

LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK



NURSYAMSI

B4

13020220273

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS MUSLIM INDONESIA

MAKASSAR

2025

1. Asgdll.java

- Deklarasi Kelas:
 - Kelas bernama `Asgdll` dideklarasikan sebagai kelas publik.
- Metode Main:
 - Metode `main` adalah titik awal eksekusi program. Ini adalah metode yang harus ada dalam setiap program Java.
- Kamus Variabel:
 - Variabel `f` dideklarasikan dengan tipe `float` dan diinisialisasi dengan nilai `20.0f`.
 - Variabel `f11` dideklarasikan dengan tipe `double` tetapi belum diinisialisasi.
- Algoritma:
 - Variabel `f11` diinisialisasi dengan nilai `10.0f`, yang merupakan kesalahan karena seharusnya menggunakan nilai bertipe `double` (seharusnya `10.0`).
- Output:
 - Program menggunakan `System.out.println` untuk mencetak nilai `f` dan `f11` ke konsol, dengan setiap nilai ditampilkan pada baris yang berbeda.

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac Asgdll.java
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>java Asgdll
f : 20.0
f11: 10.0
```

2. Assign.java

- Deklarasi Kelas:

Program dimulai dengan mendeklarasikan kelas bernama `Assign`. Kelas ini merupakan struktur dasar dalam pemrograman Java yang mengelompokkan kode.
- Metode `main`:

Metode `main` adalah titik masuk eksekusi program. Setiap program Java harus memiliki metode ini agar dapat dijalankan. Parameter `String[] args`

memungkinkan program menerima argumen dari baris perintah, meskipun dalam program ini tidak digunakan.

- **Variabel:**

Di bagian kamus, variabel `i` dideklarasikan dengan tipe data `int`, yang berarti dapat menyimpan bilangan bulat.

- **Pencetakan Output:**

- Program mencetak string "hello" ke konsol menggunakan `System.out.print`, diikuti dengan karakter newline (`\n`) untuk memindahkan kursor ke baris berikutnya.
- Variabel `i` kemudian diinisialisasi dengan nilai 5.
- Program selanjutnya mencetak pesan "Ini nilai i :" diikuti dengan nilai dari variabel `i`, yang hasilnya akan menunjukkan "Ini nilai i : 5".

- **Output:**

Saat program dijalankan, output yang dihasilkan adalah:

```
C:\Users\ASUS\Documents\Tugas PB0\Tugas 1\Source Code>javac Asign.java
C:\Users\ASUS\Documents\Tugas PB0\Tugas 1\Source Code>java Asign
hello
Ini nilai i :5
```

3. ASIGNi.java

- **Deklarasi Kelas:**

Program dimulai dengan mendeklarasikan kelas bernama `ASIGNi`, yang merupakan struktur dasar dalam Java untuk mengelompokkan kode.

- **Metode `main`:**

Metode `main` adalah titik masuk eksekusi program. Setiap program Java harus memiliki metode ini untuk dapat dijalankan. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.

- **Deklarasi Variabel:**

- `short ks = 1;` Deklarasi variabel `ks` dengan tipe data `short`, diinisialisasi dengan nilai 1.
- `int ki = 1;` Deklarasi variabel `ki` dengan tipe data `int`, diinisialisasi dengan nilai 1.

- `long k1 = 10000;` Deklarasi variabel `k1` dengan tipe data `long`, diinisialisasi dengan nilai `10000`.
 - `char c = 65;` Deklarasi variabel `c` dengan tipe data `char`, diinisialisasi menggunakan integer `65`, yang akan direpresentasikan sebagai karakter 'A' dalam ASCII.
 - `char c1 = 'Z';` Deklarasi variabel `c1` dengan tipe data `char`, diinisialisasi dengan karakter 'Z'.
 - `double x = 50.2f;` Deklarasi variabel `x` dengan tipe data `double`, diinisialisasi dengan nilai `50.2f` (dapat menggunakan `50.2` tanpa `f` karena `double` bisa menyimpan `float`).
 - `float y = 50.2f;` Deklarasi variabel `y` dengan tipe data `float`, diinisialisasi dengan nilai `50.2f`.
- Pencetakan Output:
 - Program mencetak nilai dari variabel karakter `c` dan `c1`, serta menampilkan hasilnya ke konsol.
 - Mencetak karakter sebagai integer dengan menggunakan variabel `c` dan `c1`.
 - Program mencetak nilai dari variabel `ks`, `ki`, dan `k1` untuk menunjukkan nilai integer yang berbeda.
 - Terakhir, program mencetak nilai dari variabel `x` dan `y`, yang menunjukkan bilangan riil.

- Output Program:

Ketika dijalankan, output program ini akan menampilkan:

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac ASIGNi.java

C:\Users\ASUS\Documents\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

- Import Kelas `Scanner`:
Program dimulai dengan mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.

- **Deklarasi Kelas:**

Kelas `BacaData` dideklarasikan sebagai kelas publik, yang merupakan struktur dasar untuk mengelompokkan kode.

- **Metode `main`:**

Metode `main` adalah titik masuk untuk eksekusi program. Setiap program Java memerlukan metode ini agar dapat dijalankan. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.

- **Deklarasi Variabel:**

- `int a;` Variabel `a` dideklarasikan dengan tipe data `int`, yang akan digunakan untuk menyimpan nilai integer yang dibaca dari input.
- `Scanner masukan;` Deklarasi objek `masukan` dari kelas `Scanner`, yang akan digunakan untuk mengambil input dari pengguna.

- **Pencetakan Pesan ke Konsol:**

Program mencetak pesan "Contoh membaca dan menulis, ketik nilai integer:" untuk meminta pengguna memasukkan nilai integer.

- **Inisialisasi Scanner:**

- `masukan = new Scanner(System.in);` Objek `masukan` diinisialisasi untuk membaca input dari sistem (konsol).

- **Membaca Input:**

- `a = masukan.nextInt();` Program menunggu pengguna untuk memasukkan nilai integer. Nilai yang dimasukkan akan disimpan dalam variabel `a`.
- Terdapat komentar yang menyarankan untuk mencoba `masukan.nextInt();` tanpa menyimpan hasilnya. Jika dilakukan, program akan membaca input tetapi tidak akan menyimpannya, sehingga tidak ada nilai yang dapat digunakan selanjutnya.

- **Mencetak Nilai yang Dibaca:**

Program mencetak nilai yang dibaca dari input menggunakan `System.out.print`, menampilkan "Nilai yang dibaca : " diikuti oleh nilai dari variabel `a`.

- **Output Program:**

Ketika dijalankan, program akan meminta pengguna untuk memasukkan nilai integer dan kemudian menampilkan nilai tersebut. Contohnya:

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac BacaData.java

C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>java BacaData
Contoh membaca dan menulis, ketik nilai integer:
8
Nilai yang dibaca : 8
```

5. Bacakar.java

- **Import Kelas:**

Program dimulai dengan mengimpor kelas `BufferedReader`, `IOException`, dan `InputStreamReader` dari paket `java.io`. Kelas-kelas ini digunakan untuk membaca input dari konsol.

- **Deklarasi Kelas:**

Kelas `Bacakar` dideklarasikan sebagai kelas publik, yang merupakan struktur dasar dalam Java untuk mengelompokkan kode.

- **Metode main:**

Metode `main` adalah titik masuk untuk eksekusi program. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah. Metode ini juga melemparkan `IOException`, yang mungkin terjadi saat membaca input.

- **Deklarasi Variabel:**

- `char cc;` Variabel `cc` dideklarasikan untuk menyimpan karakter yang dibaca dari input.
- `int bil;` Variabel `bil` dideklarasikan untuk menyimpan bilangan bulat yang dibaca dari input.

- **Inisialisasi `BufferedReader`:**

- `InputStreamReader isr = new InputStreamReader(System.in);` Objek `isr` diinisialisasi untuk membaca input dari konsol.
- `BufferedReader dataIn = new BufferedReader(isr);` Objek `dataIn` diinisialisasi untuk membaca input baris demi baris.
- Komentar juga menunjukkan cara alternatif untuk menginisialisasi `BufferedReader`.

- **Pencetakan Pesan ke Konsol:**

Program mencetak pesan "hello" dan meminta pengguna untuk memasukkan satu karakter.

