

PEMROGRAMAN JAVA

Yoannita



■ **Operator**



Special thanks : Some slides are from Josephine Petrina K & Gasim Alkaff

Operator Aritmatika

- Merupakan tanda untuk operasi aritmatika antara dua nilai

Opr	Fungsi	Contoh	Deskripsi
+	Penjumlahan	$a + b$	Penjumlahan a dan b
-	Pengurangan	$a - b$	Pengurangan a dan b
*	Perkalian	$a * b$	Perkalian a dan b
/	Pembagian	a / b	Pembagian a dan b
%	Modulus	$a \% b$	Sisa pembagian a dan b



Operator Aritmatika

Contoh Program

```
// nama file : arOp.java
class arOp {
    public static void main (String [ ] args) {
        double a = 5.0;
        double b = 2.0;
        double tambah = a + b;
        double kurang = a - b;
        double kali = a * b;
        double bagi = a / b;
        double sisa = a % b;

        System.out.println (" a + b = " + tambah); //atau :
        System.out.println (" a + b = " + (a + b));
        System.out.println (" a - b = " + kurang);
        System.out.println (" a * b = " + kali);
        System.out.println (" a / b = " + bagi);
        System.out.println (" a % b = " + sisa);
    }
}
```

Aritmatika

- Diberikan pernyataan yang membingungkan,

$$6\%2*5+4/2+88-10$$

- Kita dapat menuliskan kembali pernyataan diatas dan menambahkan beberapa tanda kurung terhadap operator precedence,

$$((6\%2)*5)+(4/2)+88-10;$$

Untuk menghindari kebingungan dalam evaluasi operasi matematika, buatlah pernyataan sesederhana mungkin dan gunakan bantuan tanda kurung.

Assignment Operator

Opr	Fungsi	Contoh	Deskripsi
=	Sama dengan	a = b	a akan bernilai sama dengan b
+=	Penjumlahan & sama dengan	a += b	$a = a + b$
-=	Pengurangan & sama dengan	a -= b	$a = a - b$
*=	Perkalian & sama dengan	a *= b	$a = a * b$
/=	Pembagian & sama dengan	a /= b	$a = a / b$
%=	Modulus & sama dengan	a %= b	$a = a \% b$



Assignment Operator

Contoh Program

```
// nama file : arOp.java
class asOp {
    public static void main (String [ ] args) {
        int a = 4;
        int b = 3;
        int c = 2;
        System.out.println ("a = " + a + "b = " + b + "c = " + c);
        c += b;    // c = c + b
        System.out.println ("c += b hasilnya c = " + c);
        c -= a;    // c = c - a
        System.out.println ("c -= a hasilnya c = " + c);
        c *= b;    // c = c * b
        System.out.println ("c *= b hasilnya c = " + c);
    }
}
```

Increment / Decrement Operator

Opr	Fungsi	Contoh	Deskripsi
++	Increment Operator	<code>nilai++</code>	Nilai ditambah satu setelah dioperasikan
		<code>++nilai</code>	Nilai ditambah satu sebelum nilai dioperasikan
--	Decrement Operator	<code>nilai--</code>	Nilai berkurang satu setelah dioperasikan
		<code>--nilai</code>	Nilai berkurang satu sebelum nilai dioperasikan



Increment / Decrement Operator

- Penulisan operator sebagai akhiran ($y = x++$ atau $y = x--$) menunjukkan bahwa y memperoleh nilai x sebelum x berubah
- Jika operator ditulis sebagai awalan ($y = ++x$ atau $y = --x$), nilai x diberikan ke y setelah terjadinya perubahan



```
class PenambahanAwalAkhir
```

```
{  
    public static void main (String[] args)  
    {  
        int x = 0;    int y = 0;  
        System.out.println("x dan y bernilai " + x + " dan " + y);  
        x++;  
        System.out.println("x++ menghasilkan "+ x);  
        ++x;  
        System.out.println("++x menghasilkan "+ x);  
  
        System.out.println("\nx dinolkan lagi!\n");  
        x = 0;  
  
        y = x++;  
        System.out.println("y = x++ (akhiran) menghasilkan : ");  
        System.out.println("x is " + x);  
        System.out.println("y is " + y);  
        y = ++x;  
        System.out.println("y = ++x (awalan) menghasilkan : ");  
        System.out.println("x is " + x);  
        System.out.println("y is " + y);  
    }  
}
```

```
x dan y bernilai 0 dan 0  
x++ menghasilkan 1  
++x menghasilkan 2  
  
x dinolkan lagi!  
  
y = x++ (akhiran) menghasilkan :  
x is 1  
y is 0  
y = ++x (awalan) menghasilkan :  
x is 2  
y is 2
```

Comparison Operator

Bernilai **True** atau **False**

Opr	Fungsi	Contoh	Deskripsi
<code>==</code>	Equality operator	<code>a == b</code>	Bernilai true jika a sama dengan b
<code>!=</code>	Inequality operator	<code>a != b</code>	Bernilai true jika a tidak sama dengan b
<code><</code>	Less than operator	<code>a < b</code>	Bernilai true jika a lebih kecil dari b
<code>></code>	Greater than operator	<code>a > b</code>	Bernilai true jika a lebih besar b
<code><=</code>	Less than or equal operator	<code>a <= b</code>	Bernilai true jika a lebih kecil atau sama dengan b
<code>>=</code>	Greater than or equal operator	<code>a >= b</code>	Bernilai true jika a lebih kecil atau sama dengan b



