

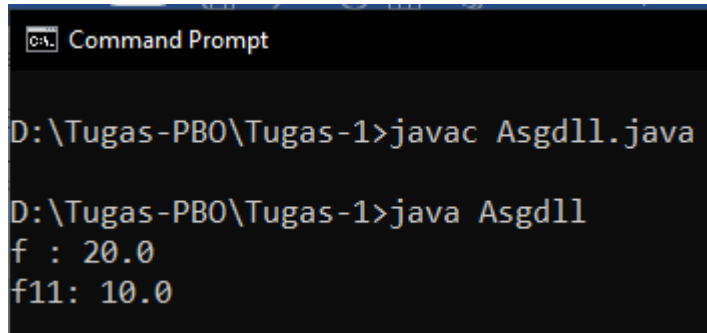
**PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Sitti Nurhalimah
Stambuk : 13020230297
Dosen : Mardiyah Hasnawi, S.Kom., M.T., MTA.

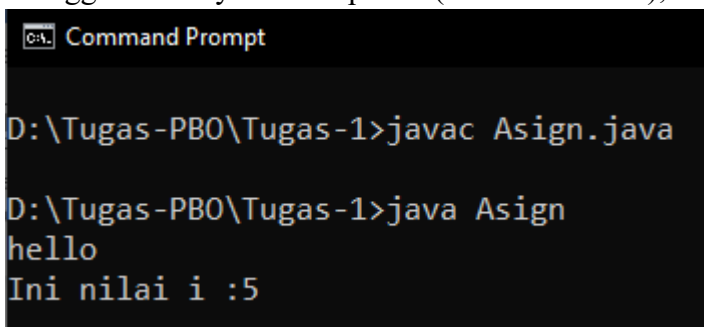
**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2024**

1. Program ini cukup sederhana, yaitu mendeklarasikan dua variabel angka dan menampilkan nilainya. Variabel `f` bertipe `float` langsung diinisialisasi dengan nilai `20.0f`, sedangkan variabel `f11` bertipe `double` awalnya belum memiliki nilai, tetapi kemudian diisi dengan `10.0f`. Saat program dijalankan, ia akan mencetak nilai dari kedua variabel tersebut dengan format:



```
Command Prompt
D:\Tugas-PBO\Tugas-1>javac Asgd11.java
D:\Tugas-PBO\Tugas-1>java Asgd11
f : 20.0
f11: 10.0
```

2. Program ini menampilkan nilai sebuah variabel bertipe `int`. Di dalam metode `main`, terdapat variabel `i` yang dideklarasikan tetapi belum langsung diberi nilai. Kemudian, program mencetak "hello" ke layar dengan menggunakan `System.out.print("hello\n");`, di mana `\n` berfungsi untuk membuat baris baru. Setelah itu, variabel `i` diberi nilai `5`, dan program mencetak "Ini nilai i :5" menggunakan `System.out.println("Ini nilai i : " + i);`.



```
Command Prompt
D:\Tugas-PBO\Tugas-1>javac Assign.java
D:\Tugas-PBO\Tugas-1>java Assign
hello
Ini nilai i :5
```

3. Program ini mendeklarasikan beberapa variabel dengan berbagai tipe data di Java, seperti `short`, `int`, `long`, `char`, `double`, dan `float`, kemudian mencetak nilainya ke layar. Variabel `ks`, `ki`, dan `kl` masing-masing bertipe `short`, `int`, dan `long`, dengan nilai `1`, `1`, dan `10000`. Untuk variabel karakter, `c` diisi dengan `65`, yang dalam tabel ASCII merepresentasikan huruf 'A', sedangkan `c1` langsung diisi dengan karakter 'Z'. Selanjutnya, ada variabel `x` bertipe `double` dan `y` bertipe `float`, keduanya memiliki nilai `50.2f`, tetapi karena presisi `double` lebih tinggi dari `float`, hasil outputnya bisa sedikit berbeda. Program kemudian mencetak semua nilai variabel tersebut ke layar, menghasilkan tampilan seperti berikut:

```
Command Prompt
D:\Tugas-PBO\Tugas-1>javac ASIGNi.java

D:\Tugas-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
```

4. Program ini menunjukkan cara membaca input bilangan bulat (integer) dari pengguna menggunakan kelas Scanner di Java. Di dalam metode main, terdapat deklarasi variabel a yang akan menyimpan nilai integer yang dimasukkan oleh pengguna. Selain itu, ada objek Scanner bernama masukan, yang digunakan untuk membaca input dari keyboard.

```
Command Prompt - java BacaData
D:\Tugas-PBO\Tugas-1>javac BacaData.java

D:\Tugas-PBO\Tugas-1>java BacaData
Contoh membaca dan menulis, ketik nilai integer:
```

