

# **LAPORAN PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Khayla Alifia Indrawati  
Stambuk : 13020230329  
Kelas : B1  
Dosen : Mardiyah Hasnawi, S.Kom., M.T.

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS MUSLIM INDONESIA  
MAKASSAR  
2025**

## 1. Class Asgdll

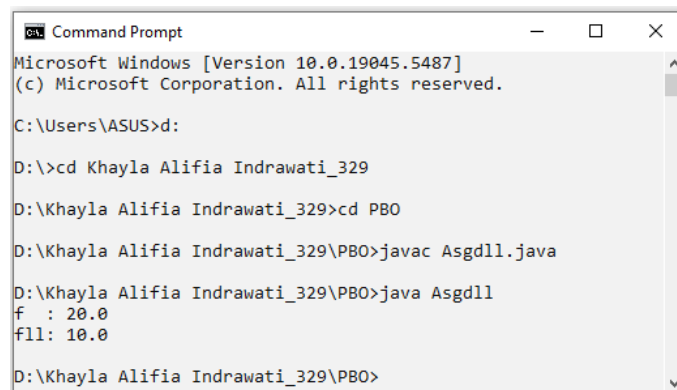
Kode program ini mendefinisikan kelas Asgdll dengan metode main yang akan dieksekusi saat program dijalankan. Di dalam metode main, terdapat dua deklarasi variabel:

- **Variabel f** bertipe data **float** dengan nilai **20.0f**. Huruf f setelah angka menunjukkan bahwa nilai tersebut adalah **float**.
- **Variabel fl** bertipe data **double**, tetapi belum diberi nilai awal.

Selanjutnya, **variabel fl** diberi nilai **10.0f**. Meskipun nilai ini bertipe data **float**, Java secara otomatis akan mengonversinya menjadi **double**, karena tipe **double** lebih besar dari **float**. Baris terakhir mencetak teks ke layar menggunakan **System.out.println()**, dengan tanda + untuk menggabungkan teks dan nilai variabel.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>d:

D:\>cd Khayla Alifia Indrawati_329

D:\Khayla Alifia Indrawati_329>cd PBO

D:\Khayla Alifia Indrawati_329\PBO>javac Asgdll.java

D:\Khayla Alifia Indrawati_329\PBO>java Asgdll
f : 20.0
fl: 10.0

D:\Khayla Alifia Indrawati_329\PBO>
```

## 2. Class Assign

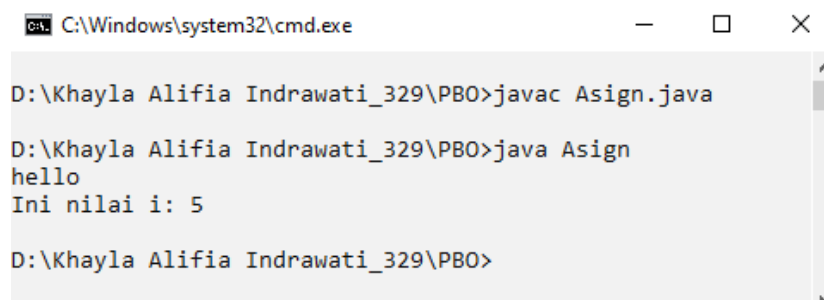
Kode program ini mendefinisikan kelas Assign dengan metode main yang akan dieksekusi saat program dijalankan. Di dalam metode main terdapat deklarasi **variabel i** bertipe data **integer (int)**, tetapi belum memiliki nilai awal.

Selanjutnya, terdapat perintah untuk mencetak teks tanpa berpindah ke baris baru secara otomatis. Perintah pindah baris diberikan dalam teks yang akan dicetak menggunakan **\n**, sehingga teks berikutnya akan berada di baris baru.

Kemudian, **variabel i** yang sebelumnya belum memiliki nilai, diberi nilai **5**. Setelah itu, terdapat perintah untuk mencetak teks menggunakan **System.out.println()**, dengan tanda + untuk menggabungkan teks dan nilai dari **variabel i**.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
C:\Windows\system32\cmd.exe

D:\Khayla Alifia Indrawati_329\PBO>javac Assign.java

D:\Khayla Alifia Indrawati_329\PBO>java Assign
hello
Ini nilai i: 5

D:\Khayla Alifia Indrawati_329\PBO>
```

### 3. Class ASIGNi

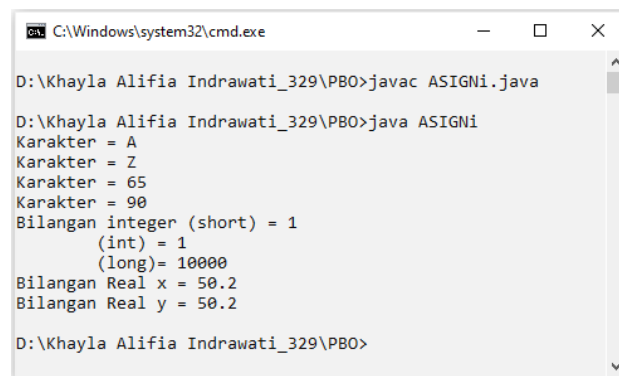
Kode program ini mendefinisikan kelas ASIGNi dengan metode main yang akan dieksekusi saat program dijalankan. Beberapa variabel yang dideklarasikan ke dalam metode main yaitu:

- **Variabel ks** dengan tipe data **short**. **Variabel ki** dengan tipe data **int** standar, yang sudah memiliki nilai awal. **Variabel kl** dengan tipe data **long** diberi nilai **10000**.
- **Variabel c** dengan tipe data **char** diberi nilai **65**. Dalam tabel ASCII, **65** adalah karakter '**A**'. **Variabel c1** dengan tipe data **char**, langsung diinisialisasi dengan karakter '**Z**'.
- **Variabel x** dengan tipe data **double**, tetapi diinisialisasi dengan nilai **50.2**.
- **Variabel y** dengan tipe data **float**, diberi nilai **50.2f**. Nilai diakhiri dengan **f** untuk menandakan bahwa nilai ini adalah **float**, bukan **double**.

Selanjutnya, diberikan perintah untuk mencetak karakter sebagai karakter dan mencetak karakter sebagai integer. Pada perintah untuk mencetak karakter sebagai integer, saya melakukan penambahan **konversi tipe data integer** sebelum **c** dan **c1**, karena tanpa hal tersebut, outputnya tidak akan mencetak karakter sebagai integer, melainkan tetap mencetak karakter sebagai karakter. Kemudian, terdapat perintah untuk mencetak **bilangan bulat** serta **bilangan riil**.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
C:\Windows\system32\cmd.exe
D:\Khayla Alifia Indrawati_329\PBO>javac ASIGNi.java
D:\Khayla Alifia Indrawati_329\PBO>java ASIGNi
Karakter = A
Karakter = Z
Karakter = 65
Karakter = 90
Bilangan integer (short) = 1
          (int) = 1
          (long)= 10000
Bilangan Real x = 50.2
Bilangan Real y = 50.2
D:\Khayla Alifia Indrawati_329\PBO>
```

### 4. Class BacaData

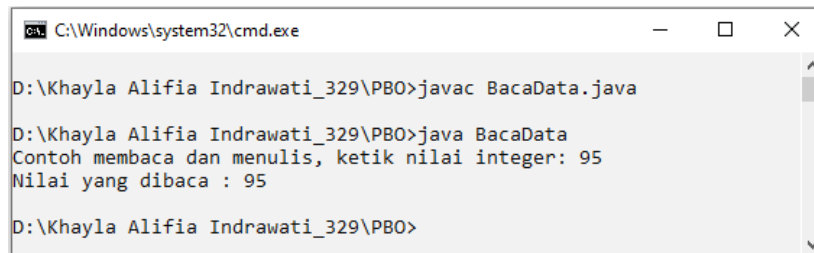
Kode program ini diawali dengan **import java.util.Scanner** yang digunakan untuk membaca input bertipe data integer dari pengguna melalui keyboard. Program ini mendefinisikan kelas BacaData dengan metode main yang akan dieksekusi Ketika program dijalankan. Adapun variabel yang dideklarasikan dalam metode utama main yaitu:

- **Variabel a** bertipe data **integer**, yang akan digunakan untuk menyimpan input dari pengguna.
- Objek **masukan** dari kelas Scanner, tetapi belum diinisialisasi.

Selanjutnya, terdapat perintah untuk menampilkan teks instruksi kepada pengguna untuk memasukkan nilai integer. Kemudian, diberikan inisialisasi objek **Scanner** yang akan membaca input dari keyboard dengan perintah **System.in**. Setelah itu, terdapat **a = masukan.nextInt();** untuk membaca input integer yang dimasukkan pengguna dan menyimpannya ke dalam **variabel a**.

Selanjutnya, terdapat perintah untuk menampilkan teks ke layar diikuti dengan nilai **a** yang telah dimasukkan oleh pengguna. Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
C:\Windows\system32\cmd.exe

D:\Khayla Alifia Indrawati_329\PBO>javac BacaData.java

D:\Khayla Alifia Indrawati_329\PBO>java BacaData
Contoh membaca dan menulis, ketik nilai integer: 95
Nilai yang dibaca : 95

D:\Khayla Alifia Indrawati_329\PBO>
```

## 5. Class Bacakar

Kode program ini mendefinisikan kelas **Bacakar** dengan metode **main** yang akan dieksekusi saat program berjalan. Pada metode **main**, terdapat penggunaan **throws IOException** yang diperlukan karena **BufferedReader** dapat menimbulkan kesalahan saat membaca input. Adapun variabel yang dideklarasikan di dalam metode **main** yaitu:

- **Variabel cc** bertipe data **char**. (menyimpan 1 karakter)
- **Variabel bil** bertipe data **integer**. (menyimpan bilangan bulat dari input)

Pada pendeklarasian, terdapat dua cara untuk membuat objek **BufferedReader**, saya menggunakan cara pertama, deklarasinya pada metode **main** yaitu:

- **InputStreamReader isr = new InputStreamReader(System.in)**
- **BufferedReader dataIn = new BufferedReader(isr)**

Selanjutnya, terdapat perintah untuk menampilkan teks “hello” lalu pindah ke baris baru (**\n**). Kemudian, terdapat perintah lagi untuk menampilkan teks instruksi agar pengguna memasukkan 1 karakter. Setelah perintah menampilkan, terdapat **cc = dataIn.readLine().charAt(0)**, digunakan untuk membaca 1 baris input dari pengguna dan mengambil karakter pertama dari input tersebut lalu menyimpannya ke **variabel cc**.

