

**PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Virkayanti Andani Putri
Stambuk : 13020220298
Dosen : Mardiyah Hasnawi, S.Kom.,M.T.,MTA.
Asisten 1 :
Asisten 2 :

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

2025

1. Asgdll.java

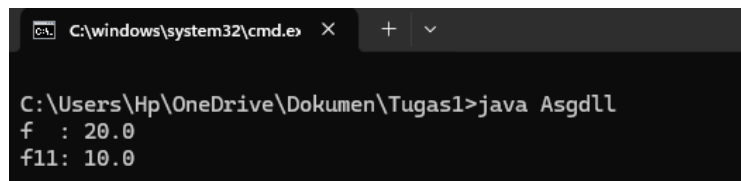
➤ Deklarasi dan Inisialisasi Variabel

- `float f = 20.0f;`
Variabel `f` bertipe `float` dengan nilai awal 20.0.
- `double fll;`
Variabel `fll` bertipe `double`, namun belum diinisialisasi.
- `fll = 10.0f;`
Variabel `fll` diberi nilai 10.0 yang berasal dari tipe `float`.

➤ Menampilkan Output

`System.out.println ("f : "+f + "\nf11: "+fll);`

- `f` memiliki nilai 20.0 (`float`).
- `fll` memiliki nilai 10.0 (`double`).
- `"\n"` adalah karakter newline (baris baru), sehingga teks berikutnya tampil di baris baru.
- Output akhirnya ditampilkan sebagai:



```
C:\windows\system32\cmd.exe X + v
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Asgdll
f : 20.0
f11: 10.0
```

2. Assign.java

➤ Deklarasi dan Inisialisasi Variabel

Pada program `Assign`, terdapat satu variabel yang dideklarasikan:

- `int i;` Deklarasi variabel `i` dengan tipe data `integer`.
- `i = 5;` Inisialisasi variabel `i` dengan nilai **5**.

Variabel `i` digunakan untuk menyimpan nilai `integer` yang nantinya akan ditampilkan pada output.

➤ Menampilkan Output

Program ini menggunakan dua perintah output:

`System.out.print("hello\n");`

- Mencetak "hello" ke layar.
- `\n` adalah karakter newline yang menyebabkan teks berikutnya muncul di baris baru.

`System.out.println("Ini nilai i : " + i);`

- Mencetak "Ini nilai i : 5", karena nilai `i` adalah 5.
- Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Assign
hello
Ini nilai i :5
```

3. ASIGNi.java

- Deklarasi dan Inisialisasi Variabel
 - short ks = 1; Bilangan bulat kecil
 - int ki = 1; Bilangan bulat
 - long kl = 10000; Bilangan bulat panjang
 - char c = 65; Huruf 'A' (65 dalam ASCII)
 - char c1 = 'Z'; Huruf 'Z'
 - double x = 50.2f; Bilangan desimal dengan presisi tinggi
 - float y = 50.2f; Bilangan desimal
- Menampilkan Output
 - System.out.println("Karakter = "+ c); Mencetak "A"
 - System.out.println("Karakter = "+ c1); Mencetak "Z"
 - Menampilkan bilangan bulat (short, int, long)
 - Menampilkan bilangan real (double lebih presisi daripada float)
 - Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\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. BacaData.java

- Deklarasi dan Inisialisasi Variabel
 - int a; Variabel untuk menyimpan input angka.
 - Scanner masukan; Variabel untuk membaca input dari pengguna.
- Menampilkan Output & Membaca Input
 - System.out.print("Contoh membaca dan menulis, ketik nilai integer:\n");
Menampilkan teks instruksi ke layar.
 - masukan = new Scanner(System.in);
Membuat objek Scanner untuk membaca input dari keyboard.
 - a = masukan.nextInt();
Membaca angka yang diketik pengguna dan menyimpannya dalam variabel a.
 - System.out.print("Nilai yang dibaca : "+ a);
Menampilkan angka yang telah dibaca dari input.
 - Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java BacaData
Contoh membaca dan menulis, ketik nilai integer:
7
Nilai yang dibaca : 7
```

