

LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Ummul Ulya Handayani Hamid

Nim : 13020220252

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS MUSLIM INDONESIA

MAKASSAR

2025

Penjelasan dari source codenya beserta outputnya

1. Asgdll.java

1. Deklarasi Kelas:

- `public class Asgdll { ... }`

Mendefinisikan kelas dengan nama Asgdll. Semua kode Java harus berada di dalam sebuah kelas.

2. Metode main:

- `public static void main(String[] args) { ... }`

Ini adalah metode utama yang akan dieksekusi pertama kali saat program dijalankan.

3. Deklarasi Variabel:

- `float f = 20.0f;`

Mendeklarasikan variabel f dengan tipe data float dan menginisialisasinya dengan nilai 20.0f. Sufiks f menunjukkan bahwa nilai tersebut adalah float.

- `double fl1;`

Mendeklarasikan variabel fl1 dengan tipe data double. double memiliki presisi lebih tinggi dibandingkan float.

4. Inisialisasi Variabel:

- `fl1 = 10.0f;`

Menginisialisasi variabel fl1 dengan nilai 10.0f. Meskipun fl1 adalah double, nilai float tetap bisa diberikan. Java akan melakukan konversi implisit dari float ke double.

5. Output:

- `System.out.println("f: "+f+"\nfl1: "+fl1);`

Mencetak nilai variabel f dan fl1 ke konsol.

- `"f: "+f` menggabungkan string `"f: "` dengan nilai variabel f.
- `"\nfl1: "+fl1` menambahkan baris baru (`\n`) dan menggabungkan string `"fl1: "` dengan nilai variabel fl1.

Menghasilkan output sebagai berikut:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Asgdll.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Asgdll
f : 20.0
fl1: 10.0
```

2. Asign.java

1. Deklarasi Kelas:

```
public class Asign { ... }
```

Kode ini mendefinisikan kelas bernama Asign. Semua kode Java harus berada di dalam sebuah kelas.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

3. Deklarasi Variabel:

```
int i;
```

Mendeklarasikan variabel `i` dengan tipe data `int` (integer). Variabel ini akan digunakan untuk menyimpan nilai bilangan bulat.

4. Program:

- Mencetak Pesan:

```
System.out.print("hello\n");
```

Mencetak string "hello" ke konsol diikuti dengan karakter newline (`\n`), yang memindahkan kursor ke baris berikutnya. Namun, karena menggunakan `print`, tidak ada baris baru setelah "hello" sampai perintah berikutnya.

- Menginisialisasi Variabel:

```
i = 5;
```

Menginisialisasi variabel `i` dengan nilai 5.

- Mencetak Nilai Variabel:

```
System.out.println("Ini nilai i : " + i);
```

Mencetak string "Ini nilai i :" diikuti dengan nilai dari variabel `i`. Karena menggunakan `println`, output akan otomatis berpindah ke baris baru setelah mencetak.

Menghasilkan output:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Assign.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Assign
hello
Ini nilai i :5
```

3. ASIGNi.java

1. Deklarasi Kelas:

```
public class ASIGNi { ... }
```

Kode ini mendefinisikan kelas bernama `ASIGNi`. Semua kode Java harus berada di dalam sebuah kelas.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

3. Deklarasi Variabel:

```
short ks = 1;
```

Mendeklarasikan variabel ks dengan tipe data short dan menginisialisasinya dengan nilai 1.

- Tipe Data Integer:

```
int ki = 1;
```

Mendeklarasikan variabel ki dengan tipe data int dan menginisialisasinya dengan nilai 1.

- Tipe Data Panjang:

```
long kl = 10000;
```

Mendeklarasikan variabel kl dengan tipe data long dan menginisialisasinya dengan nilai 10000.

- Tipe Data Karakter:

```
char c = 65; // Inisialisasi karakter dengan integer
```

```
char c1 = 'Z'; // Inisialisasi karakter dengan karakter
```

Variabel c diinisialisasi dengan nilai integer 65, yang sesuai dengan karakter 'A' dalam ASCII.

Variabel c1 diinisialisasi langsung dengan karakter 'Z'.

- Tipe Data Real:

```
double x = 50.2f;
```

```
float y = 50.2f;
```

Variabel x diinisialisasi sebagai double dengan nilai 50.2 (meskipun menggunakan sufiks f, ini akan dikonversi menjadi double). Variabel y diinisialisasi sebagai float dengan nilai 50.2.

4. Algoritma dan Output:

Kode ini mencetak berbagai informasi ke konsol:

- Menampilkan karakter yang disimpan dalam variabel c dan c1.
- Menampilkan nilai dari variabel integer (short, int, dan long).
- Menampilkan nilai dari variabel real (double dan float).

