

Tugas I

Makassar, 21 Februari 2025

**PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Serly Amalyanti  
Stambuk : 13020230203  
Dosen : Mardiyah Hasnawi, S.kom., M.T.,MTA.

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

```

Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\serly>cd/

C:\>D:

D:\>cd SEMESTER 4

D:\SEMESTER 4>cd Pemrograman berorientasi objek

D:\SEMESTER 4\Pemrograman berorientasi objek>cd Oop1

D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>dir
Volume in drive D is Data
Volume Serial Number is 8E20-59E7

Directory of D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1

02/26/2025  08:27 PM  <DIR>          .
02/26/2025  06:35 PM  <DIR>          ..
02/26/2025  08:10 PM                273 oop_kode1.java
02/26/2025  08:16 PM                395 oop_kode10.java
02/26/2025  08:16 PM                321 oop_kode11.java
02/26/2025  08:16 PM                701 oop_kode12.java
02/26/2025  08:16 PM                845 oop_kode13.java
02/26/2025  08:17 PM                536 oop_kode14.java
02/26/2025  08:17 PM                552 oop_kode15.java
02/26/2025  08:17 PM           1,126 oop_kode16.java
02/26/2025  08:14 PM                278 oop_kode2.java
02/26/2025  08:14 PM           1,082 oop_kode3.java
02/26/2025  08:14 PM                527 oop_kode4.java
02/26/2025  08:14 PM           1,023 oop_kode5.java
02/26/2025  08:15 PM                738 oop_kode6.java
02/26/2025  08:15 PM                928 oop_kode7.java
02/26/2025  08:15 PM                489 oop_kode8.java
02/26/2025  08:15 PM                856 oop_kode9.java
                16 File(s)              10,670 bytes
                2 Dir(s)  261,030,486,016 bytes free

```

## 1. Program 1

```

D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>javac oop_kode1.java

D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>java oop_kode1
f : 20.0
f11: 10.0

```

Penjelasan : *oop\_kode1.java* adalah sebuah program sederhana dalam bahasa Java yang menunjukkan cara penggunaan data float dan double. Program ini memiliki dua variabel, yaitu Variabel *f* bertipe float dengan nilai *20.0f*, sedangkan *f11* bertipe double dan diberi nilai *10.0f*. Dalam metode main, program mencetak nilai kedua variabel menggunakan `System.out.println`, dengan karakter `\n` untuk membuat baris baru. Hasil outputnya seperti pada gambar yang di highlight. Program ini menunjukkan perbedaan antara float dan double, di mana double memiliki presisi lebih tinggi dan dapat menyimpan nilai dari float tanpa masalah.

## 2. Program 2

```

D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>javac oop_kode2.java

D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>java oop_kode2
hello
Ini nilai i :5

```

Penjelasan : *oop\_kode2.java* menunjukkan penggunaan tipe data integer (int) dan cara mencetak teks di Java. Variabel *i* dideklarasikan sebagai int dan diberi nilai 5. Program mencetak *"hello"* dengan \n untuk pindah baris, lalu menampilkan *Ini nilai i :5* menggunakan System.out.println. Saat dijalankan, outputnya: seperti gambar yang di highlight.

### 3. Program 3

```
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>javac oop_kode3.java
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>java oop_kode3
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 : *op\_kode3.java* menunjukkan penggunaan berbagai tipe data primitif di Java, seperti *char* yang memiliki karakter ( 'A', 'Z', dan 'z'), Selain itu, program juga menampilkan bilangan bulat dalam beberapa tipe, yaitu *short* dengan nilai 1, *int* dengan nilai 1, dan *long* dengan nilai 10000. Selanjutnya, program mencetak bilangan *real* *x* = 50.20000076293945 dan *y* = 50.2, yang menunjukkan perbedaan presisi pada tipe double. Saat dijalankan, outputnya: seperti gambar yang di-highlight.

### 4. Program 4

```
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>javac oop_kode4.java
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>java oop_kode4
Contoh membaca dan menulis, ketik nilai integer:
8
Nilai yang dibaca : 8
```

Penjelasan : *oop\_kode4.java* menunjukkan cara membaca dan menampilkan nilai integer di Java menggunakan input pengguna. Program meminta pengguna untuk mengetikkan sebuah bilangan bulat dengan pesan *"Contoh membaca dan menulis, ketik nilai integer:"*. Setelah pengguna memasukkan angka misalnya 5, program membaca input tersebut dan mencetak *"Nilai yang dibaca: 5"*. Saat dijalankan, outputnya: seperti gambar yang di-highlight.

## 5. Program 5

```
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>javac oop_kode5.java
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>java oop_kode5
hello
baca 1 karakter : serly
baca 1 bilangan : 4
s
4
bye
```

Penjelasan : *oop\_kode5.java* menunjukkan penggunaan input pengguna di Java. Program mencetak "*hello*", lalu membaca satu karakter dan satu bilangan yang dimasukkan oleh pengguna. Input karakter yang diberikan adalah "serly", tetapi hanya karakter pertama yang digunakan. Selanjutnya, program membaca dan mencetak bilangan 8, kemudian menampilkan nilai tersebut. Terakhir, program mencetak "bye" sebagai tanda bahwa eksekusi telah selesai. Saat dijalankan, outputnya: seperti gambar yang di-highlight.

## 6. Program 6

