

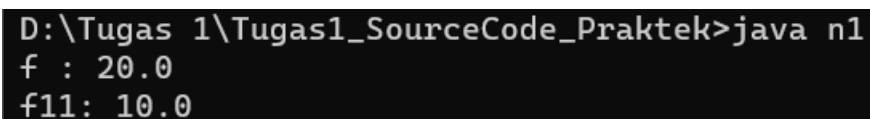
Nama : Adri Linrung  
Nim : 13020220145  
Kelas : A4

1. Kode program

//13020220145,Adri Linrung, Kamis 29 Feb 2024

```
public class n1 {  
    public static void main(String[] args) {  
        /* Kamus */  
        float f = 20.0f;  
        double fll;  
        /* Algoritma */  
        fll = 10.0f;  
        System.out.println("f : " + f + "\nf11: " + fll);  
    }  
}
```

Output :



```
D:\Tugas 1\Tugas1_SourceCode_Praktek>java n1  
f : 20.0  
f11: 10.0
```

Penjelasan setiap bagian dari program:

1. `public class n1`: Mendefinisikan sebuah kelas Java dengan nama T1. Kelas ini memiliki satu metode utama yang akan dieksekusi saat program dijalankan.
2. `public static void main(String[] args)`: Metode utama program Java. Pada Java, setiap program harus memiliki satu metode `main` yang akan dieksekusi terlebih dahulu saat program dijalankan. Metode ini menerima argumen berupa array string.
3. `float f = 20.0f`;; Mendeklarasikan variabel `f` dengan tipe data float dan menginisialisasinya dengan nilai 20.0. Suffix `f` menandakan bahwa nilai tersebut adalah float.
4. `double fll`;; Mendeklarasikan variabel `fll` dengan tipe data double tanpa inisialisasi.
5. `fll = 10.0f`;; Menginisialisasi variabel `fll` dengan nilai 10.0. Karena `fll` adalah tipe data double, maka nilai float akan secara implisit di-cast menjadi double saat di-assign ke `fll`.
6. `System.out.println("f : " + f + "\nf11: " + fll)`;; Mencetak nilai variabel `f` dan `fll` ke layar. Ini dilakukan dengan menggunakan `System.out.println`, yang merupakan metode untuk mencetak keluaran ke konsol. Ekspresi `"f : " + f + "\nf11: " + fll` adalah string yang terdiri dari teks konstan dan nilai variabel yang

dimasukkan menggunakan operator +. Saat menjalankan program, nilai dari variabel f dan fll akan digabungkan dengan teks dan dicetak ke konsol.

## 2. Kode program

//13020220145,Adri Linrung, Kamis 29 Feb 2024

```
public class n2 {  
    public static void main(String[] args) {  
        /* Kamus */  
        int i;  
        /* Program */  
        System.out.print("hello\n");  
        i = 5;  
        System.out.println("Ini nilai i : " + i);  
    }  
}
```

Output :

```
D:\Tugas 1\Tugas1_SourceCode_Praktek>java n2  
hello  
Ini nilai i : 5
```

Penjelasan setiap bagian dari program:

1. `public class T2`: Mendefinisikan sebuah kelas Java dengan nama T2.
2. `public static void main(String[] args)`: Metode utama program Java. Pada Java, setiap program harus memiliki satu metode main yang akan dieksekusi terlebih dahulu saat program dijalankan. Metode ini menerima argumen berupa array string.
3. `int i`;; Mendeklarasikan variabel i dengan tipe data int. Variabel i digunakan untuk menyimpan bilangan bulat.
4. `System.out.print("hello\n");`: Mencetak teks "hello" ke layar tanpa ganti baris (dengan menggunakan `System.out.print`). Teks "hello" diakhiri dengan karakter baris baru (`\n`) untuk membuat output berikutnya dimulai dari baris baru.
5. `i = 5`;; Menginisialisasi variabel i dengan nilai 5.
6. `System.out.println("Ini nilai i : " + i)`;; Mencetak teks "Ini nilai i : " diikuti dengan nilai dari variabel i. Pada Java, kita dapat menggunakan operator + untuk menggabungkan teks dan nilai variabel dalam satu pernyataan cetak. Ekspresi ini akan mencetak nilai dari variabel i setelah teks "Ini nilai i : " ke layar, diikuti dengan ganti baris karena menggunakan `println`.

