

**LAPORAN**  
**PEMOGRAMAN ORIENTASI OBJEK**



Nama : Muh. Alif Singgih Aksari

Nim : 13020220042

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS MUSLIM INDONESIA**

**MAKASSAR**

**2023**

#### SOURCE CODE 1:

```
D:\latihan java>java Asgdll
f : 20.0
f11: 10.0

D:\latihan java>
```

#### PENJELASAN PROGRAM :

Program ini adalah sebuah program Java sederhana yang memiliki kelas bernama Asgdll. Berikut adalah penjelasan singkat dari kode program ini:

- `public class Asgdll`: Mendefinisikan kelas dengan nama Asgdll. Kelas ini merupakan tempat di mana program Java dimulai, dan method main akan dijalankan pertama kali.
- `public static void main(String[] args)`: Method main adalah titik masuk utama program Java. Semua kode yang ingin dijalankan harus berada di dalam method main. Parameter `String[] args` adalah array string yang dapat digunakan untuk menerima argumen dari baris perintah, tetapi dalam program ini tidak digunakan.
- `float f = 20.0f`;: Mendeklarasikan variabel f dengan tipe data float dan memberikan nilai awal 20.0.
- `double f11`;: Mendeklarasikan variabel f11 dengan tipe data double tanpa memberikan nilai awal.
- `f11 = 10.0f`;: Memberikan nilai 10.0 ke variabel f11.
- `System.out.println("f : " + f + "\nf11: " + f11)`;: Menampilkan nilai variabel f dan f11 ke konsol. `\n` digunakan untuk membuat baris baru dalam output.

Jadi, secara singkat, program ini mendeklarasikan dan menginisialisasi dua variabel float dan double, kemudian menampilkan nilai variabel tersebut ke konsol.

#### SOURCE CODE 2 :

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

D:\latihan java>
```

#### PENJELASAN PROGRAM :

Program ini juga adalah program Java yang sederhana. Berikut adalah penjelasan singkat dari kode program tersebut:

- `public class Asign`: Mendefinisikan kelas dengan nama `Asign`.
- `public static void main(String[] args)`: Method `main` adalah titik masuk utama program Java.
- `int i`;: Mendeklarasikan variabel `i` dengan tipe data `int` tanpa memberikan nilai awal.
- `System.out.print("hello\n");`: Menampilkan teks "hello" ke konsol tanpa membuat baris baru karena menggunakan `System.out.print` bukan `System.out.println`.
- `i = 5`;: Memberikan nilai 5 ke variabel `i`.
- `System.out.println("Ini nilai i : " + i);`: Menampilkan teks "Ini nilai i : " diikuti dengan nilai variabel `i` ke konsol, dan karena menggunakan `System.out.println`, baris baru akan dibuat setelah output ini.

Jadi, secara singkat, program ini mendeklarasikan sebuah variabel `i`, menampilkan "hello" ke konsol, memberikan nilai 5 ke variabel `i`, dan akhirnya menampilkan nilai variabel `i` ke konsol.

### SOURCE CODE 3 :

