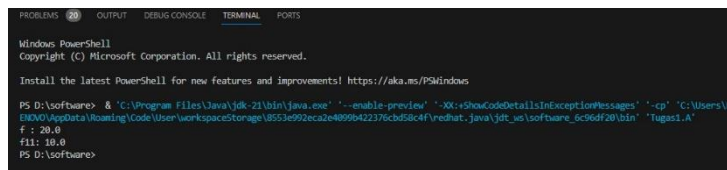


Nama: Muh. Faizal

Nim: 13020220149

Kelas: A4

```
public class A {  
    /  
    @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);  
    }  
}
```



```
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-enable-preview' '-XX:ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\I  
ENGO\AppData\Local\Roaming\Code\User\workspaceStorage\8551e992eca2e4099b422376c8d5d4f\redhat-java\jdk_as\software_6c96df20\bin' 'Tugas1.A'  
f : 20.0  
f11: 10.0  
PS D:\software>
```

Ini adalah pernyataan yang menetapkan kelas `A` ke dalam paket `Tugas1`. Paket adalah cara untuk mengelompokkan kelas-kelas terkait dalam sebuah proyek Java.

1. **public class A {** : Ini adalah deklarasi kelas Java dengan nama **A**. Nama kelas harus sama dengan nama file Java di mana kelas ini

didefinisikan.

2. **public static void main(String[] args) {** : Metode **main** adalah metode utama yang akan dieksekusi ketika program dimulai. Setiap program Java harus memiliki metode **main** sebagai titik masuknya.
3. **float f= 20.0f;** : Mendeklarasikan variabel **f** sebagai tipe data float dan menginisialisasinya dengan nilai 20.0.
4. **double f11;** : Mendeklarasikan variabel **f11** sebagai tipe data double tanpa menginisialisasinya. Ini berarti variabel **f11** memiliki nilai default, yaitu 0.0.
5. **f11=10.0f;** : Menginisialisasi variabel **f11** dengan nilai 10.0. Karena tipe data double lebih presisi dari float, tidak perlu menambahkan suffix **f** pada nilai 10.0.
6. **System.out.println ("f : "+f + "\nf11: "+f11);** : Mencetak nilai variabel **f** dan **f11** ke konsol. Operator **+** digunakan untuk menggabungkan teks dengan nilai variabel.

```

public class B {
    /
    @param args
    /

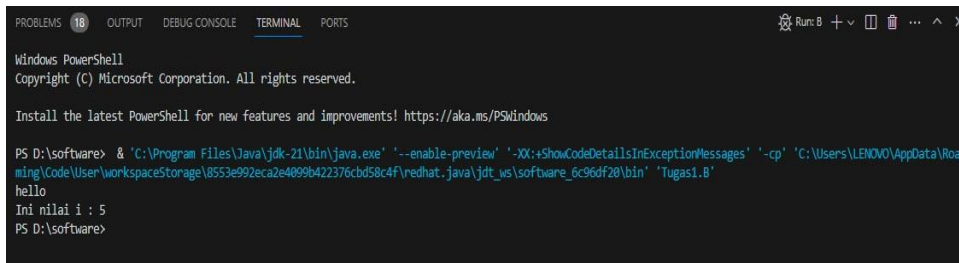
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        / Kamus
        / int
        i;

        / Program /

        System.out.print ("hello\n"); i
        = 5; System.out.println ("Ini
        nilai i : " + i);
    }
}

```



```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\LENOVO\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.B'
hello
Ini nilai i : 5
PS D:\software>

```

1. **public class B {** : Mendeklarasikan kelas Java dengan nama **B**. Nama kelas harus sama dengan nama file Java di mana kelas ini didefinisikan.
2. **public static void main(String[] args) {** : Metode **main** adalah metode utama yang akan dieksekusi ketika program dimulai. **args** adalah parameter yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. **System.out.print ("hello\n");** : Menggunakan **System.out.print** untuk mencetak teks "hello" ke konsol tanpa membuat baris baru. Karakter **\n** digunakan untuk membuat baris baru setelah teks "hello" tercetak.

4. **int i;** : Mendeklarasikan variabel **i** sebagai tipe data integer tanpa menginisialisasinya. Variabel **i** memiliki nilai default 0.
5. **i = 5;** : Menginisialisasi variabel **i** dengan nilai 5.
6. **System.out.println ("Ini nilai i : " + i);** : Menggunakan **System.out.println** untuk mencetak teks "Ini nilai i :" diikuti dengan nilai variabel **i** ke konsol. Operator **+** digunakan untuk menggabungkan teks dengan nilai variabel.

Pernyataan ini mencetak nilai variabel `i` ke konsol dengan menggunakan `System.out.println()`. Sehingga nilai `i` akan ditampilkan setelah string "Ini nilai i : " dengan menggunakan operator konkatenasi (`+`)

```

public class C {
    /
    @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);
}
}

```

```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L
ENDVO\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.C'
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
PS D:\software>

```

1. `short ks = 1; int ki = 1; long kl = 10000;` : Mendeklarasikan dan menginisialisasi variabel `ks` sebagai tipe data `short` dengan nilai 1, `ki` sebagai tipe data `int` dengan nilai 1, dan `kl` sebagai tipe data `long` dengan nilai 10000.
2. `char c = 65;` : Mendeklarasikan variabel `c` sebagai tipe data `char` dan menginisiasiasinya dengan nilai 65, yang merupakan nilai ASCII untuk karakter 'A'.
3. `char c1 = 'Z';` : Mendeklarasikan variabel `c1` sebagai tipe data `char` dan menginisiasiasinya dengan karakter 'Z'.
4. `double x = 50.2f; float y = 50.2f;` : Mendeklarasikan variabel `x` sebagai tipe data `double` dan `y` sebagai tipe data `float`, kemudian menginisiasiasinya dengan nilai 50.2f.
5. `System.out.println("Karakter = "+ c);` : Mencetak nilai variabel `c` sebagai karakter ke konsol.
6. `System.out.println("Karakter = "+ c1);` : Mencetak nilai variabel `c1` sebagai karakter ke konsol.
7. `System.out.println("Bilangan integer (short) = "+ ks);` : Mencetak nilai variabel `ks` sebagai bilangan integer ke konsol.
8. `System.out.println("\t(int) = "+ ki);` : Mencetak nilai variabel `ki` sebagai bilangan integer ke konsol.
9. `System.out.println("\t(long)= "+ kl);` : Mencetak nilai variabel `kl` sebagai bilangan long ke konsol.
10. `System.out.println("Bilangan Real x = "+ x);` : Mencetak nilai

variabel x sebagai bilangan real ke konsol.

