

Dasar-Dasar Pemrograman Java

Tipe data primitif

1. logika - boolean

Tipe data boolean diwakili oleh dua pernyataan : true dan false

2. teksual – char

Tipe data character (char) diwakili oleh karakter single Unicode.

Tipe data ini harus memiliki ciri berada dalam tanda single quotes(' ')

```
'a' //Huruf a
'\t' //A tab
\" //untuk single quotes
\" \" //untuk double quotes
```

Meskipun, String bukan merupakan tipe data primitif (namun merupakan suatu Class). String mewakili tipe data yang terdiri atas beberapa karakter.

```
String message="Hello world!"
```

3. Integral – byte, short, int & long

Tipe data yang terintegrasi dalam java menggunakan tiga bentuk- yaitu desimal, oktal atau heksadesimal.

Sebagai contoh,

```
2 //nilai desimal 2
077 //angka 0 pada awal pernyataan mengindikasikan nilai oktal
```

| <i>Integer Length</i> | <i>Name or Type</i> | <i>Range</i> |
|-----------------------|---------------------|-------------------------|
| 8 bits | byte | -2^7 to 2^7-1 |
| 16 bits | short | -2^{15} to $2^{15}-1$ |
| 32 bits | int | -2^{31} to $2^{31}-1$ |
| 64 bits | long | -2^{63} to $2^{63}-1$ |

4. Floating Point – float dan double

Tipe Floating point memiliki double sebagai default tipe datanya.

```
E or e //(add exponential value)
F or f //(float)
D or d //(double)
```

Contohnya :

```
3.14 //nilai floating-point sederhana (a double)
6.02E23 //A nilai floating-point yang besar
```



| <i>Float Length</i> | <i>Name or Type</i> | <i>Range</i> |
|---------------------|---------------------|-------------------------|
| 32 bits | float | -2^{31} to $2^{31}-1$ |
| 64 bits | double | -2^{63} to $2^{63}-1$ |

Variabel

Variabel adalah item yang digunakan data untuk menyimpan pernyataan object. Variabel memiliki tipe data dan nama.

Tipe data mengindikasikan tipe dari nilai yang dapat dibentuk oleh variabel itu sendiri.

Nama variabel harus mengikuti aturan untuk pengidentifikasian.

Deklarasi dan Inisialisasi Variabel

Untuk deklarasi variabel adalah sebagai berikut :

<data tipe> <name> [=initial value];

Catatan: Nilainya berada diantara <> adalah nilai yang disyaratkan, sementara nilai dalam tanda [] bersifat optional.

Berikut ini adalah contoh program yang mendeklarasikan dan menginisialisasi beberapa variabel,

```
public class VariableSamples
{
    public static void main( String[] args ){
        boolean result;
        char option;
        option = 'C';
        double grade = 0.0;
    }
}
```

Menampilkan Data Variabel

Untuk mengeluarkan nilai dari variabel yang diinginkan, kita dapat menggunakan perintah sebagai berikut :

System.out.println()
System.out.print()

Contoh program :

```
public class OutputVariable
{
    public static void main( String[] args ){
        int niai = 10;
        char x;
        x = 'A';
        System.out.println( niai );
        System.out.println( "Nilai x=" + x );
    }
}
```



Output :
10
Nilai x=A

System.out.println() vs. System.out.print()

Pernyataan 1:

```
System.out.print("Hello ");
System.out.print("world!");
```

Pernyataan tersebut akan menghasilkan output berikut ini pada layar :
Hello world!

Pernyataan 2:

```
System.out.println("Hello ");
System.out.println("world!");
```

Pernyataan ini akan menghasilkan output sebagai berikut pada layar :
Hello
world!

