

**LAPORAN**  
**PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Muhammad Andi Syaifullah  
Nim : 13020220015  
Dosen : Mardiyah Hasnawi, S.Kom.,M.T.


**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS MUSLIM INDONESIA**  
**MAKASSAR**

**2024**

- **Program 1**

**-OutPut Program class Asgdll Dan Penjelasan Programnya**

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Asgdll.java
f : 20.0
f11: 10.0
```



Mari kita bahas setiap bagian dari program tersebut:

**1. Tujuan Program:**

Program ini memiliki tujuan sederhana untuk menginisialisasi dua variabel floating-point (`f` dan `f11`), kemudian mencetak nilai keduanya. Ini memberikan pemahaman dasar tentang deklarasi, inisialisasi, dan penggunaan variabel dalam bahasa pemrograman Java.

**2. Kamus (Kamus merupakan istilah dalam Bahasa Indonesia yang digunakan untuk menyatakan variabel):**

- `float f = 20.0f;`: Mendeklarasikan dan menginisialisasi variabel `f` dengan tipe data float dan nilai awal 20.0f.
- `double f11;`: Mendeklarasikan variabel `f11` dengan tipe data double tanpa memberikan nilai awal.

**3. Algoritma:**

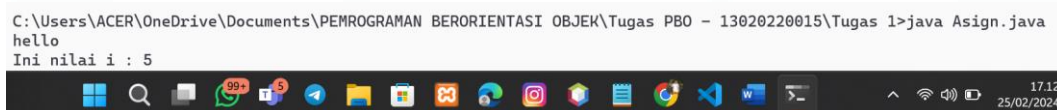
- `f11 = 10.0f;`: Menginisialisasi variabel `f11` dengan nilai 10.0f. (Catatan: Ada kesalahan kecil di sini, karena `f11` dideklarasikan sebagai double, seharusnya nilai ini adalah 10.0 tanpa 'f' atau `f11` perlu dideklarasikan sebagai float).
- `System.out.println("f: " + f + "\nf11: " + f11);`: Mencetak nilai variabel `f` dan `f11` ke konsol.

**4. Keyword yang Digunakan / Tipe Data:**

- `public`: Kata kunci yang menunjukkan bahwa method (`main` dalam hal ini) dapat diakses dari mana saja.
- `static`: Kata kunci yang menunjukkan bahwa metode ini bersifat statis, artinya dapat dipanggil tanpa menginstansiasi objek dari kelasnya.
- `void`: Tipe data yang menunjukkan bahwa metode `main` tidak mengembalikan nilai.
- `float`: Tipe data yang digunakan untuk menyimpan nilai desimal dengan presisi yang lebih rendah dibandingkan dengan `double`.
- `double`: Tipe data yang digunakan untuk menyimpan nilai desimal dengan presisi yang lebih tinggi dibandingkan dengan `float`.
- `System.out.println`: Metode untuk mencetak output ke konsol.

## • Program 2

### -OutPut Program class Assign Dan Penjelasan Programnya



```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Assign.java
hello
Ini nilai i : 5
```

#### 1. Tujuan Program:

Program ini memiliki tujuan sederhana untuk mendeklarasikan sebuah variabel `i`, memberinya nilai 5, dan kemudian mencetak nilai tersebut ke konsol. Ini bertujuan untuk memberikan pemahaman tentang deklarasi, inisialisasi, dan penggunaan variabel dalam bahasa pemrograman Java.

#### 2. Keyword yang Digunakan / Tipe Data:

- `public class Assign`: Mendeklarasikan kelas dengan nama "Assign". Kata kunci `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa

metode ini bersifat statis, dan `void` menunjukkan bahwa metode ini tidak mengembalikan nilai.

- `int i;`: Mendeklarasikan variabel `i` dengan tipe data int (integer).

### 3. Algoritma:

- `System.out.print("hello\n");`: Mencetak string "hello" tanpa baris baru (newline).

- `i = 5;`: Memberikan nilai 5 pada variabel `i`.

- `System.out.println("Ini nilai i : " + i);`: Mencetak nilai variabel `i` dengan pesan tambahan ke konsol.

### 4. Output:

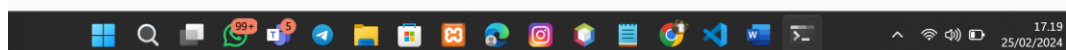
- Program ini akan mencetak "hello" tanpa baris baru, dan kemudian mencetak "Ini nilai i : 5" pada baris berikutnya.

Program ini dirancang untuk memperkenalkan konsep dasar deklarasi variabel, inisialisasi, dan penggunaan `System.out.println` untuk mencetak output di Java.

## • Program 3

-OutPut Program class ASIGNi Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java ASIGNi.java
Karakter = A
Karakter = Z
Karakter = A
Karakter = Z
Bilangan integer (short) = 1
(int) = 1
(long)= 10000
Bilangan Real x = 50.20000076293945
Bilangan Real y = 50.2
```



### 1. Tujuan Program:

Program ini bertujuan untuk mendeklarasikan beberapa variabel dengan tipe data yang berbeda, memberikan nilai pada variabel-variabel tersebut, dan kemudian mencetak nilai-nilai tersebut ke konsol. Ini memberikan pemahaman tentang deklarasi, inisialisasi, dan penggunaan variabel dengan tipe data yang berbeda dalam bahasa pemrograman Java.

## 2. Keyword yang Digunakan / Tipe Data:

- `public class ASIGNi`: Mendeklarasikan kelas dengan nama "ASIGNi". `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini bersifat statis, dan `void` menunjukkan bahwa metode ini tidak mengembalikan nilai.
- Tipe Data:
  - `short`: Tipe data untuk bilangan bulat pendek (16-bit).
  - `int`: Tipe data untuk bilangan bulat (32-bit).
  - `long`: Tipe data untuk bilangan bulat panjang (64-bit).
  - `char`: Tipe data untuk karakter (16-bit Unicode).
  - `double`: Tipe data untuk bilangan real (64-bit).
  - `float`: Tipe data untuk bilangan real dengan presisi rendah (32-bit).

