

LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Sulistriawati Ramadhani

Nim : 13020220294

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR

2025

1. Asgdll.java

Program ini mendefinisikan kelas utama bernama *Asgdll*. Di dalam metode *main*, terdapat pendefinisian dua variabel:

- Satu variabel bertipe *float* diinisialisasi langsung dengan nilai 20.0f.
- Variabel kedua bertipe *double* dideklarasikan tanpa nilai awal, kemudian diberikan nilai 10.0 (konversi implisit dari float ke double). Selanjutnya, nilai kedua variabel tersebut dicetak ke konsol dengan menggabungkan teks dan karakter newline untuk memisahkan baris. Program ini berfungsi sebagai contoh dasar pendefinisian variabel, penugasan nilai, dan output ke layar.

Output

```
C:\Users\Asus>D:

D:\>cd D:\Tugas PBO\Tugas 1\Source Code

D:\Tugas PBO\Tugas 1\Source Code>javac Asgdll.java

D:\Tugas PBO\Tugas 1\Source Code>java Asgdll
f : 20.0
f11: 10.0
```

2. Asign.java

Pada program ini, kelas *Asign* menjadi titik awal eksekusi melalui metode *main*.

- Pertama, sebuah variabel integer dideklarasikan tanpa nilai, kemudian program mencetak pesan "hello" ke layar dengan karakter newline untuk membuat baris baru.
- Setelah itu, variabel tersebut diisi dengan angka 5 dan dicetak kembali bersama dengan penjelasan mengenai nilainya. Program ini mengilustrasikan penggunaan variabel sederhana dan bagaimana output dapat ditampilkan secara langsung di konsol.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Asign.java

D:\Tugas PBO\Tugas 1\Source Code>java Asign
hello
Ini nilai i :5
```

3. ASIGNi.java

Dalam program ini, beragam tipe data primitif dieksplorasi. Kelas *ASIGNi* mendeklarasikan:

- Variabel dengan tipe *short*, *int*, dan *long* untuk bilangan bulat,
- Dua variabel *char*, satu diberi nilai numerik (65, yang mewakili 'A' dalam tabel ASCII) dan satu lagi diinisialisasi secara langsung dengan 'Z',
- Dan juga variabel bertipe *double* dan *float*. Seluruh nilai yang dideklarasikan tersebut kemudian ditampilkan ke layar. Program ini membantu memahami perbedaan serta penggunaan masing-masing tipe data.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac ASIGNi.java

D:\Tugas PBO\Tugas 1\Source Code>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
```

4. BacaData.java

Program ini mendemonstrasikan cara membaca input dari pengguna dengan menggunakan kelas *Scanner*.

- Di dalam metode *main*, sebuah objek *Scanner* dibuat untuk menangkap input dari keyboard.
- Program kemudian membaca sebuah bilangan bulat yang dimasukkan oleh pengguna dan menyimpannya dalam sebuah variabel.
- Akhirnya, nilai tersebut ditampilkan ke konsol dengan pesan yang mendeskripsikan hasil input. Ini merupakan contoh dasar interaksi antara program dan pengguna melalui masukan konsol.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac BacaData.java

D:\Tugas PBO\Tugas 1\Source Code>java BacaData
Contoh membaca dan menulis, ketik nilai integer
12
Nilai yang dibaca : 12
```

5. Bacakar.java

Di sini, program menggunakan kelas *BufferedReader* bersama dengan *InputStreamReader* untuk menangani masukan secara manual.

- Program meminta pengguna untuk memasukkan sebuah karakter dan kemudian sebuah bilangan bulat.
- Pembacaan dilakukan dengan mengambil baris input, kemudian karakter pertama diambil untuk merepresentasikan input karakter, dan baris berikutnya diubah menjadi integer.
- Kedua nilai tersebut dicetak ke layar bersama pesan penutup. Program ini menekankan pada teknik membaca input menggunakan stream secara lebih mendasar.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Bacakar.java
D:\Tugas PBO\Tugas 1\Source Code>java Bacakar
hello
baca 1 karakter : a
baca 1 bilangan : 1
a
1
bye
```

6. Casting1.java

Pada program ini, teknik *casting* atau konversi tipe data antar primitif dijelaskan.

