

Tugas I

Makassar, 21 Februari 2025

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Alexandria Kayla Kazya Putri Nur
Nim : 13020230001
Dosen : Mardiyah Hasnawi, S.Kom., M.T.

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

Tugas Program

Program 1

output :

```
D:\TUGAS 1>javac Asgdll.java

D:\TUGAS 1>java Asgdll
f : 20.0
f11: 10.0
```

keterangan :

Program ini menampilkan nilai variable f yang bertipe float dengan nilai 20.0, dan kedua menampilkan nilai variable f11 yang bertipe double dengan nilai 10.0 Meskipun f11 diberikan nilai 10.0f (yang bertipe float), Java secara otomatis mengkonversinya menjadi tipe double karena tipe double memiliki kapasitas yang lebih besar. Hasil akhirnya, kedua variabel ini dicetak dalam format yang sama, yaitu f : 20.0 dan f11 : 10.0, meskipun tipe data mereka berbeda.

Program 2

output :

```
D:\TUGAS 1>javac Asign.java

D:\TUGAS 1>java Asign
hello
Ini nilai i :5
```

keterangan :

Program ini mendeklarasikan variabel i bertipe integer. Pertama, menggunakan `System.out.print("hello\n");`, program mencetak hello dan berpindah ke baris baru. Kemudian, variabel i diberi nilai 5 dengan `i = 5;`. Selanjutnya, `System.out.println("Ini nilai i : " + i);` mencetak "Ini nilai i : " diikuti dengan nilai dari variabel i, yaitu 5. Program ini menghasilkan dua baris output: "hello" pada baris pertama dan "Ini nilai i :5" pada baris kedua.

Program 3

output :

```
D:\TUGAS 1>javac ASIGNi.java

D:\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
```

keterangan :

Program ini mendeklarasikan beberapa variabel dengan tipe data yang berbeda yaitu short ks, int ki, long kl, char c dan c1, double x, dan float y. Variabel c diinisialisasi dengan nilai 65 yang direpresentasikan sebagai karakter ASCII 'A', dan c1 diinisialisasi dengan karakter 'Z'. Program kemudian mencetak nilai variabel tersebut, baik sebagai karakter maupun nilai numeriknya ASCII. Hasil output menunjukkan pencetakan karakter c sebagai 'A' dan c1 sebagai 'Z', serta nilai integer untuk ks, ki, kl, dan nilai real untuk x dan y. Program ini menunjukkan penggunaan berbagai tipe data dan cara mencetak karakter dan nilai numeriknya.

Program 4

output :

```
D:\TUGAS 1>javac BacaData.java

D:\TUGAS 1>java BacaData
Contoh membaca dan menulis, ketik nilai integer: 1
Nilai yang dibaca: 1
```

keterangan :

Program ini menggunakan kelas Scanner untuk membaca input dari pengguna. Dimulai dengan mendeklarasikan variabel a bertipe integer dan objek Scanner bernama masukan. Program kemudian mencetak pesan yang meminta pengguna untuk memasukkan nilai integer. Setelah itu, masukan.nextInt() digunakan untuk membaca nilai yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel a. Nilai tersebut kemudian dicetak ke layar. Jika pengguna memasukkan angka, misalnya 1, maka program akan mencetak "Nilai yang dibaca: 1". Namun, jika input yang dimasukkan bukan angka, program akan menghasilkan kesalahan karena nextInt() hanya bisa menerima input integer.

Program 5

output :

```
D:\TUGAS 1>javac Bacakar.java

D:\TUGAS 1>java Bacakar
hello
baca 1 karakter : alexa
baca 1 bilangan : 8
a
8
bye
D:\TUGAS 1>|
```

keterangan :

Program ini menggunakan BufferedReader untuk membaca input dari pengguna melalui keyboard. Program diawali dengan mencetak "hello" sebagai salam pembuka. Selanjutnya, program meminta pengguna memasukkan satu karakter dengan menampilkan pesan "baca 1 karakter :". Jika pengguna mengetikkan kata seperti "alexa", program hanya mengambil karakter pertama, yaitu 'a', menggunakan charAt(0). Setelah itu, program meminta pengguna memasukkan satu bilangan dengan menampilkan "baca 1 bilangan :". Input yang diberikan,

misalnya 8, dibaca sebagai string lalu dikonversi menjadi bilangan bulat (int) menggunakan Integer.parseInt(). Program kemudian mencetak karakter 'a' dan angka 8 sebagai hasil input. Terakhir, program mencetak "bye", menandakan bahwa eksekusi telah selesai. Program ini memanfaatkan BufferedReader untuk membaca input secara efisien, tetapi memiliki keterbatasan dalam membaca karakter karena hanya mengambil huruf pertama dari input string.