Kemudian, program menunggu pengguna mengetikkan sebuah angka dan menekan Enter. Nilai yang diketik akan disimpan dalam variabel a menggunakan `masukan.nextInt()`, lalu program mencetak hasilnya dengan format:

```
D:\Tugas-PBO\Tugas-1>java BacaData
Contoh membaca dan menulis, ketik nilai integer:
5
Nilai yang dibaca : 5
```

5. Program ini menggunakan `BufferedReader` untuk membaca input dari pengguna, baik dalam bentuk karakter (`char`) maupun bilangan (`int`). Di dalam metode main, objek `BufferedReader` dibuat dengan dua cara: pertama menggunakan `InputStreamReader`, dan yang kedua langsung dalam satu baris kode. Saat dijalankan, program akan mencetak:

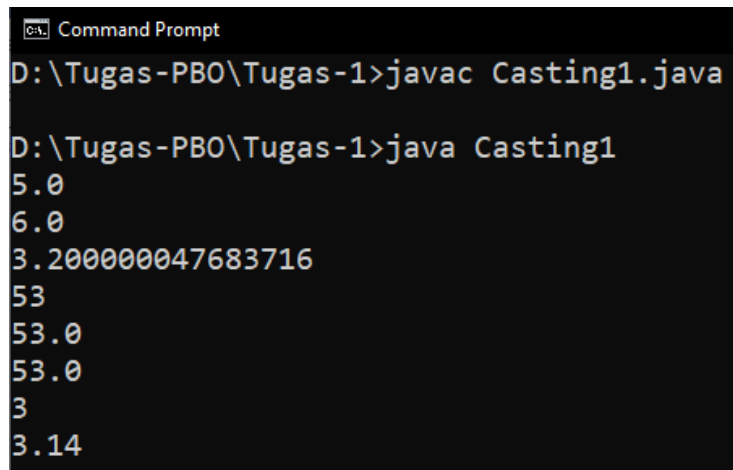
```
Command Prompt - java Bacakar
D:\Tugas-PBO\Tugas-1>javac Bacakar.java

D:\Tugas-PBO\Tugas-1>java Bacakar
hello
baca 1 karakter : _
```

Lalu, program menunggu pengguna memasukkan satu karakter dan menekan Enter. Karakter yang diketik akan disimpan. Setelah itu, program mencetak:

```
D:\Tugas-PBO\Tugas-1>java Bacakar
hello
baca 1 karakter : H
baca 1 bilangan : 5
H
5
bye
```

6. Program ini mendemonstrasikan penggunaan casting eksplisit di Java, yaitu proses mengubah satu tipe data ke tipe lainnya secara langsung. Dalam metode main, terdapat beberapa variabel dengan tipe int, float, char, dan double, yang kemudian dikonversi ke tipe lain menggunakan casting. Variabel a bertipe int dikonversi ke float, menghasilkan output 5.0, sedangkan b dikonversi ke double, menghasilkan 6.0. Variabel e bertipe float diubah menjadi double, tetap bernilai 3.2, tetapi dengan presisi lebih tinggi. Hal menarik terjadi pada variabel g bertipe char dengan nilai '5', yang dalam kode ASCII memiliki nilai numerik 53. Saat dikonversi ke int, hasilnya adalah 53, sedangkan ketika diubah ke float atau double, hasilnya menjadi 53.0. Sementara itu, variabel k bertipe double dengan nilai 3.14 dikonversi ke int, sehingga bagian desimalnya hilang, menghasilkan 3, sedangkan saat dikonversi ke float, nilainya tetap 3.14. Secara keseluruhan, program ini menunjukkan bagaimana Java menangani perubahan tipe data dengan casting eksplisit menggunakan tanda kurung (<tipe_data>).



```

C:\> Command Prompt
D:\Tugas-PBO\Tugas-1>javac Casting1.java

D:\Tugas-PBO\Tugas-1>java Casting1
5.0
6.0
3.200000047683716
53
53.0
53.0
3
3.14

```

7. Program ini menunjukkan cara melakukan konversi tipe data menggunakan kelas wrapper di Java, seperti Integer, Double, Float, dan String. Konversi ini berguna saat perlu mengubah nilai antar berbagai tipe data, terutama antara String dan tipe numerik.

