

**PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Audita Cahyani Amiruddin
Stambuk : 13020230101
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.java - 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 f11;
        /* Algoritma */
        f11=10.0f;
        System.out.println ("f : "+f +
                               "\nf11: "+f11);
    }
}
```

```
D:\Tugas PBO>javac Asgdll.java
```

```
D:\Tugas PBO>java Asgdll
f : 20.0
f11: 10.0
```

Penjelasan : Kode Java ini dimulai dengan mendeklarasikan dua variabel: f yang bertipe float dan diberi nilai 20.0f, serta f11 yang bertipe double. Kemudian, variabel f11 diberi nilai 10.0f. Setelah itu, kode ini mencetak nilai dari kedua variabel ke konsol menggunakan perintah System.out.println(). Output yang dihasilkan akan menampilkan nilai f (20.0) dan nilai f11 (10.0) di baris yang berbeda.

2.

```
Untitled - 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);
    }
}
```

```
D:\Tugas PBO>javac Asign.java
```

```
D:\Tugas PBO>java Asign
hello
Ini nilai i :5
```

Penjelasan : Kode Java ini, bernama ASIGNi, memiliki fungsi yang sangat sederhana: menampilkan pesan "hello" dan nilai sebuah variabel integer. Pertama, di dalam metode main, sebuah variabel integer bernama i dideklarasikan. Kemudian, program mencetak kata "hello" ke konsol, diikuti dengan karakter newline (\n) yang membuat baris baru. Selanjutnya, variabel i diberi nilai 5. Terakhir, program mencetak pesan "Ini nilai i : " yang diikuti dengan nilai variabel i (yaitu 5) ke konsol. Jadi, ketika kode ini dijalankan, output yang dihasilkan akan terdiri dari dua baris: baris pertama menampilkan "hello", dan baris kedua menampilkan "Ini nilai i :5".

```
3.  /* 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);
        }
    }
```

```
D:\Tugas PBO>javac ASIGNi.java

D:\Tugas PBO>java ASIGNi
Karakter = A
Karakter = Z
Karakter = A
Karakter = Z
Bilangan integer (short) = 1
\t(int) = 1
\t(long)= 10000
Bilangan Real x = 50.20000076293945
Bilangan Real y = 50.2
```

Penjelasan : Kode Java ASIGNi ini dirancang untuk menunjukkan bagaimana cara mendeklarasikan dan menginisialisasi berbagai tipe data dasar dalam Java. Diawali dengan deklarasi variabel-variabel seperti ks bertipe short, ki bertipe int, dan kl bertipe long, yang semuanya merupakan tipe data bilangan bulat dengan rentang nilai yang berbeda. Kemudian, variabel c dan c1 dideklarasikan sebagai tipe char untuk menyimpan karakter, di mana c diinisialisasi dengan nilai ASCII 65 ('A') dan c1 dengan karakter 'Z'. Selanjutnya, variabel x dan y dideklarasikan sebagai tipe double dan float untuk menyimpan bilangan riil. Bagian inti dari kode ini adalah serangkaian perintah System.out.println() yang mencetak nilai-nilai variabel tersebut ke konsol. Menariknya, karakter dicetak baik sebagai karakter maupun nilai integer ASCII-nya, menyoroti fleksibilitas tipe data char.

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:\Tugas PBO>javac BacaData.java
```

```
D:\Tugas PBO>java BacaData
```

```
Contoh membaca dan menulis, ketik nilai integer:
```

```
11
```

```
Nilai yang dibaca : 11
```

Penjelasan : Kode Java BacaData ini adalah contoh sederhana bagaimana cara menerima input bilangan bulat dari pengguna melalui konsol. Pertama-tama, program mengimpor kelas Scanner dari paket java.util, yang merupakan alat untuk membaca input dari berbagai sumber, termasuk keyboard. Kemudian, di dalam metode main, sebuah variabel integer a dan sebuah objek Scanner bernama masukan dideklarasikan. Program lalu mencetak pesan yang meminta pengguna untuk memasukkan nilai integer. Objek Scanner yang dibuat kemudian digunakan untuk membaca input integer yang dimasukkan pengguna melalui metode nextInt() dan menyimpannya dalam variabel a. Terakhir, program mencetak kembali nilai yang telah dibaca ke konsol. Jika baris kode a = masukan.nextInt(); diubah menjadi masukan.nextInt(); saja, maka nilai integer yang dimasukkan pengguna akan dibaca tetapi tidak disimpan di variabel manapun, sehingga nilainya akan hilang dan tidak dapat digunakan dalam program. Secara keseluruhan, kode ini menunjukkan dasar-dasar interaksi dengan pengguna melalui input konsol di Java.

