

Tugas 1

Makassar, 21 Februari 2025

## **TEORI PEMROGRAMAN BERBASIS OBJEK**



Nama : Nur Aqidah Safanikiah  
Stambuk : 13020230121  
Frekuensi : TI\_PBO-12 (B2)  
Dosen : Mardiyah Hasnawi, S.Kom.,M.T.,MTA

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

### 1. Output Program

```
D:\xampp\htdocs\Code\PBO\Tugas 1>javac Asgdll.java

D:\xampp\htdocs\Code\PBO\Tugas 1>java Asgdll
f : 20.0
fill: 10.0
```

Penjelasan :

Dalam program Asgdll ini, variabel bertipe float bernama **f** dideklarasikan dan diinisialisasi dengan nilai 20.0, sedangkan variabel double **fill** dideklarasikan tanpa nilai awal. Kemudian, **fill** diisi dengan nilai 10.0. Untuk menampilkan hasilnya, digunakan **System.out.println**, yang mencetak nilai **f** dan **fill** dengan format yang rapi, di mana **\n** digunakan untuk pindah baris.

### 2. Output Program

```
D:\xampp\htdocs\Code\PBO\Tugas 1>javac Assign.java

D:\xampp\htdocs\Code\PBO\Tugas 1>java Assign
hello
Ini nilai i :5
```

Penjelasan :

Dalam program Asgdll. Mendefinisikan variabel bertipe data integer bernama "**i**". Kemudian, program mencetak kata "hello" diikuti dengan pindah baris menggunakan **System.out.print("hello\n")**. Setelah itu, variabel **i** diinisialisasi dengan nilai 5. Terakhir, program mencetak nilai dari **i** dengan menggunakan **System.out.println("Ini nilai i : " + i)**, yang akan menampilkan "Ini nilai i : 5" di layar.

### 3. Output Program

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

D:\xampp\htdocs\Code\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 :

Pada program ASIGNi merupakan contoh sederhana dalam Java yang menunjukkan berbagai tipe data dasar seperti short, int, long, char, double, dan float. Program ini mendeklarasikan beberapa variabel, di mana karakter dapat diisi dengan angka (seperti `c = 65` untuk 'A') atau huruf (seperti `c1 = 'Z'`). Kemudian, program mencetak nilai-nilai variabel tersebut, baik sebagai huruf maupun angka, serta menampilkan nilai dari semua variabel angka dengan menggunakan perintah *System.out.println*.

#### 4. Output Program

```
D:\xampp\htdocs\Code\PB0\Tugas 1>javac BacaData.java

D:\xampp\htdocs\Code\PB0\Tugas 1>java BacaData
Contoh membaca dan menulis, ketik nilai integer: 5
8
Nilai yang dibaca : 8
```

Penjelasan :

Program **BacaData** menggunakan kelas *Scanner* dari Java untuk membaca input angka dari pengguna. Pertama program menampilkan pesan untuk memasukkan nilai, kemudian membuat objek *Scanner* untuk membaca input. **Method nextInt()** digunakan untuk mengambil angka yang diketikkan pengguna dan menyimpannya ke variabel **a**. Terakhir, program menampilkan nilai yang telah dibaca tersebut.

#### 5. Output Program

```
D:\xampp\htdocs\Code\PB0\Tugas 1>javac Bacakar.java

D:\xampp\htdocs\Code\PB0\Tugas 1>java Bacakar
hello
baca 1 karakter : I
baca 1 bilangan : 8
I
8
bye
```

Penjelasan :

Pada program **Bacakar** menggunakan **BufferedReader** untuk membaca input dari pengguna. Program mendeklarasikan variabel karakter **cc** dan integer **bil**, lalu membuat dua objek **BufferedReader** untuk membaca input. Setelah menampilkan "*hello*", program meminta dan membaca satu karakter dari pengguna menggunakan **readLine().charAt(0)** dan satu bilangan dengan **Integer.parseInt(dataIn.readLine())**. Akhirnya, program menampilkan karakter dan bilangan yang dibaca, diikuti dengan pesan "*bye*".

#### 6. Output Program

```
D:\xampp\htdocs\Code\PBO\Tugas 1>javac Casting1.java

D:\xampp\htdocs\Code\PBO\Tugas 1>java Casting1
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14
```

Penjelasan :

Pada program **Casting1** menunjukkan bagaimana konsep casting (konversi) tipe data primitif di Java. Di dalam program, beberapa variabel dideklarasikan, termasuk integer, float, char, dan double. Program ini mencetak hasil konversi dari satu tipe data ke tipe data lainnya. Misalnya, integer **a** diubah menjadi float, dan integer **b** diubah menjadi double. Karakter **g** juga dikonversi menjadi integer, float, dan double, yang menghasilkan nilai ASCII dari karakter tersebut. Program ini mengilustrasikan dua jenis konversi: *widening* (ke tipe dengan rentang lebih besar) dan *narrowing* (ke tipe dengan rentang lebih kecil).

