

PRAKTIKUM 3

OPERATOR

A. TUJUAN PEMBELAJARAN

1. Mengetahui berbagai macam bentuk operator
2. Memahami penggunaan berbagai macam jenis operator yang ada di Java

B. DASAR TEORI

Operator dapat diklasifikasikan menjadi 2 bentuk, yaitu unary operator dan binary operator. Unary operator adalah operator yang hanya melibatkan 1 operan. Sedangkan binary operator adalah operator yang melibatkan 2 operan. Java mempunyai berbagai macam jenis operator yang dapat digolongkan menjadi operator aritmatika, increment-decrement, bitwise, boolean, logik, shift (geser), penugasan, kombinasi dan kondisi.

Arithmetic operator (operator aritmatika) adalah operator yang berfungsi untuk operasi aritmatika. Yang termasuk dalam arithmetic operator bisa dilihat pada Tabel 3.1.

Tabel 3.1. Arithmetic operator

Arithmetic Operator	Keterangan
+	Operasi penambahan
-	Operasi pengurangan
*	Operasi perkalian
/	Operasi pembagian
%	Operasi modulus

Increment – decrement operator adalah operator yang berguna untuk menaikkan 1 nilai (increment) dan menurunkan 1 nilai (decrement). Yang termasuk increment-

decrement operator bisa dilihat pada Tabel 3.2. Berdasarkan urutan eksekusi penaikan dan penurunan nilainya, increment-decrement operator ini dapat diklasifikasikan menjadi 2 macam, yaitu pre-increment/decrement dan post-increment/decrement.

Tabel 3.2. Arithmetic operator

Increment-Decrement Operator	Keterangan
++	increment
--	decrement

Bitwise operator adalah operator yang dipakai untuk operasi bit pada nilai operan. Yang termasuk bitwise operator bisa dilihat pada Tabel 3.3.

Tabel 3.3. Bitwise Operator

Bitwise Operator	Keterangan
~	Operasi complement
&	Operasi AND
	Operasi OR
^	Operasi XOR

Boolean operator (operator boolean) adalah operator yang mengharuskan operannya bertipe boolean (true atau false). Yang termasuk boolean operator bisa dilihat pada Tabel 3.4.

Tabel 3.4. Boolean Operator

Logical Operator	Keterangan
!	Operasi negasi (NOT)
&	Operasi AND
	Operasi OR
^	Operasi XOR
&&	Operasi AND (short circuit)

	Operasi OR (short circuit)
--	----------------------------

Operator `!`, `&`, `|` dan `^` mempunyai implementasi yang sama sebagaimana ketika ia menjadi bitwise operator. Hanya saja di logical operator, operan yang dilibatkan disini harus bertipe boolean, yang hanya mempunyai nilai true atau false.

Logical operator (operator logika) adalah operator yang sering dipakai untuk operasi perbandingan dan selalu menghasilkan suatu nilai bertipe boolean (true atau false). Yang termasuk logical operator bisa dilihat pada Tabel 3.5.

Tabel 3.5. Logical Operator

Logical Operator	Keterangan
<code>==</code>	Operasi perbandingan sama dengan
<code>!=</code>	Operasi perbandingan tidak sama dengan
<code>></code>	Operasi perbandingan lebih besar
<code>>=</code>	Operasi perbandingan lebih besar sama dengan
<code><</code>	Operasi perbandingan lebih kecil
<code><=</code>	Operasi perbandingan lebih kecil sama dengan

Shift operator (operator geser) adalah operator yang berfungsi untuk menggeser susunan bit pada suatu nilai. Yang termasuk dalam shift operator dapat dilihat pada Tabel 3.6.

Tabel 3.6. Shift Operator

Shift Operator	Keterangan
<code>>></code>	right shift
<code>>>></code>	unsigned right shift
<code><<</code>	left shift

Combination operator (operator kombinasi) adalah operator yang terdiri dari gabungan 2 operator. Biasanya combination operator ini dipakai untuk mempersingkat waktu penulisan program. Yang termasuk operator combination bisa dilihat pada Tabel

3.7.

Tabel 3.7. Combination Operator

Combination Operator	Keterangan
<code>+=</code>	Gabungan dari operator <code>=</code> dan <code>+</code>
<code>-=</code>	Gabungan dari operator <code>=</code> dan <code>-</code>
<code>*=</code>	Gabungan dari operator <code>=</code> dan <code>*</code>
<code>/=</code>	Gabungan dari operator <code>=</code> dan <code>/</code>
<code>%=</code>	Gabungan dari operator <code>=</code> dan <code>%</code>
<code>>>=</code>	Gabungan dari operator <code>=</code> dan <code>>></code>
<code>>>>=</code>	Gabungan dari operator <code>=</code> dan <code>>>></code>
<code><<=</code>	Gabungan dari operator <code>=</code> dan <code><<</code>
<code>&=</code>	Gabungan dari operator <code>=</code> dan <code>&</code>
<code> =</code>	Gabungan dari operator <code>=</code> dan <code> </code>
<code>^=</code>	Gabungan dari operator <code>=</code> dan <code>^</code>

