

LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK



Nama : M. Akbar Amiruddin
Nim : 13020220023

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

1. Asgdll.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut Asgdll.

Metode Main:

- Metode main adalah titik masuk utama program Java. Program Java selalu memulai eksekusi dari metode main. Metode ini menerima argumen berupa array String (String[] args). Argumen ini dapat digunakan untuk berinteraksi dengan program dari baris perintah.

Variabel dan Tipe Data:

- float f = 20.0f;; Mendeklarasikan variabel f dengan tipe data float dan memberinya nilai awal 20.0.
- double fl; Mendeklarasikan variabel fl dengan tipe data double tanpa memberikan nilai awal.

Inisialisasi Variabel:

- fl = 10.0f;; Memberikan nilai 10.0 pada variabel fl. Meskipun nilai ini bertipe float, itu bisa diterima dalam variabel double karena konversi tipe data secara implisit.

Menampilkan Output:

- System.out.println("f : " + f + "\nf11: " + fl); Baris ini mencetak nilai dari variabel f dan fl ke konsol. Tanda + digunakan untuk menggabungkan teks dan nilai variabel. \n digunakan untuk memberi baris baru.

Keseluruhan program ini sederhana, ia hanya mendeklarasikan beberapa variabel, memberi nilai pada beberapa variabel, dan kemudian mencetak nilai-nilai tersebut ke konsol.

2. Asign.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut Asign.

Metode Main:

- Metode main adalah titik masuk utama program Java. Ini adalah tempat di mana eksekusi program dimulai. Metode ini menerima argumen berupa array String (String[] args), meskipun dalam kasus ini, argumen tersebut tidak digunakan.

Deklarasi Variabel:

- `int i;` Mendeklarasikan variabel `i` dengan tipe data integer tanpa memberikan nilai awal.

Menampilkan Output:

- `System.out.print("hello\n");` Baris ini mencetak string "hello" ke konsol tanpa membuat baris baru. `\n` digunakan untuk membuat baris baru setelah string "hello" tercetak.
- `i = 7;` Memberikan nilai 7 pada variabel `i`.

Menampilkan Output Lagi:

- `System.out.println("Ini nilai i : " + i);` Baris ini mencetak teks "Ini nilai i :" diikuti oleh nilai dari variabel `i` ke konsol. Tanda `+` digunakan untuk menggabungkan teks dan nilai variabel.

3. ASIGNi.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut `ASIGNi`.

Metode Main:

- Metode main adalah titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `short ks = 1;` Mendeklarasikan variabel `ks` dengan tipe data short dan memberikannya nilai awal 1.
- `int ki = 1;` Mendeklarasikan variabel `ki` dengan tipe data integer dan memberikannya nilai awal 1.
- `long kl = 10000;` Mendeklarasikan variabel `kl` dengan tipe data long dan memberikannya nilai awal 10000.
- `char c = 65;` Mendeklarasikan variabel `c` dengan tipe data char dan memberikannya nilai 65 yang mewakili karakter 'A' dalam ASCII.
- `char c1 = 'Z';` Mendeklarasikan variabel `c1` dengan tipe data char dan memberikannya nilai 'Z'.
- `double x = 50.2f;` Mendeklarasikan variabel `x` dengan tipe data double dan memberikannya nilai awal 50.2.
- `float y = 50.2f;` Mendeklarasikan variabel `y` dengan tipe data float dan memberikannya nilai awal 50.2.

Menampilkan Output: Program mencetak nilai dari variabel-variabel yang telah dideklarasikan ke konsol menggunakan `System.out.println()`.

- `System.out.println("Karakter = " + c);` Mencetak nilai karakter dari variabel `c`.

