

# Operator

Kita sudah belajar tentang bagaimana suatu variabel dideklarasikan dan diinisiasi. Selanjutnya kita akan berusaha memahami aksi yang akan diberlakukan terhadap variabel tersebut, misalnya mengubah, menambah, membagi variabel atau aksi lainnya. Untuk itu, kita akan menggunakan operator.

Operator dalam Java merupakan suatu simbol yang digunakan untuk melakukan suatu operasi tertentu (memanipulasi, mengolah) satu atau lebih variabel. Variabel yang dioperasikan disebut sebagai operand. Bahasa Java memiliki banyak operator yang dapat digunakan dan bisa dikelompokkan menjadi beberapa kategori, seperti :

- Operasi Assignment
- Operasi Arithmetic
- Operasi Unary
- Operasi Equality and Relational
- Operasi Conditional

Selengkapnya, silakan merujuk ke <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>.

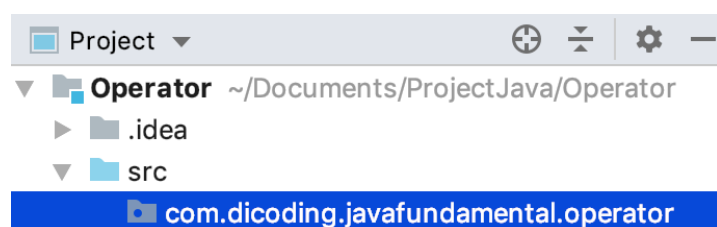
## Operasi Assignment

Operasi *Assignment* adalah salah satu operator yang akan sering ditemui dalam pemrograman Java. *Assignment Variable* (penempatan/pengalamatan variabel) merupakan operator yang cukup simpel dan diwakilkan dengan simbol operator '='. Operator *assignment* memberikan nilai pada sebuah variabel yang ada di sebelah kiri (*left operand*) dengan nilai yang ada di sebelah kanan (*right operand*).

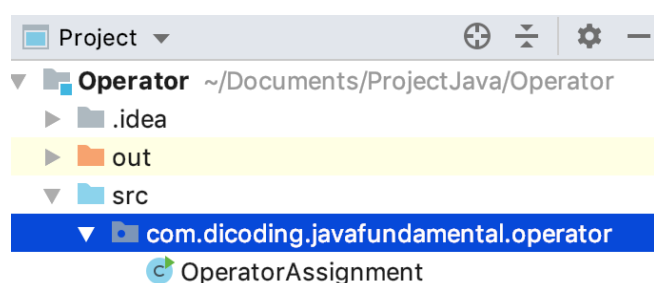
## Codelab Operasi Assignment

Langsung saja kita praktekan melalui kode di bawah ini:

1. Buatlah proyek baru dengan nama Operator dengan nama package com.dicoding.javafundamental.operator di dalamnya:



2. Buatlah sebuah kelas baru di dalamnya dengan nama **OperatorAssignment** :



3. Tambahkan kode berikut di dalam kelas `OperatorAssignment` :

```
1. package com.dicoding.javafundamental.operator;
2.
3. public class OperatorAssignment {
4.
5.     public static void main(String[] args) {
6.         int value = 2;
7.         int anotherValue = 19;
8.         System.out.println("Data pada value adalah ->> " + value);
9.         System.out.println("Data pada anotherValue adalah ->> " + anotherValue);
10.    }
11. }
```

4. Selanjutnya jalankanlah kode di atas pada IDE yang kalian gunakan. Bila sukses, seharusnya Console akan menampilkan output seperti ini.

```
Data pada value adalah ->> 2
Data pada anotherValue adalah ->> 19
```

## Bedah Code Operasi Assignment

Untuk penjelasannya, lihat bagian kode di bawah ini:

```
1. int value = 2;
```

Potongan kode di atas `value` merupakan left operand. Angka 2 merupakan sebuah nilai *integer* (right operand). Sesuai dengan penjelasan operasi *assignment*, nilai `value` akan menjadi nilai *integer* 2. Operasi ini bisa juga disebut sebagai *copy*, karena nilai left operand akan mengkopi nilai right operand.

