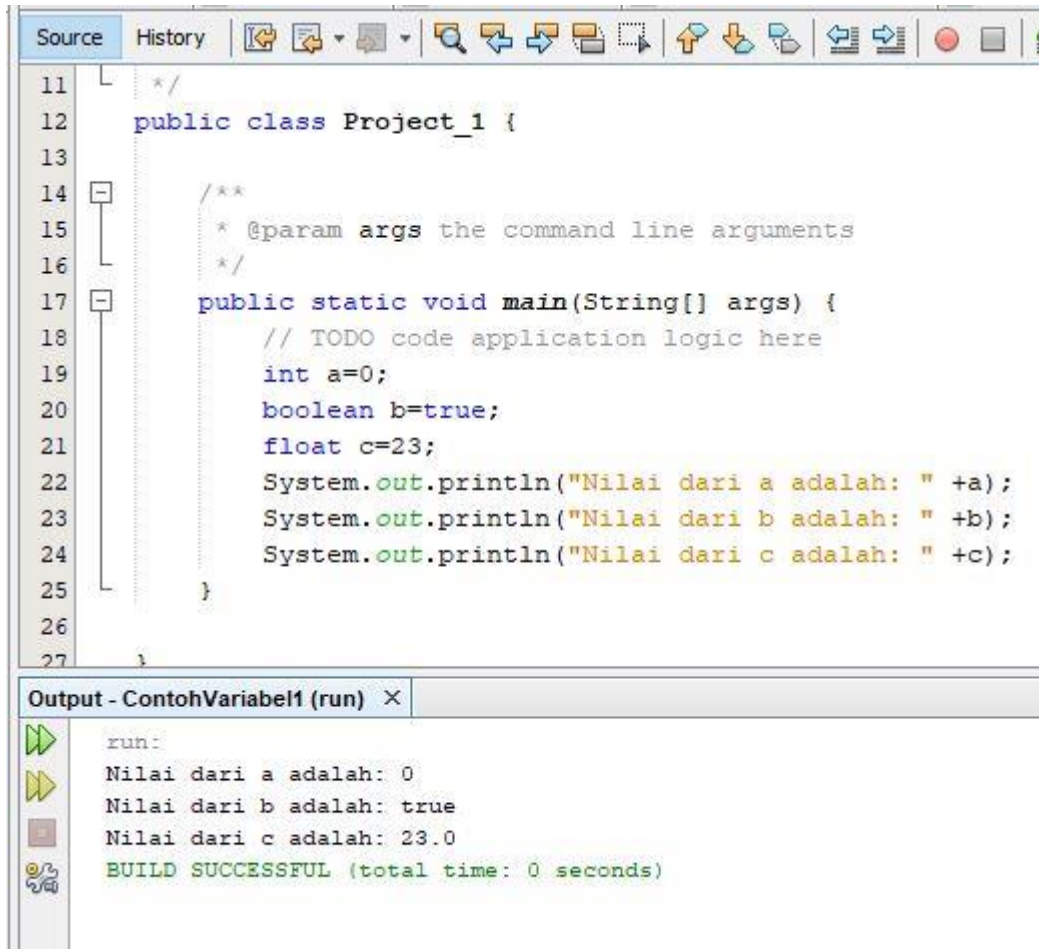


Nama : AHMAD RUDianto

Nim : 21157201114

VARIABEL



```
11  */
12  public class Project_1 {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          int a=0;
20          boolean b=true;
21          float c=23;
22          System.out.println("Nilai dari a adalah: " +a);
23          System.out.println("Nilai dari b adalah: " +b);
24          System.out.println("Nilai dari c adalah: " +c);
25      }
26
27  }
```

