

**LAPORAN**  
**PEMROGRAMAN BERORIENTASI OBJEK**



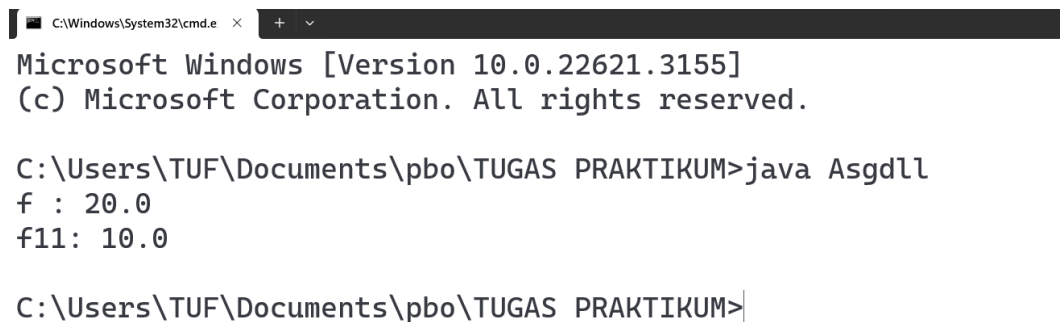
Nama : Narendra Awangga  
Stambuk : 13020220022  
Kelas : A1  
Dosen : Mardiyah Hasnawi, S.Kom.,M.T

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS MUSLIM INDONESIA**  
**MAKASSAR**  
**2023**

## 1. Kode program

```
public class Asgdll {  
  
    public static void main(String[] args) {  
  
        float f= 20.0f;  
        double fl;  
  
        fl=10.0f;  
        System.out.println ("f : "+f+ "\nf11: "+fl);  
    }  
}
```

Output :



```
C:\Windows\System32\cmd.e  x  +  v  
Microsoft Windows [Version 10.0.22621.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Asgdll  
f : 20.0  
f11: 10.0  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>
```

Penjelasan Program :

Program tersebut adalah program Java sederhana yang memiliki tujuan untuk mendeklarasikan dan menginisialisasi dua variabel bertipe data bilangan pecahan (floating point), yaitu **float f** dan **double fl**, kemudian mencetak nilai dari kedua variabel tersebut ke konsol.

Penjelasan singkat tentang keyword yang digunakan dari variabel/tipe data/kelas:

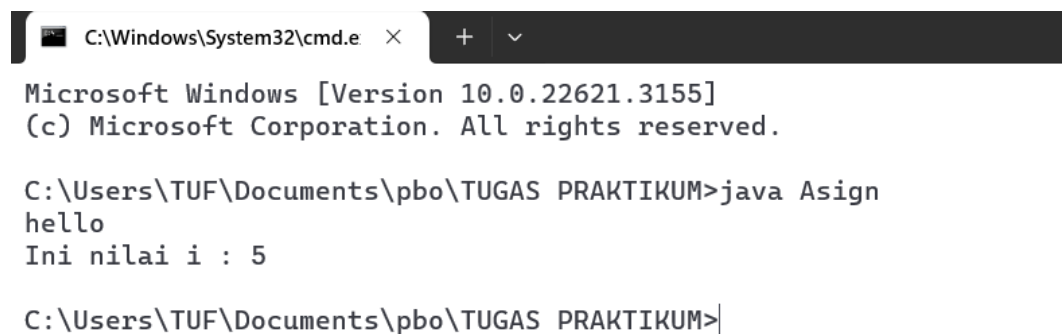
1. **public class Asgdll {**: Mendeklarasikan sebuah kelas dengan nama **Asgdll**. Kata kunci **public** menunjukkan bahwa kelas ini dapat diakses dari luar paket. Nama kelas diikuti oleh tanda kurung buka **{** menandakan dimulainya blok kode untuk kelas.

2. **public static void main(String[] args) {**: Mendefinisikan metode utama (**main**) yang merupakan titik awal eksekusi program. Kata kunci **public** menandakan bahwa metode ini dapat diakses dari luar kelas, **static** menunjukkan bahwa metode ini milik kelas itu sendiri dan tidak memerlukan instansiasi objek, **void** menunjukkan bahwa metode ini tidak mengembalikan nilai. **String[] args** adalah parameter yang diterima oleh metode utama yang digunakan untuk meneruskan argumen baris perintah ke program.
3. **float f= 20.0f;**: Mendeklarasikan variabel **f** sebagai tipe data **float** dan memberikannya nilai awal 20.0f. Nilai **20.0f** adalah literal float.
4. **double fl;**: Mendeklarasikan variabel **fl** sebagai tipe data **double** tanpa memberikan nilai awal.
5. **fl=10.0f;**: Memberi nilai 10.0f ke variabel **fl**. Meskipun **fl** adalah variabel bertipe **double**, nilai float dapat dengan aman ditingkatkan menjadi double. Dalam konteks ini, konversi dari float ke double terjadi secara otomatis.
6. **System.out.println ("f : "+f + "\nf11: "+fl);**: Mencetak nilai **f** dan **fl** ke konsol. **System.out.println** adalah metode yang digunakan untuk mencetak ke konsol. Operator **+** digunakan untuk menggabungkan string dengan nilai variabel. Baris baru (**\n**) digunakan untuk membuat output lebih terstruktur.
7. **}**: Penutup dari blok kode untuk metode **main**.
8. **}**: Penutup dari blok kode untuk kelas **Asgdll**.

## 2. Kode Program

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

Output :



```
C:\Windows\System32\cmd.e  X  +  v  
Microsoft Windows [Version 10.0.22621.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Asign  
hello  
Ini nilai i : 5  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>
```

Penjelasan Program :

Program tersebut memiliki tujuan untuk mencetak pesan "hello" ke konsol, kemudian menginisialisasi variabel `i` dengan nilai 5, dan mencetak nilai variabel `i` beserta pesan penjelasan ke konsol.

Penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas:

1. `public class Asign` {: Mendeklarasikan kelas `Asign`. Kata kunci `public` menandakan bahwa kelas ini dapat diakses dari luar paket.
2. `public static void main(String[] args)` {: Mendeklarasikan metode `main`, yang merupakan titik awal eksekusi program. Kata kunci `public` menunjukkan

bahwa metode ini dapat diakses dari luar kelas, **static** menunjukkan bahwa metode ini milik kelas itu sendiri dan tidak memerlukan instansiasi objek, **void** menunjukkan bahwa metode ini tidak mengembalikan nilai. **String[] args** adalah parameter yang diterima oleh metode **main**, yang digunakan untuk meneruskan argumen baris perintah ke program.

3. **int i**; Mendeklarasikan variabel **i** sebagai tipe data **int** tanpa memberikan nilai awal.
4. **System.out.print ("hello\n");**; Mencetak pesan "hello" ke konsol tanpa baris baru (**\n** digunakan untuk membuat baris baru setelah mencetak pesan).
5. **i = 5**; Menginisialisasi variabel **i** dengan nilai 5.
6. **System.out.println ("Ini nilai i : " + i);**; Mencetak nilai variabel **i** beserta pesan penjelasan ke konsol. **System.out.println** digunakan untuk mencetak nilai variabel **i** ke konsol, sementara operator **+** digunakan untuk menggabungkan string dengan nilai variabel **i**.

Dengan demikian, program ini sederhana dan tujuannya adalah untuk mendemonstrasikan penggunaan variabel, pencetakan pesan ke konsol, dan inisialisasi nilai variabel dalam bahasa pemrograman Java.