## 3. Algoritma:

- `short ks = 1;`, `int ki = 1;`, `long kl = 10000;`: Mendeklarasikan dan menginisialisasi variabel dengan tipe data short, int, dan long.
- `char c = 65;`: Menginisialisasi variabel karakter dengan nilai ASCII 65 (mewakili karakter 'A').

- `char c1 = 'Z';`: Mendeklarasikan dan menginisialisasi variabel karakter dengan nilai 'Z'.
- `double x = 50.2f;`, `float y = 50.2f;`: Mendeklarasikan dan menginisialisasi variabel dengan tipe data double dan float.
- Mencetak nilai variabel-variabel ke konsol menggunakan `System.out.println`.

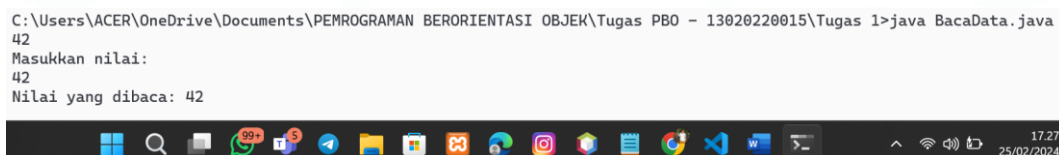
#### 4. Output:

- Program ini akan mencetak nilai-nilai variabel ke konsol.

Program ini dirancang untuk memberikan pemahaman tentang berbagai tipe data dan cara menginisialisasi serta mencetak nilai-nilai variabel di Java.

#### • Program 4

##### -OutPut Program class BacaData Dan Penjelasan Programnya



```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java BacaData.java
42
Masukkan nilai:
42
Nilai yang dibaca: 42
```

#### 1. Tujuan Program:

Program ini bertujuan untuk membaca nilai dari pengguna menggunakan kelas `Scanner` dan mencetak nilai yang dibaca ke konsol. Ini memberikan pemahaman tentang penggunaan kelas `Scanner` untuk mendapatkan input dari pengguna dalam bahasa pemrograman Java.

#### 2. Keyword yang Digunakan / Tipe Data:

- `import java.util.Scanner;`: Mengimpor kelas `Scanner` dari pustaka utilitas Java untuk digunakan dalam program.

- `public class BacaData`: Mendeklarasikan kelas dengan nama "BacaData". Kata kunci `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini bersifat statis, dan `void` menunjukkan bahwa metode ini tidak mengembalikan nilai.
- `int a`: Mendeklarasikan variabel `a` dengan tipe data `int` (integer).
- `Scanner masukan`: Mendeklarasikan variabel `masukan` dengan tipe data `Scanner`.

### 3. Algoritma:

- `System.out.println(42)`: Mencetak angka 42 ke konsol.
- `masukan = new Scanner(System.in)`: Membuat objek `Scanner` baru untuk membaca input dari `System.in` (konsol).
- `System.out.println("Masukkan nilai: ")`: Mencetak pesan meminta pengguna untuk memasukkan nilai.
- `a = masukan.nextInt()`: Membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `a`.
- `System.out.println("Nilai yang dibaca: " + a)`: Mencetak nilai yang dibaca dari pengguna ke konsol.

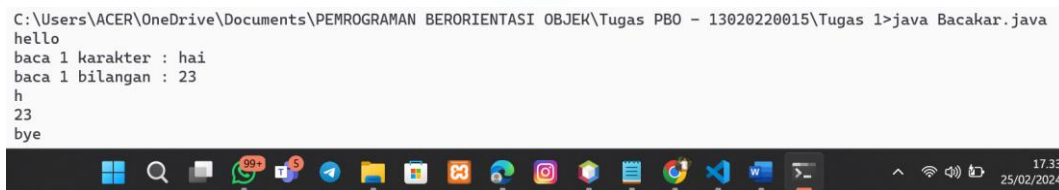
### 4. Output:

- Program ini akan mencetak angka 42, meminta pengguna untuk memasukkan nilai, dan kemudian mencetak nilai yang dimasukkan oleh pengguna.

Program ini dirancang untuk memberikan contoh penggunaan kelas `Scanner` untuk membaca input dari pengguna dalam bentuk bilangan bulat di Java.

- **Program 5**

-OutPut Program class Bacakar Dan Penjelasan Programnya



```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Bacakar.java
hello
baca 1 karakter : hai
baca 1 bilangan : 23
h
23
bye
```

1. Tujuan Program:

Program ini bertujuan untuk membaca satu karakter dan satu bilangan dari pengguna, kemudian mencetak nilai-nilai tersebut ke konsol. Program ini memberikan contoh penggunaan kelas `BufferedReader` untuk membaca input dari pengguna dalam bentuk karakter dan bilangan bulat di Java.

2. Keyword yang Digunakan / Tipe Data:

- `import java.io.BufferedReader;`: Mengimpor kelas `BufferedReader` dari pustaka input-output Java.
- `import java.io.IOException;`: Mengimpor kelas `IOException` untuk menangani eksepsi yang dapat terjadi saat bekerja dengan input-output.
- `import java.io.InputStreamReader;`: Mengimpor kelas `InputStreamReader` untuk mengonversi byte input stream menjadi karakter.
- `public class Bacakar`: Mendeklarasikan kelas dengan nama "Bacakar". Kata kunci `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.