Output - ContohVariabel1 (run) ×

```
run:
Nilai dari a adalah: 0
Nilai dari b adalah: true
Nilai dari c adalah: 23.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Sebelum membuat variabel di atas, lebih baik membuat string terlebih dahulu.

Dengan menggunakan tipe data int,boolean dan float

sebagai contoh

int a=0

boolean b=true;

float c=23;

Fungsi dari int (integer) adalah untuk mendefinisikan variable numerik seluruh nomor yang tidak menggunakan koma.

Fungsi dari boolean adalah suatu tipe data yang hanya mempunyai dua nilai, yaitu true dan false (benar dan salah).

Fungsi float adalah sebuah format bilangan yang dapat digunakan untuk mempresentasikan hasil sebuah nilai sangat besar atau sangat kecil.

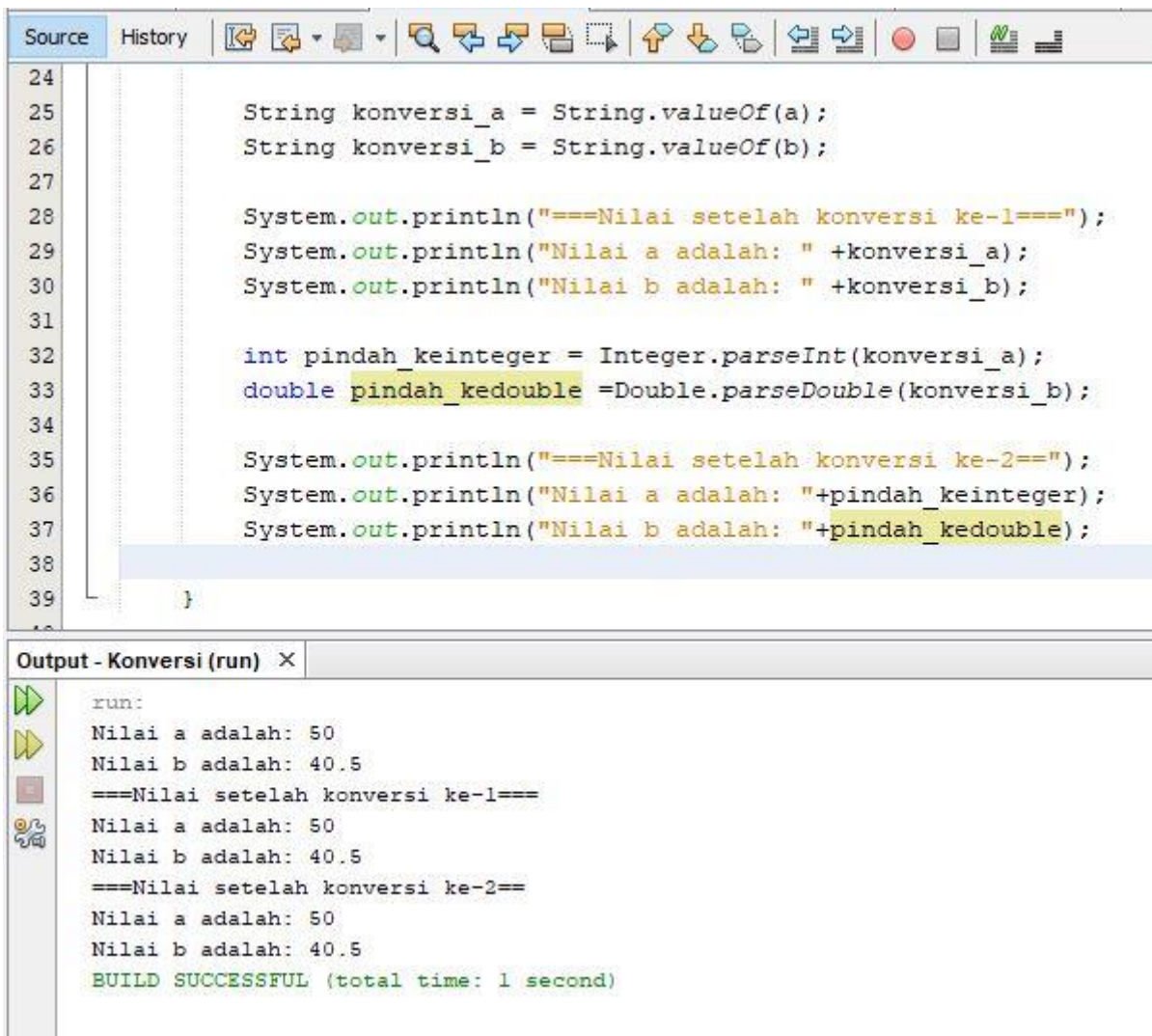
Setelah itu kita membuat kode program yang menghasilkan output contoh di bawah ini, setelah itu Run