Baris selanjutnya, terdapat perintah untuk menampilkan instruksi memasukkan bilangan bulat. Setelah perintah menampilkan, terdapat **bil = Integer.parseInt(dataIn.readLine())**, digunakan untuk membaca input sebagai string dan mengonversi string tersebut menjadi bilangan integer.

Selanjutnya, terdapat perintah menampilkan karakter **cc** dan bilangan **bil** yang dimasukkan pengguna serta menggunakan **\n** untuk pindah ke baris baru. Dan terakhir perintah untuk menampilkan teks “bye” lalu pindah ke baris baru.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**

```
C:\Windows\system32\cmd.exe
D:\Khayla Alifia Indrawati_329\PB0>javac Bacakar.java
D:\Khayla Alifia Indrawati_329\PB0>java Bacakar
hello
baca 1 karakter : K
baca 1 bilangan : 18
K
18
bye
D:\Khayla Alifia Indrawati_329\PB0>
```

## 6. Class Casting1

Kode program ini mendefinisikan kelas **Casting1** yang menunjukkan penggunaan casting dalam tipe data primitive. Program ini mendefinisikan kelas **Casting1** dengan metode main yang berisi beberapa variabel:

- **Variabel a** dan **b** bertipe data **integer**, diberi nilai 5 dan 6.
- **Variabel d** dan **e** bertipe data **float**, diberi nilai 2.f dan 3.2f.
- **Variabel g** bertipe data **char**, diberi karakter '5' dan memiliki nilai ASCII 53.
- **Variabel k** bertipe data **double**, diberi nilai 3.14.

Pada bagian **System.out.println()**, program mencetak hasil konversi dari satu tipe ke tipe lain yaitu:

- **(float)a** yang mengubah integer menjadi float, **(double)b** yang mengubah integer menjadi double, **(int)d** yang mengubah float menjadi integer dengan menghilangkan angka decimal.
- **(double)e** yang mengubah float menjadi double, **(int)g** yang mengonversi karakter '5' menjadi kode ASCII-nya, yaitu 53. Dan begitu seterusnya (saling mengkonversi) hingga program selesai.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**

```
C:\Windows\system32\cmd.exe
D:\Khayla Alifia Indrawati_329\PB0>javac Casting1.java
D:\Khayla Alifia Indrawati_329\PB0>java Casting1
5.0
6.0
2
3.200000047683716
53
53.0
53.0
3
3.14
D:\Khayla Alifia Indrawati_329\PB0>
```

## 7. Class Casting2

Kode program ini mendefinisikan kelas **Casting2** yang menunjukkan penggunaan casting dalam tipe data primitif, dengan metode utama main, yang mendeklarasikan beberapa variabel:

- **Variabel a** dan **b** bertipe data **integer**. **Variabel d** dan **e** bertipe data **float**.

- **Variabel g** bertipe data **char**. **Variabel k** bertipe data **double**.
- **Variabel n, m, l** bertipe data **string**.

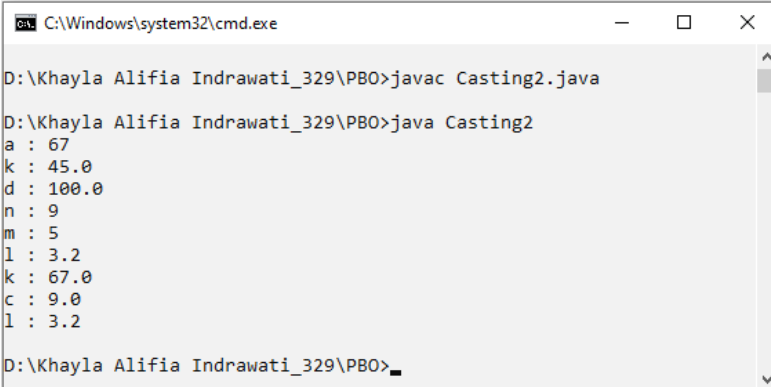
**Konversi pertama** yaitu string ke tipe numerik, dimana **Integer.parseInt(n)**, mengubah string “67” menjadi integer 67 dan menyimpannya pada **variabel a**. **Double.parseDouble(m)**, mengubah string “45” menjadi double 45.0 dan menyimpannya pada **variabel k**. **Float.parseFloat(l)**, mengubah string “100” menjadi float 100.0 dan menyimpannya pada **variabel d**. Kemudian, terdapat perintah untuk menampilkan hasil konversi ke layar.

**Konversi kedua** yaitu tipe numerik dan karakter ke string, dimana **String.valueOf(b)**, mengubah integer 9 menjadi string “9” dan menyimpannya pada **variabel n**. **String.valueOf(g)**, mengubah karakter ‘5’ menjadi string “5” dan menyimpannya pada **variabel m**. **String.valueOf(e)**, mengubah float 3.2 menjadi string “3.2” dan menyimpannya pada **variabel l**. Kemudian, terdapat perintah untuk menampilkan hasil konversi ke layar.