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac ASIGNi.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> 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. BacaData.java

1. Import Kelas Scanner:

```
import java.util.Scanner;
```

Baris ini mengimpor kelas Scanner dari package java.util. Kelas Scanner digunakan untuk membaca input dari berbagai sumber, termasuk input dari keyboard.

2. Deklarasi Kelas:

```
public class BacaData { ... }
```

Kode ini mendefinisikan kelas bernama BacaData. Semua kode Java harus berada di dalam sebuah kelas.

3. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

4. Deklarasi Variabel:

```
int a;
```

```
Scanner masukan;
```

- `int a;`: Mendeklarasikan variabel `a` dengan tipe data `int` (integer). Variabel ini akan digunakan untuk menyimpan nilai integer yang diinput oleh pengguna.
- `Scanner masukan;`: Mendeklarasikan variabel `masukan` dengan tipe data `Scanner`. Variabel ini akan digunakan untuk membuat objek `Scanner` yang akan membaca input dari `System.in` (standard input, biasanya keyboard).

5. Program:

- Menampilkan Pesan:

```
System.out.print("Contoh membaca dan menulis, ketik nilai integer: \n");
```

Mencetak pesan ke konsol yang meminta pengguna untuk memasukkan nilai integer.

- Membuat Objek Scanner:

```
masukan = new Scanner(System.in);
```

Membuat objek `Scanner` baru yang membaca input dari `System.in`. Objek ini ditugaskan ke variabel `masukan`.

- Membaca Input Integer:

```
a = masukan.nextInt();
```

Menggunakan metode `nextInt()` dari objek `masukan` untuk membaca nilai integer dari input pengguna. Nilai ini disimpan dalam variabel `a`.

- Menampilkan Nilai yang Dibaca:

```
System.out.print("Nilai yang dibaca : " + a);
```

Mencetak pesan ke konsol yang menampilkan nilai integer yang telah dibaca dari input pengguna.

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac BacaData.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java BacaData
Contoh membaca dan menulis, ketik nilai integer:
8
Nilai yang dibaca : 8
```

5. Bacakar.java

1. Import Kelas yang Diperlukan:

```
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;
```

Baris ini mengimpor kelas-kelas yang diperlukan untuk membaca input dari pengguna:

- `BufferedReader`: Digunakan untuk membaca teks dari input stream dengan efisiensi tinggi.
- `IOException`: Diperlukan untuk menangani kemungkinan kesalahan saat membaca input.
- `InputStreamReader`: Mengkonversi byte stream menjadi karakter stream.

2. Deklarasi Kelas:

```
public class Bacakar { ... }
```

Kode ini mendefinisikan kelas bernama `Bacakar`.

3. Metode main:

```
public static void main(String[] args) throws IOException { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan. Metode ini juga menyatakan bahwa ia dapat melempar `IOException`, yang berarti jika terjadi kesalahan saat membaca input, program akan menghentikan eksekusi.

4. Deklarasi Variabel:

```
char cc;
```

```
int bil;
```

- `char cc`;; Mendeklarasikan variabel `cc` dengan tipe data karakter (`char`) untuk menyimpan karakter yang dibaca dari input.
- `int bil`;; Mendeklarasikan variabel `bil` dengan tipe data integer (`int`) untuk menyimpan bilangan bulat yang dibaca dari input.

5. Membuat Objek untuk Membaca Input:

```
InputStreamReader isr = new InputStreamReader(System.in);
```

```
BufferedReader dataIn = new BufferedReader(isr);
```

```
BufferedReader dataIn = new BufferedReader(new InputStreamReader(System.in));
```

- Membuat objek `InputStreamReader` yang akan membaca dari `System.in` (input standar, biasanya keyboard).
- Membuat objek `BufferedReader` (`dataIn` dan `dataIn`) untuk membaca input dengan efisiensi tinggi.

6. Algoritma dan Output:

- Menampilkan Pesan:

```
System.out.print("hello\n");
```

```
System.out.print("baca 1 karakter : ");
```

Mencetak pesan "hello" dan meminta pengguna untuk memasukkan satu karakter.

- Membaca Karakter:

```
cc = dataIn.readLine().charAt(0);
```

Menggunakan metode `readLine()` untuk membaca seluruh baris input sebagai string, kemudian mengambil karakter pertama dengan `charAt(0)` dan menyimpannya dalam variabel `cc`.

- Menampilkan Pesan untuk Bilangan:

```
System.out.print("baca 1 bilangan : ");
```

Mencetak pesan meminta pengguna untuk memasukkan satu bilangan bulat.