```
System.out.println("Nilai dari A adalah " +a);
```

```
System.out.println("Nilai dari A adalah " +b);
```

```
System.out.println("Nilai dari A adalah " +c);
```

CONVERSI



The screenshot shows an IDE with a Java source file and its output. The source code defines variables `a` and `b`, converts them to strings, then back to integer and double types, and prints the results at each stage. The output window shows the execution of this code, displaying the original values, the string conversion, and the subsequent parsing back to numeric types.

```
24
25     String konversi_a = String.valueOf(a);
26     String konversi_b = String.valueOf(b);
27
28     System.out.println("===Nilai setelah konversi ke-1===");
29     System.out.println("Nilai a adalah: " +konversi_a);
30     System.out.println("Nilai b adalah: " +konversi_b);
31
32     int pindah_keinteger = Integer.parseInt(konversi_a);
33     double pindah_kedouble =Double.parseDouble(konversi_b);
34
35     System.out.println("===Nilai setelah konversi ke-2===");
36     System.out.println("Nilai a adalah: "+pindah_keinteger);
37     System.out.println("Nilai b adalah: "+pindah_kedouble);
38
39 }
```

Output - Konversi (run) X

```
run:
Nilai a adalah: 50
Nilai b adalah: 40.5
===Nilai setelah konversi ke-1===
Nilai a adalah: 50
Nilai b adalah: 40.5
===Nilai setelah konversi ke-2===
Nilai a adalah: 50
Nilai b adalah: 40.5
BUILD SUCCESSFUL (total time: 1 second)
```

Sama seperti di percobaan 1, cuman saya akan menjelaskan fungsi dari double dan string konveersi

Fungsi dari double adalah tipe data untuk angka, sama seperti int, tapi Double dapat menggunakan koma, Dan harus di kasih perintah input data .

Contoh:

```
System.out.println("nilai a adalah: " +a);
```

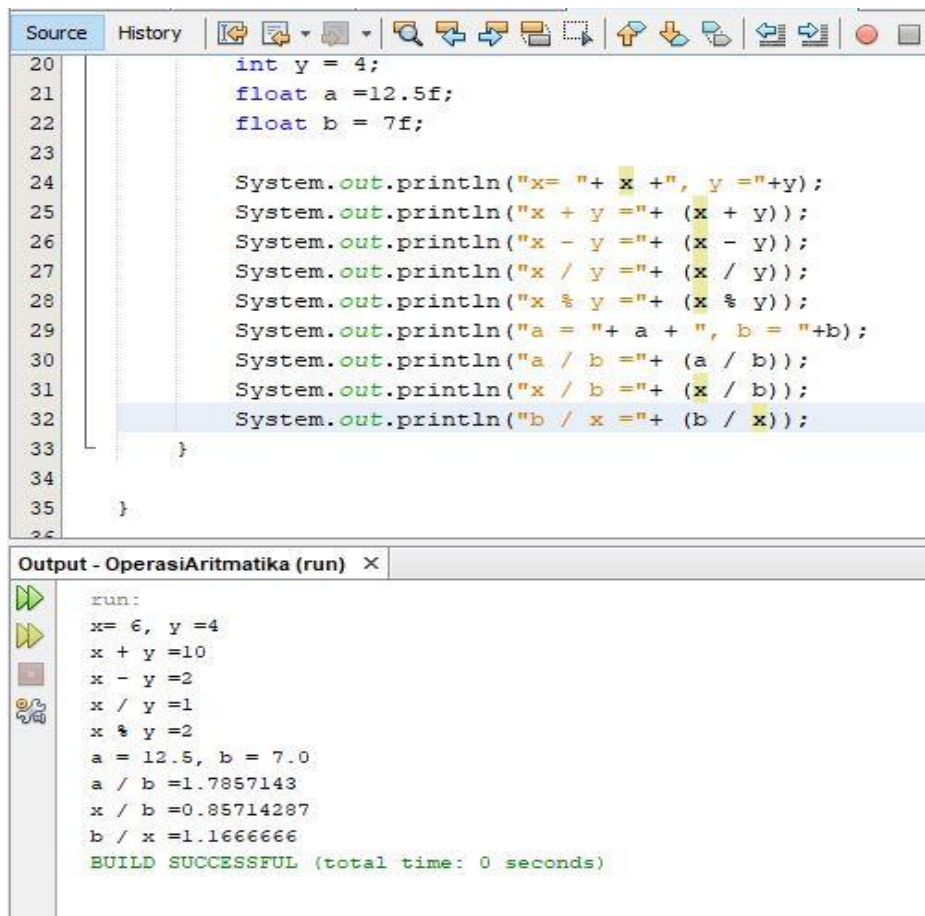
```
System.out.println("nilai b adalah: " +b);
```

```
String konversi_a = String.valueOf(a); // konfersi dari integer ke string
```

```
String konversi_b = String.valueOf(b); // konfersi dari double ke string
```

Dan di lanjutkan hitung nilai setelan konversi seperti di atas.

OPERATOR ARITMATIKA



```
Source History
20      int y = 4;
21      float a =12.5f;
22      float b = 7f;
23
24      System.out.println("x= " + x +", y =" +y);
25      System.out.println("x + y =" + (x + y));
26      System.out.println("x - y =" + (x - y));
27      System.out.println("x / y =" + (x / y));
28      System.out.println("x % y =" + (x % y));
29      System.out.println("a = " + a + ", b = " +b);
30      System.out.println("a / b =" + (a / b));
31      System.out.println("x / b =" + (x / b));
32      System.out.println("b / x =" + (b / x));
