

Tugas 1

Makassar, 28 Februari 2025

**PRAKTIKUM
PEMROGRAMAN
BERORIENTASI OBJEK**



Nama : Selfi
Stambuk : 13020230104
Frekuensi : TI_PBO-12 (B2)
Dosen : Mardiyah Hasnawi, S.Kom, M.T.

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2025**

1.

```
*Asgdll - Notepad
File Edit Format View Help
public class Asgdll {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus */
        float f= 20.0f;
        double fll;

        /* Algoritma */
        fll=10.0f;
        System.out.println ("f : "+f +
                                "\nf11: "+fll);
    }
}
```

```
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>java --version
java 23.0.2 2025-01-21
Java(TM) SE Runtime Environment (build 23.0.2+7-58)
Java HotSpot(TM) 64-Bit Server VM (build 23.0.2+7-58, mixed mode, sharing)

C:\Users\LENOVO>c:
C:\Users\LENOVO>c:
C:\Users\LENOVO>cd..
C:\Users>cd..
C:\>c Tugas 1
'C' is not recognized as an internal or external command,
operable program or batch file.
C:\>cd Tugas
C:\>cd Tugas 1
C:\Tugas 1>javac Asgdll.java
C:\Tugas 1>java Asgdll
f : 20.0
f11: 10.0
```

Penjelasan: Pada bagian algoritma, variabel fll kemudian diisi dengan nilai 10.0. Setelah itu, program menampilkan nilai kedua variabel menggunakan System.out.println(). Hasil output akan mencetak f dengan nilai 20.0 dan fll dengan nilai 10.0, sesuai format yang telah ditentukan. Program ini memperlihatkan cara mendefinisikan variabel, menetapkan nilai, dan mencetak hasilnya dalam bahasa pemrograman Java.

2.

```
*Asign - Notepad
File Edit Format View Help
public class Asign {
    /**
     * @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:\Tugas 1>javac Asign.java
C:\Tugas 1>java Asign
hello
Ini nilai i : 5
```

Penjelasan: Program dimulai dengan mendeklarasikan kelas `Asign` dan metode `main`, kemudian variabel `i` bertipe `int` dideklarasikan tanpa nilai awal. Selanjutnya, program mencetak teks "hello" dan berpindah ke baris baru, lalu memberikan nilai 5 ke variabel `i`. Setelah itu, program mencetak teks "Ini nilai i : 5" dan berakhir setelah menampilkan output yang telah ditentukan.

3.

```
ASIGNi - Notepad
File Edit Format View Help
/* Deskripsi : */
/* 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);
    }
}
```

```

C:\Tugas 1>javac ASIGN1.java

C:\Tugas 1>Java ASIGN1
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
C:\Tugas 1>_

```

Penjelasan: Program ini mendeklarasikan berbagai jenis variabel, termasuk bilangan bulat (short, int, long), karakter (char), dan bilangan riil (double, float). Variabel `c` diinisialisasi dengan nilai 65, yang direpresentasikan sebagai karakter `A` berdasarkan kode ASCII, sementara `c1` langsung diberi karakter `'Z'`. Selanjutnya, program mencetak nilai karakter dalam bentuk aslinya serta sebagai bilangan ASCII. Setelah itu, program menampilkan nilai variabel bilangan bulat dan bilangan riil ke layar. Program ini menunjukkan bagaimana mendefinisikan variabel, menginisialisasinya, dan mencetak hasilnya dalam Java.

4.

```

BacaData - Notepad
File Edit Format View Help
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);
    }
}

```

```


C:\Tugas 1>javac BacaData.java

C:\Tugas 1>Java BacaData
Contoh membaca dan menulis, ketik nilai integer:
50
Nilai yang dibaca : 50
C:\Tugas 1>

```

Penjelasan: Program ini meminta pengguna memasukkan sebuah bilangan bulat menggunakan **class Scanner** dan kemudian mencetak kembali nilai yang telah dimasukkan. Diawali dengan deklarasi variabel `a` sebagai tempat menyimpan nilai integer dan objek `Scanner` untuk membaca input. Program menampilkan pesan instruksi kepada pengguna, lalu menangkap angka yang diketik dan menyimpannya dalam variabel `a`. Setelah itu, nilai tersebut ditampilkan kembali sebagai output. Jika perintah `masukan.nextInt()` ; diketik dua kali tanpa menyediakan dua angka, program akan menunggu input tambahan atau menghasilkan error.

