

Tugas I

Makassar, 21 Februari 2024

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Ghadiyahtul Fadiyah Ahmad
Nim : 13020230085
Dosen : Mardiyah Hasnawi, S.Kom., M.T.

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

TUGAS PROGRAM PBO

Program 1

output :

```
D:\PBO\Tugas1>javac Asgd11.java
D:\PBO\Tugas1>java Asgd11.java
f : 20.0
f11: 10.0
D:\PBO\Tugas1>
```

keterangan :

Program ini mencetak nilai dari variabel *f* yang bertipe *float* dengan angka 20.0, lalu variabel *f11* yang bertipe *double* dengan angka 10.0. Awalnya, *f11* diberi nilai 10.0f, yang sebenarnya bertipe *float*, tetapi Java secara otomatis mengubahnya menjadi *double* karena tipe ini memiliki kapasitas yang lebih besar. Meskipun tipe datanya berbeda, hasil output yang ditampilkan tetap sama, yaitu *f*: 20.0 dan *f11*: 10.0.

Program 2

output :

```
D:\PBO\Tugas1>javac Asign.java
D:\PBO\Tugas1>java Asign.java
hello
Ini nilai i :5
D:\PBO\Tugas1>_
```

keterangan :

Program ini membuat variabel *i* dengan tipe *integer*. Pertama, perintah `System.out.print("hello\n");` digunakan untuk menampilkan teks "hello" lalu pindah ke baris berikutnya. Setelah itu, variabel *i* diisi dengan angka 5 menggunakan `i = 5;`. Kemudian, perintah `System.out.println("Ini nilai i : " + i);` akan menampilkan teks "Ini nilai i : " diikuti dengan angka dari variabel *i*, yaitu 5. Hasil akhirnya, program akan menghasilkan dua baris output: "hello" di baris pertama dan "Ini nilai i :5" di baris kedua.

Program 3

output :

```
D:\PBO\Tugas1>javac ASIGNi.java
D:\PBO\Tugas1>java ASIGNi.java
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
D:\PBO\Tugas1>_
```

keterangan :

Program ini mendefinisikan beberapa variabel dengan berbagai tipe data, termasuk *short*, *int*, *long*, *char*, *double*, dan *float*. Variabel *c* diberi nilai 65, yang dalam tabel ASCII mewakili karakter 'A', sedangkan *c1* langsung diinisialisasi dengan karakter 'Z'. Selanjutnya, program menampilkan nilai dari setiap variabel, baik dalam bentuk

karakter maupun angka yang sesuai. Hasil eksekusi menunjukkan bahwa c ditampilkan sebagai 'A' dan c1 sebagai 'Z', sementara variabel numerik lainnya mencetak nilai sesuai dengan tipe datanya. Program ini mengilustrasikan bagaimana berbagai tipe data digunakan serta bagaimana karakter dapat direpresentasikan dalam bentuk angka ASCII.

Program 4

output :

```
D:\PBO\Tugas1>javac BacaData.java

D:\PBO\Tugas1>java BacaData.java
Contoh membaca dan menulis, ketik nilai integer:
1
Nilai yang dibaca : 1
D:\PBO\Tugas1>_
```

keterangan :

Program ini memanfaatkan kelas Scanner untuk mengambil input dari pengguna. Pertama, sebuah variabel a bertipe int dideklarasikan, bersama dengan sebuah objek Scanner bernama masukan. Program kemudian menampilkan instruksi kepada pengguna untuk memasukkan angka. Input yang diberikan pengguna dibaca menggunakan masukan.nextInt() dan disimpan dalam variabel a, lalu ditampilkan kembali ke layar. Misalnya, jika pengguna memasukkan angka 1, output yang muncul adalah "Nilai yang dibaca: 1". Namun, jika pengguna memasukkan data selain angka, program akan mengalami error karena metode nextInt() hanya dapat memproses input berupa bilangan bulat.

