

LAPORAN

PEMOGRAMAN ORIENTASI OBJEK



Nama : Muh Ilham Nur Hidayat Akbar

Nim :13020220041

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

MAKASSAR 2023

Kode Program 1 :

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

Penjelasan Program :

Program di atas adalah program sederhana dalam bahasa pemrograman Java yang bertujuan untuk menunjukkan penggunaan tipe data float dan double serta operasi dasar dalam Java.

Berikut penjelasan singkat mengenai kode program di atas:

1. Program dimulai dengan mendefinisikan kelas `Asgdll`.
2. Di dalam kelas tersebut, terdapat method `main` yang merupakan method utama yang akan dieksekusi pertama kali ketika program dijalankan.
3. Di dalam method `main`, terdapat deklarasi variabel `f` dengan tipe data float dan diinisialisasi dengan nilai 20.0.
4. Selanjutnya, terdapat deklarasi variabel `fl` dengan tipe data double tanpa inisialisasi nilai.
5. Kemudian dilakukan operasi pengisian nilai variabel `fl` dengan nilai 10.0f. Perlu diperhatikan bahwa walaupun `fl` memiliki tipe data double, namun nilai yang diberikan adalah float (10.0f). Hal ini sah dalam Java karena float secara otomatis akan di-promote menjadi double ketika diperlukan.
6. Selanjutnya, dilakukan pencetakan (print) variabel `f` dan `fl` dengan menggunakan `System.out.println()`.
7. Program kemudian mencetak nilai `f` dan `fl` ke layar.

Tujuan dari program ini sepertinya adalah untuk menunjukkan bahwa dalam Java, tipe data float dan double dapat digunakan untuk menyimpan bilangan desimal dengan presisi yang berbeda. Selain itu, program juga menunjukkan bahwa dalam operasi di Java, tipe data float dapat di-promote menjadi double jika diperlukan.

Kode Program 2 :

```
public class Asign {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
  
        int i;  
  
        System.out.print ("hello\n"); i = 7;  
        System.out.println ("Ini nilai i : " + i);  
    }  
}
```

Penjelasan Program :

Program di atas adalah contoh sederhana dari sebuah program Java yang mencetak pesan ke layar. Berikut adalah penjelasan kode program tersebut:

1. Deklarasi Kelas: Program dimulai dengan deklarasi kelas `Asign`. Nama kelas harus sesuai dengan nama file yang menyimpan kode program (dalam hal ini, Asign.java).
2. Metode Utama: Metode utama `main` adalah titik masuk utama untuk program Java. Semua eksekusi program dimulai dari sini.
3. Komentar: Komentar dalam kode digunakan untuk memberikan penjelasan tentang bagaimana kode bekerja atau apa yang dilakukan oleh bagian-bagian tertentu dari kode tersebut. Komentar ini tidak akan dieksekusi oleh program dan hanya digunakan untuk tujuan dokumentasi.
4. Deklarasi Variabel: Variabel `i` bertipe data integer (`int`) dideklarasikan di dalam metode `main`.
5. Mencetak Pesan: Program mencetak pesan "hello" ke layar menggunakan metode `System.out.print()`. Pesan tersebut diakhiri dengan karakter newline (`\n`), sehingga pesan berikutnya akan dicetak di baris baru.
6. Memberi Nilai pada Variabel: Nilai 5 diberikan pada variabel `i` setelah mencetak pesan.
7. Mencetak Nilai Variabel: Program mencetak nilai variabel `i` dengan menggunakan metode `System.out.println()`. Pesan yang dicetak adalah "Ini nilai i : " diikuti dengan nilai dari variabel `i`.

Tujuan dari program diatas adalah untuk mengilustrasikan konsep dasar dalam pemrograman Java, termasuk deklarasi variabel, pencetakan pesan ke layar, dan penggunaan metode `main` sebagai titik masuk program. Program ini juga menunjukkan cara mendeklarasikan variabel, memberikan nilai pada variabel, dan mencetak nilai variabel ke layar.

Kode Program 3:

```
public class ASIGNi {
```

```

/**
 * @param args
 */
public static void main(String[] args) {

    short ks = 1;
    int ki = 1;
    long kl = 10000;
    char c = 65;

    char c1 = 'Z';
    double x = 50.2f;
    float y = 50.2f;

    System.out.println("Karakter = " + c);
    System.out.println("Karakter = " + c1);

    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);
}
}

```

Penjelasan Program :

Kode program di atas merupakan program Java sederhana yang mendemonstrasikan penggunaan beberapa tipe data dasar dalam bahasa pemrograman Java. Tujuan utama dari program ini adalah untuk menunjukkan cara mendeklarasikan dan menggunakan variabel dengan tipe data yang berbeda.