Pada bagian pertama, program mengubah String menjadi tipe numerik. Variabel `n` yang awalnya bertipe String berisi "67" dikonversi menjadi Integer menggunakan `Integer.parseInt(n)`, sehingga `a` sekarang bernilai 67. Variabel `m` yang berisi "45" dikonversi ke Double dengan `Double.parseDouble(m)`, menjadikan `k` bernilai 45.0. Sedangkan `l` yang berisi "100" dikonversi ke Float dengan `Float.parseFloat(l)`, sehingga `d` menjadi 100.0.

Pada bagian kedua, program mengubah tipe numerik menjadi String. Variabel `b` (int) dikonversi ke String menggunakan `String.valueOf(b)`, sehingga `n` berisi "9". Variabel `g` (char dengan nilai '5') dikonversi ke String, sehingga `m` sekarang berisi "5". Begitu pula dengan `e` (float dengan nilai 3.2), yang dikonversi menjadi String, menjadikan `l` berisi "3.2".

Pada bagian terakhir, program mengubah tipe numerik satu sama lain menggunakan metode `valueOf()`. `a` dikonversi menjadi Double, lalu dipaksa menjadi int, sehingga `k` tetap 67. Kemudian, `b` (int) diubah menjadi Double menggunakan `Integer.valueOf(b).doubleValue()`, sehingga `c` bernilai 9.0.

Secara keseluruhan, program ini mendemonstrasikan berbagai cara melakukan konversi antar tipe data di Java, baik dari String ke angka maupun sebaliknya, serta antar tipe numerik menggunakan kelas wrapper.

```

Command Prompt
D:\Tugas-PBO\Tugas-1>javac Casting2.java

D:\Tugas-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

```

8. rogram ini menunjukkan bagaimana menggunakan operator ternary (?:) untuk memilih antara dua nilai berdasarkan kondisi tertentu, sehingga lebih ringkas dibandingkan dengan if-else.

```

Command Prompt
D:\Tugas-PBO\Tugas-1>java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1

```

9. program ini menunjukkan bahwa pembagian antar bilangan bulat menghasilkan bilangan bulat, tetapi jika ingin mendapatkan hasil desimal, harus menggunakan tipe float atau double, baik melalui variabel bertipe float atau dengan casting langsung menggunakan (float).

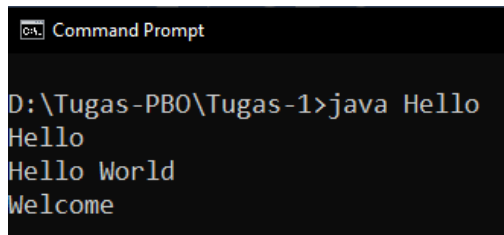
```

Command Prompt
D:\Tugas-PBO\Tugas-1>javac Ekspresi1.java

D:\Tugas-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

```

10. Program ini memperlihatkan perbedaan antara print (mencetak tanpa pindah baris) dan println (mencetak lalu pindah ke baris berikutnya), serta penggunaan \n untuk mengatur format output secara manual.



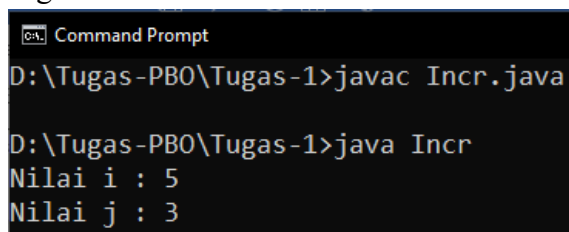
```

C:\> Command Prompt

D:\Tugas-PBO\Tugas-1>java Hello
Hello
Hello World
Welcome

```

11. Program ini menunjukkan perbedaan antara post-increment ($i++$) dan pre-increment ($++i$) dalam Java. Saat $j = i++$ dijalankan, nilai i yang awalnya 3 diberikan ke j terlebih dahulu, baru kemudian i bertambah menjadi 4. Setelah itu, dalam `System.out.println()`, digunakan $++i$, yang berarti i bertambah satu lagi menjadi 5 sebelum dicetak. Akhirnya, program mencetak "Nilai i : 5" dan "Nilai j : 3", menegaskan bahwa post-increment menggunakan nilai lama sebelum bertambah, sedangkan pre-increment menambah nilai dulu sebelum digunakan.



```

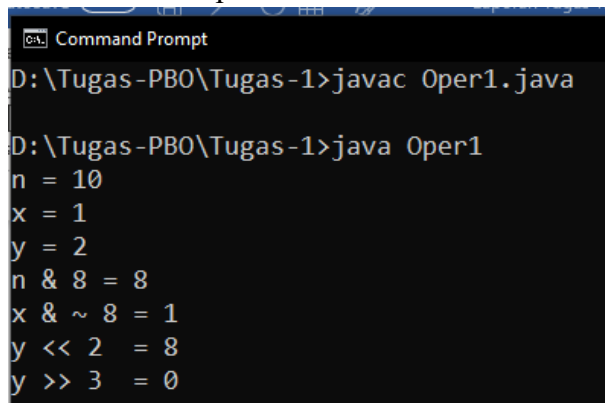
C:\> Command Prompt

D:\Tugas-PBO\Tugas-1>javac Incr.java

D:\Tugas-PBO\Tugas-1>java Incr
Nilai i : 5
Nilai j : 3