- Membaca Bilangan Bulat:

```
bil = Integer.parseInt(dataIn.readLine());
```

Menggunakan metode `readLine()` untuk membaca seluruh baris input sebagai string, kemudian mengonversi string tersebut menjadi integer menggunakan `Integer.parseInt()` dan menyimpannya dalam variabel `bil`.

7. Menampilkan Hasil:

```
System.out.print(cc + "\n" + bil + "\n");
```

```
System.out.print("bye \n");
```

Mencetak nilai karakter yang dibaca (`cc`), nilai bilangan bulat (`bil`), dan pesan "bye".

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Bacakar.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Bacakar
hello
baca 1 karakter : u
baca 1 bilangan : 1
u
1
bye
```

6. Casting1.java

1. Deklarasi Kelas:

```
public class Casting1 { ... }
```

Kode ini mendefinisikan kelas bernama `Casting1`.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

3. Deklarasi dan Inisialisasi Variabel:

```
int a = 5, b = 6;
```

```
float d = 2.f, e = 3.2f;
```

```
char g = '5';
```

```
double k = 3.14;
```

- `int a = 5;` dan `int b = 6;;` Mendeklarasikan dua variabel integer.
- `float d = 2.f;` dan `float e = 3.2f;;` Mendeklarasikan dua variabel float.
- `char g = '5';` Mendeklarasikan variabel karakter yang diinisialisasi dengan karakter '5'.
- `double k = 3.14;;` Mendeklarasikan variabel double.

4. Casting dan Output:

Kode ini mencetak hasil dari berbagai operasi casting ke konsol:

```
-System.out.println((float)a); // int ke float
```

Mengonversi integer a (5) menjadi float (5.0).

```
-System.out.println((double)b); // int ke double
```

Mengonversi integer b (6) menjadi double (6.0).

```
-System.out.println((int)d); // float ke int
```

Mengonversi float d (2.0) menjadi integer (2). Nilai desimal dibuang.

```
-System.out.println((double)e); // float ke double
```

Mengonversi float e (3.2) menjadi double (3.2).

```
-System.out.println((int)g); // char ke int (ASCII)
```

Mengonversi karakter g ('5') menjadi integer berdasarkan nilai ASCII-nya, yaitu 53.

```
-System.out.println((float)g); // char ke float (ASCII)
```

Mengonversi karakter g ('5') menjadi float berdasarkan nilai ASCII-nya, yaitu 53.0.

```
-System.out.println((double)g); // char ke double (ASCII)
```

Mengonversi karakter g ('5') menjadi double berdasarkan nilai ASCII-nya, yaitu 53.0.

```
-System.out.println((int)k); // double ke int
```

Mengonversi double k (3.14) menjadi integer (3). Nilai desimal dibuang.

```
-System.out.println((float)k); // double ke float
```

Mengonversi double k (3.14) menjadi float (3.14).

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Casting1.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Casting1
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14
```

7. Casting2.java

1. Deklarasi Kelas:

```
public class Casting2 { ... }
```

Kode ini mendefinisikan kelas bernama Casting2.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

3. Deklarasi dan Inisialisasi Variabel:

```
int a = 8, b = 9;
```

```
float d = 2.f, e = 3.2f;
```

```
char g = '5';
```

```
double k = 3.14;
```

```
String n = "67", m = "45", l = "100";
```

- `int a = 8, b = 9;;` Mendeklarasikan dua variabel integer.
- `float d = 2.f, e = 3.2f;;` Mendeklarasikan dua variabel float.
- `char g = '5';` Mendeklarasikan variabel karakter yang diinisialisasi dengan karakter '5'.
- `double k = 3.14;;` Mendeklarasikan variabel double.
- `String n = "67", m = "45", l = "100";` Mendeklarasikan tiga variabel string yang berisi nilai numerik.

4. Konversi String ke Numerik:

```
a = Integer.parseInt(n); /*Konversi String ke Integer*/
```

```
k = Double.parseDouble(m); /*Konversi String ke Double*/
```

```
d = Float.parseFloat(l); /*Konversi String ke Float*/
```

```
System.out.println("a : " + a + "\nk : " + k + "\nd : " + d);
```

- `a = Integer.parseInt(n);`: Mengonversi string `n` ("67") menjadi integer (67).
- `k = Double.parseDouble(m);`: Mengonversi string `m` ("45") menjadi double (45.0).
- `d = Float.parseFloat(l);`: Mengonversi string `l` ("100") menjadi float (100.0).
- Kemudian, nilai-nilai ini dicetak ke konsol.

