

Tugas I

Makassar, 28 Februari 2025

**PRAKTIKUM
PEMROGRAMAN
BERORIENTASI OBJEK**



Nama : Aqilah Jamellya Wulandari Putrisyah

Nim : 13020230155

Frekuensi : TI_PBO-12(B2)

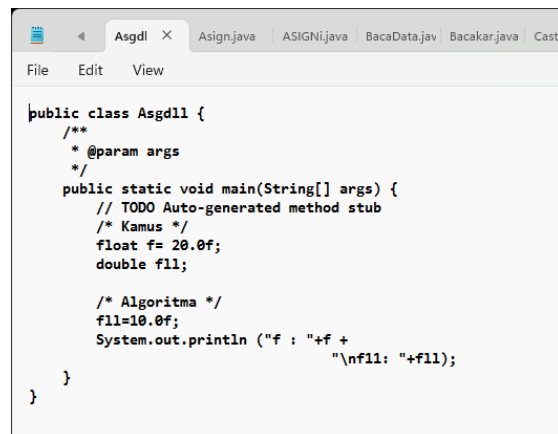
Dosen : Mardiyah Hasnawi, S.Kom., M.T.

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

2025

TUGAS 1 PBO

1.

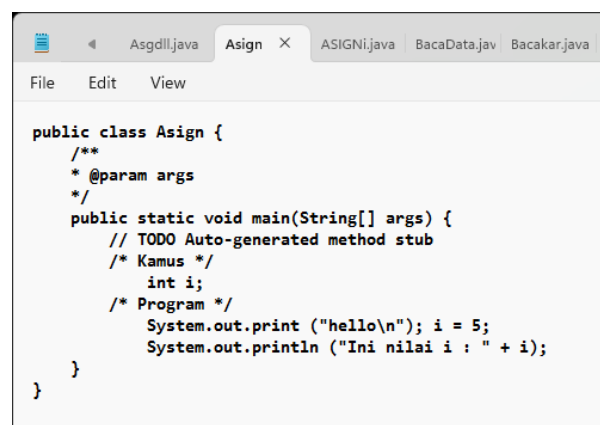


```
public class Asgdll {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        /* Kamus */  
        float f= 20.0f;  
        double fl1;  
  
        /* Algoritma */  
        fl1=10.0f;  
        System.out.println ("f : "+f +  
                               "\nf11: "+fl1);  
    }  
}
```

```
D:\KULIAH\SEMESTER 4\PBO>javac asgdll.java  
  
D:\KULIAH\SEMESTER 4\PBO>java asgdll  
Error: Could not find or load main class asgdll  
Caused by: java.lang.NoClassDefFoundError: asgdll (wrong name: Asgdll)  
  
D:\KULIAH\SEMESTER 4\PBO>java Asgdll  
f : 20.0  
f11: 10.0  
  
D:\KULIAH\SEMESTER 4\PBO>
```

Penjelasannya: Program ini dimulai dengan deklarasi kelas `Asgdll`, yang berisi metode `main` sebagai titik masuk utama. Di dalam metode `main`, terdapat deklarasi variabel `f` bertipe `float` yang diinisialisasi dengan nilai `20.0f`, serta variabel `fl1` bertipe `double` yang belum diberi nilai awal. Selanjutnya, dalam bagian algoritma, variabel `fl1` diberikan nilai `10.0`. Kemudian, program mencetak nilai variabel `f` dan `fl1` menggunakan `System.out.println()`. Output yang dihasilkan akan menampilkan nilai `f` sebagai `20.0` dan `fl1` sebagai `10.0` dalam format yang telah ditentukan. Program ini menunjukkan cara mendeklarasikan variabel, memberikan nilai, dan mencetaknya menggunakan Java.

2.



```
public class Assign {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        /* Kamus */  
        int i;  
        /* Program */  
        System.out.print ("hello\n"); i = 5;  
        System.out.println ("Ini nilai i : " + i);  
    }  
}
```

```
C:\WINDOWS\system32\CMD X + v
C:\belajar>javac Assign.java
C:\belajar>java Assign
hello
Ini nilai i : 5
C:\belajar>|
```

Penjelasannya: Program ini dimulai dengan deklarasi kelas `Assign` yang berisi metode `main` sebagai titik masuk utama. Di dalam metode ini, variabel `i` bertipe `int` dideklarasikan tanpa nilai awal. Program kemudian mencetak sebuah teks menggunakan `System.out.print()`, yang mencetak teks tanpa berpindah ke baris baru, tetapi dalam hal ini terdapat karakter newline (`\n`) yang menyebabkan perpindahan ke baris berikutnya. Setelah itu, variabel `i` diberikan nilai. Selanjutnya, `System.out.println()` digunakan untuk mencetak teks yang berisi nilai dari variabel `i`. Program ini menunjukkan bagaimana mendeklarasikan variabel, memberikan nilai, serta menampilkan output dengan metode `print` dan `println` dalam Java.

