

TUGAS 1

Laporan Pemrograman berorientasi objek

Nama : Tizar Fitrah Anugerah

Nim : 13020220005

Tugas Praktek

Kasus 1

Output:

```
1. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java Asgdll
   f : 20.0
   f11: 10.0
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk menentukan bahwa kelas Asgdll dan metode main dapat diakses dari mana saja dalam program atau paket Java yang berbeda.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam konteks ini, kelas Asgdll didefinisikan.
3. **static**: digunakan dalam konteks metode main. Ini berarti bahwa metode main bukan bagian dari suatu instance kelas, tetapi terkait dengan kelas itu sendiri. Dengan kata lain, Anda dapat memanggil metode main tanpa membuat objek dari kelas Asgdll.
4. **void**: digunakan sebelum nama metode main. Dalam deklarasi main, void menunjukkan bahwa metode tidak mengembalikan nilai apapun setelah dieksekusi.
5. **main**: Metode main adalah titik awal dari eksekusi program Java. Ketika Anda menjalankan program Java, JDK (Java Development Kit) akan mencari metode main dan memulai eksekusi program
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **float**: digunakan untuk merepresentasikan bilangan pecahan (floating-point) dengan presisi yang lebih rendah dibandingkan dengan tipe data double. Variabel f dideklarasikan sebagai float dengan nilai awal 20.0f.
8. **double**: digunakan untuk merepresentasikan bilangan pecahan (floating-point) dengan presisi yang lebih tinggi daripada tipe data float. Variabel f11 dideklarasikan sebagai double tanpa memberikan nilai awal.
9. **System.out.println**: System adalah kelas yang menyediakan akses ke sistem, out adalah objek PrintStream yang merupakan output standar, dan println adalah metode untuk mencetak nilai ke output.

Kasus 2

Output :

```
2. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java Assign  
hello  
Ini nilai i : 5
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas Assign sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Assign.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas Assign. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**:. Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **int**: Int tipe data yang digunakan untuk mendeklarasikan variabel i sebagai bilangan bulat (integer).
8. **System.out.print()**: perintah untuk mencetak teks ke konsol tanpa menambahkan karakter baris baru. Dalam program ini, "hello" dicetak ke konsol.
9. **System.out.println**: Dalam program ini, teks "Ini nilai i : " digabungkan dengan nilai variabel i dan kemudian dicetak ke konsol dengan penambahan baris baru.

Kasus 3

Output :

```
3. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>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
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas Assign sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Assign.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas Assign. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**:. Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **short, int, long**: Ini adalah tipe data untuk menyimpan bilangan bulat dengan ukuran yang berbeda. short memiliki ukuran 16 bit, int memiliki ukuran 32 bit, dan long memiliki ukuran 64 bit.
8. **char**: Ini adalah tipe data untuk menyimpan karakter. Dalam program ini, karakter "A" dan "z" disimpan dalam variabel c dan c1 secara berurutan.
9. **double, float**: Ini adalah tipe data untuk menyimpan bilangan real dengan presisi yang berbeda. double digunakan untuk presisi ganda (64 bit), sementara float digunakan untuk presisi tunggal (32 bit).
10. **System.out.println**: System adalah kelas yang menyediakan akses ke sistem, out adalah objek output standar, dan println adalah metode yang digunakan untuk mencetak teks ke konsol dengan menambahkan karakter baris baru setelahnya

11. `short ks = 1;` : Mendeklarasikan variabel `ks` dengan tipe data `short` dan memberikan nilai awal 1.
12. `int ki = 1;` : Mendeklarasikan variabel `ki` dengan tipe data `int` dan memberikan nilai awal 1.
13. `long kl = 10000;` : Mendeklarasikan variabel `kl` dengan tipe data `long` dan memberikan nilai awal 10000.
14. `char c = 65;` : Mendeklarasikan variabel `c` dengan tipe data `char` dan memberikan nilai awal 65, yang merupakan representasi ASCII untuk huruf 'A'.
15. `char c1 = 'z';` : Mendeklarasikan variabel `c1` dengan tipe data `char` dan memberikan nilai awal 'z'.
16. `double x = 50.2f;` : Mendeklarasikan variabel `x` dengan tipe data `double` dan memberikan nilai awal 50.2 (yang direpresentasikan oleh 50.2f yang merupakan float).
17. `float y = 50.2f;` : Mendeklarasikan variabel `y` dengan tipe data `float` dan memberikan nilai awal 50.2f.

Kasus 4

Output :

