

JAWABAN

1. Tugas Praktek : Praktek Program Java (terlampir)

1	<div data-bbox="300 371 1390 853"><div data-bbox="300 371 1002 853"><pre>1 // Ardiansyah 2 // 13020220091 3 // 25-05-2024 4 5 public class Asgd11 { 6 7 public static void main(String[] args) { 8 // Kamus 9 float f = 20.0f; 10 double f11; 11 12 // Algoritma 13 f11 = 10.0f; 14 System.out.println("f : " + f + "\nf11: " + f11); 15 16 // Menambahkan pencetakan nilai f11 17 System.out.println("f11: " + f11); 18 } 19 } 20</pre></div><div data-bbox="1002 371 1390 853"><pre>f : 20.0 f11: 10.0 f11: 10.0</pre></div></div>
	<pre>// Ardiansyah // 13020220091 // 25-05-2024 public class Asgd11 { public static void main(String[] args) { // Kamus float f = 20.0f; double f11; // Algoritma f11 = 10.0f; System.out.println("f : " + f + "\nf11: " + f11); // Menambahkan pencetakan nilai f11 System.out.println("f11: " + f11); } }</pre>

	<p>Program Java ini adalah contoh sederhana yang menunjukkan penggunaan variabel dan pencetakan nilai-nilai variabel tersebut. Berikut adalah penjelasan singkat mengenai setiap bagian dari program:</p> <ol style="list-style-type: none"> 1. <code>`public class Asgdll { }`</code>: Definisi kelas utama Java bernama <code>`Asgdll`</code>, yang akan menjadi titik awal eksekusi program. 2. <code>`public static void main(String[] args) { }`</code>: Metode utama (<code>`main method`</code>) dari kelas <code>`Asgdll`</code>, dieksekusi pertama kali saat program dijalankan. Parameter <code>`args`</code> adalah array string untuk menerima argumen dari baris perintah, namun tidak digunakan dalam contoh ini. 3. <code>`float f = 20.0f;`</code>: Mendeklarasikan variabel <code>`f`</code> dengan tipe data <code>`float`</code> dan nilai awal 20.0. Suffix <code>`f`</code> menunjukkan bahwa nilai tersebut adalah float. 4. <code>`double fl;`</code>: Mendeklarasikan variabel <code>`fl`</code> dengan tipe data <code>`double`</code> tanpa memberikan nilai awal. Secara default, nilai variabel <code>`fl`</code> adalah <code>`0.0`</code>. 5. <code>`fl = 10.0f;`</code>: Menginisialisasi variabel <code>`fl`</code> dengan nilai 10.0. Meskipun <code>`fl`</code> adalah tipe data <code>`double`</code>, ia menerima nilai dari tipe data <code>`float`</code> (dalam hal ini, 10.0f), karena tipe data <code>`float`</code> dapat di-promosikan menjadi <code>`double`</code>. 6. <code>`System.out.println("f : " + f + "\nfl: " + fl);`</code>: Mencetak nilai variabel <code>`f`</code> dan <code>`fl`</code> ke layar. <code>`System.out.println()`</code> digunakan untuk mencetak ke konsol. Ekspresi <code>`"f : " + f + "\nfl: " + fl`</code> menggabungkan string dengan nilai variabel <code>`f`</code> dan <code>`fl`</code>, lalu mencetaknya ke konsol. 7. <code>`System.out.println("fl: " + fl);`</code>: Mencetak nilai variabel <code>`fl`</code> ke layar. Ini menunjukkan cara nilai variabel dapat dicetak lebih dari satu kali dalam program. <p>Tujuan program ini adalah menunjukkan bagaimana variabel dideklarasikan, diinisialisasi, dan dicetak dalam bahasa pemrograman Java.</p>
--	---

2

```
1 |
2
3 // Ardiansyah
4 // 13020220091
5 // 25-05-2024
6 public class Asign {
7
8     public static void main(String[] args) {
9         // Kamus
10        int i;
11
12        // Program
13        System.out.print("hello\n");
14        i = 5;
15        System.out.println("Ini nilai i : " + i);
16    }
17 }
18
```

```
hello
Ini nilai i : 5
```

```
// Ardiansyah
// 13020220091
// 25-05-2024
public class Asign {

    public static void main(String[] args) {

        // Kamus
        int i;

        // Program
        System.out.print("hello\n");
        i = 5;
        System.out.println("Ini nilai i : " + i);
    }
}
```

Program tersebut adalah contoh sederhana yang bertujuan untuk menunjukkan bagaimana cara mendeklarasikan sebuah variabel, menginisialisasi nilai ke dalam variabel, dan mencetak nilai variabel tersebut ke layar.