## 7. Output Program

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

D:\xampp\htdocs\Code\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** menunjukkan konversi tipe data dalam bahasa pemrograman Java antar berbagai tipe data. Program ini dimulai dengan mendeklarasikan beberapa variabel dengan tipe data **primitif** seperti **integer**, **float**, **character**, dan **double**, serta variabel bertipe **String**. Selanjutnya, program melakukan serangkaian konversi dari **String** ke tipe data primitif menggunakan metode parsing seperti **Integer.parseInt()**, **Double.parseDouble()**, dan **Float.parseFloat()**, dimana nilai-nilai string seperti "67", "45", dan "100" diubah menjadi representasi numeriknya. Program kemudian melanjutkan dengan melakukan konversi sebaliknya, yaitu dari tipe data primitif ke **String** menggunakan metode **String.valueOf()**, yang mengubah

nilai numerik dan karakter menjadi representasi string. Di bagian akhir, program menunjukkan konversi yang lebih kompleks antar tipe numerik menggunakan kelas pembungkus (*wrapper classes*), seperti mengkonversi integer ke double melalui **Integer.valueOf(b).doubleValue()**. Setiap langkah konversi diikuti dengan output ke konsol yang menampilkan hasil dari proses tersebut, memberikan gambaran jelas tentang bagaimana data dapat ditransformasikan antar berbagai tipe dalam Java tanpa kehilangan nilai aslinya atau dengan perubahan yang dapat diprediksi.

## 8. Output Program

```
D:\xampp\htdocs\Code\PBO\Tugas 1>javac Ekspresi.java

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

Penjelasan:

Pada program Ekspresi menunjukkan bagaimana penggunaan operator kondisional di Java. Program ini mendeklarasikan dua variabel integer ( $x=1$  dan  $y=2$ ), menampilkan nilai keduanya, lalu menggunakan operator ternary ( $x < y$ ) ?  $x : y$  untuk mengevaluasi kondisi. Karena  $x$  memang lebih kecil dari  $y$ , kondisi bernilai benar dan hasil ekspresi adalah nilai  $x$  (yaitu 1). Operator kondisional ini berfungsi seperti pernyataan if-else dalam bentuk singkat, memeriksa kondisi dan mengembalikan nilai pertama jika benar atau nilai kedua jika salah.

## 9. Output Program

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

D:\xampp\htdocs\Code\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
```

Penjelasan:

Pada program Ekspresi1 menunjukkan cara kerja pembagian integer dan casting (konversi) di Java. Program menggunakan variabel  $x=1$ ,  $y=2$  dan menunjukkan bahwa pembagian integer ( $x/y$ ) menghasilkan 0 karena hasil dibulatkan ke bawah. Dengan mengkonversi nilai ke float (melalui variabel  $fx$ ,  $fy$  atau langsung dengan casting  $(float)x/(float)y$ ), hasil pembagian menjadi 0.5 yang lebih akurat. Ketika nilai diubah menjadi  $x=10$  dan  $y=3$ , pembagian

integer menghasilkan 3, menunjukkan perbedaan antara pembagian integer dan float di Java.

#### 10. Output Program

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

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

Penjelasan :

Pada program Hello menunjukkan dasar penggunaan fungsi output di Java. Program ini berisi metode main yang mengeksekusi beberapa pernyataan untuk menampilkan teks ke layar. Pertama, program menggunakan `System.out.print("Hello")` untuk menampilkan kata "Hello" tanpa baris baru. Kemudian, `System.out.print("\nHello ")` menampilkan baris baru diikuti kata "Hello " dengan spasi di akhir. Selanjutnya, `System.out.println("World")` menampilkan kata "World" dan menambahkan baris baru secara otomatis. Terakhir, `System.out.println("Welcome")` menampilkan kata "Welcome" diikuti baris baru. Hasilnya adalah tampilan teks "Hello" pada baris pertama, "Hello World" pada baris kedua, dan "Welcome" pada baris ketiga.

#### 11. Output Program

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

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

Penjelasan :

Program **Incr** ini menunjukkan penggunaan operator increment (`++`). Variabel `i` diinisialisasi dengan nilai 3, dan kemudian `j` diisi dengan nilai `i` sebelum increment, sehingga `j` menjadi 3. Setelah itu, `i` ditingkatkan menjadi 4. Saat mencetak nilai, `i` diincrement lagi menggunakan prefix (`++i`), sehingga nilainya menjadi 5. Program akan menampilkan output "Nilai i : 5" dan "Nilai j : 3".

#### 12. Output Program

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

D:\xampp\htdocs\Code\PBO\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 ini menunjukkan penggunaan operator *bitwise* dan *shift* pada variabel integer. Tiga variabel diinisialisasi: `n` dengan 10 (1010 dalam biner), `x` dengan 1 (0001), dan `y` dengan 2 (0010). Program mencetak nilai variabel tersebut, lalu melakukan beberapa operasi: `n & 8` menghasilkan 8, `x & ~8` menghasilkan 1, `y << 2` menggeser `y` dua posisi ke kiri menjadi 8, dan `y >> 3` menggeser `y` tiga posisi ke kanan menjadi 0. Hasil dari setiap operasi ditampilkan, menunjukkan cara kerja operator bitwise dan shift.