- `System.out.println("Karakter = "+ c1);`; Mencetak nilai karakter dari variabel `c1`.
- `System.out.println("Bilangan integer (short) = "+ ks);`; Mencetak nilai dari variabel `ks`.
- `System.out.println("\t(int) = "+ ki);`; Mencetak nilai dari variabel `ki`.
- `System.out.println("\t(long)= "+ kl);`; Mencetak nilai dari variabel `kl`.
- `System.out.println("Bilangan Real x = "+ x);`; Mencetak nilai dari variabel `x`.
- `System.out.println("Bilangan Real y = "+ y);`; Mencetak nilai dari variabel `y`.

4. BacaData.java

Import Pustaka:

- Baris pertama `import java.util.Scanner;` mengimpor pustaka `Scanner` dari paket `java.util`. Pustaka ini memungkinkan program untuk membaca input dari pengguna melalui konsol.

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut `BacaData`.

Metode Main:

- Metode main adalah titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `int a;` Mendeklarasikan variabel `a` dengan tipe data integer yang akan digunakan untuk menyimpan nilai yang akan dibaca dari input pengguna.
- `Scanner masukan;` Mendeklarasikan variabel `masukan` dengan tipe data `Scanner`. Variabel ini akan digunakan untuk mengambil input dari pengguna.

Membaca Input dari Pengguna:

- `masukan = new Scanner(System.in);`; Menciptakan objek `Scanner` baru dengan menggunakan objek `System.in` sebagai sumber input. `System.in` mewakili input standar, yang pada umumnya adalah keyboard.
- `a = masukan.nextInt();`; Membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel `a`.

Menampilkan Output:

- `System.out.print("Nilai yang dibaca : " + a);`; Mencetak nilai yang telah dibaca dari input pengguna ke konsol.

5. Bacakar.java

Import Pustaka:

- Kode mengimpor pustaka yang diperlukan untuk melakukan operasi input-output, yaitu `java.io.BufferedReader`, `java.io.IOException`, dan `java.io.InputStreamReader`.

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut `Bacakar`.

Metode Main:

- Metode main adalah titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `char cc`; Mendeklarasikan variabel `cc` dengan tipe data `char` untuk menyimpan satu karakter.
- `int bil`; Mendeklarasikan variabel `bil` dengan tipe data `integer` untuk menyimpan bilangan bulat.

Membuat Objek Untuk Membaca Input:

- `InputStreamReader isr = new InputStreamReader(System.in);`; Menciptakan objek `InputStreamReader` yang digunakan untuk membaca input dari `System.in` (input standar).
- `BufferedReader dataIn = new BufferedReader(isr);`; Menciptakan objek `BufferedReader` yang digunakan untuk membaca input dari `isr`.

Meminta Input dari Pengguna:

- `System.out.print("baca 1 karakter : ");`; Menampilkan pesan untuk meminta pengguna memasukkan satu karakter.
- `cc = dataIn.readLine().charAt(0);`; Membaca satu baris input dari pengguna dan mengambil karakter pertama dari baris tersebut.
- `System.out.print("baca 1 bilangan : ");`; Menampilkan pesan untuk meminta pengguna memasukkan satu bilangan.
- `bil = Integer.parseInt(dataIn.readLine());`; Membaca satu baris input dari pengguna dan mengonversinya menjadi bilangan integer.

Menampilkan Output:

- `System.out.print(cc + "\n" + bil + "\n");`; Menampilkan karakter dan bilangan yang telah dibaca dari input pengguna.
- `System.out.print("bye \n");`; Menampilkan pesan perpisahan.

6. Casting1.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut Casting1.

Metode Main:

- Metode main adalah titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `int a = 5, b = 6;`: Mendeklarasikan dua variabel bertipe integer, a dengan nilai 5 dan b dengan nilai 6.
- `float d = 2.f, e = 3.2f;`: Mendeklarasikan dua variabel bertipe float, d dengan nilai 2.0 dan e dengan nilai 3.2.
- `char g = '5';`: Mendeklarasikan variabel bertipe char g dengan nilai '5'.
- `double k = 3.14;`: Mendeklarasikan variabel bertipe double k dengan nilai 3.14.