- `public static void main(String[] args) throws IOException`: Mendeklarasikan metode `main` dan menyatakan bahwa metode ini dapat melempar eksepsi `IOException`.
- `char cc; int bil`;: Mendeklarasikan variabel `cc` dengan tipe data `char` dan `bil` dengan tipe data `int`.
- `InputStreamReader isr = new InputStreamReader(System.in)`;: Membuat objek `InputStreamReader` untuk membaca input dari `System.in` (konsol).
- `BufferedReader dataIn = new BufferedReader(isr)`;: Membuat objek `BufferedReader` untuk membaca karakter dari input stream.
- `BufferedReader datAIn = new BufferedReader(new InputStreamReader(System.in))`;: Membuat objek `BufferedReader` untuk membaca karakter dari input stream. (Ini tampaknya tidak digunakan dalam program).

### 3. Algoritma:

- `System.out.print("hello\n")`;: Mencetak pesan "hello" ke konsol.
- `System.out.print("baca 1 karakter : ")`;: Mencetak pesan meminta pengguna untuk memasukkan satu karakter.
- `cc = dataIn.readLine().charAt(0)`;: Membaca satu karakter dari pengguna dan menyimpannya dalam variabel `cc`.
- `System.out.print("baca 1 bilangan : ")`;: Mencetak pesan meminta pengguna untuk memasukkan satu bilangan.
- `bil = Integer.parseInt(datAIn.readLine())`;: Membaca satu bilangan bulat dari pengguna dan menyimpannya dalam variabel `bil`.
- `System.out.print(cc + "\n" + bil + "\n")`;: Mencetak nilai karakter dan bilangan yang dibaca ke konsol.
- `System.out.print("bye \n")`;: Mencetak pesan "bye" ke konsol.

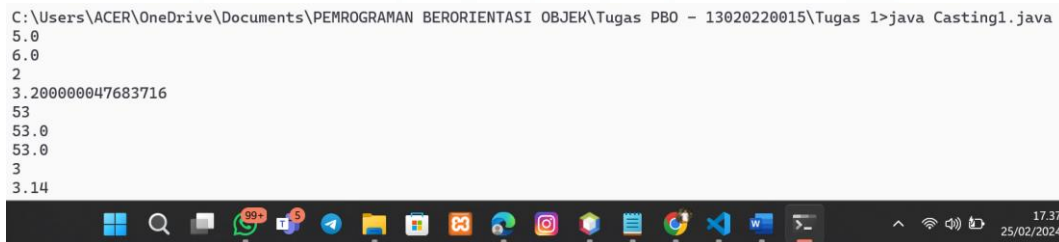
#### 4. Output:

- Program ini akan mencetak pesan, meminta pengguna untuk memasukkan satu karakter dan satu bilangan, kemudian mencetak nilai-nilai tersebut ke konsol.

Program ini memberikan contoh penggunaan kelas `BufferedReader` untuk membaca input dari pengguna dalam bentuk karakter dan bilangan bulat di Java.

- **Program 6**

-OutPut Program class Casting1 Dan Penjelasan Programnya



```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Casting1.java
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14
```

#### Penjelasan Program:

Program ini bertujuan untuk menunjukkan contoh penggunaan *\*casting\** (konversi tipe data) di dalam Java. Berbagai nilai dengan tipe data yang berbeda di-*\*cast\** ke tipe data lainnya, dan hasilnya dicetak ke konsol.

#### Keyword yang Digunakan / Tipe Data:

- ``public class Casting1``: Mendeklarasikan kelas dengan nama "Casting1". ``public`` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- ``public static void main(String[] args)``: Mendeklarasikan metode ``main``, yang merupakan titik awal eksekusi program. ``public`` menunjukkan bahwa metode ini dapat diakses dari luar kelas, ``static`` menunjukkan bahwa metode ini bersifat statis, dan ``void`` menunjukkan bahwa metode ini tidak mengembalikan nilai.

- `int a=5, b=6;`: Mendeklarasikan dan menginisialisasi variabel `a` dan `b` dengan tipe data `int`.
- `float d=2.5, e=3.2;`: Mendeklarasikan dan menginisialisasi variabel `d` dan `e` dengan tipe data `float`.
- `char g='5';`: Mendeklarasikan dan menginisialisasi variabel `g` dengan tipe data `char`.
- `double k=3.14;`: Mendeklarasikan dan menginisialisasi variabel `k` dengan tipe data `double`.

#### Algoritma:

- `System.out.println((float)a);`: Melakukan *casting* dari `int` ke `float` dan mencetak hasilnya ke konsol.
- `System.out.println((double)b);`: Melakukan *casting* dari `int` ke `double` dan mencetak hasilnya ke konsol.
- `System.out.println((int)d);`: Melakukan *casting* dari `float` ke `int` dan mencetak hasilnya ke konsol.
- `System.out.println((double)e);`: Melakukan *casting* dari `float` ke `double` dan mencetak hasilnya ke konsol.
- `System.out.println((int)g);`: Melakukan *casting* dari `char` ke `int` (berdasarkan nilai ASCII) dan mencetak hasilnya ke konsol.
- `System.out.println((float)g);`: Melakukan *casting* dari `char` ke `float` (berdasarkan nilai ASCII) dan mencetak hasilnya ke konsol.
- `System.out.println((double)g);`: Melakukan *casting* dari `char` ke `double` (berdasarkan nilai ASCII) dan mencetak hasilnya ke konsol.
- `System.out.println((int)k);`: Melakukan *casting* dari `double` ke `int` dan mencetak hasilnya ke konsol.
- `System.out.println((float)k);`: Melakukan *casting* dari `double` ke `float` dan mencetak hasilnya ke konsol.

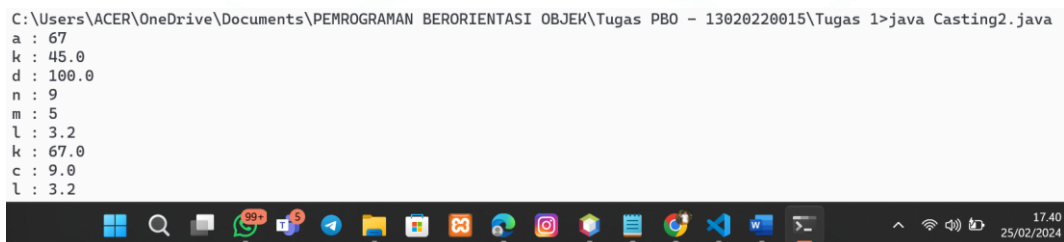
Output:

- Program ini akan mencetak hasil dari operasi \*casting\* ke konsol.

