

Java Language

Operator Logika

Operator yang digunakan pada Tipe data **Boolean** untuk menentukan logika antar variabel atau nilai . dapat bernilai true atau false. Ada empat operasi pada operator logika. Diantaranya : AND (&&) , OR (||) , XOR (^) , NAGASI (!).

Keterangan :

Nilai Boolean itu ada dua , **true** dan **false** :

True bernilai -> 1 (2,3,4,--seterusnya) nilai akan di compres menjadi nilai boolean yaitu 1.

False bernilai -> 0

Kita dapat melakukan operasi matematika pada boolena. Contoh pada logika diantanya :

1. AND (&&) melakukan operasi matematika dengan Perkalian. Contoh :

Int a , b,c ;

a	b	c	Output (c)
0	0	$0 * 0 = 0$	False
0	1	$0 * 1 = 0$	False
1	0	$1 * 0 = 0$	False
1	1	$1 * 1 = 1$	True

```
1 package com.tutorial;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         // Operator Logika
7         boolean a,b,c ;
8
9         System.out.println("==== AND (&&) ====");
10        a = false ;
11        b = false ;
12        c = (a && b) ;
13        System.out.println(a + " && " + b + " = " + c );
14        a = false ;
15        b = true ;
16        c = (a && b) ;
17        System.out.println (a + " && " + b + " = " + c );
18        a = true ;
19        b = false ;
20        c = (a && b) ;
21        System.out.println(a + " && " + b + " = " + c );
22        a = true ;
23        b = true ;
24        c = (a && b) ;
25        System.out.println(a + " && " + b + " = " + c );
26
27    }
28 }
29
30
```

Run: Main x

```
"C:\Program Files\Java\jdk-10.0.2\bin\java.exe" "-
==== AND (&&) ====
false && false = false
false && true = false
true && false = false
true && true = true

Process finished with exit code 0
```

Keterangan : memberikan nilai true jika kedua nilai bernilai true. Tetapi akan bernilai false jika salah satu nilai bernilai false. Contoh pada program :

```

public class Main {
    public static void main(String[] args) {
        //Contoh :

        int nilaiAND = 10 ;
        //kedua nilai perbandingan bernilai true (benar) -> true
        System.out.println( nilaiAND < 15 && nilaiAND > 5 );

        //salah satu nilai bernilai false - > false
        System.out.println(nilaiAND > 15 && nilaiAND > 5 );
    }
}

```

true
false

2. OR (||) melakukan operasi matematika dengan penjumlahan

Int a,b,c ;

a	b	c	Output (c)
0	0	0 + 0 = 0	False
0	1	0 + 1 = 1	True
1	0	1 + 0 = 1	True
1	1	1 + 1 = 2	True

```

package com.tutorial;

public class Main {
    public static void main(String[] args) {
        // Operator Logika
        boolean a,b,c ;

        //=====OR=====

        System.out.println("==== OR ( || ) ====");
        a = false ;
        b = false ;
        c = (a || b) ;
        System.out.println(a + " || " + b + " = " + c );
        a = false ;
        b = true ;
        c = (a || b) ;
        System.out.println(a + " || " + b + " = " + c );
        a = true ;
        b = false ;
        c = (a || b) ;
        System.out.println(a + " || " + b + " = " + c );
        a = true ;
        b = true ;
        c = (a || b) ;
        System.out.println(a + " || " + b + " = " + c );
    }
}

```

Run: "C:\Program Files\Java\jdk-10.0.2\bin\java.exe" "-javaag
==== OR (||) ====
false || false = false
false || true = true
true || false = true
true || true = true
Process finished with exit code 0

Keterangan : memberikan nilai false jika kedua nilai bernilai false. Tetapi akan bernilai true jika salah satu nilai bernilai true. Contoh pada program :

```

public class Main {
    public static void main(String[] args) {

        //Contoh :
        int nilaiOR = 10 ;
        //kedua nilai perbandingan bernilai false (salah) -> false
        System.out.println(nilaiOR < 5 || nilaiOR > 15 );

        //salah satu nilai bernilai true (benar) - > true
        System.out.println(nilaiOR > 15 || nilaiOR > 5 );
    }
}

