

**LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Putri Habiba Syari
Stambuk : 13020230333
Kelas : B1
Dosen : Mardiyah Hasnawi, S.Kom., M.T.

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

1. Asgddl.java

Program ini mendeklarasikan dua variabel, yaitu `f` bertipe `float` yang diinisialisasi dengan nilai `20.0f`, dan `f11` bertipe `double` tanpa nilai awal. Kemudian, `f11` diberikan nilai `10.0f`, yang merupakan bilangan bertipe `float`, tetapi secara otomatis dikonversi menjadi `double` karena `double` memiliki presisi lebih tinggi daripada `float`. Setelah itu, program mencetak nilai kedua variabel menggunakan `System.out.println()`, di mana `\n` digunakan untuk membuat baris baru agar hasil output terlihat lebih rapi. Program ini menunjukkan bagaimana deklarasi, inisialisasi, dan konversi tipe data `float` ke `double` dapat dilakukan dalam Java.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Asgddl.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Asgddl
f : 20.0
f11: 10.0
```

2. Asign.java

Program ini mendeklarasikan variabel `i` bertipe `int` tanpa langsung memberikan nilai awal. Kemudian, program mencetak "hello" diikuti oleh karakter `\n`, yang berfungsi untuk membuat baris baru. Setelah itu, variabel `i` diberikan nilai `5`, lalu program mencetak "Ini nilai i : " diikuti dengan nilai dari `i`. Program ini menunjukkan penggunaan dasar variabel dalam Java, termasuk deklarasi, inisialisasi, serta bagaimana mencetak teks dan nilai variabel menggunakan `System.out.println()` dan `System.out.print()`.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Asign.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Asign
hello
Ini nilai i : 5
```

3. ASIGNi.java

Program Java ini mendeklarasikan dan menginisialisasi berbagai tipe data primitif, kemudian mencetak nilainya ke layar. Variabel `ks`, `ki`, dan `kl` masing-masing bertipe `short`, `int`, dan `long`, yang merepresentasikan bilangan bulat dengan rentang nilai berbeda, ditampilkan dengan format yang menggunakan karakter tab (`\t`) agar lebih rapi. Variabel `c` dan `c1` bertipe `char` diberi nilai `65` (karakter 'A' dalam ASCII) dan 'Z', lalu dicetak dalam bentuk karakter maupun nilai numeriknya (`int`). Untuk tipe data real, `x` bertipe `double` dan `y` bertipe `float`, dengan nilai `50.2`, di mana `float` harus menggunakan sufiks `f`. Program ini menunjukkan bagaimana Java menangani berbagai tipe data numerik dan karakter, serta cara mencetaknya dengan format yang lebih terstruktur.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac ASIGNi.java

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java ASIGNi
Karakter = A
Karakter = Z
Karakter = 65
Karakter = 90
Bilangan integer (short) = 1
          (int) = 1
          (long) = 10000
Bilangan Real x = 50.2
Bilangan Real y = 50.2
```

4. **BacaData.java**

Program ini menggunakan kelas Scanner untuk membaca input dari pengguna dan mencetaknya ke layar. Pertama, variabel a bertipe int dideklarasikan untuk menyimpan nilai yang akan dimasukkan oleh pengguna. Kemudian, objek Scanner bernama masukan dibuat untuk membaca input dari keyboard. Program menampilkan pesan kepada pengguna untuk memasukkan sebuah bilangan integer, lalu membaca input tersebut menggunakan masukan.nextInt(), yang mengambil angka yang dimasukkan dan menyimpannya ke dalam variabel a. Akhirnya, program mencetak kembali nilai yang telah dibaca dengan format "Nilai yang dibaca: " + a. Program ini mendemonstrasikan cara menerima dan menampilkan data integer menggunakan Scanner dalam Java.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac BacaData.java

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java BacaData
Contoh membaca dan menulis, ketik nilai integer:
13
Nilai yang dibaca: 13
```

5. **Bacakar.java**

Program ini menggunakan BufferedReader untuk membaca input dari pengguna secara efisien. BufferedReader bekerja bersama InputStreamReader untuk mengubah input dari keyboard (System.in) menjadi karakter atau string, sedangkan IOException digunakan untuk menangani kemungkinan kesalahan saat membaca input. Program pertama mencetak "hello", lalu meminta pengguna memasukkan satu karakter dengan dataIn.readLine().charAt(0), serta sebuah bilangan yang dikonversi ke int menggunakan Integer.parseInt(dataIn.readLine()). Setelah itu, program mencetak hasil input dan menutup dengan pesan "bye". Metode ini lebih cepat dibandingkan Scanner, terutama untuk membaca teks dalam jumlah besar.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Bacakar.java