Program ini dirancang untuk memberikan contoh bagaimana \*casting\* tipe data dapat dilakukan di Java, dengan mengubah nilai dari satu tipe data ke tipe data lainnya.

- **Program 7**

-OutPut Program class Casting2 Dan Penjelasan Programnya



```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Casting2.java
a : 67
k : 45.0
d : 100.0
n : 9
m : 5
l : 3.2
k : 67.0
c : 9.0
l : 3.2
```

Penjelasan Program:

Program ini bertujuan untuk menunjukkan contoh penggunaan \*casting\* atau konversi antara tipe data di Java, khususnya konversi antara tipe data primitif dan tipe data String.

Keyword yang Digunakan / Tipe Data:

- `public class Casting2`: Mendeklarasikan kelas dengan nama "Casting2". `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini bersifat statis, dan `void` menunj- `int a=8, b=9;`
- `float d=2.f, e=3.2f;`

- `char g='5';`
- `double k=3.14;`
- `String n="67", m="45", l="100";`

Algoritma:

- `a = Integer.parseInt(n);`: Mengubah nilai `n` (String) menjadi nilai `a` (int) menggunakan metode `parseInt` dari kelas `Integer`.
- `k = Double.parseDouble(m);`: Mengubah nilai `m` (String) menjadi nilai `k` (double) menggunakan metode `parseDouble` dari kelas `Double`.
- `d = Float.parseFloat(l);`: Mengubah nilai `l` (String) menjadi nilai `d` (float) menggunakan metode `parseFloat` dari kelas `Float`.
- `n = String.valueOf(b);`: Mengubah nilai `b` (int) menjadi nilai `n` (String) menggunakan metode `valueOf` dari kelas `String`.
- `m = String.valueOf(g);`: Mengubah nilai `g` (char) menjadi nilai `m` (String) menggunakan metode `valueOf` dari kelas `String`.
- `l = String.valueOf(e);`: Mengubah nilai `e` (float) menjadi nilai `l` (String) menggunakan metode `valueOf` dari kelas `String`.
- `k = Double.valueOf(a).intValue();`: Mengubah nilai `a` (int) menjadi `Double` dan kemudian mengambil nilainya sebagai int.
- `double c = Integer.valueOf(b).doubleValue();`: Mengubah nilai `b` (int) menjadi `Integer` dan kemudian mengambil nilainya sebagai double.

Output:

- Program ini akan mencetak nilai variabel-variabel setelah proses konversi ke konsol.

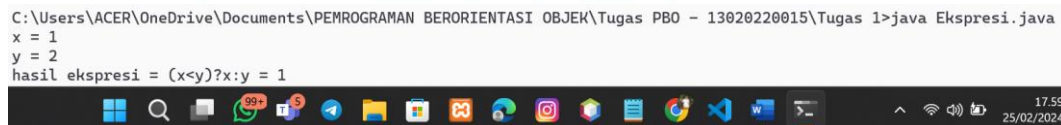
Catatan:

- Metode `parseInt`, `parseDouble`, `parseFloat`, `valueOf`, `intValue`, dan `doubleValue` adalah bagian dari kelas-kelas bawaan Java yang menyediakan fungsi konversi tipe data.

Program ini memberikan contoh penggunaan berbagai metode untuk melakukan konversi antara tipe data primitif dan String di Java.

- **Program 8**

-OutPut Program class Ekspresi Dan Penjelasan Programnya



```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Ekspresi.java
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
```

Penjelasan Program:

Program ini bertujuan untuk memberikan contoh penggunaan ekspresi ternary (conditional) atau operator kondisional dalam Java. Ekspresi ternary digunakan untuk menggantikan struktur pemilihan (if-else) dalam kasus sederhana.

Keyword yang Digunakan / Tipe Data:

- `public class Ekspresi`: Mendeklarasikan kelas dengan nama "Ekspresi". `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini bersifat statis, dan `void` menunjukkan bahwa metode ini tidak mengembalikan nilai.
- Deklarasi dan inisialisasi variabel:
  - `int x = 1;`

- `int y = 2;`

Algoritma:

- `System.out.print("x = " + x + "\n");`: Mencetak nilai variabel `x` ke konsol.
- `System.out.print("y = " + y + "\n");`: Mencetak nilai variabel `y` ke konsol.
- `System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y));`: Mencetak hasil ekspresi ternary `(x < y) ? x : y` ke konsol. Ekspresi ini menghasilkan nilai `x` jika `x` kurang dari `y`, dan menghasilkan nilai `y` jika tidak.

Output:

- Program ini akan mencetak nilai variabel `x` dan `y`, serta hasil dari ekspresi ternary ke konsol.

Catatan:

- Operator ternary memiliki bentuk umum: `(kondisi) ? nilai_jika_true : nilai_jika_false`.
- Jika kondisi benar, nilai yang dikembalikan adalah nilai jika benar; jika tidak, nilai yang dikembalikan adalah nilai jika salah.

Program ini memberikan contoh sederhana penggunaan ekspresi ternary untuk memilih nilai berdasarkan kondisi tertentu.