## 3. Kode program

```
public class n3 {
```

```

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

    /* KAMUS */
    short ks = 1;
    int ki = 1;
    long kl = 10000L; // Perhatikan penambahan "L" pada inisialisasi long
    char c = 65; /* inisialisasi karakter dengan integer */
    char c1 = 'Z'; /* inisialisasi karakter dengan karakter */
    double x = 50.2;
    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 = " + (int) c); // Casting c ke int
    System.out.println("Karakter = " + (int) c1); // Casting c1 ke int
    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);
}
}

```

Output :

```

D:\Tugas 1\Tugas1_SourceCode_Praaktek>java n3
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

```

Penjelasan setiap bagian dari program:

1. public class T3: Mendefinisikan sebuah kelas Java dengan nama T3.
2. public static void main(String[] args): Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter args adalah

array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.

3. `short ks = 1;` Mendeklarasikan variabel `ks` dengan tipe data `short` dan menginisialisasinya dengan nilai 1.

4. `int ki = 1;` Mendeklarasikan variabel `ki` dengan tipe data `int` dan menginisialisasinya dengan nilai 1.

5. `long kl = 10000L;` Mendeklarasikan variabel `kl` dengan tipe data `long` dan menginisialisasinya dengan nilai 10000. Suffix "`L`" menandakan bahwa nilai tersebut adalah `long`.

6. `char c = 65;` Mendeklarasikan variabel `c` dengan tipe data `char` dan menginisialisasinya dengan nilai 65. Dalam Unicode, 65 mewakili karakter '`A`'.

7. `char c1 = 'Z';` Mendeklarasikan variabel `c1` dengan tipe data `char` dan menginisialisasinya dengan karakter '`Z`'.

8. `double x = 50.2;` Mendeklarasikan variabel `x` dengan tipe data `double` dan menginisialisasinya dengan nilai 50.2.

9. `float y = 50.2f;` Mendeklarasikan variabel `y` dengan tipe data `float` dan menginisialisasinya dengan nilai 50.2. Suffix "`f`" menandakan bahwa nilai tersebut adalah `float`.

10. Pada bagian Algoritma, nilai-nilai dari variabel-variabel tersebut dicetak ke layar dengan menggunakan `System.out.println()`. Ini adalah pernyataan cetak yang digunakan untuk menampilkan teks dan nilai variabel ke konsol. Beberapa nilai variabel juga dicetak setelah melakukan casting ke tipe data yang sesuai menggunakan operator (`tipeData`) di depan variabel.

#### 4. Kode program

//13020220145,Adri Linrung, Kamis 29 Feb 2024

```
import java.util.Scanner;
```

```
public class n4{
```

```
    public static void main(String[] args) {
```

```
        // 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(); // Membaca input dari pengguna
```

```

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

        // Menutup scanner setelah digunakan
        masukan.close();
    }
}

```

Output :

```

D:\Tugas 1\Tugas1_SourceCode_Praktek>java n4
Contoh membaca dan menulis, ketik nilai integer:
145
Nilai yang dibaca : 145

```

Penjelasan setiap bagian dari program:

1. `import java.util.Scanner;`: Mengimpor kelas Scanner dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class T4 {`: Mendefinisikan sebuah kelas Java dengan nama T4.
3. `public static void main(String[] args) {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter `args` adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
4. `int a;`: Mendeklarasikan variabel `a` dengan tipe data `int`. Variabel ini akan digunakan untuk menyimpan nilai yang akan dimasukkan oleh pengguna.
5. `Scanner masukan;`: Mendeklarasikan variabel `masukan` dengan tipe `Scanner`. Variabel ini akan digunakan untuk membaca input dari pengguna.
6. `System.out.print("Contoh membaca dan menulis, ketik nilai integer: \n");`: Mencetak teks "Contoh membaca dan menulis, ketik nilai integer:" ke layar. Teks ini memberikan petunjuk kepada pengguna untuk memasukkan nilai integer.
7. `masukan = new Scanner(System.in);`: Membuat objek dari kelas `Scanner` dengan menggunakan konstruktor `Scanner(System.in)`. Objek ini akan digunakan untuk membaca input dari sistem masukan standar (biasanya keyboard).
8. `a = masukan.nextInt();`: Memanggil metode `nextInt()` dari objek `masukan` untuk membaca nilai integer yang dimasukkan oleh pengguna, dan menyimpannya ke dalam variabel `a`.
9. `System.out.print("Nilai yang dibaca : " + a);`: Mencetak nilai yang dibaca dari input pengguna ke layar.

10. masukan.close();: Menutup objek Scanner setelah selesai digunakan. Ini adalah praktik yang baik untuk membebaskan sumber daya yang digunakan oleh objek Scanner.