### 3. Kode program

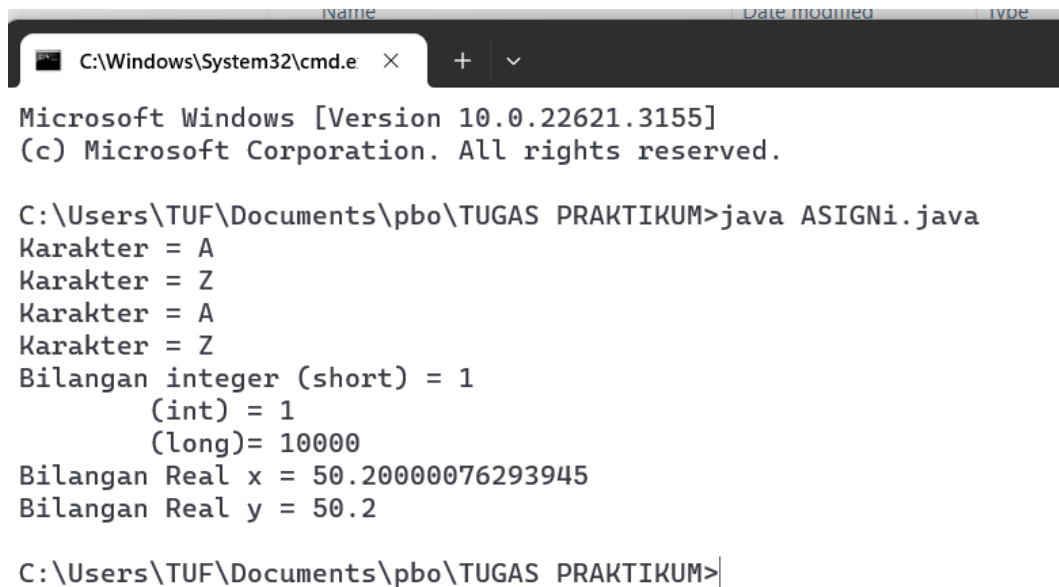
```
public class ASIGNi {  
  
    public static void main(String[] args) {  
  
        short ks = 1;  
        int ki = 1;  
        long kl = 10000;  
        char c = 65;  
        char c1 = 'Z';  
        double x = 50.2f;  
        float y = 50.2f;  
  
        System.out.println("Karakter = " + c);  
        System.out.println("Karakter = " + c1);  
    }  
}
```

```

        System.out.println("Karakter = " + c);
        System.out.println("Karakter = " + c1);
        System.out.println("Bilangan integer (short) = " +
            ks);
        System.out.println("\t(int) = " + ki);
        System.out.println("\t(long)= " + kl);
        System.out.println("Bilangan Real x = " + x);
        System.out.println("Bilangan Real y = " + y);
    }
}

```

Output :



```

Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java ASIGNi.java
Karakter = A
Karakter = Z
Karakter = A
Karakter = Z
Bilangan integer (short) = 1
        (int) = 1
        (long)= 10000
Bilangan Real x = 50.20000076293945
Bilangan Real y = 50.2

C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>

```

Penjelasan Program :

Program tersebut memiliki tujuan untuk mendemonstrasikan penggunaan berbagai jenis variabel dan tipe data dalam bahasa pemrograman Java, serta cara mencetak nilai-nilai variabel tersebut ke konsol.

Penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas:

1. **public class ASIGNi** {}: Mendeklarasikan kelas **ASIGNi**. Kata kunci **public** menandakan bahwa kelas ini dapat diakses dari luar paket.
2. **public static void main(String[] args)** {}: Mendeklarasikan metode **main**, yang merupakan titik awal eksekusi program. Kata kunci **public** menunjukkan

bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini milik kelas itu sendiri dan tidak memerlukan instansiasi objek, `void` menunjukkan bahwa metode ini tidak mengembalikan nilai. `String[] args` adalah parameter yang diterima oleh metode `main`, yang digunakan untuk meneruskan argumen baris perintah ke program.

3. `short ks = 1;`, `int ki = 1;`, `long kl = 10000;`: Mendeklarasikan variabel bertipe data bilangan bulat (`short`, `int`, dan `long`) dan memberikan nilai awal ke masing-masing variabel.
4. `char c = 65;`, `char c1 = 'Z';`: Mendeklarasikan variabel bertipe data karakter (`char`) dan memberikan nilai awal baik dalam bentuk integer maupun karakter.
5. `double x = 50.2f;`, `float y = 50.2f;`: Mendeklarasikan variabel bertipe data bilangan real (`double` dan `float`) dan memberikan nilai awal ke masing-masing variabel.
6. `System.out.println`: Menggunakan metode `println` dari kelas `System.out` untuk mencetak nilai-nilai variabel ke konsol. `println` digunakan untuk mencetak nilai dengan tambahan baris baru setelah setiap cetakan.

Tujuan dari program ini adalah untuk menunjukkan cara mendeklarasikan, menginisialisasi, dan mencetak nilai variabel-variabel dengan berbagai jenis tipe data yang tersedia dalam Java. Dengan demikian, program ini berfungsi sebagai contoh penggunaan variabel dan tipe data dalam bahasa pemrograman Java.

#### 4. Kode Program

```
import java.util.Scanner;
public class BacaData {

    public static void main(String[] args) {
        int a;
        Scanner masukan;
        System.out.print("Contoh membaca dan menulis, ketik nilai integer:
\n");
        masukan = new Scanner(System.in);
        a = masukan.nextInt();
        System.out.print("Nilai yang dibaca : " + a);
    }
}
```

Output :

```
C:\Windows\System32\cmd.e × + v
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java BacaData.java
Contoh membaca dan menulis, ketik nilai integer:
5
Nilai yang dibaca : 5
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>
```

Penjelasan Program :

Program tersebut memiliki tujuan untuk mendemonstrasikan penggunaan kelas **Scanner** dalam bahasa pemrograman Java untuk membaca input dari pengguna.

Penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas:

1. **import java.util.Scanner;**: Mengimpor kelas **Scanner** dari paket **java.util**, yang akan digunakan untuk membaca input dari pengguna.
2. **public class BacaData {**: Mendeklarasikan kelas **BacaData**. Kata kunci **public** menandakan bahwa kelas ini dapat diakses dari luar paket.
3. **public static void main(String[] args) {**: Mendeklarasikan metode **main**, yang merupakan titik awal eksekusi program. Kata kunci **public** menunjukkan bahwa metode ini dapat diakses dari luar kelas, **static** menunjukkan bahwa metode ini milik kelas itu sendiri dan tidak memerlukan instansiasi objek, **void** menunjukkan bahwa metode ini tidak mengembalikan nilai. **String[] args** adalah parameter yang diterima oleh metode **main**, yang digunakan untuk meneruskan argumen baris perintah ke program.
4. **int a;**: Mendeklarasikan variabel **a** sebagai tipe data **int** untuk menyimpan nilai integer yang akan dimasukkan oleh pengguna.
5. **Scanner masukan;**: Mendeklarasikan variabel **masukan** sebagai objek dari kelas **Scanner**. Ini akan digunakan untuk membaca input dari pengguna.



6. `System.out.print ("Contoh membaca dan menulis, ketik nilai integer: \n");` Mencetak pesan ke konsol untuk meminta pengguna untuk memasukkan nilai integer.
7. `masukan = new Scanner(System.in);` Membuat objek `Scanner` baru dengan parameter `System.in`, yang berarti akan membaca input dari konsol.
8. `a = masukan.nextInt();` Membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel `a`. Metode `nextInt()` dari objek `masukan` digunakan untuk membaca nilai integer.
9. `System.out.print ("Nilai yang dibaca : "+ a);` Mencetak nilai yang telah dibaca dari pengguna ke konsol.

Tujuan dari program ini adalah untuk mendemonstrasikan cara menggunakan kelas `Scanner` untuk membaca input dari pengguna dan kemudian mencetaknya ke konsol. Dengan menggunakan objek `Scanner`, program ini memungkinkan pengguna untuk memberikan input dari keyboard dan kemudian mencetak kembali nilai tersebut.

## 5. Kode Program

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Bacakar {

    public static void main(String[] args) throws IOException {

        char cc;
        int bil;

        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader dataIn = new BufferedReader(isr);

        BufferedReader datAIn = new BufferedReader(new
        InputStreamReader(System.in));
```

```

        System.out.print("hello\n");
        System.out.print("baca 1 karakter : ");

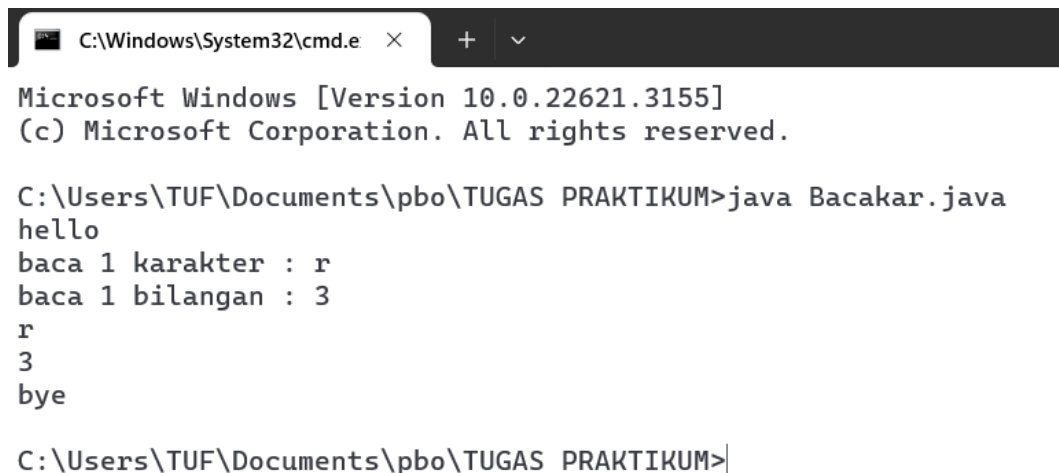
        cc = dataIn.readLine().charAt(0);
        System.out.print("baca 1 bilangan : ");

        bil = Integer.parseInt(dataIn.readLine());