## • Program 9

-OutPut Program class Ekspresi1 Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Ekspresi1.java
x/y (format integer) = 0
x/y (format float) = 0
x/y (format integer) = 0.5
x/y (format float) = 0.5
float(x)/float(y) (format integer)= 0.5
float(x)/float(y) (format float) = 0.5
x/y (format integer) = 3
x/y (format float) = 3
```

### Penjelasan Program:

Program ini bertujuan untuk memberikan contoh konversi tipe data dalam operasi pembagian dengan menggunakan operator `/`. Program ini menunjukkan bagaimana pembagian dengan tipe data integer dapat menghasilkan hasil yang berbeda dengan pembagian menggunakan tipe data float.

### Keyword yang Digunakan / Tipe Data:

- `public class Ekspresi1`: Mendeklarasikan kelas dengan nama "Ekspresi1". `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini bersifat statis, dan `void` menunjukkan bahwa metode ini tidak mengembalikan nilai.
- Deklarasi dan inisialisasi variabel:
  - `int x = 1`;
  - `int y = 2`;
  - `float fx`;
  - `float fy`;

### Algoritma:



- `System.out.print("x/y (format integer) = "+ x/y);`: Mencetak hasil pembagian `x` dan `y` dalam format integer ke konsol.
- `System.out.print("\nx/y (format float) = "+ x/y);`: Mencetak hasil pembagian `x` dan `y` dalam format float ke konsol.
- `fx=x; fy=y;`: Mengkonversi nilai `x` dan `y` ke tipe data float.
- `System.out.print("\nx/y (format integer) = "+ fx/fy);`: Mencetak hasil pembagian `fx` dan `fy` dalam format integer ke konsol.
- `System.out.print("\nx/y (format float) = "+ fx/fy);`: Mencetak hasil pembagian `fx` dan `fy` dalam format float ke konsol.
- `System.out.print("\nfloat(x)/float(y) (format integer)= "+ (float)x/(float)y);`: Mencetak hasil pembagian `(float)x/(float)y` dalam format integer ke konsol.
- `System.out.print("\nfloat(x)/float(y) (format float) = "+ (float)x/(float)y);`: Mencetak hasil pembagian `(float)x/(float)y` dalam format float ke konsol.
- `x = 10; y = 3;`: Mengganti nilai `x` dan `y`.
- `System.out.print("\nx/y (format integer) = "+ x/y);`: Mencetak hasil pembagian `x` dan `y` setelah nilai `x` dan `y` diganti ke konsol.
- `System.out.print("\nx/y (format float) = "+ x/y);`: Mencetak hasil pembagian `x` dan `y` dalam format float ke konsol.

Output:

- Program ini akan mencetak hasil-hasil pembagian ke konsol dengan berbagai format.

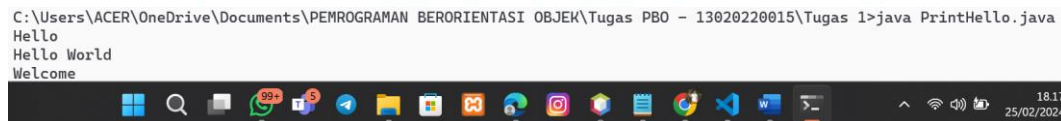
Catatan:

- Operator pembagian `/` dalam Java akan menghasilkan nilai integer jika kedua operandnya berupa integer. Jika setidaknya salah satu operand adalah float, hasilnya akan float.

Program ini memberikan contoh bagaimana tipe data dalam operasi pembagian dapat mempengaruhi hasil, terutama ketika menggunakan operator `/` dengan tipe data integer.

- **Program 10**

-OutPut Program class PrintHello Dan Penjelasan Programnya



```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java PrintHello.java
Hello
Hello World
Welcome
```

Penjelasan Program:

Program ini bertujuan untuk mencetak beberapa pesan ke konsol menggunakan berbagai metode `print` dan `println`. Program ini memberikan contoh penggunaan perintah cetak dalam Java.

Keyword yang Digunakan / Tipe Data:

- `public class PrintHello`: Mendeklarasikan kelas dengan nama "PrintHello". `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.

- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini bersifat statis, dan `void` menunjukkan bahwa metode ini tidak mengembalikan nilai.

Algoritma:

- `System.out.print("Hello");``: Mencetak pesan "Hello" ke konsol tanpa pindah ke baris baru.
- `System.out.print("\nHello ");``: Mencetak pesan "Hello" ke konsol dan pindah ke baris baru.
- `System.out.println("World");``: Mencetak pesan "World" ke konsol dan pindah ke baris baru.
- `System.out.println("Welcome");``: Mencetak pesan "Welcome" ke konsol dan pindah ke baris baru.

#### Output:

- Program ini akan mencetak pesan-pesan ke konsol, dan karena digunakan metode `println``, setiap pesan akan tercetak di baris baru.

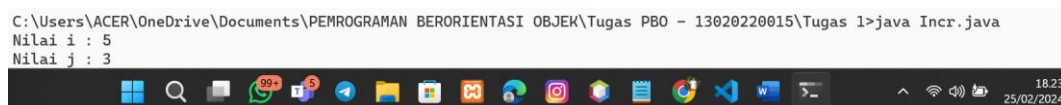
#### Catatan:

- Metode `print`` digunakan untuk mencetak teks tanpa pindah ke baris baru.
- Metode `println`` digunakan untuk mencetak teks dan pindah ke baris baru setelah mencetak.

Program ini memberikan contoh penggunaan perintah cetak dalam Java untuk mencetak pesan ke konsol.

### • Program 11

#### -OutPut Program class Incr Dan Penjelasan Programnya