```
4. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java BacaData
Contoh membaca dan menulis,ketik integer :
4
Nilai yang di baca : 4
```

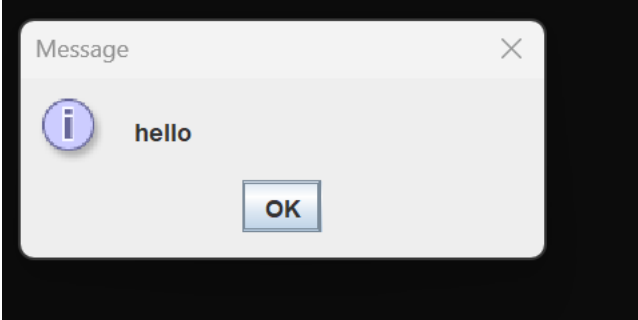
Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **Import java.util.Scanner:** Baris pertama dari program mengimpor kelas Scanner yang terdapat dalam paket java.util. Ini diperlukan agar kita dapat menggunakan kelas Scanner untuk membaca input dari pengguna.
2. **public:** digunakan untuk mendeklarasikan kelas BacaData sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
3. **class:** Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah BacaData.
4. **static:** membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas BacaData. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
5. **void:** digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
6. **main:** Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
7. **String[] args:** String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
8. **int a :** Mendeklarasikan variabel a sebagai tipe data integer yang akan digunakan untuk menyimpan nilai yang akan dimasukkan oleh pengguna.
9. **Scanner masukan:** Mendeklarasikan variabel masukan sebagai objek dari kelas Scanner yang akan digunakan untuk membaca input dari pengguna.
10. **System.out.println("Contoh membaca dan menulis,ketik integer : ");** mencetak pesan ke layar yang meminta pengguna untuk memasukkan nilai integer.
11. **Membuat Objek Scanner:** masukan = new Scanner(System.in); membuat objek Scanner dengan parameter System.in. Ini akan mengalihkan input dari keyboard pengguna ke objek masukan.
12. **Membaca Input:** a = masukan.nextInt(); membaca input yang dimasukkan oleh pengguna dan menyimpannya dalam variabel a. Metode nextInt() digunakan untuk membaca input integer dari pengguna.
13. **Mencetak Output:** System.out.print("Nilai yang di baca : " + a); mencetak nilai yang telah dibaca dari input pengguna.

Kasus 5

Output :

```
5. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java Bacakar
hello
baca 1 karakter : a
baca 1 bilangan : 1
a
```



Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **import**: Keyword import digunakan untuk mengimpor paket atau kelas dari paket tertentu ke dalam program Java. Dalam program ini, import digunakan untuk mengimpor kelas-kelas yang diperlukan untuk membaca input dari pengguna dan menampilkan dialog dalam GUI.
2. **java.io.BufferedReader**: Ini adalah kelas yang digunakan untuk membaca teks dari input stream dengan membaca beberapa karakter sekaligus. Dalam program ini, BufferedReader digunakan untuk membaca input dari System.in (keyboard).
3. **java.io.InputStreamReader**: Kelas InputStreamReader digunakan untuk membaca input stream byte dan menerjemahkannya menjadi karakter.
4. **java.io.IOException**: Ini adalah pengecualian yang mungkin terjadi saat berinteraksi dengan input/output (IO) dalam Java. Dalam program ini, pengecualian IOException ditangkap dan dihandle dalam metode main dengan menggunakan kata kunci throws.
5. **javax.swing.JOptionPane**: Ini adalah kelas yang digunakan untuk menampilkan dialog dalam GUI Java. Dalam program ini, JOptionPane digunakan untuk menampilkan dialog pesan dan meminta input dari pengguna.
6. **public**: digunakan untuk mendeklarasikan kelas Bacakar sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
7. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Bacakar.
8. **public static void main(String[] args) throws IOException**: metode utama yang akan dieksekusi oleh Java saat menjalankan program. Metode ini memiliki kata kunci public, static, dan void. public menunjukkan bahwa metode ini dapat diakses dari luar kelas. static menunjukkan bahwa metode ini adalah metode kelas (bukan metode instance) sehingga

dapat dipanggil tanpa membuat objek dari kelasnya. void menunjukkan bahwa metode ini tidak mengembalikan nilai apapun. throws IOException menunjukkan bahwa metode dapat melempar pengecualian IOException.

9. **System.out.print**: perintah untuk mencetak teks ke konsol. Dalam program ini, digunakan untuk mencetak pesan-pesan tertentu ke layar.
10. **dataIn.readLine().charAt(0)**: membaca karakter dari input yang diberikan oleh pengguna. dataIn.readLine() membaca satu baris teks dari input, dan charAt(0) digunakan untuk mengambil karakter pertama dari baris tersebut.
11. **Integer.parseInt(dataIn.readLine())**: membaca bilangan bulat dari input yang diberikan oleh pengguna. dataIn.readLine() membaca satu baris teks dari input, dan Integer.parseInt() digunakan untuk mengonversi teks tersebut menjadi bilangan bulat.
12. **JOptionPane.showInputDialog()**: metode yang digunakan untuk menampilkan dialog input dari pengguna dalam GUI. Dalam program ini, digunakan untuk meminta pengguna untuk memasukkan karakter.
13. **JOptionPane.showMessageDialog()**: metode yang digunakan untuk menampilkan dialog pesan dalam GUI. Dalam program ini, digunakan untuk menampilkan pesan sambutan.
14. **System.out.println(cc + "\n" + bil + "\n")**: perintah untuk mencetak karakter dan bilangan bulat yang telah dibaca dari input pengguna ke konsol.
15. **cc**: Ini adalah variabel yang digunakan untuk menyimpan karakter yang dibaca dari input pengguna.
16. **bil**: variabel yang digunakan untuk menyimpan bilangan bulat yang dibaca dari input pengguna.
17. **String kar**: yang digunakan untuk menyimpan karakter yang dimasukkan melalui dialog input GUI.

Kasus 6

Output :