- **Membaca Input Karakter:**

- `cc = dataIn.readLine().charAt(0);` Menggunakan `readLine()` untuk membaca satu baris input dari pengguna dan mengambil karakter pertama dari baris tersebut untuk disimpan di variabel `cc`.
- **Membaca Input Bilangan Bulat:**
 - Program meminta pengguna untuk memasukkan bilangan bulat.
 - `bil = Integer.parseInt(dataIn.readLine());` Membaca satu baris input, mengonversinya menjadi integer, dan menyimpannya dalam variabel `bil`.
- **Mencetak Nilai yang Dibaca:**
Program mencetak nilai karakter dan bilangan yang dibaca. Kemudian mencetak "bye" sebagai tanda akhir program.
- **Output Program:**
Ketika dijalankan, program akan meminta pengguna untuk memasukkan karakter dan bilangan bulat. Contoh outputnya bisa seperti ini:

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac Bacakar.java

C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>java Bacakar
hello
baca 1 karakter : a
baca 1 bilangan : 4
a
4
bye
```

6. Casting1.java

- **Deklarasi Kelas:**
Program dimulai dengan mendeklarasikan kelas bernama `Casting1`, yang berfungsi sebagai wadah untuk kode.
- **Metode `main`:**
Metode `main` adalah titik masuk untuk eksekusi program. Ini adalah metode yang akan dijalankan saat program dieksekusi. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.
- **Deklarasi Variabel:**
 - `int a = 5, b = 6;` Dua variabel integer `a` dan `b` diinisialisasi dengan nilai 5 dan 6.

- `float d = 2.f, e = 3.2f;`: Dua variabel float `d` dan `e` diinisialisasi dengan nilai 2.0 dan 3.2.
- `char g = '5';`: Variabel karakter `g` diinisialisasi dengan karakter '5'.
- `double k = 3.14;`: Variabel double `k` diinisialisasi dengan nilai 3.14.

- **Pencetakan Hasil Casting:**

Program melakukan berbagai jenis casting dan mencetak hasilnya ke konsol dengan menggunakan `System.out.println()`:

- `(float)a`: Mengonversi `a` (integer) menjadi float. Hasilnya adalah 5.0.
- `(double)b`: Mengonversi `b` (integer) menjadi double. Hasilnya adalah 6.0.
- `(int)d`: Mengonversi `d` (float) menjadi integer. Hasilnya adalah 2 (desimal dibuang).
- `(double)e`: Mengonversi `e` (float) menjadi double. Hasilnya adalah 3.2.
- `(int)g`: Mengonversi `g` (char) menjadi integer. Hasilnya adalah 53, yang merupakan nilai ASCII untuk karakter '5'.
- `(float)g`: Mengonversi `g` (char) menjadi float. Hasilnya juga 53.0.
- `(double)g`: Mengonversi `g` (char) menjadi double. Hasilnya adalah 53.0.
- `(int)k`: Mengonversi `k` (double) menjadi integer. Hasilnya adalah 3 (desimal dibuang).
- `(float)k`: Mengonversi `k` (double) menjadi float. Hasilnya adalah 3.14.

- **Output Program:**

Ketika dijalankan, program akan mencetak hasil dari setiap operasi casting ke konsol. Hasilnya akan terlihat seperti ini:

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac Casting1.java
C:\Users\ASUS\Documents\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

- **Deklarasi Kelas:**

Program dimulai dengan mendeklarasikan kelas bernama `Casting2`.

- **Metode main:**

Metode `main` adalah titik masuk untuk eksekusi program. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.

- **Deklarasi Variabel:**

- `int a = 8, b = 9;;` Dua variabel integer `a` dan `b` diinisialisasi dengan nilai 8 dan 9.
- `float d = 2.f, e = 3.2f;;` Dua variabel float `d` dan `e` diinisialisasi dengan nilai 2.0 dan 3.2.
- `char g = '5';` Variabel karakter `g` diinisialisasi dengan karakter '5'.
- `double k = 3.14;;` Variabel double `k` diinisialisasi dengan nilai 3.14.
- `String n = "67", m = "45", l = "100";` Tiga variabel string `n`, `m`, dan `l` diinisialisasi dengan string "67", "45", dan "100".

- **Konversi dari String ke Tipe Primitif:**

- `a = Integer.parseInt(n);` Mengonversi string `n` menjadi integer dan menyimpannya di `a`.
- `k = Double.parseDouble(m);` Mengonversi string `m` menjadi double dan menyimpannya di `k`.
- `d = Float.parseFloat(l);` Mengonversi string `l` menjadi float dan menyimpannya di `d`.
- Program mencetak nilai `a`, `k`, dan `d`.

- **Konversi dari Tipe Primitif ke String:**

- `n = String.valueOf(b);` Mengonversi integer `b` menjadi string dan menyimpannya di `n`.
- `m = String.valueOf(g);` Mengonversi karakter `g` menjadi string dan menyimpannya di `m`.
- `l = String.valueOf(e);` Mengonversi float `e` menjadi string dan menyimpannya di `l`.
- Program mencetak nilai `n`, `m`, dan `l`.

- **Konversi menggunakan `Integer.valueOf()` dan `Double.valueOf()`:**

- `k = Double.valueOf(a).intValue();` Mengonversi integer `a` menjadi objek `Double`, lalu mengambil nilai integer dari objek tersebut dan menyimpannya di `k`.
- `double c = Integer.valueOf(b).doubleValue();` Mengonversi integer `b` menjadi objek `Integer`, lalu mengambil nilai double dari objek tersebut dan menyimpannya di `c`.
- Program mencetak nilai `k`, `c`, dan `l`.

- **Output Program:**

Ketika dijalankan, program akan mencetak hasil dari setiap konversi. Contoh outputnya bisa seperti ini:

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac Casting2.java

C:\Users\ASUS\Documents\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

- **Deklarasi Kelas:**

Program dimulai dengan mendeklarasikan kelas bernama `Ekspresi`.

- **Metode `main`:**

Metode `main` adalah titik masuk untuk eksekusi program. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.

- **Deklarasi Variabel:**

- `int x = 1;` Variabel `x` dideklarasikan dan diinisialisasi dengan nilai 1.
- `int y = 2;` Variabel `y` dideklarasikan dan diinisialisasi dengan nilai 2.

- **Pencetakan Nilai Variabel:**

- Program mencetak nilai dari variabel `x` dan `y` menggunakan `System.out.print()`, menampilkan :

`x = 1`

`y = 2`

- **Penggunaan Ekspresi Kondisional:**

- Program menggunakan ekspresi ternary untuk membandingkan `x` dan `y`.

- Ekspresi $(x < y) ? x : y$ akan memeriksa apakah x kurang dari y . Jika benar, nilai x akan dipilih; jika salah, nilai y akan dipilih.
- Hasil dari ekspresi ini dicetak ke konsol. Dalam kasus ini, karena $1 < 2$ adalah benar, hasilnya adalah 1.
- Pencetakan Hasil Ekspresi:
 - Program mencetak hasil ekspresi ke konsol, menampilkan:

```
C:\Users\ASUS\Documents\Tugas PB0\Tugas 1\Source Code>javac Ekspresi.java
C:\Users\ASUS\Documents\Tugas PB0\Tugas 1\Source Code>java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
```

9. Ekspresi1.java

1. Deklarasi Kelas:
Program dimulai dengan mendeklarasikan kelas bernama Ekspresi1.
2. Metode main:
Metode main adalah titik masuk untuk eksekusi program. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.
3. Deklarasi Variabel:
 - `int x = 1;` Variabel x dideklarasikan dan diinisialisasi dengan nilai 1.
 - `int y = 2;` Variabel y dideklarasikan dan diinisialisasi dengan nilai 2.
 - `float fx; float fy;` Dua variabel float fx dan fy dideklarasikan tetapi belum diinisialisasi.
4. Pencetakan Hasil Pembagian:
 - `System.out.print("x/y (format integer) = "+ x/y);`
Mencetak hasil pembagian x dan y sebagai integer. Karena kedua variabel adalah integer, hasil pembagian juga akan menjadi integer (kebalikan dibuang) dan akan mencetak 0.
 - `System.out.print("\nx/y (format float) = "+ x/y);`
Mencetak hasil pembagian x dan y sebagai float, tetapi hasilnya masih 0.0 karena pembagian dilakukan antara dua integer.
5. Inisialisasi Variabel Float:
 - `fx = x; dan fy = y;` Variabel fx dan fy diinisialisasi dengan nilai dari x dan y , sehingga mereka sekarang memiliki tipe data float.
6. Pencetakan Hasil Pembagian Float:

- o `System.out.print("\nx/y (format integer) = "+ fx/fy);:`
Mencetak hasil pembagian `fx` dan `fy` sebagai integer. Ini akan menghasilkan hasil yang lebih tepat sesuai dengan nilai float, tetapi hasil ini juga akan menjadi integer (kebalikan dibuang).
- o `System.out.print("\nx/y (format float) = "+ fx/fy);:`
Mencetak hasil pembagian `fx` dan `fy` sebagai float, yang akan memberikan hasil yang akurat dari 1.0.

7. Casting:

- o `System.out.print("\nfloat(x)/float(y) (format integer) = "+ (float)x/(float)y);:` Mengonversi `x` dan `y` ke float sebelum melakukan pembagian, hasilnya akan dicetak sebagai integer (kebalikan dibuang).
- o `System.out.print("\nfloat(x)/float(y) (format float) = "+ (float)x/(float)y);:` Mengonversi `x` dan `y` ke float sebelum melakukan pembagian, hasilnya akan dicetak sebagai float.

8. Mengubah Nilai `x` dan `y`:

- o `x = 10; y = 3;:` Nilai `x` dan `y` diubah menjadi 10 dan 3.

9. Pencetakan Hasil Pembagian dengan Nilai Baru:

- o `System.out.print("\nx/y (format integer) = "+ x/y);:`
Mencetak hasil pembagian 10 / 3 sebagai integer, yang akan menghasilkan 3.
- o `System.out.print("\nx/y (format float) = "+ x/y);:`
Mencetak hasil pembagian 10 / 3 sebagai float, yang akan menghasilkan 3.3333....

Output Program

Ketika dijalankan, output program ini akan terlihat seperti ini:

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac Ekspresi1.java
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>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
```