Casting Tipe Data:

- (float)a: Mengubah nilai variabel a yang bertipe int menjadi float.
- (double)b: Mengubah nilai variabel b yang bertipe int menjadi double.
- (int)d: Mengubah nilai variabel d yang bertipe float menjadi int.
- (double)e: Mengubah nilai variabel e yang bertipe float menjadi double.
- (int)g: Mengubah nilai variabel g yang bertipe char (ASCII) menjadi int.
- (float)g: Mengubah nilai variabel g yang bertipe char (ASCII) menjadi float.
- (double)g: Mengubah nilai variabel g yang bertipe char (ASCII) menjadi double.
- (int)k: Mengubah nilai variabel k yang bertipe double menjadi int.
- (float)k: Mengubah nilai variabel k yang bertipe double menjadi float.

Menampilkan Output:

- Program mencetak hasil casting dari setiap operasi ke konsol menggunakan `System.out.println()`.

Hasil casting dapat memberikan output yang berbeda tergantung pada nilai yang di-cast dan tipe data yang dituju. Dalam kasus ini, program mencoba untuk melakukan casting antara beberapa tipe data primitif yang umum digunakan dalam Java. Casting dilakukan untuk mengubah tipe data dari satu tipe ke tipe lain yang kompatibel.

7. Casting2.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut Casting2.

Metode Main:

- Metode main adalah titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `int a = 8, b = 9;`: Mendeklarasikan dua variabel bertipe integer, a dengan nilai 8 dan b dengan nilai 9.
- `float d = 2.f, e = 3.2f;`: Mendeklarasikan dua variabel bertipe float, d dengan nilai 2.0 dan e dengan nilai 3.2.
- `char g = '5';`: Mendeklarasikan variabel bertipe char g dengan nilai '5'.
- `double k = 3.14;`: Mendeklarasikan variabel bertipe double k dengan nilai 3.14.
- `String n = "67", m = "45", l = "100";`: Mendeklarasikan tiga variabel bertipe String, n dengan nilai "67", m dengan nilai "45", dan l dengan nilai "100".

Penggunaan Metode Parsing dan Konversi:

- `Integer.parseInt(n)`: Mengonversi nilai dari variabel String n menjadi integer dan menyimpannya kembali ke variabel a.
- `Double.parseDouble(m)`: Mengonversi nilai dari variabel String m menjadi double dan menyimpannya kembali ke variabel k.
- `Float.parseFloat(l)`: Mengonversi nilai dari variabel String l menjadi float dan menyimpannya kembali ke variabel d.
- `String.valueOf(b)`: Mengonversi nilai dari variabel integer b menjadi String dan menyimpannya kembali ke variabel n.
- `String.valueOf(g)`: Mengonversi nilai dari variabel char g menjadi String dan menyimpannya kembali ke variabel m.
- `String.valueOf(e)`: Mengonversi nilai dari variabel float e menjadi String dan menyimpannya kembali ke variabel l.
- `Double.valueOf(a).intValue()`: Mengonversi nilai dari variabel integer a menjadi double dan kemudian menjadi integer kembali, dan menyimpannya ke variabel k.
- `Integer.valueOf(b).doubleValue()`: Mengonversi nilai dari variabel integer b menjadi double dan menyimpannya ke variabel c.

Menampilkan Output:

- Program mencetak nilai dari variabel-variabel yang telah diubah atau dikonversi ke konsol menggunakan `System.out.println()`.

8. Ekspresi.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut Ekspresi.

Metode Main:

- Metode main adalah titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `int x = 1;` Mendeklarasikan variabel x dengan tipe data integer dan memberikannya nilai 1.
- `int y = 2;` Mendeklarasikan variabel y dengan tipe data integer dan memberikannya nilai 2.

Penggunaan Operator Kondisional (Ternary Operator):

`(x < y) ? x : y`: Ini adalah contoh penggunaan operator kondisional (ternary operator) dalam Java.