        System.out.print(cc + "\n" + bil + "\n");
        System.out.print("bye \n");
    }
}

```

Output :



```

C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Bacakar.java
hello
baca 1 karakter : r
baca 1 bilangan : 3
r
3
bye

C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>

```

Penjelasan Program :

Program ini bertujuan untuk mendemonstrasikan beberapa metode untuk membaca input dari pengguna menggunakan kelas-kelas seperti **BufferedReader**, **InputStreamReader**, dan **JOptionPane**.

Penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas:

1. **import java.io.BufferedReader;**, **import java.io.IOException;**, **import java.io.InputStreamReader;**: Mengimpor kelas-kelas yang diperlukan dari paket **java.io** untuk membaca input dari pengguna.
2. **public static void main(String[] args) throws IOException {:** Mendeklarasikan metode **main** dan menambahkan **throws IOException** ke

deklarasinya. Ini berarti bahwa metode `main` akan menangani pengecualian `IOException` dengan melemparkannya ke atas (biasanya ke metode pemanggil). Ini diperlukan karena kita menggunakan kelas-kelas yang terkait dengan IO, yang dapat memunculkan pengecualian IO saat berinteraksi dengan input/output.

3. `char cc; int bil;` Mendeklarasikan variabel `cc` sebagai tipe data `char` dan `bil` sebagai tipe data `int`.
4. `InputStreamReader isr = new InputStreamReader(System.in);` Membuat objek `InputStreamReader` yang akan digunakan untuk membaca input dari `System.in`.
5. `BufferedReader dataIn = new BufferedReader(isr);` Membuat objek `BufferedReader` dengan objek `InputStreamReader` sebagai parameternya. Ini digunakan untuk membaca input dari pengguna melalui `System.in`.
6. `cc = dataIn.readLine().charAt(0);` Membaca satu karakter dari input yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `cc`. `readLine()` membaca sebuah baris input, sedangkan `charAt(0)` mengambil karakter pertama dari baris tersebut.
7. `bil = Integer.parseInt(dataIn.readLine());` Membaca satu bilangan dari input yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `bil`. `parseInt()` digunakan untuk mengonversi string input menjadi bilangan integer.
8. `System.out.print(cc + "\n" + bil + "\n");` Mencetak karakter dan bilangan yang telah dibaca dari pengguna ke konsol.
9. `/*String kar = JOptionPane.showInputDialog("Karakter 1 : ");  
System.out.println(kar);*/` Ini adalah contoh penggunaan `JOptionPane` yang dikomentari. `JOptionPane.showInputDialog()` digunakan untuk menampilkan dialog input kepada pengguna.

10. `/*JOptionPane.showMessageDialog(null,"hello");*/`: Ini adalah contoh penggunaan `JOptionPane` yang dikomentari. `JOptionPane.showMessageDialog()` digunakan untuk menampilkan dialog pesan kepada pengguna.

Program ini memberikan beberapa contoh cara membaca input dari pengguna menggunakan berbagai kelas dalam Java, seperti `BufferedReader`, `InputStreamReader`, dan `JOptionPane`. Dengan ini, program menunjukkan beberapa pendekatan yang berbeda untuk interaksi dengan pengguna dalam aplikasi Java.

## 6. Kode program

```
public class Casting1 {  
  
    public static void main(String[] args) {  
        int a=5, b=6;  
        float d=2.f,e=3.2f;  
        char g='5';  
        double k=3.14;  
        System.out.println((float)a);  
        System.out.println((double)b);  
        System.out.println((int)d);  
        System.out.println((double)e);  
        System.out.println((int)g);  
        System.out.println((float)g);  
        System.out.println((double)g);  
        System.out.println((int)k);  
        System.out.println((float)k);  
    }  
}
```

Output :

```
C:\Windows\System32\cmd.e  ×  +  v
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Casting1.java
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14

C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>
```

Penjelasan Program :

Program ini bertujuan untuk mendemonstrasikan penggunaan operator *casting* untuk mengubah tipe data primitif menjadi tipe data primitif yang lain.

Penjelasan singkat tentang casting dan tipe data primitif:

*Casting* adalah proses mengubah tipe data satu menjadi tipe data yang lain. Dalam konteks ini, kita menggunakan *casting* untuk mengubah nilai dari satu tipe data primitif menjadi tipe data primitif yang lain.

Tipe data primitif yang digunakan dalam program ini adalah **int**, **float**, **char**, dan **double**. **int** adalah tipe data untuk bilangan bulat, **float** untuk bilangan pecahan dengan presisi rendah, **char** untuk karakter, dan **double** untuk bilangan pecahan dengan presisi ganda.

Berikut adalah penjelasan setiap baris dalam program:

1. **int a=5, b=6;**: Mendeklarasikan variabel **a** dan **b** dengan tipe data **int** dan memberikan nilai masing-masing 5 dan 6.
2. **float d=2.f,e=3.2f;**: Mendeklarasikan variabel **d** dan **e** dengan tipe data **float** dan memberikan nilai masing-masing 2.0 dan 3.2.
3. **char g='5';**: Mendeklarasikan variabel **g** dengan tipe data **char** dan memberikan nilai karakter '5'.
4. **double k=3.14;**: Mendeklarasikan variabel **k** dengan tipe data **double** dan memberikan nilai 3.14.

5. **System.out.println((float)a);**: Mencetak nilai variabel **a** setelah di-*cast* menjadi tipe data **float**. Outputnya adalah 5.0.
6. **System.out.println((double)b);**: Mencetak nilai variabel **b** setelah di-*cast* menjadi tipe data **double**. Outputnya adalah 6.0.
7. **System.out.println((int)d);**: Mencetak nilai variabel **d** setelah di-*cast* menjadi tipe data **int**. Outputnya adalah 2.
8. **System.out.println((double)e);**: Mencetak nilai variabel **e** setelah di-*cast* menjadi tipe data **double**. Outputnya adalah 3.2.
9. **System.out.println((int)g);**: Mencetak nilai variabel **g** setelah di-*cast* menjadi tipe data **int**. Karena nilai karakter '5' diasumsikan sebagai nilai ASCII-nya, maka outputnya adalah 53, yaitu nilai ASCII dari karakter '5'.
10. **System.out.println((float)g);**: Mencetak nilai variabel **g** setelah di-*cast* menjadi tipe data **float**. Karena nilai karakter '5' diasumsikan sebagai nilai ASCII-nya, maka outputnya adalah 53.0.
11. **System.out.println((double)g);**: Mencetak nilai variabel **g** setelah di-*cast* menjadi tipe data **double**. Karena nilai karakter '5' diasumsikan sebagai nilai ASCII-nya, maka outputnya adalah 53.0.
12. **System.out.println((int)k);**: Mencetak nilai variabel **k** setelah di-*cast* menjadi tipe data **int**. Outputnya adalah 3, karena bagian desimal akan diabaikan.
13. **System.out.println((float)k);**: Mencetak nilai variabel **k** setelah di-*cast* menjadi tipe data **float**. Outputnya adalah 3.14, karena tipe data **float** dapat menyimpan bilangan pecahan dengan presisi lebih rendah dibandingkan **double**.

## 7. Kode Program

```
public class Casting2{  
    public static void main(String[] args) {  
  
        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";

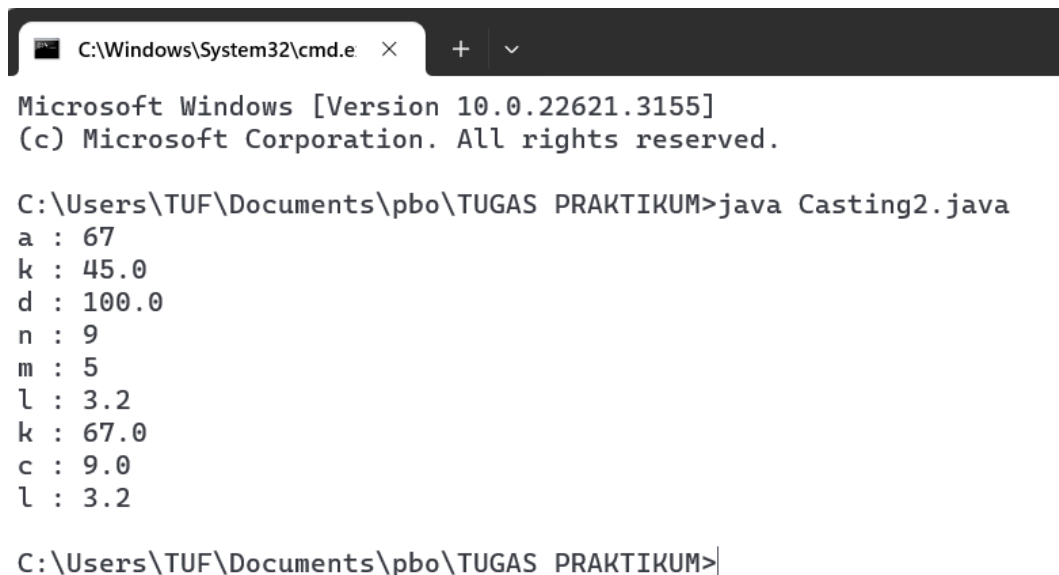
a = Integer.parseInt(n);
k = Double.parseDouble(m);
d = Float.parseFloat(l);
System.out.println("a : " + a + "\nk : " + k + "\nd : " + d);
n = String.valueOf(b);
m = String.valueOf(g);
l = String.valueOf(e);
System.out.println("n : " + n + "\nm : " + m + "\nl : " + l);
k = Double.valueOf(a).intValue();

double c = Integer.valueOf(b).doubleValue();
System.out.println("k : " + k + "\nc : " + c + "\nl : " + l);
}