## Operasi Aritmatika

Java memiliki banyak operator yang digunakan untuk membantu perhitungan Aritmatika. Operator Aritmatika pada Java antara lain :

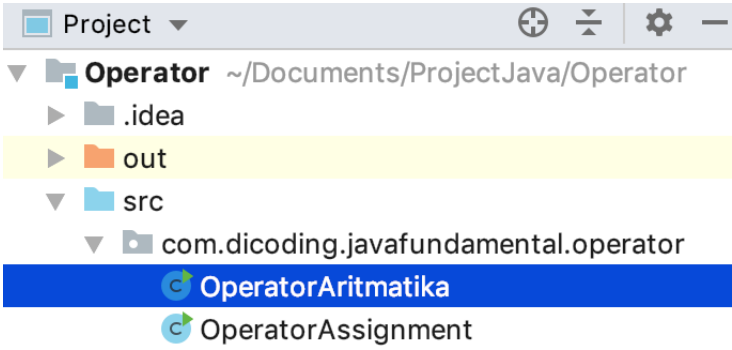
Operator	Deskripsi
+	Operator penambahan
-	Operator pengurangan
*	Operator pengalian
/	Operator pembagian
%	Operator sisa hasil

Pastinya kita sudah familiar dengan simbol-simbol operator di atas pada pelajaran Matematika dasar. Mungkin hanya satu simbol yang terlihat asing dan baru untuk Anda, yakni ‘%’ (persen) sebagai operator untuk menghitung sisa hasil bagi atau dapat disebut mod / modulus. Modulus adalah sebuah operasi yang menghasilkan sisa dari pembagian dari suatu bilangan terhadap bilangan lainnya.

## Codelab Operasi Aritmatika

Langsung saja kita praktekan melalui kode di bawah ini:

1. Bukalah proyek Operator dan tambahkan kelas baru didalamnya dengan nama `OperatorAritmatika`.



2. Tambahkan kode di bawah ini ke dalam kelas `OperatorAritmatika`:

```
1. package com.dicoding.javafundamental.operator;
2.
3. public class OperatorAritmatika {
4.
5.     public static void main(String[] args) {
6.         System.out.println("Operasi Penjumlahan");
7.         int hasilPenjumlahan = 5 + 1;
8.         System.out.println("Hasil 5 + 1 = " + hasilPenjumlahan);
9.         System.out.println();
10.
11.        System.out.println("Operasi Pengurangan");
12.        int hasilPengurangan = 4 - 1;
13.        System.out.println("Hasil 4 - 1 = " + hasilPengurangan);
14.        System.out.println();
15.    }
```

3. Selanjutnya jalankanlah kode di atas pada IDE yang kalian gunakan. Bila sukses, seharusnya Console akan menampilkan output seperti ini.

Operasi Penjumlahan

Hasil  $5 + 1 = 6$

Operasi Pengurangan

Hasil  $4 - 1 = 3$

Operasi Pengalian

Hasil  $5 * 5 = 25$

Operasi Pembagian

Hasil  $20 / 2 = 10$

Operasi Habis bagi

Hasil  $8 \% 2 = 0$

Hasil  $9 \% 2 = 1$

## Bedah Code Operasi Aritmatika

Kode di atas merupakan implementasi simbol-simbol Aritmatika yang didukung oleh Java. Seperti pada kode “`int hasilPenjumlahan = 5 + 1`”, tentunya ini cukup mudah dipahami bagi kita yang sudah mengenal matematika dasar.

