

Tugas 1

Makassar, 21 Februari 2025

**PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Hikmalia
Stambuk : 13020230233
Dosen : Mardiyah Hasnawi, S.Kom., MT,MTA

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

1. Program Java “Asgdll” merupakan contoh pemrograman sederhana, dimana mendeklarasikan dua variabel yang bertipe data float dan double, pada program ini f yang bertipe float dan diinisialisasi dengan nilai 20.0f, sedangkan f11 yang bertipe double dan kemudian diinisialisasi dengan nilai 10.0f. Saat Program ini dijalankan menggunakan perintah System.out.println maka akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Asgdll.java

D:\TUGAS-PBO\Tugas1>java Asgdll
f : 20.0
f11: 10.0
```

2. Program Java “Asign” merupakan contoh pemrograman sederhana dimana akan mencetak teks ke layar pada program ini, mendeklarasikan variabel ‘i’ dengan tipe data integer yang kemudian akan diberikan nilai 5. Selanjutnya Program akan menggunakan perintah System.out.print("hello\n"); untuk menampilkan teks "hello" yang diikuti dengan baris baru (\n). Setelah itu, program mencetak teks "Ini nilai i :" diikuti dengan nilai variabel i. Ketika dijalankan, program akan menghasilkan output berupa "hello" di baris pertama, dan "Ini nilai i :5" di baris berikutnya sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Asign.java

D:\TUGAS-PBO\Tugas1>java Asign
hello
Ini nilai i :5
```

3. Program ini bertujuan menunjukkan bagaimana berbagai jenis data disimpan dan ditampilkan dalam Java mulai dari bilangan bulat, karakter, dan bilangan desimal. Dalam program, terdapat beberapa variabel antara lain ks, ki, dan kl yang masing-masing bertipe short, int, dan long dimana digunakan untuk menyimpan angka bulat. Selain itu, ada dua variabel karakter, yaitu c yang diberi nilai 65 dan c1 yang langsung diberi nilai 'Z'. Nilai 65 pada variabel c sebenarnya mewakili karakter 'A' dalam tabel ASCII. Untuk bilangan desimal, program menggunakan dua tipe data, yaitu double pada variabel x dan float pada variabel y, di mana double memiliki tingkat presisi lebih tinggi dibandingkan float. Saat program dijalankan, karakter 'A' dan 'Z' ditampilkan dua kali, pertama dalam bentuk karakter dan kedua dalam program mencetak nilai bilangan bulat dengan berbagai tipe data yang berbeda sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac ASIGNi.java

D:\TUGAS-PBO\Tugas1>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. Program ini adalah contoh sederhana penggunaan kelas `Scanner` dalam Java untuk membaca input dari pengguna. Program meminta pengguna memasukkan sebuah angka, lalu angka tersebut disimpan dalam variabel `a` bertipe `int`. Objek `Scanner` yang diberi nama `masukan` digunakan untuk membaca input dari keyboard. Setelah mengetikkan angka dan menekan Enter, program menyimpan angka itu dalam variabel `a`, lalu mencetaknya kembali ke layar. Jika baris `a = masukan.nextInt();` dihapus, program tidak bisa membaca input, sehingga variabel `a` tidak akan memiliki nilai, yang bisa menyebabkan error atau hasil yang tidak diinginkan, sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac BacaData.java

D:\TUGAS-PBO\Tugas1>java BacaData
Contoh membaca dan menulis, ketik nilai integer:
7
Nilai yang dibaca : 7
```

5. Program ini adalah contoh penggunaan `BufferedReader` dalam Java untuk membaca input dari pengguna, baik berupa huruf maupun angka. Saat dijalankan, program pertama-tama menampilkan teks "hello" di layar. Setelah itu, pengguna diminta memasukkan satu huruf, yang kemudian disimpan dalam variabel `cc`. Selanjutnya, pengguna diminta memasukkan sebuah angka, yang dikonversi menjadi bilangan bulat (`int`) dan disimpan dalam variabel `bil`. Setelah menerima kedua input tersebut, program akan menampilkan kembali huruf dan angka yang dimasukkan, lalu mencetak pesan "bye" sebelum selesai. Jika program tidak membaca input dengan benar, bisa terjadi kesalahan sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Bacakar.java

D:\TUGAS-PBO\Tugas1>java Bacakar
hello
baca 1 karakter : H
baca 1 bilangan : 7
H
7
bye
```