33  }
34
35  }
```

Output - OperasiAritmatika (run) ×

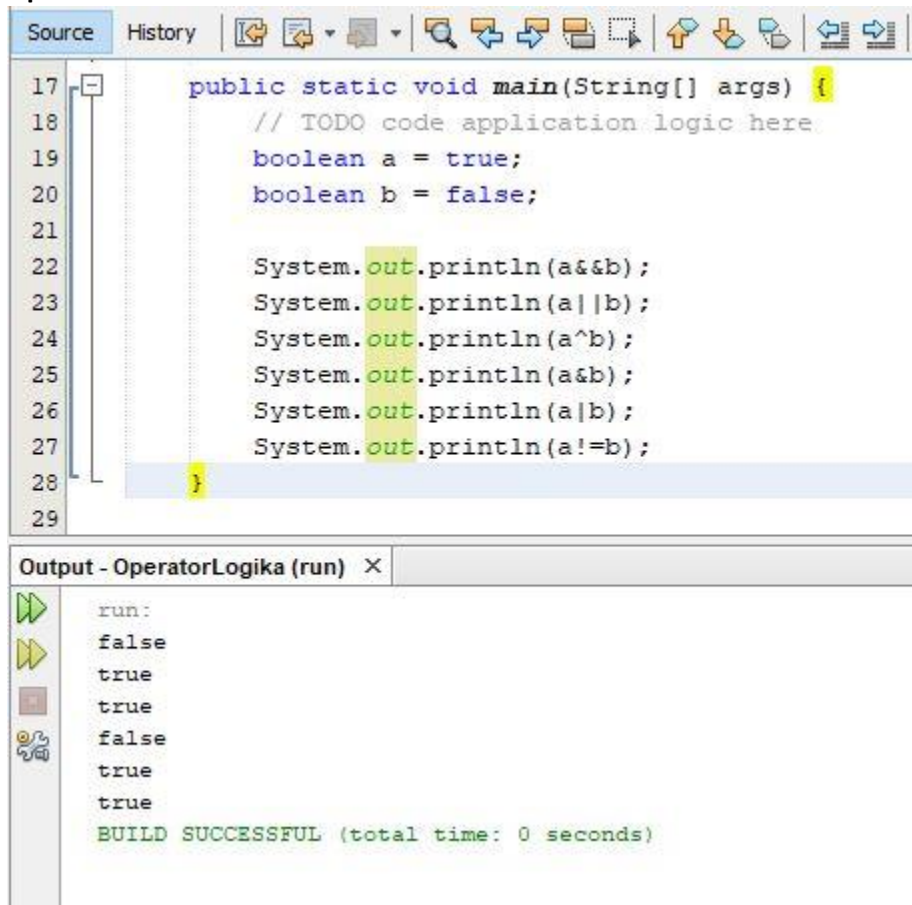
```
run:
x= 6, y =4
x + y =10
x - y =2
x / y =1
x % y =2
a = 12.5, b = 7.0
a / b =1.7857143
x / b =0.85714287
b / x =1.16666666
BUILD SUCCESSFUL (total time: 0 seconds)
```

Percobaan selanjutnya, yaitu mengenai penjumlahan, pengurangan, perkalian, pembagian, pada jenis data short, int, float.

Short yang berfungsi digunakan pada aplikasi khusus yang berkaitan dengan memory.

Contoh string pada gambar di atas.

Operator



The screenshot shows an IDE with a Java source file. The code defines a `main` method that initializes two boolean variables, `a` and `b`, and then prints the results of various logical and relational operations. Below the code, the 'Output' window shows the execution results, which are the boolean values `false`, `true`, `true`, `false`, `true`, `true`, and a successful build message.