Operators

Dalam Java, ada beberapa tipe operator. Ada operator aritmatika, operator relasi, operator logika, dan operator kondisi. Operator ini mengikuti macam-macam prioritas yang pasti jadi compilernya akan tahu yang mana operator untuk dijalankan lebih dulu dalam kasus beberapa operator yang dipakai bersama-sama dalam satu pernyataan.

| <i>Operator</i> | <i>Use</i> | <i>Description</i> |
|------------------------|-------------------|---|
| + | op1 + op2 | Adds op1 and op2 |
| * | op1 * op2 | Multiplies op1 by op2 |
| / | op1 / op2 | Divides op1 by op2 |
| % | op1 % op2 | Computes the remainder of dividing op1 by op2 |
| - | op1 - op2 | Subtracts op2 from op1 |

Contoh program penggunaan operator-operator :

```
public class aritmatikaDemo
{
    public static void main(String[] args)
    {
        int i = 37;
        int j = 42;
        double x = 27.475;
        double y = 7.22;
        System.out.println("Nilai Variabel...");
        System.out.println(" i = " + i);
    }
}
```



```

System.out.println(" j = " + j);
System.out.println(" x = " + x);
System.out.println(" y = " + y);
System.out.println("Penjumlahan...");
System.out.println(" i + j = " + (i + j));
System.out.println(" x + y = " + (x + y));
System.out.println("Pengurangan...");
System.out.println(" i - j = " + (i - j));
System.out.println(" x - y = " + (x - y));
System.out.println("Perkalian...");
System.out.println(" i * j = " + (i * j));
System.out.println(" x * y = " + (x * y));
System.out.println("Pembagian...");
System.out.println(" i / j = " + (i / j));
System.out.println(" x / y = " + (x / y));
System.out.println("Sisa Hasil Bagi...");
System.out.println(" i % j = " + (i % j));
System.out.println(" x % y = " + (x % y));
System.out.println("Gabungan...");
System.out.println(" j + y = " + (j + y));
System.out.println(" i * x = " + (i * x));
}
}

```

Output program :

```

Nilai Variabel...
i = 37
j = 42
x = 27.475
y = 7.22
Penjumlahan...
i + j = 79
x + y = 34.695
Pengurangan...
i - j = -5
x - y = 20.255
Perkalian...
i * j = 1554
x * y = 198.37
Pembagian...
i / j = 0
x / y = 3.8054
Sisa Hasil Bagi...
i % j = 37
x % y = 5.815
Gabungan...
j + y = 49.22
i * x = 1016.58

```



Operator Increment dan Decrement

Dari sisi operator dasar aritmatika, java juga terdiri atas operator unary increment (++) dan operator unary decrement (--). operator increment dan decrement menambah dan mengurangi nilai yang tersimpan dalam bentuk variabel angka terhadap nilai 1. Sebagai contoh :

```
count = count + 1; //increment nilai count dengan nilai 1
count++;
```

| Operator | Use | Description |
|-----------------|------------|--|
| ++ | op++ | Increments op by 1; evaluates to the value of op before it was incremented |
| ++ | ++op | Increments op by 1; evaluates to the value of op after it was incremented |
| -- | op-- | Decrements op by 1; evaluates to the value of op before it was decremented |
| -- | --op | Decrements op by 1; evaluates to the value of op after it was decremented |

Operator increment dan decrement dapat ditempatkan sebelum atau sesudah operand.

Contoh 1 :

```
int i = 10,
int j = 3;
int k = 0;
k = ++j + i; //akan menghasilkan k = 4+10 = 14
```

Contoh 2 :

```
int i = 10,
int j = 3;
int k = 0;
k = j++ + i; //akan menghasilkan k = 3+10 = 13
```

Operator Relasi

Operator Relasi membandingkan dua nilai dan menentukan keterhubungan diantara nilai-nilai tersebut. Hasil keluarannya berupa **nilai boolean** yaitu true atau false.