**Konversi ketiga** yaitu tipe integer ke double dan sebaliknya, dimana **Double.valueOf(a).intValue()**, akan mengubah integer 67 menjadi double, lalu Kembali ke integer 67, yang kemudian disimpan pada **variabel k**. Dan **Integer.valueOf(b).doubleValue()**, akan mengubah integer 9 menjadi double 9.0 dan menyimpannya pada **variabel c**. Kemudian, terdapat perintah untuk menampilkan hasil konversi ke layar.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```

C:\Windows\system32\cmd.exe

D:\Khayla Alifia Indrawati_329\PBO>javac Casting2.java

D:\Khayla Alifia Indrawati_329\PBO>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

D:\Khayla Alifia Indrawati_329\PBO>

```

## 8. Class Ekspresi

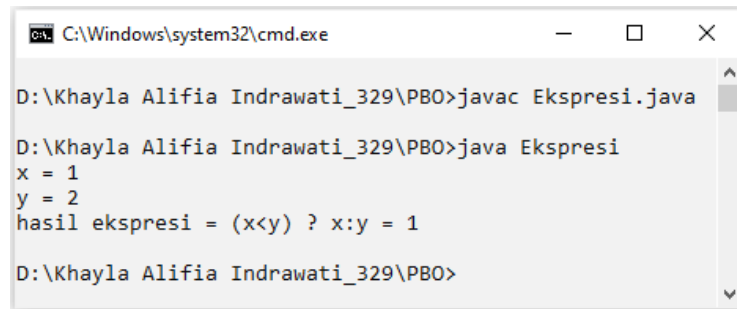
Kode program ini mendefinisikan kelas Ekspresi dengan metode main, yang mendeklarasikan dua variabel: **x** dan **y** dengan tipe data integer, diberi nilai 1 dan 2.

Selanjutnya, program menampilkan nilai awal dari **x** dan **y** menggunakan **System.out.print()**, dengan **\n** untuk pindah ke baris baru. Kemudian, menampilkan hasil operator kondisional.

Cara operator ini bekerja yaitu saat kondisi (**x < y**), yaitu **1 < 2** maka bernilai **true**. Jika kondisi **true**, maka hasilnya adalah **x(1)**, sebaliknya jika kondisi **false**, maka hasilnya adalah **y (2)**. Karena kondisi **1 < 2 true**, maka ekspresi menghasilkan 1, yang kemudian ditampilkan sebagai output.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
C:\Windows\system32\cmd.exe

D:\Khayla Alifia Indrawati_329\PB0>javac Ekspresi.java

D:\Khayla Alifia Indrawati_329\PB0>java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y) ? x:y = 1

D:\Khayla Alifia Indrawati_329\PB0>
```

## 9. Class Ekspresi1

Program ini mendefinisikan kelas Ekspresi1 dengan metode main, yang mendeklarasikan beberapa variabel:

- **Variabel x dan y** bertipe data **integer**, diberi nilai awal 1 dan 2.
- **Variabel fx dan fy** bertipe data **float**.

**Konversi pertama** yaitu pembagian integer, dimana **x/y (format integer)** adalah pembagian dua bilangan bulat akan menghasilkan bilangan bulat, jadi **x/y = 1/2** hasilnya 0 karena pecahan dibuang. Kondisi kedua yaitu **x/y (format float)** hasilnya tetap 0 walaupun “format float” karena tipe datanya masih int.

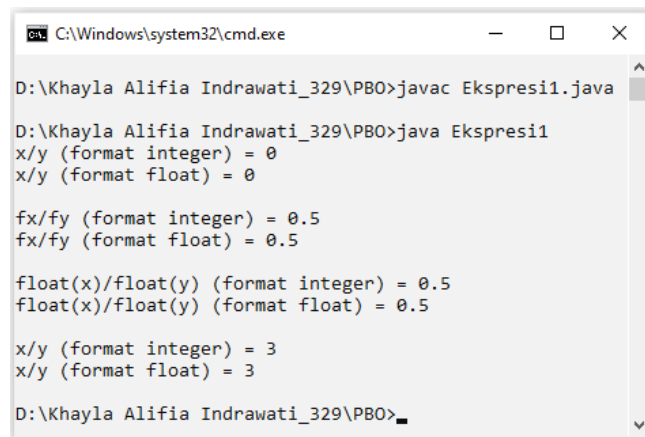
**Konversi kedua** yaitu konversi ke float, dimana fx dan fy diberikan nilai dari x dan y. Karena fx dan fy bertipe data float, maka operasi **fx/fy** menghasilkan nilai desimal 0.5 baik “format integer” maupun “format float”.

**Konversi ketiga** yaitu casting secara langsung, dimana (float)x dan (float)y mengubah x dan y menjadi float sebelum pembagian dilakukan. Hasilnya tetap 0.5, karena operasi dilakukan dengan tipe data float.

