

**LAPORAN
PEMROGRAMAN BERBASIS OBJEK**



Nama : Elva Ruhustina
Stambuk : 13020230288
Kelas : TI_PBO-14 (B4)
Dosen : Mardiyah Hasnawi, S.Kom., M.T., MTA

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

Tugas 1 Pemrograman Berbasis Objek – Elva Ruhustina

1. Output:

```
D:\Elva Ruhustina\Tugas PBO\Tugas 1>javac Asgdll.java
D:\Elva Ruhustina\Tugas PBO\Tugas 1>java Asgdll
f      : 20.0
fill   : 10.0
```

Penjelasan

Dalam program dengan kelas Asgdll, terdapat dua variabel: f (float) dengan nilai awal 20.0f dan fill (double), yang kemudian diisi 10.0f. Karena fill bertipe double, nilai 10.0f dikonversi otomatis tanpa perlu deklarasi khusus. Saat program dijalankan, outputnya "f : 20.0" dan "fill : 10.0", di mana fill tetap bertipe double, tetapi tidak menampilkan angka desimal tambahan karena nilai awalnya berasal dari 10.0f.

2. Output:

```
D:\Elva Ruhustina\Tugas PBO\Tugas 1>javac Assign.java
D:\Elva Ruhustina\Tugas PBO\Tugas 1>java Assign
hello
Ini nilai i :5
```

Penjelasan

Dalam program dengan kelas Assign, terdapat variabel i yang dideklarasikan sebagai int, tetapi belum langsung diberi nilai awal. Setelah itu, i diinisialisasi dengan 5. Output ditampilkan dengan System.out.print("hello\n");, yang mencetak "hello" lalu pindah ke baris baru. Kemudian, System.out.println("Ini nilai i: " + i); mencetak "Ini nilai i: 5".

3. Output:

```
D:\Elva Ruhustina\Tugas PBO\Tugas 1>javac ASIGNi.java
D:\Elva Ruhustina\Tugas PBO\Tugas 1>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 :

Dalam program dengan kelas ASIGNi, terdapat beberapa variabel dengan berbagai tipe data. ks (short) dan ki (int) memiliki nilai 1, sedangkan kl (long) menyimpan 10.000. Variabel c (char) diberi nilai 65, yang dalam kode ASCII merepresentasikan karakter 'A', sementara c1 (char) langsung diinisialisasi dengan 'z'. Untuk tipe desimal, x (double) diberi nilai 50.2f, yang dikonversi otomatis dari float ke double sehingga nilainya menjadi 50.20000076293945. Di sisi lain, y (float) dengan nilai 50.2f tetap menampilkan 50.2, sesuai dengan presisi float.

4. Output:

```
D:\Elva Ruhustina\Tugas PBO\Tugas 1>javac BacaData.java

D:\Elva Ruhustina\Tugas PBO\Tugas 1>java BacaData
Contoh membaca dan menulis, ketik nilai integer:
88
Nilai yang dibaca : 88
```

Penjelasan:

Program BacaData menggunakan Scanner dari java.util untuk membaca input angka dari pengguna. Program ini memiliki variabel a (integer) untuk menyimpan angka yang dimasukkan dan objek masukan (Scanner) untuk menangani input. Saat dijalankan, pengguna diminta memasukkan angka, yang kemudian disimpan dalam variabel a. Setelah itu, program menampilkan output berupa "Nilai yang dibaca: [angka]". Misalnya, jika pengguna mengetik 88, maka program akan menampilkan "Nilai yang dibaca: 88"

5. Output:

```
D:\Elva Ruhustina\Tugas PBO\Tugas 1>java Bacakar
hello
baca 1 karakter : e
baca 1 bilangan : 8
e
8
bye
```

Penjelasan :

Program Bacakar menggunakan BufferedReader dan InputStreamReader untuk membaca input dari pengguna serta menangani pengecualian dengan IOException. Variabel cc (char) menyimpan karakter pertama yang dimasukkan, sedangkan bil (int) menyimpan angka yang dikonversi dari string. Saat dijalankan, program mencetak "hello", lalu meminta pengguna memasukkan satu karakter dan satu angka. Misalnya, jika pengguna mengetik 'e' diikuti angka 8, program menampilkan 'e' dan 8. Terakhir, program mencetak "bye" sebelum selesai.

6. Output:

```
D:\Elva Ruhustina\Tugas PBO\Tugas 1>javac Casting1.java

D:\Elva Ruhustina\Tugas PBO\Tugas 1>java Casting1
5.0
6.0
2
3.200000047683716
53
53.0
53.0
3
3.14
```

Penjelasan :

Program Casting1 mendeklarasikan variabel dengan tipe data berbeda dan menerapkan konversi tipe. Variabel a (int) dikonversi ke float menjadi 5.0, sedangkan b (int) dikonversi ke double menjadi 6.0. Variabel d (float) dikonversi ke int menjadi 2, sementara e (float) dikonversi ke double tetap 3.2 karena kompatibel. Variabel g (char) dikonversi ke int menjadi 53 sesuai nilai ASCII dari '5'. Variabel k (double) dikonversi ke int menjadi 3 (desimal terpotong) dan ke float tetap 3.14, meskipun dengan presisi lebih rendah.

7. Output:

```
D:\Elva Ruhustina\Tugas PBO\Tugas 1>javac Casting2.java

D:\Elva Ruhustina\Tugas PBO\Tugas 1>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 Casting2 melakukan berbagai konversi data, seperti mengubah teks jadi angka, angka jadi teks, dan konversi antarjenis angka. Misalnya, String "67" diubah ke int (67), "45" jadi double (45.0), dan "100" jadi float (100.0). Sebaliknya, angka juga bisa diubah ke String, seperti 9 jadi "9", karakter '5' jadi "5", dan 3.2f jadi "3.2". Ada juga konversi antarangka, misalnya variabel a (int 67) diubah ke double, lalu balik lagi ke int tanpa perubahan. Variabel b (int 9) diubah ke double (9.0) dan disimpan di c. Terakhir, variabel l, yang awalnya "100", berubah jadi "3.2" setelah dikonversi pakai String.valueOf(e), karena e adalah float dengan nilai 3.2f.

8. Output:

```
D:\Elva Ruhustina\Tugas PBO\Tugas 1>javac Ekspresi.java

D:\Elva Ruhustina\Tugas PBO\Tugas 1>java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y) ? x:y = 1
```

Penjelasan :

Program Ekspresi punya dua variabel, x = 1 dan y = 2, lalu nilainya dicetak pakai System.out.print(). Setelah itu, program pakai operator ternary (x < y) ? x : y buat ngecek kondisi. Karena 1 < 2 itu true, maka hasilnya adalah x, yaitu 1. Terakhir, program nampilin hasilnya di konsol dengan format: "hasil ekspresi = (x < y) ? x : y = 1".

9. Output:

```
hasil ekspresi = (x<y) ? x:y = 1
D:\Elva Ruhustina\Tugas PBO\Tugas 1>javac Ekspresi1.java

D:\Elva Ruhustina\Tugas PBO\Tugas 1>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
D:\Elva Ruhustina\Tugas PBO\Tugas 1>
```

Penjelasan :

Program Ekspresi1 menunjukkan perbedaan hasil pembagian antara int dan float di Java.

- Pertama, ada variabel $x = 1$ dan $y = 2$ yang bertipe int, serta fx dan fy yang bertipe float. Saat x / y dihitung, hasilnya 0 karena pembagian antara dua bilangan int selalu dibulatkan ke bawah. Tapi kalau fx / fy , hasilnya 0.5, karena float bisa menyimpan angka desimal.