5.

```

bacakar.java - 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");
    }
}

```

```

D:\Tugas PBO>javac Bacakar.java

```

```

D:\Tugas PBO>java Bacakar
hello
baca 1 karakter : d
baca 1 bilangan : 2
d
2
bye

```

Penjelasan: Kode Java Bacakar ini dirancang untuk membaca input dari pengguna, baik berupa karakter tunggal maupun bilangan bulat, menggunakan `BufferedReader`. Pertama, kode mengimpor kelas-kelas yang diperlukan dari paket `java.io`, yaitu `BufferedReader`, `IOException`, dan `InputStreamReader`, yang memungkinkan program untuk membaca input dari aliran karakter. Di dalam metode `main`, variabel `cc` (karakter) dan `bil` (bilangan bulat) dideklarasikan. Dua objek `BufferedReader` dibuat, `dataIn` dan `dataIn`, keduanya terhubung ke input standar (keyboard) melalui `InputStreamReader`. Program kemudian mencetak "hello" ke konsol. Selanjutnya, meminta pengguna untuk memasukkan satu karakter, yang kemudian dibaca menggunakan `dataIn.readLine().charAt(0)` dan disimpan di variabel `cc`. Setelah itu, program meminta pengguna untuk memasukkan satu bilangan bulat, yang dibaca menggunakan `dataIn.readLine()` dan diubah menjadi integer menggunakan `Integer.parseInt()` sebelum disimpan di variabel `bil`. Kode yang dikomentari menunjukkan penggunaan `JOptionPane` dari `Swing`, yang akan menampilkan dialog input dan output, namun itu sedang tidak aktif. Akhirnya, program mencetak nilai `cc` dan `bil` ke konsol, diikuti dengan "bye".

6.

```

*Untitled - 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

    }
}

```

```

D:\Tugas PBO>javac Casting1.java

D:\Tugas PBO>java Casting1
5.0
6.0
2
3.200000047683716
53
53.0
53.0
3
3.14

```

Penjelasan: Kode Java Casting1 ini memperlihatkan bagaimana proses casting atau konversi tipe data primitif dilakukan secara eksplisit. Diawali dengan deklarasi beberapa variabel dengan tipe data yang berbeda, seperti int, float, char, dan double. Kemudian, setiap variabel dikonversi ke tipe data lain menggunakan operator casting (tipe_data), dan hasilnya dicetak ke konsol. Misalnya, variabel a yang bertipe int dikonversi ke float menggunakan (float)a. Begitu pula, variabel g yang bertipe char dikonversi ke int, float, dan double, di mana konversi ini menghasilkan nilai ASCII dari karakter tersebut. Proses casting ini memungkinkan kita untuk mengubah tipe data suatu nilai, tetapi perlu diingat bahwa konversi dari tipe data yang lebih besar ke yang lebih kecil dapat menyebabkan hilangnya informasi, seperti bagian desimal yang dihilangkan saat mengkonversi double ke int.

7.

```

/*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=2.5f,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);
    }
}

```

```
D:\Tugas PBO>javac Casting2.java
```

```
D:\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

```

Penjelasan: Kode Java Casting2 ini mendemonstrasikan bagaimana konversi tipe data dilakukan, khususnya antara tipe data primitif dan tipe data kelas. Diawali dengan deklarasi variabel-variabel dengan tipe data int, float, char, double, dan String. Bagian penting dari kode ini adalah bagaimana String dikonversi menjadi tipe data numerik menggunakan metode seperti Integer.parseInt(), Double.parseDouble(), dan Float.parseFloat(). Sebaliknya, tipe data numerik dan karakter dikonversi menjadi String menggunakan String.valueOf(). Kode ini juga menunjukkan penggunaan kelas pembungkus (wrapper classes) seperti Integer dan Double untuk melakukan konversi antar tipe data numerik. Misalnya, integer dikonversi menjadi Double dan sebaliknya. Proses casting ini penting dalam pemrograman Java, terutama ketika berhadapan dengan input dari pengguna yang seringkali berupa String. Secara keseluruhan, kode ini memberikan contoh yang jelas tentang bagaimana konversi tipe data dilakukan dalam konteks tipe data primitif dan kelas.