#### 5. Kode program

```
//13020220145,Adri Linrung, Kamis 29 Feb 2024
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class n5 {
    public static void main(String[] args) throws IOException {
        // 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 */
    }
}
```

Output :

```
D:\Tugas 1\Tugas1_SourceCode_Praktek>java n5
hello
baca 1 karakter : A
baca 1 bilangan : 145
A
145
bye
```

Penjelasan setiap bagian dari program:

1. `import java.io.BufferedReader;;` Mendatangkan kelas `BufferedReader` dari paket `java.io`. Kelas ini digunakan untuk membaca input karakter dari sumber data seperti `System.in`.
2. `import java.io.IOException;;` Mendatangkan kelas `IOException` dari paket `java.io`. Kelas ini digunakan untuk menangani pengecualian yang mungkin terjadi saat bekerja dengan operasi masukan/keluaran (I/O).
3. `import java.io.InputStreamReader;;` Mendatangkan kelas `InputStreamReader` dari paket `java.io`. Kelas ini digunakan untuk membaca karakter dari aliran input byte.
4. `public class T5 {`: Mendefinisikan sebuah kelas Java dengan nama `T5`.
5. `public static void main(String[] args) throws IOException {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter `args` adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan. Pernyataan `throws IOException` menandakan bahwa metode ini dapat menimbulkan pengecualian `IOException` yang perlu ditangani.
6. Deklarasi variabel `cc` dan `bil` yang akan digunakan untuk menyimpan karakter dan bilangan yang dimasukkan oleh pengguna.
7. `InputStreamReader isr = new InputStreamReader(System.in);`: Membuat objek `InputStreamReader` yang digunakan untuk membaca karakter dari aliran input standar (`System.in`).
8. `BufferedReader dataIn = new BufferedReader(isr);`: Membuat objek `BufferedReader` yang akan membungkus objek `InputStreamReader`. Ini memungkinkan kita untuk membaca input secara linier.
9. `BufferedReader dataIn = new BufferedReader(new InputStreamReader(System.in));`: Cara lain untuk membuat objek `BufferedReader` langsung dari `System.in` tanpa perantara objek `InputStreamReader`.

10. Mencetak teks "hello" dan meminta pengguna untuk memasukkan satu karakter dan satu bilangan.

11. `cc = dataIn.readLine().charAt(0);` Membaca satu baris teks dari pengguna menggunakan objek `dataIn`, kemudian mengambil karakter pertama dari baris tersebut dan menyimpannya ke dalam variabel `cc`.

12. `bil = Integer.parseInt(dataIn.readLine());` Membaca satu baris teks dari pengguna menggunakan objek `dataIn`, kemudian mengonversi teks tersebut menjadi bilangan bulat menggunakan metode `parseInt()` dari kelas `Integer`, dan menyimpannya ke dalam variabel `bil`.

#### 6. Kode program

```
//13020220145,Adri Linrung, Kamis 29 Feb
2024
/*Casting menggunakan tipe data primitif*/
public class n6 {
    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
    }
}
```

Output :

```
D:\Tugas 1\Tugas1_SourceCode_Praktek>java n6
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14
```



Penjelasan setiap bagian dari program:

1. `public class T6 {`: Mendefinisikan sebuah kelas Java dengan nama T6.
2. `public static void main(String[] args) {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter `args` adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. Deklarasi variabel `a, b, d, e, g,` dan `k` dengan tipe data `int, float, char,` dan `double`.
4. `System.out.println((float) a);`: Mencetak nilai variabel `a` setelah dilakukan casting ke tipe data `float`. Operasi casting dilakukan dengan menempatkan (tipeData) di depan nilai yang ingin di-casting.
5. `System.out.println((double) b);`: Mencetak nilai variabel `b` setelah dilakukan casting ke tipe data `double`.
6. `System.out.println((int) d);`: Mencetak nilai variabel `d` setelah dilakukan casting ke tipe data `int`.
7. `System.out.println((double) e);`: Mencetak nilai variabel `e` setelah dilakukan casting ke tipe data `double`.
8. `System.out.println((int) g);`: Mencetak nilai variabel `g` setelah dilakukan casting ke tipe data `int`. Nilai karakter `g` diubah menjadi nilai ASCII-nya.
9. `System.out.println((float) g);`: Mencetak nilai variabel `g` setelah dilakukan casting ke tipe data `float`. Nilai karakter `g` diubah menjadi nilai ASCII-nya.
10. `System.out.println((double) g);`: Mencetak nilai variabel `g` setelah dilakukan casting ke tipe data `double`. Nilai karakter `g` diubah menjadi nilai ASCII-nya.
11. `System.out.println((int) k);`: Mencetak nilai variabel `k` setelah dilakukan casting ke tipe data `int`.
12. `System.out.println((float) k);`: Mencetak nilai variabel `k` setelah dilakukan casting ke tipe data `float`.