```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Incr.java
Nilai i : 5
Nilai j : 3
```

#### Penjelasan Program:

Program ini bertujuan untuk memberikan contoh penggunaan operator increment (`++`) dalam Java. Program ini menunjukkan perbedaan antara pre-increment (`++i`) dan post-increment (`i++`).

Keyword yang Digunakan / Tipe Data:

- `public class Incr`: Mendeklarasikan kelas dengan nama "Incr". `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini bersifat statis, dan `void` menunjukkan bahwa metode ini tidak mengembalikan nilai.
- Deklarasi dan inisialisasi variabel:
  - `int i, j`

Algoritma:

- `i = 3`: Menginisialisasi variabel `i` dengan nilai 3.
- `j = i++`: Menggunakan post-increment, yaitu nilai `i` diberikan kepada `j` sebelum `i` ditingkatkan. Sehingga, nilai `j` adalah 3 dan nilai `i` menjadi 4.
- `System.out.println ("Nilai i : " + (++i) + "\nNilai j : " + j)`: Mencetak nilai `i` setelah pre-increment dan nilai `j` setelah post-increment ke konsol.

Output:

- Program ini akan mencetak nilai `i` setelah pre-increment dan nilai `j` setelah post-increment ke konsol.

Catatan:

- Pre-increment (`++i`) akan meningkatkan nilai variabel sebelum nilai variabel tersebut digunakan.
- Post-increment (`i++`) akan menggunakan nilai variabel sebelum nilai variabel tersebut ditingkatkan.

Program ini memberikan contoh perbedaan antara pre-increment dan post-increment dalam Java.

## • Program 12

### -OutPut Program class Oper1 Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Oper1.java
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

### Penjelasan Program:

Program ini bertujuan untuk memberikan contoh penggunaan beberapa operator bitwise (operasi bit) dalam Java. Program ini menunjukkan penggunaan operator AND (`&`), NOT (`~`), dan shift (`<<`, `>>`).

### Keyword yang Digunakan / Tipe Data:

- `public class Oper1`: Mendeklarasikan kelas dengan nama "Oper1".  
`public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini bersifat statis, dan `void` menunjukkan bahwa metode ini tidak mengembalikan nilai.
- Deklarasi dan inisialisasi variabel:

```
- `int n = 10; /* 1010 */  
- `int x = 1; /* 1 */  
- `int y = 2; /* 10 */
```

Algoritma:

- ``System.out.println ("n = " + n);``: Mencetak nilai variabel ``n`` ke konsol.
- ``System.out.println ("x = " + x);``: Mencetak nilai variabel ``x`` ke konsol.
- ``System.out.println ("y = " + y);``: Mencetak nilai variabel ``y`` ke konsol.
- ``System.out.println("n & 8 = " + (n & 8));``: Melakukan operasi bitwise AND (``&``) antara nilai ``n`` dan ``8`` (binary 1010 AND 1000) dan mencetak hasilnya ke konsol.
- ``System.out.println ("x & ~ 8 = " + (x & ~8));``: Melakukan operasi bitwise AND (``&``) antara nilai ``x`` dan komplement (``~``) dari ``8`` (binary 1 AND 0111) dan mencetak hasilnya ke konsol.
- ``System.out.println ("y << 2 = " + (y << 2));``: Melakukan operasi bitwise shift left (``<<``) pada nilai ``y`` sebanyak 2 bit (binary 10 ==> 1000) dan mencetak hasilnya ke konsol.
- ``System.out.println ("y >> 3 = " + (y >>3));``: Melakukan operasi bitwise shift right (``>>``) pada nilai ``y`` sebanyak 3 bit (binary 10 ==> 0000) dan mencetak hasilnya ke konsol.

Output:

- Program ini akan mencetak nilai variabel ``n``, ``x``, dan ``y``, serta hasil dari operasi bitwise ke konsol.

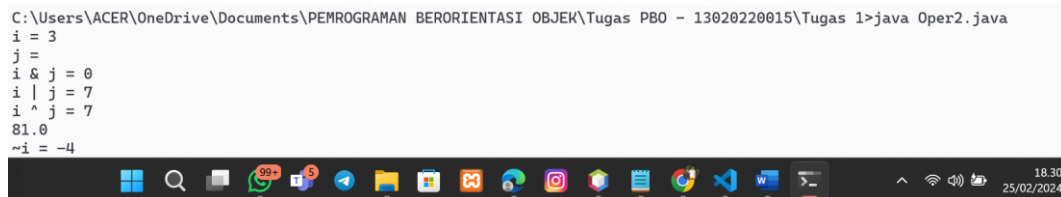
Catatan:

- Operasi bitwise AND (`&`): Setiap bit pada posisi yang sama di kedua operand harus 1 untuk menghasilkan 1.
- Operasi bitwise NOT (`~`): Mengganti setiap bit 0 menjadi 1, dan setiap bit 1 menjadi 0.
- Operasi bitwise shift left (`<<`): Menggeser bit ke kiri sebanyak n posisi.
- Operasi bitwise shift right (`>>`): Menggeser bit ke kanan sebanyak n posisi.

Program ini memberikan contoh penggunaan beberapa operator bitwise untuk manipulasi bit dalam nilai variabel.

### • Program 13

#### -OutPut Program class Oper2 Dan Penjelasan Programnya