```

false
true

3. XOR (^) exclusive OR : akan bernilai true jika kedua nilai berbeda. Tetapi akan bernilai false jika kedua nilai bernilai sama. Contoh :

Int a,b,c ;

a	b	c	Output (c)
0	0	0	False
0	1	1	True
1	0	1	True
1	1	0	False

```

package com.tutorial;

public class Main {

    public static void main(String[] args) {

        // Operator Logika
        boolean a,b,c ;

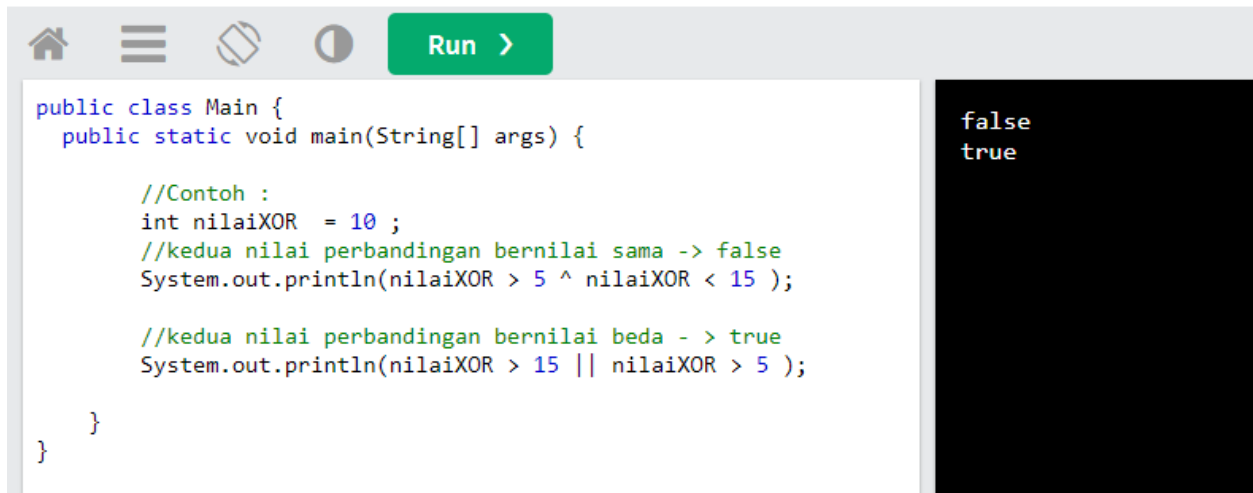
        //=====XOR

        System.out.println("==== XOR ( ^ ) ====");
        a = false ;
        b = false ;
        c = (a ^ b) ;
        System.out.println(a + " ^ " + b + " = " + c);
        a = false ;
        b = true ;
        c = (a ^ b) ;
        System.out.println(a + " ^ " + b + " = " + c);
        a = true ;
        b = false ;
        c = (a ^ b) ;
        System.out.println(a + " ^ " + b + " = " + c);
        a = true ;
        b = true ;
        c = (a ^ b) ;
        System.out.println(a + " ^ " + b + " = " + c);
    }
}

```

Run: "C:\Program Files\Java\jdk-10.0.2\bin\java.exe" "-javaagent:..."
 ==== XOR (^) ====
 false ^ false = false
 false ^ true = true
 true ^ false = true
 true ^ true = false
 Process finished with exit code 0

Contoh Program :