```
D:\latihan java>javac ASIGNi.java

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

### PENJELASAN PROGRAM

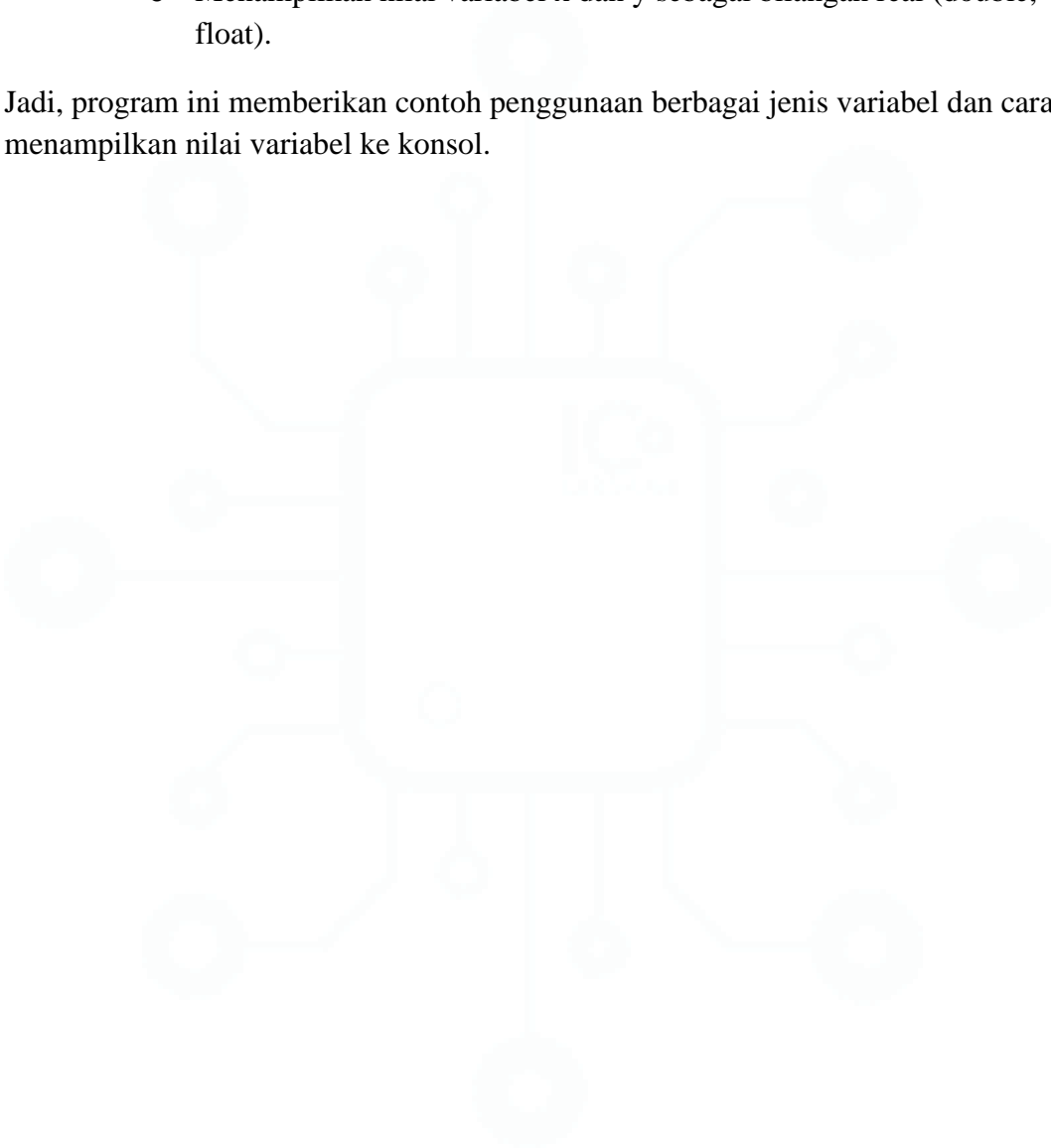
Program ini adalah contoh sederhana dalam mendefinisikan dan menggunakan berbagai jenis variabel dalam bahasa pemrograman Java. Berikut adalah penjelasan singkat dari kode program tersebut:

1. `public class ASIGNi`: Mendefinisikan kelas dengan nama ASIGNi.
2. `public static void main(String[] args)`: Method main adalah titik masuk utama program Java.
3. Deklarasi variabel-variabel:
  - `short ks = 1;` Mendeklarasikan variabel ks dengan tipe data short dan memberikan nilai awal 1.
  - `int ki = 1;` Mendeklarasikan variabel ki dengan tipe data int dan memberikan nilai awal 1.
  - `long kl = 10000;` Mendeklarasikan variabel kl dengan tipe data long dan memberikan nilai awal 10000.
  - `char c = 65;` Mendeklarasikan variabel c dengan tipe data char dan memberikan nilai awal 65 (menggunakan representasi integer untuk karakter).
  - `char c1 = 'Z';` Mendeklarasikan variabel c1 dengan tipe data char dan memberikan nilai awal 'Z'.
  - `double x = 50.2f;` Mendeklarasikan variabel x dengan tipe data double dan memberikan nilai awal 50.2f.
  - `float y = 50.2f;` Mendeklarasikan variabel y dengan tipe data float dan memberikan nilai awal 50.2f.

4. Menampilkan nilai variabel-variabel ke konsol:

- Menampilkan nilai variabel karakter c dan c1 sebagai karakter.
- Menampilkan nilai variabel karakter c dan c1 sebagai integer (representasi karakter dalam ASCII).
- Menampilkan nilai variabel ks, ki, dan kl sebagai bilangan integer (short, int, long).
- Menampilkan nilai variabel x dan y sebagai bilangan real (double, float).