5. Konversi Numerik ke String:

```
n = String.valueOf(b); /*Konversi Integer ke String*/
m = String.valueOf(g); /*Konversi Karakter ke String*/
l = String.valueOf(e); /*Konversi Float ke String*/
System.out.println("n : " + n + "\nm : " + m + "\nl : " + l);
```

- `n = String.valueOf(b);`: Mengonversi integer `b` (9) menjadi string ("9").
- `m = String.valueOf(g);`: Mengonversi karakter `g` ('5') menjadi string ("5").
- `l = String.valueOf(e);`: Mengonversi float `e` (3.2) menjadi string ("3.2").
- Kemudian, nilai-nilai ini dicetak ke konsol.

6. Konversi Integer ke Double (dengan cara lain):

```
k = Double.valueOf(a).intValue(); /*Konversi Integer ke Double*/
double c = Integer.valueOf(b).doubleValue();
System.out.println("k : " + k + "\nc : " + c + "\nl : " + l);
```

- `k = Double.valueOf(a).intValue();`: Mengonversi integer `a` (67) menjadi objek Double, lalu diambil nilai integer-nya, sehingga nilai desimalnya hilang (67.0 menjadi 67).
- `double c = Integer.valueOf(b).doubleValue();`: Mengonversi integer `b` (9) menjadi objek Integer, lalu diambil nilai double-nya (9 menjadi 9.0).
- Kemudian, nilai-nilai ini dicetak ke konsol.

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Casting2.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> 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. Ekspresi.java

1. Deklarasi Kelas:

```
public class Ekspresi { ... }
```

Kode ini mendefinisikan kelas bernama Ekspresi.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

3. Deklarasi dan Inisialisasi Variabel:

```
int x = 1;
```

```
int y = 2;
```

Mendeklarasikan dan menginisialisasi dua variabel integer x dan y.

4. Output Nilai Variabel:

```
System.out.print("x = " + x + "\n");
```

```
System.out.print("y = " + y + "\n");
```

Mencetak nilai variabel x dan y ke konsol.

5. Ekspresi Kondisional (Ternary Operator):

```
System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y));
```

Ini adalah bagian inti dari kode. Mari kita pecah:

- $(x < y) ? x : y$: Ini adalah ekspresi kondisional (ternary operator).
 - $(x < y)$: Ini adalah kondisi yang dievaluasi. Jika x kurang dari y, kondisi ini bernilai true. Jika tidak, bernilai false.
 - $? x : y$: Bagian ini menentukan nilai yang akan dihasilkan berdasarkan kondisi.
 - Jika kondisi $(x < y)$ bernilai true, ekspresi akan menghasilkan nilai x.
 - Jika kondisi $(x < y)$ bernilai false, ekspresi akan menghasilkan nilai y.
- Dalam kasus ini, karena x adalah 1 dan y adalah 2, maka $(x < y)$ bernilai true, sehingga ekspresi menghasilkan nilai x, yaitu 1.
- Tanda kurung $((x < y) ? x : y)$ digunakan untuk memastikan bahwa ekspresi kondisional dievaluasi terlebih dahulu sebelum digabungkan dengan string.

Outputnya yaitu:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Ekspresi.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
```

9. Ekspresi1.java

1. Deklarasi Kelas:

```
public class Ekspresi1 { ... }
```

Kode ini mendefinisikan kelas bernama Ekspresi1.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

3. Deklarasi dan Inisialisasi Variabel:

```
int x = 1;
```

```
int y = 2;
```

```
float fx;
```

```
float fy;
```

- `int x = 1;` dan `int y = 2;;` Mendeklarasikan dan menginisialisasi dua variabel integer.
- `float fx;` dan `float fy;;` Mendeklarasikan dua variabel float tanpa inisialisasi.

4. Operasi Pembagian:

```
System.out.print("x/y (format integer) = " + x/y);
```

```
System.out.print("\nx/y (format float) = " + x/y);
```

- `x/y (format integer)`: Melakukan pembagian antara dua integer (x dan y). Karena keduanya adalah integer, hasilnya juga akan menjadi integer. Dalam hal ini, hasilnya adalah 0 karena pembagian 1 dibagi 2 menghasilkan 0 dalam pembagian integer.
- `x/y (format float)`: Melakukan pembagian antara dua integer. Hasilnya tetap 0 karena kedua operand masih bertipe integer.

5. Menghindari Hasil Nol dengan Casting:

```
fx = x;
```

```
fy = y;
```

```
System.out.print("\nx/y (format integer) = " + fx/fy);
```

```
System.out.print("\nx/y (format float) = " + fx/fy);
```

- Di sini, `fx` dan `fy` diinisialisasi dengan nilai dari `x` dan `y`. Dengan cara ini, `fx` dan `fy` sekarang bertipe float.
- `fx/fy`: Melakukan pembagian antara dua float. Hasilnya adalah 0.5, yang dicetak ke konsol.

6. Casting Langsung:

```
System.out.print("\nfloat(x)/float(y) (format integer) = " + (float)x/(float)y);
```

```
System.out.print("\nfloat(x)/float(y) (format float) = " + (float)x/(float)y);
```

- `(float)x/(float)y`: Melakukan casting pada `x` dan `y` menjadi float sebelum melakukan pembagian. Hasilnya adalah 0.5 untuk format float.

7. Mengubah Nilai Variabel:

```
x = 10;
y = 3;
System.out.print("\nx/y (format integer) = " + x/y);
System.out.print("\nx/y (format float) = " + x/y);
```

- Mengubah nilai x menjadi 10 dan y menjadi 3.
- Pembagian antara x dan y dalam format integer akan menghasilkan 3 (10 dibagi 3).
- Pembagian dalam format float juga akan menghasilkan 3.3333... tetapi karena keduanya masih bertipe integer, output akan tetap dalam format integer.

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Ekspresil.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Ekspresil
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. Hello.java

1. Deklarasi Kelas:

```
public class Hello { ... }
```

Kode ini mendefinisikan kelas bernama Hello. Dalam Java, semua kode harus berada di dalam sebuah kelas.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan. Metode ini adalah titik masuk untuk aplikasi Java.

3. Menuliskan Pesan ke Layar:

- Menggunakan print:

```
System.out.print("Hello");
```

Baris ini mencetak string "Hello" ke konsol. Metode print tidak menambahkan karakter baru setelah mencetak, sehingga kursor tetap berada di akhir output.

- Menggunakan print dengan Newline:

```
System.out.print("\nHello ");
```

Baris ini mencetak newline (\n) terlebih dahulu, lalu mencetak "Hello ". Dengan cara ini, output berikutnya akan dimulai pada baris baru.

- Menggunakan println:

```
System.out.println("World");
```

Baris ini mencetak "World" ke konsol dan secara otomatis menambahkan karakter newline setelahnya. Ini berarti kursor akan berpindah ke baris berikutnya setelah mencetak.

- Mencetak Pesan Selanjutnya:

```
System.out.println("Welcome");
```

Baris ini mencetak "Welcome" ke konsol, juga dengan menambahkan newline di akhir.

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Hello.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Hello
Hello
Hello World
Welcome
```

11. Incr.java

1. Deklarasi Kelas:

```
public class Incr { ... }
```

Kode ini mendefinisikan kelas bernama Incr.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

3. Deklarasi Variabel:

```
int i, j;
```

Mendeklarasikan dua variabel integer i dan j.

4. Inisialisasi dan Penggunaan Operator Increment:

```
i = 3;
```

```
j = i++;
```

- `i = 3;`: Menginisialisasi variabel i dengan nilai 3.
- `j = i++;`: Ini adalah contoh penggunaan post-increment. Dalam hal ini:
 - Nilai dari i (yang saat ini adalah 3) akan diberikan ke j, sehingga j menjadi 3.
 - Setelah itu, nilai i akan diincrement (ditambah 1), sehingga i menjadi 4.

5. Mencetak Hasil:

```
System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j);
```

- ++i: Ini adalah contoh penggunaan pre-increment. Nilai i (yang sekarang adalah 4) akan diincrement terlebih dahulu menjadi 5, dan kemudian nilai baru (5) akan digunakan dalam output.
- Mencetak nilai dari i dan j ke konsol.

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Incr.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Incr
Nilai i : 5
Nilai j : 3
```

12. Oper1.java

1. Deklarasi Kelas:

```
public class Oper1 { ... }
```

Kode ini mendefinisikan kelas bernama Oper1.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

3. Deklarasi dan Inisialisasi Variabel:

```
int n = 10; /* 1010 */
```

```
int x = 1; /* 0001 */
```

```
int y = 2; /* 0010 */
```

- `int n = 10;`: Menginisialisasi variabel n dengan nilai 10, yang dalam biner adalah 1010.
- `int x = 1;`: Menginisialisasi variabel x dengan nilai 1, yang dalam biner adalah 0001.
- `int y = 2;`: Menginisialisasi variabel y dengan nilai 2, yang dalam biner adalah 0010.

4. Mencetak Nilai Variabel:

```
System.out.println("n = " + n);
```

```
System.out.println("x = " + x);
```

```
System.out.println("y = " + y);
```

Mencetak nilai dari variabel n, x, dan y ke konsol.

5. Penggunaan Operator Bitwise:

- AND Bitwise (&):

```
System.out.println("n & 8 = " + (n & 8)); /* 1010 AND 1000 */
```

- Operasi ini melakukan AND bitwise antara n (1010) dan 8 (1000). Hasilnya adalah:

1010

AND

1000

1000 (8 dalam desimal)

- NOT Bitwise (~) dan AND Bitwise (&):