7. Kode program

```
//13020220145,Adri Linrung, Kamis 29 Feb 2024
/*Casting menggunakan tipe data Class*/
public class n7 {
    /**
     * @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);
}
}

```

Output :

```

D:\Tugas 1\Tugas1_SourceCode_Praktek>java n7
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 setiap bagian dari program:

1. public class T7 {: Mendefinisikan sebuah kelas Java dengan nama T7.
2. public static void main(String[] args) {: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter args adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.

3. Deklarasi variabel-variabel a, b, d, e, g, k, n, m, dan l dengan tipe data yang sesuai.

4. `a = Integer.parseInt(n);` Mengonversi nilai dari string n menjadi tipe data integer menggunakan metode `parseInt()` dari kelas Integer, kemudian disimpan kembali ke variabel a.

5. `k = Double.parseDouble(m);` Mengonversi nilai dari string m menjadi tipe data double menggunakan metode `parseDouble()` dari kelas Double, kemudian disimpan kembali ke variabel k.

6. `d = Float.parseFloat(l);` Mengonversi nilai dari string l menjadi tipe data float menggunakan metode `parseFloat()` dari kelas Float, kemudian disimpan kembali ke variabel d.

7. `n = String.valueOf(b);` Mengonversi nilai dari variabel b menjadi string menggunakan metode `valueOf()` dari kelas String, kemudian disimpan kembali ke variabel n.

8. `m = String.valueOf(g);` Mengonversi nilai dari variabel g menjadi string menggunakan metode `valueOf()` dari kelas String, kemudian disimpan kembali ke variabel m.

9. `l = String.valueOf(e);` Mengonversi nilai dari variabel e menjadi string menggunakan metode `valueOf()` dari kelas String, kemudian disimpan kembali ke variabel l.

10. `k = Double.valueOf(a).intValue();` Mengonversi nilai dari variabel a menjadi double menggunakan metode `valueOf()` dari kelas Double, kemudian dilakukan konversi menjadi integer menggunakan metode `intValue()`, dan hasilnya disimpan kembali ke variabel k.

11. `double c = Integer.valueOf(b).doubleValue();` Mengonversi nilai dari variabel b menjadi integer menggunakan metode `valueOf()` dari kelas Integer, kemudian dilakukan konversi menjadi double menggunakan metode `doubleValue()`, dan hasilnya disimpan ke dalam variabel c.

#### 8. Kode program

```
//13020220145,Adri Linrung, Kamis
29 Feb 2024
/* pemakaian operator kondisional */
public class n8 {
    /**
     * @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*/
}
}

```

Output :

```

D:\Tugas 1\Tugas1_SourceCode_Praktek>javac n8.java

D:\Tugas 1\Tugas1_SourceCode_Praktek>java n8
x = 1
y = 2
hasil ekspresi = (x < y) ? x : y = 1

```

Penjelasan setiap bagian dari program:

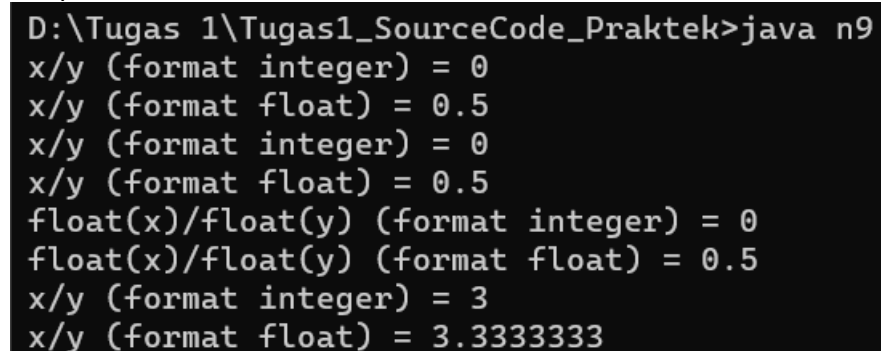
1. `public class T8 {`: Mendefinisikan sebuah kelas Java dengan nama T8.
2. `public static void main(String[] args) {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter args adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. `int x = 1;`: Mendeklarasikan variabel x dengan tipe data integer dan memberinya nilai awal 1.
4. `int y = 2;`: Mendeklarasikan variabel y dengan tipe data integer dan memberinya nilai awal 2.
5. `System.out.print("x = " + x + "\n");`: Mencetak nilai variabel x ke layar dengan format tertentu.
6. `System.out.print("y = " + y + "\n");`: Mencetak nilai variabel y ke layar dengan format tertentu.
7. `System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y));`: Mencetak hasil dari ekspresi ternary `(x < y) ? x : y`. Ekspresi ini akan menghasilkan nilai x jika kondisi `x < y` benar, dan menghasilkan nilai y jika kondisi tersebut salah.
  - `(x < y)` merupakan kondisi.
  - `?` merupakan operator ternary yang menunjukkan bahwa kondisi yang akan diuji.
  - `x` adalah nilai yang akan dihasilkan jika kondisi benar.
  - `:` menandakan bahwa nilai berikutnya akan dihasilkan jika kondisi salah.
  - `y` adalah nilai yang akan dihasilkan jika kondisi salah.

Hasil ekspresi tersebut kemudian dicetak ke layar bersama dengan pesan tambahan.

#### 9. Kode program