6. Program ini mendemonstrasikan cara mengubah tipe data dalam Java melalui proses Casting, yaitu konversi dari satu tipe data ke tipe lainnya. Program ini

bekerja dengan beberapa tipe data, seperti int, float, double, dan char, lalu mengonversinya ke tipe lain dan mencetak hasilnya. Misalnya, angka 5 yang bertipe int dikonversi menjadi 5.0 dalam float, dan angka 6 dalam int dikonversi menjadi 6.0 dalam double. Saat mengonversi angka desimal ke bilangan bulat, seperti dari 2.0 (float) menjadi 2 (int) atau dari 3.14 (double) menjadi 3 (int), bagian desimalnya akan dihilangkan. Selain itu, perbedaan dalam penyimpanan angka juga terlihat saat 3.2 (float) dikonversi ke double, di mana hasilnya menjadi 3.200000047683716 karena adanya perbedaan presisi dalam representasi angka di komputer. Program ini juga menunjukkan bagaimana karakter direpresentasikan dalam bentuk angka di dalam sistem komputer. Sebagai contoh, karakter '5' saat dikonversi ke int menghasilkan 53, karena dalam tabel ASCII, setiap karakter memiliki nilai angka tersendiri. Ketika karakter ini dikonversi ke float atau double, hasilnya tetap 53.0, mengikuti aturan konversi karakter ke angka, Sehingga akan menampilkan output pada gambar dibawah ini

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Casting1.java
```

```
D:\TUGAS-PBO\Tugas1>java Casting1
```

```
5.0
```

```
6.0
```

```
3.200000047683716
```

```
53
```

```
53.0
```

```
53.0
```

```
3
```

```
3.14
```

7. Program ini menjelaskan cara mengubah jenis data di Java, seperti mengonversi teks menjadi angka, angka menjadi teks, dan mengubah angka dari satu jenis ke jenis lainnya. Pertama, program mengubah teks yang berisi angka menjadi angka sebenarnya. Misalnya, teks "67" diubah menjadi 67 dalam bentuk bilangan bulat (int), "45" menjadi 45.0 dalam bentuk desimal (double), dan "100" menjadi 100.0 (float). Ini berguna ketika kita mendapatkan angka dari input pengguna yang awalnya berupa teks dan ingin menggunakannya dalam perhitungan. Selanjutnya, program mengubah angka dan karakter menjadi teks, seperti 9 menjadi "9", huruf '5' menjadi "5", dan angka desimal 3.2 tetap menjadi "3.2". Terakhir, program juga menunjukkan cara mengubah angka dari satu tipe ke tipe lain, misalnya 67 bisa diubah menjadi 67.0 agar bisa digunakan dalam perhitungan dengan angka desimal, atau 9 diubah menjadi 9.0 sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Casting2.java

D:\TUGAS-PBO\Tugas1>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. Program ini menjelaskan cara menggunakan Operator kondisional (ternary operator) `?:` dalam Java untuk memilih nilai berdasarkan kondisi tertentu. Program ini dimulai dengan mendeklarasikan dua variabel integer, yaitu `x = 1` dan `y = 2`, lalu mencetak nilai keduanya. Setelah itu, program mengevaluasi ekspresi `(x < y) ? x : y`, yang berfungsi untuk menentukan nilai berdasarkan hasil perbandingan. Operator ini bekerja dengan cara mengecek kondisi sebelum tanda `?`. Jika kondisi tersebut benar (`true`), maka nilai setelah `?` yang akan digunakan, sedangkan jika salah (`false`), nilai setelah `:` yang akan digunakan. Dalam program ini, kondisi `x < y` atau `1 < 2` bernilai `true`, sehingga hasil ekspresi adalah nilai `x`, yaitu `1`. sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Ekspresi.java

D:\TUGAS-PBO\Tugas1>java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
```