5.



```

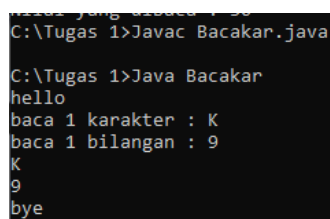
Bacakar - Notepad
File Edit Format View Help
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

//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");
    }
}

```



```

C:\Tugas 1>javac Bacakar.java

C:\Tugas 1>Java Bacakar
hello
baca 1 karakter : K
baca 1 bilangan : 9
K
9
bye

```

Penjelasan: Program ini dimulai dengan menampilkan teks "hello" sebagai output awal. Selanjutnya, program meminta pengguna untuk memasukkan satu karakter dengan menampilkan pesan "baca 1 karakter :", kemudian membaca input tersebut menggunakan **BufferedReader** dan menyimpannya dalam variabel `cc`. Setelah itu, program meminta pengguna memasukkan sebuah bilangan bulat dengan menampilkan pesan "baca

1 bilangan :", lalu membaca input yang diberikan, mengonversinya menjadi tipe `int`, dan menyimpannya dalam variabel `bil`. Terakhir, program mencetak kembali karakter dan bilangan yang telah dimasukkan oleh pengguna.

6.

```
Casting1 - Notepad
File Edit Format View Help
/*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.f,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
    }
}
```

```
C:\Tugas 1>JavaC Casting1.java
C:\Tugas 1>Java Casting1
5.0
6.0
2
3.200000047683716
53
53.0
53.0
3
3.14
```

Penjelasan: Program ini mendemonstrasikan proses **casting** atau konversi tipe data dalam Java menggunakan tipe data primitif. Dimulai dengan mendeklarasikan variabel bertipe `int`, `float`, `char`, dan `double`, program kemudian melakukan konversi eksplisit antara tipe-tipe tersebut. Nilai integer dikonversi ke `float` dan `double`, sedangkan nilai `float` dikonversi ke `int` dan `double`. Karakter `char` juga dikonversi ke `int`, `float`, dan `double`, yang menghasilkan nilai ASCII dari karakter tersebut. Selain itu, nilai `double` dikonversi ke `int` dan `float`, yang menyebabkan hilangnya bagian desimal dalam konversi ke integer. Hasil konversi ini ditampilkan menggunakan `System.out.println()`, yang menunjukkan bagaimana Java menangani perubahan tipe data dalam berbagai situasi.

7.

```

Casting2 - Notepad
File Edit Format View Help
/*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(a); /*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+"\n : "+l);
    }
}

```

```

C:\Tugas 1>javac Casting2.java

C:\Tugas 1>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

```

Penjelasan: Program ini menunjukkan cara melakukan konversi tipe data menggunakan **kelas bawaan Java**. Dimulai dengan mendeklarasikan variabel bertipe **int, float, char, double, dan String**, program kemudian melakukan konversi **String ke Integer, Double, dan Float** menggunakan metode `parseInt()`, `parseDouble()`, dan `parseFloat()`. Selanjutnya, program mengonversi **Integer, Karakter, dan Float ke String** menggunakan `String.valueOf()`. Selain itu, dilakukan konversi antara **Integer dan Double** menggunakan metode `valueOf()`, yang memungkinkan perubahan tipe data tanpa kehilangan presisi secara signifikan. Hasil dari setiap konversi kemudian ditampilkan ke layar menggunakan `System.out.println()`.