Jadi, program ini memberikan contoh penggunaan berbagai jenis variabel dan cara menampilkan nilai variabel ke konsol.



#### SOURCE CODE 4 :

```
D:\latihan java>javac BacaData.java

D:\latihan java>java BacaData
Contoh membaca dan menulis, ketik nilai integer:
5
Nilai yang dibaca : 5
```

#### PENJELASAN PROGRAM :

Program ini adalah contoh sederhana dalam membaca input dari pengguna menggunakan kelas Scanner pada Java. Berikut adalah penjelasan singkat dari kode program tersebut:

1. `import java.util.Scanner;`: Mengimpor kelas Scanner dari paket `java.util` untuk digunakan dalam program.
2. `public class BacaData:` Mendefinisikan kelas dengan nama `BacaData`.
3. `public static void main(String[] args):` Method `main` adalah titik masuk utama program Java.
4. `int a;`: Mendeklarasikan variabel `a` dengan tipe data `int` untuk menyimpan nilai yang akan diinput.
5. `Scanner masukan;`: Mendeklarasikan variabel `masukan` dengan tipe data `Scanner`, yang akan digunakan untuk membaca input dari pengguna.
6. `System.out.print("Contoh membaca dan menulis, ketik nilai integer: \n");`: Menampilkan teks ke konsol untuk memberikan petunjuk kepada pengguna untuk memasukkan nilai integer.
7. `masukan = new Scanner(System.in);`: Membuat objek `Scanner` baru untuk membaca input dari `System.in` (input dari pengguna).
8. `a = masukan.nextInt();`: Membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel `a`.
9. `System.out.print("Nilai yang dibaca : " + a);`: Menampilkan nilai yang telah dibaca dari pengguna ke konsol.

Jadi, program ini meminta pengguna untuk memasukkan nilai integer, membaca input tersebut menggunakan `Scanner`, dan kemudian menampilkan nilai yang telah dibaca ke konsol.

#### SOURCE CODE 5 :

```
D:\latihan java>javac Bacakar.java

D:\latihan java>java Bacakar
hello
baca 1 karakter : alif
baca 1 bilangan : 15
a
15
bye
```

#### PENJELASAN PROGRAM :

Program ini adalah contoh penggunaan kelas `BufferedReader` untuk membaca input dari pengguna, serta mendemonstrasikan beberapa cara untuk membaca karakter dan bilangan. Berikut adalah penjelasan singkat dari kode program tersebut:

1. `import java.io.BufferedReader;`: Mengimpor kelas `BufferedReader` dari paket `java.io` untuk membaca input karakter.
2. `import java.io.IOException;`: Mengimpor kelas `IOException` dari paket `java.io` karena metode `readLine()` dapat menghasilkan `IOException`.
3. `import java.io.InputStreamReader;`: Mengimpor kelas `InputStreamReader` dari paket `java.io` untuk mengonversi `System.in` menjadi stream yang dapat dibaca.
4. `public static void main(String[] args) throws IOException:` Method `main` menyatakan bahwa program dapat melemparkan `IOException`.
5. Deklarasi variabel:
  - `char cc;`: Mendeklarasikan variabel `cc` untuk menyimpan karakter yang akan diinput.
  - `int bil;`: Mendeklarasikan variabel `bil` untuk menyimpan bilangan integer yang akan diinput.
  - `InputStreamReader isr = new InputStreamReader(System.in);`: Membuat objek `InputStreamReader` untuk membaca input dari `System.in`.
  - `BufferedReader dataIn = new BufferedReader(isr);`: Membuat objek `BufferedReader` untuk membaca input dari `InputStreamReader`.

6. `System.out.print("hello\n");`:: Menampilkan teks "hello" ke konsol.
7. `System.out.print("baca 1 karakter : ");`:: Menampilkan teks untuk memberi petunjuk kepada pengguna.
8. `cc = dataIn.readLine().charAt(0);`:: Membaca satu karakter yang dimasukkan oleh pengguna menggunakan `BufferedReader` dan menyimpannya dalam variabel `cc`.
9. `System.out.print("baca 1 bilangan : ");`:: Menampilkan teks untuk memberi petunjuk kepada pengguna.
10. `bil = Integer.parseInt(dataIn.readLine());`:: Membaca satu bilangan integer yang dimasukkan oleh pengguna menggunakan `BufferedReader` dan mengonversinya ke dalam tipe data integer.
11. Menampilkan nilai yang telah dibaca ke konsol:
  - `System.out.print(cc + "\n" + bil + "\n");`
12. `System.out.print("bye \n");`:: Menampilkan teks "bye" ke konsol.