5. Bacakar.java

- Deklarasi dan Inisialisasi Variabel
 - Variabel untuk menyimpan input:
 - char cc; Menyimpan satu karakter yang dibaca dari input.
 - int bil; Menyimpan bilangan bulat yang dibaca dari input.
 - Deklarasi untuk membaca input dari keyboard:
 - InputStreamReader isr = new InputStreamReader(System.in);
 - BufferedReader dataIn = new BufferedReader(isr);
 - BufferedReader datAIn = new BufferedReader(new InputStreamReader(System.in));
 Digunakan untuk membaca input dari pengguna.
- Menampilkan Output & Membaca Input
 - System.out.print("hello\n"); Menampilkan teks "hello".
 - System.out.print("baca 1 karakter : "); Menampilkan prompt untuk input karakter.
 - cc = dataIn.readLine().charAt(0); Membaca satu karakter dari input.
 - System.out.print("baca 1 bilangan : "); Menampilkan prompt untuk input bilangan.
 - bil = Integer.parseInt(datAIn.readLine()); Membaca angka dari input dan mengubahnya ke tipe int.
 - System.out.print(cc + "\n" + bil + "\n"); Menampilkan karakter dan angka yang telah dibaca.
 - System.out.print("bye \n"); Menampilkan teks "bye".
 - Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Bacakar
hello
baca 1 karakter : a
baca 1 bilangan : 1
a
1
bye
```

6. Casting1.java

- Deklarasi dan Inisialisasi Variabel
 - Bilangan bulat (int)
 - int a = 5, b = 6;

- Bilangan desimal (float dan double)

float d = 2.f, e = 3.2f;

double k = 3.14;

- Karakter (char)

char g = '5';

➤ Menampilkan Output dengan Type Casting

- System.out.println((float)a);
int a (5) diubah ke float, hasilnya 5.0.
- System.out.println((double)b);
int b (6) diubah ke double, hasilnya 6.0.
- System.out.println((int)d);
float d (2.0) diubah ke int, hasilnya 2 (dibulatkan ke bawah).
- System.out.println((double)e);
float e (3.2) diubah ke double, hasilnya 3.200000047683716 karena presisi double lebih tinggi.
- System.out.println((int)g);
char g ('5') diubah ke int, hasilnya 53 (nilai ASCII dari karakter '5').
- System.out.println((float)g);
char g ('5') diubah ke float, hasilnya 53.0 (nilai ASCII dalam bentuk float).
- System.out.println((double)g);
char g ('5') diubah ke double, hasilnya 53.0 (nilai ASCII dalam bentuk double).
- System.out.println((int)k);
double k (3.14) diubah ke int, hasilnya 3 (dibulatkan ke bawah).
- System.out.println((float)k);
double k (3.14) diubah ke float, hasilnya 3.14
- Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Casting1
5.0
6.0
2
3.200000047683716
53
53.0
53.0
3
3.14
```

7. Casting2.java

➤ Deklarasi dan Inisialisasi Variabel

- Bilangan Bulat (int)
int a = 8, b = 9;
- Bilangan Desimal (float dan double)
float d = 2.f, e = 3.2f;
double k = 3.14;

- Karakter (char)
char g = '5';
 - Tipe String (String)
String n = "67", m = "45", l = "100";
- Menampilkan Output dengan Konversi Tipe Data
- Konversi dari String ke Tipe Numerik
 - a = Integer.parseInt(n); // String "67" → int 67
 - k = Double.parseDouble(m); // String "45" → double 45.0
 - d = Float.parseFloat(l); // String "100" → float 100.0
 - Konversi dari Tipe Numerik ke String
 - n = String.valueOf(b); // int 9 → String "9"
 - m = String.valueOf(g); // char '5' → String "5"
 - l = String.valueOf(e); // float 3.2 → String "3.2"
 - Konversi dari Tipe Numerik ke Tipe Lain
 - k = Double.valueOf(a).intValue(); // int 67 → double → int 67
 - double c = Integer.valueOf(b).doubleValue(); // int 9 → double 9.0
 - Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\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. Ekspresi.java

- Deklarasi dan Inisialisasi Variabel
- Program ini mendeklarasikan dua variabel bertipe int:
- ```
int x = 1;
int y = 2;
```
- x diberi nilai 1
  - y diberi nilai 2
- Menampilkan Output
- Program mencetak nilai dari x dan y:
- ```
System.out.print("x = " + x + "\n");
System.out.print("y = " + y + "\n");
```
- Evaluasi Ekspresi dengan Operator Ternary
- Kode berikut menggunakan operator ternary (? :):
- ```
System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y));
```
- (x < y) ? x : y berarti:
    - Jika x < y bernilai true, maka hasilnya adalah x
    - Jika false, maka hasilnya adalah y
  - Karena 1 < 2 bernilai true, maka hasilnya adalah x (yaitu 1).

- Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
```

## 9. Ekspresi1.java

### ➤ Deklarasi dan Inisialisasi Variabel

- Deklarasi:
  - a. int x, y; mendeklarasikan variabel x dan y dengan tipe data int.
  - b. float fx, fy; mendeklarasikan variabel fx dan fy dengan tipe data float.
- Inisialisasi:
  - a. x = 1; dan y = 2; memberi nilai awal 1 dan 2 ke variabel x dan y.
  - b. fx dan fy belum diberikan nilai awal, tetapi nanti akan diisi dengan nilai x dan y.

### ➤ Menampilkan Output

Menampilkan hasil operasi pembagian (/) dalam berbagai format.

#### a. Operasi Pembagian dengan Tipe int

```
System.out.print ("x/y (format integer) = "+ x/y);
```

```
System.out.print ("\nx/y (format float) = "+ x/y);
```

- x / y adalah pembagian antara dua bilangan int.
- Hasil 1 / 2 adalah 0, karena dalam int hasil desimal akan dibulatkan ke bawah.

#### b. Konversi int ke float sebelum Operasi

```
fx = x;
```

```
fy = y;
```

```
System.out.print ("\nx/y (format integer) = "+ fx/fy);
```

```
System.out.print ("\nx/y (format float) = "+ fx/fy);
```

- fx = x; dan fy = y; mengubah nilai x dan y menjadi float.
- fx / fy sekarang menjadi 1.0 / 2.0, sehingga hasilnya adalah 0.5.

#### c. Menggunakan Casting ke float dalam Operasi

```
System.out.print ("\nfloat(x)/float(y) (format integer) = "+ (float)x/(float)y);
```

```
System.out.print ("\nfloat(x)/float(y) (format float) = "+ (float)x/(float)y);
```

- (float)x / (float)y mengubah x dan y menjadi float sebelum operasi.
- Hasil 1.0 / 2.0 tetap 0.5.

#### d. Mengubah Nilai x dan y, lalu Menampilkan Hasil Baru

```
x = 10;
```

```
y = 3;
```

```
System.out.print ("\nx/y (format integer) = "+ x/y);
```

```
System.out.print ("\nx/y (format float) = "+ x/y);
```

- Nilai x diubah menjadi 10 dan y menjadi 3.

- 10 / 3 dalam int menghasilkan 3 karena hasil desimal dibulatkan ke bawah.
- 10 / 3 tetap 3 karena masih dalam format int.
- Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\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. PrintHello.java

Program ini menggunakan System.out.print() dan System.out.println() untuk mencetak teks ke layar. Berikut penjelasan per barisnya:

- Baris 1

```
System.out.print("Hello");
```

System.out.print() digunakan untuk mencetak teks tanpa berpindah ke baris baru.

- Baris 2

```
System.out.print("\nHello ");
```

\n (newline) digunakan untuk membuat baris baru sebelum teks "Hello ".

- Baris 3

```
System.out.println("World");
```

System.out.println() mencetak teks dan otomatis berpindah ke baris baru setelahnya.

- Baris 4

```
System.out.println("Welcome");
```

System.out.println() mencetak "Welcome" lalu berpindah ke baris baru.

- Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Hello
Hello
Hello World
Welcome
```

## 11. Incr.java

- Deklarasi dan Inisialisasi Variabel

Program ini menggunakan dua variabel yaitu i dan j.

- Deklarasi



int i, j; Mendeklarasikan dua variabel bertipe integer (int), yaitu i dan j.

- Inisialisasi Variabel
  - i = 3; Memberikan nilai awal 3 kepada variabel i.
  - j = i++;
    - Operator post-increment (i++) digunakan.
    - Nilai i sebelum peningkatan (3) diberikan ke j.
    - Setelah itu, i bertambah menjadi 4.

➤ Menampilkan Output

Bagian berikut mencetak nilai i dan j:

```
System.out.println ("Nilai i : " + (++i) +
"\nNilai j : " + j);
```

- ++i : Operator pre-increment, yaitu meningkatkan nilai i terlebih dahulu, baru mencetaknya.
- \n : Membuat baris baru agar output lebih rapi.

