Tugas 1 Makassar, 21 Februari 2025

# TEORI

# PEMROGRAMAN BERBASIS OBJEK



Nama : Nisa Awaliyah Fazni

Nim : 13020230028

Frekuensi : TI\_PBO (B1)

Dosen : Mardiyyah Hasnawi, S.Kom.,M.T.,MTA.

# PROGRAM STUDI TEKNIK INFORMATIKA

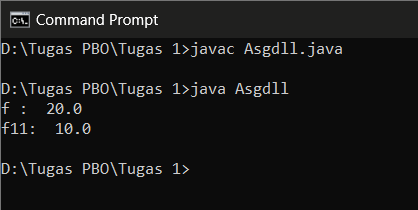
# FAKULTAS ILMU KOMPUTER

# UNIVERSITAS MUSLIM INDONESIA

# MAKASSAR

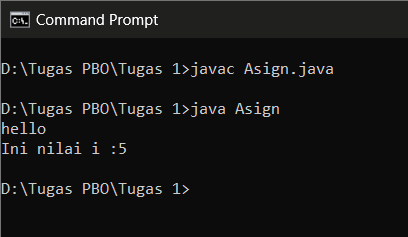
**2025**

Program 1 :



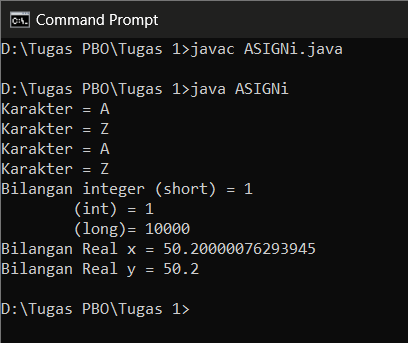
Keterangan : Program Java ini mendeklarasikan variabel f bertipe **float** dengan nilai **20.0f** dan fll bertipe **double** dengan nilai **10.0f**. Kedua nilai tersebut kemudian dicetak menggunakan System.out.println(), dengan \n untuk membuat baris baru. Saat dijalankan di Command Prompt, output yang muncul adalah nilai variabel f dan fll, yaitu **20.0** dan **10.0**. Program berjalan tanpa error karena tipe data sudah sesuai, dan penggunaan f pada angka memastikan bahwa nilai tersebut dikenali sebagai **float** oleh Java.

Program 2 :



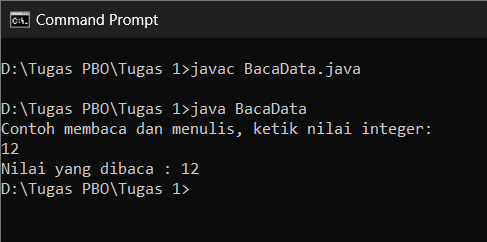
Keterangan : Program Java ini mendeklarasikan variabel i bertipe **int,** lalu mencetak teks "hello" dengan \n untuk pindah baris. Setelah itu, variabel i diberi nilai **5** dan dicetak menggunakan System.out.println(). Saat dijalankan di Command Prompt, output yang dihasilkan adalah "hello" di baris pertama, lalu "Ini nilai i :5" di baris berikutnya. Program berjalan dengan baik karena sintaksnya benar dan variabel telah diinisialisasi sebelum digunakan.

Program 3 :



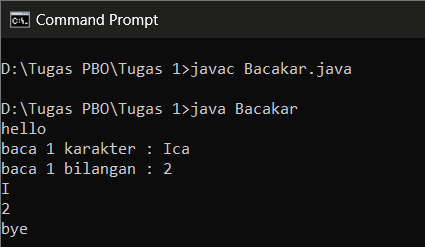
Keterangan : Program Java ini mendeklarasikan beberapa variabel dengan tipe data berbeda, termasuk short, int, long, char, double, dan float. Variabel c diberi nilai **65**, yang dikonversi menjadi karakter **'A'**, sedangkan c1 langsung diinisialisasi dengan **'Z'**. Variabel numerik dicetak dengan format berbeda untuk menunjukkan berbagai tipe data. Saat dijalankan di Command Prompt, program mencetak karakter 'A' dan 'Z' beberapa kali, serta menampilkan nilai integer dan bilangan real dengan tingkat presisi yang berbeda.