```
System.out.println("x & ~8 = " + (x & ~8)); /* 1 AND 0111 */
```

- Di sini, ~8 mengubah biner dari 8 (1000) menjadi NOT-nya, yaitu:

~1000 = 0111 (7 dalam desimal)

- Kemudian, operasi AND dilakukan antara x (0001) dan hasil NOT:

0001

AND

0111

0001 (1 dalam desimal)

- Pergeseran Kiri (<<):

```
System.out.println("y << 2 = " + (y << 2)); /* 10 ==> 1000 = 8 */
```

- Operasi ini menggeser bit dari y (0010) ke kiri sebanyak dua posisi:

0010 << 2 = 1000 (8 dalam desimal)

- Pergeseran Kanan (>>):

```
System.out.println("y >> 3 = " + (y >> 3)); /* 10 ==> 0000 = 0 */
```

- Operasi ini menggeser bit dari y (0010) ke kanan sebanyak tiga posisi:

0010 >> 3 = 0000 (0 dalam desimal)

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Oper1.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Oper1
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

13. Oper2.java

1. Deklarasi Kelas:


```
public class Oper2 { ... }
```

Kode ini mendefinisikan kelas bernama Oper2.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

3. Deklarasi dan Inisialisasi Variabel:

```
char i, j;
```

```
i = 3; /* 00000011 dalam biner */
```

```
j = 4; /* 00000100 dalam biner */
```

- `char i = 3;` Menginisialisasi variabel karakter `i` dengan nilai 3, yang dalam biner adalah 00000011.
- `char j = 4;` Menginisialisasi variabel karakter `j` dengan nilai 4, yang dalam biner adalah 00000100.

4. Mencetak Nilai Variabel:

```
System.out.println("i = " + (int) i);
```

```
System.out.println("j = " + j);
```

- Mencetak nilai dari `i` setelah mengonversinya menjadi integer (hasilnya adalah 3).
- Mencetak nilai dari `j` (hasilnya adalah 4).

5. Penggunaan Operator Bitwise:

- AND Bitwise (`&`):

```
System.out.println("i & j = " + (i & j)); /* 0: 00000000 dalam biner */
```

- Operasi ini melakukan AND bitwise antara `i` (00000011) dan `j` (00000100):

```
00000011
```

```
AND
```

```
00000100
```

```
-----
```

```
00000000 (0 dalam desimal)
```

- OR Bitwise (`|`):

```
System.out.println("i | j = " + (i | j)); /* 7: 00000111 biner */
```

- Operasi ini melakukan OR bitwise antara `i` dan `j`:

```
00000011
```

```
OR
```

```
00000100
```

00000111 (7 dalam desimal)

- XOR Bitwise (^):

```
System.out.println("i ^ j = " + (i ^ j)); /* 7: 00000111 biner */
```

- Operasi ini melakukan XOR bitwise antara i dan j. Hasilnya sama dengan OR karena kedua operand memiliki bit yang berbeda:

00000011

XOR

00000100

00000111 (7 dalam desimal)

- Catatan: Operator ^ di Java bukan untuk pangkat, melainkan untuk operasi XOR.

6. Pemangkatan menggunakan Kelas Math:

```
System.out.println(Math.pow(i, j)); /* Class Math memiliki method pow(a,b) untuk pemangkatan */
```

- Menggunakan metode pow(a, b) dari kelas Math untuk menghitung 3⁴ (3 pangkat 4), hasilnya adalah 81.081.0.

7. NOT Bitwise (~):

```
System.out.println(" ~i = " + ~i); /* -4: 11111100 biner */
```

- Operasi ini melakukan NOT bitwise pada nilai i. Hasilnya adalah kebalikan dari semua bit:

~00000011 = 11111100 (-4 dalam desimal menggunakan representasi dua's complement)

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Oper2.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Oper2
i = 3
j =
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
```

14. Oper3.java

1. Deklarasi Kelas:

```
public class Oper3 { ... }
```

Kode ini mendefinisikan kelas bernama Oper3.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

3. Penggunaan Operator Logika:

- Operator AND Logika (&&):

```
if (true && true) { System.out.println(true && true); }
```

- Pernyataan ini menggunakan operator AND logika (&&). Karena kedua operand adalah true, hasilnya juga true. Baris ini mencetak true ke konsol.

- Operator AND Bitwise (&):

```
if (true & true) { System.out.println(true & false); }
```

- Pernyataan ini menggunakan operator AND bitwise (&). Hasil dari true & true adalah true, sehingga blok kode dieksekusi.
- Namun, di dalam blok tersebut, System.out.println(true & false); mencetak hasil dari true & false, yang adalah false.