11. *System.out.println("Bilangan Real y = "+ y); : Mencetak nilai*

variabel y sebagai bilangan real ke konsol.

Kode tersebut bertujuan untuk menunjukkan cara mendeklarasikan, menginisialisasi, dan menggunakan beberapa tipe data primitif dalam pemrograman Java, serta cara mencetak nilai variabel ke konsol

```

import
java.util.Scanner
r; public class
BacaData {
/ contoh membaca integer menggunakan Class Scanner/
/
@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);
}
}

```

```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L
ENOV0\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.BacaData'

Contoh membaca dan menulis, ketik nilai integer:
5
Nilai yang dibaca : 5
PS D:\software>

```


1. **import java.util.Scanner;** : Mendeklarasikan impor untuk kelas **Scanner** dari paket **java.util**. Ini diperlukan untuk menggunakan kelas **Scanner** dalam kode Anda.
2. **public class BacaData {** : Mendeklarasikan kelas Java dengan nama **BacaData**. Nama kelas harus sama dengan nama file Java di mana kelas ini didefinisikan.
3. **public static void main(String[] args) {** : Metode **main** adalah metode utama yang akan dieksekusi ketika program dimulai. **args** adalah parameter yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
4. **int a;** : Mendeklarasikan variabel **a** sebagai tipe data integer untuk menyimpan nilai yang akan dibaca dari input pengguna.
5. **Scanner masukan;** : Mendeklarasikan variabel **masukan** sebagai objek dari kelas **Scanner** untuk membaca input dari pengguna.
6. **System.out.print ("Contoh membaca dan menulis, ketik nilai integer: \n");** : Mencetak teks ke konsol untuk meminta pengguna untuk memasukkan nilai integer.
7. **masukan = new Scanner(System.in);** : Membuat objek **Scanner** baru dengan menggunakan **System.in** sebagai sumber input, yang akan membaca input dari keyboard.
8. **a = masukan.nextInt();** : Membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel **a**.
9. **/ coba ketik : masukan.nextInt(); Apa akibatnya ?/** : Ini adalah komentar yang memberikan instruksi kepada pengguna untuk mencoba mengetik **masukan.nextInt();** dan meminta penjelasan tentang apa yang akan terjadi. Namun, karena ini adalah komentar, perintah ini tidak akan dieksekusi oleh program.
10. **System.out.print ("Nilai yang dibaca : "+ a);** : Mencetak nilai yang telah dibaca dari input pengguna ke konsol.

Kode tersebut bertujuan untuk menunjukkan cara menggunakan kelas **Scanner** untuk membaca input dari pengguna dalam bentuk bilangan bulat (integer) dan mencetak nilai yang dibaca ke konsol.

```

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 /

}

}

```

```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L
ENOV0\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.Bacakar'

hello
baca 1 karakter : a
baca 1 bilangan : 3
a
3
bye
PS D:\software>

```

1. **import java.io.BufferedReader; import java.io.IOException; import java.io.InputStreamReader;** : Mendeklarasikan impor untuk beberapa kelas yang diperlukan untuk membaca input dari pengguna. Ini termasuk kelas **BufferedReader** untuk membaca input karakter, **IOException** untuk menangani eksepsi input/output, dan **InputStreamReader** untuk membaca input stream.
2. **public class Bacakar {** : Mendeklarasikan kelas Java dengan nama **Bacakar**. Nama kelas harus sama dengan nama file Java di mana kelas ini didefinisikan.
3. **public static void main(String[] args) throws IOException {** : Metode **main** adalah metode utama yang akan dieksekusi ketika program dimulai. **args** adalah parameter yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan. Karena kita menggunakan **BufferedReader** yang mungkin melemparkan **IOException**, kita menggunakan **throws IOException** untuk menangani kemungkinan eksepsi.
4. **char cc; int bil;** : Mendeklarasikan variabel **cc** sebagai tipe data karakter dan **bil** sebagai tipe data integer untuk menyimpan nilai yang akan dibaca dari input pengguna.
5. **InputStreamReader isr = new InputStreamReader(System.in);**
BufferedReader dataIn = new BufferedReader(isr); : Membuat objek **InputStreamReader** untuk membaca input stream dari **System.in**, dan objek **BufferedReader** untuk membaca input karakter dari **InputStreamReader**.
6. **BufferedReader dataIn = new BufferedReader(new**
InputStreamReader(System.in)); : Contoh lain untuk membuat objek

BufferedReader tanpa menggunakan variabel **isr**.

7. **System.out.print ("hello\n");** : Mencetak teks "hello" ke konsol.
8. **cc =dataIn.readLine().charAt(0);** : Membaca satu baris input dari pengguna menggunakan **BufferedReader**, kemudian mengambil karakter pertama dari baris tersebut dan menyimpannya dalam variabel **cc**.
9. **bil =Integer.parseInt(dataIn.readLine());** : Membaca satu baris input dari pengguna menggunakan **BufferedReader**, kemudian mengonversi string tersebut menjadi integer dan menyimpannya dalam variabel **bil**.
10. Komentar yang dimulai dengan **//** adalah komentar satu baris, sedangkan komentar yang dimulai dengan **/*** dan diakhiri dengan ***/** adalah komentar blok yang dapat mencakup beberapa baris.

Kode tersebut bertujuan untuk menunjukkan cara menggunakan kelas **BufferedReader** untuk membaca input dari pengguna melalui konsol dalam bentuk karakter dan bilangan bulat, serta cara mencetak nilai yang dibaca ke konsol.

```

public class Casting1 {
    /Casting menggunakan tipe data
    primitif/ 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
    }
}

```

```
PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L
ENOV\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.Casting1'

5.0
6.0
2
3.200000047683716
53
53.0
53.0
3
3.14
PS D:\software>
```

Penjelasan

```
public class Casting1 {
```

Ini adalah deklarasi kelas `Casting1`, yang dinyatakan sebagai kelas publik.

Ini berarti kelas `Casting1` dapat diakses dari luar paket di mana kelas tersebut berada.