1. Deklarasi variabel:

- `short ks = 1;`: Mendeklarasikan variabel `ks` dengan tipe data `short` dan memberikan nilai awal 1.
- `int ki = 1;`: Mendeklarasikan variabel `ki` dengan tipe data `int` dan memberikan nilai awal 1.
- `long kl = 10000;`: Mendeklarasikan variabel `kl` dengan tipe data `long` dan memberikan nilai awal 10000.
- `char c = 65;`: Mendeklarasikan variabel `c` dengan tipe data `char` dan memberikan nilai awal 65, yang merupakan representasi karakter 'A' dalam ASCII.
- `char c1 = 'Z';`: Mendeklarasikan variabel `c1` dengan tipe data `char` dan memberikan nilai awal 'Z'.

- `double x = 50.2f;`: Mendeklarasikan variabel `x` dengan tipe data `double` dan memberikan nilai awal 50.2f (sebuah nilai floating-point).
- `float y = 50.2f;`: Mendeklarasikan variabel `y` dengan tipe data `float` dan memberikan nilai awal 50.2f.

2. Menampilkan output ke layar:

- `System.out.println("Karakter = " + c);`: Menampilkan nilai dari variabel `c` ke layar dengan tambahan teks "Karakter = ".
- `System.out.println("Karakter = " + c1);`: Menampilkan nilai dari variabel `c1` ke layar dengan tambahan teks "Karakter = ".
- `System.out.println("Bilangan integer (short) = " + ks);`: Menampilkan nilai dari variabel `ks` ke layar dengan tambahan teks "Bilangan integer (short) = ".
- `System.out.println("\t(int) = " + ki);`: Menampilkan nilai dari variabel `ki` ke layar dengan tambahan teks "(int) = ".
- `System.out.println("\t(long)= " + kl);`: Menampilkan nilai dari variabel `kl` ke layar dengan tambahan teks "(long) = ".
- `System.out.println("Bilangan Real x = " + x);`: Menampilkan nilai dari variabel `x` ke layar dengan tambahan teks "Bilangan Real x = ".
- `System.out.println("Bilangan Real y = " + y);`: Menampilkan nilai dari variabel `y` ke layar dengan tambahan teks "Bilangan Real y = ".

Tujuan program ini adalah untuk memberikan contoh penggunaan berbagai tipe data dasar dalam Java dan menunjukkan bagaimana nilai variabel dapat ditampilkan ke layar. Program ini dapat membantu pemula dalam memahami dasar-dasar deklarasi variabel dan penggunaan tipe data dalam Java.

Kode Program 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: ");
        masukan = new Scanner(System.in);
        a = masukan.nextInt(); /* coba ketik :
        masukan.nextInt();
```

Apa akibatnya ?*/

```
System.out.print ("Nilai yang dibaca : "+ a);  
    }  
}
```

Penjelasan Program :

Kode program di atas adalah contoh sederhana untuk membaca input integer dari pengguna menggunakan kelas Scanner dalam bahasa pemrograman Java. Berikut penjelasan singkat mengenai setiap bagian dari kode tersebut:

1. `import java.util.Scanner;`: Baris ini mengimpor kelas Scanner dari pustaka util di dalam paket Java. Kelas Scanner digunakan untuk membaca input dari pengguna.
2. `public class BacaData {`: Ini adalah deklarasi kelas dengan nama "BacaData". Semua kode program akan terdapat di dalam tubuh kelas ini.
3. `public static void main(String[] args) {`: Ini adalah metode utama (main method) yang akan dieksekusi ketika program dijalankan. Semua pernyataan program akan dijalankan di dalam metode ini.
4. `int a;`: Mendeklarasikan variabel `a` yang akan digunakan untuk menyimpan nilai integer yang dibaca dari pengguna.
5. `Scanner masukan;`: Mendeklarasikan objek Scanner dengan nama `masukan`, yang akan digunakan untuk membaca input dari pengguna.
6. `System.out.print ("Contoh membaca dan menulis, ketik nilai integer: ");`: Menampilkan pesan permintaan kepada pengguna untuk memasukkan nilai integer.
7. `masukan = new Scanner(System.in);`: Menciptakan objek Scanner yang terhubung dengan input dari sistem (`System.in`), sehingga dapat digunakan untuk membaca nilai yang dimasukkan oleh pengguna.
8. `a = masukan.nextInt();`: Menggunakan metode `nextInt()` dari objek Scanner untuk membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `a`.
9. `System.out.print ("Nilai yang dibaca : "+ a);`: Menampilkan nilai yang telah dibaca dari pengguna.

Program ini bertujuan untuk memberikan contoh penggunaan kelas Scanner untuk membaca nilai integer dari pengguna dan menampilkan nilai tersebut. Input integer yang dimasukkan oleh pengguna akan disimpan dalam variabel `a`, dan kemudian nilai tersebut akan ditampilkan kembali ke pengguna. Program ini dirancang untuk memberikan pemahaman dasar tentang penggunaan kelas Scanner dalam membaca input dari pengguna dalam bentuk integer.

Kode Program 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");

/* Tambahkan program membaca/input data
menggunakan Class Scanner, Class Console dan Class
JOptionPane */
}
}