**Konversi terakhir** yaitu mengubah nilai x dan y, dimana nilai x diinisialisasi menjadi 10, dan y menjadi 3. **Kondisi pertama** yaitu x/y (format integer), jika diikalkulasikan  $10/3 = 3.333$ , tetapi hasil yang diambil hanya 3, dan pecahan dibuang karena bertipe data integer. **Kondisi kedua** (format float) juga tetap menghasilkan 3, meskipun disebut “format float”, tetapi tipe datanya masih integer.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
C:\Windows\system32\cmd.exe

D:\Khayla Alifia Indrawati_329\PBO>javac Ekspresi1.java

D:\Khayla Alifia Indrawati_329\PBO>java Ekspresi1
x/y (format integer) = 0
x/y (format float) = 0

fx/fy (format integer) = 0.5
fx/fy (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

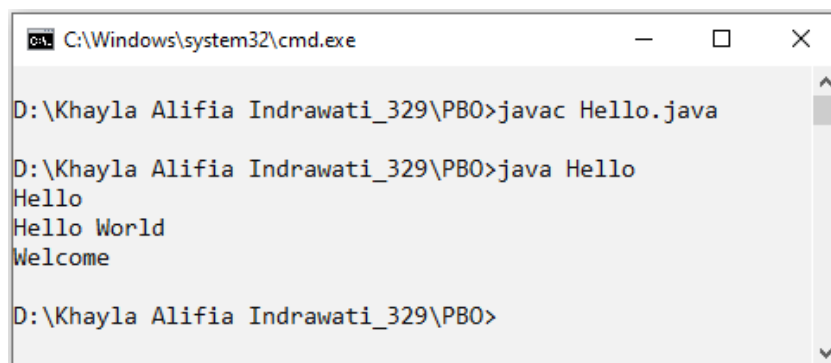
D:\Khayla Alifia Indrawati_329\PBO>
```

## 10. Class Hello

Kode program ini mendefinisikan kelas **Hello** dengan metode main, yang akan dieksekusi saat program dijalankan. Ini adalah program sederhana karena tidak ada deklarasi variabel di dalamnya, melainkan langsung memberikan perintah untuk menampilkan teks ke layar dengan **System.out.print** dan **System.out.println**. Program ini menunjukkan perbedaan antara **print** (yang tidak berpindah baris) dan **println** (yang otomatis berpindah baris setelah mencetak teks).

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
C:\Windows\system32\cmd.exe

D:\Khayla Alifia Indrawati_329\PBO>javac Hello.java

D:\Khayla Alifia Indrawati_329\PBO>java Hello
Hello
Hello World
Welcome

D:\Khayla Alifia Indrawati_329\PBO>
```

## 11. Class Incr

Program ini mendefinisikan kelas **Incr** dengan metode main, yang akan dieksekusi saat program dijalankan. Program ini memiliki deklarasi:

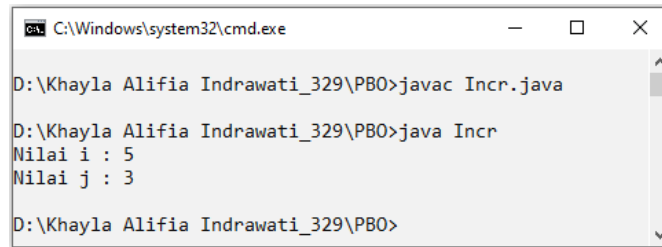
- **Variabel i dan j** dengan tipe data **integer**. **Variabel i** diberi nilai awal **3**.
- **Variabel j** diberikan nilai **i++** (post-increment), artinya nilai **i** disimpan ke **j** terlebih dahulu, baru kemudian **i** bertambah **1**. Setelah baris ini dieksekusi:
  - **j = 3** (karena nilai awal **i** adalah **3**).
  - **i = 4** (karena setelah itu **i++** meningkatkan nilainya).

Selanjutnya, program mencetak hasil ke layar menggunakan **pre-increment** (**++i**), yang menambah **i** sebelum ditampilkan. Karena sebelumnya **i = 4**, maka setelah **++i**, nilainya menjadi **5**. Sementara itu, **j** tetap **3** karena tidak berubah setelah deklarasi sebelumnya.



Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
C:\Windows\system32\cmd.exe
D:\Khayla Alifia Indrawati_329\PB0>javac Incr.java
D:\Khayla Alifia Indrawati_329\PB0>java Incr
Nilai i : 5
Nilai j : 3
D:\Khayla Alifia Indrawati_329\PB0>
```

## 12. Class Oper1

Kode program ini mendefinisikan kelas Oper1 dengan metode utama main yang akan dieksekusi saat program dijalankan, dengan deklarasi:

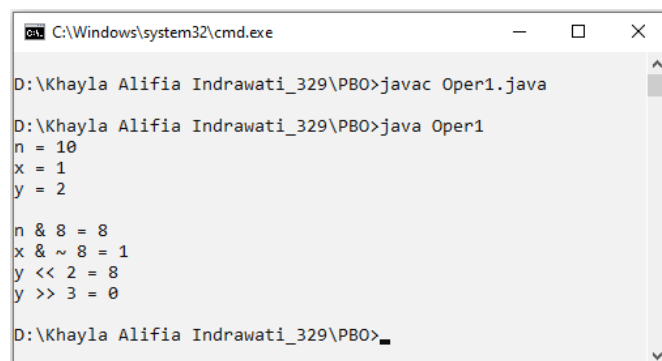
- **Variabel n** bertipe data **integer** diberi nilai awal 10 dan bentuk biner 1010.
- **Variabel x** bertipe data **integer** diberi nilai awal 1 dan bentuk biner 0001.
- **Variabel y** bertipe data **integer** diberi nilai awal 2 dan bentuk biner 0010.

Selanjutnya, program mencetak nilai awal dari masing-masing variabel. Kemudian mencetak hasil operasi bit ke layar, di mana:

- Untuk  $n \& 8 = 1010 \& 1000 = 1000$  (8 dalam desimal). Hal ini dikarenakan menggunakan operator AND (&) yang hanya akan menghasilkan 1 jika keduanya bernilai 1.
- Untuk  $x \& \sim 8 = 0001 \& \sim(1000) = 0001 \& 0111 = 0001$  (1 dalam desimal). Hal ini dikarenakan menggunakan operator NOT(~) yang membalik bit, sehingga 1000 menjadi 0111, dan penggunaan operator AND(&) yang hanya akan menghasilkan 1 jika keduanya bernilai 1.
- Untuk  $y \<< 2 = 0010 \<< 2 = 1000$  (8 dalam desimal). Hal ini dikarenakan menggunakan operator left shift (<<) yaitu menggeser bit ke kiri sebanyak 2 posisi.
- Untuk  $y \>> 3 = 0010 \>> 3 = 0000$  (0 dalam desimal). Hal ini dikarenakan menggunakan operator right shift (>>) yaitu menggeser bit ke kanan sebanyak 3 posisi.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
C:\Windows\system32\cmd.exe
D:\Khayla Alifia Indrawati_329\PB0>javac Oper1.java
D:\Khayla Alifia Indrawati_329\PB0>java Oper1
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
D:\Khayla Alifia Indrawati_329\PB0>
```