```
//13020220145,Adri Linrung, Kamis 29 Feb 2024
/* pembagian integer, casting */
public class n9 {
    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);
    }
}
```

Output :



```
D:\Tugas 1\Tugas1_SourceCode_Praaktek>java n9
x/y (format integer) = 0
x/y (format float) = 0.5
x/y (format integer) = 0
x/y (format float) = 0.5
float(x)/float(y) (format integer) = 0
float(x)/float(y) (format float) = 0.5
x/y (format integer) = 3
x/y (format float) = 3.333333
```

Penjelasan setiap bagian dari program:

1. public class T9 {: Mendefinisikan sebuah kelas Java dengan nama T9.

2. `public static void main(String[] args) {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter `args` adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. `int x = 1;`: Mendeklarasikan variabel `x` dengan tipe data integer dan memberinya nilai awal 1.
4. `int y = 2;`: Mendeklarasikan variabel `y` dengan tipe data integer dan memberinya nilai awal 2.
5. `float fx;`: Mendeklarasikan variabel `fx` dengan tipe data float.
6. `float fy;`: Mendeklarasikan variabel `fy` dengan tipe data float.
7. `System.out.print("x/y (format integer) = " + x / y);`: Mencetak hasil dari operasi pembagian `x / y` dalam format integer ke layar.
8. `System.out.print("\nx/y (format float) = " + x / y);`: Mencetak hasil dari operasi pembagian `x / y` dalam format float ke layar. Karena `x` dan `y` bertipe data integer, maka operasi ini akan menghasilkan nilai integer.
9. `fx = x;`: Melakukan penugasan nilai dari `x` ke `fx`.
10. `fy = y;`: Melakukan penugasan nilai dari `y` ke `fy`.
11. `System.out.print("\nx/y (format integer) = " + fx / fy);`: Mencetak hasil dari operasi pembagian `fx / fy` dalam format integer ke layar. Karena `fx` dan `fy` bertipe data float, maka operasi ini akan menghasilkan nilai float.
12. `System.out.print("\nx/y (format float) = " + fx / fy);`: Mencetak hasil dari operasi pembagian `fx / fy` dalam format float ke layar.
13. `System.out.print("\nfloat(x)/float(y) (format integer)= " + (float) x / (float) y);`: Mencetak hasil dari operasi pembagian `(float) x / (float) y` dalam format integer ke layar. Operasi casting dilakukan sebelum pembagian untuk memastikan bahwa operasi pembagian dilakukan dengan tipe data float.
14. `System.out.print("\nfloat(x)/float(y) (format float) = " + (float) x / (float) y);`: Mencetak hasil dari operasi pembagian `(float) x / (float) y` dalam format float ke layar.
15. `x = 10;`: Mengubah nilai variabel `x` menjadi 10.
16. `y = 3;`: Mengubah nilai variabel `y` menjadi 3.

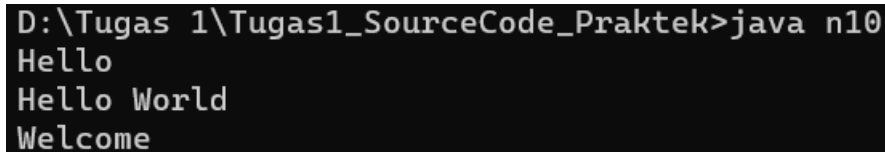
17. `System.out.print("\nx/y (format integer) = " + x / y);`; Mencetak hasil dari operasi pembagian `x / y` dalam format integer ke layar.

18. `System.out.print("\nx/y (format float) = " + x / y);`; Mencetak hasil dari operasi pembagian `x / y` dalam format float ke layar.

10. Kode program

//13020220145,Adri Linrung, Kamis 29 Feb 2024