```
6. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java Casting
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas Casting sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Casting.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas Casting. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**:. Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **int a=5,b=6; float d=2.f,e=3.2f; char g='5'; double k=3.14**;; deklarasi dan inisialisasi beberapa variabel dengan tipe data yang berbeda:
 - a dan b dideklarasikan sebagai int dan diberi nilai 5 dan 6.
 - d dan e dideklarasikan sebagai float dan diberi nilai 2.f (2.0) dan 3.2f.
 - g dideklarasikan sebagai char dan diberi nilai '5'.
 - k dideklarasikan sebagai double dan diberi nilai 3.14.
8. **System.out.println((float)a)**;; mencetak nilai dari variabel a yang telah di-casting menjadi float. Dalam hal ini, nilai a yang awalnya integer 5 akan diubah menjadi float 5.0 dan kemudian dicetak.

9. **System.out.println((double)b);**:mencetak nilai dari variabel b yang telah di-casting menjadi double. Dalam hal ini, nilai b yang awalnya integer 6 akan diubah menjadi double 6.0 dan kemudian dicetak.
10. **System.out.println((int)d);**:mencetak nilai dari variabel d yang telah di-casting menjadi int. Dalam hal ini, nilai d yang awalnya float 2.0 akan diubah menjadi integer 2 dan kemudian dicetak.
11. **System.out.println((double)e);**:mencetak nilai dari variabel e yang telah di-casting menjadi double. Dalam hal ini, nilai e yang awalnya float 3.2 akan diubah menjadi double 3.2 dan kemudian dicetak.
12. **System.out.println((int)g);**:mencetak nilai dari variabel g yang telah di-casting menjadi int. Dalam hal ini, karakter '5' akan diubah menjadi integer sesuai dengan kode ASCII-nya (53) dan kemudian dicetak.
13. **System.out.println((float)g);**:mencetak nilai dari variabel g yang telah di-casting menjadi float. Dalam hal ini, karakter '5' akan diubah menjadi float sesuai dengan nilai ASCII-nya (53.0) dan kemudian dicetak.
14. **System.out.println((double)g);**:mencetak nilai dari variabel g yang telah di-casting menjadi double. Dalam hal ini, karakter '5' akan diubah menjadi double sesuai dengan nilai ASCII-nya (53.0) dan kemudian dicetak.
15. **System.out.println((int)k);**:mencetak nilai dari variabel k yang telah di-casting menjadi int. Dalam hal ini, nilai k yang awalnya double 3.14 akan diubah menjadi integer 3 dan kemudian dicetak.
16. **System.out.println((float)k);**:mencetak nilai dari variabel k yang telah di-casting menjadi float. Dalam hal ini, nilai k yang awalnya double 3.14 akan diubah menjadi float 3.14 dan kemudian dicetak.