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

```
    // Isi method main
}
```

Ini adalah metode utama (`main`) yang merupakan titik awal eksekusi untuk program Java. Setiap program Java harus memiliki satu metode `main` yang dijalankan saat program tersebut dijalankan. Parameter `args` adalah parameter dari metode `main` yang dapat digunakan untuk menerima argumen dari baris perintah saat menjalankan program.

```
2. int a=5, b=6;
```

```
    float d=2.f, e=3.2f;
```

```
    char g='5';
```

```
    double k=3.14;
```

Ini adalah bagian deklarasi variabel lokal. Variabel-variabel tersebut dideklarasikan dan diinisialisasi dengan tipe data yang sesuai.

```
3. 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
```

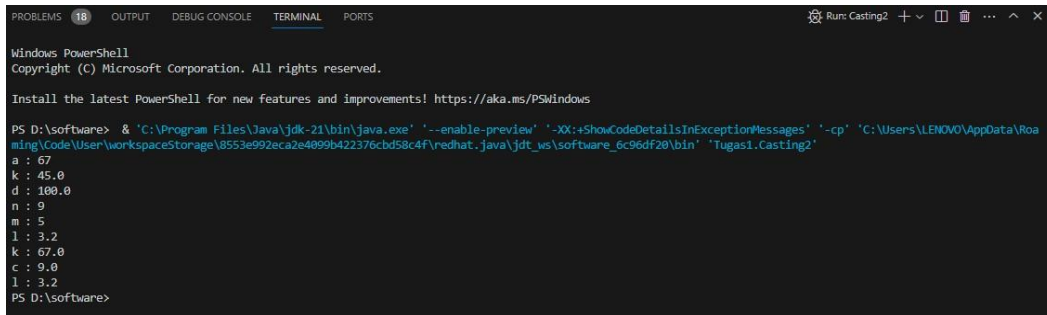
```
<-- int System.out.println((float)k); // double <--  
float
```

Ini adalah bagian dari program yang menunjukkan operasi casting antar tipe data primitif. Secara umum, casting digunakan untuk mengubah tipe data dari satu tipe ke tipe yang lain. Pada program ini, dilakukan beberapa operasi casting yang ditunjukkan dengan menggunakan tanda kurung dan tipe data di depan nilai yang akan di-cast.

```

public class Casting2 {
    /Casting menggunakan tipe data Class/
    /
    @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);
    }
}

```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\LENOVO\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.Casting2'
a : 67
k : 45.0
d : 100.0
m : 9
l : 3.2
n : 67.0
c : 9.0
l : 3.2
PS D:\software>
```

1. **public class Casting2 {** : Mendeklarasikan kelas Java dengan nama **Casting2**. Nama kelas harus sama dengan nama file Java di mana kelas ini didefinisikan.
2. **public static void main(String[] args) {** : Metode **main** adalah metode utama yang akan dieksekusi ketika program dimulai. **args** adalah parameter yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. **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";** : Mendeklarasikan dan menginisialisasi beberapa variabel dengan tipe data yang berbeda.
4. **a = Integer.parseInt(n);** : Mengkonversi nilai dari variabel **n** yang bertipe **String** ke tipe data **int** menggunakan metode **parseInt** dari kelas **Integer**.
5. **k = Double.parseDouble(m);** : Mengkonversi nilai dari variabel **m** yang bertipe **String** ke tipe data **double** menggunakan metode **parseDouble** dari kelas **Double**.
6. **d = Float.parseFloat(l);** : Mengkonversi nilai dari variabel **l** yang bertipe **String** ke tipe data **float** menggunakan metode **parseFloat** dari kelas **Float**.
7. **n = String.valueOf(b);** : Mengkonversi nilai dari variabel **b** yang bertipe **int** ke tipe data **String** menggunakan metode **valueOf** dari kelas **String**.
8. **m = String.valueOf(g);** : Mengkonversi nilai dari variabel **g** yang bertipe **char** ke tipe data **String** menggunakan metode **valueOf** dari kelas **String**.
9. **l = String.valueOf(e);** : Mengkonversi nilai dari variabel **e** yang bertipe **float** ke tipe data **String** menggunakan metode **valueOf** dari kelas **String**.
10. **k = Double.valueOf(a).intValue();** : Mengkonversi nilai dari variabel **a** yang bertipe **int** ke tipe data **double** menggunakan metode **valueOf** dari kelas **Double**, kemudian mengkonversi kembali ke **int** menggunakan metode **intValue**.
11. **double c = Integer.valueOf(b).doubleValue();** : Mengkonversi nilai dari variabel **b** yang bertipe **int** ke tipe data **double** menggunakan metode **valueOf** dari kelas **Integer**, kemudian mengkonversi ke **double** menggunakan metode **doubleValue**.

System.out.println("a : "+a+"\nk : "+k+"\nd : "+d); : Mencetak

nilai variabel **a**, **k**, dan **d** ke konsol.

System.out.println("n : "+n+"\nm : "+m+"\nl : "+l); : Mencetak nilai variabel **n**, **m**, dan **l** ke konsol.

System.out.println("k : "+k+"\nc : "+c+"\nl : "+l); : Mencetak nilai variabel **k**, **c**, dan **l** ke konsol

Kode tersebut bertujuan untuk menunjukkan berbagai contoh konversi tipe data antara tipe data primitif dan objek String menggunakan metode *parsing* dan *valueOf*.

```

public class Ekspresi {

    / pemakaian operator kondisional /

    /

    @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));

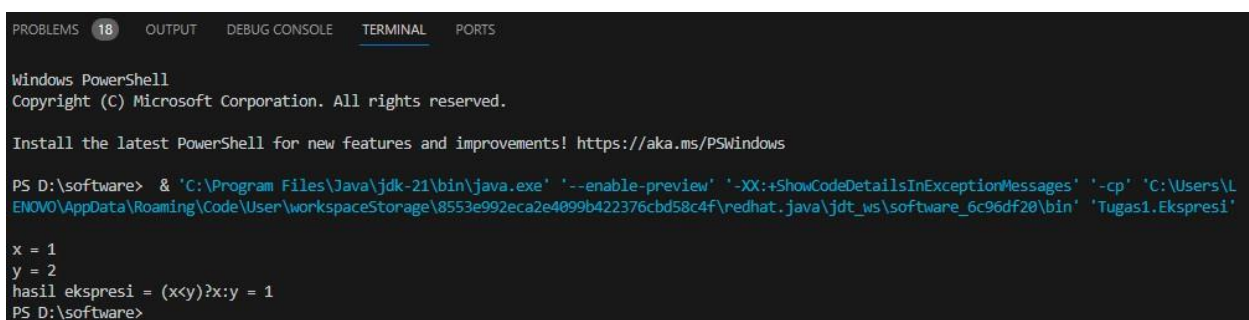
        /Gunakan dalam kurung "(statemen dan kondisi)"