8.

```

Ekspresi - Notepad
File Edit Format View Help
/* 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*/
    }
}

```

```
C:\Tugas 1>javac Ekspresi.java

C:\Tugas 1>java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
```

Penjelasan: Program ini mendemonstrasikan **pemakaian operator kondisional (ternary operator)** dalam Java untuk memilih nilai berdasarkan suatu kondisi. Program dimulai dengan mendeklarasikan dua variabel, `x` dan `y`, masing-masing bernilai **1 dan 2**. Kemudian, program mencetak nilai kedua variabel tersebut. Setelah itu, program menggunakan **operator ternary (condition) `? true_value : false_value`** untuk mengevaluasi ekspresi `(x < y) ? x : y`. Jika kondisi `x < y` benar, maka hasilnya adalah `x`, tetapi jika salah, hasilnya adalah `y`. Dalam kasus ini, karena `x` lebih kecil dari `y`, ekspresi akan menghasilkan nilai `x`, yaitu **1**. Hasil evaluasi ini kemudian ditampilkan ke layar menggunakan `System.out.print()`.

9.

```
Ekspresi1 - Notepad
File Edit Format View Help
/* pembagian integer, casting */
public class Ekspresi1 {

    /**
     * @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);
    }
}
```

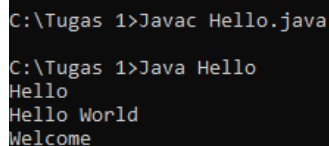
```
C:\Tugas 1>java Ekspresi1
x/y (format integer) = 0
x/y (format float) = 0
x/y (format integer) = 0.5
x/y (format float) = 0.5
float(x)/float(y) (format integer) = 0.5
float(x)/float(y) (format float) = 0.5
x/y (format integer) = 3
x/y (format float) = 3
```

Penjelasan: Program ini menunjukkan bagaimana **pembagian integer dan casting** bekerja dalam Java. Dimulai dengan mendeklarasikan variabel **`x` dan `y`** bertipe `int`, serta **`fx` dan `fy`**

bertipe `float`, program kemudian melakukan berbagai operasi pembagian. Saat x/y dihitung tanpa casting, hasilnya berupa bilangan bulat karena Java secara otomatis membulatkan ke bawah dalam pembagian integer. Untuk menghindari hasil nol atau bilangan bulat, variabel `x` dan `y` disalin ke `fx` dan `fy`, yang bertipe `float`, sehingga pembagian menghasilkan bilangan desimal. Selain itu, dilakukan **casting langsung** dengan `(float)x / (float)y`, yang memastikan hasil tidak kehilangan presisi. Terakhir, nilai `x` dan `y` diubah untuk menunjukkan bagaimana pembagian integer dan float berperilaku dengan angka yang berbeda. Program ini memperjelas pentingnya casting saat bekerja dengan operasi pembagian agar hasil lebih akurat sesuai kebutuhan.

10.

```
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");  
    }  
}
```



```
C:\Tugas 1>javac Hello.java  
  
C:\Tugas 1>java Hello  
Hello  
Hello World  
Welcome
```

Penjelasan: Program ini menunjukkan cara **menampilkan teks ke layar** menggunakan `System.out.print()` dan `System.out.println()` dalam Java.

Dimulai dengan mencetak kata **"Hello"** menggunakan `System.out.print()`, yang menampilkan teks tanpa berpindah ke baris baru. Selanjutnya, perintah `System.out.print("\nHello ")` mencetak "Hello" dengan **karakter newline** (`\n`) yang membuat teks dimulai dari baris baru. Kemudian, `System.out.println("World")` mencetak "World" dan **secara otomatis pindah ke baris berikutnya**. Terakhir, `System.out.println("Welcome")` mencetak "Welcome" pada baris baru.

Program ini memperlihatkan perbedaan antara `print()` yang mencetak teks dalam satu baris tanpa berpindah ke baris baru, serta `println()` yang mencetak teks dan otomatis berpindah ke baris berikutnya.