}

```

Output :



```

C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Casting2.java
a : 67
k : 45.0
d : 100.0
n : 9
m : 5
l : 3.2
k : 67.0
c : 9.0
l : 3.2

C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>

```

Penjelasan Program :

Program ini bertujuan untuk mendemonstrasikan penggunaan metode *parsing* dan konversi antara tipe data primitif dan objek String dalam bahasa pemrograman Java.

Penjelasan tentang metode parsing dan konversi:

*Parsing* adalah proses mengubah tipe data objek String menjadi tipe data primitif tertentu, seperti `int`, `float`, atau `double`.

Konversi adalah proses mengubah tipe data primitif menjadi tipe data objek String, atau sebaliknya.

Berikut adalah penjelasan setiap baris dalam program:

1. `int a=8,b=9;, float d=2.f,e=3.2f;, char g='5';, double k=3.14;;` Mendeklarasikan beberapa variabel dengan tipe data primitif yang berbeda.
2. `String n="67",m="45",l="100";`: Mendeklarasikan tiga variabel dengan tipe data String dan memberikan nilai masing-masing.
3. `a = Integer.parseInt(n);`: Mengubah nilai dari variabel `n` (String) menjadi tipe data `int` menggunakan metode `parseInt()`. Hasilnya akan disimpan dalam variabel `a`.
4. `k = Double.parseDouble(m);`: Mengubah nilai dari variabel `m` (String) menjadi tipe data `double` menggunakan metode `parseDouble()`. Hasilnya akan disimpan dalam variabel `k`.
5. `d = Float.parseFloat(l);`: Mengubah nilai dari variabel `l` (String) menjadi tipe data `float` menggunakan metode `parseFloat()`. Hasilnya akan disimpan dalam variabel `d`.
6. `n = String.valueOf(b);, m = String.valueOf(g);, l = String.valueOf(e);`: Mengonversi nilai dari variabel `b`, `g`, dan `e` menjadi tipe data String menggunakan metode `valueOf()` dari kelas `String`. Hasilnya akan disimpan kembali dalam variabel yang bersangkutan.
7. `k = Double.valueOf(a).intValue();`: Mengubah nilai dari variabel `a` (int) menjadi tipe data `double`, kemudian diubah lagi menjadi tipe data `int` menggunakan metode `intValue()`. Hasilnya akan disimpan dalam variabel `k`.
8. `double c = Integer.valueOf(b).doubleValue();`: Mengubah nilai dari variabel `b` (int) menjadi tipe data `double` menggunakan metode `valueOf()`, kemudian diubah lagi menjadi tipe data `double`. Hasilnya akan disimpan dalam variabel `c`.
9. `System.out.println("a : "+a+"\nk : "+k+"\nd : "+d);, System.out.println("n : "+n+"\nm : "+m+"\nl : "+l);, System.out.println("k : "+k+"\nc : "+c+"\nl : "+l);`: Mencetak nilai variabel-variabel tersebut ke konsol.

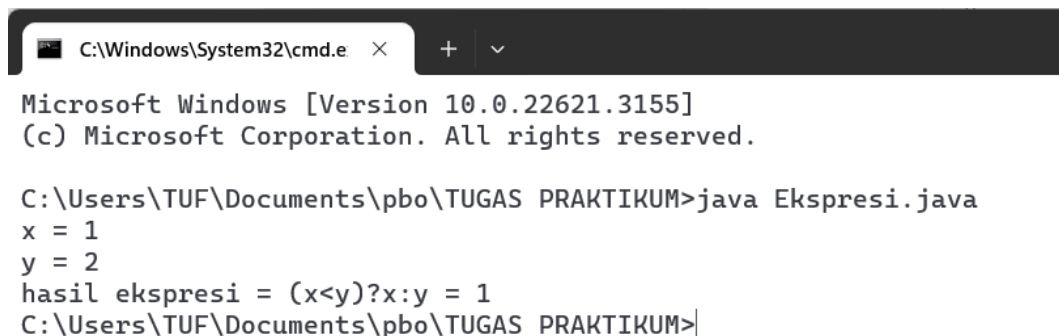


Program ini menunjukkan bagaimana mengubah tipe data antara tipe data primitif dan objek String menggunakan metode *parsing* seperti `parseInt()` dan `parseDouble()`, serta menggunakan konversi antara tipe data primitif dan objek String menggunakan metode `valueOf()`

## 8. Kode Program

```
public class Ekspresi {  
    public static void main(String[] args) {  
  
        int x = 1;  
        int y = 2;  
  
        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));  
  
    }  
  
}
```

Output :



```
C:\Windows\System32\cmd.e  X  +  v  
Microsoft Windows [Version 10.0.22621.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Ekspresi.java  
x = 1  
y = 2  
hasil ekspresi = (x<y)?x:y = 1  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>
```

Penjelasan Program :

Program ini bertujuan untuk mendemonstrasikan penggunaan ekspresi kondisional (ternary operator) dalam bahasa pemrograman Java. Ekspresi kondisional digunakan untuk mengevaluasi kondisi dan menghasilkan nilai berdasarkan kondisi tersebut dalam satu baris kode.

Berikut adalah penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas dalam program:

1. **public class Ekspresi {**: Mendeklarasikan kelas utama bernama **Ekspresi**.
2. **public static void main(String[] args) {**: Mendeklarasikan metode utama (**main**) yang merupakan titik awal eksekusi program. Keyword **public** menunjukkan bahwa metode ini dapat diakses dari luar kelas, **static** menunjukkan bahwa metode ini milik kelas itu sendiri dan dapat dipanggil tanpa membuat objek kelas, **void** menunjukkan bahwa metode ini tidak mengembalikan nilai, dan **String[] args** adalah parameter yang menerima argumen baris perintah ke program.
3. **int x = 1;**, **int y = 2;**: Mendeklarasikan dua variabel **x** dan **y** dengan tipe data **int** dan memberikan nilai masing-masing 1 dan 2.
4. **System.out.print("x = " + x + "\n");**, **System.out.print("y = " + y + "\n");**: Menggunakan **System.out.print** untuk mencetak nilai dari variabel **x** dan **y** ke konsol.
5. Ekspresi kondisional:

```
System.out. System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x  
: y));
```

- **(x < y) ? x : y**: Ini adalah ekspresi kondisional (ternary operator). Jika kondisi **(x < y)** benar, maka nilai ekspresi ini adalah **x**, jika tidak maka nilainya adalah **y**.
- **System.out.print**: Mencetak hasil ekspresi kondisional ke konsol.

Program ini bertujuan untuk menunjukkan cara menggunakan ekspresi kondisional untuk memilih nilai berdasarkan kondisi tertentu dalam satu baris kode.

## 9. Kode Program

```
public class Ekspresi1 {  
    public static void main(String[] args) {  
  
        int x = 1;  
        int y = 2;  
        float fx;  
        float fy;  
  
        System.out.print("x/y (format integer) = " + x / y);  
        System.out.print("\nx/y (format float) = " + x / y);  
  
        fx = x;  
        fy = y;
```

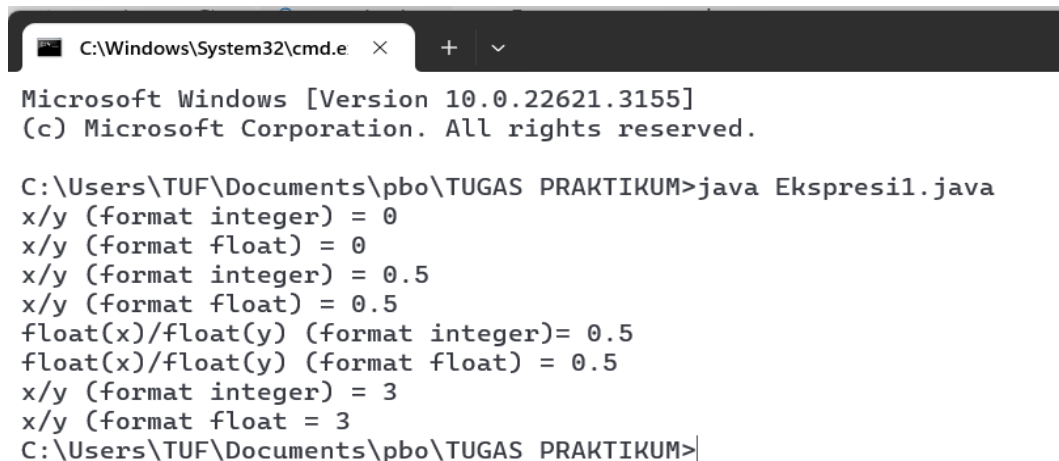
```

System.out.print("\nx/y (format integer) = " + fx / fy);
System.out.print("\nx/y (format float) = " + fx / fy);

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 :



```

Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Ekspresi1.java
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
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>

```

Penjelasan Pogram :

Program ini bertujuan untuk mendemonstrasikan penggunaan pembagian integer dan casting dalam bahasa pemrograman Java.