3.

```
Asgdll.java Assign.java ASIGNi X BacaData.jav Bacakar.java Cast
File Edit View
/* Program ini berisi contoh sederhana untuk mendefinisikan */
/* variabel-variabel bilangan bulat (short int, int, long int), */
/* karakter, bilangan riil, */
public class ASIGNi {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* KAMUS */
        short ks = 1;
        int ki = 1;
        long kl = 10000;
        char c = 65; /* inisialisasi karakter dengan integer */
        char c1 = 'Z'; /* inisialisasi karakter dengan karakter */
        double x = 50.2f;
        float y = 50.2f;
        /* Algoritma */

        /* penulisan karakter sebagai karakter */
        System.out.println ("Karakter = "+ c);
        System.out.println ("Karakter = "+ c1);

        /* penulisan karakter sebagai integer */
        System.out.println ("Karakter = "+ c);
        System.out.println ("Karakter = "+ c1);

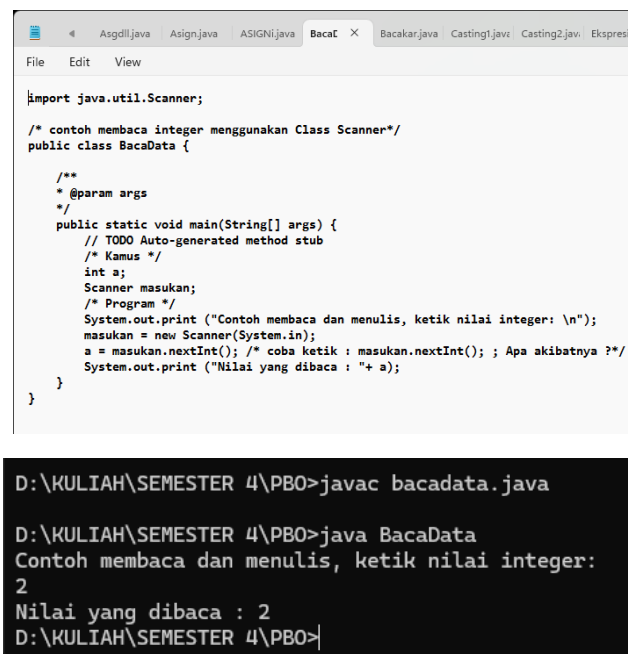
        System.out.println ("Bilangan integer (short) = "+ ks);
        System.out.println ("\t(int) = "+ ki);
        System.out.println ("\t(long)= "+ kl);
        System.out.println ("Bilangan Real x = "+ x);
        System.out.println ("Bilangan Real y = "+ y);
    }
}
Ln 1, Col 1 1,165 characters
```

```
D:\KULIAH\SEMESTER 4\PBO>javac ASIGNi.java
D:\KULIAH\SEMESTER 4\PBO>java ASIGNi
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
D:\KULIAH\SEMESTER 4\PBO>|
```

Penjelasannya: Program ini mendemonstrasikan deklarasi dan penggunaan berbagai jenis variabel, termasuk bilangan bulat, karakter, dan bilangan riil. Program dimulai dengan mendefinisikan kelas

`ASIGNi`, yang berisi metode `main` sebagai titik masuk eksekusi. Di dalam metode ini, beberapa variabel dideklarasikan dan diinisialisasi, termasuk `short`, `int`, dan `long` untuk bilangan bulat, `char` untuk karakter, serta `double` dan `float` untuk bilangan riil. Variabel karakter `c` diinisialisasi dengan nilai numerik, yang dikonversi ke karakter ASCII yang sesuai, sedangkan `c1` langsung diinisialisasi dengan karakter `Z`. Program kemudian mencetak nilai karakter baik dalam bentuk karakter maupun sebagai integer ASCII. Setelah itu, bilangan bulat dan bilangan riil yang telah dideklarasikan ditampilkan ke layar menggunakan `System.out.println()`, dengan beberapa nilai yang diformat menggunakan tabulasi (`\t`). Program ini menunjukkan bagaimana berbagai tipe data diinisialisasi, dikonversi, dan ditampilkan dalam Java, serta bagaimana karakter dapat direpresentasikan sebagai angka dalam kode ASCII.

4.



```
import java.util.Scanner;

/* contoh membaca integer menggunakan Class Scanner*/
public class BacaData {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus */
        int a;
        Scanner masukan;
        /* Program */
        System.out.print ("Contoh membaca dan menulis, ketik nilai integer: \n");
        masukan = new Scanner(System.in);
        a = masukan.nextInt(); /* coba ketik : masukan.nextInt(); ; Apa akibatnya ?*/
        System.out.print ("Nilai yang dibaca : "+ a);
    }
}
```

```
D:\KULIAH\SEMESTER 4\PBO>javac bacadata.java

D:\KULIAH\SEMESTER 4\PBO>java BacaData
Contoh membaca dan menulis, ketik nilai integer:
2
Nilai yang dibaca : 2
D:\KULIAH\SEMESTER 4\PBO>
```

Penjelasannya: Program ini menggunakan kelas `Scanner` untuk membaca input dari pengguna. Program dimulai dengan mengimpor `java.util.Scanner`, lalu mendeklarasikan kelas `BacaData` yang berisi metode `main` sebagai titik masuk utama. Di dalam metode `main`, variabel `a` bertipe `int` dideklarasikan tanpa nilai awal, dan objek `Scanner` bernama `masukan` juga dideklarasikan. Program kemudian mencetak teks instruksi kepada pengguna agar memasukkan nilai integer. Selanjutnya, objek `Scanner` dibuat menggunakan `new Scanner(System.in)`, memungkinkan program membaca input dari keyboard. Variabel `a` kemudian diisi dengan nilai integer yang dimasukkan pengguna melalui metode `nextInt()`. Terakhir, program mencetak kembali nilai yang telah dibaca menggunakan `System.out.print()`. Program ini menunjukkan cara menerima input angka dari pengguna dan mencetaknya kembali menggunakan `Scanner` dalam Java.

5.

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

//import javax.swing.*;

public class Bacakar {
    /**
     * @param args
     * @throws IOException
     */
    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        /* Kamus */
        char cc;
        int bil;
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader dataIn = new BufferedReader(isr);
        // atau
        BufferedReader dataIn = new BufferedReader(new InputStreamReader(System.in));
        /* Algoritma */
        System.out.print ("hello\n");

        System.out.print("baca 1 karakter : ");
        //perintah baca karakter cc
        cc =dataIn.readLine().charAt(0);
        System.out.print("baca 1 bilangan : ");
        //perintah baca bil
        bil =Integer.parseInt(dataIn.readLine());
        /*String kar = JOptionPane.showInputDialog("Karakter 1 : ");

        System.out.println(kar);*/
        //JOptionPane.showMessageDialog(null, "hello");
        System.out.print (cc +"\n" +bil+"\n");
        System.out.print ("bye \n");
    }
}

```