Program 4:



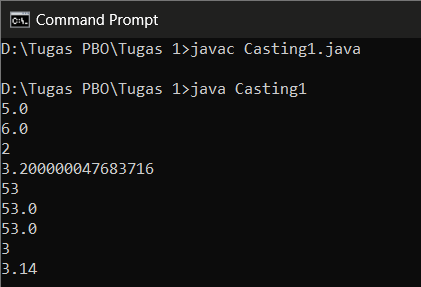
Keterangan : Program ini menggunakan **Scanner** untuk membaca input angka dari pengguna. Setelah menampilkan pesan **"Contoh membaca dan menulis, ketik nilai integer:"**, program menunggu pengguna memasukkan bilangan integer. Nilai yang dimasukkan kemudian disimpan dalam variabel a dan ditampilkan kembali dalam format **"Nilai yang dibaca : [angka]"**. Jika pengguna memasukkan angka **12** , maka output yang muncul adalah **"Nilai yang dibaca : 12"**. Namun, jika input bukan angka, program akan mengalami error karena nextInt() hanya menerima bilangan bulat.

Program 5 :



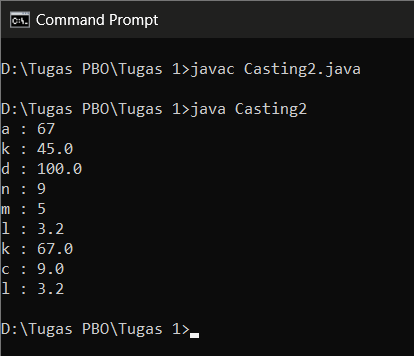
Keterangan : Program ini menggunakan **BufferedReader** untuk membaca input dari pengguna. Setelah menampilkan pesan **"hello"**, program meminta pengguna memasukkan satu karakter yang disimpan dalam variabel cc, lalu meminta input bilangan bulat yang disimpan dalam bil. Kedua nilai ini kemudian ditampilkan kembali di layar. Jika pengguna memasukkan **'A'** dan **10**, maka outputnya adalah **"A\n10"**, diakhiri dengan pesan **"bye"**. Program akan error jika input bilangan bukan angka karena Integer.parseInt() hanya menerima bilangan bulat.

Program 6 :

****

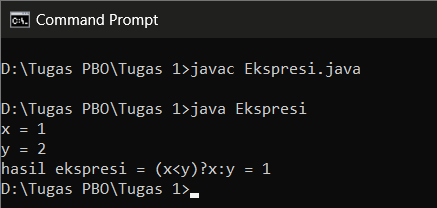
Keterangan : Program ini mendemonstrasikan **type casting** dalam Java dengan mengonversi berbagai tipe data, seperti int, float, char, dan double. Variabel a dan b dikonversi ke **float** dan **double**, sedangkan d dan e dikonversi ke **int** dan **double**. Karakter '5' dikonversi ke **int** berdasarkan kode ASCII-nya (**53**), lalu ke **float** dan **double**. Nilai **3.14** dalam k juga dikonversi ke **int** dan **float**. Hasilnya ditampilkan dalam output sesuai dengan tipe data setelah konversi.

Program 7 :



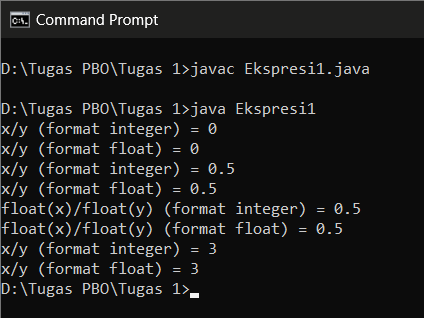
Keterangan : Program ini menunjukkan penggunaan **type casting** dan **konversi tipe data** dalam Java. Nilai string seperti "67", "45", dan "100" dikonversi menjadi int, double, dan float, kemudian ditampilkan sebagai **a : 67, k : 45.0, dan d : 100.0**. Sebaliknya, nilai numerik dikonversi ke string menggunakan String.valueOf(). Selanjutnya, nilai a dikonversi ke double, dan b dikonversi ke double juga, yang menghasilkan output akhir **k : 67.0, c : 9.0, dan l : 3.2**.