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:\Tugas PBO>javac Ekspresi.java

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

Penjelasan: Kode Java Ekspresi ini secara khusus menunjukkan penerapan operator kondisional, atau yang sering disebut sebagai operator ternary, dalam konteks yang sangat sederhana. Diawali dengan deklarasi dua variabel integer, x dan y, yang masing-masing diberi nilai 1 dan 2. Kemudian, program mencetak nilai kedua variabel tersebut ke konsol. Bagian inti dari kode ini adalah penggunaan operator kondisional $(x < y) ? x : y$. Ekspresi ini memeriksa apakah x kurang dari y. Jika kondisi ini benar, maka nilai x akan dihasilkan; jika tidak, nilai y yang akan dihasilkan. Dalam kasus ini, karena 1 memang kurang dari 2, maka nilai x (yaitu 1) yang dicetak. Penggunaan tanda kurung pada $((x < y) ? x : y)$ berfungsi untuk mempertegas bahwa kode tersebut adalah satu kesatuan perintah. Operator kondisional ini merupakan cara ringkas untuk menulis pernyataan if-else sederhana dalam satu baris, dan sering digunakan untuk memilih antara dua nilai berdasarkan kondisi tertentu.

9.  *Untitled - Notepad

```
File Edit Format View Help
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);

    }
}
```

```
D:\Tugas PBO>javac Ekspresi1.java

D:\Tugas PBO>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: Kode Java Ekspresi ini menunjukkan bagaimana pembagian integer dan casting mempengaruhi hasil operasi aritmatika. Kode dimulai dengan mendeklarasikan variabel integer x dan y, dan variabel float fx dan fy. Ketika x dibagi dengan y (1/2), hasilnya adalah 0 karena pembagian integer menghilangkan bagian desimal. Meskipun dicetak dalam format float, hasilnya tetap 0 karena pembagian dilakukan pada integer. Selanjutnya, x dan y dikonversi menjadi float dan disimpan di fx dan fy. Ketika fx dibagi dengan fy (1.0/2.0), hasilnya adalah 0.5, tetapi jika dicetak sebagai integer, hasilnya menjadi 0. Untuk mendapatkan hasil float yang benar, pembagian harus dilakukan pada float dan dicetak sebagai float. Casting eksplisit juga digunakan untuk mengubah x dan y menjadi float sebelum pembagian, yang menghasilkan 0.5. Kode ini kemudian mengubah nilai x dan y menjadi 10 dan 3, dan melakukan pembagian integer, yang menghasilkan 3.

10.

```
Untitled - Notepad
File Edit Format View Help
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:\Tugas PBO>javac Hello.java
```

```
D:\Tugas PBO>java Hello
```

```
Hello
```

```
Hello World
```

```
Welcome
```

Penjelasan: Kode Java Hello ini adalah contoh sederhana untuk menampilkan teks ke konsol. Program ini dimulai dengan mendefinisikan kelas Hello dan metode main, yang merupakan titik awal eksekusi program. Kemudian, program menggunakan System.out.print untuk mencetak string "Hello" ke konsol. Perintah ini tidak menambahkan baris baru setelah mencetak teks. Selanjutnya, program mencetak baris baru (\n) dan string "Hello " menggunakan System.out.print lagi. Kemudian, program menggunakan System.out.println untuk mencetak string "World" dan menambahkan baris baru setelahnya. Terakhir, program menggunakan System.out.println lagi untuk mencetak string "Welcome" dan menambahkan baris baru. Perbedaan utama antara System.out.print dan System.out.println adalah bahwa System.out.println menambahkan baris baru setelah mencetak teks, sedangkan System.out.print tidak.

11.