Program 6

output :

```
D:\TUGAS 1>javac Casting1.java
D:\TUGAS 1>java Casting1
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14
```

keterangan :

Program ini menggunakan beberapa variabel dideklarasikan dengan tipe data berbeda, seperti int, float, char, dan double. Selanjutnya, program melakukan berbagai konversi dan mencetak hasilnya. Misalnya, int dikonversi ke float atau double tanpa kehilangan data, sedangkan float yang dikonversi ke int akan kehilangan bagian desimalnya. Untuk karakter (char), ketika dikonversi ke int, hasilnya adalah nilai ASCII dari karakter tersebut. Dalam kasus ini, karakter '5' memiliki nilai ASCII 53, sehingga (int) 5 menghasilkan 53, dan konversi ke float atau double tetap mempertahankan nilai 53.0. Selain itu, program juga menunjukkan bahwa mengonversi double ke int akan membuang bagian desimalnya. Misalnya, nilai 3.14 saat dikonversi ke int menjadi 3, sedangkan konversi ke float tetap mempertahankan 3.14. Kesimpulannya, casting eksplisit diperlukan saat melakukan narrowing conversion untuk menghindari kehilangan data, sementara widening conversion terjadi secara otomatis tanpa risiko kehilangan informasi.

Program 7

output :

```
D:\TUGAS 1>javac Casting2.java
D:\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
```

keterangan :

Program ini mendemonstrasikan konversi tipe data menggunakan kelas wrapper di Java, yang memungkinkan perubahan antara String, Integer, Float, Double, dan char. Pertama, program mengonversi nilai string menjadi tipe numerik menggunakan metode `parseInt()`, `parseDouble()`, dan `parseFloat()`. Hasilnya, string "67" dikonversi menjadi int 67, "45" menjadi double 45.0, dan "100" menjadi float 100.0. program mengubah nilai numerik ke string menggunakan `String.valueOf()`, di mana int 9 dikonversi ke "9", char '5' menjadi "5", dan float 3.2 menjadi "3.2". program melakukan konversi Integer ke Double dan sebaliknya. int 67 dikonversi ke Double, lalu kembali menjadi int, sedangkan int 9 diubah menjadi double 9.0. Program ini menunjukkan bagaimana Java menangani casting tipe data dengan kelas wrapper, yang sering digunakan dalam pemrosesan input, konversi data, dan manipulasi variabel dalam program yang lebih kompleks.

Program 8

output :

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

keterangan :

Program ini mendemonstrasikan penggunaan operator kondisional (ternary operator) dalam Java, yang berfungsi sebagai alternatif singkat dari pernyataan if else. Di dalam program, dua variabel x dan y dengan nilai awal 1 dan 2 dideklarasikan. Program kemudian mencetak nilai kedua variabel sebelum mengevaluasi ekspresi ternary `((x < y) ? x : y)`. Karena kondisi `x < y` bernilai true, maka ekspresi mengembalikan nilai x. Jika kondisi tersebut bernilai false, ekspresi akan mengembalikan nilai y. Dengan menggunakan operator ternary, program dapat menentukan hasil dari suatu kondisi tanpa perlu menuliskan if else, membuat kode lebih ringkas dan efisien.

Program 9

ouput :

```
D:\TUGAS 1>javac Ekspresi1.java
D:\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
```

keterangan :

Program Ekspresi1.java menunjukkan bagaimana Java menangani pembagian integer dan casting. Saat dua angka bertipe int dibagi, hasilnya akan berupa bilangan bulat karena Java secara otomatis membulatkan ke bawah dan mengabaikan nilai pecahan. Hal ini terlihat saat $x = 1$ dan $y = 2$, di mana x / y menghasilkan 0 karena hasil asli 0.5 dibulatkan ke bawah. Begitu pula ketika $x = 10$ dan $y = 3$, hasil $10 / 3$ tetap 3 tanpa desimal. Untuk mendapatkan hasil pecahan, program mengubah x dan y menjadi float dengan menggunakan variabel fx dan fy . Saat dilakukan pembagian fx / fy , hasilnya adalah 0.5, sesuai dengan operasi pembagian bilangan desimal. Selain itu, program juga menggunakan casting eksplisit $(float)x / (float)y$ untuk memastikan bahwa hasil pembagian tetap dalam bentuk desimal.