- Variabel-variabel dengan tipe *int*, *float*, *char*, dan *double* dideklarasikan dengan nilai awal tertentu.
- Kemudian, dilakukan konversi eksplisit seperti mengubah integer ke float, atau karakter ke nilai numerik berdasarkan kode ASCII.
- Hasil setiap operasi casting ditampilkan ke konsol, sehingga memberikan gambaran mengenai bagaimana data dapat dipindahkan antar tipe secara eksplisit maupun implisit.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Casting1.java
D:\Tugas PBO\Tugas 1\Source Code>java Casting1
5.0
6.0
2
3.200000047683716
53
53.0
53.0
3
3.14
```

7. Casting2.java

Program ini menunjukkan cara mengubah data antar tipe yang lebih kompleks dengan memanfaatkan metode parsing dan konversi string.

- Nilai-nilai yang awalnya berupa string (misalnya "67", "45", "100") diubah ke dalam tipe numerik (integer, double, atau float) menggunakan metode *parse*.
- Sebaliknya, beberapa nilai numerik dikonversi menjadi string menggunakan metode *valueOf*
- Selain itu, terdapat juga contoh penggunaan metode untuk mengubah nilai integer ke double dan sebaliknya. Semua perubahan ini diikuti dengan pencetakan hasil untuk memperlihatkan transformasi tipe data secara jelas.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Casting2.java
D:\Tugas PBO\Tugas 1\Source Code>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
```

8. Ekspresi.java

Program ini fokus pada penggunaan operator kondisional (ternary operator) untuk menentukan nilai berdasarkan kondisi perbandingan.

- Dua variabel integer diinisialisasi, dan operator ternary digunakan untuk memilih nilai terkecil di antara keduanya.
- Hasil perbandingan dan keputusan tersebut kemudian dicetak ke layar bersama informasi awal kedua variabel. Dengan demikian, program ini menekankan penulisan ekspresi kondisional yang ringkas dalam Java.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Ekspresi.java
D:\Tugas PBO\Tugas 1\Source Code>java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
```

9. Ekspresi1.java

Dalam program ini, perbedaan antara pembagian bilangan bulat dan pembagian dengan tipe desimal diperlihatkan.

- Awalnya, pembagian antara dua integer menghasilkan bilangan bulat (tanpa angka di belakang koma), meskipun dicetak dalam format float.
- Untuk mendapatkan hasil dengan pecahan, dilakukan *casting* dari integer ke float sebelum operasi pembagian.
- Semua langkah perhitungan kemudian dicetak, memperlihatkan bagaimana *casting* dapat mempengaruhi hasil operasi aritmatika.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Ekspresi1.java

D:\Tugas PBO\Tugas 1\Source Code>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
```

10. PrintHello.java

Program ini merupakan demonstrasi sederhana penggunaan metode *print* dan *println*.

- Kelas *Hello* mencetak beberapa pesan seperti "Hello", "World", dan "Welcome".
- Perbedaan mendasar antara *print* (yang tidak otomatis menambahkan baris baru) dan *println* (yang menambahkan newline setelah teks) ditampilkan melalui urutan pencetakan pesan. Dengan demikian, program ini memberikan pemahaman mengenai cara mengatur format output pada konsol.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Hello.java

D:\Tugas PBO\Tugas 1\Source Code>java Hello
Hello
Hello World
Welcome
```

11. Incr.java

Program ini mengilustrasikan perbedaan antara operator post-increment dan pre-increment.

- Dua variabel integer dideklarasikan dan diinisialisasi.
- Salah satu variabel dioperasikan dengan post-increment saat nilainya disimpan ke variabel lain, sehingga nilai asli yang disimpan berbeda dari nilai setelah increment.
- Selanjutnya, pre-increment digunakan sehingga nilai variabel bertambah sebelum dicetak. Hasilnya menunjukkan perbedaan cara kerja kedua operator tersebut dalam konteks assignment dan output.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Incr.java

D:\Tugas PBO\Tugas 1\Source Code>java Incr
Nilai i : 5
Nilai j : 3
```

12. Oper1.java

Program ini berfokus pada penerapan operasi bitwise.