Program 5

output :

```
D:\PBO\Tugas1>javac Bacakar.java

D:\PBO\Tugas1>java Bacakar.java
hello
baca 1 karakter : f
baca 1 bilangan : 1
f
1
bye
D:\PBO\Tugas1>_
```

keterangan :

Program ini menggunakan BufferedReader untuk membaca input dari pengguna. Diawali dengan mencetak "hello" sebagai sapaan awal, program kemudian meminta pengguna untuk memasukkan satu karakter dengan menampilkan pesan "baca 1 karakter :". Jika pengguna mengetikkan kata seperti "fadiyah", program hanya mengambil karakter pertama, yaitu 'f', dengan metode charAt(0). Selanjutnya, program meminta pengguna untuk memasukkan sebuah angka dengan pesan "baca 1 bilangan :". Input yang diberikan akan dibaca sebagai string, lalu dikonversi menjadi bilangan bulat menggunakan Integer.parseInt(). Sebagai contoh,

jika pengguna memasukkan angka 1, maka program akan menyimpan dan mencetak angka tersebut.

Setelah menampilkan karakter serta angka yang dimasukkan pengguna, program mengakhiri eksekusi dengan mencetak "bye". Penggunaan `BufferedReader` dalam program ini memungkinkan pengolahan input secara efisien, tetapi metode yang digunakan hanya mengambil karakter pertama dari input string yang lebih panjang.

Program 6

output :

```
D:\PBO\Tugas1>javac Casting1.java
D:\PBO\Tugas1>java Casting1.java
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14
D:\PBO\Tugas1>
```

keterangan :

Program ini mendeklarasikan variabel dengan tipe data berbeda (`int`, `float`, `char`, `double`) dan melakukan konversi tipe. **Widening conversion** (misalnya dari `int` ke `double`) terjadi otomatis tanpa kehilangan data, sementara **narrowing conversion** (misalnya dari `double` ke `int`) membutuhkan casting eksplisit dan menghilangkan bagian desimal. Untuk `char`, konversi ke `int` menghasilkan nilai ASCII, misalnya `'5'` menjadi 53. Program ini menunjukkan pentingnya memahami konversi data agar tidak kehilangan informasi saat mengubah tipe variabel.

Program 7

output :

```
D:\PBO\Tugas1>javac Casting2.java
D:\PBO\Tugas1>java Casting2.java
a : 67
k : 45.0
d : 100.0
n : 9
m : 5
l : 3.2
k : 67.0
c : 9.0
l : 3.2
D:\PBO\Tugas1>
```

keterangan :

Program ini menunjukkan konversi tipe data menggunakan **kelas wrapper** di Java. `String` dikonversi ke tipe numerik dengan `parseInt()`, `parseDouble()`, dan `parseFloat()`, misalnya `"67"` menjadi `int` 67. Sebaliknya, nilai numerik dikonversi ke `string` dengan `String.valueOf()`, seperti `int` 9 menjadi `"9"`. Program juga mengonversi `Integer` ke

Double dan sebaliknya. Penggunaan kelas wrapper ini memudahkan pemrosesan input, manipulasi data, dan konversi tipe dalam Java.

Program 8

output :

```
D:\PBO\Tugas1>javac Ekspresi.java
D:\PBO\Tugas1>java Ekspresi.java
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
D:\PBO\Tugas1>
```

keterangan :

Program ini mendemonstrasikan penggunaan **operator ternary** dalam Java sebagai alternatif dari pernyataan if-else. Operator ini digunakan untuk menentukan nilai berdasarkan suatu kondisi dengan sintaks yang lebih ringkas.

Di dalam program, terdapat dua variabel, x dan y, yang masing-masing diinisialisasi dengan nilai 1 dan 2. Program pertama-tama mencetak nilai dari kedua variabel tersebut, lalu mengevaluasi ekspresi **(x < y) ? x : y**. Ekspresi ini bekerja dengan cara mengecek kondisi $x < y$. Jika kondisi tersebut **benar (true)**, maka nilai x akan dikembalikan. Sebaliknya, jika kondisi **salah (false)**, maka nilai y yang akan dikembalikan. Penggunaan operator ternary ini memungkinkan program menentukan hasil suatu kondisi tanpa perlu menuliskan blok if-else, sehingga kode menjadi lebih ringkas dan mudah dibaca. Operator ini sering digunakan dalam situasi di mana hanya ada dua kemungkinan hasil, seperti memilih nilai minimum atau maksimum di antara dua angka.