Proses Nilai Variabel

1. Sebelum cetak output:
  - i = 4 (karena sebelumnya i++ telah menambah nilai i menjadi 4)
  - j = 3 (karena j = i++ menyimpan nilai i sebelum kenaikan)
2. Saat perintah ++i dieksekusi:
  - i bertambah menjadi 5 sebelum dicetak.
- Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Incr
Nilai i : 5
Nilai j : 3
```

## 12. Oper1.java

➤ Deklarasi dan Inisialisasi Variabel

Program ini menggunakan tiga variabel integer, yaitu n, x, dan y.

- Deklarasi Variabel
  - int n, x, y; → Mendeklarasikan variabel n, x, dan y bertipe integer (int).
- Inisialisasi Variabel
  - n = 10; → n diinisialisasi dengan nilai 10 (biner: 1010).
  - x = 1; → x diinisialisasi dengan nilai 1 (biner: 0001).
  - y = 2; → y diinisialisasi dengan nilai 2 (biner: 0010).

➤ Menampilkan Output

Bagian berikut mencetak nilai variabel dan hasil operasi bitwise:

```
System.out.println ("n = " + n);
System.out.println ("x = " + x);
System.out.println ("y = " + y);
```

Operasi Bitwise dan Shift

1. Operasi AND (&)  
System.out.println ("n & 8 = " + (n & 8));
  - n = 10 → biner: 1010

- $8 = 8 \rightarrow$  biner: 1000
  - Operasi AND (&):  
1010  
1000  
----  
1000 (8 dalam desimal)
  - Output:  $n \& 8 = 8$
2. Operasi NOT (~) dan AND (&)  
System.out.println ("x & ~ 8 = " + (x & ~8));
- $x = 1 \rightarrow$  biner: 0001
  - $8 = 8 \rightarrow$  biner: 1000, maka NOT ~8:  
~1000 = 0111
  - Operasi AND (&):  
0001  
0111  
----  
0001 (1 dalam desimal)
  - Output:  $x \& \sim 8 = 1$
3. Operasi Left Shift (<<)  
System.out.println ("y << 2 = " + (y << 2));
- $y = 2 \rightarrow$  biner: 0010
  - Left shift ( $y \ll 2$ ) menggeser ke kiri 2 bit:  
0010 << 2 = 1000
  - 1000 dalam desimal = 8.
  - Output:  $y \ll 2 = 8$
4. Operasi Right Shift (>>)  
System.out.println ("y >> 3 = " + (y >> 3));
- $y = 2 \rightarrow$  biner: 0010
  - Right shift ( $y \gg 3$ ) menggeser ke kanan 3 bit:  
0010 >> 3 = 0000
  - 0000 dalam desimal = 0.
  - Output:  $y \gg 3 = 0$
  - Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Oper1
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

### 13. Oper2.java

- Deklarasi dan Inisialisasi Variabel
  - Deklarasi variabel:  
Variabel i dan j dideklarasikan sebagai tipe char, yang sebenarnya lebih cocok untuk menyimpan karakter daripada angka.
  - Inisialisasi variabel:  
i = 3; // 3 dalam desimal (00000011 dalam biner)

j = 4; // 4 dalam desimal (00000100 dalam biner)  
Variabel i diisi dengan nilai 3 dan j dengan nilai 4.

➤ Menampilkan Output

Setiap baris kode System.out.println() digunakan untuk mencetak hasil operasi berikut:

1. Menampilkan nilai variabel i dan j  
System.out.println("i = " + (int) i);  
System.out.println("j = " + j);  
Output:  
i = 3  
j = 4
2. Operasi AND  
System.out.println("i & j = " + (i & j));
  - i & j berarti 00000011 & 00000100 = 00000000
  - Hasilnya: 0  
i & j = 0
3. Operasi OR  
System.out.println("i | j = " + (i | j));
  - i | j berarti 00000011 | 00000100 = 00000111
  - Hasilnya: 7  
Output:  
i | j = 7
4. Operasi XOR  
System.out.println("i ^ j = " + (i ^ j));
  - i ^ j berarti 00000011 ^ 00000100 = 00000111
  - Hasilnya: 7  
Output:  
i ^ j = 7
5. Pemangkatan (Exponentiation)  
System.out.println(Math.pow(i, j));
  - Math.pow(i, j) menghitung i pangkat j, yaitu  $3^4 = 81.0$   
Output:  
81.0
6. Negasi Bitwise  
System.out.println(" ~i = " + ~i);
  - ~i membalik semua bit dari i = 3 (00000011), menjadi 11111100 dalam representasi dua komplemen.
  - Nilai tersebut setara dengan -4 dalam desimal.
  - Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Oper2
i = 3
j =
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
```

## 14. Oper3.java

Menggunakan pernyataan if dengan operasi logika pada nilai boolean (true dan false).