```

Untitled - Notepad
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);
    }
}

```

```

D:\Tugas PBO>javac Incr.java

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

```

Penjelasan; Kode Java Incr ini secara khusus menggambarkan perbedaan antara penggunaan operator increment postfix (i++) dan prefix (++i). Program dimulai dengan mendeklarasikan dua variabel integer, i dan j. Variabel i kemudian diinisialisasi dengan nilai 3. Pada baris j = i++;, nilai i yang awalnya 3 diberikan kepada j terlebih dahulu, dan setelah itu, nilai i ditingkatkan menjadi 4. Ini adalah contoh penggunaan operator increment postfix. Selanjutnya, pada baris System.out.println ("Nilai i : " + (++i) + "\nNilai j : " + j);, operator increment prefix ++i digunakan. Ini berarti nilai i ditingkatkan terlebih dahulu menjadi 5, dan nilai yang baru ini (5) digunakan dalam perintah System.out.println. Sementara itu, nilai j tetap 3, karena hanya diinisialisasi sekali dengan nilai i sebelum i ditingkatkan oleh operator postfix. Perbedaan utama antara postfix dan prefix adalah urutan eksekusi: postfix menggunakan nilai variabel terlebih dahulu, lalu meningkatkannya, sedangkan prefix meningkatkannya terlebih dahulu, lalu menggunakan nilai yang baru.

12.

```

*Untitled - 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 */

    }
}

```

```
D:\Tugas PBO>javac Oper1.java
```

```
D:\Tugas PBO>java Oper1
```

```

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

```

Penjelasan: Kode Java Oper1 ini mendemonstrasikan penggunaan beberapa operator bitwise, yang bekerja langsung pada representasi biner dari nilai-nilai integer. Dimulai dengan deklarasi tiga variabel integer, n, x, dan y, yang masing-masing diinisialisasi dengan nilai 10, 1, dan 2. Program kemudian mencetak nilai-nilai ini ke konsol. Bagian utama dari kode ini adalah penggunaan operator bitwise seperti AND (&), NOT (~), left shift (<<), dan right shift (>>). Operasi n & 8 melakukan operasi AND bitwise antara 10 (1010 dalam biner) dan 8 (1000 dalam biner), menghasilkan 8 (1000), karena hanya bit yang sama-sama 1 yang menghasilkan 1. Operasi x & ~8 melakukan operasi AND bitwise antara 1 dan hasil dari NOT bitwise dari 8. Operator NOT membalik semua bit dari 8, dan kemudian operasi AND dilakukan. Operasi y << 2 melakukan left shift bitwise pada y (2) sebanyak 2 posisi, menggeser bit ke kiri dan menambahkan 0 di sebelah kanan, menghasilkan 8. Operasi y >> 3 melakukan right shift bitwise pada y sebanyak 3 posisi, menggeser bit ke kanan dan menambahkan 0 di sebelah kiri, menghasilkan 0. Operator bitwise ini sering digunakan dalam pemrograman tingkat rendah untuk manipulasi bit dan optimasi kinerja.

13.

```

Untitled - 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 */
    }
}

```

```

D:\Tugas PBO>javac Oper2.java

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

```

Penjelasan: Kode Java Oper2 ini mengilustrasikan penggunaan operator bitwise pada tipe data karakter. Program ini dimulai dengan mendeklarasikan dua variabel karakter, i dan j, yang masing-masing diinisialisasi dengan nilai 3 dan 4. Kemudian, program mencetak nilai-nilai ini ke konsol, dengan i di-cast ke integer untuk menampilkan nilai numeriknya. Bagian utama dari kode ini adalah penggunaan operator bitwise seperti AND (&), OR (|), XOR (^), dan NOT (~). Operasi i & j melakukan operasi AND bitwise antara 3 (00000011 dalam biner) dan 4 (00000100 dalam biner), menghasilkan 0 (00000000), karena tidak ada bit yang sama-sama 1 pada posisi yang sama. Operasi i | j melakukan operasi OR bitwise, menghasilkan 7 (00000111), karena setiap bit yang 1 pada salah satu atau kedua operand menghasilkan 1. Operasi i ^ j melakukan operasi XOR bitwise, juga menghasilkan 7, karena setiap bit yang berbeda menghasilkan 1. Kode ini juga menunjukkan bahwa ^ bukan operator pangkat di Java, dengan menggunakan Math.pow(i, j) untuk menghitung 3 pangkat 4. Terakhir, operasi ~i melakukan operasi NOT bitwise pada i, membalik semua bit dan menghasilkan -4.

14.