Program 10

output :

```
D:\TUGAS 1>javac Hello.java

D:\TUGAS 1>java Hello
Hello
Hello World
Welcome
```

keterangan :

Program Hello.java menampilkan teks ke layar menggunakan perintah `System.out.print()` dan `System.out.println()`. Perintah `System.out.print("Hello");` mencetak kata "Hello" tanpa pindah ke baris baru. Kemudian, `System.out.print("\nHello ");` mencetak "Hello " dengan karakter `\n`, yang menyebabkan teks dipindahkan ke baris berikutnya sebelum mencetak kata "Hello ". Setelah itu, `System.out.println("World");` mencetak kata "World" dan secara otomatis berpindah ke baris berikutnya karena menggunakan `println()`. Terakhir, `System.out.println("Welcome");` mencetak kata "Welcome" di baris baru.

Program 11

output :

```
D:\TUGAS 1>javac Incr.java

D:\TUGAS 1>java Incr
Nilai i : 5
Nilai j : 3
```

keterangan :

Program ini mendemonstrasikan efek dari operator increment (`++`), khususnya perbedaan antara pre-increment (`++i`) dan post-increment (`i++`). Program dimulai dengan deklarasi variabel `int i` dan `int j`, kemudian `i` diinisialisasi dengan nilai 3. Selanjutnya, terdapat pernyataan `j = i++;` yang menggunakan post-increment, di mana nilai `i` yang lama (3) disimpan ke dalam `j`, lalu `i` bertambah menjadi 4. Pada pernyataan `System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j);`, operator pre-increment (`++i`) digunakan, sehingga nilai `i` bertambah dulu menjadi 5 sebelum dicetak. Sementara itu, `j` tetap bernilai 3 karena saat penyimpanan ke `j` sebelumnya, nilai `i` yang lama digunakan sebelum peningkatan terjadi.

Program 12

output :

```
D:\TUGAS 1>javac Oper1.java

D:\TUGAS 1>java Oper1
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

keterangan :

Program ini mendemonstrasikan operasi bitwise pada bilangan biner. Variabel $n = 10$ (1010), $x = 1$ (0001), dan $y = 2$ (0010) dicetak terlebih dahulu. Kemudian, $n \& 8$ (1010 AND 1000) menghasilkan 8. Operasi $x \& \sim 8$ menggunakan komplemen bitwise ~ 8 (1111 0111) dan menghasilkan 1. Operasi $y \ll 2$ (left shift) menggeser bit y dua posisi ke kiri menjadi 1000 (8). Terakhir, $y \gg 3$ (right shift) menggeser y tiga posisi ke kanan menjadi 0000 (0). Output ini menunjukkan bagaimana operasi bitwise bekerja dalam manipulasi angka biner.

Program 13

output :

```
D:\TUGAS 1>javac Oper2.java

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

keterangan :

Program ini menggunakan operator bitwise dan relasional pada variabel char. $i = 3$ (00000011 biner) dan $j = 4$ (00000100 biner). Operasi $i \& j$ (AND) menghasilkan 0 (00000000) karena tidak ada bit yang sama-sama 1. $i | j$ (OR) menghasilkan 7 (00000111), dan $i \wedge j$ (XOR) juga 7 karena bit yang berbeda menjadi 1. Pemangkatan $\text{Math.pow}(i, j)$ menghitung $3^4 = 81$. Terakhir, $\sim i$ (NOT) mengubah 3 menjadi -4 dalam bentuk komplemen dua (11111100). Program ini menunjukkan cara kerja operasi bitwise dan pemangkatan dalam Java.