Berikut adalah penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas dalam program:

1. **public class Ekspresi1** {}: Mendeklarasikan kelas utama bernama **Ekspresi1**.
2. **public static void main(String[] args)** {}: Mendeklarasikan metode utama (**main**) yang merupakan titik awal eksekusi program. Keyword **public** menunjukkan bahwa metode ini dapat diakses dari luar kelas, **static** menunjukkan bahwa metode ini milik kelas itu sendiri dan dapat dipanggil tanpa membuat objek kelas, **void** menunjukkan bahwa metode ini tidak

mengembalikan nilai, dan `String[] args` adalah parameter yang menerima argumen baris perintah ke program.

3. `int x = 1; int y = 2; float fx; float fy;` Mendeklarasikan dua variabel `x` dan `y` dengan tipe data `int` dan memberikan nilai masing-masing 1 dan 2. Selain itu, dua variabel `fx` dan `fy` dengan tipe data `float` dideklarasikan tanpa nilai awal.
4. Ekspresi pembagian:

```
System.out.print ("x/y (format integer) = "+ x/y);
```

```
System.out.print ("\nx/y (format float) = "+ x/y);
```

- Dalam ekspresi pertama, pembagian `x/y` dilakukan menggunakan tipe data `int`, sehingga hanya bagian bulat dari hasil pembagian yang diambil. Ini disebut pembagian integer.
- Dalam ekspresi kedua, pembagian `x/y` juga dilakukan menggunakan tipe data `int`, namun hasilnya dicetak dalam format float. Hal ini dapat menimbulkan kesalahpahaman karena nilai yang dicetak masih dalam format integer.

5. Casting:

```
System.out.print ("\nx/y (format integer) = "+ fx/fy);
```

```
System.out.print ("\nx/y (format float) = "+ fx/fy);
```

```
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);
```

- Untuk menghindari pembagian integer, nilai variabel `x` dan `y` di-*assign* ke variabel `fx` dan `fy` yang bertipe data `float` sebelum melakukan pembagian.
- Melakukan *casting* dengan mengubah tipe data `x` dan `y` menjadi `float` sebelum melakukan pembagian. Ini memastikan bahwa pembagian dilakukan dalam format float.

6. `x = 10; y = 3;` Memberikan nilai baru kepada variabel `x` dan `y`.

7. Ekspresi pembagian dengan nilai baru:

```
System.out.print ("\nx/y (format integer) = "+ x/y);
```

```
System.out.print ("\nx/y (format float) = "+ x/y);
```

- Dalam ekspresi pertama, pembagian `x/y` menggunakan tipe data `int`, sehingga hanya bagian bulat dari hasil pembagian yang diambil. Ini disebut pembagian integer.

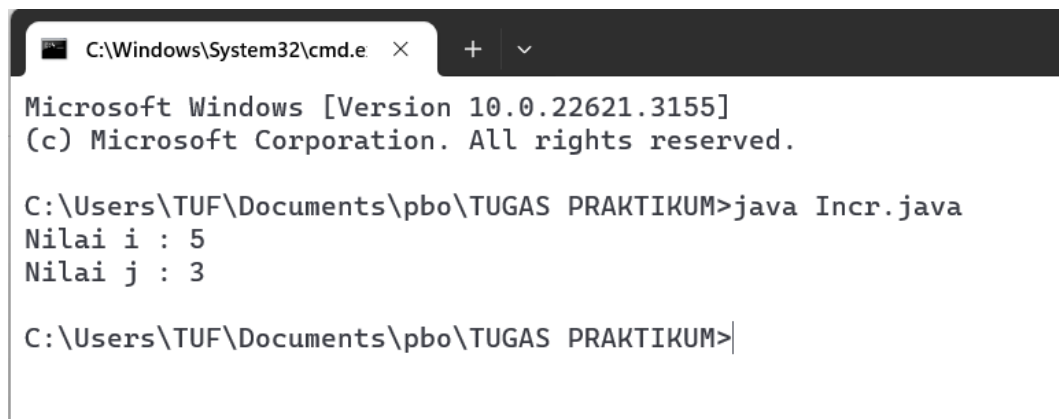
- Dalam ekspresi kedua, pembagian `x/y` juga dilakukan menggunakan tipe data `int`, namun hasilnya dicetak dalam format float. Hal ini dapat menimbulkan kesalahpahaman karena nilai yang dicetak masih dalam format integer.

Tujuan dari program ini adalah untuk menunjukkan perbedaan antara pembagian integer dan float, serta penggunaan `casting` untuk menghindari pembagian integer.

## 10. Kode Program

```
public class Incr {  
    public static void main(String[] args) {  
  
        int i, j;  
  
        i = 3;  
        j = i++;  
        System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j);  
    }  
}
```

Output :



```
C:\Windows\System32\cmd.e  ×  +  ▾  
Microsoft Windows [Version 10.0.22621.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Incr.java  
Nilai i : 5  
Nilai j : 3  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>
```

Penjelasan Program :

Program ini bertujuan untuk mendemonstrasikan berbagai cara untuk mencetak teks ke layar menggunakan perintah `System.out.print` dan `System.out.println` dalam bahasa pemrograman Java.

Berikut adalah penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas dalam program:

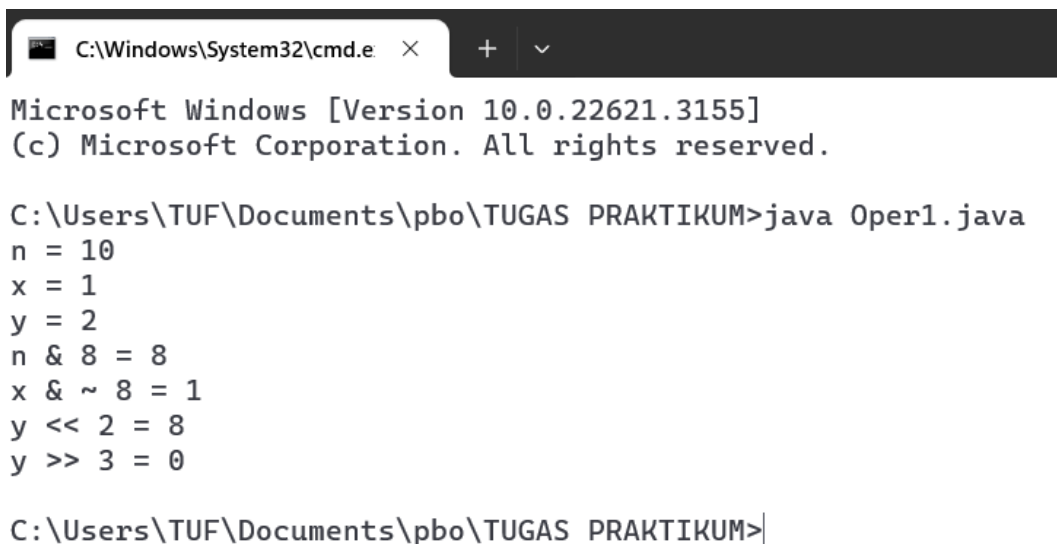
1. **public class PrintHello** {: Mendeklarasikan kelas utama bernama **PrintHello**.
2. **public static void main(String[] args)** {: Mendeklarasikan metode utama (**main**) yang merupakan titik awal eksekusi program. Keyword **public** menunjukkan bahwa metode ini dapat diakses dari luar kelas, **static** menunjukkan bahwa metode ini milik kelas itu sendiri dan dapat dipanggil tanpa membuat objek kelas, **void** menunjukkan bahwa metode ini tidak mengembalikan nilai, dan **String[] args** adalah parameter yang menerima argumen baris perintah ke program.
3. **System.out.print("Hello");**: Mencetak teks "Hello" ke layar tanpa mengganti baris.
4. **System.out.print("\nHello ");**: Mencetak teks "Hello" ke layar dengan mengganti baris setelahnya.
5. **System.out.println("World");**: Mencetak teks "World" ke layar dan mengganti baris setelahnya.
6. **System.out.println("Welcome");**: Mencetak teks "Welcome" ke layar dan mengganti baris setelahnya.