```

C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Oper2.java
i = 3
j = 0
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4

```

#### Penjelasan Program:

Program ini bertujuan untuk memberikan contoh penggunaan beberapa operator bitwise (operasi bit) dan operasi matematika sederhana pada karakter (`char`) dalam Java. Program ini menunjukkan penggunaan operator AND (`&`), OR (`|`), XOR (`^`), NOT (`~`), dan fungsi `Math.pow()`.

#### Keyword yang Digunakan / Tipe Data:

- `public class Oper2`: Mendeklarasikan kelas dengan nama "Oper2".  
`public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa

metode ini dapat diakses dari luar kelas, ``static`` menunjukkan bahwa metode ini bersifat statis, dan ``void`` menunjukkan bahwa metode ini tidak mengembalikan nilai.

- Deklarasi dan inisialisasi variabel:

- ``char i, j;``

Algoritma:

- ``i = 3;``: Menginisialisasi variabel ``i`` dengan nilai 3 (dalam representasi karakter).
- ``j = 4;``: Menginisialisasi variabel ``j`` dengan nilai 4 (dalam representasi karakter).
- ``System.out.println("i = " + (int) i);``: Mencetak nilai variabel ``i`` dalam bentuk integer ke konsol.
- ``System.out.println("j = " + j);``: Mencetak nilai variabel ``j`` ke konsol.
- ``System.out.println("i & j = " + (i & j));``: Melakukan operasi bitwise AND (``&``) antara nilai ``i`` dan ``j`` dan mencetak hasilnya ke konsol.
- ``System.out.println("i | j = " + (i | j));``: Melakukan operasi bitwise OR (``|``) antara nilai ``i`` dan ``j`` dan mencetak hasilnya ke konsol.
- ``System.out.println("i ^ j = " + (i ^ j));``: Melakukan operasi bitwise XOR (``^``) antara nilai ``i`` dan ``j`` dan mencetak hasilnya ke konsol.
- ``System.out.println(Math.pow(i, j));``: Menghitung nilai ``i`` pangkat ``j`` menggunakan fungsi ``Math.pow()`` dan mencetak hasilnya ke konsol.
- ``System.out.println("~i = " + ~i);``: Melakukan operasi bitwise NOT (``~``) pada nilai ``i`` dan mencetak hasilnya ke konsol.

Output:

- Program ini akan mencetak nilai variabel ``i`` dan ``j``, serta hasil dari operasi bitwise dan fungsi pangkat ke konsol.



Catatan:

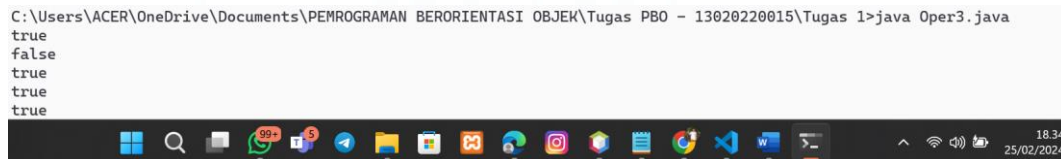
- Operator bitwise (`&`, `|`, `^`, `~`) bekerja pada level bit.
- Fungsi `Math.pow(a, b)` mengembalikan `a` pangkat `b`.

Program ini memberikan contoh penggunaan operator bitwise dan fungsi matematika sederhana pada karakter dalam Java.

### • Program 14

-OutPut Program class Oper3 Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Oper3.java
true
false
true
true
true
```



Penjelasan Program:

Program ini bertujuan untuk memberikan contoh penggunaan operator logika (`&&`, `&`, `||`, `|`) dalam Java. Program ini menunjukkan perbedaan antara operator logika AND (`&&`), OR (`||`) dan operator bitwise AND (`&`), OR (`|`).

Keyword yang Digunakan / Tipe Data:

- `public class Oper3`: Mendeklarasikan kelas dengan nama "Oper3".  
`public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini bersifat statis, dan `void` menunjukkan bahwa metode ini tidak mengembalikan nilai.

Algoritma:

- `if (true && true) { System.out.println(true && true); }`: Menggunakan operator logika AND (`&&`) dalam kondisi if, yang akan mengevaluasi ke `true` jika kedua operannya `true`. Ini mencetak `true` ke konsol.
- `if (true & true) { System.out.println(true & false); }`: Menggunakan operator bitwise AND (`&`) dalam kondisi if, yang akan mengevaluasi ke `true` jika kedua operannya `true`. Ini mencetak `true` ke konsol.
- `if (true) { System.out.println(true); }`: Kondisi if sederhana yang selalu bernilai `true`, mencetak `true` ke konsol.
- `if (true || true) { System.out.println(true); }`: Menggunakan operator logika OR (`||`) dalam kondisi if, yang akan mengevaluasi ke `true` jika setidaknya satu operannya `true`. Ini mencetak `true` ke konsol.
- `if (true | false) { System.out.println(true | false); }`: Menggunakan operator bitwise OR (`|`) dalam kondisi if, yang akan mengevaluasi ke `true` jika setidaknya satu operannya `true`. Ini mencetak `true` ke konsol.

#### Output:

- Program ini akan mencetak hasil dari evaluasi kondisi-kondisi if ke konsol.

#### Catatan:


- Operator logika (`&&`, `||`) menggunakan short-circuit evaluation, yang berarti evaluasi dihentikan jika hasil sudah jelas dari operand pertama.
- Operator bitwise (`&`, `|`) selalu mengevaluasi kedua operannya, tanpa memperdulikan hasil operand pertama.

Program ini memberikan contoh perbedaan antara operator logika dan operator bitwise dalam kondisi if.

