

BAB III OPERATOR

Operator Java merupakan karakter khusus yang berupa simbol atau tanda yang memerintahkan *compiler* untuk melakukan berbagai *operasi* terhadap sejumlah *operand*. Perintah operasi dispesifikasikan oleh operator, dimana operand-nya adalah variabel, pernyataan, atau besaran literal.

Operator yang dibahas pada bagian ini adalah operator aritmatika, increment dan decrement, assignment (penugasan), relasi, logical, dan bitwise.

A. Operator Aritmatika

Sama halnya dengan semua bahasa pemrograman, Java menyediakan operator-operator aritmatika untuk manipulasi variabel data numerik. Operator-operator tersebut antara lain :

Operator	Penggunaan	Deskripsi
+	Op1 + Op2	Menambahkan Op1 dengan Op2
-	Op1 – Op2	Mengurangkan Op1 dengan Op2
*	Op1 * Op2	Mengalikan Op1 dengan Op2
/	Op1 / Op2	Membagi Op1 dengan Op2
%	Op1 % Op2	Menghasilkan sisa hasil bagi Op1 dengan Op2

Latihan 4. Aritmatika.java

```
class Aritmatika{
    public static void main(String[] args) {
        int a = 20;
        int b = 10;
        System.out.println("Penggunaan Operator Aritmatika ");
        System.out.println("Nilai awal a adalah : "+a);
        System.out.println("Nilai awal b adalah : "+b);
        System.out.println("Hasil dari a + b = " +(a + b));
        System.out.println("Hasil dari a - b = " +(a - b));
        System.out.println("Hasil dari a / b = " +(a / b));
        System.out.println("Hasil dari a * b = " +(a * b));
        System.out.println("Hasil dari a % b = " +(a % b));
    }
}
```

```
C:\ Command Prompt
G:\JAVAA>javac Aritmatika.java
G:\JAVAA>java Aritmatika
Penggunaan Operator Aritmatika
Nilai awal a adalah : 20
Nilai awal b adalah : 10
Hasil dari a + b = 30
Hasil dari a - b = 10
Hasil dari a / b = 2
Hasil dari a * b = 200
Hasil dari a % b = 0
```

**B. Operator Increment dan Decrement**

Operator Increment dan Decrement digunakan untuk menaikkan atau menurunkan suatu nilai integer (bilangan bulat) sebanyak satu satuan, dan hanya dapat digunakan pada variabel.

Ada dua versi operator increment maupun decrement, yaitu prefix dan postfix. Prefix berarti operator digunakan sebelum variabel atau ekspresi, dan postfix berarti operator digunakan sesudahnya. Penjelasan selengkapnya yaitu:

Operator	Penggunaan	Deskripsi
++	Op++	Op dinaikkan nilainya 1 setelah dilakukan operasi pada Op
	++Op	Op dinaikkan nilainya 1 sebelum dilakukan operasi pada Op
--	Op--	Op diturunkan nilainya 1 setelah dilakukan operasi pada Op
	--Op	Op diturunkan nilainya 1 sebelum dilakukan operasi pada Op

**Latihan 5. IncrementDecrement.java**

```
class IncrementDecrement{
    public static void main (String[] args){
        int i = 1;
        System.out.println("i : " + i);
        System.out.println("++i : " + ++i);
        System.out.println("i++ : " + i++);
        System.out.println("i : " + i);
        System.out.println("--i : " + --i);
        System.out.println("i-- : " + i--);
        System.out.println("i : " + i);
    }
}
```

```

C:\ Command Prompt
G:\JAVAA>javac IncrementDecrement.java
G:\JAVAA>java IncrementDecrement
i : 1
++i : 2
i++ : 2
i : 3
--i : 2
i-- : 2
i : 1

```

C. Operator Assignment ( Penugasan )

Operator assignment dalam Java digunakan untuk memberikan sebuah nilai ke sebuah variabel. Operator assignment hanya berupa '=', namun selain itu dalam Java dikenal beberapa *shortcut assignment operator* yang penting, yang digambarkan dalam tabel berikut :

Operator	Penggunaan	Ekuivalen Dengan
+=	Op1 += Op2	Op1 = Op1 + Op2
-=	Op1 -= Op2	Op1 = Op1 - Op2
*=	Op1 *= Op2	Op1 = Op1 * Op2
/=	Op1 /= Op2	Op1 = Op1 / Op2
%=	Op1 %= Op2	Op1 = Op1 % Op2
&=	Op1 &= Op2	Op1 = Op1 & Op2
!=	Op1 != Op2	Op1 = Op1 ! Op2
^=	Op1 ^= Op2	Op1 = Op1 ^ Op2
<<=	Op1 <<= Op2	Op1 = Op1 << Op2
>>=	Op1 >>= Op2	Op1 = Op1 >> Op2
>>>=	Op1 >>>= Op2	Op1 = Op1 >>> Op2

Latihan 6. Assignment.java

```

class Assignment {
    public static void main(String[] args) {
        int var = 10;
        int a,b,c;
        a = b = c = 100;
        int d,e,f;
        f = 200;
        e = f;
        d = e;
    }
}