Kasus 7

Output :

```
7. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>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
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas Casting2 sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Casting2.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas Casting2. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**:. Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **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";**: Ini adalah deklarasi dan inisialisasi beberapa variabel dengan tipe data yang berbeda:
 - a dan b dideklarasikan sebagai int dan diberi nilai 8 dan 9.
 - d dan e dideklarasikan sebagai float dan diberi nilai 2.f (2.0) dan 3.2f.
 - g dideklarasikan sebagai char dan diberi nilai '5'.
 - k dideklarasikan sebagai double dan diberi nilai 3.14.
 - n, m, dan l dideklarasikan sebagai String dan diberi nilai "67", "45", dan "100" secara berurutan.
8. **Integer.parseInt(n); Double.parseDouble(m); Float.parseFloat(l);**: Ini adalah metode yang digunakan untuk mengonversi string menjadi tipe data primitif yang sesuai:

- Integer.parseInt(n) mengonversi string "67" menjadi nilai integer dan disimpan dalam variabel a.
 - Double.parseDouble(m) mengonversi string "45" menjadi nilai double dan disimpan dalam variabel k.
 - Float.parseFloat(l) mengonversi string "100" menjadi nilai float dan disimpan dalam variabel d.
9. **System.out.println("a : "+a+"\nk : "+k+"\nd : "+d);** Ini adalah perintah untuk mencetak nilai variabel a, k, dan d ke layar dengan menggunakan metode println() dari kelas System.out. Hasilnya adalah mencetak nilai variabel tersebut dengan pesan yang terformat.
10. **String.valueOf(b); String.valueOf(g); String.valueOf(e);** Ini adalah metode yang digunakan untuk mengonversi tipe data primitif menjadi string:
- String.valueOf(b) mengonversi nilai integer b menjadi string dan disimpan dalam variabel n.
 - String.valueOf(g) mengonversi karakter '5' menjadi string dan disimpan dalam variabel m.
 - String.valueOf(e) mengonversi nilai float 3.2f menjadi string dan disimpan dalam variabel l.
11. **System.out.println("n : "+n+"\nm : "+m+"\nl : "+l);** Ini adalah perintah untuk mencetak nilai variabel n, m, dan l ke layar dengan menggunakan metode println() dari kelas System.out.
12. **Double.valueOf(a).intValue(); Integer.valueOf(b).doubleValue();** Ini adalah metode yang digunakan untuk mengubah tipe data primitif menjadi objek wrapper dan kemudian mengembalikan tipe data primitif lagi:
- Double.valueOf(a).intValue() mengonversi nilai integer a menjadi objek Double, kemudian mengambil nilainya sebagai integer.
 - Integer.valueOf(b).doubleValue() mengonversi nilai integer b menjadi objek Integer, kemudian mengambil nilainya sebagai double.
13. **System.out.println("k : "+k+"\nc : "+c+"\n l : "+l);** Ini adalah perintah untuk mencetak nilai variabel k, c, dan l ke layar dengan menggunakan metode println() dari kelas System.out.

Kasus 8

Output :

```
8. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java Ekspresi
x = 1
y = 2
Hasil ekspresi = (x<y) ? x:y = 1
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas Ekspresi sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Ekspresi.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas Ekspresi. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**:. Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **int x = 1; int y = 2;**: deklarasi dan inisialisasi dua variabel x dan y dengan tipe data integer, masing-masing diberi nilai 1 dan 2.
8. **System.out.print("x = "+x+ "\n");**: perintah untuk mencetak teks "x = " dan nilai dari variabel x ke konsol, diikuti dengan karakter baris baru (\n). Hasilnya akan dicetak dalam satu baris.
9. **System.out.print("y = "+y+ "\n");**: perintah untuk mencetak teks "y = " dan nilai dari variabel y ke konsol, diikuti dengan karakter baris baru (\n). Hasilnya akan dicetak dalam satu baris.
10. **System.out.print("Hasil ekspresi = (x<y) ? x:y = " + ((x < y) ? x : y));**: perintah untuk mencetak hasil dari ekspresi ternary $(x < y) ? x : y$ ke konsol. Ekspresi ternary ini akan mengevaluasi apakah x lebih kecil dari y. Jika benar, maka nilai x akan dicetak, jika tidak, maka nilai y yang akan dicetak.
 - $(x < y) ? x : y$ adalah ekspresi ternary. Jika kondisi $(x < y)$ benar, maka nilai dari ekspresi ini adalah x, jika tidak, maka nilai dari ekspresi ini adalah y.
11. **System.out.print()**:perintah untuk mencetak teks ke konsol.