Program ini juga memiliki beberapa bagian yang di-comment (`//`), yang merupakan bagian yang di-nonaktifkan dan hanya sebagai contoh cara menggunakan kelas `JOptionPane` untuk membaca karakter dan menampilkan pesan dialog. Bagian ini perlu diaktifkan kembali dengan menghapus tanda komentar jika Anda ingin menggunakannya.

Program juga mencatat untuk menambahkan program membaca/input data menggunakan `Scanner`, `Console`, dan `JOptionPane`. Namun, implementasinya belum diberikan dalam kode yang disediakan.



#### SOURCE CODE 6 :

```
D:\latihan java>javac Casting1.java

D:\latihan java>java Casting1
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14
```

#### PENJELASAN PROGRAM :

Program ini adalah contoh penggunaan operator casting untuk mengubah tipe data primitif dalam bahasa pemrograman Java. Berikut adalah penjelasan singkat dari kode program tersebut:

1. `public class Casting1`: Mendefinisikan kelas dengan nama `Casting1`.
2. `public static void main(String[] args)`: Method `main` adalah titik masuk utama program Java.
3. Mendeklarasikan beberapa variabel dengan berbagai tipe data:
  - `int a=5, b=6`:: Mendeklarasikan dua variabel bertipe `int` dengan nilai awal 5 dan 6.
  - `float d=2.f, e=3.2f`:: Mendeklarasikan dua variabel bertipe `float` dengan nilai awal 2.0 dan 3.2.
  - `char g='5'`:: Mendeklarasikan variabel bertipe `char` dengan nilai awal '5'.
  - `double k=3.14`:: Mendeklarasikan variabel bertipe `double` dengan nilai awal 3.14.
4. Melakukan operasi casting dan menampilkan hasilnya ke konsol:
  - `(float)a`: Mengubah variabel `a` (`int`) menjadi `float` dan mencetak hasilnya.
  - `(double)b`: Mengubah variabel `b` (`int`) menjadi `double` dan mencetak hasilnya.
  - `(int)d`: Mengubah variabel `d` (`float`) menjadi `int` dan mencetak hasilnya.

- (double)e: Mengubah variabel e (float) menjadi double dan mencetak hasilnya.
- (int)g: Mengubah variabel g (char) menjadi int (nilai ASCII) dan mencetak hasilnya.
- (float)g: Mengubah variabel g (char) menjadi float (nilai ASCII) dan mencetak hasilnya.
- (double)g: Mengubah variabel g (char) menjadi double (nilai ASCII) dan mencetak hasilnya.
- (int)k: Mengubah variabel k (double) menjadi int dan mencetak hasilnya.
- (float)k: Mengubah variabel k (double) menjadi float dan mencetak hasilnya.

Program ini memberikan contoh bagaimana operator casting digunakan untuk mengubah tipe data primitif satu menjadi tipe data primitif lainnya.

#### SOURCE CODE 7 :

```
D:\latihan java>javac Casting2.java

D:\latihan java>java Casting2
a : 67
k : 45.0
d : 100.0
n : 9
m : 5
l : 3.2
k : 67.0
c : 9.0
l : 3.2
```

#### PENJELASAN PROGRAM :

Program ini adalah contoh penggunaan casting pada tipe data primitif dan tipe data kelas (class) di Java. Berikut adalah penjelasan singkat dari kode program tersebut:

##### 1. Casting pada Tipe Data Primitif:

- `System.out.println((float)a);`: Casting nilai variabel a (int) ke float.
- `System.out.println((double)b);`: Casting nilai variabel b (int) ke double.
- `System.out.println((int)d);`: Casting nilai variabel d (float) ke int.
- `System.out.println((double)e);`: Casting nilai variabel e (float) ke double.
- `System.out.println((int)g);`: Casting nilai variabel g (char) ke int (menggunakan representasi ASCII).
- `System.out.println((float)g);`: Casting nilai variabel g (char) ke float (menggunakan representasi ASCII).
- `System.out.println((double)g);`: Casting nilai variabel g (char) ke double (menggunakan representasi ASCII).
- `System.out.println((int)k);`: Casting nilai variabel k (double) ke int.
- `System.out.println((float)k);`: Casting nilai variabel k (double) ke float.