### 13. Class Oper2

Kode program ini mendefinisikan kelas Oper2 dengan metode main, yang akan dieksekusi saat dijalankan, dengan deklarasi:

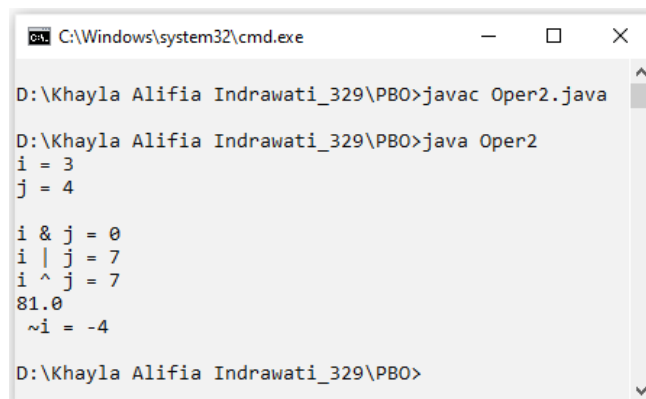
- **Variabel i** bertipe data **char**, diberi nilai awal 3 dan bentuk biner 00000011
- **Variabel j** bertipe data **char**, diberi nilai awal 4 dan bentuk biner 00000100

Selanjutnya, program mencetak nilai awal dari masing-masing variabel. Kemudian mencetak hasil operasi bit ke layar, di mana:

- Untuk **i & j = 00000011 & 00000100 = 00000000** (0 dalam desimal). Hal ini dikarenakan menggunakan operator AND (&) yang hanya akan menghasilkan 1 jika keduanya bernilai 1.
- Untuk **i | j = 00000011 | 00000100 = 00000111** (7 dalam desimal). Hal ini dikarenakan menggunakan operator OR (|) yang akan menghasilkan 1 jika salah satu bit bernilai 1.
- Untuk **i ^ j = 00000011 ^ 00000100 = 00000111** (7 dalam desimal). Hal ini dikarenakan menggunakan operator XOR (^) yang akan menghasilkan 1 jika kedua bit berbeda.
- Untuk **Math.pow(i, j)**, dimana i sebagai bilangan bulat dan j sebagai pangkat, sehingga menghitung  $3^4 = 81$ . Math.pow(a, b) digunakan untuk menghitung pemangkatan di java, bukan menggunakan operator “^”.
- **~i = ~(00000011) = 11111100**. Hal ini dikarenakan menggunakan operator NOT(~) yang membalik bit, yang dalam representasi bilangan negatif (komplemen dua) bernilai -4 dalam desimal.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
C:\Windows\system32\cmd.exe