Kasus 9

Output :

```
9. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>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
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas Ekspresil sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Ekspresil.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas Ekspresil. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**.. Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **int x = 1; int y = 2;** Ini adalah deklarasi dan inisialisasi dua variabel x dan y dengan tipe data integer, masing-masing diberi nilai 1 dan 2.
8. **float fx; float fy;** Ini adalah deklarasi dua variabel fx dan fy dengan tipe data float.
9. **System.out.print("x/y (format integer) = "+x/y);** perintah untuk mencetak hasil dari operasi pembagian x dengan y (dalam format integer) ke konsol. Namun, perlu dicatat bahwa ini menggunakan pembagian integer, sehingga hasilnya akan menjadi integer juga, yaitu 0.
10. **System.out.print("\nx/y (format float) = "+x/y);** perintah untuk mencetak hasil dari operasi pembagian x dengan y (dalam format float) ke konsol. Tetapi, sama seperti sebelumnya, ini juga menggunakan pembagian integer, sehingga hasilnya akan menjadi integer juga, yaitu 0.
11. **fx=x; fy=y;** perintah untuk mengkonversi nilai dari variabel x dan y ke tipe data float, sehingga nilai-nilai tersebut dapat dibagi secara tepat.

12. **System.out.print("\nx/y (format integer) = " +fx/fy);**perintah untuk mencetak hasil dari operasi pembagian fx dengan fy (dalam format integer) ke konsol, setelah nilai-nilai x dan y dikonversi menjadi float.
13. **System.out.print("\nx/y (format float) = " +fx/fy);** perintah untuk mencetak hasil dari operasi pembagian fx dengan fy (dalam format float) ke konsol.
14. **System.out.print("\nfloat (x) / float(y) (format integer) = "+(float)x/(float)y);** perintah untuk mencetak hasil dari operasi pembagian x yang sudah dikonversi menjadi float dengan y yang sudah dikonversi menjadi float (dalam format integer) ke konsol.
15. **System.out.print("\nfloat (x) / float(y) (format float) = "+(float)x/(float)y);** perintah untuk mencetak hasil dari operasi pembagian x yang sudah dikonversi menjadi float dengan y yang sudah dikonversi menjadi float (dalam format float) ke konsol.
16. **x = 10; y = 3;** Mengubah nilai dari x menjadi 10 dan y menjadi 3.
17. **System.out.print("\nx/y (format integer) = "+x/y);**perintah untuk mencetak hasil dari operasi pembagian x dengan y (dalam format integer) ke konsol. Namun, karena ini menggunakan pembagian integer, hasilnya akan menjadi integer juga, yaitu 3.
18. **System.out.print("\nx/y (format float) = "+x/y);** perintah untuk mencetak hasil dari operasi pembagian x dengan y (dalam format float) ke konsol. Tetapi, sama seperti sebelumnya, ini juga menggunakan pembagian integer, sehingga hasilnya akan menjadi integer juga, yaitu 3.

Kasus 10

Output:

```
10. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java Incr
    Nilai i : 5
    Nilai j : 3
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas Incr sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Incr.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas Incr. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**:. Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **int i, j**;; Ini adalah deklarasi dua variabel i dan j dengan tipe data integer.
8. **i = 3**;; Menginisialisasi variabel i dengan nilai 3.
9. **j = i++**;; operasi penugasan yang akan menugaskan nilai i ke j, kemudian menaikkan nilai i secara bertahap. Jadi, nilai j akan menjadi 3, karena nilai i diteruskan sebelum peningkatan.
10. **System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j)**;; Ini adalah perintah untuk mencetak nilai dari variabel i dan j ke konsol.
 - ++i adalah operator peningkatan yang menambahkan 1 ke nilai i sebelum nilai i digunakan dalam ekspresi. Jadi, nilai i akan menjadi 5 sebelum dicetak.
 - j akan tetap menjadi 3, karena nilai i yang ditugaskan ke j sebelum peningkatan.