##### 2. Casting pada Tipe Data Class:

- `a = Integer.parseInt(n);`: Konversi String n ke Integer dan disimpan di variabel a.

- `k = Double.parseDouble(m);`: Konversi String `m` ke Double dan disimpan di variabel `k`.
- `d = Float.parseFloat(l);`: Konversi String `l` ke Float dan disimpan di variabel `d`.
- `n = String.valueOf(b);`: Konversi Integer `b` ke String dan disimpan di variabel `n`.
- `m = String.valueOf(g);`: Konversi Karakter `g` ke String dan disimpan di variabel `m`.
- `l = String.valueOf(e);`: Konversi Float `e` ke String dan disimpan di variabel `l`.
- `k = Double.valueOf(a).intValue();`: Konversi Integer `a` ke Double, kemudian ambil nilai Integer dan disimpan di variabel `k`.
- `double c = Integer.valueOf(b).doubleValue();`: Konversi Integer `b` ke Double dan disimpan di variabel `c`.

Program ini menunjukkan penggunaan berbagai operasi casting untuk mengonversi antara tipe data primitif dan tipe data kelas (String) dalam bahasa Java.

#### SOURCE CODE 8 :

```
D:\latihan java>javac Ekspresi.java

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

#### PENJELASAN PROGRAM :

Program ini merupakan contoh penggunaan operator kondisional (ternary) dalam bahasa Java. Berikut adalah penjelasan singkat dari kode program tersebut:

1. `int x = 1;;` Mendeklarasikan variabel `x` dan memberikan nilai 1.
2. `int y = 2;;` Mendeklarasikan variabel `y` dan memberikan nilai 2.
3. Menampilkan nilai variabel `x` dan `y` ke konsol:

```
System.out.print("x = " + x + "\n");
```

```
System.out.print("y = " + y + "\n")
```

4. Menggunakan operator kondisional (ternary) untuk memilih nilai antara `x` dan `y` berdasarkan kondisi (`x < y`):

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

Jika kondisi (`x < y`) benar, maka hasil ekspresi adalah nilai `x`, jika tidak benar, maka hasil ekspresi adalah nilai `y`.

Jadi, program ini menunjukkan cara menggunakan operator kondisional untuk membuat ekspresi bersyarat yang memilih nilai berdasarkan suatu kondisi. Dalam hal ini, hasilnya akan menampilkan nilai `x` jika `x` kurang dari `y`, dan nilai `y` jika sebaliknya.

#### SOURCE CODE 9 :

```
D:\latihan java>javac Ekspresi1.java

D:\latihan java>java Ekspresi1
x/y (format integer) = 0
x/y (format float) = 0
x/y (format integer) = 0.5
x/y (format float) = 0.5
float(x)/float(y) (format integer)= 0.5
float(x)/float(y) (format float) = 0.5
x/y (format integer) = 3
x/y (format float = 3
```

#### PENJELASAN PROGRAM :

Program tersebut mencoba untuk menjelaskan konsep pembagian dan casting dalam Java. Berikut adalah poin-poin utama dari program tersebut:

1. Pembagian dengan Tipe Data Integer:
  - Ketika dua bilangan integer dibagi menggunakan operator / pada Java, hasilnya juga akan menjadi integer. Ini berarti angka desimal dihapus dan hanya menyisakan bilangan bulat.
  - Contoh: `int x = 1; int y = 2; System.out.print("x/y (format integer) = " + x/y);`
2. Penggunaan Tipe Data Float untuk Hasil Pembagian yang Akurat:
  - Untuk mendapatkan hasil pembagian yang akurat dengan angka desimal, variabel x dan y diubah ke tipe data float sebelum dibagi.
  - Contoh: `float fx = x; float fy = y; System.out.print("\nx/y (format float) = " + fx/fy);`
3. Casting untuk Mendapatkan Hasil Pembagian dalam Tipe Data Float:
  - Alternatifnya, kita dapat menggunakan casting untuk mengubah tipe data bilangan bulat ke float sebelum melakukan pembagian.
  - Contoh: `System.out.print("\nfloat(x)/float(y) (format integer) = " + (float)x/(float)y);`
4. Pembagian dengan Nilai Variabel yang Berubah:

- Setelah mengganti nilai variabel x dan y, program menunjukkan hasil pembagian baru.
- Contoh: `x = 10; y = 3; System.out.print("\nx/y (format integer) = " + x/y);`

Program tersebut memberikan pemahaman tentang cara pembagian bekerja dengan tipe data integer, bagaimana mengonversi variabel ke tipe data float untuk hasil pembagian yang akurat, dan bagaimana menggunakan casting untuk melakukan pembagian dalam tipe data float.



SOURCE CODE 10 :