Selanjutnya adalah implementasi dari operasi hasil bagi atau yang dipanggil dengan Modulus (mod).

```
1. int hasilSisa = 8 % 2;  
2. int hasilSisaLain = 9 % 2;
```

Modulus yang berarti habis dibagi, bila dicetak akan menghasilkan keluaran berikut ini:

```
1. Operasi Habis bagi  
2. Hasil 8 % 2 = 0  
3. Hasil 9 % 2 = 1
```

Dari output di atas, dapat disimpulkan bahwa nilai variabel 8 tidak bersisa bila dibagi dengan variabel 2. Sementara itu, nilai variabel 9 jika dibagi dengan variabel yang sama (2), menyisakan sisa hasil bagi 1.

## Operasi Unary

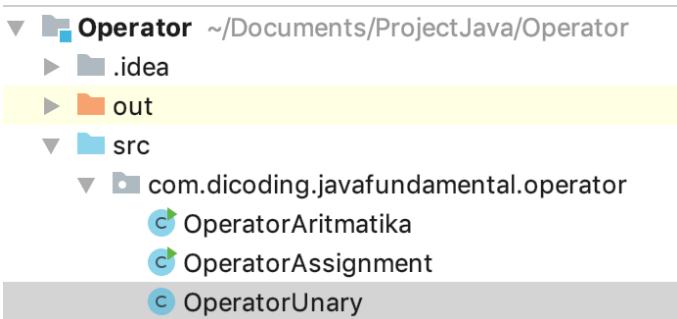
Merupakan operator yang melakukan berbagai operasi Aritmatik yang hanya melibatkan satu operand. Operasi-operasi tersebut dapat berupa penambahan, pengurangan atau membalik suatu nilai dari boolean. Dalam bahasa pemrograman Java, berikut inilah yang termasuk Operator Unary :

Operator	Deskripsi
+	Operator Unary positif
-	Operator Unary negatif
++	Operator penambahan; peningkatan nilai sebesar 1 point
--	Operator pengurangan;; pengurangan nilai sebesar 1 point
!	Operator komplemet logika, operator; membalikan sebuah nilai boolean

## Codelab Operasi Unary

Mari kita coba kode di bawah ini untuk melihat bagaimana implementasi Operator di atas.

1. Bukalah proyek Operator dan tambahkan kelas baru didalamnya dengan nama `OperatorUnary`.



2. Tambahkan kode di bawah ini ke dalam kelas `OperatorUnary`:

```
1. package com.dicoding.javafundamental.operator;
2.
3. public class OperatorUnary {
4.
5.     public static void main(String[] args) {
6.         System.out.println("Operator Unary Plus");
7.         int nilaiAwal = 5;
8.         int hasil = +nilaiAwal;
9.         System.out.println("Hasil +5 = " + hasil);
10.        System.out.println();
11.
12.        System.out.println("Operator Unary Minus");
13.        int nilaiAwal2 = 5;
14.        nilaiAwal2 = -nilaiAwal2;
15.        System.out.println("Hasil -5 = " + nilaiAwal2);
16.    }
17. }
```

3. Selanjutnya jalankanlah kode di atas pada IDE yang kalian gunakan. Bila sukses, seharusnya Console akan menampilkan output seperti ini.

Operator Unary Plus

Hasil +5 = 5

Operator Unary Minus

Hasil -5 = -5

Operator peningkatan nilai sebesar 1 point

Hasil 5++ = 6

Operator pengurangan nilai sebesar 1 point

Hasil 5-- = 4

Operator komplemen logika

Hasil !false = true

## Bedah Code Operasi Unary

### Operasi Unary Plus

```
1. int value = +5;
```

Operator yang menunjukkan bila value yang diberikan operator ini adalah positif. Memang secara default, bilangan selalu positif.

### Operasi Unary Minus

```
1. int value = -5;
```

Seperti halnya Operator Plus, Operator Minus bernilai kebalikannya.

### Operasi Unary Increment

```
1. int value = 5;  
2. value++;
```

Operator yang memberikan penambahan nilai sebesar 1 point untuk value yang di-increment. Operator ini biasa digunakan untuk pengulangan, yang akan dijelaskan pada materi lainnya.