Conditional operator (operator kondisional) adalah operator yang dipakai untuk operasi kondisi (persyaratan), sama sebagaimana if-then-else dan hanya berlaku untuk pernyataan tunggal. Operator ini mengembalikan suatu nilai yang benar sesuai dengan kondisi yang diberikan. Conditional operator (operator kondisional) ini hanya ada 1 macam, yaitu `?` disertai dengan tanda `:` (titik dua). Jika kondisi persyaratan yang terletak di sebelah kiri tanda `?` bernilai benar, maka pernyataan yang berada di sebelah kiri tanda `:` yang akan diambil. Demikian juga sebaliknya, jika kondisi persyaratan bernilai salah, maka pernyataan yang berada di sebelah kanan tanda `:` yang akan diambil.

C. TUGAS PENDAHULUAN

Buatlah makalah yang berisi tentang berbagai macam operator dengan disertai contoh penggunaan dan outputnya.

D. PERCOBAAN

Percobaan 1 : Melakukan increment dan decrement nilai

```
class IncDec {
    public static void main (String args[]) {
        int x = 8, y = 13;
        System.out.println("x = " + x);
        System.out.println("y = " + y);
        System.out.println("x = " + ++x);
        System.out.println("y = " + y++);
        System.out.println("x = " + x--);
        System.out.println("y = " + --y);
    }
}
```

Percobaan 2 : Melakukan operasi bit

```
class Bitwise {
    public static void main (String args[]) {
        int x = 5, y = 6;
        System.out.println("x = " + x);
        System.out.println("y = " + y);
        System.out.println("x & y = " + (x & y));
        System.out.println("x | y = " + (x | y));
        System.out.println("x ^ y = " + (x ^ y));
    }
}
```

Percobaan 3 : Melakukan operasi komplemen

```
class BitwiseComplement {
    public static void main (String args[]) {
        int x = 8;
        System.out.println("x = " + x);
        int y = ~x;
        System.out.println("y = " + y);
    }
}
```

Percobaan 4 : Melakukan operasi shift

```
class Shift {
    public static void main (String args[]) {
        int x = 7;
        System.out.println("x = " + x);
        System.out.println("x >> 2 = " + (x >> 2));
        System.out.println("x << 1 = " + (x << 1));
        System.out.println("x >>> 1 = " + (x >>> 1));
    }
}
```

Percobaan 5 : Menggunakan logical operator

```
class LogicalOperator {
    public static void main (String args[]) {
        int x = 7, y = 11, z = 11;
        System.out.println("x = " + x);
        System.out.println("y = " + y);
        System.out.println("z = " + z);
        System.out.println("x < y = " + (x < y));
        System.out.println("x > z = " + (x > z));
        System.out.println("y <= z = " + (y <= z));
        System.out.println("x >= y = " + (x >= y));
        System.out.println("y == z = " + (y == z));
        System.out.println("x != y = " + (x != z));
    }
}
```

Percobaan 6 : Menggunakan operator boolean and

```
public class BooleanAnd {
    public static void main(String args[]) {
        int a=5, b=7;
        if ((a<2) & (b++<10)) b+=2;
        System.out.println(b);
    }
}
```

Percobaan 7 : Menggunakan operator boolean and short-circuit

```
public class ShortCircuitBooleanAnd {
    public static void main(String args[]) {
        int a=5, b=7;
        if ((a<2) && (b++<10)) b+=2;
        System.out.println(b);
    }
}
```

Percobaan 8 : Menggunakan boolean or

```
public class BooleanOr {
    public static void main(String args[]) {
        int a=5, b=7;
        if ((a>2) | (b++<10)) b+=2;
        System.out.println(b);
    }
}
```

Percobaan 9 : Menggunakan boolean or short-circuit

```
public class ShortCircuitBooleanOr {
    public static void main(String args[]) {
        int a=5, b=7;
        if ((a>2) || (b++<10)) b+=2;
        System.out.println(b);
    }
}
```

Percobaan 10 : Menggunakan operator kondisi

```
class Conditional {
    public static void main (String args[]) {
        int x = 0;
        boolean isEven = false;
        System.out.println("x = " + x);
        x = isEven ? 4 : 7;
        System.out.println("x = " + x);
    }
}
```


Jelaskan apa yang terjadi pada potongan program berikut ini:

```
byte a=-1;  
a=(byte) (a >>> 2);
```

LAPORAN RESMI

Kumpulkan hasil percobaan di atas dan tambahkan analisa untuk tiap percobaan, latihan, dan tugas yang telah dibuat.