```

Penjelasan Program :

Program di atas adalah program sederhana dalam bahasa pemrograman Java yang digunakan untuk membaca satu karakter dan satu bilangan dari pengguna. Berikut adalah penjelasan singkat mengenai kode program tersebut:

1. `import` statement:

- `import java.io.BufferedReader;`: Mengimpor kelas `BufferedReader` dari paket `java.io`, yang digunakan untuk membaca input dari pengguna.
- `import java.io.IOException;`: Mengimpor kelas `IOException` dari paket `java.io`, yang diperlukan karena metode `readLine()` dapat menimbulkan `IOException`.
- `import java.io.InputStreamReader;`: Mengimpor kelas `InputStreamReader` dari paket `java.io`, yang digunakan untuk membaca input stream.
- Komentar: Ada juga komentar yang menggunakan `//` untuk memberikan penjelasan di bagian ini.

2. Deklarasi kelas `Bacakar`:

- Deklarasi kelas dimulai dengan `public class Bacakar {`.
- Di dalamnya terdapat metode `main` yang merupakan titik masuk eksekusi program.

3. Deklarasi variabel dan objek:

- `char cc;`: Variabel untuk menyimpan satu karakter.
- `int bil;`: Variabel untuk menyimpan satu bilangan bulat.
- `InputStreamReader isr = new InputStreamReader(System.in);`: Membuat objek `InputStreamReader` untuk membaca input dari `System.in`.
- `BufferedReader dataIn = new BufferedReader(isr);`: Membuat objek `BufferedReader` untuk membaca data dari `InputStreamReader`.

4. Membaca input:

- `System.out.print("baca 1 karakter : ");`: Menampilkan pesan ke pengguna untuk meminta input karakter.
- `cc = dataIn.readLine().charAt(0);`: Membaca satu baris input dari pengguna, kemudian mengambil karakter pertama dan menyimpannya dalam variabel `cc`.
- `System.out.print("baca 1 bilangan : ");`: Menampilkan pesan ke pengguna untuk meminta input bilangan.
- `bil = Integer.parseInt(dataIn.readLine());`: Membaca satu baris input dari pengguna dan mengonversinya menjadi bilangan bulat, lalu menyimpannya dalam variabel `bil`.

5. Menampilkan output:

- `System.out.print(cc + "\n" + bil + "\n");`: Menampilkan karakter dan bilangan yang telah dibaca.
- `System.out.print("bye \n");`: Menampilkan pesan "bye".

6. Komentar:

- Terdapat beberapa komentar yang memberikan penjelasan atau instruksi untuk tambahan program menggunakan `Scanner`, `Console`, dan `JOptionPane`.

Tujuan utama program ini adalah untuk membaca satu karakter dan satu bilangan dari pengguna, kemudian menampilkan karakter dan bilangan tersebut sebagai output. Program ini juga memberikan beberapa contoh penggunaan komentar dan beberapa perintah yang di-comment-out

(tidak aktif), seperti penggunaan `JOptionPane`, yang mungkin merupakan eksperimen atau potensi pengembangan program lebih lanjut.

Kode Program 6 :

```
public class Casting1 {  
  
    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  
    }  
}
```

Penjelasan Program :

Program ini adalah contoh penggunaan operator casting dalam bahasa pemrograman Java. Operator casting digunakan untuk mengubah tipe data satu ke tipe data lainnya. Berikut adalah penjelasan setiap baris kode:

1. `int a=5, b=6;`: Mendeklarasikan dua variabel bertipe int, `a` dan `b`, dan memberi nilai masing-masing 5 dan 6.
2. `float d=2.f, e=3.2f;`: Mendeklarasikan dua variabel bertipe float, `d` dan `e`, dan memberi nilai masing-masing 2.f dan 3.2f.
3. `char g='5';`: Mendeklarasikan variabel bertipe char, `g`, dan memberi nilai '5'. Perlu dicatat bahwa karakter '5' di sini disimpan dalam bentuk karakter (char), bukan angka.
4. `double k=3.14;`: Mendeklarasikan variabel bertipe double, `k`, dan memberi nilai 3.14.
5. `System.out.println((float)a);`: Mencetak hasil casting variabel `a` dari int ke float. Hasilnya akan menjadi 5.0f.
6. `System.out.println((double)b);`: Mencetak hasil casting variabel `b` dari int ke double. Hasilnya akan menjadi 6.0.
7. `System.out.println((int)d);`: Mencetak hasil casting variabel `d` dari float ke int. Hasilnya akan menjadi 2.