Kasus 11

Output :

```
11. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java Oper1
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas Oper1 sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Oper1.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas Oper1. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**: Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **int n = 10; int x = 1; int y = 2;**: deklarasi dan inisialisasi tiga variabel n, x, dan y dengan tipe data integer, masing-masing diberi nilai 10, 1, dan 2.
8. **System.out.println("n = "+n);**: perintah untuk mencetak nilai dari variabel n ke konsol.
9. **System.out.println("x = "+x);**: perintah untuk mencetak nilai dari variabel x ke konsol.
10. **System.out.println("y = "+y);**: perintah untuk mencetak nilai dari variabel y ke konsol.
11. **System.out.println("n & 8 = " + (n & 8));**: operasi bitwise AND (&) antara nilai variabel n dan 8, kemudian hasilnya dicetak ke konsol. Operasi bitwise AND menghasilkan nilai yang sama dengan 1 hanya jika kedua bit yang sesuai dalam representasi biner dari kedua operand adalah 1.
12. **System.out.println("x & ~ 8 = "+ (x & ~8));**: operasi bitwise AND (&) antara nilai variabel x dan komplement dari 8 (~8), kemudian hasilnya dicetak ke konsol. ~8 menghasilkan nilai yang sama dengan 1 jika bit yang sesuai dalam representasi biner

dari 8 adalah 0. Operasi bitwise AND menghasilkan nilai yang sama dengan 1 hanya jika kedua bit yang sesuai dalam representasi biner dari kedua operand adalah 1.

13. **System.out.println("y << 2 = "+ (y << 2));**: operasi bitwise shift ke kiri (<<) pada nilai variabel y sebanyak 2 bit, kemudian hasilnya dicetak ke konsol. Operasi bitwise shift ke kiri menggeser semua bit dalam operand ke kiri sejumlah langkah yang ditentukan, dan bit-bit yang kosong di sebelah kanan diisi dengan 0.
14. **System.out.println("y >> 3 = "+ (y >> 3));**: operasi bitwise shift ke kanan (>>) pada nilai variabel y sebanyak 3 bit, kemudian hasilnya dicetak ke konsol. Operasi bitwise shift ke kanan menggeser semua bit dalam operand ke kanan sejumlah langkah yang ditentukan, dan bit-bit yang kosong di sebelah kiri diisi dengan nilai bit paling signifikan.

Kasus 12

Output :

```
12. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java Oper2
i = 3
j =
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas Oper2 sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Oper2.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas Oper2. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**: Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **char i,j**:: deklarasi dua variabel i dan j dengan tipe data char (karakter).
8. **i = 3; j = 4**:: Menginisialisasi variabel i dengan nilai 3 dan variabel j dengan nilai 4. Perlu dicatat bahwa dalam Java, karakter juga dapat diwakili oleh kode ASCII.
9. **System.out.println("i = " +(int) i)**:: perintah untuk mencetak nilai dari variabel i ke konsol. Dalam hal ini, kita mengkonversi karakter i menjadi integer dengan (int) karena kita ingin mencetak nilai dalam kode ASCII.
10. **System.out.println("j = "+j)**:: Ini adalah perintah untuk mencetak nilai dari variabel j ke konsol.
11. **System.out.println("i & j = "+(i & j))**:: operasi bitwise AND (&) antara nilai variabel i dan j, kemudian hasilnya dicetak ke konsol. Operasi bitwise AND menghasilkan nilai yang sama dengan 1 hanya jika kedua bit yang sesuai dalam representasi biner dari kedua operand adalah 1.
12. **System.out.println("i | j = "+(i | j))**:: operasi bitwise OR (|) antara nilai variabel i dan j, kemudian hasilnya dicetak ke konsol. Operasi bitwise OR menghasilkan nilai yang

sama dengan 1 jika salah satu bit yang sesuai dalam representasi biner dari kedua operand adalah 1.

13. **System.out.println("i ^ j = " + (i^j));**: operasi bitwise XOR (^) antara nilai variabel i dan j, kemudian hasilnya dicetak ke konsol. Operasi bitwise XOR menghasilkan nilai yang sama dengan 1 hanya jika satu dari dua bit yang sesuai dalam representasi biner dari kedua operand adalah 1, tetapi tidak keduanya.
14. **System.out.println(Math.pow(i,j));**: perintah untuk mencetak hasil dari Math.pow(i,j) ke konsol, yang merupakan pangkat i dengan basis j.
15. **System.out.println("~i = " + ~i);**: operasi bitwise NOT (~) pada nilai variabel i, kemudian hasilnya dicetak ke konsol. Operasi bitwise NOT mengubah setiap bit dalam operand menjadi kebalikannya.