Program 14

output :

```
D:\TUGAS 1>javac Oper3.java

D:\TUGAS 1>java Oper3
true
false
true
true
true
```

keterangan :

Program ini mendemonstrasikan penggunaan operator logika (&&, ||) dan bitwise (&, |) dalam Java. Pada kondisi pertama, true && true menghasilkan true, sehingga true dicetak. Kondisi kedua menggunakan true & true, yang juga bernilai true, tetapi kesalahan terjadi pada System.out.println(true & false), karena seharusnya true & true. Jika diperbaiki, outputnya akan false. Selanjutnya, pernyataan if (true) mencetak true karena kondisinya selalu benar. Kondisi true || true juga bernilai true, sehingga mencetak true. Terakhir, true | false menggunakan operator bitwise OR, yang menghasilkan true dan mencetak true. Program ini menunjukkan perbedaan antara operator logika dan bitwise dalam Java.

Program 15

output :

```
D:\TUGAS 1>javac Oper4.java

D:\TUGAS 1>java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4
```

keterangan :

Program ini mendemonstrasikan penggunaan operator ternary (?:) dalam Java. Dimulai dengan inisialisasi i = 0, j = 0, c = 8, dan d = 10. Operator ternary pertama (((int)c > (int)d) ? c : d) membandingkan c (8) dan d (10). Karena 8 > 10 adalah false, maka nilai d (10) dipilih dan disimpan di e, sehingga output pertama adalah Nilai e = 10. Selanjutnya, operator ternary ((i > j) ? i : j) membandingkan i = 0 dan j = 0. Karena 0 > 0 adalah false, maka j dipilih, sehingga k = 0, dan output kedua adalah Nilai k = 0. Kemudian, nilai i diubah menjadi 2 dan j menjadi 3. Operator ternary ((i++ > j++) ? i : j) membandingkan i = 2 dan j = 3. Karena 2 > 3 adalah false, maka j dipilih. Namun, karena operator j++ digunakan, nilai j meningkat menjadi 4 setelah evaluasi, sehingga k = 4 dan output terakhir adalah Nilai k = 4.

Program 16

output :

```
D:\TUGAS 1>javac Oprator.java

D:\TUGAS 1>java Oprator
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output
Bool1 && Bool2: false
Bool1 || Bool2: true
!Bool1: false
Bool1 ^ Bool2: true
i + j: 7
i - j: 3
i * j: 10
i / j: 2
i % j: 1
x + y: 10.0
x - y: 0.0
x * y: 25.0
x / y: 1.0
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
```


keterangan :

Program ini mendemonstrasikan berbagai operasi pada tipe data termasuk operasi logika, aritmatika, dan relasional. Pertama, program mencetak teks pengantar: "Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output". Selanjutnya, program melakukan operasi logika dengan variabel Boolean Bool1 = true dan Bool2 = false. Hasil operasi Bool1 && Bool2 (AND) adalah false karena salah satu operand bernilai false, sedangkan Bool1 || Bool2 (OR) bernilai true karena salah satu operand true. Operasi !Bool1 (NOT) membalik nilai true menjadi false, dan Bool1 ^ Bool2 (XOR) menghasilkan true karena operand memiliki nilai yang berbeda. Kemudian, program melakukan operasi aritmatika menggunakan variabel i = 5 dan j = 2. Operasi penjumlahan i + j menghasilkan 7, pengurangan i - j menghasilkan 3, perkalian i * j menghasilkan 10, dan pembagian bulat i / j menghasilkan 2 karena hasil desimal dibulatkan ke bawah. Operasi i % j (modulo) menghasilkan 1, yaitu sisa dari pembagian 5 oleh 2. Setelah itu, operasi aritmatika juga diterapkan pada angka desimal (float) dengan x = 5.0 dan y = 5.0. Hasil operasi x + y adalah 10.0, x - y adalah 0.0, x * y adalah 25.0, dan x / y adalah 1.0. Selanjutnya, program membandingkan nilai i dan j menggunakan operasi relasional. Hasil i == j adalah false karena 5 tidak sama dengan 2, sedangkan i != j adalah true karena keduanya berbeda. Operasi i < j menghasilkan false karena 5 lebih besar dari 2, sementara i > j adalah true. Operasi i <= j menghasilkan false, dan i >= j menghasilkan true. Operasi yang sama dilakukan pada variabel x dan y. Karena x dan y memiliki nilai yang sama, hasil x != y, x < y, dan x > y adalah false, sedangkan x <= y dan x >= y bernilai true.