```
public class n10 {  
    /**  
     * @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 1\Tugas1_SourceCode_Praktek>java n10  
Hello  
Hello World  
Welcome
```

Output :

Penjelasan setiap bagian dari program:

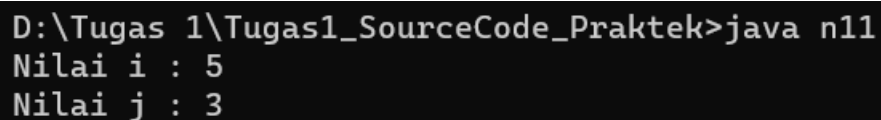
1. `public class T10 {`: Mendefinisikan sebuah kelas Java dengan nama T10.
2. `public static void main(String[] args) {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter `args` adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. `System.out.print("Hello");`: Menggunakan method `print()` dari kelas `System.out` untuk menampilkan teks "Hello" ke layar tanpa mengganti baris.
4. `System.out.print("\nHello ");`: Menggunakan method `print()` dari kelas `System.out` untuk menampilkan teks "Hello" ke layar dan mengganti baris.
5. `System.out.println("World");`: Menggunakan method `println()` dari kelas `System.out` untuk menampilkan teks "World" ke layar dan mengganti baris.

6. `System.out.println("Welcome");`;; Menggunakan method `println()` dari kelas `System.out` untuk menampilkan teks "Welcome" ke layar dan mengganti baris.

#### 11. Kode program

```
//13020220145,Adri  
Linrung, Kamis 29 Feb 2024  
/* Efek dari operator ++ */  
public class n11 {  
    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);  
    }  
}
```

Output :



```
D:\Tugas 1\Tugas1_SourceCode_Praktek>java n11  
Nilai i : 5  
Nilai j : 3
```

Penjelasan setiap bagian dari program:

1. `public class T11 {`: Mendefinisikan sebuah kelas Java dengan nama T11.
2. `public static void main(String[] args) {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter `args` adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. `int i, j;`: Deklarasi variabel `i` dan `j` dengan tipe data integer.
4. `i = 3;`: Memberi nilai awal 3 pada variabel `i`.
5. `j = i++;`: Memberikan nilai variabel `i` ke `j`, kemudian nilai variabel `i` ditambah 1.
6. `System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j);`: Mencetak nilai variabel `i` setelah penambahan 1 (`++i`) dan nilai variabel `j` ke layar. Perhatikan bahwa (`++i`) digunakan untuk menampilkan nilai `i` setelah penambahan 1. Sedangkan `j` adalah nilai `i` sebelum penambahan 1.

#### 12. Kode program

```
//13020220145,Adri Linrung, Kamis 29 Feb 2024  
/* pemakaian beberapa operator terhadap bit */
```

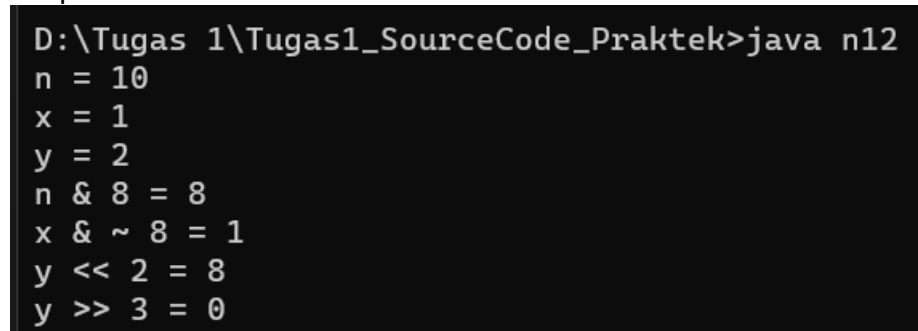


```

public class n12 {
    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==> 100 = 4 */
        System.out.println("y >> 3 = " + (y >> 3)); /* 10==>0000 = 0 */
    }
}

```

Output :



```

D:\Tugas 1\Tugas1_SourceCode_Praktek>java n12
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 4
y >> 3 = 0

```

Penjelasan setiap bagian dari program:

1. `public class T12 {`: Mendefinisikan sebuah kelas Java dengan nama T12.
2. `public static void main(String[] args) {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter args adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. `int n = 10; /* 1010 */`: Mendeklarasikan variabel n dengan tipe data integer dan memberinya nilai 10 dalam representasi biner (1010).
4. `int x = 1; /* 1 */`: Mendeklarasikan variabel x dengan tipe data integer dan memberinya nilai 1 dalam representasi biner (1).
5. `int y = 2; /* 10 */`: Mendeklarasikan variabel y dengan tipe data integer dan memberinya nilai 2 dalam representasi biner (10).
6. `System.out.println("n = " + n);`: Menampilkan nilai variabel n ke layar.

7. `System.out.println("x = " + x);`; Menampilkan nilai variabel x ke layar.
8. `System.out.println("y = " + y);`; Menampilkan nilai variabel y ke layar.
9. `System.out.println("n & 8 = " + (n & 8)); /* 1010 AND 1000 */`; Menampilkan hasil operasi bitwise AND antara nilai n dan 8 ke layar.
10. `System.out.println("x & ~8 = " + (x & ~8)); /* 1 AND 0111 */`; Menampilkan hasil operasi bitwise AND antara nilai x dan hasil complement dari 8 ke layar.
11. `System.out.println("y << 2 = " + (y << 2)); /* 10==> 1000 = 8 */`; Menampilkan hasil operasi bitwise shift left (<<) pada nilai y sebanyak 2 bit ke layar.
12. `System.out.println("y >> 3 = " + (y >> 3)); /* 10 ==> 0000 = 0 */`; Menampilkan hasil operasi bitwise shift right (>>) pada nilai y sebanyak 3 bit ke layar.

### 13. Kode program