```

Karakter yang dibaca : 2
D:\KULIAH\SEMESTER 4\PBO>javac bacakar.java

D:\KULIAH\SEMESTER 4\PBO>java Bacakar
hello
baca 1 karakter : B
baca 1 bilangan : 2
B
2
bye
D:\KULIAH\SEMESTER 4\PBO>

```

Penjelasannya: Program 'Bacakar' membaca input dari pengguna menggunakan 'BufferedReader' dan menampilkan hasilnya ke layar. Program dimulai dengan mendeklarasikan variabel 'cc' untuk menyimpan karakter dan 'bil' untuk menyimpan bilangan bulat. Objek 'InputStreamReader' dan 'BufferedReader' dibuat untuk membaca input dari pengguna. Program mencetak "hello" sebagai salam pembuka, lalu meminta pengguna untuk memasukkan satu karakter yang dibaca menggunakan 'dataIn.readLine().charAt(0)'. Selanjutnya, program meminta pengguna memasukkan bilangan yang dibaca menggunakan 'Integer.parseInt(dataIn.readLine())', yang mengonversi string input menjadi integer. Setelah membaca input, program mencetak kembali karakter dan bilangan yang telah dimasukkan, lalu menampilkan pesan "bye" sebagai penutup. Program ini mendemonstrasikan cara membaca input dari pengguna dalam bentuk karakter dan bilangan menggunakan 'BufferedReader' di Java.

6.

```

/*Casting menggunakan tipe data primitif*/
public class Casting1 {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int a=5,b=6;
        float d=2.5f,e=3.2f;
        char g='5';
        double k=3.14;

        System.out.println((float)a); // int <-- float
        System.out.println((double)b); // int <-- double
        System.out.println((int)d); // float <-- int
        System.out.println((double)e); // float <-- double
        System.out.println((int)g); // char <-- int (ASCII)
        System.out.println((float)g); // char <-- float (ASCII)
        System.out.println((double)g); // char <-- double (ASCII)
        System.out.println((int)k); // double <-- int
        System.out.println((float)k); // double <-- float
    }
}

```

```
D:\KULIAH\SEMESTER 4\PBO>javac casting1.java

D:\KULIAH\SEMESTER 4\PBO>java Casting1
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14

D:\KULIAH\SEMESTER 4\PBO>
```

Penjelasannya: Program `Casting1` ini mendemonstrasikan proses konversi tipe data primitif di Java menggunakan casting eksplisit. Program dimulai dengan deklarasi dan inisialisasi variabel bertipe integer, float, char, dan double. Kemudian, setiap variabel dikonversi ke tipe data lain menggunakan tanda kurung `(tipe data)`. Integer dikonversi ke float dan double, float dikonversi ke integer dan double, sedangkan karakter dikonversi ke integer, float, dan double berdasarkan nilai ASCII-nya. Double juga dikonversi ke integer dan float. Program ini menunjukkan bagaimana casting eksplisit dapat mengubah representasi data dari satu tipe ke tipe lainnya di Java, terutama dalam menangani nilai numerik dan karakter ASCII.

7.

```
File Edit View

/*Casting menggunakan tipe data Class*/
public class Casting2 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        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";

        a = Integer.parseInt(n); /*Konversi String ke Integer*/
        k = Double.parseDouble(m); /*Konversi String ke Double*/
        d = Float.parseFloat(l); /*Konversi String ke Float*/
        System.out.println("a : "+a+"\nk : "+k+"\nd : "+d);

        n = String.valueOf(b); /*Konversi Integer ke String*/
        m = String.valueOf(g); /*Konversi Karakter ke String*/
        l = String.valueOf(e); /*Konversi Float ke String*/
        System.out.println("n : "+n+"\nm : "+m+"\nl : "+l);

        k = Double.valueOf(a).intValue(); /*Konversi Integer ke Double*/
        double c = Integer.valueOf(b).doubleValue();

        System.out.println("k : "+k+"\nc : "+c+"\nl : "+l);
    }
}
```

```
C:\WINDOWS\system32\CMD X + v

D:\KULIAH\SEMESTER 4\Tugas PBO>javac casting2.java

D:\KULIAH\SEMESTER 4\Tugas PBO>java Casting2
a : 67
k : 45.0
d : 100.0
n : 9
m : 5
l : 3.2
k : 67.0
c : 9.0
l : 3.2

D:\KULIAH\SEMESTER 4\Tugas PBO>
```

Penjelasannya: Program `Casting2` ini mendemonstrasikan penggunaan casting dan konversi tipe data dalam Java menggunakan kelas wrapper seperti `Integer`, `Double`, `Float`, dan `String`. Pertama, variabel bertipe integer, float, char, double, dan string diinisialisasi dengan nilai tertentu. Selanjutnya, dilakukan konversi tipe data:

- `a = Integer.parseInt(n);` mengonversi String `"67"` menjadi integer.

- `k = Double.parseDouble(m);` mengonversi String `"45"` menjadi double.
- `d = Float.parseFloat(l);` mengonversi String `"100"` menjadi float.

Setelah konversi ini, nilai baru dari `a`, `k`, dan `d` ditampilkan. Kemudian, dilakukan konversi dari tipe numerik atau karakter ke String:

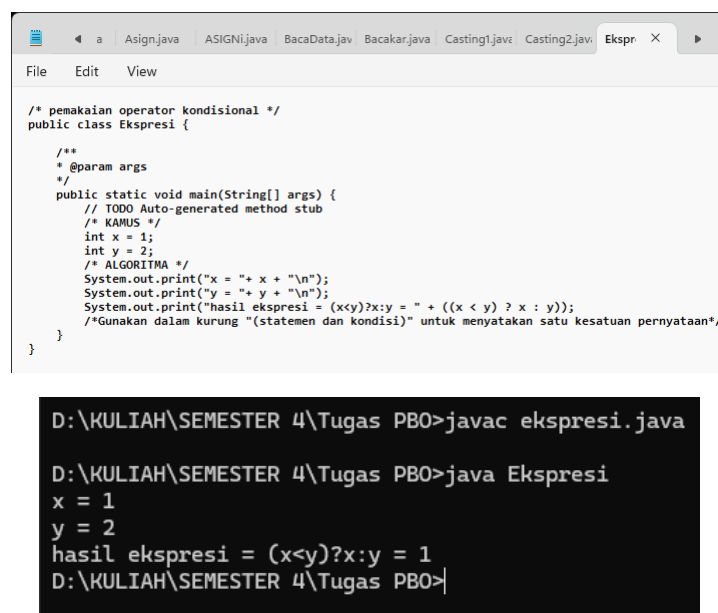
- `n = String.valueOf(b);` mengonversi integer `b` ke String.
- `m = String.valueOf(g);` mengonversi char `g` ke String.
- `l = String.valueOf(e);` mengonversi float `e` ke String.

Hasil konversi ini juga ditampilkan. Selanjutnya, dilakukan konversi antara tipe numerik yang berbeda:

- `k = Double.valueOf(a).intValue();` mengonversi integer `a` menjadi double, lalu dikonversi kembali ke integer.
- `double c = Integer.valueOf(b).doubleValue();` mengonversi integer `b` menjadi double.

Terakhir, nilai `k`, `c`, dan `l` ditampilkan sebagai hasil akhir. Program ini menunjukkan bagaimana Java menangani konversi tipe data, baik dari String ke tipe numerik, dari tipe numerik ke String, maupun antar tipe numerik menggunakan metode bawaan dari kelas wrapper.

8.