8. `System.out.println((double)e);`: Mencetak hasil casting variabel `e` dari float ke double. Hasilnya akan menjadi 3.2.

9. `System.out.println((int)g);`: Mencetak hasil casting variabel `g` dari char ke int (ASCII). Karena karakter '5' memiliki nilai ASCII 53, hasilnya akan menjadi 53.

10. `System.out.println((float)g);`: Mencetak hasil casting variabel `g` dari char ke float (ASCII). Karena karakter '5' memiliki nilai ASCII 53, hasilnya akan menjadi 53.0f.

11. `System.out.println((double)g);`: Mencetak hasil casting variabel `g` dari char ke double (ASCII). Karena karakter '5' memiliki nilai ASCII 53, hasilnya akan menjadi 53.0.

12. `System.out.println((int)k);`: Mencetak hasil casting variabel `k` dari double ke int. Hasilnya akan menjadi 3.

13. `System.out.println((float)k);`: Mencetak hasil casting variabel `k` dari double ke float. Hasilnya akan menjadi 3.14f.

Tujuan dari program ini adalah menunjukkan bagaimana operator casting dapat digunakan untuk mengubah tipe data antar variabel dengan tipe data yang berbeda.

Kode Program 7 :

```
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);  
    }  
}
```

```
}  
}
```

Penjelasan Program :

Program ini adalah contoh penggunaan operasi konversi tipe data (casting) di dalam bahasa pemrograman Java. Tujuan utama dari program ini adalah untuk menunjukkan cara melakukan konversi antara tipe data primitif, seperti integer, float, double, char, dan string.

1. `int a=8, b=9;`
 - Mendeklarasikan dua variabel bertipe data integer (`a`` dan `b``) dan memberi nilai awal.
2. `float d=2.f, e=3.2f;`
 - Mendeklarasikan dua variabel bertipe data float (`d`` dan `e``) dan memberi nilai awal.
3. `char g='5';`
 - Mendeklarasikan variabel bertipe data char (`g``) dan memberi nilai awal berupa karakter '5'.
4. `double k=3.14;`
 - Mendeklarasikan variabel bertipe data double (`k``) dan memberi nilai awal.
5. `String n="67", m="45", l="100";`
 - Mendeklarasikan tiga variabel bertipe data String (`n``, `m``, dan `l``) dan memberi nilai awal dalam bentuk string.
6. `a = Integer.parseInt(n);`
 - Mengkonversi string `n`` ke integer dan menyimpannya di variabel `a``.
7. `k = Double.parseDouble(m);`
 - Mengkonversi string `m`` ke double dan menyimpannya di variabel `k``.
8. `d = Float.parseFloat(l);`
 - Mengkonversi string `l`` ke float dan menyimpannya di variabel `d``.
9. `n = String.valueOf(b);`
 - Mengkonversi nilai integer `b`` ke string dan menyimpannya di variabel `n``.
10. `m = String.valueOf(g);`
 - Mengkonversi karakter `g`` ke string dan menyimpannya di variabel `m``.
11. `l = String.valueOf(e);`
 - Mengkonversi nilai float `e`` ke string dan menyimpannya di variabel `l``.
12. `k = Double.valueOf(a).intValue();`
 - Mengkonversi nilai integer `a`` ke double, kemudian mengambil nilai integer-nya dan menyimpannya di variabel `k``.
13. `double c = Integer.valueOf(b).doubleValue();`
 - Mengkonversi nilai integer `b`` ke double dan menyimpannya di variabel `c``.

14. `System.out.println("a : "+a+"\nk : "+k+"\nd : "+d);``
- Menampilkan hasil konversi dalam bentuk string untuk variabel ``a``, ``k``, dan ``d``.
15. `System.out.println("n : "+n+"\nm : "+m+"\nl : "+l);``
- Menampilkan hasil konversi dalam bentuk string untuk variabel ``n``, ``m``, dan ``l``.
16. `System.out.println("k : "+k+"\nc : "+c+"\nl : "+l);``
- Menampilkan hasil konversi dalam bentuk string untuk variabel ``k``, ``c``, dan ``l``.