Program 8 :



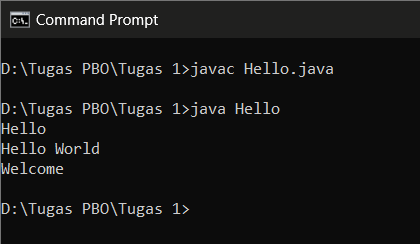
Keterangan : Program ini mendemonstrasikan penggunaan **operator ternary (**?:**)** dalam Java. Variabel x dan y masing-masing diberi nilai **1** dan **2**. Program mengevaluasi ekspresi (x < y) ? x : y, yang berarti jika x lebih kecil dari y, maka hasilnya adalah x, jika tidak, hasilnya y. Karena **1 < 2** bernilai **true**, maka hasilnya adalah **1**, yang kemudian ditampilkan sebagai **"hasil ekspresi = (x<y)?x:y = 1"**.

Program 9 :



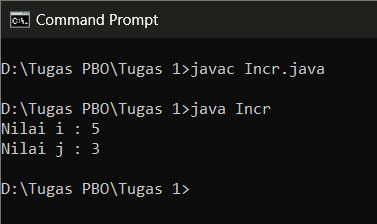
Keterangan : Program ini mendemonstrasikan perbedaan **pembagian dalam tipe data integer dan float** di Java. Saat membagi dua bilangan bulat (int), hasilnya akan berupa integer (**x/y = 0**). Jika salah satu operand dikonversi ke float, hasilnya akan dalam bentuk desimal (**x/y = 0.5**). Konversi eksplisit menggunakan float(x)/float(y) memastikan hasil pembagian tetap dalam format desimal. Selain itu, program juga menunjukkan pembagian integer dengan hasil lebih besar (**x/y = 3**), yang mungkin berasal dari nilai berbeda dalam perhitungan lainnya.

Program 10 :



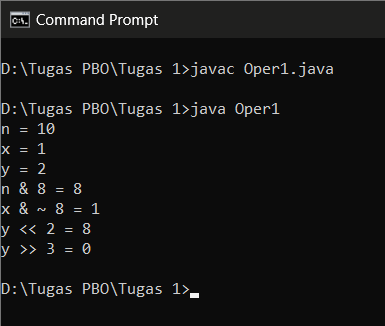
Keterangan : Program ini mendemonstrasikan **perbedaan antara** print **dan** println **dalam Java.** Pernyataan System.out.print("Hello") mencetak teks tanpa pindah baris, sedangkan System.out.print("\nHello ") menambahkan baris baru sebelum mencetak "Hello ". Kemudian, System.out.println("World") mencetak "World" dan langsung pindah ke baris baru, diikuti oleh System.out.println("Welcome") yang mencetak "Welcome" di baris berikutnya. Hasil akhirnya adalah tiga baris output: "Hello", "Hello World", dan "Welcome".

Program 11 :



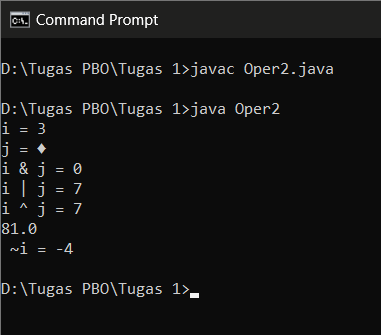
Keterangan : Program ini mendemonstrasikan **operator increment (++) dalam Java**. Variabel i diinisialisasi dengan nilai 3, lalu j = i++ menyimpan nilai i ke j sebelum i bertambah menjadi 4. Kemudian, ++i menaikkan i menjadi 5 sebelum dicetak. Akhirnya, output menampilkan **"Nilai i : 5"** dan **"Nilai j : 3"**, karena j menyimpan nilai awal i sebelum peningkatan pertama terjadi.

Program 12 :



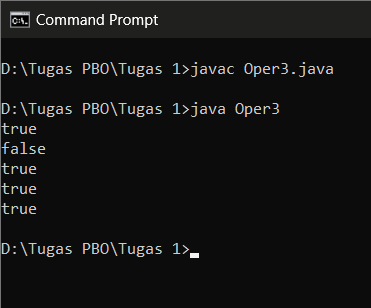
Keterangan : Program ini mendemonstrasikan **operator bitwise dalam Java**. Operator & melakukan operasi AND bitwise antara n dan 8, menghasilkan 8. Operator ~ (NOT) membalik semua bit dari 8, lalu AND dengan x, menghasilkan 1. Operator << (left shift) menggeser bit y ke kiri sebanyak 2, menghasilkan 8, sementara >> (right shift) menggeser y ke kanan sebanyak 3, menghasilkan 0. Output program mencerminkan hasil dari operasi bitwise ini.