```

/* pemakaian operator kondisional */
public class Ekspresi {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* KAMUS */
        int x = 1;
        int y = 2;
        /* ALGORITMA */
        System.out.print("x = " + x + "\n");
        System.out.print("y = " + y + "\n");
        System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y));
        /*Gunakan dalam kurung "(statemen dan kondisi)" untuk menyatakan satu kesatuan pernyataan*/
    }
}

```

```

D:\KULIAH\SEMESTER 4\Tugas PBO>javac ekspresi.java

D:\KULIAH\SEMESTER 4\Tugas PBO>java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
D:\KULIAH\SEMESTER 4\Tugas PBO>

```

Penjelasannya: Program `Ekspres` ini mendemonstrasikan penggunaan operator kondisional (ternary operator) `? :` dalam Java untuk menentukan nilai berdasarkan suatu kondisi. Pertama, variabel `x` dan `y` bertipe `int` diinisialisasi dengan nilai `1` dan `2`. Program mencetak nilai `x` dan `y` ke layar menggunakan `System.out.print()`. Selanjutnya, ekspresi `(x < y) ? x : y` dijalankan. Operator `? :` bekerja dengan cara mengevaluasi kondisi `x < y`. Jika kondisi benar, maka nilai sebelum tanda `:` dipilih (`x`). Jika salah, nilai setelah tanda `:` dipilih (`y`). Karena `x = 1` dan `y = 2`, kondisi `x < y` bernilai `true`, sehingga hasil ekspresi adalah `x` (nilai `1`). Hasil ini kemudian ditampilkan ke layar. Program ini menunjukkan cara menggunakan operator ternary untuk membuat pernyataan if-else sederhana dalam satu baris kode.

9.

```
File Edit View

/* pembagian integer, casting */
public class Ekspresil {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* KAMUS */
        int x = 1; int y = 2; float fx; float fy;

        /* ALGORITMA */

        System.out.print ("x/y (format integer) = " + x/y);
        System.out.print ("\nx/y (format float) = " + x/y);
        /* supaya hasilnya tidak nol */
        fx=x;
        fy=y;

        System.out.print ("\nx/y (format integer) = " + fx/fy);
        System.out.print ("\nx/y (format float) = " + fx/fy);
        /* casting */

        System.out.print ("\nfloat(x)/float(y) (format integer) = " + (float)x/(float)y);
        System.out.print ("\nfloat(x)/float(y) (format float) = " + (float)x/(float)y);

        x = 10; y = 3;

        System.out.print ("\nx/y (format integer) = " + x/y);
        System.out.print ("\nx/y (format float) = " + x/y);
    }
}
```

```
D:\KULIAH\SEMESTER 4\Tugas PBO>javac ekspresil.java

D:\KULIAH\SEMESTER 4\Tugas PBO>java Ekspresil
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
D:\KULIAH\SEMESTER 4\Tugas PBO>
```

Penjelasannya: Program 'Ekspresil' ini mendemonstrasikan pembagian bilangan bulat ('int') dan bilangan desimal ('float') dalam Java serta penggunaan casting untuk mengonversi tipe data. Pertama, dua variabel 'x' dan 'y' bertipe 'int' diinisialisasi dengan nilai '1' dan '2'. Saat 'x/y' dihitung, karena kedua operand bertipe 'int', hasilnya juga 'int', sehingga hanya bagian bilangan bulat yang ditampilkan tanpa desimal. Kemudian, nilai 'x' dan 'y' diberikan ke variabel 'fx' dan 'fy' yang bertipe 'float', sehingga pembagian antara 'fx/fy' menghasilkan nilai desimal. Selanjutnya, casting eksplisit dilakukan dengan '(float)x/(float)y', yang mengonversi 'x' dan 'y' menjadi 'float' sebelum operasi pembagian, sehingga hasilnya tetap dalam format desimal. Setelah itu, 'x' dan 'y' diperbarui menjadi '10' dan '3', lalu operasi pembagian dilakukan lagi. Saat 'x/y' dihitung tanpa casting, hasilnya tetap 'int', sedangkan untuk pembagian dengan 'float', hasilnya memiliki nilai desimal. Program ini menunjukkan perbedaan antara pembagian integer yang hanya menghasilkan bilangan bulat, pembagian float yang memberikan hasil dengan desimal, serta penggunaan casting untuk mengubah tipe data agar mendapatkan hasil yang lebih akurat dalam operasi matematika.

10.