- Beberapa variabel integer diinisialisasi dengan nilai tertentu.
- Operator bitwise seperti AND, NOT, serta pergeseran bit (shift left dan shift right) diterapkan pada variabel tersebut.
- Masing-masing operasi menghasilkan nilai yang dicetak ke layar, sehingga memberikan gambaran tentang manipulasi bit secara langsung dalam representasi biner.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Oper1.java

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

13. Oper2.java

Di sini, operasi bitwise dilakukan dengan variabel bertipe *char*.

- Nilai-nilai karakter (yang juga dapat direpresentasikan secara biner) digunakan dalam operasi AND, OR, XOR, dan NOT.
- Selain itu, program juga memanfaatkan metode *Math.pow* untuk operasi pemangkatan, meskipun hal tersebut berada di luar ranah operasi bitwise.
- Output yang dihasilkan menjelaskan bagaimana operasi bitwise bekerja pada nilai karakter dan hasil perhitungan matematika.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Oper2.java

D:\Tugas PBO\Tugas 1\Source Code>java Oper2
i = 3
j = ♦
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
```

14. Oper3.java

Program ini menguji beberapa operasi logika dalam Java.

- Dengan menggunakan pernyataan *if*, dievaluasi beberapa kondisi logika yang menggunakan operator logika seperti AND (&& dan &), serta OR (|| dan |).
- Kondisi-kondisi tersebut dieksekusi jika bernilai benar, dan hasil evaluasinya ditampilkan ke layar. Program ini menyoroti perbedaan antara operator logika biasa dan operator bitwise yang juga dapat digunakan dalam konteks boolean.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Oper3.java

D:\Tugas PBO\Tugas 1\Source Code>java Oper3
true
false
true
true
true
```


15. Oper4.java

Program ini memanfaatkan operator ternary untuk memilih nilai berdasarkan perbandingan.

- Terdapat variabel integer dan karakter yang dibandingkan menggunakan operator ternary untuk menentukan nilai mana yang lebih besar atau lebih kecil.
- Setelah penugasan awal, nilai variabel juga dimodifikasi menggunakan operasi increment (post-increment) dan dibandingkan kembali, dengan hasilnya ditampilkan ke konsol. Hal ini menggambarkan penggunaan operator kondisional yang fleksibel dalam pemilihan nilai di antara variabel.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Oper4.java  
  
D:\Tugas PBO\Tugas 1\Source Code>java Oper4  
Nilai e = 10  
Nilai k = 0  
Nilai k = 4
```

16. Operator.java

Program ini merangkum penerapan berbagai jenis operator dalam Java.

- Variabel dengan tipe *boolean*, *int*, dan *float* dideklarasikan untuk menunjukkan operasi logika (seperti AND, OR, NOT, XOR), aritmatika (penjumlahan, pengurangan, perkalian, pembagian, modulo), dan relasional (sama dengan, tidak sama dengan, lebih kecil, lebih besar, dan sebagainya).
- Walaupun output tidak langsung dicetak, kode ini menguraikan cara kerja masing-masing operator dalam berbagai konteks tipe data. Program ini berfungsi sebagai referensi komprehensif untuk memahami dasar-dasar operator di dalam bahasa Java.

Output

```
D:\Tugas PBO\Tugas 1\Source Code>javac Oprator.java  
  
D:\Tugas PBO\Tugas 1\Source Code>java Oprator  
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output
```