```
//13020220145,Adri Linrung, Kamis 29 Feb 2024
/* pemakaian beberapa operator terhadap RELATIONAL DAN bit */
public class n13 {
    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 = " + (int) 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 */
    }
}
```

Output :

```

D:\Tugas 1\Tugas1_SourceCode_Praktek>java n13
i = 3
j =
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4

```

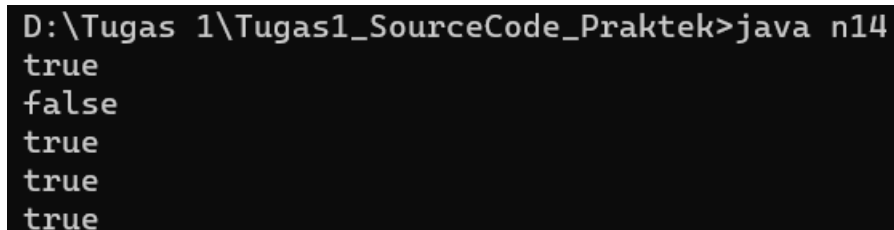
Penjelasan setiap bagian dari program:

1. `public class T13 {`: Mendefinisikan sebuah kelas Java dengan nama T13.
2. `public static void main(String[] args) {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter `args` adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. `char i, j;`: Mendeklarasikan dua variabel `i` dan `j` dengan tipe data `char`.
4. `i = 3; /* 00000011 dalam biner */`: Memberikan nilai 3 ke variabel `i` dalam representasi biner (00000011).
5. `j = 4; /* 00000100 dalam biner */`: Memberikan nilai 4 ke variabel `j` dalam representasi biner (00000100).
6. `System.out.println("i = " + (int) i);`: Menampilkan nilai variabel `i` dalam format integer ke layar.
7. `System.out.println("j = " + (int) j);`: Menampilkan nilai variabel `j` dalam format integer ke layar.
8. `System.out.println("i & j = " + (i & j));`: Menampilkan hasil operasi bitwise AND antara nilai `i` dan `j` ke layar.
9. `System.out.println("i | j = " + (i | j));`: Menampilkan hasil operasi bitwise OR antara nilai `i` dan `j` ke layar.
10. `System.out.println("i ^ j = " + (i ^ j));`: Menampilkan hasil operasi bitwise XOR antara nilai `i` dan `j` ke layar.
11. `System.out.println(Math.pow(i, j));`: Menampilkan hasil pangkat dari nilai `i` dan `j` ke layar. Namun, perlu diperhatikan bahwa operator `^` dalam bahasa Java bukanlah operator pangkat, melainkan operator XOR. Untuk melakukan operasi pangkat, digunakan method `Math.pow()` dari kelas `Math`.
12. `System.out.println("~i = " + ~i);`: Menampilkan hasil operasi bitwise NOT dari nilai `i` ke layar.

#### 14. Kode program

//13020220145,Adri Linrung, Kamis 29 Feb 2024

```
public class n14 {  
    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 1\Tugas1_SourceCode_Praktek>java n14  
true  
false  
true  
true  
true
```

Output :

Penjelasan setiap bagian dari program:

1. `public class T14 {`: Mendefinisikan sebuah kelas Java dengan nama T14.
2. `public static void main(String[] args) {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter `args` adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. `if (true && true) { System.out.println(true && true); }`: Statement if untuk mengecek apakah kedua kondisi `true && true` benar. Jika benar, maka akan mencetak hasil `true && true` ke layar.
4. `if (true & true) { System.out.println(true & false); }`: Statement if untuk mengecek apakah kedua kondisi `true & true` benar. Operator `&` merupakan

operator bitwise AND, sehingga akan melakukan operasi bitwise AND terhadap nilai boolean true dan false. Hasilnya adalah false, dan hasil tersebut akan dicetak ke layar.

5. `if (true) { System.out.println(true); }`: Statement if untuk mengecek apakah kondisi true benar. Jika benar, maka akan mencetak true ke layar.

6. `if (true || true) { System.out.println(true); }`: Statement if untuk mengecek apakah salah satu dari kedua kondisi `true || true` benar. Karena salah satu kondisi benar, maka hasilnya adalah true, dan hasil tersebut akan dicetak ke layar.

7. `if (true | false) { System.out.println(true | false); }`: Statement if untuk mengecek apakah salah satu dari kedua kondisi `true | false` benar. Operator `|` merupakan operator bitwise OR, sehingga akan melakukan operasi bitwise OR terhadap nilai boolean true dan false. Hasilnya adalah true, dan hasil tersebut akan dicetak ke layar.

#### 15. Kode program