Program ini bertujuan untuk memberikan pemahaman tentang penggunaan berbagai metode konversi tipe data di Java.

Kode Program 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*/

    }
}
```

Penjelasan Program :

Program di atas adalah contoh penggunaan operator kondisional (ternary operator) dalam bahasa pemrograman Java. Tujuan dari program ini adalah untuk menunjukkan bagaimana cara menggunakan operator kondisional untuk melakukan pengecekan kondisi dan mengembalikan nilai berdasarkan kondisi tersebut.

Berikut adalah penjelasan singkat mengenai kode program tersebut:

1. Deklarasi variabel: Variabel `x` dan `y` dideklarasikan dengan tipe data `int` dan diberi nilai masing-masing `1` dan `2`.

2. Pemakaian operator kondisional: Pada baris `System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y));`, terjadi pemakaian operator kondisional. Operator kondisional memiliki bentuk `(kondisi) ? nilai_jika_true : nilai_jika_false`. Artinya, jika kondisi dalam kurung `(x < y)` terpenuhi (true), maka nilai yang dikembalikan adalah nilai `x`, jika tidak terpenuhi (false), maka nilai yang dikembalikan adalah nilai `y`.

3. Output: Program mencetak nilai dari variabel `x` dan `y`, kemudian mencetak hasil dari ekspresi `(x < y) ? x : y`. Hasil ekspresi ini adalah nilai dari variabel `x` jika kondisi `(x < y)` terpenuhi, dan nilai dari variabel `y` jika kondisi tersebut tidak terpenuhi.

Jadi, tujuan dari program ini adalah untuk mengilustrasikan cara menggunakan operator kondisional untuk mengembalikan nilai berdasarkan kondisi yang diberikan. Dalam kasus ini, program mencetak nilai variabel `x` jika `x < y`, dan mencetak nilai variabel `y` jika tidak.

Kode Program 9 :

```
/* pembagian integer, casting */
public class Ekspresi1 {
    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);
    }
}
```

Penjelasan Program :

Program di atas adalah contoh kode dalam bahasa pemrograman Java yang bertujuan untuk mendemonstrasikan pembagian integer dan casting dalam pemrograman.

Tujuan dari program ini adalah untuk memperlihatkan perbedaan hasil pembagian bilangan integer dan bilangan float, serta penggunaan casting untuk mengubah tipe data bilangan integer menjadi bilangan float.

1. Deklarasi variabel: Program pertama-tama mendeklarasikan variabel x dan y sebagai bilangan integer, serta variabel fx dan fy sebagai bilangan float.
2. Operasi pembagian integer: Program mencetak hasil dari pembagian x oleh y dalam format integer dan float. Namun, karena x dan y adalah bilangan integer, hasil pembagiannya juga akan menjadi bilangan integer.
3. Menggunakan variabel float: Untuk mendapatkan hasil pembagian dalam format float, nilai x dan y ditugaskan ke variabel float fx dan fy. Hal ini dilakukan untuk mengubah tipe data bilangan integer menjadi float sebelum melakukan pembagian.
4. Operasi casting: Casting dilakukan dengan cara mengubah tipe data variabel x dan y menjadi float saat melakukan pembagian. Ini dilakukan dengan menempatkan tipe data yang diinginkan di depan variabel yang akan di-cast.
5. Percobaan pembagian baru: Setelah melakukan casting, program mencoba lagi untuk membagi nilai x dan y dengan nilai yang baru. Ini menunjukkan bahwa dengan pembagian bilangan integer, hasilnya akan dibulatkan ke bawah, sedangkan pembagian dengan bilangan float memberikan hasil yang lebih presisi.

Kode Program 10 :