```
17 public static void main(String[] args) {
18     // TODO code application logic here
19     boolean a = true;
20     boolean b = false;
21
22     System.out.println(a&&b);
23     System.out.println(a||b);
24     System.out.println(a^b);
25     System.out.println(a&b);
26     System.out.println(a|b);
27     System.out.println(a!=b);
28 }
29
```

Output - OperatorLogika (run) ×

```
run:
false
true
true
false
true
true
BUILD SUCCESSFUL (total time: 0 seconds)
```

Pada percobaan yang ke 4 ini adalah Operator Relasi, yaitu membandingkan 2 nilai dan menentukan keterhubungan diantara nilai nilai tersebut. Hasil keluarannya berupa nilai Boolean yaitu true dan false.

Dan di beri simbol :

Arti dari symbol :

`==` adalah Sama dengan

`>` adalah lebih besar

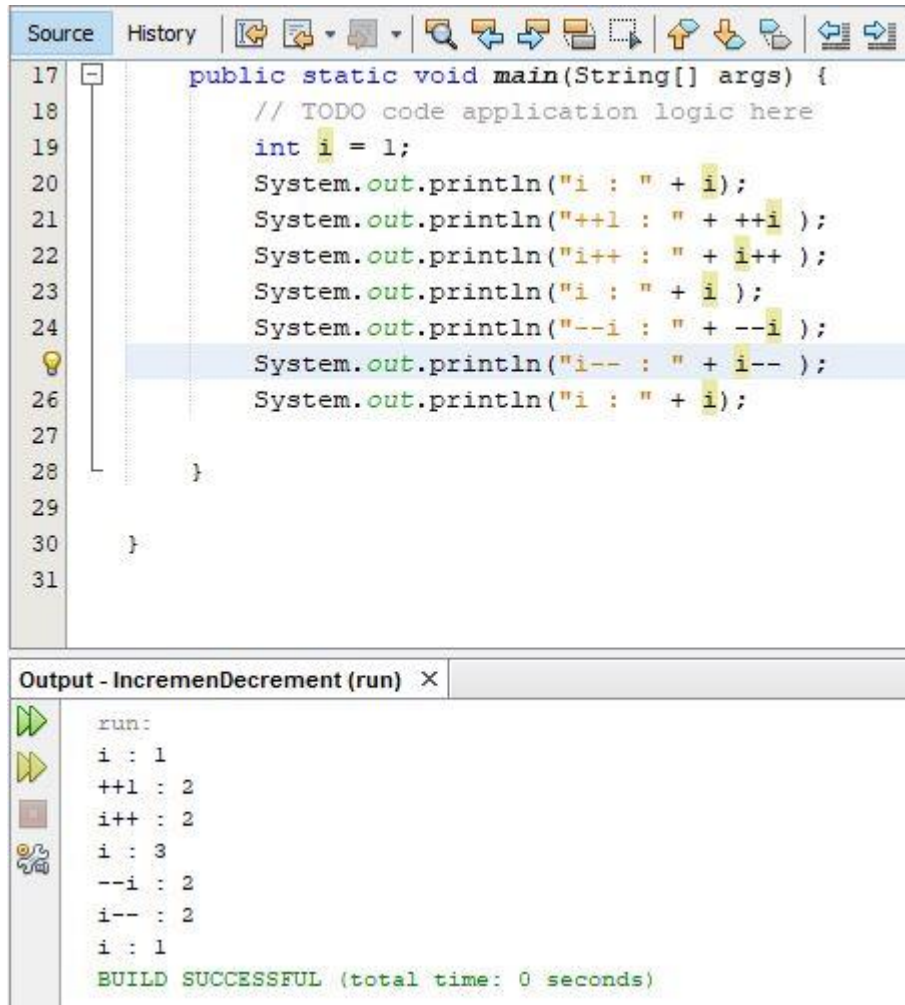
`!=` adalah tidak sama

`<=` adalah lebih kecil atau sama dengan

`<` adalah lebih kecil

`>=` adalah lebih besar atau sama dengan

IncrementDecrement



The screenshot shows an IDE with a Java source file. The code defines a `main` method that initializes `i` to 1 and prints its value. It then demonstrates pre-increment (`++i`), post-increment (`i++`), pre-decrement (`--i`), and post-decrement (`i--`) operations, printing the value of `i` after each operation. Finally, it prints the value of `i` again. The output window shows the results of these operations.

```
17 public static void main(String[] args) {  
18     // TODO code application logic here  
19     int i = 1;  
20     System.out.println("i : " + i);  
21     System.out.println("++i : " + ++i );  
22     System.out.println("i++ : " + i++ );  
23     System.out.println("i : " + i );  
24     System.out.println("--i : " + --i );  
25     System.out.println("i-- : " + i-- );  
26     System.out.println("i : " + i);  
27  
28 }  
29  
30 }  
31
```

Output - IncremenDecrement (run) X