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Bacakar
hello
baca 1 karakter: P
baca 1 bilangan: 7
P
7
bye
```

6. **Casting1.java**

Program ini mendemonstrasikan **casting** (konversi tipe data) dalam Java dengan mengubah berbagai tipe data secara eksplisit. Variabel a dan b bertipe int, d dan e bertipe float, g bertipe char (berisi karakter '5'), serta k bertipe double. Dalam proses casting, int dapat diubah menjadi float atau double, sedangkan float dapat dikonversi ke int (menghilangkan desimal) atau double (menjaga nilai). char dikonversi ke int, float, dan double berdasarkan nilai ASCII ('5' memiliki nilai ASCII 53). Sementara itu, double dapat diubah menjadi int (membuang desimal) atau float (tetap mempertahankan desimal). Program ini menunjukkan bagaimana Java menangani konversi tipe data dengan aturan tertentu, terutama dalam kehilangan atau mempertahankan presisi nilai.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Casting1.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Casting1
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14
```

7. Casting2.java

Program ini mendemonstrasikan **konversi tipe data (casting dan parsing)** dalam Java, baik dari **string ke numerik** maupun **numerik ke string**. Variabel a, b, d, e, g, dan k digunakan untuk menyimpan nilai numerik dan karakter, sementara n, m, dan l adalah string yang merepresentasikan angka. Program pertama mengonversi string ke tipe numerik menggunakan Integer.parseInt(), Double.parseDouble(), dan Float.parseFloat(), yang masing-masing berfungsi untuk mengubah string menjadi int, double, dan float. Selanjutnya, nilai numerik dikonversi kembali ke string menggunakan String.valueOf(). Terakhir, Double.valueOf(a).intValue() mengubah a menjadi Double sebelum dikembalikan ke int, dan Integer.valueOf(b).doubleValue() mengubah b menjadi Integer lalu dikonversi ke double. Program ini menunjukkan bagaimana Java menangani konversi antara string dan tipe numerik dengan metode bawaan yang efisien, memungkinkan pemrosesan data dari berbagai sumber seperti input pengguna atau file teks.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Casting2.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\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
```

8. Ekspresi.java

Program ini mendemonstrasikan **operator ternary (? :)** dalam Java untuk menentukan nilai berdasarkan kondisi tertentu. Variabel x dan y masing-masing diberi nilai 1 dan 2, lalu dicetak ke layar. Pernyataan ((x < y) ? x : y) adalah ekspresi ternary yang berarti: *jika x lebih kecil dari*

y, maka hasilnya adalah x, jika tidak, hasilnya adalah y. Karena $x = 1$ dan $y = 2$, kondisi $x < y$ bernilai **true**, sehingga ekspresi menghasilkan x (1). Program ini menunjukkan cara sederhana menggunakan operator ternary untuk memilih nilai berdasarkan kondisi tertentu.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Ekspresi.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
```

9. Ekspresi1.java

Program ini mendemonstrasikan **perbedaan pembagian antara tipe data integer (int) dan float (float)** dalam Java. Variabel x dan y dideklarasikan sebagai int, sementara fx dan fy sebagai float. Program pertama melakukan pembagian x/y dengan kedua variabel bertipe int, sehingga hasilnya juga bertipe int (bagian desimal dibuang). Selanjutnya, fx dan fy diisi dengan nilai x dan y, lalu dilakukan pembagian dengan tipe float, yang menghasilkan nilai desimal. Program juga menunjukkan cara melakukan **casting eksplisit** dengan (float)x / (float)y, memastikan hasilnya berupa nilai float meskipun x dan y bertipe int. Terakhir, program mengubah nilai x menjadi 10 dan y menjadi 3, lalu mengulang operasi pembagian untuk memperlihatkan perbedaan hasil antara pembagian integer dan float.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Ekspresi1.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Ekspresi1
x/y (format integer) = 0
x/y (format float) = 0
fx/fy (format integer) = 0.5
fx/fy (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. Hello.java

Program ini mendefinisikan kelas Hello, yang menunjukkan perbedaan antara **print()** dan **println()** dalam mencetak output ke layar. **System.out.print("Hello");** mencetak teks "Hello" tanpa pindah baris, kemudian **\n** dalam **System.out.print("\nHello ");** membuat baris baru sebelum mencetak "Hello ". **System.out.println("World");** mencetak "World" dan langsung pindah ke baris berikutnya, sedangkan **System.out.println("Welcome");** mencetak "Welcome" di baris baru. Program ini menunjukkan bahwa **print() mencetak teks tanpa berpindah baris, sementara println() mencetak teks lalu pindah ke baris berikutnya.**

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Hello.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Hello
Hello
Hello World
Welcome
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>
```

11. Incr.java

Program ini mendemonstrasikan penggunaan operator increment (++) pada variabel integer. Variabel i diinisialisasi dengan nilai 3, lalu j diberikan nilai i++, yang merupakan **post-increment**, sehingga j menyimpan nilai 3 sebelum i bertambah menjadi 4. Nilai j tetap 3 karena j = i++ menyimpan nilai awal i sebelum peningkatan. Pada pernyataan System.out.println(), ++i digunakan, yang merupakan **pre-increment**, sehingga i bertambah menjadi 5 sebelum dicetak. Program ini menunjukkan bahwa **post-increment (i++) mengembalikan nilai sebelum peningkatan**, sedangkan **pre-increment (++i) mengembalikan nilai setelah peningkatan**.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Incr.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Incr
Nilai i : 5
Nilai j : 3
```