D:\Khayla Alifia Indrawati_329\PBO>javac Oper2.java

D:\Khayla Alifia Indrawati_329\PBO>java Oper2
i = 3
j = 4

i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4

D:\Khayla Alifia Indrawati_329\PBO>
```

### 14. Class Oper3

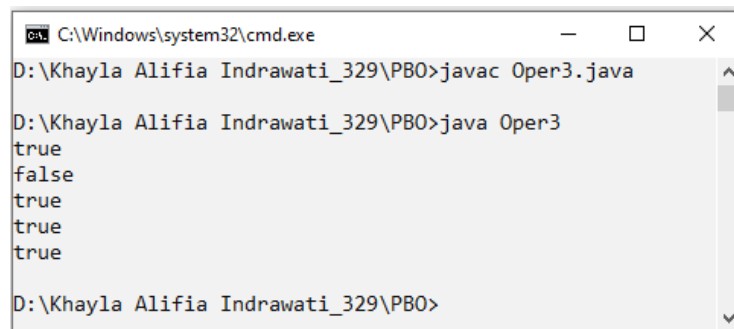
Kode program ini mendefinisikan kelas Oper3 dengan metode main, yang akan dieksekusi saat program dijalankan. Program ini tidak memiliki deklarasi variabel, tetapi langsung menggunakan kondisi dalam pernyataan if untuk mengevaluasi ekspresi logika dan menampilkan hasilnya. Berikut cara bekerja setiap kondisi yang ada pada algoritma program:

- Untuk **if (true && true)** hasilnya menampilkan true, karena menggunakan operator AND logis (&&) akan menghasilkan true jika keduanya bernilai true.

- Untuk **if (true & true)** dieksekusi, tetapi **System.out.println(true & false)** tidak mencetak apapun karena ekspresi dalam println adalah false. Operator AND bitwise (&) tetap memeriksa keduanya meskipun hasil akhir sudah bisa ditentukan.
- Untuk **if (true)** hasilnya akan menampilkan true, karena kondisi dalam if sudah true.
- Untuk **if (true || true)** hasilnya menampilkan true, karena menggunakan operator OR Logis (||) akan langsung menghasilkan true jika salah satunya bernilai true. Operator ini tidak perlu memeriksa keduanya.
- Untuk **if (true | false)** hasilnya menampilkan true, karena menggunakan operator OR bitwise (|) akan menghasilkan true jika salah satunya bernilai true. Operator ini juga tetap memeriksa kedua operasi.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```

C:\Windows\system32\cmd.exe
D:\Khayla Alifia Indrawati_329\PB0>javac Oper3.java
D:\Khayla Alifia Indrawati_329\PB0>java Oper3
true
false
true
true
true
D:\Khayla Alifia Indrawati_329\PB0>

```

## 15. Class Oper4

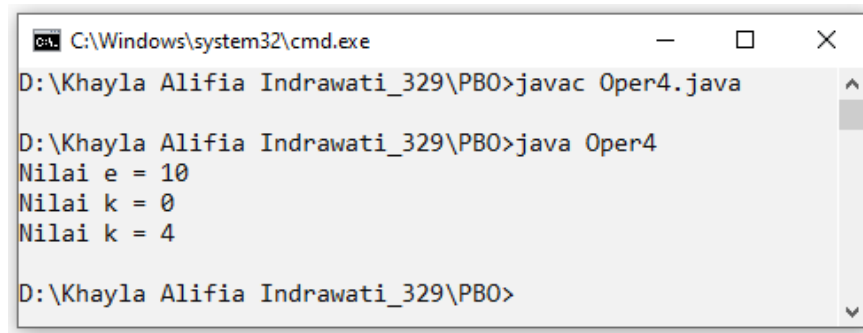
Kode program ini mendefinisikan kelas Oper4 dengan metode main, yang akan dieksekusi saat program dijalankan. Program ini menggunakan operator ternary untuk menentukan nilai berdasarkan kondisi tertentu. Adapun deklarasi variabel dalam metode main yaitu:

- **Variabel i dan j** bertipe data **integer**, diberi nilai awal 0.
- **Variabel c dan d** bertipe data **char**, diberi nilai awal 8 dan 10
- **Variabel e** akan menampung hasil dari kondisi **((int)c > (int)d ? c : d)**
- **Variabel k** akan menampung hasil dari kondisi **((i > j) ? i : j)**, di mana i dan j bernilai 0 sehingga k = 0.

Selanjutnya, terdapat perintah untuk menampilkan nilai awal yaitu menampilkan nilai e = 10 dan k = 0. Kemudian, terdapat inisialisasi untuk nilai i = 2 dan j = 3 lalu diberikan variabel k untuk menampung hasil dari kondisi **((i++ > j++) ? i : j)**, dimana i++ menggunakan nilai 2, lalu i bertambah menjadi 3 dan j++ menggunakan nilai 3, lalu j bertambah menjadi 4. Karena kondisi **i++ > j++** adalah false, maka k = j.