Program tersebut adalah contoh sederhana yang bertujuan untuk menunjukkan cara mendeklarasikan sebuah variabel, menginisialisasi nilai ke dalam variabel, dan mencetak nilai variabel tersebut ke layar.

Dalam program ini, terdapat kelas yang dideklarasikan dengan nama "Asign". Metode utama dari kelas ini, yang dinamai "main", merupakan titik masuk utama

	<p>untuk eksekusi program. Parameter "args" dalam metode main digunakan untuk menerima argumen baris perintah.</p> <p>Variabel "i" dideklarasikan dengan tipe data int (integer), kemudian diinisialisasi dengan nilai 5. Kemudian, nilai variabel "i" dicetak ke layar menggunakan metode println dari kelas System, yang digabungkan dengan string "Ini nilai i : ". Perintah "System.out.print("hello\n")" juga digunakan untuk mencetak string "hello" ke konsol.</p>
3	 <pre> 1 // Ardiansyah 2 // 13020220091 3 // 25-05-2024 4 public class ASIGNi { 5 public static void main(String[] args) { 6 // KAMUS 7 short ks = 1; 8 int ki = 1; 9 long kl = 10000; 10 char c = 65; /* inisialisasi karakter dengan integer */ 11 char c1 = 'Z'; /* inisialisasi karakter dengan karakter */ 12 double x = 50.2f; 13 float y = 50.2f; 14 // Algoritma 15 // penulisan karakter sebagai karakter 16 System.out.println("Karakter = " + c); 17 System.out.println("Karakter = " + c1); 18 // penulisan karakter sebagai integer 19 System.out.println("Karakter = " + (int) c); 20 System.out.println("Karakter = " + (int) c1); 21 System.out.println("Bilangan integer (short) = " + ks); 22 System.out.println("\t(int) = " + ki); 23 System.out.println("\t(long) = " + kl); 24 } 25 } </pre> <pre> Karakter = A Karakter = Z Karakter = 65 Karakter = 90 Bilangan integer (short) = 1 (int) = 1 (long) = 10000 Bilangan Real x = 50.20000076293945 Bilangan Real y = 50.2 </pre>
	<pre> // Ardiansyah // 13020220091 // 25-05-2024 public class ASIGNi { public static void main(String[] args) { // KAMUS short ks = 1; int ki = 1; long kl = 10000; char c = 65; /* inisialisasi karakter dengan integer */ char c1 = 'Z'; /* inisialisasi karakter dengan karakter */ double x = 50.2f; float y = 50.2f; // Algoritma // penulisan karakter sebagai karakter System.out.println("Karakter = " + c); System.out.println("Karakter = " + c1); } } </pre>

	<pre> // penulisan karakter sebagai integer System.out.println("Karakter = " + (int) c); System.out.println("Karakter = " + (int) c1); System.out.println("Bilangan integer (short) = " + ks); System.out.println("\t(int) = " + ki); System.out.println("\t(long)= " + kl); System.out.println("Bilangan Real x = " + x); System.out.println("Bilangan Real y = " + y); } } </pre>
	<p>Program ini bertujuan untuk menunjukkan contoh penggunaan dan inisialisasi variabel-variabel dengan berbagai tipe data di dalam bahasa pemrograman Java.</p> <p>Dalam program ini, kelas dideklarasikan dengan nama "ASIGNi", yang merupakan titik masuk utama untuk eksekusi program. Metode utama dari kelas ini dinamai "main", yang harus ada dalam setiap program Java dan digunakan untuk memulai eksekusi program. Parameter "args" dalam metode main biasanya digunakan untuk menerima argumen baris perintah.</p> <p>Selanjutnya, variabel-variabel dideklarasikan dan diinisialisasi dengan berbagai tipe data. Variabel-variabel dengan tipe data bilangan bulat (short, int, long) dideklarasikan dan diinisialisasi. Selain itu, variabel karakter (char) juga dideklarasikan dan diinisialisasi, di mana karakter 'Z' direpresentasikan oleh nilai ASCII-nya, yaitu 65. Variabel-variabel dengan tipe data bilangan riil (double dan float) juga dideklarasikan dan diinisialisasi.</p> <p>Nilai-nilai variabel kemudian dicetak ke konsol menggunakan metode println dari kelas System. Pada beberapa kasus, karakter escape sequence seperti "\t" digunakan untuk memberikan indentasi pada output yang dihasilkan.</p>