        untuk menyatakan satu kesatuan pernyataan/

    }

}

```



```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L
ENOV0\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.Ekspresi'

x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
PS D:\software>

```

1. **public class Ekspresi {** : Mendeklarasikan kelas Java dengan nama **Ekspresi**. Nama kelas harus sama dengan nama file Java di mana kelas ini didefinisikan.
2. **public static void main(String[] args) {** : Metode **main** adalah metode utama yang akan dieksekusi ketika program dimulai. **args** adalah parameter yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. **int x = 1; int y = 2;** : Mendeklarasikan variabel **x** dan **y** sebagai

tipe data integer dan menginisialisasinya dengan nilai 1 dan 2.

- 4. **System.out.print("x = " + x + "\n"); System.out.print("y = " + y + "\n");** : Mencetak nilai variabel **x** dan **y** ke konsol.*
- 5. **System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y));** : Menggunakan operator kondisional (ternary) untuk mengevaluasi ekspresi **(x < y) ? x : y**. Jika kondisi **(x < y)** benar, maka nilai **x** akan dipilih, jika tidak, nilai **y** akan dipilih. Hasil dari ekspresi ini kemudian dicetak ke konsol.*

Bagian ini mencetak nilai variabel `x` dan `y` ke konsol menggunakan `System.out.print()`. Kemudian, sebuah ekspresi kondisional digunakan untuk mengevaluasi apakah nilai `x` kurang dari nilai `y`. Jika benar, maka hasilnya adalah nilai `x`, jika salah, maka hasilnya adalah nilai `y`. Hasil ekspresi tersebut kemudian dicetak ke konsol. Ekspresi kondisional digunakan di dalam tanda kurung dan diikuti oleh operator ternary (`? :`). Jika kondisi dalam tanda kurung bernilai benar, maka ekspresi setelah tanda tanya (`?`) akan dievaluasi, dan jika kondisi tersebut salah, maka ekspresi setelah titik dua (`:`) akan dievaluasi.

```

public class Ekpresil {

    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);

    }

}

```

```
PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L
ENOV0\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat_java\jdt_ws\software_6c96df20\bin' 'Tugas1.Ekpresil'

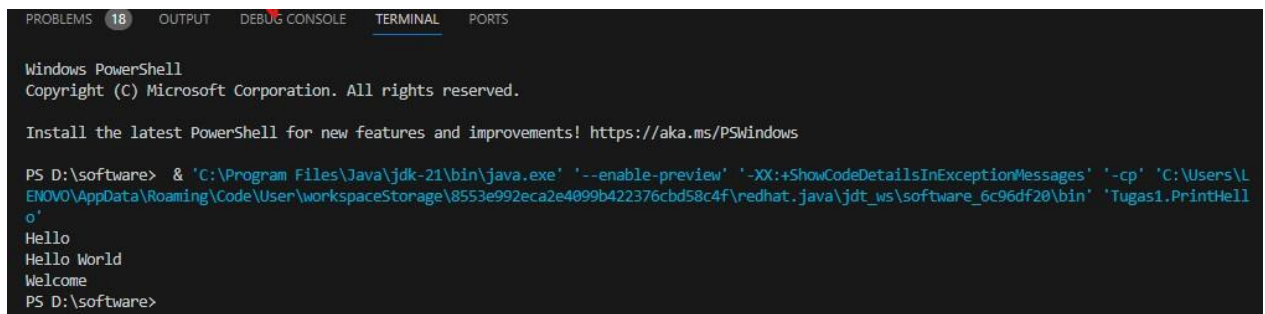
x/y (format integer) = 0
x/y (format float) = 0
x/y (format integer) = 0.5
x/y (format float) = 0.5
float(x)/float(y) (format integer)=0.5
float(x)/float(y) (format float) =0.5
x/y (format integer) = 3
x/y (format float) = 3
PS D:\software>
```

Penjelasan:

1. **public class Ekpresil {** : Mendeklarasikan kelas Java dengan nama **Ekpresil**. Nama kelas harus sama dengan nama file Java di mana kelas ini didefinisikan.
2. **public static void main(String[] args) {** : Metode **main** adalah metode utama yang akan dieksekusi ketika program dimulai. **args** adalah parameter yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. **int x = 1; int y = 2; float fx; float fy;** : Mendeklarasikan variabel **x** dan **y** sebagai tipe data integer dan menginisialisasinya dengan nilai 1 dan 2. Mendeklarasikan variabel **fx** dan **fy** sebagai tipe data float tanpa inisialisasi.
4. **System.out.print ("x/y (format integer) = "+ x/y);** : Mencetak hasil pembagian **x** dengan **y** dalam format integer. Pada baris ini, hasilnya adalah 0 karena pembagian integer akan menghasilkan nilai bulat tanpa pembulatan.
5. **System.out.print ("\nx/y (format float) = "+ x/y);** : Mencetak hasil pembagian **x** dengan **y** dalam format float. Meskipun **x** dan **y** bertipe integer, hasilnya akan dianggap float karena hasil pembagian adalah float.
6. **fx=x; fy=y;** : Mengkonversi nilai **x** dan **y** ke float dan menyimpannya dalam variabel **fx** dan **fy**.
7. **System.out.print ("\nx/y (format integer) = "+ fx/fy);** : Mencetak hasil pembagian **fx** dengan **fy** dalam format integer. Karena **fx** dan **fy** adalah float, hasilnya adalah nilai float yang sesuai.
8. **System.out.print ("\nx/y (format float) = "+ fx/fy);** : Mencetak hasil pembagian **fx** dengan **fy** dalam format float.
9. **System.out.print("\nfloat(x)/float(y) (format integer)="+(float)x/(float)y);** : Menggunakan operasi casting untuk mengkonversi **x** dan **y** ke float sebelum melakukan pembagian, dan kemudian mencetak hasilnya dalam format integer.
10. **System.out.print("\nfloat(x)/float(y) (format float) ="+(float)x/(float)y);** : Menggunakan operasi casting untuk mengkonversi **x** dan **y** ke float sebelum melakukan pembagian, dan kemudian mencetak hasilnya dalam format float.
11. **x = 10; y = 3;** : Mengubah nilai **x** menjadi 10 dan **y** menjadi 3.
12. **System.out.print ("\nx/y (format integer) = "+ x/y);** : Mencetak hasil pembagian **x** dengan **y** setelah nilai **x** dan **y** diubah dalam format integer.
13. **System.out.print ("\nx/y (format float = "+ x/y);** : Mencetak

hasil pembagian **x** dengan **y** setelah nilai **x** dan **y** diubah dalam format float.