| Operator | Use | Description |
|-----------------|------------|-------------------------------------|
| > | op1 > op2 | op1 is greater than op2 |
| >= | op1 >= op2 | op1 is greater than or equal to op2 |
| < | op1 < op2 | op1 is less than op2 |
| <= | op1 <= op2 | op1 is less than or equal to op2 |
| == | op1 == op2 | op1 and op2 are equal |
| != | op1 != op2 | op1 and op2 are not equal |



Contoh program operator Relasi :

```
public class RelasiDemo
{
    public static void main(String[] args) {
        int i = 37;
        int j = 42;
        int k = 42;
        System.out.println("Nilai Variabel...");
        System.out.println(" i = " + i);
        System.out.println(" j = " + j);
        System.out.println(" k = " + k);
        System.out.println("lebih besar dari...");
        System.out.println(" i > j = " + (i > j));
        System.out.println(" j > i = " + (j > i));
        System.out.println(" k > j = " + (k > j));
        System.out.println("lebih besar atau sama dengan...");
        System.out.println(" i >= j = " + (i >= j));
        System.out.println(" j >= i = " + (j >= i));
        System.out.println(" k >= j = " + (k >= j));
        System.out.println("lebih kecil dari...");
        System.out.println(" i < j = " + (i < j));
        System.out.println(" j < i = " + (j < i));
        System.out.println(" k < j = " + (k < j));
        System.out.println("lebih kecil atau sama dengan...");
        System.out.println(" i <= j = " + (i <= j));
        System.out.println(" j <= i = " + (j <= i));
        System.out.println(" k <= j = " + (k <= j));
        System.out.println("sama dengan...");
        System.out.println(" i == j = " + (i == j));
        System.out.println(" k == j = " + (k == j));
        System.out.println("tidak sama dengan...");
        System.out.println(" i != j = " + (i != j));
        System.out.println(" k != j = " + (k != j));
    }
}
```

Output program :

```
Nilai variabel...
i = 37
j = 42
k = 42
lebih besar dari...
i > j = false
j > i = true
k > j = false
lebih besar dari atau sama dengan...
i >= j = false
j >= i = true
k >= j = true
lebih kecil dari...
i < j = true
j < i = false
```



```

k < j = false
lebih kecil dari atau sama dengan...
i <= j = true
j <= i = false
k <= j = true
sama dengan...
i == j = false
k == j = true
tidak sama dengan...
i != j = true
k != j = false
    
```

Operator logika

Operator logika memiliki satu atau lebih operand boolean yang menghasilkan nilai boolean. Ada enam operator logika yaitu: && (logika AND), & (boolean logika AND), || (logika OR), | (boolean logika inclusive OR), ^ (boolean logika exclusive OR), dan ! (logika NOT).

&& (logika AND) dan & (boolean logika AND)

| x1 | x2 | Result |
|-------|-------|--------|
| TRUE | TRUE | TRUE |
| TRUE | FALSE | FALSE |
| FALSE | TRUE | FALSE |
| FALSE | FALSE | FALSE |

Contoh program menggunakan logika dan boolean AND :

```

public class TestAND
{
    public static void main( String[] args ){
        int i = 0;
        int j = 10;
        boolean test= false;
        test = (i > 10) && (j++ > 9);
        System.out.println(i);
        System.out.println(j);
        System.out.println(test);
        test = (i > 10) & (j++ > 9);
        System.out.println(i);
        System.out.println(j);
        System.out.println(test);
    }
}
    
```

Output program :

```

0
10
false
0
11
false
    
```



|| (logika OR) dan | (boolean logika inclusive OR)

| x1 | x2 | Result |
|-----------|-----------|---------------|
| TRUE | TRUE | TRUE |
| TRUE | FALSE | TRUE |
| FALSE | TRUE | TRUE |
| FALSE | FALSE | FALSE |