```
//13020220145,Adri Linrung, Kamis 29 Feb 2024
/* Operator terner */
public class n15 {
    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);
    }
}
```

Output :

```
D:\Tugas 1\Tugas1_SourceCode_Praktek>java n15
Nilai e = 10
Nilai k = 0
Nilai k = 4
```

Penjelasan setiap bagian dari program:

1. `public class T15 {`: Mendefinisikan sebuah kelas Java dengan nama T15.
  2. `public static void main(String[] args) {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter args adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
  3. `int i = 0; int j = 0;`: Mendeklarasikan dua variabel bertipe integer i dan j, dan menginisialisasikan keduanya dengan nilai 0.
  4. `char c = 8; char d = 10;`: Mendeklarasikan dua variabel bertipe karakter c dan d, dan menginisialisasikan keduanya dengan nilai 8 dan 10 secara berurutan.
  5. `int e = (((int) c > (int) d) ? c : d);`: Menggunakan operator ternary untuk menentukan nilai variabel e. Jika nilai c lebih besar dari nilai d, maka nilai e akan sama dengan nilai c, jika tidak, maka nilai e akan sama dengan nilai d.
  6. `int k = ((i > j) ? i : j);`: Menggunakan operator ternary untuk menentukan nilai variabel k. Jika nilai i lebih besar dari nilai j, maka nilai k akan sama dengan nilai i, jika tidak, maka nilai k akan sama dengan nilai j.
  7. `System.out.print("Nilai e = " + e);`: Mencetak nilai variabel e ke layar.
  8. `System.out.print("\nNilai k = " + k);`: Mencetak nilai variabel k ke layar.
  9. `i = 2; j = 3; k = ((i++ > j++) ? i : j);`: Mengubah nilai variabel i menjadi 2 dan j menjadi 3. Kemudian menggunakan operator ternary untuk menentukan nilai variabel k. Pada operasi ini, nilai i dan j akan diinkremen (bertambah satu) setelah perbandingan dilakukan. Jika nilai i sebelum diinkremen lebih besar dari nilai j sebelum diinkremen, maka nilai k akan sama dengan nilai i, jika tidak, maka nilai k akan sama dengan nilai j.
16. Kode program
- ```
//13020220145,Adri Linrung, Kamis 29 Feb 2024
/* Contoh pengoperasian variabel bertipe dasar */
public class n16 {
    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);
}
}

```

Output :

```

D:\Sahabat Umi\SMT 4\Pbo\Program\Tugas 1>java T16
Silahkan baca teksnya dan tambahkan perintah program di bawah
ini untuk menampilkan output program

```

Penjelasan setiap bagian dari program:

1. public class T16 { : Mendefinisikan sebuah kelas Java dengan nama T16.

2. `public static void main(String[] args) {`: Metode utama program Java. Metode ini akan dieksekusi terlebih dahulu saat program dijalankan. Parameter `args` adalah array dari string yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.

3. Mendeklarasikan beberapa variabel bertipe data dasar:

- `boolean Bool1, Bool2, TF;`: Mendeklarasikan tiga variabel bertipe boolean.
- `int i, j, hsl;`: Mendeklarasikan tiga variabel bertipe integer.
- `float x, y, res;`: Mendeklarasikan tiga variabel bertipe float.

4. Menggunakan operasi-operasi dasar:

- `TF = Bool1 && Bool2;`: Operasi boolean AND antara Bool1 dan Bool2.
- `TF = Bool1 || Bool2;`: Operasi boolean OR antara Bool1 dan Bool2.
- `TF = !Bool1;`: Operasi boolean NOT dari Bool1.
- `TF = Bool1 ^ Bool2;`: Operasi boolean XOR antara Bool1 dan Bool2.
- Operasi numerik seperti penjumlahan, pengurangan, perkalian, dan pembagian serta modulus menggunakan variabel `i` dan `j`.
- Operasi relasional numerik seperti `==`, `!=`, `<`, `>`, `<=`, dan `>=` yang menghasilkan nilai boolean dan menyimpannya dalam variabel `TF`.

5. `System.out.println("Silahkan baca teksnya dan tambahkan perintah program di bawah ini untuk menampilkan output program");`: Menampilkan pesan untuk menambahkan perintah program di bawahnya.