- Jika kondisi `(x < y)` bernilai benar, maka nilai dari ekspresi adalah x.
- Jika kondisi `(x < y)` bernilai salah, maka nilai dari ekspresi adalah y.

Ekspresi ini digunakan untuk membandingkan nilai x dan y, dan kemudian mencetak nilai yang lebih kecil di antara keduanya.

Menampilkan Output:

- `System.out.print("x = " + x + "\n");` Menampilkan nilai dari variabel x.
- `System.out.print("y = " + y + "\n");` Menampilkan nilai dari variabel y.
- `System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y));` Menampilkan hasil dari ekspresi menggunakan operator kondisional.

Operator kondisional (`? :`) ini memungkinkan untuk menulis ekspresi yang singkat untuk menentukan nilai berdasarkan kondisi tertentu. Dalam kasus ini, kita memilih nilai x jika x kurang dari y, dan jika tidak, kita memilih nilai y.

9. Ekspresi1.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut Ekspresi1.

Metode Main:

- Metode main adalah titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `int x = 1; int y = 2; float fx; float fy;` Mendeklarasikan beberapa variabel, x dan y sebagai integer, fx dan fy sebagai float.

Operasi Pembagian:

- `System.out.print("x/y (format integer) = "+ x/y);`: Melakukan pembagian dua integer (x dan y) tanpa casting, yang menghasilkan hasil integer. Hasil ini adalah hasil pembagian bulat, yang hanya menampilkan bagian integer dari hasil pembagian, tanpa angka desimal.
- `System.out.print("\nx/y (format float) = "+ x/y);`: Operasi pembagian yang sama seperti sebelumnya, tetapi hasilnya ditampilkan sebagai float. Namun, karena kedua operan adalah integer, hasilnya masih merupakan integer, bahkan ketika ditampilkan sebagai float.

Casting: Untuk menghindari pembagian integer, dilakukan casting dari integer ke float sebelum melakukan pembagian.

- `fx = x; fy = y;`: Menetapkan nilai variabel x dan y ke variabel fx dan fy yang bertipe float.
- `System.out.print ("\nx/y (format integer) = "+ fx/fy);`: Melakukan pembagian fx dan fy yang bertipe float, hasilnya adalah float.
- `System.out.print ("\nx/y (format float) = "+ fx/fy);`: Operasi pembagian yang sama seperti sebelumnya, tetapi hasilnya ditampilkan sebagai float.

Casting Langsung: Melakukan operasi pembagian dengan melakukan casting langsung ke float.

- `System.out.print("\nfloat(x)/float(y) (format integer)= "+ (float)x/(float)y);`: Melakukan pembagian x dan y yang sudah dicasting ke float, sehingga hasilnya adalah float.
- `System.out.print("\nfloat(x)/float(y) (format float) = "+ (float)x/(float)y);`: Operasi pembagian yang sama seperti sebelumnya, tetapi hasilnya ditampilkan sebagai float.

Pengujian Hasil Pembagian:

- Terdapat pengujian hasil pembagian dengan nilai x = 10 dan y = 3. Namun, karena kedua nilai tersebut adalah integer, hasil pembagiannya masih merupakan integer, bahkan ketika ditampilkan sebagai float.

Kode ini mendemonstrasikan bagaimana casting dapat digunakan untuk mengontrol tipe data yang dihasilkan dari operasi pembagian dalam Java, khususnya ketika kita ingin hasil pembagian berupa float.

10. PrintHello.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut PrintHello.

Metode Main:

- Terdapat metode main yang merupakan titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Menuliskan Teks ke Layar:

- `System.out.print("Hello");`: Menampilkan teks "Hello" di layar tanpa mengganti baris (tanpa karakter newline).
- `System.out.print("\nHello ");`: Menampilkan teks "Hello" di layar dengan mengganti baris (menggunakan karakter newline `\n`).
- `System.out.println("World");`: Menampilkan teks "World" di layar dengan mengganti baris. Metode `println` akan menambahkan karakter newline secara otomatis setelah menampilkan teks.
- `System.out.println("Welcome");`: Menampilkan teks "Welcome" di layar dengan mengganti baris.