Program 9

ouput :

```
D:\PBO\Tugas1>javac Ekspresi1.java
D:\PBO\Tugas1>java Ekspresi1.java
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
D:\PBO\Tugas1>_
```

keterangan :

Program **Ekspresil.java** menunjukkan cara Java menangani pembagian bilangan bulat dan penggunaan **casting**. Saat dua int dibagi, hasilnya dibulatkan ke bawah, misalnya $1 / 2$ menghasilkan 0 dan $10 / 3$ menghasilkan 3. Untuk mendapatkan hasil desimal, program menggunakan **casting** dengan float, seperti $(float)x / (float)y$ atau

menggunakan variabel float (fx / fy). Dengan casting, hasil pembagian tetap dalam bentuk pecahan, misalnya 1.0 / 2.0 menghasilkan 0.5.

Program 10

output :

```
D:\PBO\Tugas1>javac Hello.java

D:\PBO\Tugas1>java Hello.java
Hello
Hello World
Welcome

D:\PBO\Tugas1>_
```

keterangan :

Program Hello.java menampilkan teks ke layar menggunakan System.out.print() dan System.out.println(). System.out.print("Hello") mencetak "Hello" tanpa pindah baris, sedangkan System.out.print("\nHello ") mencetak "Hello " setelah berpindah ke baris baru karena adanya \n. System.out.println("World") mencetak "World" dan otomatis berpindah ke baris berikutnya. Terakhir, System.out.println("Welcome") mencetak "Welcome" di baris baru. Program ini menunjukkan perbedaan print(), yang tidak berpindah baris, dan println(), yang langsung berpindah ke baris berikutnya setelah mencetak teks.

Program 11

output :

```
D:\PBO\Tugas1>javac Incr.java

D:\PBO\Tugas1>java Incr.java
Nilai i : 5
Nilai j : 3

D:\PBO\Tugas1>_
```

keterangan :

Program ini menunjukkan bagaimana operator increment (++) bekerja, dengan menyoroti perbedaan antara pre-increment (++i) dan post-increment (i++). Awalnya, sebuah variabel i bertipe int dideklarasikan dan diberi nilai 3. Kemudian, terdapat pernyataan j = i++; yang menggunakan post-increment. Dalam kasus ini, nilai i sebelum peningkatan (3) disimpan ke j, sementara i sendiri bertambah menjadi 4 setelahnya. Selanjutnya, pada pernyataan System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j);, pre-increment (++i) digunakan, yang berarti nilai i ditingkatkan terlebih dahulu menjadi 5 sebelum ditampilkan. Sedangkan j tetap menyimpan nilai 3

karena saat pengisian j, post-increment membuatnya menerima nilai i sebelum perubahan terjadi.

Program 12

output :

```
D:\PBO\Tugas1>javac Oper1.java
D:\PBO\Tugas1>java Oper1.java
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
D:\PBO\Tugas1>_
```

keterangan :

Program ini mendemonstrasikan penggunaan operasi bitwise pada bilangan biner. Variabel n = 10 (1010), x = 1 (0001), dan y = 2 (0010) pertama-tama ditampilkan. Kemudian, dilakukan beberapa operasi: n & 8 (1010 AND 1000) menghasilkan 8 karena hanya bit yang bernilai 1 di kedua operand yang dipertahankan, sedangkan x & ~8 menggunakan komplement bitwise ~8 (1111 0111) sehingga hasilnya tetap 1. Operasi y << 2 menggeser bit y dua langkah ke kiri, mengubahnya menjadi 1000 (8 dalam desimal), sementara y >> 3 menggeser y tiga langkah ke kanan, menghasilkan 0000 (0). Hasil ini memperlihatkan bagaimana operator bitwise bekerja dalam manipulasi angka biner.

Program 13

output :

```
D:\PBO\Tugas1>javac Oper2.java
D:\PBO\Tugas1>java Oper2.java
i = 3
j = 4
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
D:\PBO\Tugas1>_
```

keterangan :

Program ini mendemonstrasikan penggunaan operator bitwise dan relasional pada variabel char. Variabel i memiliki nilai 3 (00000011 dalam biner) dan j bernilai 4 (00000100 dalam biner). Operasi i & j (AND) menghasilkan 0 (00000000) karena tidak ada bit yang sama-sama bernilai 1, sedangkan i | j (OR) menghasilkan 7 (00000111) karena menggabungkan bit yang bernilai 1 dari kedua operand. Operasi i

$\wedge j$ (XOR) juga menghasilkan 7 karena hanya bit yang berbeda yang menjadi 1. Selain itu, pemangkatan dengan `Math.pow(i, j)` menghitung 3^4 dan menghasilkan 81. Terakhir, operasi `~i` (NOT) membalik semua bit dalam representasi komplement dua, mengubah 3 menjadi -4 (11111100 dalam biner). Program ini menggambarkan bagaimana operator bitwise dan fungsi pemangkatan bekerja dalam Java.