Kasus 13

Output :

```
13. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java Oper3
true
false
true
true
true
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas Oper3 sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Oper3.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas Oper3. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**: Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **char i,j**;; Ini adalah deklarasi dua variabel i dan j dengan tipe data char (karakter).
8. **i = 3; j = 4**;; Menginisialisasi variabel i dengan nilai 3 dan variabel j dengan nilai 4. Perlu dicatat bahwa dalam Java, karakter juga dapat diwakili oleh kode ASCII.
9. **System.out.println("i = " +(int) i)**;; perintah untuk mencetak nilai dari variabel i ke konsol. Dalam hal ini, kita mengkonversi karakter i menjadi integer dengan (int) karena kita ingin mencetak nilai dalam kode ASCII.
10. **System.out.println("j = "+j)**;; Ini adalah perintah untuk mencetak nilai dari variabel j ke konsol.
11. **System.out.println("i & j = "+(i & j))**;; operasi bitwise AND (&) antara nilai variabel i dan j, kemudian hasilnya dicetak ke konsol. Operasi bitwise AND menghasilkan nilai yang sama dengan 1 hanya jika kedua bit yang sesuai dalam representasi biner dari kedua operand adalah 1.
12. **System.out.println("i | j = "+(i | j))**;; operasi bitwise OR (|) antara nilai variabel i dan j, kemudian hasilnya dicetak ke konsol. Operasi bitwise OR menghasilkan nilai yang sama dengan 1 jika salah satu bit yang sesuai dalam representasi biner dari kedua operand adalah 1.

13. **System.out.println("i ^ j = " + (i^j));**: operasi bitwise XOR (^) antara nilai variabel i dan j, kemudian hasilnya dicetak ke konsol. Operasi bitwise XOR menghasilkan nilai yang sama dengan 1 hanya jika satu dari dua bit yang sesuai dalam representasi biner dari kedua operand adalah 1, tetapi tidak keduanya.
14. **System.out.println(Math.pow(i,j));**: perintah untuk mencetak hasil dari Math.pow(i,j) ke konsol, yang merupakan pangkat i dengan basis j.
15. **System.out.println("~i = " + ~i);**: operasi bitwise NOT (~) pada nilai variabel i, kemudian hasilnya dicetak ke konsol. Operasi bitwise NOT mengubah setiap bit dalam operand menjadi kebalikannya.

Kasus 14

Output:

```
14. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java Oper4
    Nilai e = 10
    Nilai k = 0
    Nilai k = 4
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas Oper4 sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah Oper4.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas Oper4. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**:. Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **int i = 0; int j = 0;**: deklarasi dan inisialisasi dua variabel i dan j dengan tipe data integer, masing-masing diberi nilai awal 0.
8. **char c = 8; char d = 10;**: deklarasi dan inisialisasi dua variabel c dan d dengan tipe data char (karakter), masing-masing diberi nilai ASCII 8 dan 10.
9. **int e = (((int)c > (int)d) ? c : d);**: operasi ternary (?:) yang mengevaluasi apakah nilai dari variabel c lebih besar dari nilai variabel d. Jika benar, maka nilai e akan diisi dengan nilai c, jika tidak, maka nilai e akan diisi dengan nilai d.
10. **int k = ((i > j) ? i : j);**: operasi ternary yang mengevaluasi apakah nilai dari variabel i lebih besar dari nilai variabel j. Jika benar, maka nilai k akan diisi dengan nilai i, jika tidak, maka nilai k akan diisi dengan nilai j.
11. **System.out.print("Nilai e = " +e);**: perintah untuk mencetak nilai dari variabel e ke konsol.
12. **System.out.print("\nNilai k = "+k);**: perintah untuk mencetak nilai dari variabel k ke konsol.
13. **i = 2; j = 3;**: Mengubah nilai dari variabel i menjadi 2 dan variabel j menjadi 3.
14. **k = ((i ++ > j++) ? i : j);**: operasi ternary yang mengevaluasi apakah nilai dari variabel i yang sudah ditambahkan 1 lebih besar dari nilai variabel j yang juga sudah ditambahkan 1. Jika benar, maka nilai k akan diisi dengan nilai i, jika tidak, maka nilai

k akan diisi dengan nilai j. Perlu dicatat bahwa i++ dan j++ akan mengembalikan nilai sebelum penambahan, sehingga i dan j tidak akan bertambah sampai setelah pengevaluasian.

15. **System.out.print ("\nNilai k = "+k);** Ini adalah perintah untuk mencetak nilai dari variabel k ke konsol setelah operasi ternary yang kedua.