11.

```
File Edit Format View Help
/* 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);
    }
}
```

```
C:\Tugas 1>javac Incr.java
```

```
C:\Tugas 1>java Incr
Nilai i : 5
Nilai j : 3
```

Penjelasan: Program ini mendemonstrasikan **efek dari operator ++ (increment)** dalam Java, yang digunakan untuk menambah nilai variabel sebanyak satu. Dimulai dengan deklarasi variabel `i` dan `j`, program menginisialisasi `i` dengan nilai **3**, kemudian menetapkan `j = i++`. Karena `i++` adalah **post-increment**, nilai `j` akan disimpan terlebih dahulu sebagai **3**, kemudian nilai `i` bertambah menjadi **4** setelah pernyataan ini dieksekusi. Selanjutnya, perintah `System.out.println()` mencetak:

- **(++i)**, yaitu **pre-increment**, yang berarti `i` bertambah **sebelum** dicetak, sehingga menjadi **5**.
- `j`, yang tetap bernilai **3**, karena pada saat assignment `j = i++`, nilai `i` belum bertambah.

Program ini menunjukkan perbedaan antara **pre-increment (++i)** yang menaikkan nilai sebelum digunakan, dan **post-increment (i++)** yang menaikkan nilai setelah digunakan dalam ekspresi.

12.

```
Oper1 - Notepad
File Edit Format View Help
/* 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 */

    }
}
```

```
C:\Tugas 1>Javac Oper1.java
```

```
C:\Tugas 1>Java Oper1
```

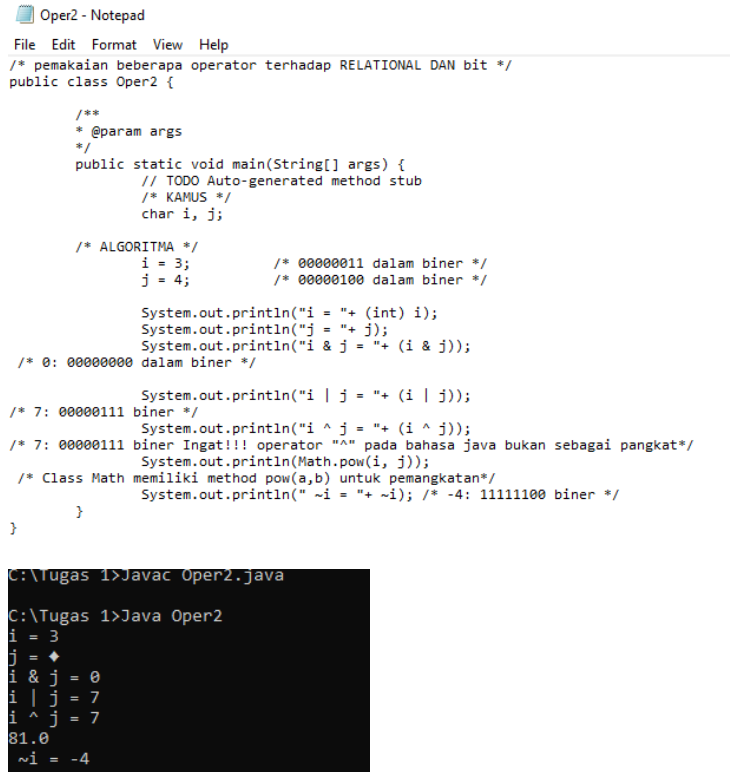
```
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

Penjelasan: Program ini mendemonstrasikan **penggunaan operator bitwise** dalam Java, yang beroperasi pada representasi biner angka. Dimulai dengan mendeklarasikan variabel *n*, *x*, dan *y*, yang masing-masing memiliki nilai **10 (1010₂)**, **1 (1₂)**, dan **2 (10₂)**. Kemudian, program menerapkan beberapa operasi bitwise:

1. ***n* & 8 (Bitwise AND)**
 - o *n* (1010₂) **AND** 8 (1000₂) menghasilkan 1000₂ atau **8**.
2. ***x* & ~8 (Bitwise AND dengan NOT)**
 - o ~8 menghasilkan **komplemen bit** dari 1000₂, yaitu 0111₂.
 - o *x* (0001₂) **AND** 0111₂ menghasilkan 0001₂ atau **1**.
3. ***y* << 2 (Bitwise Left Shift)**
 - o *y* (10₂) digeser **2 bit ke kiri**, menjadi 1000₂ atau **8**.
4. ***y* >> 3 (Bitwise Right Shift)**
 - o *y* (10₂) digeser **3 bit ke kanan**, menjadi 0000₂ atau **0**.

Program ini memperlihatkan bagaimana **operator bitwise** seperti & (AND), ~ (NOT), << (left shift), dan >> (right shift) bekerja dalam manipulasi angka di tingkat biner.

13.