Program 14

output :

```
D:\PBO\Tugas1>javac Oper3.java
D:\PBO\Tugas1>java Oper3.java
true
false
true
true
true
D:\PBO\Tugas1>
```

keterangan :

Program ini menunjukkan perbedaan antara operator logika (`&&`, `||`) dan operator bitwise (`&`, `|`) dalam Java. Pada kondisi pertama, ekspresi `true && true` bernilai `true`, sehingga hasilnya dicetak sebagai `true`. Selanjutnya, `true & true` juga menghasilkan `true`, namun ada kesalahan dalam `System.out.println(true & false)`, yang seharusnya menggunakan `true & true`. Jika diperbaiki, outputnya akan menjadi `false`. Kemudian, pernyataan `if (true)` akan mencetak `true` karena kondisi selalu benar. Pada ekspresi `true || true`, hasilnya tetap `true`, sehingga `true` dicetak. Terakhir, operator bitwise OR pada `true | false` menghasilkan `true`, sehingga outputnya tetap `true`. Program ini memperlihatkan bagaimana operator logika dan bitwise bekerja serta perbedaannya dalam evaluasi kondisi.

Program 15

output :

```
D:\PBO\Tugas1>javac Oper4.java
D:\PBO\Tugas1>java Oper4.java
Nilai e = 10
Nilai k = 0
Nilai k = 4
D:\PBO\Tugas1>
```

keterangan :

Program ini mendemonstrasikan operator ternary (`?:`) dalam Java. Variabel `c = 8` dan `d = 10` dibandingkan, karena `8 > 10` salah, `d` dipilih, sehingga `e = 10`. Kemudian, `i = 0`

dan $j = 0$ dibandingkan, hasilnya $k = 0$ karena $0 > 0$ salah. Setelah $i = 2$ dan $j = 3$, ekspresi $i++ > j++$ dievaluasi, karena $2 > 3$ salah, j dipilih, lalu bertambah menjadi 4, sehingga $k = 4$.

Program 16

output :

```
D:\PBO\Tugas1>javac Oprator.java
D:\PBO\Tugas1>java Oprator.java
Hasil dari berbagai operasi:

Boolean AND      : false
Boolean OR       : true
Boolean NOT      : false
Boolean XOR      : true

Operasi Numerik (Integer):
Penjumlahan      : 7
Pengurangan     : 3
Perkalian        : 10
Pembagian bulat  : 2
Modulo           : 1

Operasi Numerik (Float):
Penjumlahan      : 10.0
Pengurangan     : 0.0
Perkalian        : 25.0
Pembagian        : 1.0

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

D:\PBO\Tugas1>
```

keterangan :

Program ini menunjukkan berbagai operasi pada tipe data, termasuk logika, aritmatika, dan relasional. Pertama, program mencetak teks pengantar sebelum menjalankan operasi logika dengan $Bool1 = true$ dan $Bool2 = false$. Hasil $Bool1 \&\& Bool2$ adalah false, $Bool1 \parallel Bool2$ bernilai true, $!Bool1$ mengubah true menjadi false, dan $Bool1 \wedge Bool2$ menghasilkan true. Selanjutnya, operasi aritmatika dengan $i = 5$ dan $j = 2$ menghasilkan $i + j = 7$, $i - j = 3$, $i * j = 10$, $i / j = 2$ (karena dibulatkan), dan $i \% j = 1$. Pada operasi float dengan $x = 5.0$ dan $y = 5.0$, hasilnya adalah $x + y = 10.0$, $x - y = 0.0$, $x * y = 25.0$, dan $x / y = 1.0$. Untuk operasi relasional, $i == j$ bernilai false, $i != j$ bernilai true, $i < j$ bernilai false, $i > j$ bernilai true, $i <= j$ bernilai false, dan $i >= j$ bernilai true. Pada x dan y , karena nilainya sama, $x != y$, $x < y$, dan $x > y$ bernilai false, sedangkan $x <= y$ dan $x >= y$ bernilai true.