Kasus 15

Output :

```
15 . E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java PrintHelo
Hello
Hello Worldwelcome
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas PrintHelo sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah PrintHelo.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas PrintHelo. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**: Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **System.out.print("Hello");**: perintah untuk mencetak teks "Hello" ke konsol tanpa karakter baris baru.
8. **System.out.print("\nHello ");**: perintah untuk mencetak teks "Hello" ke konsol dengan karakter baris baru (\n) di awalnya, sehingga teks tersebut dicetak di baris baru.
9. **System.out.print("World ");**: perintah untuk mencetak teks "World" ke konsol tanpa karakter baris baru.
10. **System.out.print("welcome");**: perintah untuk mencetak teks "welcome" ke konsol tanpa karakter baris baru.

Kasus 16

Output :

```
16. E:\SEMESTER 4\PBO\Tugas 1\SourceCode>java Oprator
Silahkan Baca Teksnya Dan Tambahkan Program Di Bawah Ini Untuk Menampilkan Output Program
Bool1 || Bool2 = true
!Bool1 = false
Bool1 ^ Bool2 = true
i + j = 7
i - j = 3
i / j = 2
i * j = 10
i / j = 2
i % j = 1
x + y = 10.0
x - y = 0.0
x / y = 1.0
x * y = 25.0
i == j = false
i != j = true
i < j = false
i > j = true
i <= j = false
i >= j = true
x != y = false
x < y = false
x > y = false
x <= y = true
x >= y = true
```

Penjelasan Program yang di gunakan dari variabel/tipe data/kelas :

1. **public**: digunakan untuk mendeklarasikan kelas PrintHelo sebagai public, yang berarti kelas ini dapat diakses dari mana saja.
2. **class**: Kata kunci class digunakan untuk mendefinisikan sebuah kelas dalam Java. Dalam kasus ini, kelas yang didefinisikan adalah PrintHelo.
3. **static**: membuat metode main menjadi statis. Ini berarti metode main dapat dipanggil tanpa membuat objek dari kelas PrintHelo. Metode main yang statis adalah metode utama yang dipanggil saat program dijalankan.
4. **void**: digunakan sebelum nama metode main. Dalam kasus ini, void menunjukkan bahwa metode main tidak mengembalikan nilai apapun.
5. **main**:. Metode main adalah titik awal eksekusi program Java. Saat program dijalankan, JDK (Java Development Kit) akan mencari metode main dan menjalankan kode di dalamnya.
6. **String[] args**: String[] menunjukkan bahwa parameter args adalah array dari objek String. Parameter ini digunakan untuk menerima argumen baris perintah yang diteruskan saat menjalankan program.
7. **boolean Bool1, Bool2, TF**:: deklarasi tiga variabel boolean Bool1, Bool2, dan TF yang digunakan untuk menyimpan nilai kebenaran dari operasi logika.
8. **int i, j, hsl**:: deklarasi tiga variabel integer i, j, dan hsl yang digunakan untuk operasi aritmatika.

9. **float x, y, res;**: deklarasi tiga variabel floating-point x, y, dan res yang juga digunakan untuk operasi aritmatika.
10. **System.out.println("Silahkan Baca Teksnya Dan Tambahkan Program Di Bawah Ini Untuk Menampilkan Output Program");**: perintah untuk mencetak teks ke konsol.
11. **Bool1 = true; Bool2 = false; TF = Bool1 && Bool2;**: operasi logika AND (&&) antara Bool1 dan Bool2, kemudian hasilnya disimpan dalam variabel TF.
12. **TF = Bool1 || Bool2;**: operasi logika OR (||) antara Bool1 dan Bool2, kemudian hasilnya disimpan dalam variabel TF.
13. **TF = !Bool1;**: operasi logika NOT (!) pada Bool1, kemudian hasilnya disimpan dalam variabel TF.
14. **TF = Bool1 ^ Bool2;**: operasi logika XOR (^) antara Bool1 dan Bool2, kemudian hasilnya disimpan dalam variabel TF.
15. **hsl = i + j; hsl = i - j; hsl = i / j; hsl = i * j; hsl = i / j; hsl = i % j;**: operasi aritmatika antara i dan j, kemudian hasilnya disimpan dalam variabel hsl. Operasi yang dilakukan adalah penambahan (+), pengurangan (-), pembagian (/), perkalian (*), dan sisa pembagian (%).
16. **res = x + y; res = x - y; res = x / y; res = x * y;**: operasi aritmatika antara x dan y, kemudian hasilnya disimpan dalam variabel res.
17. **TF = (i == j); TF = (i != j); TF = (i < j); TF = (i > j); TF = (i <= j); TF = (i >= j);**: operasi perbandingan antara i dan j, kemudian hasilnya disimpan dalam variabel TF. Operasi yang dilakukan adalah kesetaraan (==), ketidaksamaan (!=), lebih kecil dari (<), lebih besar dari (>), kurang dari atau sama dengan (<=), dan lebih besar dari atau sama dengan (>=).
18. **TF = (x != y); TF = (x < y); TF = (x > y); TF = (x <= y); TF = (x >= y);**: operasi perbandingan antara x dan y, kemudian hasilnya disimpan dalam variabel TF.