```
D:\latihan java>javac PrintHello.java

D:\latihan java>java PrintHello
Hello
Hello World
Welcome
```

#### PENJELASAN PROGRAM :

Program ini adalah contoh sederhana dalam bahasa Java yang menunjukkan beberapa cara untuk menampilkan teks di layar menggunakan perintah `System.out.print` dan `System.out.println`. Berikut adalah penjelasan singkat dari kode program tersebut:

1. `System.out.print("Hello");`: Menampilkan teks "Hello" di layar tanpa mengganti baris.
2. `System.out.print("\nHello ");`: Menampilkan teks "Hello" di layar dengan mengganti baris sebelumnya. `\n` digunakan untuk membuat baris baru.
3. `System.out.println("World");`: Menampilkan teks "World" di layar dengan membuat baris baru setelahnya.
4. `System.out.println("Welcome");`: Menampilkan teks "Welcome" di layar dengan membuat baris baru setelahnya.

Penting untuk dicatat bahwa perbedaan antara `System.out.print` dan `System.out.println` adalah bahwa yang pertama tidak membuat baris baru setelah menampilkan teks, sedangkan yang kedua membuat baris baru setelah menampilkan teks.



#### SOURCE CODE 11

```
D:\latihan java>javac Incr.java  
  
D:\latihan java>java Incr  
Nilai i : 5  
Nilai j : 3
```

#### PENJELASAN PROGRAM :

Program ini adalah contoh dalam bahasa Java yang menunjukkan efek dari operator ++ (increment). Berikut adalah penjelasan singkat dari kode program tersebut:

1. `int i, j;;` Mendeklarasikan dua variabel integer, `i` dan `j`.
2. `i = 3;;` Memberikan nilai awal 3 ke variabel `i`.
3. `j = i++;;` Mengeksekusi operator ++ yang ditempatkan setelah variabel (`i++`). Nilai `i` disalin ke `j`, lalu nilai `i` ditingkatkan sebesar 1. Dengan kata lain, `j` mendapatkan nilai awal `i` sebelum peningkatan.
4. `System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j);` Menampilkan nilai `i` setelah di-increment secara prefix (ditambah 1 sebelum digunakan) dan nilai `j` ke layar.

Dengan demikian, program ini memberikan ilustrasi tentang bagaimana operator increment (++) bekerja dalam konteks post-increment (`i++`) dan pre-increment (`++i`).

#### SOURCE CODE 12 :

```
D:\latihan java>javac Oper1.java

D:\latihan java>java Oper1
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

#### PENJELASAN PROGRAM :

Program tersebut merupakan contoh penggunaan beberapa operator bitwise pada variabel bertipe data integer. Berikut adalah poin-poin utama dari program tersebut:

1. Inisialisasi Variabel:
  - `int n = 10;` Mendeklarasikan variabel `n` dengan nilai 10 (dalam representasi biner, 1010).
  - `int x = 1;` Mendeklarasikan variabel `x` dengan nilai 1 (dalam representasi biner, 1).
  - `int y = 2;` Mendeklarasikan variabel `y` dengan nilai 2 (dalam representasi biner, 10).
2. Operasi Bitwise AND (&):
  - `n & 8`: Melakukan operasi AND antara `n` dan 8 (dalam representasi biner, 1010 AND 1000). Hasilnya adalah 8 karena hanya bit ketiga yang memiliki nilai 1 pada `n`.
3. Operasi Bitwise AND (&) dan NOT (~):
  - `x & ~8`: Melakukan operasi AND antara `x` dan NOT dari 8 (dalam representasi biner, 1 AND 0111). Hasilnya adalah 0 karena operasi NOT mengubah 8 menjadi 0111.
4. Operasi Bitwise Shift Left (<<):
  - `y << 2`: Melakukan operasi shift left pada nilai `y` sebanyak 2 bit (dalam representasi biner, 10 << 2). Hasilnya adalah 8 karena nilai `y` digeser ke kiri.

#### 5. Operasi Bitwise Shift Right (>>):

- $y \gg 3$ : Melakukan operasi shift right pada nilai  $y$  sebanyak 3 bit (dalam representasi biner,  $10 \gg 3$ ). Hasilnya adalah 0 karena nilai  $y$  digeser ke kanan.