10. Hello.java

• Deklarasi Kelas:

Program dimulai dengan mendeklarasikan kelas bernama `Hello`.

- Metode `main`:

Metode `main` adalah titik masuk untuk eksekusi program. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.

- Pencetakan Teks ke Konsol:

- `System.out.print("Hello");` Mencetak teks "Hello" ke konsol tanpa menambahkan garis baru setelahnya. Hasil output saat itu adalah:

Hello

- `System.out.println("World");` Mencetak "World" ke konsol dan secara otomatis menambahkan garis baru setelahnya. Hasil output diperbarui menjadi:

Hello

Hello World

- `System.out.println("Welcome");` Mencetak "Welcome" ke konsol dengan garis baru setelahnya. Hasil output akhir adalah:

Hello

Hello World

Welcome

Output Program

Ketika dijalankan, output program ini akan terlihat seperti ini:

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac Hello.java
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>java Hello
Hello
Hello World
Welcome
```

11. Incr.java

- Deklarasi Kelas:

Program dimulai dengan mendeklarasikan kelas bernama `Incr`.

- Metode `main`:

Metode `main` adalah titik masuk untuk eksekusi program. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.

- Deklarasi Variabel:

- `int i, j`;: Dua variabel integer `i` dan `j` dideklarasikan.

- Inisialisasi dan Penggunaan Operator Increment:

- `i = 3`;: Variabel `i` diinisialisasi dengan nilai 3.
 - `j = i++`;: Di sini, nilai `i` (yaitu 3) diberikan ke `j`, dan kemudian `i` di-increment. Namun, karena ini menggunakan post-increment (`i++`), nilai `i` yang digunakan untuk `j` adalah nilai sebelum increment, yaitu 3. Setelah baris ini, `i` menjadi 4, dan `j` tetap 3.

- Pencetakan Hasil:

- `System.out.println ("Nilai i : " + (++i) + "\nNilai j : " + j)`;:
 - `++i` adalah pre-increment, yang berarti nilai `i` akan di-increment terlebih dahulu sebelum digunakan. Jadi, `i` yang sebelumnya 4 akan menjadi 5.
 - Program mencetak nilai `i` sebagai 5 dan nilai `j` sebagai 3.
 - Hasil output dari pernyataan ini adalah :

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac Incr.java
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>java Incr
Nilai i : 5
Nilai j : 3
```

12. Oper1.java

- Deklarasi Kelas:

Program dimulai dengan mendeklarasikan kelas bernama `Oper1`.