Penjelasan:

- `System.out.print` digunakan untuk menampilkan teks atau nilai ke layar tanpa mengganti baris.
- `System.out.println` juga menampilkan teks atau nilai ke layar, tetapi menambahkan karakter newline (ganti baris) setelah teks atau nilai tersebut.
- Dalam contoh ini, teks "Hello" ditampilkan dua kali, sekali dengan `System.out.print` dan sekali dengan `System.out.println`.
- Teks "World" dan "Welcome" juga ditampilkan menggunakan `System.out.println`, sehingga setelah masing-masing teks ditampilkan, baris baru ditambahkan ke layar.

Kesimpulannya, kode tersebut mencetak beberapa pesan ke layar termasuk "Hello", "World", dan "Welcome" dengan menggunakan method `print` dan `println`.

11. Incr.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut `Incr`.

Metode Main:

- Terdapat metode main yang merupakan titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `int i, j;`: Mendeklarasikan dua variabel bertipe integer `i` dan `j`.

Penggunaan Operator ++:

- `i = 3;`: Memberikan nilai 3 ke variabel `i`.

- `j = i++;` Menggunakan operator postfix increment (`i++`), yang berarti nilai dari `i` akan diambil terlebih dahulu untuk variabel `j`, dan setelah itu nilai `i` akan ditambahkan satu. Oleh karena itu, nilai `j` akan menjadi 3, dan nilai `i` akan menjadi 4.
- `System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j);` Pada bagian ini, nilai `i` di-preincrement (`++i`), yang berarti nilai `i` akan ditambahkan satu terlebih dahulu sebelum digunakan dalam operasi cetak. Nilai `j` yang telah ditetapkan sebelumnya tidak berubah karena menggunakan operator post-increment (`i++`).

Output:

- Program mencetak nilai variabel `i` setelah pre-increment dan nilai variabel `j` ke konsol.

Hasilnya, nilai `i` yang pertama kali dicetak adalah 5 (karena sudah di-increment sebelum dicetak), sementara nilai `j` yang telah diset sebelumnya tetap 3 (karena menggunakan operator post-increment saat pengisian nilai ke `j`).

12. Oper1.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut `Oper1`.

Metode Main:

- Terdapat metode main yang merupakan titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `int n = 10;` Mendeklarasikan variabel `n` dengan nilai 10, yang dalam representasi binernya adalah 1010.
- `int x = 1;` Mendeklarasikan variabel `x` dengan nilai 1, yang dalam representasi binernya adalah 0001.
- `int y = 2;` Mendeklarasikan variabel `y` dengan nilai 2, yang dalam representasi binernya adalah 0010.

Operasi Bitwise:

- `System.out.println("n & 8 = " + (n & 8));` Melakukan operasi bitwise AND (`&`) antara nilai `n` dan 8. Operasi bitwise AND akan menghasilkan 1 hanya jika kedua bit yang bersesuaian adalah 1, sehingga hasilnya adalah 8 (1000).
- `System.out.println("x & ~8 = " + (x & ~8));` Melakukan operasi bitwise AND antara nilai `x` dan hasil dari operasi bitwise NOT (`~`) dari 8. Operasi NOT akan mengubah setiap bit 0 menjadi 1 dan sebaliknya. Sehingga, hasilnya adalah 0 (0000).

- `System.out.println("y << 2 = " + (y << 2));`: Melakukan operasi bitwise shift left (<<) pada nilai y sebanyak 2 bit. Operasi ini akan menggeser bit-bit ke kiri sebanyak 2 posisi, sehingga hasilnya adalah 8 (1000).
- `System.out.println("y >> 3 = " + (y >> 3));`: Melakukan operasi bitwise shift right (>>) pada nilai y sebanyak 3 bit. Operasi ini akan menggeser bit-bit ke kanan sebanyak 3 posisi, sehingga hasilnya adalah 0 (0000).