Baris 1: `if (true && true) { System.out.println(true && true); }`

- `true && true` adalah true karena kedua operand bernilai true.
- `System.out.println(true && true);` mencetak true.

Output: true

Baris 2: `if (true & true) { System.out.println(true & false); }`

- `true & true` adalah true, sehingga perintah dalam blok if akan dijalankan.
- `System.out.println(true & false);`
  - `true & false` menggunakan operator `&` (bitwise AND) yang mengevaluasi kedua operand.
  - `true & false` menghasilkan false, sehingga dicetak false.

Output: false

Baris 3: `if (true) { System.out.println(true); }`

- `if (true)` selalu benar, sehingga perintah dalam blok akan dijalankan.
- `System.out.println(true);` mencetak true.

Output: true

Baris 4: `if (true || true) { System.out.println(true); }`

- `true || true` menggunakan operator OR .
- Karena salah satu operand bernilai true, ekspresi menghasilkan true.
- `System.out.println(true);` mencetak true.

Output: true

Baris 5: `if (true | false) { System.out.println(true | false); }`

- `true | false` menggunakan operator `|` (bitwise OR).
- `true | false` menghasilkan true.
- `System.out.println(true | false);` mencetak true.

Output: true

- Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Oper3
true
false
true
true
true
```

## 15. Oper4.java

### ➤ Deklarasi dan Inisialisasi Variabel

Program Oper4 mendeklarasikan beberapa variabel sebagai berikut:

- `int i = 0;` Variabel `i` bertipe integer dan diberi nilai 0.
- `int j = 0;` Variabel `j` bertipe integer dan diberi nilai 0.
- `char c = 8;` Variabel `c` bertipe karakter dengan nilai 8 (karakter ASCII dengan kode 8).

- `char d = 10;` Variabel `d` bertipe karakter dengan nilai 10 (karakter ASCII dengan kode 10).
- `int e = (((int)c > (int)d) ? c : d);`
  - Dibandingkan `c` dan `d` dalam bentuk integer.
  - `c = 8` dan `d = 10`, sehingga `e = d = 10`.
- `int k = ((i > j) ? i : j);`
  - `i = 0` dan `j = 0`, sehingga `k = 0` karena kedua nilai sama.

Setelah deklarasi awal, program mengubah nilai:

- `i = 2;`
- `j = 3;`
- `k = ((i++ > j++) ? i : j);`
  - `i++ > j++ → 2 > 3 (false)`
  - Maka nilai `k = j` setelah `j++`, yaitu 4.

#### ➤ Menampilkan Output

Setiap perintah `System.out.print` akan mencetak hasil:

1. `System.out.print("Nilai e = "+ e);`
    - `e = 10`, sehingga output:  
Nilai e = 10
  2. `System.out.print("\nNilai k = "+ k);`
    - `k = 0`, sehingga output:  
Nilai k = 0
  3. Setelah perubahan nilai `i` dan `j`, perintah terakhir:
    - `System.out.print("\nNilai k = "+ k);`
    - `k = 4`, sehingga output:  
Nilai k = 4
- Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4
```

## 16. Operator.java

### ➤ Deklarasi dan Inisialisasi Variabel

Pada program Operator, terdapat beberapa variabel yang dideklarasikan dan diinisialisasi:

Variabel Boolean

- `boolean Bool1, Bool2, TF;`
  - `Bool1 = true;`
  - `Bool2 = false;`
  - `TF` digunakan untuk menyimpan hasil operasi logika.

Variabel Numerik Integer

- `int i, j, hsl;`

- `i = 5;`
- `j = 2;`
- `hsl` digunakan untuk menyimpan hasil operasi numerik integer.

#### Variabel Numerik Float

- `float x, y, res;`
  - `x = 5;`
  - `y = 5;`
  - `res` digunakan untuk menyimpan hasil operasi numerik float.
- Output akhirnya ditampilkan sebagai:

```
C:\Users\Hp\OneDrive\Dokumen\Tugas1>java Oprator
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output
```