### 13. Output Program

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

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

Penjelasan :

Program Oper2 menunjukkan penggunaan operator *bitwise* dan relasional pada tipe data `char`. Variabel `i` dan `j` diinisialisasi dengan nilai `3` dan `4`, lalu dilakukan operasi bitwise seperti AND (`&`), OR (`|`), XOR (`^`), dan NOT (`~`). Hasilnya mencerminkan manipulasi bit biner dari kedua nilai tersebut. Selain itu, program menggunakan `Math.pow` untuk menghitung pemangkatan `3^4`. Semua hasil operasi dicetak ke layar, memberikan ilustrasi cara kerja operator bitwise dan fungsi pemangkatan dalam Java.

### 14. Output Program

```
D:\xampp\htdocs\Code\PB0\Tugas 1>javac Oper3.java

D:\xampp\htdocs\Code\PB0\Tugas 1>java Oper3
true
false
true
true
true
```

Penjelasan :

Program Oper3 ini menunjukkan penggunaan operator logika dan bitwise dalam beberapa kondisi. Pada pernyataan `if (true && true)`, program mencetak `true` karena kedua operand bernilai `true`. Selanjutnya, `if (true & true)` mencetak hasil dari `true & false`, yang menghasilkan `false`, karena operator AND *bitwise* mengevaluasi kedua operand tanpa mempertimbangkan nilai boolean. Pernyataan `if (true)` selalu terpenuhi, sehingga mencetak `true`. Pada `if (true || true)`, program mencetak `true` karena setidaknya satu operand bernilai `true`. Terakhir, `if (true | false)` mencetak hasil dari `true | false`, yang menghasilkan `true`. Program ini memberikan gambaran jelas tentang perbedaan antara operator logika dan *bitwise* dalam Java.

### 15. Output Program

```
D:\xampp\htdocs\Code\PB0\Tugas 1>javac Oper4.java

D:\xampp\htdocs\Code\PB0\Tugas 1>java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4
```

Penjelasan :

Program Oper4 ini menunjukkan penggunaan operator ternary untuk menentukan nilai berdasarkan kondisi. Dua variabel integer, `i` dan `j`, diinisialisasi dengan 0, dan dua karakter, `c` dan `d`, diinisialisasi dengan 8 dan 10. Operator ternary membandingkan `c` dan `d`, sehingga `e` bernilai 10, dan `k` diisi dengan nilai `j`, yaitu 0. Setelah mencetak nilai `e` dan `k`, nilai `i` diubah menjadi 2 dan `j` menjadi 3. Pada pernyataan `k = ((i++ > j++) ? i : j);`, nilai yang dibandingkan adalah 2 dan 3, sehingga `k` diisi dengan nilai `j`, yang menjadi 4 setelah increment.

### 16. Output Program



```
D:\xampp\htdocs\Code\PBO\Tugas 1>javac Oprator.java
```

```
D:\xampp\htdocs\Code\PBO\Tugas 1>java Oprator
```

```
Bool1 && Bool2 = false
```

```
Bool1 || Bool2 = true
```

```
!Bool1 = false
```

```
Bool1 ^ Bool2 = true
```

```
Nilai i = 5, j = 2
```

```
i + j = 7
```

```
i - j = 3
```

```
i / j = 2
```

```
i * j = 10
```

```
i % j = 1
```

```
Nilai i = 5
```

```
Nilai x = 5.0, y = 5.0
```

```
x + y = 10.0
```

```
x - y = 0.0
```

```
x / y = 1.0
```

```
x * y = 25.0
```

```
Operasi Relasional Integer:
```

```
i == j : false
```

```
i != j : true
```

```
i < j : false
```

```
i > j : true
```

```
i <= j : false
```

```
i >= j : true
```

```
Operasi Relasional Float:
```

```
x != y : false
```

```
x < y : false
```

```
x > y : false
```

```
x <= y : true
```

```
x >= y : true
```

```
Nilai i = 5
```

Penjelasan :

Program Oprator menunjukkan pengoperasian variabel bertipe dasar, termasuk boolean dan numerik. Di dalam metode `main`, variabel boolean (`Bool1` dan `Bool2`) dan variabel numerik (integer `i`, `j`, dan `hsl`, serta float `x` dan `y`) dideklarasikan. Program melakukan operasi logika pada variabel boolean menggunakan operator AND (`&&`), OR (`||`), NOT (`!`), dan XOR (`^`). Selanjutnya, berbagai operasi aritmetika (penjumlahan, pengurangan, pembagian, perkalian, dan modulo) dilakukan pada variabel integer dan float, dengan hasil disimpan dalam variabel `hsl` dan `res`. Terakhir, program

melakukan operasi relasional untuk membandingkan nilai variabel, menyimpan hasilnya dalam variabel boolean `TF`. Meskipun tidak ada output yang dicetak, program ini memberikan gambaran tentang berbagai operasi yang dapat dilakukan pada variabel bertipe dasar dalam Java.