```

Untitled - 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 */
    }
}

```

```
D:\Tugas PBO>javac Oper3.java
```

```
D:\Tugas PBO>java Oper3
```

```
true
false
true
true
true
```

Penjelasan: Kode Java Oper3 ini secara khusus membandingkan penggunaan operator logika dan operator bitwise dalam konteks pernyataan if. Program ini menggunakan kombinasi operator logika && (AND) dan || (OR), serta operator bitwise & dan |, untuk mengevaluasi kondisi boolean. Perbedaan utama terletak pada cara evaluasi dilakukan. Operator logika && dan || menggunakan evaluasi short-circuit, yang berarti jika hasil dapat ditentukan dari operand pertama, operand kedua tidak akan dievaluasi. Misalnya, dalam true && true, kedua operand dievaluasi, tetapi jika operand pertama adalah false, operand kedua tidak akan dievaluasi. Demikian pula, dalam true || true, jika operand pertama adalah true, operand kedua tidak akan dievaluasi. Sebaliknya, operator bitwise & dan | selalu mengevaluasi kedua operand, terlepas dari nilai operand pertama. Selain itu, operator logika && dan || dirancang untuk bekerja dengan nilai boolean, sedangkan operator bitwise & dan | dapat digunakan dengan nilai integer maupun boolean. Kode ini juga menunjukkan bahwa meskipun kedua jenis operator dapat digunakan dalam kondisi if, mereka memiliki perilaku yang berbeda, terutama dalam hal evaluasi dan tipe data yang mereka operasikan.

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:\Tugas PBO>javac Oper4.java
```

```
D:\Tugas PBO>java Oper4
```

```
Nilai e = 10
Nilai k = 0
Nilai k = 4
```

Penjelasan: Kode Java Oper4 ini berfokus pada demonstrasi penggunaan operator terner, atau operator kondisional, yang memungkinkan penulisan ekspresi kondisional dalam satu baris. Program ini dimulai dengan mendeklarasikan beberapa variabel, termasuk i, j, c, d, dan k. Penggunaan operator terner terlihat dalam baris `int e = (((int)c > (int)d) ? c : d);`, yang membandingkan nilai c dan d setelah keduanya di-cast ke integer. Karena c (8) tidak lebih besar dari d (10), maka nilai d (10) diberikan kepada e. Selanjutnya, `int k = ((i > j) ? i : j);` juga menggunakan operator terner untuk membandingkan i dan j, yang keduanya bernilai 0. Karena i tidak lebih besar dari j, maka nilai j (0) diberikan kepada k. Kemudian, program mencetak nilai e dan k ke konsol. Bagian terakhir dari kode ini menunjukkan penggunaan operator terner dengan postfix increment. Variabel i dan j diubah menjadi 2 dan 3, dan ekspresi `k = ((i++ > j++) ? i : j);` dijalankan. Meskipun i (2) tidak lebih besar dari j (3), yang membuat kondisi bernilai false, nilai j sebelum di-increment (3) diberikan kepada k. Setelah itu, i dan j di-increment menjadi 3 dan 4.

16.

```

Untitled - 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);
    }
}

```

```

D:\Tugas PBO>javac Oprator.java

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

```

Penjelasan: Kode Java Oprator ini memberikan gambaran tentang berbagai operasi yang dapat dilakukan pada tipe data dasar dalam Java, seperti boolean, integer, dan float. Program dimulai dengan mendeklarasikan variabel-variabel dengan tipe-tipe data tersebut. Kemudian, serangkaian operasi dilakukan, termasuk operasi boolean seperti AND, OR, NOT, dan XOR, operasi numerik pada integer dan float seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulo, serta operasi relasional numerik seperti perbandingan kesetaraan, ketidaksamaan, kurang dari, lebih dari, kurang dari atau sama dengan, dan lebih dari atau sama dengan. Namun, kode ini tidak menampilkan hasil dari operasi-operasi tersebut ke konsol. Untuk melihat hasil dari setiap operasi, perintah `System.out.println()` perlu ditambahkan setelah setiap operasi.