```
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>javac oop_kode6.java
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>java oop_kode6
5.0
6.0
2
0.20000000298023224
53
53.0
53.0
3
3.14
```

Penjelasan : *oop\_kode6.java* menunjukkan cara casting (konversi tipe data) di Java menggunakan tipe data primitif seperti int, float, double, dan char. Program mengubah nilai dari satu tipe ke tipe lain, termasuk konversi char ke angka ASCII. Outputnya menampilkan angka 5.0, 6.0, 2, 0.2, 53, 53.0, 3, dan 3.14, yang mencerminkan bagaimana Java menangani angka dalam berbagai format. Saat dijalankan, hasilnya seperti gambar yang di-highlight

## 7. Program 7

```
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>javac oop_kode7.java
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>java oop_kode7
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 : *oop\_kode7.java* menunjukkan konversi tipe data menggunakan kelas wrapper di Java. Program ini mengubah String ke Integer, Double, dan Float, lalu mencetak hasilnya (*a : 67, k : 45.0, d : 100.0*). Kemudian, program mengonversi Integer, Char, dan Float ke String, menghasilkan (*n : 9, m : 5, l : 3.2*). Terakhir, Integer dikonversi ke Double, menghasilkan (*k : 67.0, c : 9.0, l : 3.2*). Program ini membantu memahami cara mengonversi data di Java dengan metode bawaan seperti *parseInt()*, *parseDouble()*, dan *valueOf()*.

## 8. Program 8

```
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>javac oop_kode8.java
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>java oop_kode8
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
```

Penjelasan : *oop\_kode8.java* menunjukkan penggunaan operator kondisional (ternary operator) *? :* di Java. Variabel *x* dan *y* diberi nilai 1 dan 2. Program kemudian mengevaluasi ekspresi *(x < y) ? x : y*, yang berarti jika *x* lebih kecil dari *y*, maka hasilnya *x*; jika tidak, hasilnya *y*. Saat dijalankan, hasilnya seperti pada gambar yang di highlight.

## 9. Program 9

```
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>javac oop_kode9.java
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>java oop_kode9
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 : *oop\_kode9.java* menunjukkan pembagian bilangan menggunakan tipe data integer dan float di Java. Program ini membandingkan hasil pembagian dengan tipe data yang berbeda. Jika kedua bilangan menggunakan tipe integer, hasilnya hanya menampilkan bilangan bulat tanpa desimal. Namun, jika salah satu atau kedua bilangan menggunakan tipe float, hasilnya akan menampilkan angka desimal. Konversi eksplisit dengan *float(x)* atau *float(y)* digunakan untuk memastikan hasil pembagian dalam format desimal. Output menunjukkan bagaimana tipe data memengaruhi hasil pembagian, di mana integer hanya menampilkan bilangan bulat, sedangkan float menampilkan angka dengan desimal.

## 10. Program 10

```
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>javac oop_kode10.java
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>java oop_kode10
Hello
Hello World
Welcome
```

Penjelasan : *oop\_kode10* ini menunjukkan bagaimana penggunaan `System.out.print()` dan `System.out.println()` dalam Java untuk mencetak teks ke layar. Output menunjukkan bagaimana perbedaan antara `print()`, yang tidak menambahkan baris baru, dan `println()`, yang otomatis berpindah ke baris berikutnya setelah mencetak teks.

## 11. Program 11

```
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>javac oop_kode11.java
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>java oop_kode11
Nilai i : 5
Nilai j : 3
```

Penjelasan : *oop\_kode11* menunjukkan efek dari operator increment (`++`) dalam Java. Variabel *i* diinisialisasi dengan nilai 3, kemudian `j = i++` menyebabkan `j` menyimpan nilai *i* sebelum *i* bertambah. Karena `i++` adalah post-increment, nilai `j` tetap 3, sedangkan *i* meningkat menjadi 4. Selanjutnya, pada perintah `System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j);`, operator pre-increment (`++i`) digunakan, sehingga *i* bertambah menjadi 5 sebelum dicetak. Sementara itu, nilai `j` tetap 3. Oleh karena itu, output yang dihasilkan adalah Nilai *i* : 5 dan Nilai *j* : 3.

## 12. Program 12

```
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>javac oop_kode12.java
D:\SEMESTER 4\Pemrograman berorientasi objek\0op1>java oop_kode12
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

Penjelasan : *oop\_kode12* menunjukkan penggunaan operator bitwise dalam Java. Nilai awal diberikan untuk variabel `n = 10`, `x = 1`, dan `y = 2`. Kemudian, dilakukan beberapa operasi bitwise: Pertama, operasi AND bitwise (`&`) antara `n` dan 8 menghasilkan 8 karena hanya bit yang sama-sama bernilai 1 yang tetap 1. Selanjutnya, operasi AND dengan NOT (`& ~`) antara `x` dan komplement 8 menghasilkan 1. Operator left shift (`<<`) digunakan pada `y`, yang menggeser bit ke kiri sebanyak 2 posisi, menghasilkan 8. Terakhir, operator right shift (`>>`) menggeser `y` ke kanan sebanyak 3 posisi, menghasilkan 0. Program

ini menunjukkan bagaimana operasi bitwise dapat digunakan untuk manipulasi data biner dalam pemrograman.