```
public class PrintHello {  
    /**  
     * @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");  
    }  
}
```

Penjelasan Program :

Program diatas merupakan contoh sederhana dalam bahasa pemrograman Java. Tujuan dari program ini adalah untuk menunjukkan deklarasi dan inisialisasi variabel, serta bagaimana cara menggunakan `System.out.println()` untuk mencetak nilai variabel ke konsol.

Mari kita jelaskan bagian-bagian dari program tersebut:

1. ****Deklarasi Kelas (`public class Asgdll { ... }`)**:**
 - Program dimulai dengan deklarasi sebuah kelas bernama `Asgdll`.
2. ****Method `main`**:**
 - `public static void main(String[] args)` adalah method utama yang akan dieksekusi ketika program dijalankan. Method ini menerima argumen dalam bentuk array dari string (`args`), tetapi dalam program ini argumen tersebut tidak digunakan.
3. ****Deklarasi Variabel (`float f= 20.0f; double fl;`)**:**
 - Variabel `f` dideklarasikan sebagai float dan diinisialisasi dengan nilai 20.0.
 - Variabel `fl` dideklarasikan sebagai double tanpa inisialisasi nilai, oleh karena itu nilainya akan menjadi default.
4. ****Algoritma (`fl=10.0f; System.out.println ("f : "+f+ "\nf11: "+fl);`)**:**
 - Nilai variabel `fl` diubah menjadi 10.0.
 - Kemudian, menggunakan `System.out.println()`, program mencetak nilai dari variabel `f` dan `fl` ke konsol.
 - Pada baris ini, terjadi kesalahan penulisan, yaitu seharusnya `f11` menjadi `fl`.

Dengan demikian, tujuan program ini adalah untuk mendemonstrasikan konsep dasar dalam penggunaan variabel, inisialisasi, dan pencetakan output di Java.

Kode Program 11 :

```
/* Efek dari operator ++ */
public class Incr {
    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);
    }
}
```

Penjelasan Program :

Program ini bertujuan untuk mengilustrasikan efek dari operator ++ pada bahasa pemrograman Java. Operator ++ digunakan untuk menambahkan nilai variabel dengan 1. Dalam program ini, terdapat dua variabel, yaitu `i` dan `j`.

Berikut adalah penjelasan langkah-langkah program:

1. `i = 3;`: Variabel `i` diinisialisasi dengan nilai 3.
2. `j = i++;`: Nilai dari `i` disalin ke dalam `j`, kemudian nilai `i` ditambahkan dengan 1. Oleh karena itu, nilai `j` akan menjadi 3 (nilai awal `i` sebelum penambahan).
3. `System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j);`: Menampilkan nilai `i` dan `j` ke layar. Bagian `++i` mengakibatkan nilai `i` ditambahkan dengan 1 sebelum dicetak. Sehingga nilai `i` yang ditampilkan adalah nilai setelah penambahan. Nilai `j` tetap 3 karena telah diambil sebelum penambahan.

Sebagai contoh, jika nilai awal `i` adalah 3, maka hasil keluaran program akan menjadi:

```
...  
Nilai i : 5  
Nilai j : 3  
...
```

Hal ini karena nilai `i` yang pertama kali ditampilkan sudah ditambahkan dengan 1 (`++i`), sedangkan nilai `j` masih menyimpan nilai `i` sebelum penambahan.

Kode Program 12 :

```
public class Oper1 {  
    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 */  
    }  
}
```

Penjelasan Program :

Program ini adalah contoh sederhana dari penggunaan operator bitwise di dalam bahasa pemrograman Java. Tujuannya adalah untuk menunjukkan bagaimana operator-operator bitwise bekerja pada bilangan bulat.

Berikut adalah penjelasan singkat setiap bagian dari program:

1. ****Deklarasi Variabel:****

- `int n = 10;`: Mendeklarasikan variabel `n` dan menginisialisasinya dengan nilai 10 (dalam representasi biner: 1010).
- `int x = 1;`: Mendeklarasikan variabel `x` dan menginisialisasinya dengan nilai 1 (dalam representasi biner: 1).
- `int y = 2;`: Mendeklarasikan variabel `y` dan menginisialisasinya dengan nilai 2 (dalam representasi biner: 10).

2. ****Pemrosesan dan Output:****

- `System.out.println("n = " + n);`: Mencetak nilai `n`.
- `System.out.println("x = " + x);`: Mencetak nilai `x`.
- `System.out.println("y = " + y);`: Mencetak nilai `y`.
- `System.out.println("n & 8 = " + (n & 8));`: Melakukan operasi bitwise AND antara `n` dan 8, lalu mencetak hasilnya.
- `System.out.println("x & ~8 = " + (x & ~8));`: Melakukan operasi bitwise AND antara `x` dan komplemen dari 8, lalu mencetak hasilnya.
- `System.out.println("y << 2 = " + (y << 2));`: Melakukan operasi bitwise shift kiri pada `y` sebanyak 2 bit, lalu mencetak hasilnya.
- `System.out.println("y >> 3 = " + (y >> 3));`: Melakukan operasi bitwise shift kanan pada `y` sebanyak 3 bit, lalu mencetak hasilnya.

****Penjelasan Operasi Bitwise:****

- `&` (AND): Melakukan operasi bitwise AND antara dua bilangan.
- `~` (NOT): Melakukan operasi bitwise NOT (negasi) pada bilangan.
- `<<` (Left Shift): Menggeser bit ke kiri sebanyak `n` posisi.
- `>>` (Right Shift): Menggeser bit ke kanan sebanyak `n` posisi.

Program ini memberikan pemahaman praktis tentang bagaimana operator bitwise bekerja pada level bit. Hasil output dari operasi-operasi tersebut akan memberikan insight tentang manipulasi bit yang dilakukan.

Kode Program 13 :

