) {  
    code to be executed;  
}
```





# Syntax while Loop

The example below first sets a variable `$x` to 1 (`$x = 1`). Then, the while loop will continue to run as long as `$x` is less than, or equal to 5 (`$x <= 5`). `$x` will increase by 1 each time the loop runs (`$x++`):

```
<?php
$x = 1;

while($x <= 5) {
    echo "The number is: $x <br>";
    $x++;
}
?>
```





## PHP do...while Loop

The do...while loop will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.

Syntax:

```
do {  
    code to be executed;  
} while (condition is true);
```





## Syntax do...while Loop

The example below first sets a variable `$x` to 1 (`$x = 1`). Then, the do while loop will write some output, and then increment the variable `$x` with 1. Then the condition is checked (is `$x` less than, or equal to 5?), and the loop will continue to run as long as `$x` is less than, or equal to 5:

```
<?php
$x = 1;

do {
    echo "The number is: $x <br>";
    $x++;
} while ($x <= 5);
?>
```





## Syntax do...while Loop

Notice that in a do while loop the condition is tested **AFTER** executing the statements within the loop. This means that the do while loop would execute its statements at least once, even if the condition is false the first time.

The example below sets the \$x variable to 6, then it runs the loop, and then the condition is checked:

```
<?php
$x = 6;

do {
    echo "The number is: $x <br>";
    $x++;
} while ($x <= 5);
?>
```





# Syntax Nested Loop

```
<?php
```

```
for ($x = 1; $x <= 5; $x++) {  
    echo "Row: $x Column: ";  
    for ($y = 1; $y <= 5; $y++) {  
        echo "$y ";  
    }  
    echo "<br>";  
}  
?>
```





## Case 1: IS-1 – PHP Loops

Mencetak bilangan ganjil diantara 10 s/d 50.

Contoh outputnya :

11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49

---

Mencetak bilangan ganjil diantara 10 s/d 50, lalu hitung ada berapa

Contoh outputnya :

11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49

Jumlah bilangan ada : 20

---

Mencetak bilangan genap bukan kelipatan 4 diantara 10 s/d 50.

Contoh outputnya : 10 14 18 22 26 30 34 38 42 46 50





## Case 2: IS-1 – PHP Loops

Tampilkan menggunakan nested loop:

```
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

Tampilkan menggunakan php nested loops dan php conditionals:

```
* * * * *  
* * * *  
* * *  
* *  
*
```



## Case 1: IS-2 – PHP Loops

Mencetak bilangan genap diantara 10 s/d 50.

Contoh outputnya :

10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

---

Mencetak bilangan genap diantara 10 s/d 50, lalu hitung ada berapa

Contoh outputnya :

10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

Jumlah bilangan ada : 21

---

Mencetak bilangan ganjil bukan kelipatan 3 diantara 10 s/d 50.

Contoh outputnya : 11 13 17 19 23 25 29 31 33 35 37 41 43 47 49



## Case 2: IS-2 – PHP Loops

Tampilkan menggunakan nested loop:

```
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

Tampilkan menggunakan php nested loops dan php conditionals:

```
*  
* *  
* * *  
* * * *  
* * * * *
```





## Task – PHP Loops

Mencetak bilangan kelipatan 3 diantara 1 s/d 50.

Contoh outputnya :

3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48

---

Mencetak bilangan kelipatan 3 diantara 1 s/d 50, lalu hitung ada berapa

Contoh outputnya :

3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48

Jumlah bilangan ada : 16

---

Mencetak bilangan genap bukan kelipatan 4 diantara 10 s/d 50.

Contoh outputnya : 10 14 18 22 26 30 34 38 42 46 50



## Task – PHP Loops

Tampilkan menggunakan nested loop:

```
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

Tampilkan menggunakan php nested loops dan php conditionals:

```
* * * * *  
* * * * -  
* * * - -  
* * - - -  
* - - - -
```



## Task – PHP Loops

Mencetak bilangan kelipatan 4 diantara 11 s/d 50.

Contoh outputnya :

12 16 20 24 28 32 36 40 44 48

---

Mencetak bilangan genap diantara 10 s/d 50, lalu hitung ada berapa

Contoh outputnya :

12 16 20 24 28 32 36 40 44 48

Jumlah bilangan ada : 10

---

Mencetak bilangan ganjil bukan kelipatan 3 diantara 10 s/d 50.

Contoh outputnya : 11 13 17 19 23 25 29 31 33 35 37 41 43 47 49



# Task – PHP Loops

Tampilkan menggunakan nested loop:

```
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

Tampilkan menggunakan php nested loops dan php conditionals:

```
* - - - -  
* * - - -  
* * * - -  
* * * * -  
* * * * *
```

**NEXT:**

**PHP Forms**