```
File Edit View

public class Hello {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* menuliskan hello ke layar */
        System.out.print("Hello");
        /* menuliskan hello dan ganti baris*/
        System.out.print("\nHello ");

        /* menuliskan hello dan ganti baris*/
        System.out.println("World");

        System.out.println("Welcome");
    }
}
```



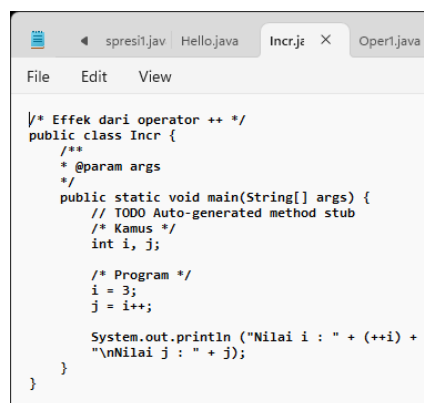
```
D:\KULIAH\SEMESTER 4\Tugas PBO>javac hello.java

D:\KULIAH\SEMESTER 4\Tugas PBO>java Hello
Hello
Hello World
Welcome

D:\KULIAH\SEMESTER 4\Tugas PBO>
```

Penjelasannya: Program 'Hello' ini merupakan program sederhana dalam Java yang menampilkan teks ke layar menggunakan 'System.out.print()' dan 'System.out.println()'. Program ini dimulai dengan mencetak kata "Hello" menggunakan 'System.out.print("Hello")', yang hanya menampilkan teks tanpa berpindah ke baris baru. Selanjutnya, 'System.out.print("\nHello ")' mencetak "Hello " setelah berpindah ke baris baru karena adanya karakter '\n' (newline). Kemudian, 'System.out.println("World")' mencetak "World" dan otomatis berpindah ke baris berikutnya karena 'println' selalu menambahkan newline di akhir outputnya. Terakhir, 'System.out.println("Welcome")' mencetak "Welcome" ke baris baru. Program ini menunjukkan perbedaan antara 'print()' yang tidak berpindah baris, 'println()' yang mencetak teks dan berpindah baris, serta penggunaan '\n' untuk berpindah baris secara manual.

11.



```
File Edit View

/* Efek dari operator ++ */
public class Incr {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus */
        int i, j;

        /* Program */
        i = 3;
        j = i++;

        System.out.println ("Nilai i : " + (++i) +
            "\nNilai j : " + j);
    }
}
```

```
D:\KULIAH\SEMESTER 4\Tugas PBO>javac incr.java

D:\KULIAH\SEMESTER 4\Tugas PBO>java Incr
Nilai i : 5
Nilai j : 3

D:\KULIAH\SEMESTER 4\Tugas PBO>
```

Penjelasannya: Program 'Incr' ini mendemonstrasikan efek dari operator increment ('++') dalam Java. Dimulai dengan mendeklarasikan dua variabel 'i' dan 'j'. Variabel 'i' diinisialisasi dengan nilai '3', lalu diberikan ke 'j' menggunakan post-increment ('i++'). Dalam ekspresi 'j = i++', nilai 'i' sebelum peningkatan ('3') diberikan terlebih dahulu ke 'j', lalu 'i' bertambah menjadi '4'. Pada perintah 'System.out.println()', ekspresi '++i' digunakan, yang merupakan pre-increment, sehingga 'i' bertambah terlebih dahulu menjadi '5' sebelum dicetak. Nilai 'j' tetap '3' karena dalam ekspresi sebelumnya, 'i' diberikan ke 'j' sebelum peningkatan terjadi. Program ini menunjukkan perbedaan antara pre-increment ('++i'), yang meningkatkan nilai sebelum digunakan, dan post-increment ('i++'), yang menggunakan nilai sebelum meningkat.

12.

```
File Edit View

/* pemakaian beberapa operator terhadap bit */
public class Oper1 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* KAMUS */

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

        /* ALGORITMA */
        System.out.println ("n = " + n);
        System.out.println ("x = " + x);
        System.out.println ("y = " + y);
        System.out.println ("n & 8 = " + (n & 8)); /* 1010 AND 1000 */
        System.out.println ("x & ~ 8 = " + (x & ~8)); /* 1 AND 0111 */
        System.out.println ("y << 2 = " + (y << 2)); /* 10 ==> 1000 = 8 */
        System.out.println ("y >> 3 = " + (y >> 3)); /* 10 ==> 0000 = 0 */
    }
}
```

```
D:\KULIAH\SEMESTER 4\Tugas PBO>javac oper1.java

D:\KULIAH\SEMESTER 4\Tugas PBO>java Oper1
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0

D:\KULIAH\SEMESTER 4\Tugas PBO>
```

Penjelasannya: Program 'Oper1' ini mendemonstrasikan penggunaan beberapa operator bitwise dalam Java. Dimulai dengan mendeklarasikan tiga variabel 'n', 'x', dan 'y' yang diinisialisasi dengan nilai '10', '1', dan '2', yang dalam bentuk biner adalah '1010', '0001', dan '0010'. Program mencetak nilai-nilai ini, lalu melakukan beberapa operasi bitwise. Operasi 'n & 8' (bitwise AND) dilakukan antara '1010' dan '1000', yang menghasilkan '1000' atau '8', karena hanya bit yang bernilai '1' di kedua angka yang dipertahankan. Operasi 'x & ~8' (bitwise AND dengan komplement bitwise) pertama-tama menghitung '~8', yaitu komplement '1000', yang menjadi '0111'. Hasil operasi '0001 & 0111' adalah '0001', atau '1'. Operasi 'y << 2' (bitwise left shift) menggeser bit 'y' ('0010') dua posisi ke kiri, menjadi '1000', yang bernilai '8' dalam desimal. Operasi 'y >> 3' (bitwise right shift) menggeser 'y' ('0010') tiga posisi ke kanan, menghasilkan '0000', yang bernilai '0' dalam desimal. Program ini memperlihatkan bagaimana operator bitwise bekerja dalam manipulasi biner, termasuk AND ('&'), NOT ('~'), left shift ('<<'), dan right shift ('>>').

13.