```
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");  
    }  
}
```



The screenshot shows a Windows PowerShell terminal window with the following content:

```
PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L  
ENOV0\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.PrintHell  
o'  
Hello  
Hello World  
Welcome  
PS D:\software>
```

1. `public class PrintHello {`

Ini adalah deklarasi kelas `PrintHello`, yang dinyatakan sebagai kelas publik. Ini berarti kelas `PrintHello` dapat diakses dari luar paket di mana kelas tersebut berada.

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


```
// Isi method main  
}
```

Ini adalah metode utama (`main`) yang merupakan titik awal eksekusi untuk program Java. Setiap program Java harus memiliki satu metode `main` yang dijalankan saat program tersebut dijalankan. Parameter `args` adalah parameter dari metode `main` yang dapat digunakan untuk menerima argumen dari baris perintah saat menjalankan program.

```
3. System.out.print("Hel
```

```
lo");
```

```
System.out.print("\nH
```

```
ello ");
```

```
System.out.println("W
```

```
orld");
```

```
System.out.println("W
```

```
elcome");
```

Dalam metode `main`, empat pernyataan `System.out.println()` atau `System.out.print()` digunakan untuk mencetak output ke konsol.

- Pernyataan pertama mencetak string "Hello" ke layar menggunakan `System.out.print()`. Perbedaan antara `print()` dan `println()` adalah `println()` menambahkan baris baru setelah mencetak, sedangkan `print()` tidak.

- Pernyataan kedua mencetak string "Hello " ke layar menggunakan `System.out.print()`, namun disertai dengan baris baru (`\n`) di awal string sehingga hasilnya akan mencetak "Hello " di baris baru.

- Pernyataan ketiga mencetak string "World" ke layar menggunakan `System.out.println()`. Karena menggunakan `println()`, maka setelah mencetak "World", akan ditambahkan baris baru.

- Pernyataan keempat mencetak string "Welcome" ke layar menggunakan `System.out.println()`. Karena menggunakan `println()`, maka setelah mencetak "Welcome", akan ditambahkan baris baru.

Dengan demikian, hasil output dari program ini akan terlihat sebagai berikut:

```
HelloWo
```

```
rld
```

```
Welcome
```

Perhatikan bahwa output "Hello" dan "Hello " tidak diikuti oleh baris baru, karena menggunakan `System.out.print()`. Sedangkan output "World" dan "Welcome" diikuti

oleh baris baru, karena menggunakan ``System.out.println()``.

```

public class Incar {

    / Efek dari operator ++ /

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        / Kamus

        / int i,

        j;

        / Program /

        i = 3;

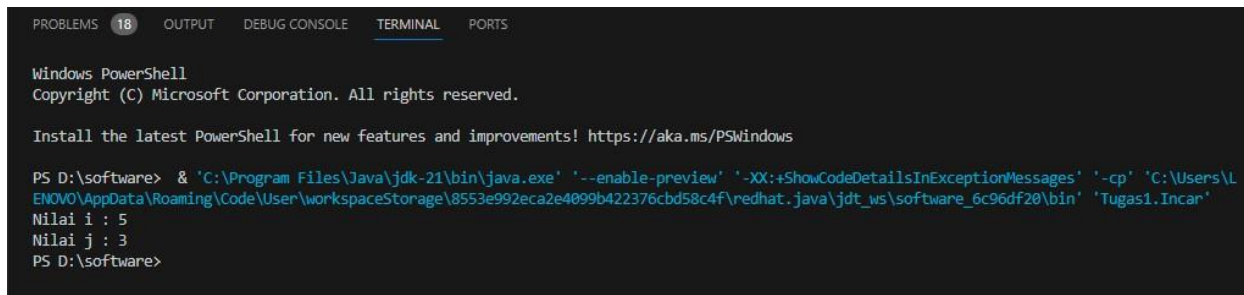
        j = i++;

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

    }

}

```



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L
ENOV0\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.Incar'
Nilai i : 5
Nilai j : 3
PS D:\software>

```

1. `public class Incar {`

Ini adalah deklarasi kelas `Incar`, yang dinyatakan sebagai kelas publik. Ini berarti kelas `Incar` dapat diakses dari luar paket di mana kelas tersebut berada.

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

```

    // Isi method main

}

```

Ini adalah metode utama (`main`) yang merupakan titik awal eksekusi untuk program Java. Setiap program Java harus memiliki satu metode `main` yang dijalankan saat program tersebut dijalankan.

Parameter ``args`` adalah parameter dari metode ``main`` yang dapat digunakan untuk menerima argumen dari baris perintah saat menjalankan program.

3. `int i, j;`

Ini adalah bagian deklarasi variabel lokal. Variabel ``i`` dan ``j`` dideklarasikan sebagai tipe data ``int`` tanpa inisialisasi.

4. `i = 3;`

`j = i++;`

Pada bagian ini, nilai ``i`` diatur menjadi 3. Kemudian, nilai ``j`` diatur menjadi nilai dari ``i`` sebelum operasi penambahan (``++``) dijalankan. Ini berarti nilai ``j`` akan menjadi 3, dan kemudian nilai ``i`` akan ditambah 1.