- Pernyataan Sederhana:

```
if (true) { System.out.println(true); }
```

- Pernyataan ini selalu benar, sehingga mencetak true ke konsol.

- Operator OR Logika (||):

```
if (true || true) { System.out.println(true); }
```

- Pernyataan ini menggunakan operator OR logika (||). Karena salah satu operand adalah true, hasilnya adalah true, sehingga mencetak true.

- Operator OR Bitwise (|):

```
if (true | false) { System.out.println(true | false); }
```

- Pernyataan ini menggunakan operator OR bitwise (|). Hasil dari true | false adalah true, sehingga blok kode dieksekusi.
- Di dalam blok tersebut, System.out.println(true | false); mencetak hasil dari true | false, yang juga adalah true.

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Oper3.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Oper3
true
false
true
true
true
```

15. Oper4.java

1. Deklarasi Kelas:

```
public class Oper4 { ... }
```

Kode ini mendefinisikan kelas bernama Oper4.

2. Metode main:

```
public static void main(String[] args) { ... }
```

Ini adalah metode utama yang dieksekusi pertama kali saat program dijalankan.

3. Deklarasi dan Inisialisasi Variabel:

```
int i = 0;
```

```
int j = 0;
```

```
char c = 8;
```

```
char d = 10;
```

- `int i = 0;` dan `int j = 0;`: Mendeklarasikan dua variabel integer dan menginisialisasinya dengan 0.
- `char c = 8;` dan `char d = 10;`: Mendeklarasikan dua variabel karakter dan menginisialisasinya dengan nilai 8 dan 10.

4. Penggunaan Ekspresi Kondisional:

- Ekspresi Kondisional dengan Tipe Data char:

```
int e = (((int)c > (int)d) ? c : d);
```

- `(int)c > (int)d`: Membandingkan nilai c dan d setelah keduanya di-casting menjadi integer. Karena 8 tidak lebih besar dari 10, kondisi ini bernilai false.
- `? c : d`: Jika kondisi bernilai true, maka nilai c akan ditugaskan ke e; jika tidak, nilai d yang akan ditugaskan. Dalam kasus ini, d (10) ditugaskan ke e.
- Ekspresi Kondisional dengan Tipe Data int:

```
int k = ((i > j) ? i : j);
```

- `(i > j)`: Membandingkan nilai i (0) dan j (0). Karena i tidak lebih besar dari j, kondisi ini bernilai false.
- `? i : j`: Jika kondisi bernilai true, maka nilai i akan ditugaskan ke k; jika tidak, nilai j yang akan ditugaskan. Dalam kasus ini, j (0) ditugaskan ke k.

5. Output Nilai Variabel:

```
System.out.print ("Nilai e = "+ e);
```

```
System.out.print ("\nNilai k = "+ k);
```

Mencetak nilai e (10) dan k (0) ke konsol.

6. Penggunaan Ekspresi Kondisional dengan Operator Increment:

```
i = 2;
```

```
j = 3;
```

```
k = ((i++>j++) ? i : j) ;
```

```
System.out.print ("\nNilai k = "+ k);
```

- `i = 2`; dan `j = 3`; Mengubah nilai `i` menjadi 2 dan `j` menjadi 3.
- `(i++ > j++)`: Ini adalah bagian yang penting. Operator `i++` dan `j++` adalah operator post-increment:
 - `i++ > j++`: Membandingkan nilai `i` (2) dengan `j` (3) sebelum melakukan increment. Karena 2 tidak lebih besar dari 3, kondisi ini bernilai false.
 - Setelah perbandingan, nilai `i` menjadi 3 dan nilai `j` menjadi 4.
 - `? i : j`: Karena kondisi bernilai false, maka nilai `j` (yang sekarang adalah 4) ditugaskan ke `k`.

7. Output Nilai Variabel Setelah Increment:

```
System.out.print ("\nNilai k = "+ k);
```

Mencetak nilai `k` (4) ke konsol.

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Oper4.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4
```

16. Operator.java

1. Deklarasi Kelas:

- `public class Operator { ... }`

Ini mendeklarasikan sebuah kelas bernama `Operator`. Dalam Java, semua kode harus berada di dalam kelas. Kata kunci `public` berarti kelas ini dapat diakses dari kelas lain.

2. Metode main:

- `public static void main(String[] args) { ... }`

Ini adalah metode utama (entry point) dari program Java. Ketika program dijalankan, metode `main` adalah tempat eksekusi dimulai.