Output:

- Program mencetak hasil dari setiap operasi ke konsol. Output dari masing-masing operasi bitwise dijelaskan dalam komentar di setiap baris kode.

13. Oper2.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut Oper2.

Metode Main:

- Terdapat metode main yang merupakan titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `char i, j;`: Mendeklarasikan dua variabel bertipe char i dan j.

Pengisian Nilai Variabel:

- `i = 3;`: Memberikan nilai 3 ke variabel i.
- `j = 4;`: Memberikan nilai 4 ke variabel j.

Operasi Bitwise:

- `System.out.println("i & j = " + (i & j));`: Melakukan operasi bitwise AND (&) antara nilai i dan j. Operasi bitwise AND akan menghasilkan 1 hanya jika kedua bit yang bersesuaian adalah 1, sehingga hasilnya adalah 0 (00000000 dalam biner).
- `System.out.println("i | j = " + (i | j));`: Melakukan operasi bitwise OR (|) antara nilai i dan j. Operasi bitwise OR akan menghasilkan 1 jika salah satu atau kedua bit yang bersesuaian adalah 1, sehingga hasilnya adalah 7 (00000111 dalam biner).
- `System.out.println("i ^ j = " + (i ^ j));`: Melakukan operasi bitwise XOR (^) antara nilai i dan j. Operasi bitwise XOR akan menghasilkan 1 jika hanya satu dari dua bit yang bersesuaian adalah 1, sehingga hasilnya adalah 7 (00000111 dalam biner).
- `System.out.println("~i = " + ~i);`: Melakukan operasi bitwise NOT (~) pada nilai i. Operasi bitwise NOT akan mengubah setiap bit 0 menjadi 1 dan sebaliknya. Sehingga, hasilnya adalah -4 (11111100 dalam biner).

Pemangkatan:

- `System.out.println(Math.pow(i, j));`: Menggunakan metode `pow(a,b)` dari kelas `Math` untuk melakukan pemangkatan. Namun, perlu diperhatikan bahwa ini tidak berhubungan dengan operasi bitwise. Metode `pow(a,b)` digunakan untuk menghitung pangkat dari `a` ke `b`.

Output:

- Program mencetak hasil dari setiap operasi ke konsol. Output dari masing-masing operasi bitwise dijelaskan dalam komentar di setiap baris kode. Operasi bitwise pada variabel bertipe `char` akan melakukan operasi pada representasi biner dari karakter tersebut.

14. Oper3.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut `Oper3`.

Metode Main:

- Terdapat metode `main` yang merupakan titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Operasi Logika:

- `if (true && true) { System.out.println(true && true); }`: Melakukan operasi logika AND (`&&`) antara dua ekspresi yang bernilai benar (`true`) dan mencetak hasilnya. Operasi AND akan menghasilkan `true` hanya jika kedua operannya juga `true`.
- `if (true & true) { System.out.println(true & false); }`: Melakukan operasi logika AND (`&`) antara dua ekspresi yang bernilai benar (`true`) dan salah (`false`). Operasi AND bitwise akan menghasilkan `false` karena salah satu operannya adalah `false`.
- `if (true) { System.out.println(true); }`: Kondisi `if` sederhana dengan kondisi yang selalu bernilai benar (`true`), sehingga blok kode di dalamnya akan selalu dieksekusi dan mencetak `true`.
- `if (true || true) { System.out.println(true); }`: Melakukan operasi logika OR (`||`) antara dua ekspresi yang bernilai benar (`true`) dan mencetak hasilnya. Operasi OR akan menghasilkan `true` karena salah satu operannya adalah `true`.
- `if (true | false) { System.out.println(true | false); }`: Melakukan operasi logika OR (`|`) antara dua ekspresi yang bernilai benar (`true`) dan salah (`false`). Operasi OR bitwise akan menghasilkan `true` karena salah satu operannya adalah `true`.