9. Program Ekspresi menunjukkan perbedaan hasil pembagian dalam Java berdasarkan tipe data yang digunakan, yaitu bilangan bulat (`int`) dan bilangan desimal (`float`). Awalnya, `x = 1` dan `y = 2`, yang keduanya bertipe `int`. Saat `x` dibagi `y` menggunakan `x / y`, hasilnya `0`, karena pembagian dalam `int` selalu mengabaikan angka desimal. Namun, ketika `x` dan `y` dikonversi ke `float`, hasil pembagian menjadi `0.5`, karena `float` mampu menyimpan angka desimal. Program kemudian mengubah nilai `x = 10` dan `y = 3`, lalu melakukan pembagian lagi. Dengan `int`, hasilnya tetap `3`, karena angka desimal dibuang. Bahkan jika ditampilkan dalam format `float`, hasilnya tetap `3`, karena perhitungan awalnya dilakukan sebagai `int`. Jika ingin hasil lebih akurat dengan angka desimal, harus digunakan tipe `float` atau `double`, baik dengan deklarasi langsung atau menggunakan casting `((float)x / (float)y)`. sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Ekspresi1.java

D:\TUGAS-PBO\Tugas1>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. Program 'Hello' ini menggunakan `System.out.print()` dan `System.out.println()` untuk mencetak teks ke layar. `print()` digunakan untuk mencetak teks tanpa berpindah ke baris baru, sedangkan `println()` mencetak teks lalu langsung berpindah ke baris berikutnya. Pada baris pertama, `System.out.print("Hello");` mencetak "Hello" tanpa berpindah ke baris baru. Kemudian, `System.out.print("\nHello ");` menggunakan `\n` untuk membuat baris baru, lalu mencetak "Hello " di awal baris tersebut tanpa berpindah ke baris baru setelahnya. Selanjutnya, `System.out.println("World");` mencetak "World" di sebelah "Hello ", kemudian berpindah ke baris baru. Terakhir, `System.out.println("Welcome");` mencetak "Welcome" di baris yang baru. Sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Hello.java

D:\TUGAS-PBO\Tugas1>java Hello
Hello
Hello World
Welcome
```

11. Program Incr ini menunjukkan cara kerja operator increment (`++`) dalam Java, yang digunakan untuk menambah nilai variabel sebanyak 1. Program dimulai dengan mendeklarasikan dua variabel `i` dan `j` bertipe integer. Kemudian, `i` diberi nilai 3, sedangkan `j` diberi nilai `i++`. Karena operator `++` berada setelah variabel (`i++` disebut post-increment), nilai `i` yang lama (3) terlebih dahulu diberikan ke `j`, lalu `i` bertambah menjadi 4. Selanjutnya, program mencetak nilai `i` menggunakan `++i`, yang disebut pre-increment, sehingga nilai `i` bertambah menjadi 5 sebelum dicetak. Sementara itu, nilai `j` tetap 3, karena sebelumnya mendapatkan nilai `i` sebelum `i` bertambah. Output yang dihasilkan adalah "Nilai `i` : 5" dan "Nilai `j` : 3". Sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Incr.java

D:\TUGAS-PBO\Tugas1>java Incr
Nilai i : 5
Nilai j : 3
```

12. Program Oper1 ini mendemonstrasikan penggunaan operator bitwise dalam Java untuk memanipulasi angka pada tingkat biner. Dimulai dengan deklarasi variabel $n = 10$, $x = 1$, dan $y = 2$, program mencetak nilai awalnya sebelum melakukan berbagai operasi bitwise. Operasi bitwise AND (&) antara n dan 8 menghasilkan 8 karena dalam biner hanya bit yang bernilai 1 di kedua angka yang dipertahankan. Operasi bitwise AND dengan negasi (& ~) antara x dan ~8 menghasilkan 1, karena negasi 8 (00001000_2) menjadi -9 (11110111_2). Selanjutnya, operasi left shift (<<) menggeser bit y (2) dua kali ke kiri, mengubah 10_2 menjadi 1000_2 , yang setara dengan 8 dalam desimal. Terakhir, operasi right shift (>>) menggeser y tiga kali ke kanan, menghapus semua bit yang ada sehingga hasilnya menjadi 0. Sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Oper1.java

D:\TUGAS-PBO\Tugas1>java Oper1
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