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

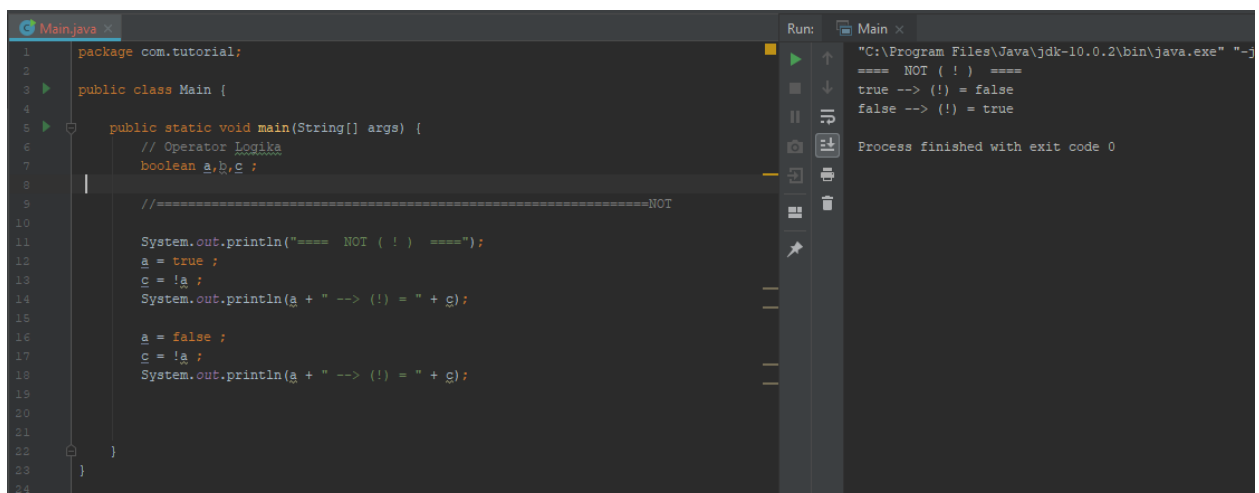
        //Contoh :
        int nilaiXOR = 10 ;
        //kedua nilai perbandingan bernilai sama -> false
        System.out.println(nilaiXOR > 5 ^ nilaiXOR < 15 );

        //kedua nilai perbandingan bernilai beda - > true
        System.out.println(nilaiXOR > 15 || nilaiXOR > 5 );

    }
}
```

false
true

4. Not atau NAGASI(!) : membalikan nilai. Dari true menjadi false dan dari false menjadi true



```
package com.tutorial;

public class Main {

    public static void main(String[] args) {
        // Operator Logika
        boolean a,b,c ;

        //=====NOT

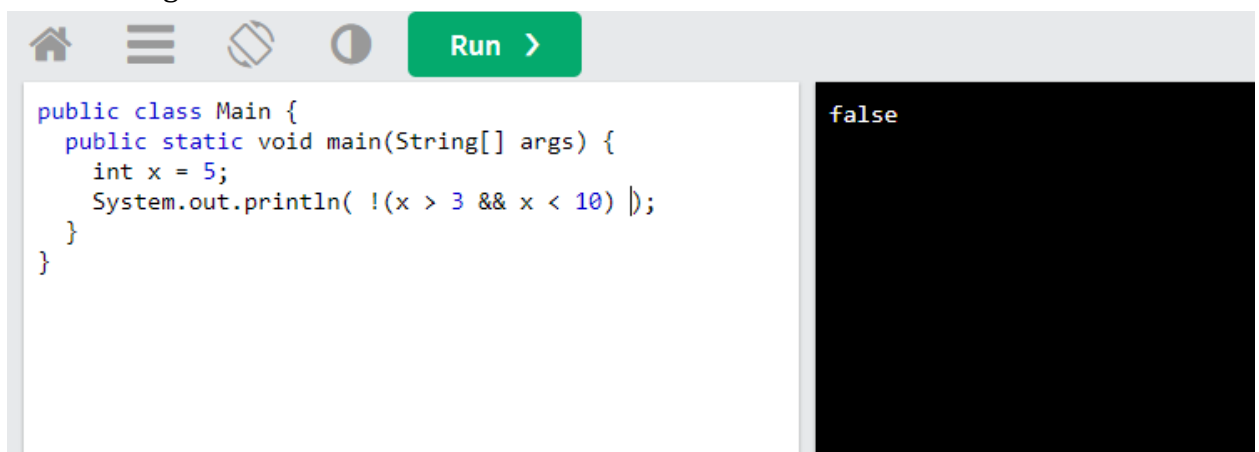
        System.out.println("==== NOT ( ! ) =====");
        a = true ;
        c = !a ;
        System.out.println(a + " --> (!) = " + c);

        a = false ;
        c = !a ;
        System.out.println(a + " --> (!) = " + c);

    }
}
```

==== NOT (!) ====
true --> (!) = false
false --> (!) = true
Process finished with exit code 0

Contoh Program :



```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        System.out.println( !(x > 3 && x < 10) );
    }
}
```

false