Tujuan dari program ini adalah untuk menunjukkan cara menggunakan perintah **System.out.print** dan **System.out.println** untuk mencetak teks ke layar dalam bahasa Java. Dengan menggunakan berbagai kombinasi dari dua perintah tersebut, program dapat menghasilkan output yang sesuai dengan kebutuhan.

## 11. Kode Program

```
public class Oper1 {  
  
    public static void main(String[] args) {  
  
        int n = 10;  
        int x = 1;  
        int y = 2;  
  
        System.out.println("n = " + n);  
        System.out.println("x = " + x);  
        System.out.println("y = " + y);  
        System.out.println("n & 8 = " + (n & 8));  
        System.out.println("x & ~ 8 = " + (x & ~8));  
        System.out.println("y << 2 = " + (y << 2));  
        System.out.println("y >> 3 = " + (y >> 3));  
    }  
}
```

Output :



```
C:\Windows\System32\cmd.e  X + v  
Microsoft Windows [Version 10.0.22621.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Oper1.java  
n = 10  
x = 1  
y = 2  
n & 8 = 8  
x & ~ 8 = 1  
y << 2 = 8  
y >> 3 = 0  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>
```

Penjelasan Program :

Program tersebut memiliki tujuan untuk mendemonstrasikan penggunaan operator bitwise pada variabel-variabel integer dalam bahasa pemrograman Java. Operator bitwise memungkinkan manipulasi bit individu dalam suatu nilai. Berikut

penjelasan mengenai penggunaan keyword dari variabel/tipe data/kelas dalam program:

1. **public class Oper1** {: Mendefinisikan sebuah kelas dengan nama **Oper1**.
2. **public static void main(String[] args)** {: Mendefinisikan sebuah metode utama (**main**) yang akan dieksekusi saat program berjalan. Kata kunci **public** menandakan bahwa metode tersebut dapat diakses dari luar kelas, **static** menandakan bahwa metode ini milik kelas itu sendiri dan dapat dipanggil tanpa membuat objek kelas, **void** menandakan bahwa metode ini tidak mengembalikan nilai, dan **String[] args** adalah parameter yang menerima argumen baris perintah ke program.
3. **Deklarasi Variabel:**  
**int n = 10**;: Mendeklarasikan variabel **n** bertipe integer dengan nilai awal 10.  
**int x = 1**;: Mendeklarasikan variabel **x** bertipe integer dengan nilai awal 1.  
**int y = 2**;: Mendeklarasikan variabel **y** bertipe integer dengan nilai awal 2.
4. **Pencetakan Nilai Variabel:**  
**System.out.println("n = " + n)**;: Mencetak nilai dari variabel **n**.  
**System.out.println("x = " + x)**;: Mencetak nilai dari variabel **x**.  
**System.out.println("y = " + y)**;: Mencetak nilai dari variabel **y**.
5. **Operasi Bitwise:**  
**System.out.println("n & 8 = " + (n & 8))**;: Melakukan operasi bitwise AND antara nilai variabel **n** dengan nilai 8.  
**System.out.println("x & ~8 = " + (x & ~8))**;: Melakukan operasi bitwise AND antara nilai variabel **x** dengan komplement dari nilai 8 (**~8**).  
**System.out.println("y << 2 = " + (y << 2))**;: Melakukan operasi bitwise shift left pada nilai variabel **y** sebanyak 2 bit.  
**System.out.println("y >> 3 = " + (y >> 3))**;: Melakukan operasi bitwise shift right pada nilai variabel **y** sebanyak 3 bit.

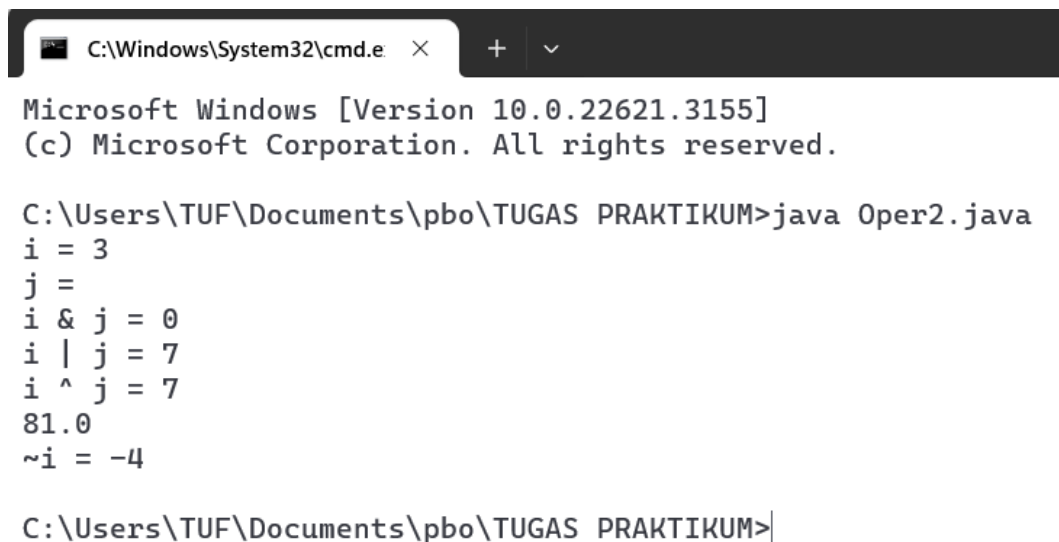
Keyword yang digunakan dari variabel/tipe data/kelas dalam program tersebut antara lain **int** untuk tipe data integer, **System.out.println** untuk mencetak output ke konsol, serta **public class** dan **public static void main** untuk mendefinisikan kelas dan metode utama dalam program.



## 12. Kode Program

```
public class Oper2 {  
    public static void main(String[] args) {  
  
        char i, j;  
  
        i = 3;  
        j = 4;  
        System.out.println("i = " + (int) i);  
        System.out.println("j = " + j);  
        System.out.println("i & j = " + (i & j));  
  
        System.out.println("i | j = " + (i | j));  
  
        System.out.println("i ^ j = " + (i ^ j));  
  
        System.out.println(Math.pow(i, j));  
  
        System.out.println("~i = " + ~i);  
    }  
}
```

Output :



```
C:\Windows\System32\cmd.e  X  +  v  
Microsoft Windows [Version 10.0.22621.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Oper2.java  
i = 3  
j =  
i & j = 0  
i | j = 7  
i ^ j = 7  
81.0  
~i = -4  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>
```

### Penjelasan Program :

Program tersebut memiliki tujuan untuk mendemonstrasikan penggunaan operator bit-wise dan beberapa operasi matematika pada variabel bertipe char dalam bahasa pemrograman Java.

Penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas:

1. **public class Oper2** {: Mendeklarasikan kelas utama bernama Oper2.
2. **public static void main(String[] args)** {: Mendeklarasikan metode main yang merupakan titik awal eksekusi program. Keyword **public** menunjukkan bahwa metode ini dapat diakses dari luar kelas, **static** menunjukkan bahwa metode ini milik kelas itu sendiri dan dapat dipanggil tanpa membuat objek kelas, **void** menunjukkan bahwa metode ini tidak mengembalikan nilai. **String[] args** adalah parameter yang diterima oleh metode main, yang digunakan untuk meneruskan argumen baris perintah ke program.
3. **char i, j**;; Mendeklarasikan dua variabel i dan j dengan tipe data char.
4. **i = 3; j = 4**;; Memberikan nilai 3 kepada variabel i dan nilai 4 kepada variabel j.
5. **System.out.println("i = "+ (int) i)**;; Mencetak nilai variabel i ke konsol dengan konversi ke tipe int menggunakan **(int)** karena variabel i bertipe char.
6. **System.out.println("j = "+ j)**;; Mencetak nilai variabel j ke konsol.
7. **System.out.println("i & j = "+ (i & j))**;; Melakukan operasi bitwise AND (**&**) antara nilai i dan j, kemudian mencetak hasilnya ke konsol.
8. **System.out.println("i | j = "+ (i | j))**;; Melakukan operasi bitwise OR (**|**) antara nilai i dan j, kemudian mencetak hasilnya ke konsol.
9. **System.out.println("i ^ j = "+ (i ^ j))**;; Melakukan operasi bitwise XOR (**^**) antara nilai i dan j, kemudian mencetak hasilnya ke konsol.
10. **System.out.println(Math.pow(i, j))**;; Menggunakan method **pow(a, b)** dari kelas **Math** untuk melakukan operasi pemangkatan, namun perlu dicatat bahwa nilai variabel i dan j diubah menjadi tipe double.