12. Oper1.java

Program ini mendemonstrasikan penggunaan operator bitwise (&, ~, <<, >>) dalam Java. Variabel n, x, dan y diinisialisasi dengan nilai 10, 1, dan 2. Operator & (AND bitwise) pada n & 8 melakukan operasi AND antara 10 (1010 dalam biner) dan 8 (1000), menghasilkan 8. Operator ~ (NOT bitwise) pada ~8 mengubah semua bit (1000 menjadi 0111 dalam representasi biner 2's complement). Operator << (left shift) pada y << 2 menggeser bit y (2 atau 0010 dalam biner) ke kiri dua kali, menghasilkan 8 (1000). Operator >> (right shift) pada y >> 3 menggeser bit y ke kanan tiga kali, yang dalam hal ini menghasilkan 0. Program ini memperlihatkan bagaimana manipulasi bit dapat mengubah nilai numerik menggunakan operasi bitwise.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Oper1.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Oper1
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

13. Oper2.java

Program ini mendemonstrasikan penggunaan operator bitwise dan operasi matematika pada variabel bertipe char yang diinisialisasi dengan nilai numerik 3 dan 4. Nilai i dan j ditampilkan sebagai integer untuk memastikan bahwa mereka diperlakukan sebagai angka. Operator & (AND), | (OR), dan ^ (XOR) menghasilkan 0, 7, dan 7. Fungsi Math.pow(i, j) menghitung 3^4, menghasilkan 81.0. Operator ~i (NOT bitwise) membalik semua bit dari 3, yang dalam representasi 2's complement menjadi -4. Program ini menunjukkan bagaimana operasi bitwise bekerja pada tipe data char yang direpresentasikan sebagai bilangan bulat serta penggunaan Math.pow() untuk perpangkatan dalam Java.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Oper2.java

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Oper2
i = 3
j = 4
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
```

14. Oper3.java

Program ini mendemonstrasikan perbedaan antara operator logika (&&, ||) dan operator bitwise (&, |) dalam Java. Pernyataan true && true menggunakan AND logika, yang hanya mengevaluasi operand kedua jika diperlukan, menghasilkan true, sedangkan true & false menggunakan AND bitwise, yang selalu mengevaluasi kedua operand, sehingga mencetak false. Pernyataan if (true) memastikan blok kode selalu dieksekusi dan mencetak true. Operator || (OR logika) pada true || true hanya mengevaluasi operand pertama karena sudah true, sementara | (OR bitwise) pada true | false mengevaluasi kedua operand dan tetap menghasilkan true. Program ini menunjukkan bahwa operator logika (&&, ||) dapat melakukan short-circuit, sedangkan operator bitwise (&, |) selalu mengevaluasi kedua operand.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Oper3.java

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Oper3
true
false
true
true
true
```

15. Oper4.java

Program ini mendemonstrasikan penggunaan operator ternary (?:) untuk menentukan nilai terbesar antara dua variabel dalam Java. Variabel i dan j diinisialisasi dengan 0, sedangkan c dan d dengan nilai karakter 8 dan 10. Ekspresi (((int)c > (int)d) ? c : d) membandingkan c dan d dalam bentuk integer, sehingga e mendapat nilai 10 karena d lebih besar. Variabel k pertama kali diisi dengan nilai maksimum antara i dan j, yaitu 0. Setelah i dan j diubah menjadi 2 dan 3, ekspresi ((i++ > j++) ? i : j) mengevaluasi perbandingan sebelum peningkatan nilai, sehingga k menjadi 4 karena j++ yang awalnya 3 dikembalikan sebagai hasil. Program ini menunjukkan bagaimana operator ternary bekerja dalam pengambilan keputusan berbasis perbandingan di Java.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Oper4.java

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4
```

16. Oprator.java

Program ini mendemonstrasikan operasi logika, aritmatika, dan relasional dalam Java dengan menampilkan hasilnya secara sistematis. Variabel boolean Bool1 (true) dan Bool2 (false)

digunakan dalam operasi logika AND (&&), OR (||), NOT (!), dan XOR (^). Variabel integer i = 5 dan j = 2 serta float x = 5 dan y = 5 digunakan untuk operasi aritmatika seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus. Selain itu, program mengevaluasi operasi relasional (==, !=, <, >, <=, >=) pada variabel integer dan float. Dengan struktur yang jelas, program ini menunjukkan bagaimana berbagai operator bekerja dalam Java dan bagaimana hasilnya ditampilkan secara sistematis untuk memudahkan pemahaman.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>javac Oprator.java

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 1>java Oprator
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output

Bool1 = true, Bool2 = false

Bool1 && Bool2 = false
Bool1 || Bool2 = true
!Bool1 = false
Bool1 ^ Bool2 = true

i = 5, j = 2

i + j = 7
i - j = 3
i / j = 2
i * j = 10
i % j = 1

x = 5.0, y = 5.0

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
```