```
File Edit View

/* pemakaian beberapa operator terhadap RELATIONAL DAN bit */
public class Oper2 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* KAMUS */
        char i, j;

        /* ALGORITMA */
        i = 3; /* 00000011 dalam biner */
        j = 4; /* 00000100 dalam biner */

        System.out.println("i = " + (int) i);
        System.out.println("j = " + j);
        System.out.println("i & j = " + (i & j)); /* 0: 00000000 dalam biner */
        System.out.println("i | j = " + (i | j)); /* 7: 00000111 biner */
        System.out.println("i ^ j = " + (i ^ j)); /* 7: 00000111 biner Ingat!!! operator '^' pada bahasa java bukan sebagai pangkat */
        System.out.println(Math.pow(i, j)); /* Class Math memiliki method pow(a,b) untuk pemangkatan */
        System.out.println(" ~i = " + ~i); /* -4: 11111100 biner */
    }
}
```

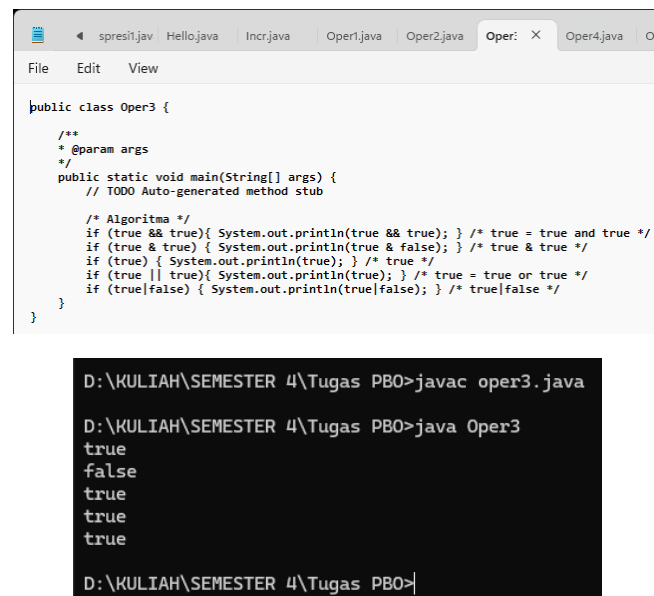
```
D:\KULIAH\SEMESTER 4\Tugas PBO>javac oper2.java

D:\KULIAH\SEMESTER 4\Tugas PBO>java Oper2
i = 3
j = 4
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4

D:\KULIAH\SEMESTER 4\Tugas PBO>
```

Penjelasannya: Program 'Oper2' ini mendemonstrasikan penggunaan operator relasional dan bitwise dalam Java. Dimulai dengan mendeklarasikan dua variabel karakter 'i' dan 'j', yang diinisialisasi dengan nilai 3 dan 4. Nilai ini direpresentasikan dalam bentuk biner sebagai '00000011' untuk 'i' dan '00000100' untuk 'j'. Program mencetak nilai numerik 'i' dan 'j' menggunakan 'System.out.println()', diikuti dengan operasi bitwise. Operasi 'i & j' (AND bitwise) menghasilkan '00000000' atau '0', karena tidak ada bit yang sama-sama '1' pada posisi yang sama. Operasi 'i | j' (OR bitwise) menghasilkan '00000111' atau '7', karena bit mana pun yang '1' akan tetap '1'. Operasi 'i ^ j' (XOR bitwise) juga menghasilkan '00000111' atau '7', karena hanya bit yang berbeda yang bernilai '1'. Program juga menunjukkan cara menghitung perpangkatan dengan 'Math.pow(i, j)', yang berarti '3^4' atau '81.0'. Terakhir, operasi '~i' (NOT bitwise) membalik semua bit 'i', menghasilkan '11111100' dalam representasi bilangan bertanda, yang setara dengan '-4' dalam sistem bilangan komplemen dua. Program ini menunjukkan bagaimana Java menangani operasi bitwise serta perbedaan antara '^' sebagai XOR dan 'Math.pow()' untuk perpangkatan.

14.



```
public class Oper3 {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        /* Algoritma */
        if (true && true) { System.out.println(true && true); } /* true = true and true */
        if (true & true) { System.out.println(true & false); } /* true & true */
        if (true) { System.out.println(true); } /* true */
        if (true || true) { System.out.println(true); } /* true = true or true */
        if (true | false) { System.out.println(true | false); } /* true | false */
    }
}

D:\KULIAH\SEMESTER 4\Tugas PBO>javac oper3.java

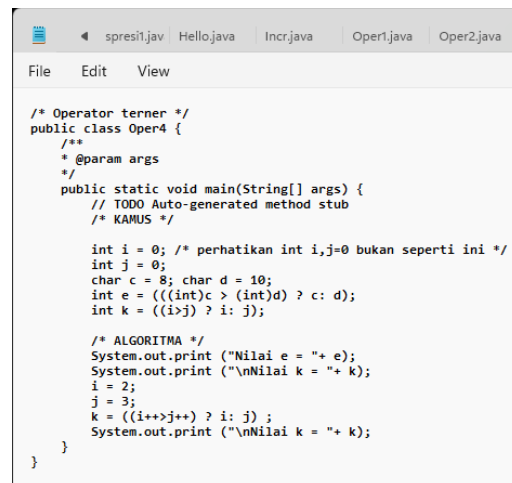
D:\KULIAH\SEMESTER 4\Tugas PBO>java Oper3
true
false
true
true
true

D:\KULIAH\SEMESTER 4\Tugas PBO>
```

Penjelasannya: Program 'Oper3' ini mendemonstrasikan penggunaan operator logika dalam Java, seperti AND ('&&' dan '&'), OR ('||' dan '|'), serta evaluasi kondisi dalam pernyataan 'if'. Program dimulai dengan mengevaluasi 'true && true', di mana kedua operand bernilai 'true', sehingga hasilnya 'true' dan dicetak ke layar. Selanjutnya, ekspresi 'true & true' juga menghasilkan 'true', tetapi ada kesalahan dalam pernyataan 'System.out.println(true & false);', yang seharusnya menggunakan 'true & true' agar sesuai dengan kondisi yang diperiksa. Evaluasi 'if(true)' akan selalu bernilai 'true', sehingga mencetak 'true' ke layar. Operator '||' digunakan dalam 'if(true || true)', yang hanya mengevaluasi operand pertama karena hasilnya sudah bisa ditentukan sebagai 'true', sehingga mencetak 'true'. Operator '|' dalam 'if(true | false)' mengevaluasi kedua operand tanpa short-circuit, tetapi karena salah satu operand adalah 'true', hasilnya tetap 'true' dan dicetak ke layar. Program ini

menunjukkan bagaimana operator logika bekerja dalam Java, terutama perbedaan antara short-circuit evaluation yang digunakan oleh `&&` dan `||` serta evaluasi penuh yang digunakan oleh `&` dan `|`.

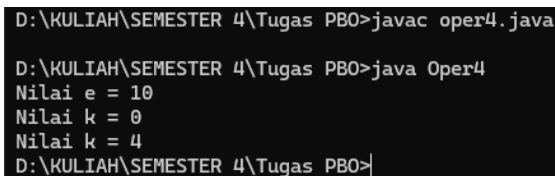
15.