```

12. Program ini mendemonstrasikan operator bitwise dalam Java, yaitu AND ($\&$), NOT (\sim), left shift (\ll), dan right shift (\gg). Operasi $n \& 8$ ($10 \& 8$) menghasilkan 8 karena bit yang sama tetap 1. Pada $x \& \sim 8$, ~ 8 membalik bit 8 menjadi ...11110111, sehingga hasilnya tetap 1. Operasi $y \ll 2$ menggeser bit y (2 atau 10 dalam biner) dua kali ke kiri menjadi 1000 (8). Sebaliknya, $y \gg 3$ menggeser 2 ke kanan tiga kali, menghasilkan 0. Program ini menunjukkan bagaimana operator bitwise dapat digunakan untuk manipulasi angka secara efisien dalam representasi biner.



```

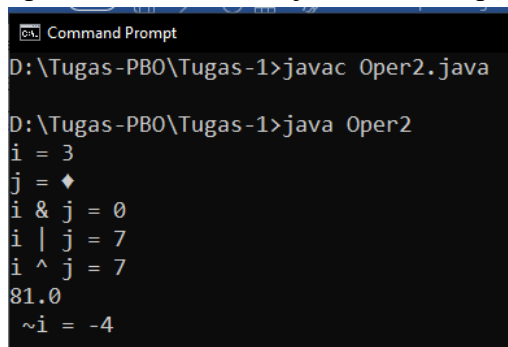
C:\> Command Prompt

D:\Tugas-PBO\Tugas-1>javac Oper1.java

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

```

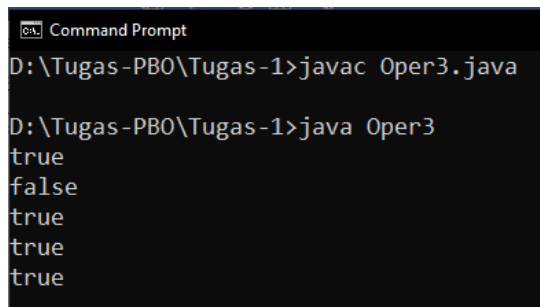
13. Program ini mendemonstrasikan penggunaan operator relasional dan bitwise dalam Java, termasuk AND (&), OR (|), XOR (^), NOT (~), dan pemangkatan (Math.pow()). Variabel i (3 atau 00000011 dalam biner) dan j (4 atau 00000100 dalam biner) digunakan untuk operasi bitwise. Hasil dari i & j adalah 0 karena tidak ada bit yang sama di posisi yang sama. i | j menghasilkan 7 (00000111), karena OR menggabungkan bit yang bernilai 1. i ^ j juga menghasilkan 7, karena XOR menghasilkan 1 hanya jika bit berbeda. Operator ^ dalam Java adalah XOR, bukan pangkat, sehingga untuk pemangkatan digunakan Math.pow(i, j), yang menghitung $3^4 = 81.0$. Operator NOT (~i) membalik semua bit 3 (00000011), menghasilkan -4 (11111100 dalam komplement dua). Program ini menunjukkan bagaimana operator bitwise bekerja dalam manipulasi angka pada level biner.



```
Command Prompt
D:\Tugas-PBO\Tugas-1>javac Oper2.java

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

14. Program ini menunjukkan perbedaan antara operator logika (&& dan ||) yang menerapkan evaluasi malas dan operator bitwise logika (& dan |) yang selalu mengevaluasi kedua operand, dengan menjalankan beberapa pernyataan kondisi yang menghasilkan nilai boolean. Saat menggunakan &&, seperti pada kondisi if (true && true), ekspresi langsung menghasilkan true tanpa perlu mengevaluasi operand kedua jika operand pertama sudah menentukan hasil, sedangkan pada penggunaan & dalam if (true & true), kedua operand selalu dievaluasi sehingga ketika ditampilkan nilai dari true & false, hasilnya false. Selain itu, pernyataan if (true) jelas selalu benar dan mencetak true, dan operator || bekerja dengan cara mengevaluasi operand kedua hanya jika diperlukan, sehingga if (true || true) menghasilkan true, sedangkan penggunaan | pada kondisi true | false menghasilkan true karena kedua operand dievaluasi. Dengan demikian, program ini memperlihatkan bagaimana perbedaan antara evaluasi malas pada operator logika dan evaluasi penuh pada operator bitwise logika mempengaruhi hasil dan efisiensi dalam penilaian kondisi.



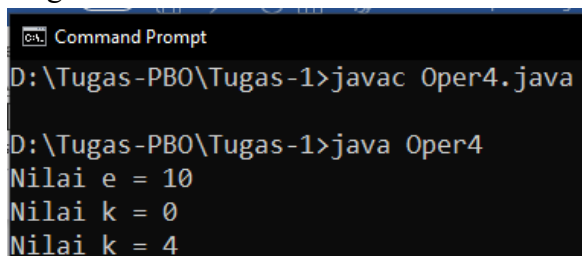
```