13. Program Oper2 menggunakan operator bitwise pada dua variabel karakter i dan j , yang diberi nilai 3 dan 4. Karena karakter juga memiliki nilai numerik dalam kode ASCII, operasi dilakukan berdasarkan bentuk biner angka tersebut. Bitwise AND (&) antara 3 dan 4 menghasilkan 0 karena tidak ada bit yang sama-sama bernilai 1. Bitwise OR (|) menghasilkan 7, karena menggabungkan bit yang bernilai 1 dari kedua angka. Bitwise XOR (^) juga menghasilkan 7, karena XOR hanya bernilai 1 jika bitnya berbeda. Kemudian, program menghitung 3^4 menggunakan Math.pow(), yang hasilnya adalah 81.0. Terakhir, operasi bitwise NOT (~) mengubah 3 menjadi -4, karena dalam sistem biner, operasi ini membalik bit dan mengubah angka menjadi negatif. Sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Oper2.java

D:\TUGAS-PBO\Tugas1>java Oper2
i = 3
j = 4
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
```

14. Program Oper3 ini menunjukkan penggunaan operator logika dalam Java, seperti AND (&& dan &), OR (|| dan |), serta penggunaan kondisi boolean dalam pernyataan if. Pada baris pertama, $\text{true} \ \&\& \ \text{true}$ bernilai true, sehingga program mencetak true. Pada baris kedua, meskipun kondisi if menggunakan $\text{true} \ \& \ \text{true}$, perintah di dalamnya justru mencetak $\text{true} \ \& \ \text{false}$, yang bernilai

false, sehingga program mencetak false. Selanjutnya, kondisi if(true) langsung bernilai true, sehingga true dicetak. Lalu, true || true juga tetap bernilai true, sehingga program kembali mencetak true. Terakhir, true | false tetap bernilai true, karena operator OR (|) akan menghasilkan true jika salah satu operand bernilai true. Secara keseluruhan, output yang dihasilkan adalah true, false, true, true, dan true. Sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Oper3.java
D:\TUGAS-PBO\Tugas1>java Oper3
true
false
true
true
true
```

15. Program Oper4 menggunakan operator ternary (? :) untuk menentukan nilai berdasarkan kondisi tertentu. Awalnya, dua variabel i dan j diinisialisasi dengan nilai 0, sedangkan variabel c dan d bertipe char dengan nilai 8 dan 10. Operator ternary pertama membandingkan c dan d, di mana d lebih besar sehingga variabel e diisi dengan nilai 10. Kemudian, operator ternary kedua membandingkan i dan j, karena nilainya sama (0), variabel k juga mendapatkan nilai 0. Setelah itu, nilai i diubah menjadi 2 dan j menjadi 3, lalu kembali dibandingkan menggunakan operator ternary dengan post-increment (i++ dan j++), sehingga k mendapatkan nilai 4 karena j lebih besar setelah operasi increment. Hasil akhirnya yang ditampilkan adalah "Nilai e = 10", "Nilai k = 0", dan "Nilai k = 4". Sehingga akan menampilkan output seperti pada gambar di bawah ini.

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Oper4.java
D:\TUGAS-PBO\Tugas1>java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4
```

16. Program Java ini mendemonstrasikan penggunaan berbagai operator, termasuk operator logika, aritmatika, dan perbandingan. Program dimulai dengan mendeklarasikan variabel bertipe boolean (Bool1, Bool2, dan TF), integer (i, j, dan hsl), serta float (x, y, dan res). Pada bagian operasi logika, program menggunakan operator AND (&&), OR (||), NOT (!), dan XOR (^). Operator AND menghasilkan false karena kedua nilai yang dibandingkan tidak sama-sama true, sedangkan OR menghasilkan true karena salah satu nilainya true. Operator NOT membalik nilai boolean, dan XOR menghasilkan true jika hanya salah satu dari dua nilai yang true. Selanjutnya, program melakukan operasi aritmatika seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus. Namun, pembagian antara dua bilangan integer (5 / 2) menghasilkan 2, bukan 2.5, karena tipe datanya integer. Operasi perbandingan juga dilakukan

untuk membandingkan nilai integer dan float dengan operator seperti ==, !=, <, >, <=, dan >=. Misalnya, 5 != 2 menghasilkan true, sedangkan 5 == 2 menghasilkan false. Sehingga menghasilkan output seperti pada gambar dibawah ini:

OUTPUT:

```
D:\TUGAS-PBO\Tugas1>javac Oprator.java

D:\TUGAS-PBO\Tugas1>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
```