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

Bagian ini mencetak nilai dari ``i`` dan ``j`` ke konsol. Perhatikan bahwa sebelum mencetak nilai ``i``, operasi penambahan (``++``) dilakukan terlebih dahulu terhadap nilai ``i``. Oleh karena itu, nilai ``i`` akan menjadi 5

(nilai setelah operasi penambahan), dan nilai ``j`` tetap 3 (karena telah ditetapkan sebelum operasi penambahan).

```

package Tugas1;

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 /

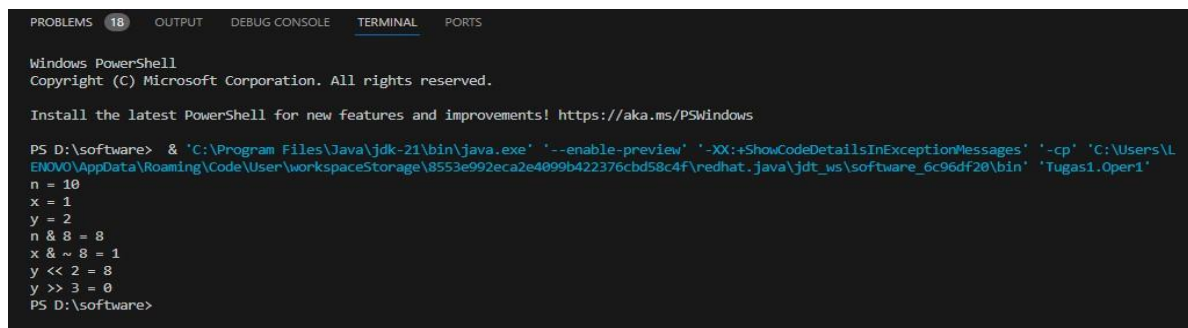
        System.out.println ("y >> 3 = "+ (y >>3)); /10

        ==>0000 = 0 /

    }

}

```



```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L
ENOV0\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat_java\jdt_ws\software_6c96df20\bin' 'Tugas1.Oper1'
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
PS D:\software>

```

1. `public class Oper1 {`

Ini adalah deklarasi kelas `Oper1`, yang dinyatakan sebagai kelas publik. Ini berarti kelas `Oper1` dapat diakses dari luar paket di mana kelas tersebut berada.

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

```
    // Isi method main

}
```

Ini adalah metode utama (`main`) yang merupakan titik awal eksekusi untuk program Java. Setiap program Java harus memiliki satu metode `main` yang dijalankan saat program tersebut dijalankan. Parameter `args` adalah parameter dari metode `main` yang dapat digunakan untuk menerima argumen dari baris perintah saat menjalankan program.

```
3. int n = 10; // 1010
```

```
    int x = 1; // 0001
```

```
    int y = 2; // 0010
```

Ini adalah bagian deklarasi variabel lokal. Variabel `n`, `x`, dan `y` dideklarasikan sebagai tipe data `int` dan diinisialisasi dengan nilai masing-masing.

```
4. System.out.println ("n = "+
```

```
    n); System.out.println ("x =
```

```
    "+ x); System.out.println ("y
```

```
    = "+ y);
```

Bagian ini mencetak nilai dari variabel `n`, `x`, dan `y` ke

```
konsol. System.out.println("n & 8 = "+ (n & 8)); // 1010 AND
```

```
100
```

Operasi bitwise AND (`&`) dilakukan antara nilai `n` dan 8. Representasi biner dari 8 adalah 1000. Jadi, operasi `n & 8` akan menghasilkan nilai 1000 karena hanya bit yang cocok dalam kedua angka akan menghasilkan 1.

```
System.out.println ("x & ~8 = "+ (x & ~8)); // 1 AND 0111
```

Operasi bitwise NOT (`~`) dilakukan terhadap nilai 8 yang menghasilkan 0111 (nilai biner yang di-complement dari 8). Kemudian, operasi bitwise AND (`&`) dilakukan antara nilai `x` (1) dan nilai hasil negasi (`~8` atau 0111). Hasilnya adalah 0001 AND 0111 yang menghasilkan 0001.

```
System.out.println ("y << 2 = "+ (y << 2)); // 10 ==> 1000 = 8
```

Operasi bitwise left shift (`<<`) dilakukan pada nilai `y` (2) sebanyak 2 kali. Operasi ini akan menggeser semua bit dalam representasi biner dari nilai

`y` ke kiri sebanyak 2 kali. Sehingga, nilai biner 10 akan menjadi 1000 setelah digeser. Hasilnya adalah 8 dalam desimal.

```
System.out.println ("y >> 3 = "+ (y >>3)); // 10 ==>0000 = 0
```

Operasi bitwise right shift (`>>`) dilakukan pada nilai `y` (2) sebanyak 3 kali. Operasi ini akan menggeser semua bit dalam representasi biner dari nilai `y` ke kanan sebanyak 3 kali. Sehingga, nilai biner 10 akan menjadi 0000 setelah digeser. Hasilnya adalah 0 dalam desimal.

```

package Tugas1;

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/

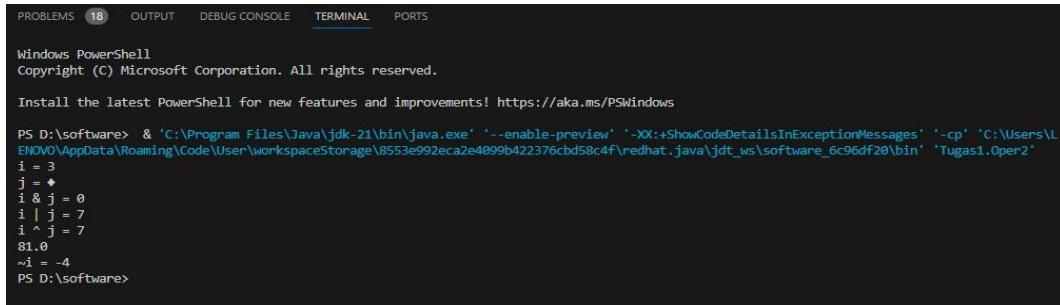
        System.out.println("~i = "+ ~i);

        / -4: 11111100 biner /

    }

}