```

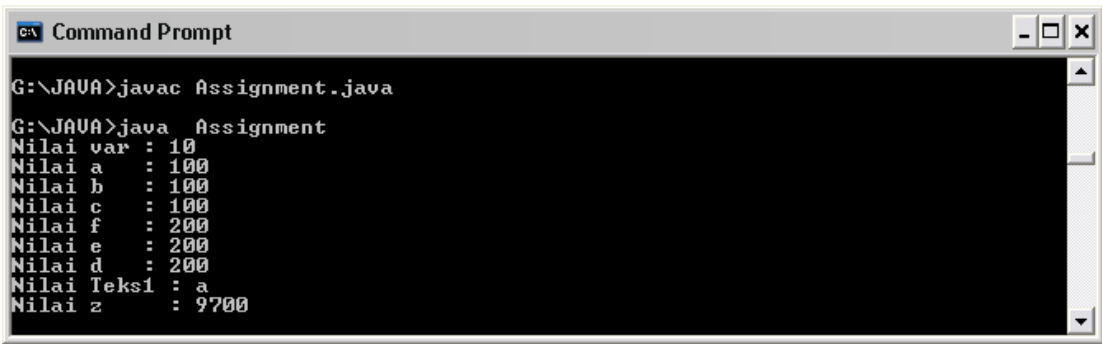
```
System.out.println("Nilai var : " + var);
System.out.println("Nilai a  : " + a);
System.out.println("Nilai b  : " + b);
System.out.println("Nilai c  : " + c);
System.out.println("Nilai f  : " + f);
System.out.println("Nilai e  : " + e);
System.out.println("Nilai d  : " + d);

int z;

char Teks1 = 'a'; // dalam Unicode karakter 'a' direpresentasikan dengan
angka 97

z = Teks1 * 100; // z = 97 * 10;

System.out.println("Nilai Teks1 : " + Teks1);
System.out.println("Nilai z   : " + z);
}
}
```



D. Operator Relasi

Operator relasi dalam Java digunakan untuk menghasilkan nilai boolean yang sering digunakan untuk mengatur alur jalannya sebuah program.

Operator	Penggunaan	Deskripsi
>	Op1 > Op2	Menghasilkan true jika Op1 lebih besar dari Op2
<	Op1 < Op2	Menghasilkan true jika Op1 lebih kecil dari Op2
>=	Op1 >= Op2	Menghasilkan true jika Op1 lebih besar atau sama dengan Op2
<=	Op1 <= Op2	Menghasilkan true jika Op1 lebih kecil atau sama dengan Op2
==	Op1 == Op2	Menghasilkan true jika Op1 sama dengan Op2
!=	Op1 != Op2	Menghasilkan true jika Op1 tidak sama dengan Op2

### ***Latihan 7. Relasi.java***

```
class Relasi{
    public static void main(String[] args) {
        int x,y,z;
        x = 100;
        y = 99;
        z = 99;
        System.out.println("Nilai x = "+x);
        System.out.println("Nilai y = "+y);
        System.out.println("Nilai z = "+z);
        // operator sama dengan
        if(y == z ){
            System.out.println("y sama dengan z");
        }else {
            System.out.println("y tidak sama dengan z");
        }
        // operator tidak sama dengan
        if(x != y ){
            System.out.println("x tidak sama dengan y");
        }else {
            System.out.println("x sama dengan y");
        }
        // operator lebih besar dari
        if(x > y ){
            System.out.println("x lebih besar dari y");
        }else {
            System.out.println("x lebih kecil dari y");
        }
        // operator lebih kecil dari
        if(y < x ){
            System.out.println("y lebih kecil dari x");
        }else {
            System.out.println("y lebih besar dari x");
        }
        // operator lebih besar dari atau sama dengan
        if(x >= y ){
            System.out.println("x lebih besar dari atau sama dengan y");
```

```

    }else {
        System.out.println("x lebih kecil dari atau sama dengany");
    }
    // operator lebih kecil dari atau sama dengan
    if(y <= x ){
        System.out.println("y lebih kecil dari atau sama dengan x");
    }else {
        System.out.println("y lebih besar dari atau sama dengan x");
    }
}
}
}

```



### E. Operator Logical

Operator ini digunakan untuk ekspresi logik yang menghasilkan nilai boolean. Operator-operator yang digunakan adalah AND ( && ), OR ( || ) dan NOT ( ! ).