Output:

- Program mencetak hasil dari setiap operasi logika ke konsol. Output ini adalah hasil dari evaluasi setiap ekspresi logika. Jika kondisi yang diberikan pada

pernyataan if adalah true, maka blok kode di dalamnya akan dieksekusi dan mencetak hasil sesuai dengan ekspresi logika yang dievaluasi.

15. Oper4.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut Oper4.

Metode Main:

- Terdapat metode main yang merupakan titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `int i = 0; int j = 0;` Mendeklarasikan dua variabel bertipe integer i dan j dan memberi nilai awal 0.
- `char c = 8; char d = 10;` Mendeklarasikan dua variabel bertipe char c dan d dan memberi nilai 8 dan 10.
- `int e = (((int)c > (int)d) ? c : d);` Menggunakan operator ternary untuk membandingkan nilai dari variabel c dan d. Jika nilai c lebih besar dari d, maka variabel e akan diisi dengan nilai c, jika tidak, maka akan diisi dengan nilai d.
- `int k = ((i > j) ? i : j);` Menggunakan operator ternary untuk membandingkan nilai dari variabel i dan j. Jika nilai i lebih besar dari j, maka variabel k akan diisi dengan nilai i, jika tidak, maka akan diisi dengan nilai j.

Output:

- `System.out.print("Nilai e = " + e);` Mencetak nilai dari variabel e.
- `System.out.print("\nNilai k = " + k);` Mencetak nilai dari variabel k.
- `i = 2; j = 3;` Memberikan nilai 2 ke variabel i dan nilai 3 ke variabel j.
- `k = ((i++ > j++) ? i : j);` Menggunakan operator ternary untuk membandingkan nilai dari variabel i dan j setelah operasi ++ (post-increment). Jika nilai i setelah diincrement lebih besar dari nilai j setelah diincrement, maka variabel k akan diisi dengan nilai i setelah diincrement, jika tidak, maka akan diisi dengan nilai j setelah diincrement.
- `System.out.print("\nNilai k = " + k);` Mencetak nilai dari variabel k setelah operasi tersebut.

16. Operator.java

Deklarasi Kelas:

- Kode dimulai dengan deklarasi kelas utama yang disebut Operator.

Metode Main:

- Terdapat metode main yang merupakan titik masuk utama program Java. Di dalam metode ini, program melakukan operasi yang didefinisikan di dalamnya.

Deklarasi Variabel:

- `boolean Bool1, Bool2, TF;`: Mendeklarasikan tiga variabel boolean Bool1, Bool2, dan TF.
- `int i, j, hsl;`: Mendeklarasikan tiga variabel bertipe integer i, j, dan hsl.
- `float x, y, res;`: Mendeklarasikan tiga variabel bertipe float x, y, dan res.

Operasi Logika:

- Dilakukan beberapa operasi logika dengan variabel boolean seperti AND (&&), OR (||), NOT (!), dan XOR (^).

Operasi Numerik:

- Dilakukan operasi numerik seperti penambahan (+), pengurangan (-), pembagian (/), perkalian (*), pembagian bulat (/), dan sisa modulo (%) menggunakan variabel bertipe integer.
- Operasi yang sama juga dilakukan menggunakan variabel bertipe float.

Operasi Relasional:

- Dilakukan operasi relasional seperti sama dengan (==), tidak sama dengan (!=), kurang dari (<), lebih dari (>), kurang dari atau sama dengan (<=), dan lebih dari atau sama dengan (>=) menggunakan variabel bertipe integer.
- Operasi yang sama juga dilakukan menggunakan variabel bertipe float.

Output:

- Tidak ada output yang ditampilkan ke layar dalam kode yang diberikan. Namun, Anda dapat menambahkan perintah `System.out.println()` untuk menampilkan hasil operasi atau nilai dari variabel-variabel tersebut ke layar. Misalnya, `System.out.println("Hasil penjumlahan i + j = " + hsl);` untuk menampilkan hasil penjumlahan variabel i dan j.