- `public`: Metode ini dapat diakses dari luar kelas.
- `static`: Metode ini terkait dengan kelas itu sendiri, bukan dengan instance (objek) dari kelas.
- `void`: Metode ini tidak mengembalikan nilai apa pun.
- `String[] args`: Ini adalah array dari argumen baris perintah yang dapat diberikan saat menjalankan program.

3. Deklarasi Variabel:

- `boolean Bool1, Bool2, TF;`

Mendeklarasikan tiga variabel boolean (Bool1, Bool2, dan TF). Variabel boolean dapat menyimpan nilai true atau false.

- `int i, j, hsl;`

Mendeklarasikan tiga variabel integer (i, j, dan hsl). Variabel integer dapat menyimpan bilangan bulat.

- `float x, y, res;`

Mendeklarasikan tiga variabel float (x, y, dan res). Variabel float dapat menyimpan bilangan desimal.

4. Inisialisasi Variabel:

- `Bool1 = true;`
- `Bool2 = false;`

Menginisialisasi Bool1 dengan true dan Bool2 dengan false.

- `i = 5;`
- `j = 2;`

Menginisialisasi i dengan 5 dan j dengan 2.

- `x = 5f;`
- `y = 5f;`

Menginisialisasi x dan y dengan 5.0. Sufiks f menunjukkan bahwa ini adalah nilai float.

Operasi dan Output

Kode ini melakukan berbagai operasi menggunakan operator-operator yang berbeda, dan setiap hasil operasi dicetak ke konsol menggunakan `System.out.println()`.

1. Operasi Logika (Boolean):

- `TF = Bool1 && Bool2; (AND)`

TF akan bernilai false karena Bool1 adalah true dan Bool2 adalah false.

- `TF = Bool1 || Bool2; (OR)`

TF akan bernilai true karena Bool1 adalah true.

- `TF = !Bool1; (NOT)`

TF akan bernilai false karena Bool1 adalah true.

- `TF = Bool1 ^ Bool2; (XOR)`

TF akan bernilai true karena Bool1 dan Bool2 memiliki nilai yang berbeda.

2. Operasi Aritmatika (Integer):

- `hsl = i + j; (Penambahan)`

hsl akan bernilai 7.

- `hsl = i - j; (Pengurangan)`

hsl akan bernilai 3.

- $hsl = i / j$; (Pembagian)

hsl akan bernilai 2 (pembagian integer).

- $hsl = i * j$; (Perkalian)

hsl akan bernilai 10.

- $hsl = i \% j$; (Modulo)

hsl akan bernilai 1 (sisa dari pembagian 5 dibagi 2).

3. Operasi Aritmatika (Float):

- $res = x + y$; (Penambahan)

res akan bernilai 10.0.

- $res = x - y$; (Pengurangan)

res akan bernilai 0.0.

- $res = x / y$; (Pembagian)

res akan bernilai 1.0.

- $res = x * y$; (Perkalian)

res akan bernilai 25.0.

4. Operasi Relasional (Integer):

- $TF = (i == j)$; (Sama dengan)

TF akan bernilai false karena 5 tidak sama dengan 2.

- $TF = (i != j)$; (Tidak sama dengan)

TF akan bernilai true karena 5 tidak sama dengan 2.

- $TF = (i < j)$; (Kurang dari)

TF akan bernilai false karena 5 tidak kurang dari 2.

- $TF = (i > j)$; (Lebih dari)

TF akan bernilai true karena 5 lebih dari 2.

- $TF = (i <= j)$; (Kurang dari atau sama dengan)

TF akan bernilai false karena 5 tidak kurang dari atau sama dengan 2.

- $TF = (i >= j)$; (Lebih dari atau sama dengan)

TF akan bernilai true karena 5 lebih dari atau sama dengan 2.

5. Operasi Relasional (Float):

- $TF = (x != y)$; (Tidak sama dengan)

TF akan bernilai false karena 5.0 sama dengan 5.0.

- $TF = (x < y)$; (Kurang dari)

TF akan bernilai false karena 5.0 tidak kurang dari 5.0.

- $TF = (x > y)$; (Lebih dari)

TF akan bernilai false karena 5.0 tidak lebih dari 5.0.

- $TF = (x \leq y)$; (Kurang dari atau sama dengan)

TF akan bernilai true karena 5.0 kurang dari atau sama dengan 5.0.

- $TF = (x \geq y)$; (Lebih dari atau sama dengan)

TF akan bernilai true karena 5.0 lebih dari atau sama dengan 5.0.

Output yang dihasilkan:

```
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> javac Operator.java
PS C:\Users\ummul\OneDrive\文件\TUGAS PBO\TUGAS 1\Source code> java Operator
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output
Bool1 AND Bool2: false
Bool1 OR Bool2: true
NOT Bool1: false
Bool1 XOR Bool2: true
```