```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L
ENOV\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.Oper2'
i = 3
j = 4
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
PS D:\software>
```

1. **package Tugas1;** : Mendefinisikan paket (package) Java dengan nama **Tugas1**. Ini adalah cara untuk mengelompokkan kelas-kelas terkait bersama dalam satu paket.
2. **public class Oper2 {** : Mendeklarasikan kelas Java dengan nama **Oper2**. Nama kelas harus sama dengan nama file Java di mana kelas ini didefinisikan.
3. **public static void main(String[] args) {** : Metode **main** adalah metode utama yang akan dieksekusi ketika program dimulai. **args** adalah parameter yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
4. **char i, j;** : Mendeklarasikan variabel **i** dan **j** sebagai tipe data **char**.
5. **i = 3;** : Menginisialisasi variabel **i** dengan nilai 3. Dalam representasi biner, ini sama dengan 00000011.
6. **j = 4;** : Menginisialisasi variabel **j** dengan nilai 4. Dalam representasi biner, ini sama dengan 00000100.
7. **System.out.println("i = "+ (int) i);** : Mencetak nilai variabel **i** sebagai integer ke layar.
8. **System.out.println("j = "+ j);** : Mencetak nilai variabel **j** ke layar.
9. **System.out.println("i & j = "+ (i & j));** : Melakukan operasi bitwise AND (&) antara nilai **i** dan **j**, dan mencetak hasilnya. Hasilnya adalah 0 (00000000) karena tidak ada bit yang cocok di kedua nilai.
10. **System.out.println("i | j = "+ (i | j));** : Melakukan operasi bitwise OR (|) antara nilai **i** dan **j**, dan mencetak hasilnya. Hasilnya adalah 7 (00000111) karena ada bit 1 di salah satu nilai.
11. **System.out.println("i ^ j = "+ (i ^ j));** : Melakukan operasi bitwise XOR (^) antara nilai **i** dan **j**, dan mencetak hasilnya. Hasilnya adalah 7 (00000111) karena hanya ada satu bit yang berbeda di antara kedua nilai.
12. **System.out.println(Math.pow(i, j));** : Mencoba untuk menggunakan operator ^ sebagai operasi pangkat, tetapi ini tidak valid untuk tipe data **char**. Sebaiknya gunakan operator bitwise XOR (^) untuk operasi XOR.
13. **System.out.println("~i = "+ ~i);** : Melakukan operasi bitwise NOT (~) pada nilai **i**, dan mencetak hasilnya. Hasilnya adalah -4 (11111100) karena nilai **i** dianggap sebagai bilangan bertanda (signed char).

```

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 /

    }

}

```

```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\LENOVO\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.Oper3'
true
false
true
true
true
true
PS D:\software>

```

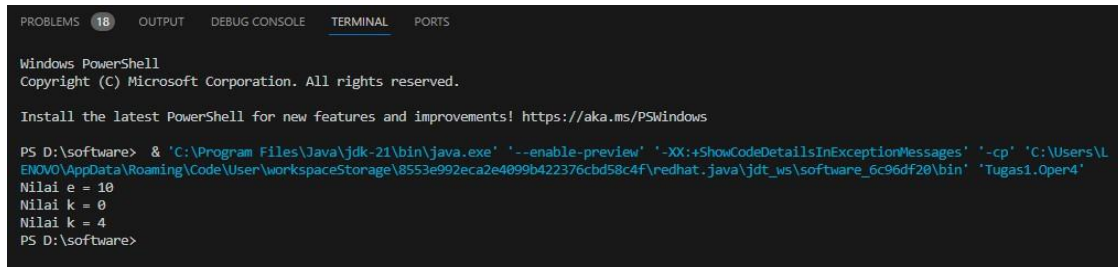
Penjelasan:

1. **public class Oper3 {** : Mendeklarasikan kelas Java dengan nama **Oper3**. Nama kelas harus sama dengan nama file Java di mana kelas ini didefinisikan.
2. **public static void main(String[] args) {** : Metode **main** adalah metode utama yang akan dieksekusi ketika program dimulai. **args** adalah parameter yang dapat digunakan untuk menerima argumen dari baris perintah saat program dijalankan.
3. **if (true && true){ System.out.println(true && true); }** : Menggunakan operator logika **&&** (AND) untuk mengecek apakah kedua kondisi bernilai **true**. Jika keduanya benar, maka blok kode dalam if statement akan dieksekusi dan mencetak **true**.

4. **if (true & true) { System.out.println(true & false); }** : Menggunakan operator logika **&** (AND bitwise) untuk mengecek apakah kedua kondisi bernilai **true**. Operator **&** akan melakukan operasi bitwise pada nilai biner dari operand-nya. Dalam konteks boolean, hasilnya tetap sama dengan operator **&&**.
5. **if (true){ System.out.println(true); }** : Mencetak **true** karena kondisinya adalah benar.
6. **if (true || true){ System.out.println(true); }** : Menggunakan operator logika **||** (OR) untuk mengecek apakah salah satu kondisi bernilai **true**. Jika salah satu benar, maka blok kode dalam if statement akan dieksekusi dan mencetak **true**.
7. **if (true|false) { System.out.println(true|false); }** : Menggunakan operator logika **|** (OR bitwise) untuk mengecek apakah salah satu kondisi bernilai **true**. Operator **|** akan melakukan operasi bitwise pada nilai biner dari operand-nya. Dalam konteks boolean, hasilnya tetap sama dengan operator **||**.

```
public class Oper4 {
    / Operator terner /
    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);
    }
}
```

```
}  
  
}
```



```
PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L  
ENOMO\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.Oper4'  
Nilai e = 10  
Nilai k = 0  
Nilai j = 4  
PS D:\software>
```

Penjelasan:

1. **int e = (((int)c > (int)d) ? c : d);** : Jika nilai **c** lebih besar dari nilai **d**, maka nilai **e** akan diisi dengan nilai **c**, jika tidak, maka nilai **e** akan diisi dengan nilai **d**.
2. **int k = ((i>j) ? i : j);** : Jika nilai **i** lebih besar dari nilai **j**, maka nilai **k** akan diisi dengan nilai **i**, jika tidak, maka nilai **k** akan diisi dengan nilai **j**.
3. **k = ((i++>j++) ? i : j);** : Ekspresi ini menunjukkan bahwa nilai **k** akan diisi dengan nilai **i** jika nilai **i** lebih besar dari nilai **j**, dan kemudian nilai **i** dan **j** akan ditambah 1.

Cara kerja operator ternary adalah sebagai berikut:

- Operator ini akan mengevaluasi ekspresi boolean di bagian pertama.
- Jika ekspresi tersebut bernilai **true**, maka ekspresi di bagian kedua akan dieksekusi dan nilai dari ekspresi tersebut akan menjadi hasil dari operator ternary.
- Jika ekspresi tersebut bernilai **false**, maka ekspresi di bagian ketiga akan dieksekusi dan nilai dari ekspresi tersebut akan menjadi hasil dari operator ternary.

Dalam kasus penggunaan operator ternary pada variabel **k** di kode Anda, karena ekspresi **i++>j++** bernilai **false** (karena **i** tidak lebih besar dari **j** setelah penambahan), maka nilai **k** diisi dengan nilai **j** yang sudah ditambah 1. Setelah ekspresi dievaluasi, nilai **i** dan **j** akan bertambah 1.