```
Oper2 - Notepad
File Edit Format View Help
/* 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 */
    }
}

C:\Tugas 1>Javac Oper2.java
C:\Tugas 1>Java Oper2
i = 3
j = 4
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
```

Penjelasan: program ini mendemonstrasikan penggunaan **operator relasional dan bitwise** dalam Java. Variabel *i* dan *j* diberikan nilai 3 dan 4, kemudian dilakukan berbagai operasi pada nilai tersebut. Pertama, program mencetak nilai *i* dan *j* dalam bentuk bilangan bulat. Kemudian, program melakukan operasi **bitwise AND (&)**, yang menghasilkan 0 karena tidak ada bit yang sama-sama bernilai 1. Selanjutnya, dilakukan **bitwise OR (|)**, yang menghasilkan 7 karena menggabungkan semua bit yang bernilai 1 dari kedua angka. Operasi **bitwise XOR (^)** juga menghasilkan 7 karena hanya mempertahankan bit yang berbeda di antara *i* dan *j*. Karena **operator ^ bukan digunakan untuk pangkat di Java**, program menggunakan `Math.pow(i, j)` untuk menghitung 3 pangkat 4, yang menghasilkan 81. Terakhir, program menggunakan **bitwise NOT (~)** untuk membalik semua bit pada *i*, yang menghasilkan -4 dalam representasi bilangan dua komplemen.

14.

```
Oper3 - Notepad
File Edit Format View Help
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 */
    }
}
```

```
C:\Tugas 1>javac Oper3.java
```

```
C:\Tugas 1>java Oper3
true
false
true
true
true
```

Penjelasan: Program ini mendemonstrasikan penggunaan **operator logika** dalam Java, yaitu **&&**, **&**, **||**, dan **|**. Pertama, program mengevaluasi **true && true**, yang menghasilkan **true**, sehingga nilai **true** dicetak ke layar. Kemudian, program mencoba mengeksekusi **true & true**, tetapi dalam perintah cetak, terjadi kesalahan karena `System.out.println(true & false);`, yang seharusnya mencetak **false** jika kondisi sebelumnya benar. Selanjutnya, program mengeksekusi **if (true)**, yang selalu bernilai **true**, sehingga perintah cetak dijalankan dan menghasilkan **true**. Lalu, evaluasi **true || true** juga menghasilkan **true**, sehingga perintah cetak kembali dijalankan. Terakhir, **true | false** menghasilkan **true**, sehingga nilai **true** kembali dicetak. Secara keseluruhan, program ini menunjukkan bagaimana **operator logika AND (&& dan &) serta OR (|| dan |)** bekerja dalam Java.

15.

```
Oper4 - Notepad
File Edit Format View Help
/* Operator ternern */
public class Oper4 {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* KAHUS */
        int i = 0; /* perhatikan int i,j=0 bukan seperti ini */
        int j = 0;
        char c = 0; 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);
    }
}
```

```
C:\Tugas 1>Javac Oper4.java
```

```
C:\Tugas 1>Java Oper4
```

```
Nilai e = 10
```

```
Nilai k = 0
```

```
Nilai k = 4
```

Penjelasan: Program ini mendemonstrasikan penggunaan **operator ternary** (**? :**) dalam Java, yang digunakan untuk membuat keputusan berdasarkan kondisi tertentu. Pertama, variabel **i** dan **j** diinisialisasi dengan **0**, sedangkan variabel **c** dan **d** memiliki nilai **8** dan **10**. Operator ternary digunakan untuk menentukan nilai **e**, dengan membandingkan **c** dan **d**. Karena **d** lebih besar, nilai **e** menjadi **10**. Selanjutnya, operator ternary juga digunakan untuk menentukan nilai **k**, dengan membandingkan **i** dan **j**, di mana hasilnya adalah **0** karena **i** dan **j** sama besar. Setelah itu, nilai **i** diubah menjadi **2** dan **j** menjadi **3**. Operator ternary kembali digunakan untuk menentukan nilai **k**, tetapi kali ini dengan **post-increment** (**i++** dan **j++**). Karena **i++** tidak lebih besar dari **j++**, maka hasilnya adalah nilai **j** setelah operasi increment, yaitu **4**. Secara keseluruhan, program ini menunjukkan bagaimana operator ternary dapat digunakan untuk memilih nilai berdasarkan kondisi, serta bagaimana **post-increment** memengaruhi hasil perbandingan.

16.

Operator - Notepad

File Edit Format View Help