Program ini memberikan contoh bagaimana operator-operator bitwise dapat digunakan untuk memanipulasi bit-bit dari nilai variabel bertipe data integer. Operasi bitwise ini berguna dalam beberapa situasi khusus, terutama dalam pemrograman rendah level atau ketika kita perlu melakukan manipulasi bit secara langsung.



#### SOURCE CODE 13 :

```
D:\latihan java>javac Oper2.java

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

#### PENJELASAN PROGRAM :

Program ini adalah contoh dalam bahasa Java yang menunjukkan penggunaan beberapa operator terhadap operasi bitwise dan operasi relational. Berikut adalah penjelasan singkat dari kode program tersebut:

1. Inisialisasi Variabel:
  - char i, j;: Mendeklarasikan dua variabel karakter, i dan j.
2. Penggunaan Operasi Bitwise:
  - i = 3;: Memberikan nilai 3 ke variabel i (dalam representasi biner, 00000011).
  - j = 4;: Memberikan nilai 4 ke variabel j (dalam representasi biner, 00000100).
3. Operasi Bitwise AND (&):
  - i & j: Melakukan operasi bitwise AND antara i dan j. Hasilnya adalah 0 karena tidak ada bit yang sama di posisi yang sama.
4. Operasi Bitwise OR (|):
  - i | j: Melakukan operasi bitwise OR antara i dan j. Hasilnya adalah 7 (dalam representasi biner, 00000111).
5. Operasi Bitwise XOR (^):
  - i ^ j: Melakukan operasi bitwise XOR antara i dan j. Hasilnya adalah 7 (dalam representasi biner, 00000111).
6. Operasi Pangkat (Math.pow):
  - Math.pow(i, j): Menggunakan method pow dari kelas Math untuk mengangkat i ke j. Catatan: Operator ^ dalam Java bukan

sebagai operator pangkat, dan digunakan method `Math.pow` untuk operasi pangkat.

7. Operasi Bitwise NOT (~):

- ~i: Melakukan operasi bitwise NOT pada i. Hasilnya adalah -4 (dalam representasi biner, 11111100).

Program ini memberikan contoh penggunaan operator bitwise AND, OR, XOR, NOT, dan operasi pangkat pada nilai karakter dalam bahasa Java.



SOURCE CODE 14 :

```
D:\latihan java>javac Oper3.java

D:\latihan java>java Oper3
true
false
true
true
true
```

#### PENJELASAN PROGRAM :

Program ini adalah contoh dalam bahasa Java yang menunjukkan penggunaan operator logika && (AND) dan || (OR), serta operator bitwise & dan | dalam konteks kondisional. Berikut adalah penjelasan singkat dari kode program tersebut:

1. Penggunaan Operator Logika && (AND):
  - `if (true && true) { System.out.println(true && true); }`: Memeriksa apakah kondisi `true && true` benar. Jika benar, maka pernyataan dalam blok `if` akan dieksekusi dan output `"true && true"` akan ditampilkan.
2. Penggunaan Operator Bitwise & dalam Kondisi:
  - `if (true & true) { System.out.println(true & false); }`: Memeriksa apakah kondisi `true & false` benar. Dalam kondisi ini, operator `&` akan melakukan operasi bitwise AND, dan output `"true & false"` akan ditampilkan jika hasilnya benar.
3. Penggunaan Operator Logika || (OR):
  - `if (true || true) { System.out.println(true); }`: Memeriksa apakah kondisi `true || true` benar. Jika benar, maka pernyataan dalam blok `if` akan dieksekusi dan output `"true"` akan ditampilkan.
4. Penggunaan Operator Bitwise | dalam Kondisi:
  - `if (true | false) { System.out.println(true | false); }`: Memeriksa apakah kondisi `true | false` benar. Dalam kondisi ini, operator `|` akan melakukan operasi bitwise OR, dan output `"true | false"` akan ditampilkan jika hasilnya benar.

#### SOURCE CODE 15 :