```

public class Operator {
    / Contoh pengoperasian variabel bertipe dasar /
    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 /
        System.out.println( "hasil dari BOOL1 && BOOL2 adalah
"+TF); TF = Bool1 || Bool2 ; / Boolean OR /
        System.out.println( "hasil dari BOOL1 || BOOL2 adalah
"+TF); TF = ! Bool1 ; / NOT /
        System.out.println( "hasil dari ! BOOL1 adalah
"+TF); TF = Bool1 ^Bool2; / XOR /
        System.out.println( "hasil dari BOOL1 ^BOOL2 adalah "+TF);
        / operasi numerik
        / i = 5; j = 2 ;
        hsl = i+j;
        System.out.println( "hasil dari i + j adalah
"+hsl); hsl = i - j;
        System.out.println( "hasil dari i - j adalah
"+hsl); hsl = i / j;
        System.out.println( "hasil dari i / j adalah "+hsl);

        hsl = i % j;
        System.out.println( "hasil dari i % j adalah
"+hsl); hsl = i /j ; / pembagian bulat /

```

```
System.out.println( "hasil dari i / j adalah  
"+hsl); hsl = i%j ; / sisa modulo /  
System.out.println( "hasil dari i % j adalah "+hsl);  
/ operasi numerik  
/ x = 5 ; y = 5 ;  
res = x + y;  
System.out.println( "hasil dari x + y adalah  
"+res); res = x - y;  
System.out.println( "hasil dari x - y adalah  
"+res); res = x / y;  
System.out.println( "hasil dari x / y adalah  
"+res); res = x * y;  
System.out.println( "hasil dari x * y adalah "+res);  
/ operasi relasional numerik  
/ TF = (i==j);  
System.out.println( "hasil dari (i==j) adalah  
"+TF); TF = (i!=j);  
System.out.println( "hasil dari (i!=j) adalah  
"+TF); TF = (i < j);  
System.out.println( "hasil dari (i<j) adalah  
"+TF); TF = (i > j);  
System.out.println( "hasil dari (i>j) adalah  
"+TF); TF = (i <= j);  
System.out.println( "hasil dari (i<=j) adalah  
"+TF); TF = (i >= j);  
System.out.println( "hasil dari (i>=j) adalah "+TF);
```

```

        / operasi relasional numerik

        / TF = (x != y);

        System.out.println( "hasil dari (x != y) adalah
        "+TF); TF = (x < y);

        System.out.println( "hasil dari (x < y) adalah
        "+TF); TF = (x > y);

        System.out.println( "hasil dari (X > y) adalah
        "+TF); TF = (x <= y);

        System.out.println( "hasil dari (x <= y) adalah
        "+TF); TF = (x >= y);

        System.out.println( "hasil dari (x >= y) adalah "+TF);

    }
}

```

```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\software> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\L
ENOV0\AppData\Roaming\Code\User\workspaceStorage\8553e992eca2e4099b422376cbd58c4f\redhat.java\jdt_ws\software_6c96df20\bin' 'Tugas1.Operator'

Silahkan baca teksnya dan tambahkan perintah program di bawah ini untuk menampilkan output program
hasil dari 800L1 && 800L2 adalah false
hasil dari 800L1 || 800L2 adalah true
hasil dari ! 800L1 adalah false
hasil dari 800L1 ^800L2 adalah true
hasil dari i + j adalah 7
hasil dari i - j adalah 3
hasil dari i / j adalah 2
hasil dari i * j adalah 10
hasil dari i / j adalah 2
hasil dari i % j adalah 1
hasil dari x + y adalah 10.0
hasil dari x - y adalah 0.0
hasil dari x / y adalah 1.0
hasil dari x * y adalah 25.0
hasil dari (i==j) adalah false
hasil dari (i!=j) adalah true
hasil dari (i<j) adalah false
hasil dari (i>j) adalah true
hasil dari (i<=j) adalah false
hasil dari (i>=j) adalah true
hasil dari (x != y) adalah false
hasil dari (x < y) adalah false
hasil dari (x > y) adalah false
hasil dari (x <= y) adalah true
hasil dari (x >= y) adalah true
PS D:\software>

```

Penjelasan:

`public class Operator`

Ini adalah deklarasi kelas ``Operator``, yang dinyatakan sebagai kelas publik. Ini berarti kelas ``Operator`` dapat diakses dari luar paket di mana kelas tersebut berada.

```

1. public static void main(String[] args) {

    // Isi method main

}

```

Ini adalah metode utama (``main``) yang merupakan titik awal eksekusi untuk program Java. Setiap program Java harus memiliki satu metode ``main`` yang dijalankan saat program tersebut dijalankan. Parameter ``args`` adalah parameter dari metode ``main`` yang dapat digunakan untuk menerima argumen dari baris perintah saat menjalankan program.

1. Deklarasi Variabel:

- `boolean Bool1, Bool2, TF;` : Deklarasi variabel boolean **Bool1**, **Bool2**, dan **TF**.
- `int i, j, hsl;` : Deklarasi variabel integer **i**, **j**, dan **hsl**.
- `float x, y, res;` : Deklarasi variabel float **x**, **y**, dan **res**.

2. Operasi Logika:

- `TF = Bool1 && Bool2;` : Operasi logika AND pada variabel boolean **Bool1** dan **Bool2**.
- `TF = Bool1 || Bool2;` : Operasi logika OR pada variabel boolean **Bool1** dan **Bool2**.
- `TF = !Bool1;` : Operasi logika NOT pada variabel boolean **Bool1**.
- `TF = Bool1 ^ Bool2;` : Operasi logika XOR pada variabel boolean **Bool1** dan **Bool2**.

3. Operasi Aritmatika:

- `hsl = i + j;` : Penjumlahan variabel integer **i** dan **j**.
- `hsl = i - j;` : Pengurangan variabel integer **i** dan **j**.
- `hsl = i * j;` : Perkalian variabel integer **i** dan **j**.
- `hsl = i / j;` : Pembagian variabel integer **i** dengan **j**.
- `hsl = i % j;` : Modulus (sisanya pembagian) variabel integer **i** dengan **j**.

4. Operasi Relasional:

- `TF = (i == j);` : Memeriksa apakah **i** sama dengan **j**.
- `TF = (i != j);` : Memeriksa apakah **i** tidak sama dengan **j**.
- `TF = (i < j);` : Memeriksa apakah **i** kurang dari **j**.
- `TF = (i > j);` : Memeriksa apakah **i** lebih besar dari **j**.
- `TF = (i <= j);` : Memeriksa apakah **i** kurang dari atau sama dengan **j**.
- `TF = (i >= j);` : Memeriksa apakah **i** lebih besar dari atau sama dengan **j**.