```
public class Oper2 {
    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 */
}
}

```

Penjelasan Program :

Program di atas adalah contoh program Java sederhana yang bertujuan untuk menunjukkan penggunaan operator bitwise dan operasi matematika dasar di Java.

Penjelasan singkat kode program:

1. ``char i, j;``: Mendeklarasikan dua variabel bertipe data char, yaitu i dan j.
2. ``i = 3;``: Menginisialisasi variabel i dengan nilai 3. Dalam representasi biner, 3 adalah 00000011.
3. ``j = 4;``: Menginisialisasi variabel j dengan nilai 4. Dalam representasi biner, 4 adalah 00000100.
4. Melakukan operasi-operasi bitwise:
 - ``System.out.println("i & j = "+ (i & j));``: Melakukan operasi bitwise AND antara i dan j. Hasilnya adalah 0 (00000000 dalam biner).
 - ``System.out.println("i | j = "+ (i | j));``: Melakukan operasi bitwise OR antara i dan j. Hasilnya adalah 7 (00000111 dalam biner).
 - ``System.out.println("i ^ j = "+ (i ^ j));``: Melakukan operasi bitwise XOR antara i dan j. Hasilnya adalah 7 (00000111 dalam biner).
5. ``System.out.println(Math.pow(i, j));``: Mencetak hasil dari pemangkatan i pangkat j menggunakan fungsi ``Math.pow()``. Namun, perlu diperhatikan bahwa ini mungkin tidak memberikan hasil yang diharapkan karena i dan j bertipe char, yang seharusnya bertipe numerik.
6. ``System.out.println("~i = "+ ~i);``: Melakukan operasi bitwise NOT terhadap i. Hasilnya adalah -4 dalam representasi biner 11111100.

Tujuan program ini adalah untuk memberikan pemahaman tentang penggunaan operator bitwise dan operasi matematika dasar dalam bahasa pemrograman Java.

Kode Program 14 :

```

public class Oper3 {

    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 */
    }
}

```

Penjelasan Program :

Program di atas adalah sebuah program Java sederhana yang menggunakan operator logika dan bitwise. Berikut adalah penjelasan singkat untuk setiap bagian dari program:

1. **if (true && true)**: Ini adalah pernyataan kondisional `if` yang menggunakan operator logika `&&` (AND). Kondisi ini selalu bernilai `true` karena keduanya adalah `true`. Pernyataan `System.out.println(true && true);` akan dicetak ke layar.
2. **if (true & true)**: Ini juga adalah pernyataan kondisional `if` dengan operator bitwise `&` (AND bitwise). Meskipun hasilnya juga `true`, namun perlu dicatat bahwa operasi bitwise akan selalu mengevaluasi kedua operand meskipun hasilnya sudah dapat diprediksi. Namun, dalam pernyataan `System.out.println(true & false);`, hasil yang dicetak adalah `false`, bukan `true` seperti pada kondisi. Ini mungkin kesalahan penulisan yang perlu diperhatikan.
3. **if (true)**: Ini adalah pernyataan `if` yang selalu bernilai `true`, dan pernyataan di dalamnya `System.out.println(true);` akan selalu dieksekusi dan mencetak `true`.
4. **if (true || true)**: Pernyataan kondisional `if` ini menggunakan operator logika `||` (OR). Kondisi ini juga selalu bernilai `true`, dan pernyataan `System.out.println(true);` akan dieksekusi dan mencetak `true`.
5. **if (true|false)**: Ini adalah pernyataan kondisional `if` dengan operator bitwise `|` (OR bitwise). Hasilnya adalah `true` dan pernyataan `System.out.println(true|false);` akan mencetak `true`.

Tujuan dari program ini sepertinya adalah untuk memberikan contoh penggunaan operator logika (`&&`, `||`) dan bitwise (`&`, `|`) pada bahasa pemrograman Java, serta memberikan pemahaman dasar tentang evaluasi kondisi dan operasi bitwise. Namun, perlu diingat bahwa ada kesalahan pada pernyataan `System.out.println(true & false);` yang mungkin perlu diperbaiki agar sesuai dengan tujuan awalnya.

Kode Program 15 :