- Metode `main`:

Metode `main` adalah titik masuk untuk eksekusi program. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.

- Deklarasi Variabel:

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

- Pencetakan Nilai Variabel:

- Program mencetak nilai dari `n`, `x`, dan `y` ke konsol :

`n = 10`

`x = 1`

`y = 2`

1. Operasi Bitwise dan Pergeseran:

- `System.out.println("n & 8 = " + (n & 8));`:
 - Operasi AND antara `n` (1010) dan `8` (1000) menghasilkan 0 (0000).
 - Outputnya adalah `n & 8 = 0`.
- `System.out.println("x & ~8 = " + (x & ~8));`:
 - `~8` adalah negasi bit dari `8` (1000), yang menghasilkan 0111.
 - Operasi AND antara `x` (0001) dan 0111 menghasilkan 1 (0001).
 - Outputnya adalah `x & ~8 = 1`.
- `System.out.println("y << 2 = " + (y << 2));`:
 - Operasi `<<` (shift left) menggeser bit `y` (0010) ke kiri sebanyak 2 posisi, menghasilkan 1000 (8).
 - Outputnya adalah `y << 2 = 8`.
- `System.out.println("y >> 3 = " + (y >> 3));`:
 - Operasi `>>` (shift right) menggeser bit `y` (0010) ke kanan sebanyak 3 posisi, menghasilkan 0000 (0).
 - Outputnya adalah `y >> 3 = 0`.

Output Program

Ketika dijalankan, output program ini akan terlihat seperti ini:

```

C:\Users\ASUS\Documents\Tugas PB0\Tugas 1\Source Code>javac Oper1.java

C:\Users\ASUS\Documents\Tugas PB0\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:
Program dimulai dengan mendeklarasikan kelas bernama Oper2.
2. Metode main:
Metode main adalah titik masuk untuk eksekusi program. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.
3. Deklarasi Variabel:
 - o `char i, j`;: Dua variabel `char` (`i` dan `j`) dideklarasikan.
4. Inisialisasi Variabel:
 - o `i = 3`;: Variabel `i` diinisialisasi dengan nilai 3, yang dalam representasi biner adalah 00000011.
 - o `j = 4`;: Variabel `j` diinisialisasi dengan nilai 4, yang dalam representasi biner adalah 00000100.
5. Pencetakan Nilai Variabel:
 - o `System.out.println("i = " + (int) i);`: Mencetak nilai `i` dengan mengonversinya menjadi integer. Hasilnya adalah 3.
 - o `System.out.println("j = " + j);`: Mencetak nilai `j`, yang hasilnya adalah 4.
6. Operasi Bitwise:
 - o `System.out.println("i & j = " + (i & j));`:
 - Operasi AND antara `i` (00000011) dan `j` (00000100) menghasilkan 0 (00000000).
 - Outputnya adalah `i & j = 0`.
 - o `System.out.println("i | j = " + (i | j));`:
 - Operasi OR antara `i` (00000011) dan `j` (00000100) menghasilkan 7 (00000111).
 - Outputnya adalah `i | j = 7`.
 - o `System.out.println("i ^ j = " + (i ^ j));`:
 - Operasi XOR antara `i` (00000011) dan `j` (00000100) juga menghasilkan 7 (00000111).

- Outputnya adalah $i \wedge j = 7$. Penting untuk dicatat bahwa operator \wedge dalam Java adalah operator XOR, bukan pangkat.
7. Pemangkatan dengan Kelas Math:
- `System.out.println(Math.pow(i, j));` Menggunakan metode `pow(a, b)` dari kelas `Math` untuk menghitung i pangkat j , yaitu 3^4 (81.0). Hasilnya adalah 81.0.
8. Negasi Bitwise:
- `System.out.println(" ~i = " + ~i);`
 - Operasi negasi bitwise `~` pada i menghasilkan -4. Dalam representasi biner, ini adalah 11111100 dalam 8 bit.
 - Hasil outputnya adalah `~i = -4`.

Output Program

Ketika dijalankan, output program ini akan terlihat seperti ini:

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac Oper2.java

C:\Users\ASUS\Documents\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:
Program dimulai dengan mendeklarasikan kelas bernama `Oper3`.
2. Metode `main`:
Metode `main` adalah titik masuk untuk eksekusi program. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.
3. Operasi Logika:
 - Pertama: `if (true && true):`
 - Ekspresi ini menggunakan operator AND pendek (`&&`). Karena kedua operand adalah `true`, maka kondisi ini benar dan mencetak `true` ke konsol.
 - Output: `true`
 - Kedua: `if (true & true):`

- Ekspresi ini menggunakan operator AND biasa (&). Meskipun ini juga benar, operator ini tidak memiliki mekanisme short-circuiting, sehingga kedua operand dievaluasi. Namun, hasil yang dicetak adalah `false` karena pernyataan berikutnya (`System.out.println(true & false)`) dieksekusi.
- Output: `false`
- Ketiga: `if (true):`
 - Ekspresi ini selalu benar, sehingga mencetak `true`.
 - Output: `true`
- Keempat: `if (true || true):`
 - Ekspresi ini menggunakan operator OR pendek (`||`). Karena salah satu operand adalah `true`, maka kondisi ini benar dan mencetak `true` ke konsol.
 - Output: `true`
- Kelima: `if (true | false):`
 - Ekspresi ini menggunakan operator OR biasa (`|`). Hasilnya adalah `true`, dan meskipun operand kedua adalah `false`, keduanya dievaluasi.
 - Output: `true`

Output Program

Ketika dijalankan, output program ini akan terlihat seperti ini:

```
C:\Users\ASUS\Documents\Tugas PB0\Tugas 1\Source Code>javac Oper3.java

C:\Users\ASUS\Documents\Tugas PB0\Tugas 1\Source Code>java Oper3
true
false
true
true
true
```

15. Oper4.java

1. Deklarasi Kelas:
Program dimulai dengan mendeklarasikan kelas bernama `Oper4`.
2. Metode `main`:
Metode `main` adalah titik masuk untuk eksekusi program. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.
3. Deklarasi Variabel:
 - `int i = 0;` Variabel `i` diinisialisasi dengan nilai 0.

- o `int j = 0;` Variabel `j` diinisialisasi dengan nilai 0.
- o `char c = 8; char d = 10;` Dua variabel karakter `c` dan `d` diinisialisasi dengan nilai 8 (karakter ASCII untuk Backspace) dan 10 (karakter ASCII untuk Line Feed).

4. Penggunaan Operator Ternary:

- o `int e = (((int)c > (int)d) ? c : d);`
 - Menggunakan operator ternary untuk membandingkan `c` dan `d`. Karena 8 tidak lebih besar dari 10, hasilnya adalah nilai 10, sehingga `e` diinisialisasi dengan 10.
- o `int k = ((i > j) ? i : j);`
 - Pada saat ini, `i` dan `j` keduanya adalah 0, sehingga hasilnya adalah 0. Jadi, `k` diinisialisasi dengan 0.

5. Pencetakan Nilai:

- o `System.out.print("Nilai e = " + e);` Mencetak nilai `e`, yaitu 10.
- o `System.out.print("\nNilai k = " + k);` Mencetak nilai `k`, yaitu 0.

6. Inisialisasi Ulang Variabel:

- o `i = 2;` Nilai `i` diubah menjadi 2.
- o `j = 3;` Nilai `j` diubah menjadi 3.

7. Penggunaan Increment dalam Operator Ternary:

- o `k = ((i++ > j++) ? i : j);`
 - Ekspresi ini membandingkan `i` (2) dan `j` (3) sebelum melakukan increment. Karena 2 tidak lebih besar dari 3, hasil dari ekspresi ini adalah nilai `j` yang sudah di-increment, yaitu 4 (karena `j++` akan meningkatkan `j` menjadi 4 setelah evaluasi).

8. Pencetakan Nilai Kembali:

- o `System.out.print("\nNilai k = " + k);` Mencetak nilai `k`, yang dihasilkan dari perbandingan sebelumnya, yaitu 4.

Output Program

Ketika dijalankan, output program ini akan terlihat seperti ini:

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac Oper4.java
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4
```

16. Operator.java

- Deklarasi Kelas:

Program dimulai dengan mendeklarasikan kelas bernama `Oprator`.