### Operasi Unary Decrement

```
1. int value = 5;  
2. value--;
```

Kebalikan dari Operator di atas, operator decrement memberikan pengurangan nilai sebesar 1 point.

### Operasi Unary Logical Complement

```
1. boolean sukses = false; //Nilai sukses adalah false
2. System.out.println("Hasil !false = " + !sukses);
```

Operator ini akan memberikan nilai kebalikan yang diberikan oleh sebuah boolean seperti pada kode di atas. Awalnya nilai dari sukses bernilai false. Namun ketika dicetak dengan diberikan operator ini, menjadi true.

## Operasi Equality and Relational

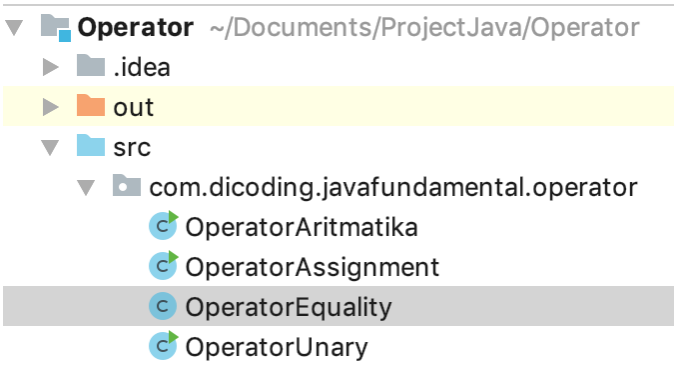
Operator Equality and Relational digunakan untuk menentukan dan membandingkan antara 2 variabel (operand) yang menghasilkan nilai True atau False. Berikut Operator Equality and Relational yang ada pada pemrograman Java :

==	equal to
!=	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

## Codelab Operasi Equality and Relational

Langsung saja kita praktikkan bagaimana menggunakan operator di atas melalui kode di bawah ini:

1. Bukalah proyek Operator dan tambahkan kelas baru didalamnya dengan nama **OperatorEquality**.



2. Tambahkan kode di bawah ini ke dalam kelas **OperatorEquality**:

```
1. package com.dicoding.javafundamental.operator;
2.
3. public class OperatorEquality {
4.
5.     public static void main(String[] args) {
6.
7.         int value = 5;
8.         int anotherValue = 4;
9.         boolean result;
10.        result = value == anotherValue;
11.
12.        System.out.println("Hasil 'value == anotherValue' adalah " + result);
13.        System.out.println();
14.
15.        System.out.println("Tidak sama dengan..");
```

3. Selanjutnya jalankanlah kode di atas pada IDE yang kalian gunakan. Bila sukses, seharusnya Console akan menampilkan output seperti ini.

Hasil 'value == anotherValue' adalah false

Tidak sama dengan..

Hasil 'value != anotherValue' adalah true

Lebih besar dari..

Hasil 'value > anotherValue' adalah true

Sama atau lebih besar dari..

Hasil 'value >= anotherValue' adalah true

Kurang dari..

Hasil 'value < anotherValue' adalah false

Sama atau kurang dari dengan..

Hasil 'value <= anotherValue' adalah false

## Bedah Code Equality and Relational

Perhatikan kode berikut:



1. `int value = 5;`

2. `int anotherValue = 4;`

3.

4. `result = value == anotherValue;`

5. `result = value != anotherValue;`

6. `result = value > anotherValue;`

7. `result = value >= anotherValue;`

8. `result = value < anotherValue;`

9. `result = value <= anotherValue;`

Operator Unary pada Java terlihat pada cuplikan kode di atas. Output-nya menghasilkan nilai berupa *true/false* sesuai kondisi nilai yang diberikan.

