

PRAKTIKUM PEMROGRAMAN WEB

CONTOH PERHITUNGAN OPERATOR



Nama : Muhammad Irfan Pratama
NIM : M3119059
KELAS : TI C 19

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH VOKASI
UNIVERSITAS SEBELAS MARET
SURAKARTA
2020

Arithmetic

Arithmetic Operator adalah operator (tanda) yang digunakan untuk melakukan perhitungan operasi aritmatika, berikut ini operatornya:

	Operator	Penggunaan di Java
Penjumlahan	+	$x + 7$
Pengurangan	-	$x - 7$
Perkalian	*	$x * 7$
Pembagian	/	$x / 7$
Sisa bagi	%	$x \% 7$

Level Prioritas Arithmetic Operator

Ketika ada operasi aritmatik yang di dalamnya terdapat beberapa operator maka cara untuk menghitungnya adalah mendahulukan arithmetic operator yang memiliki prioritas yang paling tinggi. Berikut ini urutan prioritas arithmetic operator:

Level	Operator	Asosiasi
1	$*$ $/$ $\%$	kiri ke kanan
2	$+$ $-$	kiri ke kanan
3	$=$	kanan ke kiri

Contoh:

$$d = a + b * c$$

Karena “*” level nya lebih tinggi daripada “+” maka “b * c” dihitung terlebih dahulu berikutnya hasil dari “b * c” ditambahkan dengan “a”. Hasil dari semuanya disimpan di variable “d”.

Bagaimana kalau operator di dalam operasi aritmatik tersebut memiliki level yang sama.

Contoh: $d = a / b * c$, maka dilihat asosiasinya. “/” dan “*” berada pada level yang sama maka yang dihitung terlebih dahulu adalah tergantung pada asosiasinya. Asosiasi “/” dan “*” dari kiri kanan maka “a / b” dihitung terlebih dahulu berikutnya hasil dari “a / b” dikalikan dengan c. Hasilnya disimpan di variable d.

Assignment

Assignment Operator (operator penugasan) adalah operator yang menggunakan tanda sama dengan (=) untuk mengisi sebuah nilai dalam suatu variabel. Operator Assignment digunakan untuk menyatakan suatu perintah atau fungsi perhitungan untuk memberikan nilai pada variabel dalam bentuk yang disingkat.

Operator penugasan terdiri dari :

Operator	Nama Operator	Contoh	Penjelasan
=	<u>Pengisian Nilai</u>	A = B	A = B
+=	<u>Pengisian dan penjumlahan</u>	A += B	A = A + B
-=	<u>Pengisian dan pengurangan</u>	A -= B	A = A - B
*=	<u>Pengisian dan perkalian</u>	A *= B	A = A * B
/=	<u>Pengisian dan pembagian</u>	A /= B	A = A / B
%=	<u>Pengisian dan sisa bagi</u>	A %= B	A = A % B
**=	<u>Pengisian dan pemangkatan</u>	A **= B	A = A ** B

Sebenarnya, operator penugasan sama seperti operator aritmatika dan juga digunakan untuk melakukan operasi aritmatika.

Contoh : `var jumlahList = 14;`

jumlahList += 1; //menggunakan operator penugasan penjumlahan untuk menambah nilai

Hasilnya : Variabel jumlahList akan bertambah satu. Maksud dari jumlahList += 1; adalah:

jumlahList = jumlahList + 1;

Logical

Operator yang digunakan untuk menemukan jawaban dari suatu logika ekspresi, jawaban tersebut bernilai boolean, yaitu hanya memiliki nilai benar (true) dan salah (false). Logical operator yang dibahas disini, diantaranya: (1) conditional AND (2) conditional OR (4) exclusive OR dan (5) logical NOT.

Conditional AND

Conditional AND ini menggunakan simbol “&&”. Seperti logika AND di pelajaran matematika, conditional AND ini juga memiliki table kebenaran, yaitu:

Ekspresi 1	Ekspresi 2	Expresi 1 && Ekspresi 2
false	false	false
false	true	false
true	false	false
true	true	true

Tabel kebenaran diatas menunjukkan bahwa conditional AND hanya menghasilkan true ketika semua ekspresinya bernilai true.

Conditional OR

Conditional OR ini menggunakan simbol “||”. Seperti logika OR di pelajaran matematika, conditional OR ini juga memiliki table kebenaran, yaitu:

Ekspresi 1	Ekspresi 2	Expresi 1 Ekspresi 2
false	false	false
false	true	true
true	false	true
true	true	true

Tabel kebenaran diatas menunjukkan bahwa conditional OR hanya menghasilkan false ketika semua ekspresinya bernilai false.

Exclusive OR

Exclusive OR ini menggunakan simbol “^”. Tabel kebenaran exclusive OR sebagai berikut:

Ekspresi 1	Ekspresi 2	Expresi 1 ^ Ekspresi 2
false	false	false
false	true	true
true	false	true
true	true	false

Tabel kebenaran diatas menunjukkan bahwa exclusive OR hanya menghasilkan true ketika ada salah satu dari ekspresinya bernilai true.

Logical NOT

Logical NOT menggunakan simbol “!” adalah kebalikan dari suatu nilai boolean. Tabel kebenaran untuk logical NOT sebagai berikut:

Ekspresi	!Ekspresi
false	true
true	false

Penerapan Logical

Operator Ekspresi yang dituliskan di tabel kebenaran bisa menggunakan equality dan relational operator. Berikut ini ilustrasinya:

```

1  public class Logical {
2      public static void main(String [] args){
3          int a = 4;
4          int b = 3;
5          boolean c1;
6          boolean c2;
7
8          c1 = a == b; // false
9          c2 = a > b; // true
10
11         System.out.println("c1 && c2 = "
12         + (c1 && c2) ); // c1 && c2 = false
13         System.out.println("c1 || c2 = "
14         + (c1 || c2) ); // c1 || c2 = true
15         System.out.println("c1 ^ c2 = "
16         + (c1 ^ c2) ); // c1 ^ c2 = false
17         System.out.println("!c1 = "
18         + (!c1) ); // !c1 = true
19     }

```

Baris 8 dan 9 menunjukkan suatu ekspresi yang masing-masing menggunakan equality dan relational operator. Sedangkan baris 11 – 18 merupakan contoh dari penerapan logical operator di Java.

Comparison

Operator ini membandingkan nilai di kedua sisi operan dan menentukan relasi di antara keduanya. Hal ini juga disebut sebagai operator relasional. Operator perbandingan berbagai adalah (=, !=, <, >, <=, dll)

Contoh: untuk operator perbandingan kita akan membandingkan nilai x dengan nilai y dan mencetak hasilnya apakah True atau false. Di sini, sebagaimana contoh, nilai x = 4 lebih kecil dari y = 5, jadi ketika kita mencetak nilai sebagai x > y, itu sebenarnya membandingkan nilai x ke y dan karena itu tidak benar, mengembalikan false.

```

x = 4
y = 5
print(('x > y is',x>y))

```

Demikian juga, Anda dapat mencoba operator perbandingan lainnya (x < y, x = y, x != y, dll.)