### 13. Program 13

```
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>javac oop_kode13.java
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>java oop_kode13
i = 3
j =
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
```

Penjelasan : *oop\_kode13* menunjukkan penggunaan operator bitwise di Java. Nilai  $i = 3$ , sedangkan  $j$  tampaknya tidak terinisialisasi dengan jelas. Operasi  $i \& j$  menghasilkan 0,  $i | j$  dan  $i \wedge j$  menghasilkan 7, menunjukkan bahwa  $j$  kemungkinan bernilai 7. Selain itu, terdapat hasil 81.0, kemungkinan dari operasi matematika. Operator NOT bitwise ( $\sim i$ ) mengubah 3 menjadi -4 dengan membalik bit dan mengubah tanda.

### 14. Program 14

```
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>javac oop_kode14.java
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>java oop_kode14
true
false
true
true
true
```

Penjelasan : *oop\_kode14* kemungkinan besar menggunakan operasi logika seperti AND (&&), OR (||), NOT (!), atau perbandingan (==, !=, >, <, dll.). Hasil output menampilkan nilai boolean (true dan false), yang menunjukkan bahwa program melakukan evaluasi kondisi logika dan menampilkannya di layar.

### 15. Program 15

```
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>javac oop_kode15.java
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>java oop_kode15
Nilai e = 10
Nilai k = 0
Nilai k = 4
```



Penjelasan : *oop\_kode15* menampilkan hasil operasi pada variabel *e* dan *k*. Dari outputnya, *e* memiliki nilai awal 10, sedangkan *k* mengalami perubahan dari 0 ke 4. Ini menunjukkan bahwa program mungkin menggunakan operasi bitwise, aritmatika, atau logika untuk memodifikasi nilai *k* sebelum menampilkannya.

## 16. Program 16

```
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>javac oop_kode16.java  
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>java oop_kode16  
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output
```

Penjelasan : *oop\_kode16* menampilkan pesan yang meminta pengguna untuk membaca teks dan menambahkan perintah untuk menampilkan output. Ini menunjukkan bahwa dalam kode sumbernya, mungkin ada bagian yang belum lengkap atau perlu diperbaiki agar program bisa menampilkan output sesuai yang diinginkan.

```
D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1>dir  
Volume in drive D is Data  
Volume Serial Number is 8E20-59E7  
  
Directory of D:\SEMESTER 4\Pemrograman berorientasi objek\Oop1  
  
02/26/2025 09:02 PM <DIR> .  
02/27/2025 08:28 AM <DIR> ..  
02/27/2025 02:31 PM      882 oop_kode1.class  
02/26/2025 08:10 PM      273 oop_kode1.java  
02/27/2025 02:36 PM      510 oop_kode10.class  
02/26/2025 08:16 PM      395 oop_kode10.java  
02/27/2025 02:36 PM      884 oop_kode11.class  
02/26/2025 08:16 PM      321 oop_kode11.java  
02/27/2025 02:37 PM    1,123 oop_kode12.class  
02/26/2025 08:16 PM      701 oop_kode12.java  
02/27/2025 02:37 PM    1,177 oop_kode13.class  
02/26/2025 08:16 PM      845 oop_kode13.java  
02/27/2025 02:38 PM      434 oop_kode14.class  
02/26/2025 08:17 PM      536 oop_kode14.java  
02/27/2025 02:38 PM    1,082 oop_kode15.class  
02/26/2025 08:17 PM      552 oop_kode15.java  
02/27/2025 02:38 PM      943 oop_kode16.class  
02/26/2025 08:17 PM    1,126 oop_kode16.java  
02/27/2025 02:33 PM      897 oop_kode2.class  
02/26/2025 08:14 PM      278 oop_kode2.java  
02/27/2025 02:33 PM    1,378 oop_kode3.class  
02/26/2025 08:14 PM    1,082 oop_kode3.java  
02/27/2025 02:33 PM    1,074 oop_kode4.class  
02/26/2025 08:50 PM      528 oop_kode4.java  
02/27/2025 02:34 PM    1,416 oop_kode5.class  
02/26/2025 08:56 PM    1,025 oop_kode5.java  
02/27/2025 02:34 PM      574 oop_kode6.class  
02/26/2025 08:15 PM      738 oop_kode6.java  
02/27/2025 02:35 PM    1,750 oop_kode7.class  
02/26/2025 08:15 PM      928 oop_kode7.java  
02/27/2025 02:35 PM    1,054 oop_kode8.class  
02/26/2025 08:15 PM      489 oop_kode8.java  
02/27/2025 02:36 PM    1,258 oop_kode9.class  
02/26/2025 08:15 PM      856 oop_kode9.java  
          32 File(s)      27,109 bytes  
          2 Dir(s)  261,027,803,136 bytes free
```