```
/* 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);
```

```
     }
```

```
C:\Tugas 1>Java Oprator
```

```
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output
```

Penjelasan: Program ini mendemonstrasikan penggunaan berbagai jenis variabel dalam Java, termasuk boolean, integer, dan float. Operasi logika diterapkan menggunakan operator AND, OR, NOT, dan XOR untuk menentukan hasil berdasarkan nilai boolean. Operasi aritmatika dilakukan pada bilangan bulat dan bilangan desimal, termasuk penjumlahan, pengurangan, perkalian, pembagian, dan modulus. Selain itu, program juga menggunakan operator relasional untuk membandingkan nilai integer dan float, menghasilkan nilai boolean yang menunjukkan hubungan antara dua angka. Namun, karena tidak ada perintah untuk mencetak hasil ke layar, output dari operasi ini tidak terlihat kecuali jika ditambahkan perintah cetak.

17.

```
Oprator1 - Notepad
File Edit Format View Help
public class Oprator1 {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // 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");

        // Operasi Boolean
        Bool1 = true;
        Bool2 = false;

        TF = Bool1 && Bool2; // Boolean AND
        System.out.println("Bool1 AND Bool2: " + TF);

        TF = Bool1 || Bool2; // Boolean OR
        System.out.println("Bool1 OR Bool2: " + TF);

        TF = !Bool1; // NOT
        System.out.println("NOT Bool1: " + TF);

        TF = Bool1 ^ Bool2; // XOR
        System.out.println("Bool1 XOR Bool2: " + TF);

        // Operasi numerik
        i = 5;
        j = 2;

        hsl = i + j;
        System.out.println("i + j: " + hsl);

        hsl = i - j;
        System.out.println("i - j: " + hsl);

        hsl = i / j;
        System.out.println("i / j: " + hsl);

        hsl = i * j;
        System.out.println("i * j: " + hsl);

        hsl = i % j; // sisa modulo
        System.out.println("i % j: " + hsl);

        // Operasi numerik dengan float
```

```

        TF = (x != y);
        System.out.println("x != y: " + TF);

        TF = (x < y);
        System.out.println("x < y: " + TF);

        TF = (x > y);
        System.out.println("x > y: " + TF);

        TF = (x <= y);
        System.out.println("x <= y: " + TF);

        TF = (x >= y);
        System.out.println("x >= y: " + TF);
    }
}

```

```

C:\Tugas 1>javac Oprator1.java

C:\Tugas 1>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

```

Penjelasan: Program ini dimulai dengan mendeklarasikan variabel-variabel boolean, integer, dan float. Kemudian, program mencetak teks awal yang meminta pengguna membaca teks dan menampilkan output. Selanjutnya, program menguji berbagai operasi logika seperti AND (&&), OR (||), NOT (!), dan XOR (^) dengan dua variabel boolean. Hasil dari setiap operasi ini ditampilkan di layar. Kemudian, program melakukan operasi aritmetika dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulo menggunakan variabel bertipe integer (`int`). Hasil setiap operasi juga dicetak ke layar. Selain itu, program menguji operasi relasional seperti sama dengan (`==`), tidak sama dengan (`!=`), lebih kecil (`<`), lebih besar (`>`), lebih kecil sama dengan (`<=`), dan lebih besar sama dengan (`>=`) dengan variabel bertipe integer dan float. Hasil dari setiap perbandingan juga dicetak ke layar. Program ini bertujuan untuk menunjukkan bagaimana berbagai operator bekerja dalam bahasa pemrograman Java.