- Kalau x dan y dikonversi dulu ke float sebelum dibagi $((float)x / (float)y)$, hasilnya tetap 0.5. Saat $x = 10$ dan $y = 3$, pembagian x / y sebagai int menghasilkan 3 (tanpa desimal). Tapi kalau dikonversi ke float sebelum dibagi, hasilnya 3.33333333, karena tipe float bisa menyimpan angka desimal dengan lebih presisi.

10. Output:

```
D:\Elva Ruhustina\Tugas PBO\Tugas 1>javac Hello.java

D:\Elva Ruhustina\Tugas PBO\Tugas 1>java Hello
Hello
Hello World
Welcome
```

Penjelasan :

Program Hello menunjukkan perbedaan antara `print()` dan `println()` dalam mencetak teks ke konsol. `System.out.print("Hello");` mencetak "Hello" tanpa pindah ke baris baru, sementara `System.out.print("\nHello ");` menambahkan baris baru sebelum mencetak "Hello ", sehingga teks muncul di baris berikutnya. Selanjutnya, `System.out.println("World");` mencetak "World" lalu langsung pindah ke baris baru. Terakhir, `System.out.println("Welcome");` mencetak "Welcome" dan otomatis berpindah ke baris berikutnya. Jadi, `print()` hanya mencetak teks tanpa pindah baris, sedangkan `println()` mencetak teks lalu berpindah ke baris baru secara otomatis.

11. Output:

```
D:\Elva Ruhustina\Tugas PBO\Tugas 1>javac Incr.java

D:\Elva Ruhustina\Tugas PBO\Tugas 1>java Incr
Nilai i : 5
Nilai j : 3
```

Penjelasan :

Program Incr menunjukkan perbedaan antara post-increment (`i++`) dan pre-increment (`++i`) dalam operasi peningkatan nilai variabel. Variabel i dan j bertipe int, dengan i diawali nilai 3. Saat $j = i++$; dijalankan, nilai i yang masih 3 diberikan ke j terlebih dahulu, lalu i bertambah menjadi 4. Kemudian, pada `System.out.println(++i);`, pre-increment membuat i dari 4 menjadi 5 sebelum dicetak. Sementara itu, nilai j tetap 3 karena tidak berubah setelah assignment. Akhirnya, program mencetak "Nilai i : 5" dan "Nilai j : 3".

12. Output:

```
D:\Elva Ruhustina\Tugas PB0\Tugas 1>javac Oper1.java

D:\Elva Ruhustina\Tugas PB0\Tugas 1>java Oper1
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

Penjelasan:

Program Oper1 menunjukkan cara kerja operasi bitwise di Java, seperti AND (&), NOT (~), left shift (<<), dan right shift (>>) menggunakan bilangan biner. Variabel n = 10 (biner 1010), x = 1 (0001), dan y = 2 (0010) direpresentasikan dalam bentuk biner. n & 8 membandingkan 1010 dengan 1000, menghasilkan 1000 (desimal 8). Pada x & ~8, operator NOT (~) mengubah 1000 menjadi 0111 (dalam 4-bit), sehingga 0001 & 0111 tetap 1. Operasi y << 2 menggeser 0010 ke kiri dua kali, menjadi 1000 (desimal 8), sedangkan y >> 3 menggeser 0010 ke kanan tiga kali, menghilangkan semua bit hingga tersisa 0.

13. Output:

```
D:\Elva Ruhustina\Tugas PB0\Tugas 1>javac Oper2.java

D:\Elva Ruhustina\Tugas PB0\Tugas 1>java Oper2
i = 3
j = 4
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
```

Penjelasan :

Program Oper2 menggunakan operasi bitwise dan perhitungan matematika pada variabel bertipe char. Di sini, i = 3 dan j = 4, yang dalam desimal tetap bernilai 3 dan 4. Saat dilakukan operasi AND (i & j), angka biner 00000011 (3) dan 00000100 (4) dibandingkan bit per bit, hasilnya 00000000 atau 0. Sedangkan OR (i | j) dan XOR (i ^ j) menghasilkan 00000111, yang jika dikonversi ke desimal bernilai 7. Untuk operasi matematika, Math.pow(i, j) menghitung 3 pangkat 4, hasilnya 81.0. Sementara itu, operasi NOT (~i) membalik semua bit dari 00000011 (3), yang dalam sistem two's complement menjadi -4.

14. Output:

```
D:\Elva Ruhustina\Tugas PB0\Tugas 1>javac Oper3.java

D:\Elva Ruhustina\Tugas PB0\Tugas 1>java Oper3
true
false
true
true
true
```

Penjelasan :

Program Oper3 membandingkan operator logika (&&, ||) dengan operator bitwise (&, |) dalam pernyataan if. Pada ekspresi true && true, operator AND logika menghasilkan true, sehingga mencetak "true". Sementara itu, true & false menggunakan AND bitwise, yang mengevaluasi kedua operand dan menghasilkan false, sehingga mencetak "false". Pada if (true), karena kondisinya true, blok kode langsung dieksekusi dan mencetak "true". Sedangkan pada true || true, operator OR logika menghasilkan true, sehingga mencetak "true". Terakhir, true | false menggunakan OR bitwise, yang tetap menghasilkan true dan mencetak "true". Perbedaannya, operator logika (&&, ||) hanya mengevaluasi bagian kanan jika diperlukan, sementara operator bitwise (&, |) selalu mengevaluasi kedua sisi ekspresi, meskipun hasil akhirnya sudah bisa ditentukan sebelumnya.

15. Output:

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

Penjelasan :

Program Oper4 menggunakan operator ternary (? :) untuk menentukan nilai variabel berdasarkan kondisi tertentu. Awalnya, i = 0 dan j = 0, sedangkan c = 8 dan d = 10 bertipe char, yang dikonversi ke int saat dibandingkan. Karena (int)c (8) > (int)d (10) bernilai false, dan i > j juga false, maka e = 10 dan k = 0. Setelah i dan j diperbarui menjadi 2 dan 3, ekspresi i++ > j++ dievaluasi. Dalam ekspresi ini, nilai i (2) dibandingkan dengan j (3) sebelum keduanya bertambah. Karena 2 > 3 adalah false, hasil yang dipilih adalah nilai j setelah increment, yaitu 4, sehingga k menjadi 4. Hasil akhirnya yang ditampilkan adalah "Nilai e = 10", "Nilai k = 0", dan "Nilai k = 4".

16. Output:

```
C:\WINDOWS\system32\cmd. x + v
Nilai e = 10
Nilai k = 0
Nilai k = 4
D:\Andi Nuraeni Bintang\Tugas PBO\Tugas 1>javac Oprator.java
D:\Andi Nuraeni Bintang\Tugas PBO\Tugas 1>java Oprator
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output
Bool1 AND Bool2: false
Bool1 OR Bool2: true
NOT Bool1: false
Bool1 XOR Bool2: true
i + j: 7
i - j: 3
i * j: 10
i / j: 2
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
```

Penjelasan :

Program Oprator menguji operasi logika, aritmatika, dan relasional pada tipe data boolean, integer, dan float. Dalam operasi boolean, jika Bool1 = true dan Bool2 = false, maka hasil evaluasi adalah false untuk &&, true untuk ||, false untuk !Bool1, dan true untuk ^ (XOR). Pada operasi aritmatika, dengan i = 5 dan j = 2, hasilnya adalah 7 (penjumlahan), 3 (pengurangan), 10 (perkalian), 2 (pembagian, dibulatkan), dan 1 (modulus). Untuk float (x = 5.0, y = 5.0), hasilnya 10.0, 0.0, 25.0, dan 1.0. Dalam operasi relasional, untuk integer, == bernilai false, != bernilai true, < bernilai false, > bernilai true, <= bernilai false, dan >= bernilai true. Sementara itu, untuk float, hasilnya false untuk !=, <, dan >, tetapi true untuk <= dan >=.