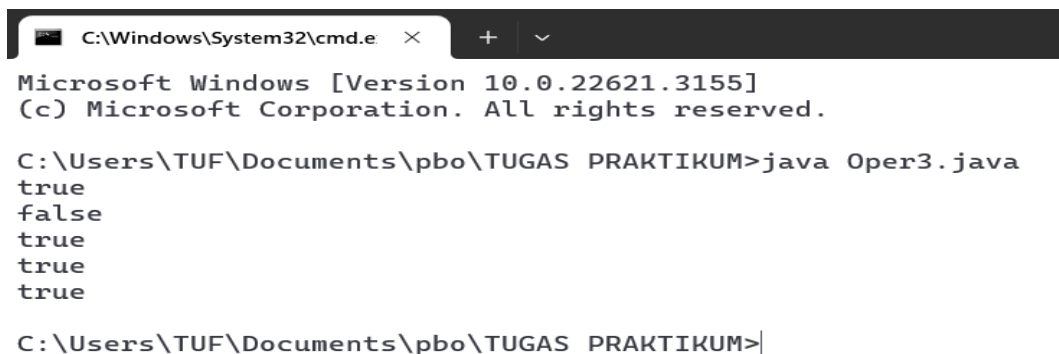
11. `System.out.println("~i = "+ ~i);`: Melakukan operasi bitwise NOT (~) pada nilai i, kemudian mencetak hasilnya ke konsol.

Tujuan dari program ini adalah untuk menunjukkan cara melakukan operasi bitwise dan beberapa operasi matematika pada variabel bertipe char dalam bahasa pemrograman Java.

### 13. Kode Program

```
public class Oper3 {  
    public static void main(String[] args) {  
  
        if (true && true) {  
            System.out.println(true && true);  
        }  
        if (true & true) {  
            System.out.println(true & false);  
        }  
        if (true) {  
            System.out.println(true);  
        }  
        if (true || true) {  
            System.out.println(true);  
        }  
        if (true | false) {  
            System.out.println(true | false);  
        }  
    }  
}
```

Output :



```
C:\Windows\System32\cmd.e  x  +  v  
Microsoft Windows [Version 10.0.22621.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Oper3.java  
true  
false  
true  
true  
true  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>
```

Penjelasan Program :

Program tersebut memiliki tujuan untuk mendemonstrasikan perbedaan antara operator logika `&&` (AND logika) dan `&` (AND bit-wise), serta antara operator logika `||` (OR logika) dan `|` (OR bit-wise) dalam bahasa pemrograman Java.

Penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas:

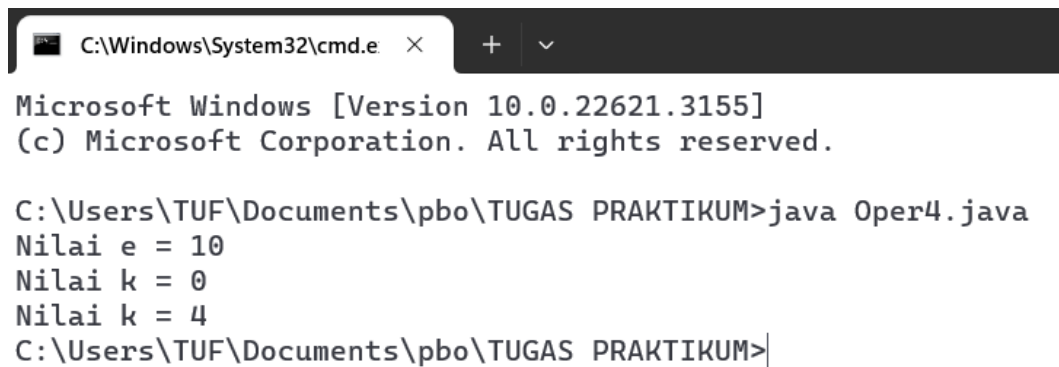
1. `public class Oper3 {`: Mendeklarasikan kelas utama bernama Oper3.
2. `public static void main(String[] args) {`: Mendeklarasikan metode main yang merupakan titik awal eksekusi program. Keyword `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas, `static` menunjukkan bahwa metode ini milik kelas itu sendiri dan dapat dipanggil tanpa membuat objek kelas, `void` menunjukkan bahwa metode ini tidak mengembalikan nilai. `String[] args` adalah parameter yang diterima oleh metode main, yang digunakan untuk meneruskan argumen baris perintah ke program.
3. `if (true && true) { System.out.println(true && true); }`: Melakukan pengecekan kondisi dengan operator logika `&&` (AND logika). Kondisi ini akan dievaluasi sebagai `true` karena kedua operandnya bernilai `true`, kemudian mencetak hasilnya ke konsol.
4. `if (true & true) { System.out.println(true & false); }`: Melakukan pengecekan kondisi dengan operator bit-wise `&` (AND bit-wise). Kondisi ini juga akan dievaluasi sebagai `true` karena kedua operandnya bernilai `true`, kemudian mencetak hasilnya ke konsol.
5. `if (true) { System.out.println(true); }`: Melakukan pengecekan kondisi yang selalu bernilai `true`, kemudian mencetak `true` ke konsol.
6. `if (true || true) { System.out.println(true); }`: Melakukan pengecekan kondisi dengan operator logika `||` (OR logika). Kondisi ini akan dievaluasi sebagai `true` karena salah satu operandnya bernilai `true`, kemudian mencetak hasilnya ke konsol.
7. `if (true | false) { System.out.println(true | false); }`: Melakukan pengecekan kondisi dengan operator bit-wise `|` (OR bit-wise). Kondisi ini juga akan dievaluasi sebagai `true` karena salah satu operandnya bernilai `true`, kemudian mencetak hasilnya ke konsol.

Tujuan dari program ini adalah untuk menunjukkan perbedaan antara operator logika dan operator bit-wise dalam operasi AND (`&&` vs `&`) serta OR (`||` vs `|`) dalam bahasa pemrograman Java.

#### 14. Kode Program

```
public class Oper4 {  
    public static void main(String[] args) {  
  
        int i = 0;  
        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 :



```
C:\Windows\System32\cmd.e  ×  +  ▾  
  
Microsoft Windows [Version 10.0.22621.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Oper4.java  
Nilai e = 10  
Nilai k = 0  
Nilai k = 4  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>
```

Penjelasan Program :

Program tersebut memiliki tujuan untuk mendemonstrasikan penggunaan operator ternary (?:) dalam bahasa pemrograman Java.

Penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas:

1. **public class Oper4** {: Mendeklarasikan kelas utama bernama Oper4.
2. **public static void main(String[] args)** {: Mendeklarasikan metode main sebagai titik awal eksekusi program. **public** menunjukkan bahwa metode ini dapat diakses dari luar kelas, **static** menunjukkan bahwa metode ini milik kelas itu sendiri dan tidak memerlukan pembuatan objek kelas, **void** menunjukkan bahwa metode ini tidak mengembalikan nilai, dan **String[] args** adalah parameter yang menerima argumen baris perintah ke program.
3. **int i = 0; int j = 0**;: Mendeklarasikan dua variabel **i** dan **j** bertipe data integer dan memberi nilai awal 0.
4. **char c = 8; char d = 10**;: Mendeklarasikan dua variabel **c** dan **d** bertipe data karakter dan memberi nilai awal masing-masing 8 dan 10.
5. **int e = (((int)c > (int)d) ? c: d)**;: Menggunakan operator ternary untuk menentukan nilai **e**. Jika nilai **c** lebih besar dari nilai **d**, maka **e** akan diisi dengan nilai **c**, jika tidak, maka akan diisi dengan nilai **d**.
6. **int k = ((i>j) ? i: j)**;: Menggunakan operator ternary untuk menentukan nilai **k**. Jika nilai **i** lebih besar dari nilai **j**, maka **k** akan diisi dengan nilai **i**, jika tidak, maka akan diisi dengan nilai **j**.
7. **System.out.print ("Nilai e = "+ e)**;: Mencetak nilai **e** ke konsol.
8. **System.out.print ("\nNilai k = "+ k)**;: Mencetak nilai **k** ke konsol.
9. **k = ((i++>j++) ? i: j)**;: Menggunakan operator ternary untuk menentukan nilai **k**, namun sebelumnya variabel **i** dan **j** akan diinkremen setelah evaluasi kondisi. Jika nilai **i** setelah diinkremen lebih besar dari nilai **j** setelah diinkremen, maka **k** akan diisi dengan nilai **i**, jika tidak, maka akan diisi dengan nilai **j**.
10. **System.out.print ("\nNilai k = "+ k)**;: Mencetak nilai **k** yang telah diubah setelah penggunaan operator ternary kedua ke konsol.

Tujuan dari program ini adalah untuk mengilustrasikan penggunaan operator ternary (?:) dalam pemrograman Java untuk mengevaluasi kondisi dan memilih nilai berdasarkan hasil evaluasi tersebut.

## 15. Kode Program

```
public class Oprator {

    public static void main(String[] args) {

        boolean Bool1, Bool2, TF;
        int i, j, hsl;
        float x, y, res;