Kemudian, terdapat perintah untuk mencetak hasil dari kondisi yang akan ditampung oleh **variabel k**. Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
C:\Windows\system32\cmd.exe
D:\Khayla Alifia Indrawati_329\PBO>javac Oper4.java
D:\Khayla Alifia Indrawati_329\PBO>java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4
D:\Khayla Alifia Indrawati_329\PBO>
```

## 16. Class Oprator

Kode program ini mendefinisikan kelas Oprator dengan metode main, yang akan dieksekusi saat program berjalan. Adapun deklarasi variabel dalam metode main yaitu:

- **Variabel Bool1, Bool2, dan TF** bertipe data **Boolean**.
- **Variabel i, j, dan hsl** bertipe data **integer**.
- **Variabel x, y, dan res** bertipe data **float**.

Kemudian terdapat perintah output yang berisi kalimat untuk menambahkan perintah menampilkan output di setiap operasi yang ada di dalam program. Lalu ada inisialisasi untuk **Bool1 = true** dan **Bool2 = false**.

**Operasi pertama** yaitu operasi logika, dimana **Bool1 && Bool2 (AND)** menghasilkan false karena salah satunya bernilai false. **Bool1 || Bool2 (OR)** menghasilkan true karena salah satu bernilai true. **!Bool1 (NOT)** menghasilkan false karena merupakan NOT sehingga membalik true. Operasi logika terakhir yaitu **Bool1 ^ Bool2** menghasilkan true karena keduanya berbeda, operasi ini merupakan operasi XOR. Masing-masing operasi logika saya berikan perintah menampilkan hasil operasi ke layar.

Terdapat inisialisasi untuk **i = 5, j = 2** sebelum dilakukan operasi kedua. **Operasi kedua** yaitu aritmetika dengan integer, dimana **hsl = i + j** menghasilkan 7 karena penjumlahan. Untuk **hsl = i - j** menghasilkan 3 karena pengurangan. Untuk **hsl = i / j** menghasilkan 2 karena pembagian dan hasil dibulatkan (integer). Untuk **hsl = i \* j** menghasilkan 10 karena perkalian. Operasi aritmetika terakhir yaitu **hsl = i % j** menghasilkan 1 karena sisa pembagian (modulo). Masing-masing operasi aritmetika dengan integer saya berikan perintah menampilkan hasil operasi ke layar.

Terdapat inisialisasi untuk **x = 5, y = 5** sebelum dilakukan operasi ketiga. **Operasi ketiga** yaitu operasi aritmetika dengan float, operasi ini sama dengan operasi aritmetika **i** dan **j** sebelumnya, dengan hasil sesuai nilai yang diberikan. Hanya saja pada operasi aritmetika untuk **x** dan **y** hasilnya adalah float dan tidak ada operasi untuk modulo. Selain itu, masing-masing operasi aritmetika dengan float saya berikan perintah menampilkan hasil operasi ke layar.

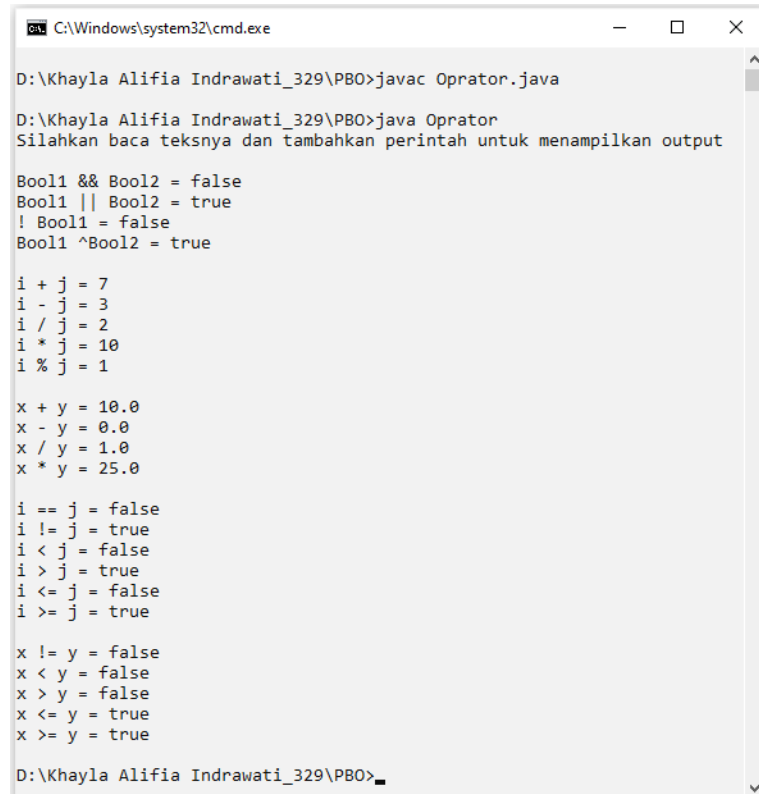
**Operasi keempat** yaitu operasi relasional untuk variabel **i** dan **j**, dimana **(i == j)** menghasilkan false karena  $5 \neq 2$ . Untuk **(i != j)** menghasilkan true karena  $5 \neq 2$ . Untuk **(i < j)** menghasilkan false karena  $5 > 2$ . Untuk **(i > j)** menghasilkan true karena  $5 > 2$ . Untuk **(i <= j)** menghasilkan false karena  $5 <= 2$ . Operasi relasional terakhir yaitu **(i <=**

j) menghasilkan true karena  $5 \geq 2$ . Masing-masing operasi relasional untuk i dan j saya berikan perintah menampilkan hasil operasi ke layar.

**Operasi kelima** yaitu operasi relasional untuk x dan y. Operasi ini sama dengan operasi relasional i dan j dengan hasil sesuai nilai yang diberikan. Hanya saja pada operasi relasional untuk x dan y tidak ada operator `==`. Masing-masing operasi relasional untuk x dan y saya berikan perintah menampilkan hasil operasi ke layar.

Program ini dijalankan menggunakan **command prompt**. Output dari program ini yaitu sebagai berikut.

**Output:**



```
C:\Windows\system32\cmd.exe

D:\Khayla Alifia Indrawati_329\PBO>javac Oprator.java

D:\Khayla Alifia Indrawati_329\PBO>java Oprator
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output

Bool1 && Bool2 = false
Bool1 || Bool2 = true
! Bool1 = false
Bool1 ^Bool2 = true

i + j = 7
i - j = 3
i / j = 2
i * j = 10
i % j = 1

x + y = 10.0
x - y = 0.0
x / y = 1.0
x * y = 25.0

i == j = false
i != j = true
i < j = false
i > j = true
i <= j = false
i >= j = true

x != y = false
x < y = false
x > y = false
x <= y = true
x >= y = true

D:\Khayla Alifia Indrawati_329\PBO>_
```