

OPERATOR, STATEMENT KONDISIONAL, DAN ITERASI PADA JAVA

OPERATOR

OPERATOR ARITMATIKA

Operator ini digunakan pada operasi-operasi aritmatika seperti penjumlahan, pengurangan, pembagian dll.

Jenis Operator Aritmatika

operator	contoh	keterangan
+	a + b	Menambahkan a dengan b
-	a - b	Mengurangkan a dengan b
*	a * b	Mengalikan a dengan b
/	a / b	Membagi a dengan b
%	a % b	Menghasilkan sisa pembagian antara a dengan b (mod)
++	a ++	Menaikkan nilai a sebesar 1
--	a --	Menurunkan nilai a sebesar 1
-	- a	Negasi dari a.

OPERATOR RELASIONAL

Untuk membandingkan 2 nilai (variabel) atau lebih digunakan operator Relasional, dimana operator ini akan mengembalikan atau menghasilkan nilai True atau False.

Jenis Operator Relasional

operator	contoh	keterangan
>	a > b	True jika a lebih besar dari b
<	a < b	True jika a lebih kecil dari b
>=	a >= b	True jika a lebih besar atau sama dengan b
<=	a <= b	True jika a lebih kecil atau sama dengan b
==	a == b	True jika a sama dengan b
!=	a != b	True jika a tidak sama dengan b

OPERATOR KONDISIONAL

Operator ini menghasilkan nilai yang sama dengan operator relasional, hanya saja penggunaannya lebih pada operasi-operasi boolean.

Jenis Operator Kondisional

operator	contoh	keterangan
&&	a && b	True jika a dan b, keduanya bernilai True
	a b	True jika a atau b bernilai True
!	! a	True jika a bernilai False

OPERATOR SHIFT DAN BITWISE

Kedua operator ini digunakan untuk memanipulasi nilai dari bitnya, sehingga diperoleh nilai yang lain.

Jenis Operator Shift dan Bitwise

operator shift		
Operator	Penggunaan	Deskripsi
>>	a >> b	Menggeser bit a ke kanan sejauh b
<<	a << b	Menggeser bit a ke kiri sejauh b
>>>	a >>> b	Menggeser bit a ke kanan sejauh b
operator bitwise		
Operator	Penggunaan	Deskripsi
&	a & b	Bitwise AND
	a b	Bitwise OR
^	a ^ b	Bitwise XOR
~	~a	Bitwise Complement

OPERATOR ASSIGNMENT

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

Jenis Operator Assignment

Operator	Contoh	Ekivalen dengan
+=	a += b	a = a + b
-=	a -= b	a = a - b
*=	a *= b	a = a * b
/=	a /= b	a = a / b
%=	a %= b	a = a % b
&=	a &= b	a = a & b
=	a = b	a = a b
^=	a ^= b	a = a ^ b
<<=	a <<= b	a = a << b
>>=	a >>= b	a = a >> b
>>>=	a >>>= b	a = a >>> b

PERNYATAAN KONDISIONAL

Statement if

```
if (ekspresi_kondisional){
    statement1;
    statement2;
    ...
}
```

Statement *if* memungkinkan sebuah program untuk dapat memilih beberapa operasi untuk di eksekusi, berdasarkan beberapa pilihan. Dalam bentuknya yang paling sederhana, bentuk *if* mengandung sebuah pernyataan tunggal yang dieksekusi jika ekspresi bersyarat adalah benar.

STATEMENT *if-else*

Untuk melakukan beberapa operasi yang berbeda jika salah satu *ekspresi_kondisional* bernilai salah, maka digunakan statement else. Bentuk *if-else* memungkinkan kode Java memungkinkan dua alternatif operasi pemrosesan : satu jika statement bersyarat adalah benar dan satu jika salah.

Bentuk statement *if - else* dengan 2 pilihan operasi pemrosesan :

```
if (ekspresi_kondisional){
    statement1;
    statement2;
    ...
}else{
    statement1;
    statement2;
    ...
}
```

Bentuk statement *if - else* dengan beberapa pilihan operasi pemrosesan

```
if (ekspresi_kondisional_A){
    statement1;
    statement2;
    ...
}else if (ekspresi_kondisional_B){
    statement1;
    statement2;
    ...
}else{
    statement1;
    statement2;
    ...
}
```

try this at home

```
public class IfElse {
    public static void main(String args[]) {
        int month = 4;
        String season;

        if (month == 12 || month == 1 || month == 2) {
            season = "Dingin";
        } else if (month == 3 || month == 4 || month == 5) {
            season = "Semi";
        } else if (month == 6 || month == 7 || month == 8) {
            season = "Panas";
        } else if (month == 9 || month == 10 || month == 11) {
            season = "Gugur";
        } else {
            season = "";
        }

        System.out.println("Bulan April masuk musim " + season + ".");
    }
}
```