        System.out.println("Silahkan Baca Teksnya Dan Tambahkan
        Program Di Bawah Ini Untuk Menampilkan Output Program");

        Bool1 = true;
        Bool2 = false;
        TF = Bool1 && Bool2;

        TF = Bool1 || Bool2;
        System.out.println("Bool1 || Bool2 = " + TF);
        TF = !Bool1;
        System.out.println("!Bool1 = " + TF);
        TF = Bool1 ^ Bool2;
        System.out.println("Bool1 ^ Bool2 = " + TF);

        i = 5;
        j = 2;
        hsl = i + j;
        System.out.println("i + j = " + hsl);
        hsl = i - j;
        System.out.println("i - j = " + hsl);
        hsl = i / j;
        System.out.println("i / j = " + hsl);
        hsl = i * j;
        System.out.println("i * j = " + hsl);
        hsl = i / j;
        System.out.println("i / j = " + hsl);
        hsl = i % j;
```

```

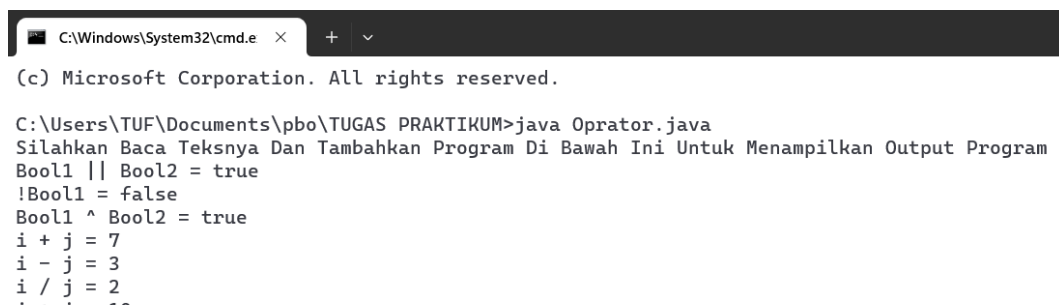
        System.out.println("i % j = " + hsl);
        x = 5;
        y = 5;
        res = x + y;
        System.out.println("x + y = " + res);
        res = x - y;
        System.out.println("x - y = " + res);
        res = x / y;
        System.out.println("x / y = " + res);
        res = x * y;
        System.out.println("x * y = " + res);

        TF = (i == j);
        System.out.println("i == j = " + TF);
        TF = (i != j);
        System.out.println("i != j = " + TF);
        TF = (i < j);
        System.out.println("i < j = " + TF);
        TF = (i > j);
        System.out.println("i > j = " + TF);
        TF = (i <= j);
        System.out.println("i <= j = " + TF);
        TF = (i >= j);
        System.out.println("i >= j = " + TF);

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

```

Output :



```

C:\Windows\System32\cmd.e  X + v
(c) Microsoft Corporation. All rights reserved.

C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java Oprator.java
Silahkan Baca Teksnya Dan Tambahkan Program Di Bawah Ini Untuk Menampilkan Output Program
Bool1 || Bool2 = true
!Bool1 = false
Bool1 ^ Bool2 = true
i + j = 7
i - j = 3
i / j = 2
i % j = 1
TF == TF = true
TF != TF = false
TF < TF = false
TF > TF = false
TF <= TF = true
TF >= TF = true
TF != TF = false
TF < TF = false
TF > TF = false
TF <= TF = true
TF >= TF = true

```



#### Penjelasan Program :

Program ini bertujuan untuk mendemonstrasikan penggunaan berbagai operator dalam bahasa pemrograman Java, seperti operator logika, operator aritmatika, dan operator perbandingan.

Penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas:

1. **public class Operator {**: Mendeklarasikan kelas utama bernama Operator.
2. **public static void main(String[] args) {**: Mendeklarasikan metode main sebagai titik awal eksekusi program. Kata kunci **public** menunjukkan bahwa metode ini dapat diakses dari luar kelas, **static** menunjukkan bahwa metode ini milik kelas itu sendiri dan tidak memerlukan pembuatan objek kelas, **void** menunjukkan bahwa metode ini tidak mengembalikan nilai, dan **String[] args** adalah parameter yang diterima oleh metode main, yang digunakan untuk meneruskan argumen baris perintah ke program.
3. Deklarasi variabel-variabel yang akan digunakan dalam program:
  - **boolean Bool1, Bool2, TF;**: Mendeklarasikan tiga variabel boolean untuk menyimpan nilai kebenaran (true atau false).
  - **int i, j, hsl;**: Mendeklarasikan tiga variabel bertipe data integer.
  - **float x, y, res;**: Mendeklarasikan tiga variabel bertipe data float.
4. Penggunaan operator logika (**&&**, **||**, **!**, **^**) dan pencetakan hasilnya ke konsol.

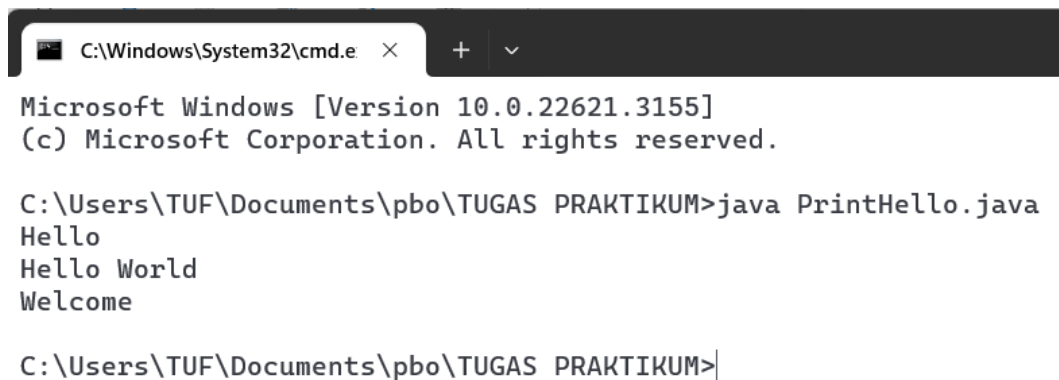
5. Penggunaan operator aritmatika (+, -, \*, /, %) pada bilangan bulat dan pecahan, serta pencetakan hasilnya ke konsol.
6. Penggunaan operator perbandingan (==, !=, <, >, <=, >=) pada bilangan bulat dan pecahan, serta pencetakan hasilnya ke konsol.

Tujuan dari program ini adalah untuk mengilustrasikan penggunaan berbagai operator dalam bahasa pemrograman Java serta menunjukkan hasil operasi dari operator-operator tersebut.

## 16. Kode Program

```
public class PrintHello {  
  
    public static void main(String[] args) {  
  
        System.out.print("Hello");  
  
        System.out.print("\nHello ");  
  
        System.out.println("World");  
  
        System.out.println("Welcome");  
    }  
}
```

Output :



```
C:\Windows\System32\cmd.e  X  +  v  
Microsoft Windows [Version 10.0.22621.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>java PrintHello.java  
Hello  
Hello World  
Welcome  
  
C:\Users\TUF\Documents\pbo\TUGAS PRAKTIKUM>
```

Penjelasan Program :

Program ini bertujuan untuk mencetak beberapa teks ke konsol dengan menggunakan beberapa metode cetak yang berbeda.

Penjelasan tentang keyword yang digunakan dari variabel/tipe data/kelas:

1. **public class PrintHello** {: Mendeklarasikan kelas utama bernama **PrintHello**.
2. **public static void main(String[] args)** {: Mendeklarasikan metode **main** sebagai titik awal eksekusi program. Kata kunci **public** menunjukkan bahwa metode ini dapat diakses dari luar kelas, **static** menunjukkan bahwa metode ini milik kelas itu sendiri dan tidak memerlukan pembuatan objek kelas, **void** menunjukkan bahwa metode ini tidak mengembalikan nilai, dan **String[] args** adalah parameter yang diterima oleh metode **main**, yang digunakan untuk meneruskan argumen baris perintah ke program.

Program ini menggunakan beberapa perintah untuk mencetak teks ke konsol:

1. **System.out.print("Hello");**: Mencetak teks "Hello" ke konsol tanpa mengganti baris.
2. **System.out.print("\nHello ");**: Mencetak teks "Hello" ke konsol dengan mengganti baris setelahnya.
3. **System.out.println("World");**: Mencetak teks "World" ke konsol dengan mengganti baris setelahnya.
4. **System.out.println("Welcome");**: Mencetak teks "Welcome" ke konsol dengan mengganti baris setelahnya.

Tujuan dari program ini adalah untuk menunjukkan cara menggunakan beberapa metode cetak dalam bahasa pemrograman Java untuk mencetak teks ke konsol. Dengan demikian, program ini berfungsi sebagai contoh penggunaan perintah cetak dalam Java