Contoh program menggunakan operator logika dan boolean OR :

```
public class TestOR
{
    public static void main( String[] args ){
        int i = 0;
        int j = 10;
        boolean test= false;
        test = (i < 10) || (j++ > 9);
        System.out.println(i);
        System.out.println(j);
        System.out.println(test);
        test = (i < 10) | (j++ > 9);
        System.out.println(i);
        System.out.println(j);
        System.out.println(test);
    }
}
```

Output Program :

```
0
10
true
0
11
true
```

^ (boolean logika ExclusiveOR)

| x1 | x2 | Result |
|-----------|-----------|---------------|
| TRUE | TRUE | FALSE |
| TRUE | FALSE | TRUE |
| FALSE | TRUE | TRUE |
| FALSE | FALSE | FALSE |

Contoh program menggunakan operator logika exclusive OR :

```
public class TestXOR
{
    public static void main( String[] args ){
        boolean val1 = true;
        boolean val2 = true;
        System.out.println(val1 ^ val2);
        val1 = false;
```




```

val2 = true;
System.out.println(val1 ^ val2);
val1 = false;
val2 = false;
System.out.println(val1 ^ val2);
val1 = true;
val2 = false;
System.out.println(val1 ^ val2);
    
```

Output Program :

```

false
true
false
true
    
```

! (logika NOT)

| <i>x1</i> | <i>Result</i> |
|------------------|----------------------|
| TRUE | FALSE |
| FALSE | TRUE |

Contoh program menggunakan operator logika NOT :

```

public class TestNOT
{
    public static void main( String[] args ){
        boolean val1 = true;
        boolean val2 = false;
        System.out.println(!val1);
        System.out.println(!val2);
    }
}
    
```

Output program :

```

false
true
    
```

Operator Kondisi(?:)

exp1?exp2:exp3

Jika exp1 bernilai true, exp2 merupakan hasil operasi.
 Jika bernilai false, kemudian exp3 merupakan hasil operasinya.

Contoh program :

```

public class kondisiOperator
{
    public static void main( String[] args ){
        String status = "";
        int grade = 80;
        //mendapatkan status pelajar
        status = (grade >= 60)?"Passed":"Fail";
        //print status
    }
}
    
```

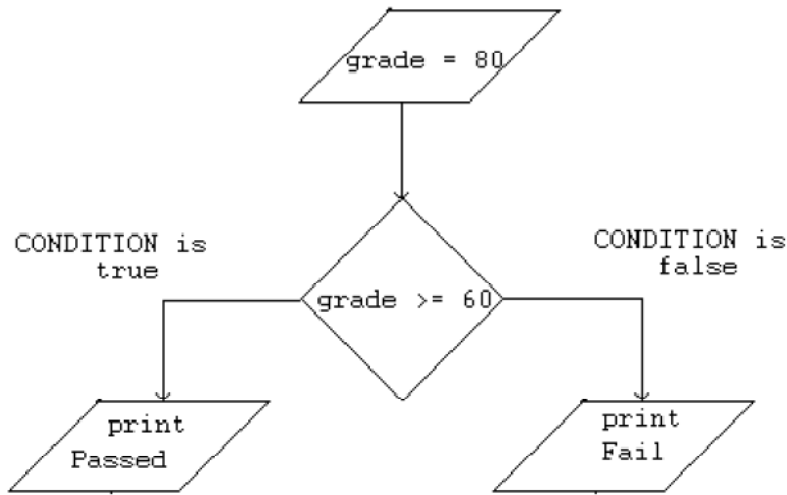


```

        System.out.println( status );
    }
}

```

Output program :
Passed



Contoh program :

```

class kondisiOperator
{
    public static void main( String[] args ){
        int score = 0;
        char answer = 'a';
        score = (answer == 'a') ? 10 : 0;
        System.out.println("Score = " + score );
    }
}

```

Output program :
Score = 10

Operator Precedence

- . [] ()
- ++ -- ! ~
- * / %
- + -
- << >> >>> <<<<
- < > <= >=
- == !=
- & |
- ^
- &&
- ||
- ?:
- =

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 :

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