A	B	A    B	A && B
True	True	True	True
True	False	True	False
False	True	True	False
False	False	False	False

#### Latihan 8. logik.java

```

class logik{
    public static void main(String[] args) {
        boolean Benar = true;
        boolean Salah = false;

        System.out.println("Hubungan OR (||)");
        System.out.println("Benar || Benar : " +(Benar||Benar));
        System.out.println("Benar || Salah : " +(Benar||Salah));
    }
}

```

```

System.out.println("Salah || Benar : " +(Salah||Benar));
System.out.println("Salah || Salah : " +(Salah||Salah));

System.out.println("Hubungan AND (&&");
System.out.println("Benar && Benar : " +(Benar&&Benar));
System.out.println("Benar && Salah : " +(Benar&&Salah));
System.out.println("Salah && Benar : " +(Salah&&Benar));
System.out.println("Salah && Salah : " +(Salah&&Salah));

System.out.println("Hubungan NOT (!)");
System.out.println("Kebalikan (NOT) dari Benar adalah: " +!Benar);
System.out.println("Kebalikan (NOT) dari Salah adalah: " +!Salah);
}
}

```



### F. Operator Bitwise

Operator ini dalam Java digunakan untuk melakukan manipulasi bit.

Operator	Penggunaan	Deskripsi
&	Op1 & Op2	Bitwise AND
	Op1   Op2	Bitwise OR
^	Op1 ^ Op2	Bitwise XOR
~	~Op	Bitwise Complement
<<	Op1 << Op2	Menggeser bit Op1 ke kiri sejauh Op2
>>	Op1 >> Op2	Menggeser bit Op1 ke kanan sejauh Op2
>>>	Op1 >>> Op2	Geser kanan tanpa mempertahankan sign (dengan nilai 0 sebagai pengisi bit paling kiri)

1. Bitwise AND

Bitwise AND akan menghasilkan bit “1”, jika kedua operator bernilai bit “1”. Operasi bitwise AND dapat digambarkan sebagai berikut :

Op1	Op2	Op1 & Op2
0	0	0
0	1	0
1	0	0
1	1	1

2. Bitwise OR

Bitwise OR akan menghasilkan bit “1”, jika salah satu operator bernilai bit “1”. Operasi bitwise OR dapat digambarkan sebagai baerikut :

Op1	Op2	Op1   Op2
0	0	0
0	1	1
1	0	1
1	1	1

3. Bitwise XOR ( Exclusive OR )

Bitwise XOR akan menghasilkan bit “1”, jika kedua operator memiliki nilai bit yang berbeda. Operasi bitwise XOR dapat digambarkan sebagai berikut :

Op1	Op2	Op1 ^ Op2
0	0	0
0	1	1
1	0	1
1	1	0

4. Bitwise Complement

Bitwise Complement akan menghasilkan bit yang berlawanan dengan bit yang dioperasikan. Operasinya dapat digambarkan sebagai berikut :

Op	Op
0	1
1	0



### ***Latihan 9. Bitwise.java***

```
class Bitwise{
    public static void main(String[] args) {
        int x,y;
        x = ~100;
        System.out.println("Nilai negasi x : "+x);
        x = 17 & 30;
        System.out.println("Nilai and : "+x);
        x = 17 | 30;
        System.out.println("Nilai or : "+x);
        x = 17 ^ 30;
        System.out.println("Nilai xor : "+x);
        x = 111;
        y = x >> 1;
        System.out.println("Nilai geser kanan : "+x);
        x = -111;
        y = x >> 1;
        System.out.println("Nilai geser kanan neg : "+x);
        x = 111;
        y = x >>> 1;
        System.out.println("Nilai geser kanan 1 bit 0 : "+x);
        x = -111;
        y = x >>> 1;
        System.out.println("Nilai geser kanan 1 bit 0 neg : "+x);
        x = 111;
        y = x << 1;
        System.out.println("Nilai geser kiri : "+x);
        x = 1000;
        y = x << 1;
        System.out.println("Nilai geser kiri : "+x);
    }
}
```

```

G:\JAVAA>javac Bitwise.java
G:\JAVAA>java Bitwise
Nilai negasi x : -101
Nilai and : 16
Nilai or : 31
Nilai xor : 15
Nilai geser kanan : 111
Nilai geser kanan neg : -111
Nilai geser kanan 1 bit 0 : 111
Nilai geser kanan 1 bit 0 neg : -111
Nilai geser kiri : 111
Nilai geser kiri : 1000

```

G. Precedence Operator

Operasi-operasi yang menggunakan operator dapat melibatkan lebih dari 1 operator dan 1 operand. Adapun urutan precedence operator dalam Java, dapat digambarkan dalam tabel berikut :

Postfix operators	[] . (params) expr++ expr--
Unary operators	++expr --expr +expr -expr ~ !
Creation or cast	New (type)expr
Multiplicative	* / %
Additive	+ -
Shift	<< >> >>>
Relational	<> <= >= instanceof
Equality	== !=
Bitwise AND	&
Bitwise exclusive OR	^
Bitwise inclusive OR	
Logical AND	&&
Logical OR	
Conditional	? :
Assignment	= += -= *= /= %= &= ^=  = <<= >>= >>>=

LATIHAN

1. Apa perbedaan X++ dengan ++X, jelaskan ?
2. Bagaimana menghitung perkalian dan pembagian dengan bilangan dua tanpa menggunakan operator matematika ?
3. Bagaimana menukar nilai dari dua variabel integer dengan tanpa menggunakan perantara ?