Program 13 :



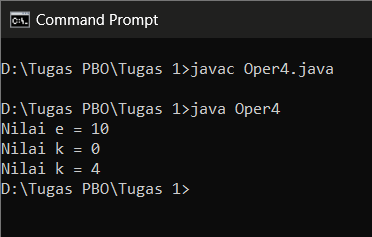
Keterangan : Program ini mendemonstrasikan **operator bitwise** pada karakter bertipe char, yang dikonversi ke nilai numerik sebelum operasi. Operator & (AND) antara i dan j menghasilkan 0, | (OR) menghasilkan 7, dan ^ (XOR) juga menghasilkan 7. Perhitungan Math.pow(i, j) menghitung 3^4 = 81.0. Operator ~ (bitwise NOT) membalik bit dari i, menghasilkan -4. Hasil ini menunjukkan cara kerja manipulasi bit dalam Java.

Program 14 :



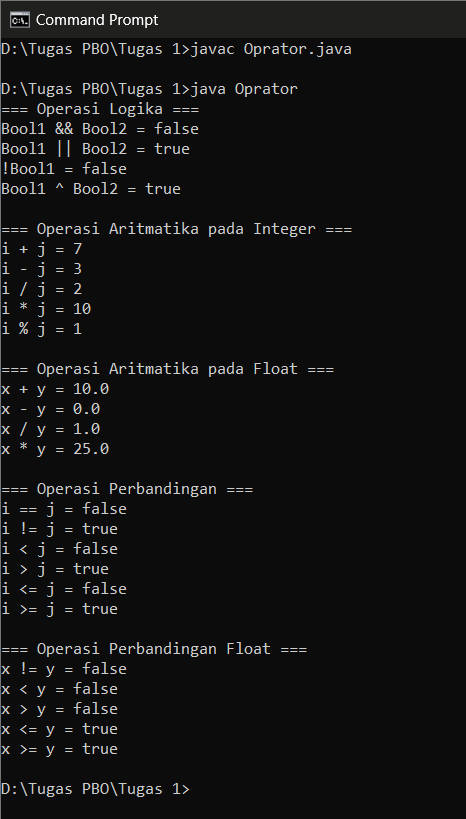
Keterangan : Program ini mendemonstrasikan penggunaan **operator logika** (&&, ||, &, dan |) dalam Java. Operator && (AND) mengevaluasi kedua kondisi dan mencetak true, sedangkan & (bitwise AND) juga mengevaluasi kedua kondisi tetapi mencetak false karena salah satu bernilai false. Operator || (OR) mengevaluasi kondisi pertama dan mencetak true, sedangkan | (bitwise OR) juga mencetak true karena salah satu kondisi true. Hasil output menunjukkan bagaimana operator ini bekerja dalam ekspresi logika.

Program 15 :



Keterangan : Program ini menggunakan **operator ternary (? :)** untuk menentukan nilai terbesar antara dua variabel. Variabel e mengambil nilai maksimum antara c (8) dan d (10), sehingga e = 10. Variabel k pertama kali membandingkan i dan j, yang awalnya 0, sehingga k = 0. Kemudian, setelah i diubah menjadi 2 dan j menjadi 3, ekspresi (i++ > j++) dievaluasi, tetapi karena i tidak lebih besar dari j, maka k diisi dengan nilai j setelah peningkatan, yaitu 4.

Program 16 :



Keterangan : Program Oprator.java ini mendemonstrasikan berbagai operasi logika, aritmatika, dan perbandingan dalam Java. Pada bagian operasi logika, program mengevaluasi ekspresi dengan operator AND (&&), OR (||), NOT (!), dan XOR (^). Kemudian, pada operasi aritmatika, program melakukan perhitungan dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus untuk tipe data integer dan float. Selanjutnya, program membandingkan nilai integer dan float menggunakan operator perbandingan (==, !=, <, >, <=, >=). Hasil eksekusi menunjukkan bagaimana operasi-operasi ini bekerja dan bagaimana Java menangani tipe data yang berbeda dalam berbagai konteks perhitungan.