Comparison Operator

Contoh Program

```
// nama file : classComp.java
class classComp {
    public static void main (String [ ] args) {
        int a = 3;
        int b = 4;
        double c = 5.0;
        boolean d;
        d = (a == b);
        System.out.println ("a = b? " + d);
    }
}
/*
```

Hasil :

false

```
*/
```

Operator Relasional

Opr	Fungsi	Contoh	Deskripsi
&&	Logical AND operator	a && b	Bernilai true jika kedua nilai benar
	Logical OR operator	a b	Bernilai true jika salah satu dari keduanya benar
!	Logical NOT operator	! b	Bernilai true jika nilai b adalah false



Operator Relasional Contoh Program

```
class LoCom
{
    public static void main(String args[])
    {
        boolean a = true;
        boolean b = true;
        boolean c = false;
        boolean e = false;
        boolean d; boolean k;

        d = a && b;
        System.out.println("true && true   = " + d);

        k = c && b;
        System.out.println("false && true  = " + k);

        System.out.println("false && false = " + (k && c));
        System.out.println("true || false  = " + (a || c));
        System.out.println("true || true   = " + (a || b));
        System.out.println("false || false = " + (e || c));
    }
}
```

Hasil :

```
true && true = true
false && true = false
false && false = false
true || false = true
true || true = true
false || false = false
```



Ternary if-then-else operators

Expression ? statement1 : statement2

- expression menyatakan besaran boolean yang akan diperiksa.
- Jika **expression bernilai benar (true)** maka **statement1 dieksekusi** atau dijalankan,
- sedangkan statement2 dijalankan jika expression bernilai salah (false).
- Selain itu kedua statement harus memiliki tipe yang sama.
- Ternary operator dapat diistilahkan :
If **Expression ? Then statement1 else : statement2**



Expression ? statement1 : statement2

```
// nama file : ternary.java
// deskripsi : Ternary Operator
// kompilasi : javac ternary.java
// eksekusi : java ternary
class ternary
{
    public static void main(String args[])
    {
        int a = 10;
        int b = 20;
        int c = (b > 0) ? (a * 10) : (a - b);
        System.out.println("hasil = " + c);
    }
}

/*
hasil = 100
*/
```

Latihan (i)

Mendapatkan nilai rata-rata dari tiga angka

- Buatlah program yang menghasilkan output nilai rata-rata dari tiga angka. Nilai dari masing-masing tiga angka tersebut adalah 10, 20 dan 45.
- Tampilan Output yang diharapkan adalah,

```
number 1 = 10  
number 2 = 20  
number 3 = 45  
Rata-rata = 25
```




[Latihan (ii)

Luas dan keliling persegi panjang]

- Buatlah program java untuk menghitung luas dan keliling persegi panjang berukuran 7 x 5 meter

Latihan (iv)

Luas Arsiran

- Buatlah program untuk menghitung luas bidang segiempat, luas bidang lingkaran, dan juga luas bidang yang diarsir pada gambar segiempat berikut ini.

Algoritma Menghitung_Luas_Bidang_Arsiran

$r = 14$

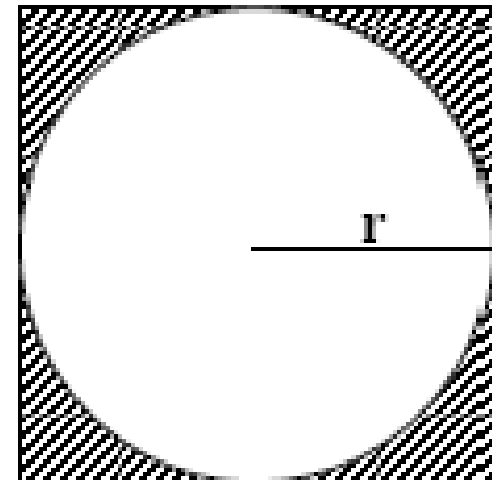
$\text{phi} \leftarrow 3.14$

$\text{Sisi} \leftarrow 2 * r$

$\text{LuasPersegi} \leftarrow \text{sisi} * \text{sisi}$

$\text{LuasLingkaran} \leftarrow \text{phi} * (r * r)$

$\text{LuasArsiran} \leftarrow \text{LuasPersegi} - \text{LuasLingkaran}$





Petunjuk Penulisan Program

(i)

1. Pada saat pembuatan blok, Anda dapat meletakkan kurung kurawal buka pada baris dengan pernyataan seperti contoh sebagai berikut ,

```
public static void main( String[] args ){
```

atau Anda dapat meletakkan kurung kurawal pada baris selanjutnya, seperti,

```
public static void main( String[] args )  
{
```

2. Anda harus memberi jarak (indent) pernyataan selanjutnya setelah awal dari blok , seperti contoh berikut,

```
public static void main( String[] args ){  
    System.out.println("Hello");  
    System.out.println("world");  
}
```



Petunjuk Penulisan Program

(ii)

1. Untuk pemberian nama dari class Java, diberikan *huruf kapital untuk huruf pertama pada nama class*. Untuk *nama method dan variabel*, huruf pertama dari kata harus dimulai dengan *huruf kecil*.
Sebagai contoh:

```
ThisIsAnExampleOfClassName  
thisIsAnExampleOfMethodName
```

```
class BinatangAir  
String ikanAirTawar  
void membacaKoran()
```

2. Pada kasus untuk identifier lebih dari satu kata, gunakan huruf kapital untuk mengindikasikan awal dari kata kecuali kata pertama.
Sebagai contoh *charArray*, *fileNumber*, *ClassName*.
3. Sebaiknya hindari menggunakan *underscores* pada awal identifier seperti *_read* atau *_write*.