4

```
1 // Created by Neha Singhania
2
3 // Ardiansyah
4 // 13020220091
5 // 25-05-2024
6 import java.util.Scanner;
7
8 /* contoh membaca integer menggunakan Class Scanner */
9 public class BacaData {
10     /**
11      * @param args
12      */
13     public static void main(String[] args) {
14         // Kamus
15         int a;
16         Scanner masukan;
17
18         // Program
19         System.out.print("Contoh membaca dan menulis, ketik nilai integer: \n");
20         masukan = new Scanner(System.in);
21         a = masukan.nextInt(); /* coba ketik: masukan.nextInt(); Apa akibatnya?
22         System.out.print("Nilai yang dibaca: " + a);
23     }
24 }
25
```

Contoh membaca dan menulis, ketik nilai integer:
Nilai yang dibaca: 89

```
// Ardiansyah
// 13020220091
// 25-05-2024
import java.util.Scanner;

/* contoh membaca integer menggunakan Class Scanner */
public class BacaData {

    /**
     * @param args
     */

    public static void main(String[] args) {

        // Kamus
        int a;
        Scanner masukan;

        // Program
        System.out.print("Contoh membaca dan menulis, ketik nilai
integer: \n");
        masukan = new Scanner(System.in);
        a = masukan.nextInt(); /* coba ketik: masukan.nextInt(); Apa
akibatnya? */
        System.out.print("Nilai yang dibaca: " + a);
    }
}
```

Program ini bertujuan untuk mendemonstrasikan penggunaan kelas Scanner untuk membaca input dari pengguna dan kemudian mencetak nilai yang dibaca ke konsol.

Pada awalnya, dilakukan impor kelas Scanner dari paket java.util agar dapat digunakan dalam program. Selanjutnya, sebuah kelas dideklarasikan dengan nama "BacaData", yang akan menjadi tempat utama untuk eksekusi program.

Metode utama dari kelas ini dinamai "main", yang merupakan titik masuk utama untuk eksekusi program dalam Java. Parameter "args" diterima oleh metode main, yang umumnya digunakan untuk menerima argumen baris perintah.

Variabel "a" dideklarasikan dengan tipe data int (integer), dan sebuah objek Scanner dideklarasikan untuk membaca input dari pengguna. Pesan instruksi kemudian dicetak ke konsol, meminta pengguna untuk memasukkan nilai integer.

Setelah itu, objek Scanner dibuat dan dihubungkan ke aliran input standar (System.in), yang memungkinkan program untuk membaca input dari keyboard pengguna. Nilai integer yang dimasukkan oleh pengguna kemudian dibaca dan disimpan dalam variabel "a" menggunakan metode nextInt() dari objek Scanner.

Akhirnya, nilai yang dibaca dari pengguna dicetak ke konsol untuk ditampilkan kepada pengguna.

5

Main.java

```

1 // Ardiansyah
2 // 13020220091
3 // 25-05-2024
4
5 import java.io.BufferedReader;
6 import java.io.IOException;
7 import java.io.InputStreamReader;
8 import java.util.Scanner;
9
10 public class Bacakar {
11     /**
12      * @param args
13      * @throws IOException
14      */
15     public static void main(String[] args) throws IOException {
16         // Kamus
17         char cc;
18         int bil;
19         InputStreamReader isr = new InputStreamReader(System.in);
20         BufferedReader dataIn = new BufferedReader(isr);