```
/* Operator ternary */
public class Oper4 {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* KAMUS */

        int i = 0; /* perhatikan int i,j=0 bukan seperti ini */
        int j = 0;
        char c = 8; char d = 10;
        int e = (((int)c > (int)d) ? c: d);
        int k = ((i>j) ? i: j);

        /* ALGORITMA */
        System.out.print ("Nilai e = " + e);
        System.out.print ("\nNilai k = " + k);
        i = 2;
        j = 3;
        k = ((i++>j++) ? i: j) ;
        System.out.print ("\nNilai k = " + k);
    }
}
```



```
D:\KULIAH\SEMESTER 4\Tugas PBO>javac oper4.java
D:\KULIAH\SEMESTER 4\Tugas PBO>java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4
D:\KULIAH\SEMESTER 4\Tugas PBO>
```

Penjelasannya: Program `Oper4` ini bertujuan untuk mendemonstrasikan penggunaan operator ternary (`?:`) dalam menentukan nilai variabel berdasarkan kondisi tertentu. Program dimulai dengan deklarasi variabel integer `i = 0`, `j = 0`, serta dua variabel karakter `c = 8` dan `d = 10` (karakter direpresentasikan dalam bentuk nilai numerik ASCII). Kemudian, dilakukan operasi ternary untuk menentukan nilai `e`, di mana jika nilai ASCII dari `c` lebih besar dari `d`, maka `e` akan berisi `c`; jika tidak, `e` akan berisi `d`. Selanjutnya, variabel `k` ditentukan dengan operator ternary, yang memilih nilai lebih besar antara `i` dan `j`, dalam hal ini `k = j` karena `i = 0` dan `j = 0`, sehingga hasilnya tetap `0`. Program kemudian mencetak nilai `e` dan `k` ke layar. Selanjutnya, `i` diperbarui menjadi `2`, dan `j` menjadi `3`. Setelah itu, operator ternary digunakan lagi untuk menetapkan nilai `k`, tetapi kali ini menggunakan post-increment (`i++` dan `j++`), yang berarti perbandingan dilakukan dengan nilai awal `i = 2` dan `j = 3`, kemudian setelah evaluasi, nilai `i` dan `j` masing-masing bertambah menjadi `3` dan `4`. Karena kondisi `i++ > j++` (`2 > 3`) adalah false, maka `k` diisi dengan nilai `j` setelah increment, yaitu `4`. Akhirnya, program mencetak nilai `k` yang baru. Secara keseluruhan, program ini menunjukkan bagaimana operator ternary dapat digunakan untuk memilih nilai berdasarkan kondisi, serta bagaimana post-increment mempengaruhi hasil perhitungan.

16.

```
File Edit View

/* Contoh pengoperasian variabel bertipe dasar */
public class Oprator {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus */

        boolean Bool1, Bool2, TF; int i, j, hsl;
        float x, y, res;
        /* algoritma */
        System.out.println("Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output");

        Bool1 = true; Bool2 = false;
        TF = Bool1 && Bool2; /* Boolean AND */
        TF = Bool1 || Bool2; /* Boolean OR */
        TF = ! Bool1; /* NOT */
        TF = Bool1 ^ Bool2; /* XOR */
        /* operasi numerik */
        i = 5; j = 2;
        hsl = i+j; hsl = i - j; hsl = i / j; hsl = i * j;
        hsl = i / j; /* pembagian bulat */
        hsl = i%j; /* sisa, modulo */
        /* operasi numerik */
        x = 5; y = 5;
        res = x + y; res = x - y; res = x / y; res = x * y;
        /* operasi relasional numerik */
        TF = (i==j); TF = (i!=j);
        TF = (i < j); TF = (i > j); TF = (i <= j); TF = (i >= j);
        /* operasi relasional numerik */
        TF = (x != y);
        TF = (x < y); TF = (x > y); TF = (x <= y); TF = (x >= y);
    }
}
```

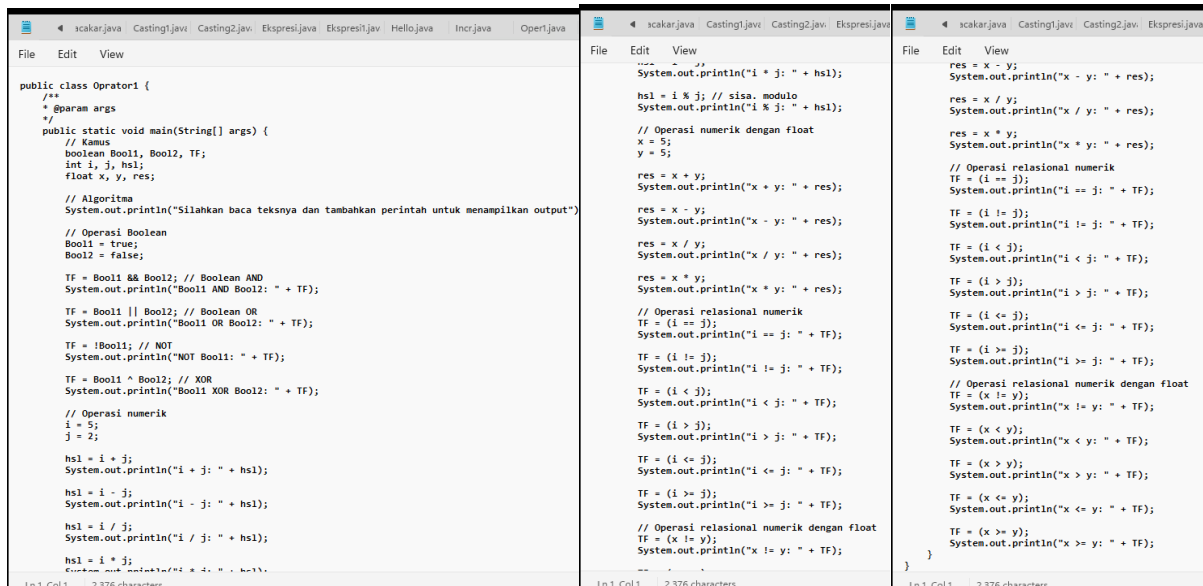
```
D:\KULIAH\SEMESTER 4\Tugas PBO>javac oprator.java

