

MODULE 2 : KONTROL KONDISI

CPMK :

Mahasiswa mengerti dan mampu menjelaskan, mengimplementasikan Operator-Operator yang ada pada PHP, Struktur Kendali, Looping dan Kondisi percabangan dengan Hypertext Preprocessor (PHP)

2.1 Materi Pengantar

Pada praktikum kali ini, anda akan mempelajari kondisi dan perulangan pada PHP. Tetapi, sebelumnya anda harus tahu operator-operator yang ada pada PHP seperti operator aritmatika, assignment, perbandingan, logika dan sebagainya.

Operator adalah suatu simbol yang dipakai untuk memanipulasi nilai suatu variabel. Variabel yang nilainya dimodifikasi oleh operator disebut operand. Misalnya 3-2, 3 dan 2 adalah operand dan - adalah operator. Dalam mengeksekusi suatu perintah ada urutan operator yang akan dilakukan oleh PHP. Urutan operator akan menentukan hasil akhir dari sebuah eksekusi. Contoh: Perhitungan $5+2*3$ akan menghasilkan nilai 11 ($2*3$ akan dihitung terlebih dahulu kemudian ditambahkan dengan 5). Hal ini karena operator * memiliki prioritas lebih tinggi daripada operator +. Urutan operator dalam PHP dari urutan tertinggi sampai terendah dapat dilihat dibawah ini:

| Prioritas | Operator |
|-----------|--|
| Tertinggi | () {} ` ! ++ -- \$ & * / % + - < > <= >= == != & ^ ! && = += -= *= /= &= = ^= .= AND (&&) XOR () |
| Terendah | OR |

a Operator Aritmatika

| Operator | Name | Example | Result |
|----------|------|---------|--------|
|----------|------|---------|--------|

| | | | |
|----|----------------|--------------|--|
| + | Addition | $\$x + \y | Sum of $\$x$ and $\$y$ |
| - | Subtraction | $\$x - \y | Difference of $\$x$ and $\$y$ |
| * | Multiplication | $\$x * \y | Product of $\$x$ and $\$y$ |
| / | Division | $\$x / \y | Quotient of $\$x$ and $\$y$ |
| % | Modulus | $\$x \% \y | Remainder of $\$x$ divided by $\$y$ |
| ** | Exponentiation | $\$x ** \y | Result of raising $\$x$ to the $\$y$ 'th power (Introduced in PHP 5.6) |

b PHP Assignment Operators

| Assignment | Same as... | Description |
|------------|--------------|---|
| $x = y$ | $x = y$ | The left operand gets set to the value of the expression on the right |
| $x += y$ | $x = x + y$ | Addition |
| $x -= y$ | $x = x - y$ | Subtraction |
| $x *= y$ | $x = x * y$ | Multiplication |
| $x /= y$ | $x = x / y$ | Division |
| $x \% = y$ | $x = x \% y$ | Modulus |

c Operator Perbandingan

| Operator | Name | Example | Result |
|----------|--------------------------|---------------|--|
| == | Equal | $\$x == \y | Returns true if $\$x$ is equal to $\$y$ |
| === | Identical | $\$x === \y | Returns true if $\$x$ is equal to $\$y$, and they are of the same type |
| != | Not equal | $\$x != \y | Returns true if $\$x$ is not equal to $\$y$ |
| <> | Not equal | $\$x <> \y | Returns true if $\$x$ is not equal to $\$y$ |
| !== | Not identical | $\$x !== \y | Returns true if $\$x$ is not equal to $\$y$, or they are not of the same type |
| > | Greater than | $\$x > \y | Returns true if $\$x$ is greater than $\$y$ |
| < | Less than | $\$x < \y | Returns true if $\$x$ is less than $\$y$ |
| >= | Greater than or equal to | $\$x >= \y | Returns true if $\$x$ is greater than or equal to $\$y$ |

| | | | |
|--------------------|-----------------------|----------------------------|--|
| <code><=</code> | Less than or equal to | <code>\$x <= \$y</code> | Returns true if \$x is less than or equal to \$y |
|--------------------|-----------------------|----------------------------|--|

d Operator Increment / Decrement

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

e Operator Logika

| Operator | Name | Example | Result |
|-------------------------|------|---------------------------------|---|
| <code>and</code> | And | <code>\$x and \$y</code> | True if both \$x and \$y are true |
| <code>or</code> | Or | <code>\$x or \$y</code> | True if either \$x or \$y is true |
| <code>xor</code> | Xor | <code>\$x xor \$y</code> | True if either \$x or \$y is true, but not both |
| <code>&&</code> | And | <code>\$x && \$y</code> | True if both \$x and \$y are true |
| <code> </code> | Or | <code>\$x \$y</code> | True if either \$x or \$y is true |
| <code>!</code> | Not | <code>!\$x</code> | True if \$x is not true |

f Operator String

| Operator | Name | Example | Result |
|-----------------|--------------------------|-------------------------------|------------------------------------|
| <code>.</code> | Concatenation | <code>\$txt1 . \$txt2</code> | Concatenation of \$txt1 and \$txt2 |
| <code>.=</code> | Concatenation assignment | <code>\$txt1 .= \$txt2</code> | Appends \$txt2 to \$txt1 |

2.2 Tugas Pendahuluan

1. Percabangan sering disebut juga dengan decision-making, memungkinkan aplikasi untuk memeriksa isi suatu variabel atau hasil perhitungan ekspresi dan mengambil tindakan yang sesuai. Sebutkan bentuk bentuk percabangan yang disediakan dalam PHP !
2. Pernyataan perulangan berguna untuk melakukan pengulangan statemen/kode program yang dieksekusi oleh webserver lebih dari sekali sesuai dengan kondisi yang diisyaratkan. Sebutkan tiga macam pengulangan dalam PHP!
3. Dalam perulangan dapat dilakukan lompatan. Ada beberapa macam fungsi untuk melewati proses yang sedang berjalan yaitu BREAK, CONTINUE dan EXIT, Jelaskan fungsi dari masing-masing fungsi tersebut !
4. Analisis perbedaan for, while, dan foreach; berikan kasus terbaik untuk tiap struktur !
5. Buat algoritma sederhana untuk menentukan bilangan prima dengan looping !