```

Run

Output

```

java -cp /tmp/V1c2oRbG1p Bacakar
hello
baca 1 karakter : A
baca 1 bilangan : 10
Masukkan teks: HYU
Teks yang dimasukkan: HYU
A
10
bye

```

```

// Ardiansyah
// 13020220091

```

```
// 25-05-2024

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

public class Bacakar {
    /**
     * @param args
     * @throws IOException
     */
    public static void main(String[] args) throws IOException {
        // Kamus
        char cc;
        int bil;
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader dataIn = new BufferedReader(isr);
        // atau
        BufferedReader dataInput = new BufferedReader(new
        InputStreamReader(System.in));

        // Algoritma
        System.out.print("hello\n");
        System.out.print("baca 1 karakter : ");
        // perintah baca karakter cc
        cc = dataIn.readLine().charAt(0);
        System.out.print("baca 1 bilangan : ");
        // perintah baca bil
        bil = Integer.parseInt(dataInput.readLine());
        /* String kar = JOptionPane.showInputDialog("Karakter 1 : ");
        * System.out.println(kar); */
        // JOptionPane.showMessageDialog(null, "hello");
        System.out.print(cc + "\n" + bil + "\n");
        System.out.print("bye \n");
        /* Tambahkan program membaca/input data
        menggunakan Class Scanner, Class Console dan Class
        JOptionPane */
    }
}
```



```
}
```

Program ini bertujuan untuk membaca input karakter dan bilangan dari pengguna dan kemudian mencetaknya ke konsol.

Pada awalnya, dilakukan impor kelas-kelas yang diperlukan seperti `BufferedReader`, `IOException`, dan `InputStreamReader` untuk melakukan operasi pembacaan input dari pengguna.

Kelas utama dideklarasikan dengan nama "Bacakar". Metode utama dari kelas ini dinamai "main" dan melemparkan `IOException` karena operasi pembacaan input dapat menyebabkan pengecualian.

Variabel `cc` dan `bil` dideklarasikan untuk menyimpan input karakter dan bilangan yang dimasukkan oleh pengguna.

Selanjutnya, objek `InputStreamReader` dan `BufferedReader` dibuat untuk membaca input karakter dari pengguna. Objek `BufferedReader` dapat dibuat langsung terhubung ke aliran input standar (`System.in`).

Pesan "hello" dicetak ke konsol untuk memberi sambutan kepada pengguna. Kemudian, karakter pertama dari baris input yang dimasukkan oleh pengguna dibaca menggunakan `BufferedReader`, sedangkan bilangan bulat diambil dengan menggunakan metode `parseInt` dari kelas `Integer`.

Input karakter dan bilangan yang dimasukkan oleh pengguna kemudian dicetak ke konsol. Pesan "bye" juga dicetak sebagai tanda akhir dari program.

Program ini bisa diperluas dengan menambahkan kemampuan membaca input menggunakan kelas `Scanner`, `Console`, atau `JOptionPane`, seperti yang dicatat dalam komentar terakhir.

6

Main.java



Run

Output

```

1 // Ardiansyah
2 // 13020220091
3 // 25-05-2024
4 public class Casting1 {
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int a = 5, b = 6;
8         float d = 2.0f, e = 3.2f;
9         char g = '5';
10        double k = 3.14;
11        System.out.println((float) a); // int <-- float
12        System.out.println((double) b); // int <-- double
13        System.out.println((int) d); // float <-- int
14        System.out.println((double) e); // float <-- double
15        System.out.println((int) g); // char <-- int (ASCII)
16        System.out.println((float) g); // char <-- float (ASCII)
17        System.out.println((double) g); // char <-- double (ASCII)
18        System.out.println((int) k); // double <-- int
19        System.out.println((float) k); // double <-- float
20    }

```

```

java -cp /tmp/V1c2oRbGlp Casting1
5.0
6.0
2
3.200000047683716
53
53.0
53.0
3
3.14