Modul I
Pengenalan Java
Statement *switch*

Bentuk umum pernyataan *switch*

```
switch ( expression ) {  
    case value_1:  
        statement1;  
        statement2;  
        . . .  
        break;  
    case value_2:  
        statement1;  
        statement2;  
        . . .  
        break;  
    [default: ]  
        statement1;  
        statement2;  
        . . .  
        break;}}
```

case: menandai posisi kode di mana eksekusi dilaksanakan.

value_1, dst.: konstanta integer atau konstanta karakter, ataupun ekspresi yang mengevaluasi keduanya.

default: sama kayak "else" pada *statement if*.

try this at home

```
public class Coba{  
    public static void main(String[] args)  
    {  
        int N=6;  
        switch (N) { /* asumsi N adalah variabel bertipe integer */  
        case 1:  
            System.out.println("Angka 1.");  
            break;  
        case 2:  
        case 4:  
        case 8:  
            System.out.println("Angka 2, 4, atau 8.");  
            System.out.println("(Kelipatan dari 2!)");  
            break;  
        case 3:  
        case 6:  
        case 9:  
            System.out.println("Angka 3, 6, or 9.");  
            System.out.println("(Faktor dari 3!)");  
            break;  
        case 5:  
            System.out.println("Angka 5.");  
            break;  
        default:  
            System.out.println("Angka 7,");  
            System.out.println("atau Angka Lebih dari Range 1-9.");  
        }  
    }  
}
```

EKSPRESI BERSYARAT (TERNARY OPERATOR)

Kita menggunakan sebuah ekspresi bersyarat untuk menggantikan sebuah bentuk if-else. Sintaks adalah sebagai berikut:

```
exp1 ? exp2 : exp3
```

Arti dari sintaks di sebelah sama dengan:
`if (exp1==true) then exp2 else exp3`

try this at home

```
public class TernaryExample{
    public static void main (String args[]){
        int x=3, y=6, m;
        int a=2,b=3,c=4;
        m = x > y ? x : y;
        System.out.println(m);
    }
}
```

Berapakah outputnya?

Answer:

STATEMENT ITERASI (PERULANGAN)

While

```
while( expression ){
    statement1;
    statement2;
    .....
}
```

Do

```
do {
    statement1;
    statement2;
    .....
} while(expression)
```

For

```
for (initialization;expression;step){
    statement1;
    statement2;
    ...
}
```

Selama expression benar, while akan dieksekusi
Hasil dari while akan dikembalikan kepada do

try this at home

```
public class Angka{
    public static void main (String[] args){
        int i;
        for(i=1;i<=10;i++){
            System.out.println(Integer.toString(i));
        }
        i=1;
        while(i<=10){
            System.out.println(Integer.toString(i));
        }
        do{
            System.out.println(Integer.toString(i));
            i++;
        }while(i<=10)
    }
}
```

Pada java terdapat 2 statement yang biasanya digunakan pada setiap bentuk iterasi diatas.

Statement tersebut yaitu:

- **break**, dapat menghentikan perulangan walaupun kondisi untuk berhenti belum terpenuhi.
- **continue**, dengan statement ini kita bisa melewatkan operasi yang dilakukan dalam iterasi sesuai dengan kondisi tertentu.

exercise for you

1. Buatlah program untuk mencari dan menampilkan bilangan prima diantara 2 bilangan yang diinputkan melalui argument.

Misal : java Prima 3 30

Output : 3 5 7 11 13 17 19 23 29

2. Buatlah program untuk penjumlahan 2 buah matrik (3x3). Nilai matrik dapat bersifat statis ataupun dirandom.

3. Buat program untuk mencari huruf yang paling banyak muncul pada suatu kalimat.

Misal : kalimat : praktikum pemrograman java



references

Modul praktikum PBO 2008-2009

Hariyanto, Bambang, "Esensi-Esensi Bahasa Pemrograman Java", Informatika Bandung, Bandung, 2003.

<http://eechhoowordpress.com/2008/08/11/download-tutorial-pemrograman-java-bahasa-indonesia/>

Linden, Peter van der. "Just Java: Edisi Bahasa Indonesia". Yogyakarta: 1997