```
/* Operator ternary */
public class Oper4 {

    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);
    }
}
```

Penjelasan Program :

Program di atas adalah contoh penggunaan operator ternary (atau conditional) dalam bahasa pemrograman Java. Operator ternary digunakan untuk mengevaluasi ekspresi boolean, dan berbeda dengan if-else statement yang lebih umum.

Tujuan program ini adalah untuk mengilustrasikan penggunaan operator ternary dalam Java dan bagaimana itu berperilaku dalam beberapa situasi.

Berikut adalah penjelasan singkat kode programnya:

1. Mendeklarasikan variabel:

- `int i = 0;`: Mendeklarasikan variabel `i` dan menginisialisasinya dengan nilai 0.
- `int j = 0;`: Mendeklarasikan variabel `j` dan menginisialisasinya dengan nilai 0.
- `char c = 8; char d = 10;`: Mendeklarasikan dua variabel bertipe char, `c` dan `d`, dan menginisialisasinya dengan nilai 8 dan 10.
- `int e = (((int)c > (int)d) ? c : d);`: Menggunakan operator ternary untuk membandingkan nilai-nilai `c` dan `d` (yang dikonversi menjadi integer) dan menginisialisasi nilai `e` dengan nilai dari `c` atau `d` tergantung pada hasil perbandingan.
- `int k = ((i > j) ? i : j);`: Menggunakan operator ternary untuk membandingkan nilai-nilai `i` dan `j`, dan menginisialisasi nilai `k` dengan nilai dari `i` atau `j` tergantung pada hasil perbandingan.

2. Menampilkan nilai `e` dan `k` ke layar.

3. Melakukan operasi `i++` dan `j++` dan mengubah nilai `k` dengan menggunakan operator ternary lagi.

Hasil output dari program ini akan menunjukkan nilai `e` yang merupakan hasil dari perbandingan antara `c` dan `d`, nilai `k` yang merupakan hasil perbandingan antara `i` dan `j` sebelum operasi `i++` dan `j++`, dan nilai `k` lagi setelah operasi `i++` dan `j++`.

Kode Program 16 :

```
/* Contoh pengoperasian variabel bertipe dasar */
public class Oprator {

    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 program di bawah ini untuk menampilkan output program");

        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);
}
}

```

Penjelasan Program :

Program di atas adalah contoh penggunaan operator-operator dasar dalam pemrograman Java. Tujuan dari program ini adalah untuk memberikan pemahaman tentang penggunaan operator-operator dasar dalam pemrograman Java, baik itu operator untuk tipe data boolean maupun tipe data numerik.

Berikut adalah penjelasan singkat mengenai tujuan dan fungsi dari masing-masing bagian kode program:

1. ****Deklarasi Variabel****: Program pertama-tama mendeklarasikan beberapa variabel yang akan digunakan dalam program. Variabel-variabel tersebut termasuk tipe data boolean (`Bool1`, `Bool2`, `TF`) dan tipe data numerik (`i`, `j`, `hsl`, `x`, `y`, `res`).
2. ****Operasi pada Tipe Data Boolean****: Program melakukan operasi-operasi dasar pada tipe data boolean, seperti AND, OR, NOT, dan XOR. Ini bertujuan untuk menunjukkan bagaimana operator-operasi tersebut bekerja dan bagaimana hasilnya disimpan dalam variabel boolean.
3. ****Operasi pada Tipe Data Numerik****: Selanjutnya, program melakukan berbagai operasi matematika dasar pada tipe data numerik (int dan float), seperti penjumlahan, pengurangan, perkalian, pembagian, dan operasi modulo. Hal ini membantu dalam memahami bagaimana operator-operasi matematika tersebut berperilaku dalam Java.
4. ****Operasi Relasional****: Program juga mencakup operasi relasional, seperti kesetaraan (`==`), ketidaksamaan (`!=`), lebih besar dari (`>`), lebih kecil dari (`<`), lebih besar dari atau sama dengan (`>=`), dan lebih kecil dari atau sama dengan (`<=`). Ini berguna untuk membandingkan nilai antara variabel-variabel numerik.
5. ****Output Program****: Program juga mencetak pesan untuk membaca teks dan menambahkan perintah program di bawahnya untuk menampilkan output program. Namun, implementasi perintah output tersebut tidak ada dalam kode yang diberikan.

Tujuan dari program ini adalah untuk memberikan pemahaman yang lebih baik tentang cara menggunakan operator-operator dasar dalam pemrograman Java dan bagaimana mereka berinteraksi dengan berbagai tipe data.