- **Program 15**

**-OutPut Program class Oper4 Dan Penjelasan Programnya**

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Oper4.java
Nilai e = 10
Nilai k = 0
Nilai k = 4
```



**Penjelasan Program:**

Program ini bertujuan untuk memberikan contoh penggunaan operator ternary (conditional) dalam Java. Program ini menunjukkan cara menggunakan operator ternary untuk mengevaluasi ekspresi bersyarat.

**Keyword yang Digunakan / Tipe Data:**

- `public class Oper4`: Mendeklarasikan kelas dengan nama "Oper4". `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini bersifat statis, dan `void` menunjukkan bahwa metode ini tidak mengembalikan nilai.

**Algoritma:**

- `int i = 0`: Mendeklarasikan dan menginisialisasi variabel `i` dengan nilai 0.
- `int j = 0`: Mendeklarasikan dan menginisialisasi variabel `j` dengan nilai 0.
- `char c = 8; char d = 10`: Mendeklarasikan dan menginisialisasi dua variabel karakter `c` dan `d` dengan nilai 8 dan 10.
- `int e = (((int)c > (int)d) ? c : d)`: Menggunakan operator ternary untuk mengevaluasi apakah `(int)c` lebih besar dari `(int)d`. Jika ya, maka `e` diisi dengan nilai `c`, jika tidak, diisi dengan nilai `d`.

- `int k = ((i > j) ? i : j);`: Menggunakan operator ternary untuk mengevaluasi apakah `i` lebih besar dari `j`. Jika ya, maka `k` diisi dengan nilai `i`, jika tidak, diisi dengan nilai `j`.
- `System.out.print ("Nilai e = " + e);`: Mencetak nilai `e` ke konsol.
- `System.out.print ("\nNilai k = " + k);`: Mencetak nilai `k` ke konsol.
- `i = 2;`: Mengganti nilai `i` menjadi 2.
- `j = 3;`: Mengganti nilai `j` menjadi 3.
- `k = ((i++ > j++) ? i : j);`: Menggunakan operator ternary untuk mengevaluasi apakah `i++` lebih besar dari `j++`. Jika ya, maka `k` diisi dengan nilai `i` setelah diinkrement, jika tidak, diisi dengan nilai `j` setelah diinkrement.
- `System.out.print ("\nNilai k = " + k);`: Mencetak nilai `k` setelah perubahan ke konsol.

Output:

- Program ini akan mencetak nilai `e` dan `k` ke konsol.

Catatan:

- Operator ternary (`? :`) digunakan untuk mengevaluasi ekspresi bersyarat. Syntaxnya adalah `(kondisi) ? nilaiJikaTrue : nilaiJikaFalse`.

Program ini memberikan contoh penggunaan operator ternary dalam Java untuk mengevaluasi ekspresi bersyarat.

- **Program 16**

-OutPut Program class Oprator Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 1>java Oprator.java
Silahkan baca teksnya dan tambahkan perintah program di bawah ini untuk menampilkan output program
false
true
false
true
0
10.0
0.0
1.0
25.0
false
true
false
true
false
true
false
false
false
true
true
```

### Penjelasan Program:

Program ini memiliki tujuan untuk memberikan contoh penggunaan berbagai operator dan operasi dalam bahasa pemrograman Java. Program ini mencakup operator logika, perbandingan, aritmetika, dan lainnya.

### Keyword yang Digunakan / Tipe Data:

- `public class Oprator`: Mendeklarasikan kelas dengan nama "Oprator".  
`public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- `public static void main(String[] args)`: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini bersifat statis, dan `void` menunjukkan bahwa metode ini tidak mengembalikan nilai.
- Deklarasi dan inisialisasi variabel:
  - `boolean Bool1, Bool2, TF` ;
  - `int i = 10, j = 5, hsl` ;
  - `float x, y, res` ;

### Algoritma:

- Operasi Logika:

- ``Bool1 = true; Bool2 = false;``
- ``TF = Bool1 && Bool2 ; System.out.println(TF);``: Melakukan operasi AND logika (``&&``) dan mencetak hasilnya.
- ``TF = Bool1 || Bool2 ; System.out.println(TF);``: Melakukan operasi OR logika (``||``) dan mencetak hasilnya.
- ``TF = ! Bool1 ; System.out.println(TF);``: Melakukan operasi NOT logika (``!``) dan mencetak hasilnya.
- ``TF = Bool1 ^ Bool2; System.out.println(TF);``: Melakukan operasi XOR logika (``^``) dan mencetak hasilnya.

- Operasi Aritmetika:

- ``hsl = i % j ; System.out.println(hsl);``: Melakukan operasi modulus (``%``) dan mencetak hasilnya.
- ``x = 5 ; y = 5 ; res = x + y; System.out.println(res);``: Melakukan operasi penambahan dan mencetak hasilnya.
- ``res = x - y; System.out.println(res);``: Melakukan operasi pengurangan dan mencetak hasilnya.
- ``res = x / y; System.out.println(res);``: Melakukan operasi pembagian dan mencetak hasilnya.
- ``res = x * y; System.out.println(res);``: Melakukan operasi perkalian dan mencetak hasilnya.

- Operasi Perbandingan:

- ``TF = (i == j); System.out.println(TF);``: Melakukan operasi perbandingan apakah ``i`` sama dengan ``j`` dan mencetak hasilnya.
- ``TF = (i != j); System.out.println(TF);``: Melakukan operasi perbandingan apakah ``i`` tidak sama dengan ``j`` dan mencetak hasilnya.

- `TF = (i < j); System.out.println(TF);`: Melakukan operasi perbandingan apakah `i` kurang dari `j` dan mencetak hasilnya.

- `TF = (i > j); System.out.println(TF);`: Melakukan operasi perbandingan apakah `i` lebih dari `j` dan mencetak hasilnya.

- `TF = (i <= j); System.out.println(TF);`: Melakukan operasi perbandingan apakah `i` kurang dari atau sama dengan `j` dan mencetak hasilnya.

- `TF = (i >= j); System.out.println(TF);`: Melakukan operasi perbandingan apakah `i` lebih dari atau sama dengan `j` dan mencetak hasilnya.

- `TF = (x != y); System.out.println(TF);`: Melakukan operasi perbandingan apakah `x` tidak sama dengan `y` dan mencetak hasilnya.

- `TF = (x < y); System.out.println(TF);`: Melakukan operasi perbandingan apakah `x` kurang dari `y` dan mencetak hasilnya.

- `TF = (x > y); System.out.println(TF);`: Melakukan operasi perbandingan apakah `x` lebih dari `y` dan mencetak hasilnya.

- `TF = (x <= y); System.out.println(TF);`: Melakukan operasi perbandingan apakah `x` kurang dari atau sama dengan `y` dan mencetak hasilnya.

- `TF = (x >= y); System.out.println(TF);`: Melakukan operasi perbandingan apakah `x` lebih dari atau sama dengan `y` dan mencetak hasilnya.

Output:

- Program ini akan mencetak hasil dari berbagai operasi ke konsol.

Catatan:

- Program ini memberikan contoh penggunaan operator logika, aritmetika, dan perbandingan dalam bahasa Java.

- Operasi perbandingan mengembalikan nilai boolean (`true` atau `false`).