```
run:  
i : 1  
++i : 2  
i++ : 2  
i : 3  
--i : 2  
i-- : 2  
i : 1  
BUILD SUCCESSFUL (total time: 0 seconds)
```

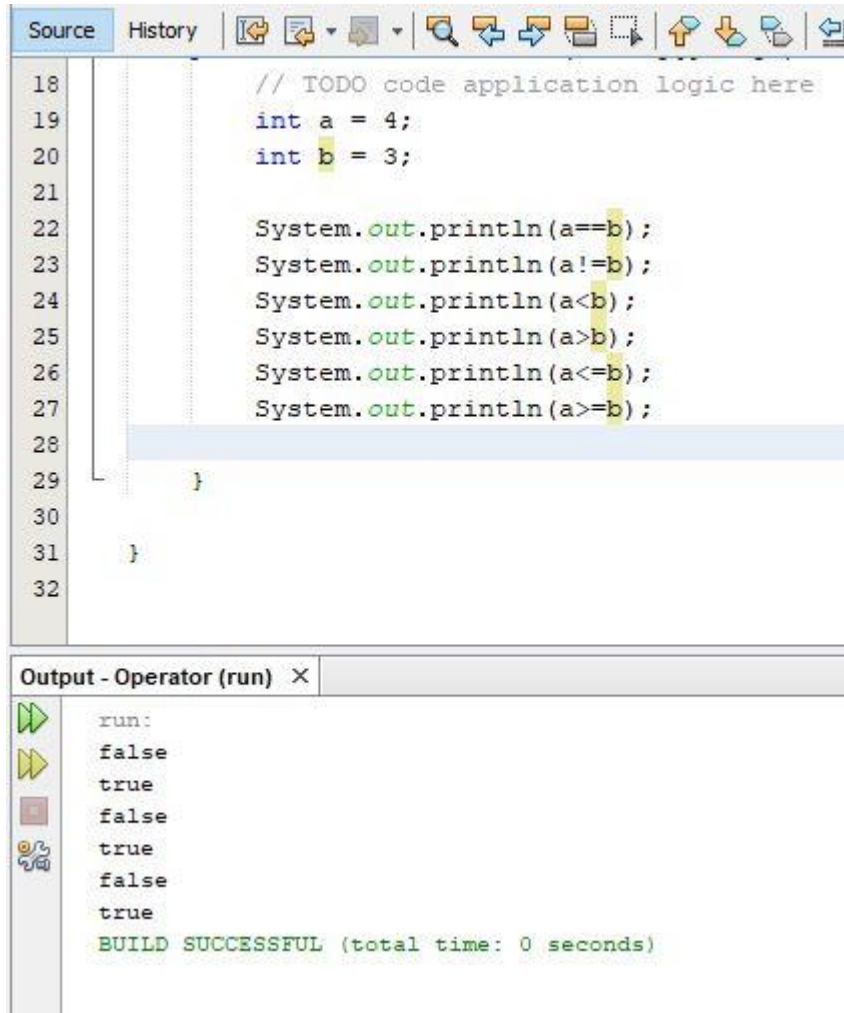
Pada percobaan yang ke 4 ini adalah incrementDecrement, yaitu

`++a` yaitu Pre-decremenet yaitu nilai a ditambah 1,kemudian operasi berjalan

`a++` yaitu Post-decrement yaitu operasi dilaksanakan kemudian a ditambah 1

`—a` yaitu Pre-decrement yaitu nilai a dikurang 1,kemudian operasi dilaksanakan 1

OPERATOR LOGIKA



The screenshot shows an IDE with a source code editor and an output window. The source code defines two integers, `a = 4` and `b = 3`, and prints the results of seven logical and relational expressions. The output window shows the results of these expressions in sequence: `run:`, `false`, `true`, `false`, `true`, `false`, `true`, and a final status message `BUILD SUCCESSFUL (total time: 0 seconds)`.

```
18 // TODO code application logic here
19 int a = 4;
20 int b = 3;
21
22 System.out.println(a==b);
23 System.out.println(a!=b);
24 System.out.println(a<b);
25 System.out.println(a>b);
26 System.out.println(a<=b);
27 System.out.println(a>=b);
28
29 }
30
31 }
32
```

Output - Operator (run) X

```
run:
false
true
false
true
false
true
BUILD SUCCESSFUL (total time: 0 seconds)
```

Pada percobaan yang terakhir ini, operator logika memiliki satu atau lebih operand Boolean yang menghasilkan nilai Boolean yaitu

`A & b` yaitu AND adalah benar jika `a` dan `b` benar

`a | b` yaitu OR adalah benar jika `a` atau `b` benar

`a ^ b` yaitu XOR adalah benar jika `a` atau `b` benar tetapi tidak keduanya.