```
D:\latihan java>javac Oper4.java

D:\latihan java>java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4
```

#### PENJELASAN PROGRAM :

Program tersebut menggunakan operator ternary, yang merupakan cara singkat untuk menyatakan ekspresi kondisional dalam satu baris. Berikut adalah poin-poin utama dari program tersebut:

1. Inisialisasi Variabel:
  - `int i = 0;` Mendeklarasikan variabel `i` dan memberikan nilai awal 0.
  - `int j = 0;` Mendeklarasikan variabel `j` dan memberikan nilai awal 0.
  - `char c = 8;` Mendeklarasikan variabel `c` dan memberikan nilai awal 8.
  - `char d = 10;` Mendeklarasikan variabel `d` dan memberikan nilai awal 10.
2. Operator Ternary pada Variabel `e`:
  - `int e = (((int)c > (int)d) ? c : d);` Operator ternary digunakan untuk membandingkan nilai variabel `c` dan `d`. Jika nilai `c` lebih besar dari `d`, maka nilai `e` akan sama dengan `c`, dan jika tidak, nilai `e` akan sama dengan `d`.
3. Operator Ternary pada Variabel `k`:
  - `int k = ((i > j) ? i : j);` Operator ternary digunakan untuk membandingkan nilai variabel `i` dan `j`. Jika nilai `i` lebih besar dari `j`, maka nilai `k` akan sama dengan `i`, dan jika tidak, nilai `k` akan sama dengan `j`.
4. Penggunaan Operator Ternary dalam Ekspresi:
  - `k = ((i++ > j++) ? i : j);` Operator ternary digunakan dalam ekspresi untuk membandingkan nilai `i` dan `j`. Jika nilai `i` awal lebih besar dari `j` awal, maka nilai `k` akan sama dengan nilai `i` sebelum diinkremen, dan jika tidak, nilai `k` akan sama dengan nilai `j` sebelum diinkremen.

Jadi, program ini menggunakan operator ternary untuk membuat keputusan kondisional dalam satu baris kode.





COURSE CODE 16 :

```
D:\latihan java>javac Oprator.java  
  
D:\latihan java>java Oprator  
Silahkan baca teksnya dan tambahkan perintah program di bawah ini untuk menampilkan output program
```

#### PENJELASAN PROGRAM :

Program ini adalah contoh pengoperasian variabel berjenis dasar dalam bahasa pemrograman Java. Berikut adalah penjelasan singkat dari kode program tersebut:

1. Deklarasi Variabel:

- `boolean Bool1, Bool2, TF;`: Mendeklarasikan tiga variabel boolean, yaitu Bool1, Bool2, dan TF.
- `int i, j, hsl;`: Mendeklarasikan dua variabel integer i dan j, serta variabel integer hsl untuk menyimpan hasil operasi.
- `float x, y, res;`: Mendeklarasikan dua variabel float x dan y, serta variabel float res untuk menyimpan hasil operasi.

2. Operasi Logika Boolean:

- `TF = Bool1 && Bool2;`: Operasi AND logika boolean.
- `TF = Bool1 || Bool2;`: Operasi OR logika boolean.
- `TF = !Bool1;`: Operasi NOT logika boolean.
- `TF = Bool1 ^ Bool2;`: Operasi XOR logika boolean.
- Operasi Numerik pada Variabel Integer (i, j, dan hsl):
  - `hsl = i + j;`: Penjumlahan.
  - `hsl = i - j;`: Pengurangan.
  - `hsl = i / j;`: Pembagian (hasilnya merupakan pembagian bulat).
  - `hsl = i * j;`: Perkalian.
  - `hsl = i % j;`: Sisa pembagian (modulo).

3. Operasi Numerik pada Variabel Float (x, y, dan res):

- `res = x + y;`: Penjumlahan.
- `res = x - y;`: Pengurangan.
- `res = x / y;`: Pembagian.
- `res = x * y;`: Perkalian.

4. Operasi Relasional Numerik:

- `TF = (i == j);`: Sama dengan.
- `TF = (i != j);`: Tidak sama dengan.
- `TF = (i < j);`: Kurang dari.

- $TF = (i > j);$  Lebih dari.
  - $TF = (i \leq j);$  Kurang dari atau sama dengan.
  - $TF = (i \geq j);$  Lebih dari atau sama dengan.
5. Operasi Relasional Numerik pada Variabel Float (x dan y):
- $TF = (x \neq y);$  Tidak sama dengan.
  - $TF = (x < y);$  Kurang dari.
  - $TF = (x > y);$  Lebih dari.
  - $TF = (x \leq y);$  Kurang dari atau sama dengan.
  - $TF = (x \geq y);$  Lebih dari atau sama dengan.

Program ini memberikan contoh penggunaan berbagai operasi pada variabel-variabel dengan tipe data dasar dalam Java, termasuk operasi logika boolean, numerik, dan relasional.