D:\KULIAH\SEMESTER 4\Tugas PBO>java Oprator
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output

D:\KULIAH\SEMESTER 4\Tugas PBO>
```

Penjelasannya: Program 'Oprator' ini bertujuan untuk mendemonstrasikan berbagai operasi dasar pada variabel dengan tipe data boolean, integer, dan float. Program dimulai dengan mendeklarasikan variabel boolean ('Bool1', 'Bool2', 'TF'), integer ('i', 'j', 'hsl'), serta float ('x', 'y', 'res'). Kemudian, program mencetak sebuah pesan sebagai pemberitahuan awal. Selanjutnya, program melakukan operasi logika menggunakan variabel boolean. 'Bool1' diatur 'true', sedangkan 'Bool2' diatur 'false'. Operasi yang dilakukan mencakup AND ('&&'), OR ('||'), NOT ('!'), dan XOR ('^'), dengan hasilnya disimpan dalam variabel 'TF', namun tidak ditampilkan ke layar karena tidak ada perintah 'System.out.println()'. Setelah itu, program menjalankan operasi aritmatika menggunakan variabel integer 'i = 5' dan 'j = 2'. Operasi yang dilakukan meliputi penjumlahan, pengurangan, pembagian, perkalian, serta modulus ('%') untuk mendapatkan sisa hasil bagi. Namun, hasil dari operasi-operasi ini hanya disimpan dalam 'hsl' tanpa ditampilkan. Operasi serupa dilakukan untuk tipe data float dengan variabel 'x = 5' dan 'y = 5', dan hasilnya disimpan dalam 'res', tetapi juga tidak ditampilkan. Program juga mengevaluasi operasi relasional untuk membandingkan dua nilai integer dan float menggunakan operator seperti sama dengan ('=='), tidak sama dengan ('!='), lebih kecil ('<'), lebih besar ('>'), lebih kecil atau sama dengan ('<='), dan lebih besar atau sama dengan ('>='). Hasil dari perbandingan ini disimpan dalam variabel 'TF', tetapi karena tidak ada perintah untuk mencetak output, pengguna tidak akan melihat hasil perbandingan di layar. Meskipun program ini mendemonstrasikan berbagai operasi logika, aritmatika, dan perbandingan dalam Java, kurangnya perintah 'System.out.println()' untuk mencetak hasil membuatnya tidak menampilkan output kepada pengguna.

***yang dapat menampilkan output:**



```

C:\WINDOWS\system32\CMD
D:\KULIAH\SEMESTER 4\Tugas PBO>javac oprator1.java

D:\KULIAH\SEMESTER 4\Tugas PBO>java Oprator1
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output
Bool1 AND Bool2: false
Bool1 OR Bool2: true
NOT Bool1: false
Bool1 XOR Bool2: true
i + j: 7
i - j: 3
i / j: 2
i * j: 10
i % j: 1
x + y: 10.0
x - y: 0.0
x / y: 1.0
x * y: 25.0
i == j: false
i != j: true
i < j: false
i > j: true
i <= j: false
i >= j: true
x != y: false
x < y: false
x > y: false
x <= y: true
x >= y: true

D:\KULIAH\SEMESTER 4\Tugas PBO>

```

Penjelasannya: Program 'Oprator1' ini digunakan untuk mendemonstrasikan berbagai operasi dasar dalam pemrograman, seperti operasi boolean, aritmatika, dan perbandingan. Program dimulai dengan mendeklarasikan variabel boolean ('Bool1', 'Bool2', 'TF'), integer ('i', 'j', 'hsl'), serta float ('x', 'y', 'res'). Kemudian, program menampilkan teks awal sebagai pemberitahuan. Selanjutnya, program melakukan operasi boolean dengan nilai 'Bool1' yang diatur 'true' dan 'Bool2' yang diatur 'false'. Operasi logika yang digunakan meliputi AND ('&&'), OR ('||'), NOT ('!'), dan XOR ('^'), lalu hasilnya ditampilkan ke layar. Setelah itu, program menjalankan operasi aritmatika menggunakan integer, yaitu penjumlahan, pengurangan, pembagian, perkalian, dan modulus, lalu hasilnya disimpan dalam variabel 'hsl' dan ditampilkan. Operasi yang sama juga dilakukan untuk tipe data float, dengan hasil yang disimpan dalam 'res' dan ditampilkan ke layar. Selain itu, program juga membandingkan dua angka menggunakan operator relasional seperti sama dengan ('=='), tidak sama dengan ('!='), lebih kecil ('<'), lebih besar ('>'), lebih kecil atau sama dengan ('<='), dan lebih besar atau sama dengan ('>='). Perbandingan ini dilakukan baik untuk variabel integer ('i' dan 'j') maupun float ('x' dan 'y'), dengan hasil yang disimpan dalam 'TF' dan ditampilkan ke layar. Program ini bertujuan untuk menunjukkan bagaimana operasi logika, aritmatika, dan perbandingan bekerja dalam Java dengan menampilkan hasil dari setiap operasi yang dilakukan.