## Operasi Conditional

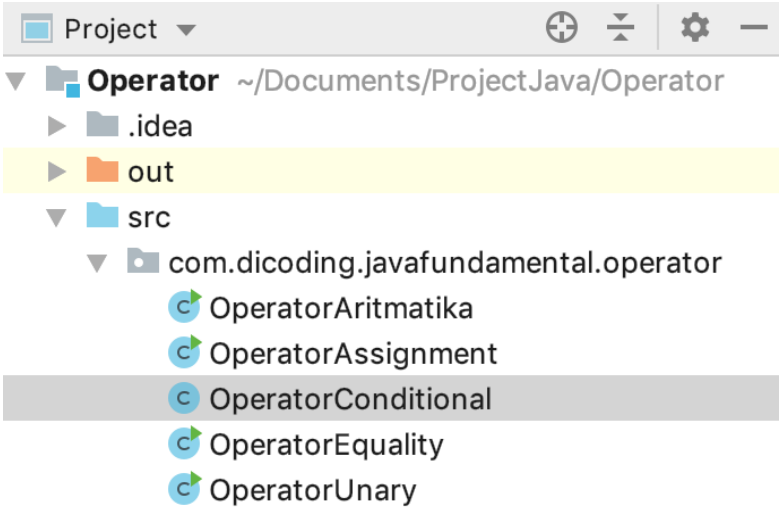
Operator Conditional digunakan pada dua atau lebih ekspresi boolean. Pada operator ini, nilai dari operator kedua, ketiga dst menjadi pertimbangan hasil dari true atau false, bila diperlukan. Hal ini tergantung dari operator Condition yang digunakan, yang jenisnya adalah:

&&	Conditional-AND
	Conditional-OR

### Codelab Operasi Conditional

Mari kita praktikkan bagaimana menggunakan Operasi Condition di atas:

1. Bukalah proyek Operator dan tambahkan kelas baru didalamnya dengan nama `OperatorConditional`.



2. Tambahkan kode di bawah ini ke dalam kelas `OperatorConditional`:

```
1. package com.dicoding.javafundamental.operator;
2.
3. public class OperatorConditional {
4.
5.     public static void main(String[] args) {
6.         int value = 4;
7.         int anotherValue = 5;
8.
9.         System.out.println("Conditional AND");
10.        boolean result = value == 3 && anotherValue == 5;
11.        boolean anotherResult = value != 3 && anotherValue == 5;
12.
13.        System.out.println("Hasil Operator AND pada syarat value == 3 dan anotherValue == 5 adalah " + result);
14.        System.out.println("Hasil Operator AND pada syarat value != 3 dan anotherValue == 5 adalah " + anotherResult);
15.        System.out.println();
```

3. Selanjutnya jalankanlah kode di atas pada IDE yang kalian gunakan. Bila sukses, seharusnya Console akan menampilkan output seperti ini.

Conditional AND

Hasil Operator AND pada syarat value == 3 dan anotherValue == 5 adalah false

Hasil Operator AND pada syarat value != 3 dan anotherValue == 5 adalah true

Conditional OR

Hasil Operator OR pada syarat value == 3 dan anotherValue == 5 adalah true

Hasil Operator OR pada syarat value != 3 dan anotherValue == 5 adalah true

## Bedah Code Operasi Conditional

### Conditional AND

```
1. boolean result = value == 3 && anotherValue == 5;
2. boolean anotherResult = value != 3 && anotherValue == 5;
```

Pada kode yang menerapkan Operator AND di atas ini, terlihat bila ada salah satunya tidak memenuhi syarat yang sudah diberikan, hasilnya adalah nilai false.

### Conditional OR

```
1. result = value == 3 || anotherValue == 5;
2. anotherResult = value != 3 || anotherValue == 5;
```

Berkebalikan dengan operator Conditional AND, Operator OR akan memberikan nilai *true* apabila salah satunya sudah memenuhi syarat.

Jika Anda masih merasa kesulitan, silakan periksa kode dari kami di [Source Code Operator](#).

[← Sebelumnya](#)

[Selanjutnya →](#)