C:\ Command Prompt
D:\Tugas-PBO\Tugas-1>javac Oper3.java

D:\Tugas-PBO\Tugas-1>java Oper3
true
false
true
true
true

```

15. Program ini mendemonstrasikan penggunaan operator ternary ($? :$) dalam Java untuk menentukan nilai berdasarkan kondisi tertentu. Operator ternary bekerja seperti pernyataan if-else dalam satu baris kode. Pertama, variabel e ditentukan menggunakan $((int)c > (int)d) ? c : d$, yang membandingkan c (8) dan d (10), lalu memilih nilai terbesar, yaitu 10, sehingga $e = 10$. Selanjutnya, variabel k ditentukan dengan $(i > j) ? i : j$, yang membandingkan i (0) dan j (0), karena nilainya sama, k tetap 0. Setelah i dan j diperbarui menjadi 2 dan 3, pernyataan $k = ((i++ > j++) ? i : j)$; dieksekusi. Karena $i++ > j++$ ($2 > 3$) adalah false, maka k mengambil nilai j , tetapi karena $j++$ berarti j bertambah setelah dibandingkan, nilai k menjadi 4. Program ini menunjukkan bagaimana operator ternary dapat digunakan untuk memilih nilai dengan cara yang ringkas dan efisien.



```

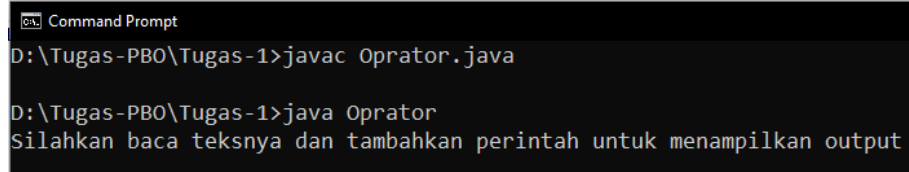
C:\ Command Prompt
D:\Tugas-PBO\Tugas-1>javac Oper4.java

D:\Tugas-PBO\Tugas-1>java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4

```

16. Program ini menunjukkan berbagai operasi dasar pada tipe data boolean, integer, dan float dalam Java. Operasi logika dilakukan pada variabel Bool1 dan Bool2, seperti AND ($\&\&$), OR ($\|\|$), NOT ($!$), dan XOR ($^$), yang menentukan nilai TF berdasarkan kondisi boolean yang diberikan. Kemudian, operasi aritmetika dilakukan dengan variabel integer i dan j , termasuk penjumlahan ($+$), pengurangan ($-$), perkalian ($*$), pembagian ($/$), dan modulus ($\%$), di mana pembagian antara dua integer akan menghasilkan hasil pembagian bulat (tanpa desimal). Hal yang sama diterapkan pada variabel float x dan y , tetapi pembagian pada tipe float akan menghasilkan nilai desimal. Selanjutnya, program ini juga mencakup operasi relasional yang membandingkan dua nilai numerik, seperti sama dengan ($==$), tidak sama dengan ($!=$), lebih kecil ($<$), lebih besar ($>$), serta operator kurang dari atau sama dengan ($<=$) dan lebih dari atau sama dengan ($>=$), yang semuanya menghasilkan nilai boolean (true atau false). Meskipun program belum

memiliki perintah `System.out.println()` untuk menampilkan hasil, konsep dasarnya memperlihatkan bagaimana Java menangani operasi logika, aritmetika, dan relasional secara efisien.



```
Command Prompt
D:\Tugas-PBO\Tugas-1>javac Oprator.java

D:\Tugas-PBO\Tugas-1>java Oprator
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output
```