- Metode `main`:

Metode `main` adalah titik masuk untuk eksekusi program. Parameter `String[] args` memungkinkan program menerima argumen dari baris perintah.

- Deklarasi Variabel:

- `boolean Bool1, Bool2, TF;;` Tiga variabel boolean untuk menyimpan nilai logika.
- `int i, j, hsl;;` Tiga variabel integer, di mana `hsl` akan digunakan untuk menyimpan hasil operasi.
- `float x, y, res;;` Tiga variabel float untuk menyimpan nilai desimal dan hasil operasi.

- Pencetakan Teks:

- `System.out.println("Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output");` Mencetak instruksi ke konsol.

- Operasi Boolean:

- `Bool1 = true; Bool2 = false;;` Menginisialisasi `Bool1` dengan `true` dan `Bool2` dengan `false`.
- `TF = Bool1 && Bool2;;` Menggunakan operator AND (`&&`), hasilnya `false`.
- `TF = Bool1 || Bool2;;` Menggunakan operator OR (`||`), hasilnya `true`.
- `TF = !Bool1;;` Menggunakan operator NOT (`!`), hasilnya `false`.
- `TF = Bool1 ^ Bool2;;` Menggunakan operator XOR (`^`), hasilnya `true` (karena hanya satu operand yang `true`).

- Operasi Numerik:

- `i = 5; j = 2;;` Menginisialisasi `i` dengan 5 dan `j` dengan 2.
- `hsl = i + j;;` Penjumlahan, hasilnya 7.
- `hsl = i - j;;` Pengurangan, hasilnya 3.
- `hsl = i / j;;` Pembagian bulat, hasilnya 2.
- `hsl = i * j;;` Perkalian, hasilnya 10.
- `hsl = i % j;;` Modulo, hasilnya 1 (sisa pembagian).

- Operasi Numerik dengan Float:

- `x = 5; y = 5;` Menginisialisasi `x` dan `y` dengan `5.0`.
- `res = x + y;` Penjumlahan, hasilnya `10.0`.
- `res = x - y;` Pengurangan, hasilnya `0.0`.
- `res = x / y;` Pembagian, hasilnya `1.0`.
- `res = x * y;` Perkalian, hasilnya `25.0`.

- Operasi Relasional:

- `TF = (i == j);` Memeriksa kesetaraan, hasilnya `false`.
- `TF = (i != j);` Memeriksa ketidaksamaan, hasilnya `true`.
- `TF = (i < j);` Memeriksa apakah `i` kurang dari `j`, hasilnya `false`.
- `TF = (i > j);` Memeriksa apakah `i` lebih besar dari `j`, hasilnya `true`.
- `TF = (i <= j);` Memeriksa apakah `i` kurang dari atau sama dengan `j`, hasilnya `false`.
- `TF = (i >= j);` Memeriksa apakah `i` lebih besar atau sama dengan `j`, hasilnya `true`.

- Operasi Relasional dengan Float:

- `TF = (x != y);` Memeriksa ketidaksamaan, hasilnya `false`.
- `TF = (x < y);` Memeriksa apakah `x` kurang dari `y`, hasilnya `false`.
- `TF = (x > y);` Memeriksa apakah `x` lebih besar dari `y`, hasilnya `false`.
- `TF = (x <= y);` Memeriksa apakah `x` kurang dari atau sama dengan `y`, hasilnya `true`.
- `TF = (x >= y);` Memeriksa apakah `x` lebih besar atau sama dengan `y`, hasilnya `true`.

OUTPUT PROGRAM

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>javac Oprator.java
```

```
C:\Users\ASUS\Documents\Tugas PBO\Tugas 1\Source Code>java Oprator  
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output
```