```

```

// Ardiansyah
// 13020220091
// 25-05-2024

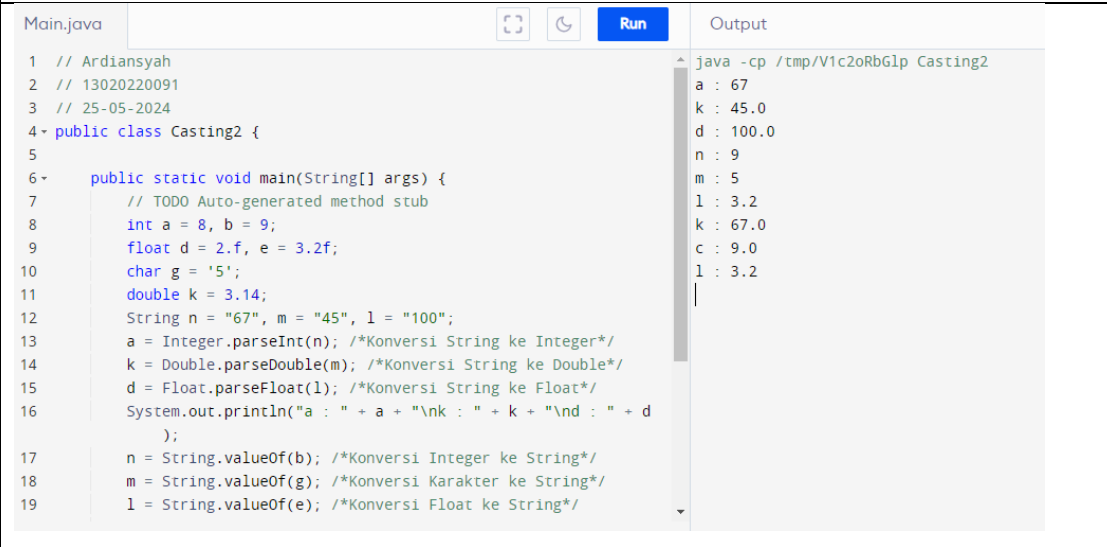
public class Casting1 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int a = 5, b = 6;
        float d = 2.0f, e = 3.2f;
        char g = '5';
        double k = 3.14;

        System.out.println((float) a); // int <-- float
        System.out.println((double) b); // int <-- double
        System.out.println((int) d); // float <-- int
        System.out.println((double) e); // float <-- double
        System.out.println((int) g); // char <-- int (ASCII)
        System.out.println((float) g); // char <-- float (ASCII)
        System.out.println((double) g); // char <-- double (ASCII)
        System.out.println((int) k); // double <-- int
        System.out.println((float) k); // double <-- float
    }
}

```

Penjelasan tiap bagian dari program tersebut adalah sebagai berikut:

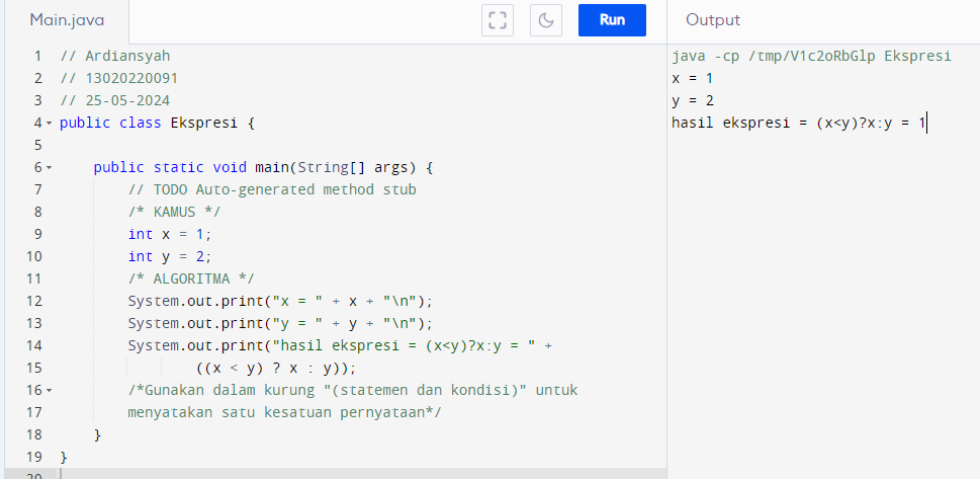
	<p>Kelas yang dideklarasikan dinamai "Casting1", yang merupakan titik masuk utama untuk eksekusi program Java. Dalam metode utama (main method) dari kelas ini, beberapa variabel dideklarasikan dan diinisialisasi dengan nilai awal. Variabel-variabel tersebut meliputi dua variabel bertipe data int (a dan b), dua variabel bertipe data float (d dan e), satu variabel bertipe data char (g), dan satu variabel bertipe data double (k).</p> <p>Selanjutnya, dilakukan proses casting antara tipe data primitif dalam Java. Proses casting dilakukan untuk mengubah tipe data suatu variabel menjadi tipe data lainnya sesuai kebutuhan. Misalnya, nilai variabel a yang bertipe int dicasting menjadi tipe float, kemudian dicetak ke konsol. Hal yang sama dilakukan untuk variabel b, d, e, g, dan k dengan berbagai jenis casting yang sesuai.</p> <p>Tujuan dari program ini adalah untuk memperlihatkan bagaimana proses casting dilakukan antara tipe data primitif dalam Java, serta dampaknya terhadap nilai variabel yang dicasting. Dengan demikian, program ini membantu untuk memahami konsep casting dalam bahasa pemrograman Java.</p>
7	 <pre> Main.java 1 // Ardiansyah 2 // 13020220091 3 // 25-05-2024 4 public class Casting2 { 5 6 public static void main(String[] args) { 7 // TODO Auto-generated method stub 8 int a = 8, b = 9; 9 float d = 2.5f, e = 3.2f; 10 char g = 'S'; 11 double k = 3.14; 12 String n = "67", m = "45", l = "100"; 13 a = Integer.parseInt(n); /*Konversi String ke Integer*/ 14 k = Double.parseDouble(m); /*Konversi String ke Double*/ 15 d = Float.parseFloat(l); /*Konversi String ke Float*/ 16 System.out.println("a : " + a + "\nk : " + k + "\nd : " + d 17); 18 n = String.valueOf(b); /*Konversi Integer ke String*/ 19 m = String.valueOf(g); /*Konversi Karakter ke String*/ 20 l = String.valueOf(e); /*Konversi Float ke String*/ 21 } 22 } </pre> <pre> Output java -cp /tmp/V1c2oRbGlp 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 </pre>
	<pre> // Ardiansyah // 13020220091 // 25-05-2024 public class Casting2 { public static void main(String[] args) { </pre>

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

Program ini bertujuan untuk mendemonstrasikan proses casting (pengubahan tipe data) antara tipe data primitif dan tipe data kelas (Class) dalam bahasa pemrograman Java.

Dalam program ini, sebuah kelas dideklarasikan dengan nama "Casting2", yang akan menjadi titik masuk utama untuk eksekusi program Java. Metode utama dari kelas ini dinamai "main" dan digunakan untuk menginisialisasi variabel-variabel serta melakukan proses casting di dalamnya.

Beberapa variabel dideklarasikan dan diinisialisasi dengan nilai awal, termasuk variabel bertipe data int (a dan b), float (d dan e), char (g), double (k), dan String (n, m, l).

	<p>Proses casting dilakukan antara berbagai tipe data, seperti dari String ke int (melalui metode <code>parseInt()</code>), dari String ke double (melalui metode <code>parseDouble()</code>), dan dari String ke float (melalui metode <code>parseFloat()</code>). Selain itu, juga dilakukan casting dari int ke double (melalui metode <code>valueOf()</code> dan <code>doubleValue()</code>), serta dari int ke float (melalui metode <code>valueOf()</code> dan <code>intValue()</code>).</p> <p>Hasil dari proses casting kemudian dicetak ke konsol untuk diperiksa. Ini membantu memperlihatkan bagaimana proses casting dilakukan antara tipe data primitif dan tipe data kelas dalam Java, serta dampaknya terhadap nilai variabel yang dicasting. Dengan demikian, tujuan dari program ini adalah untuk memperlihatkan konsep casting dalam bahasa pemrograman Java secara praktis.</p>
8	 <p>The screenshot shows an IDE with a file named <code>Main.java</code>. The code defines a class <code>Ekspresi</code> with a <code>main</code> method. It declares two integer variables, <code>x</code> and <code>y</code>, with values 1 and 2 respectively. It then prints the values of <code>x</code> and <code>y</code>. Finally, it prints the result of a conditional expression: <code>(x < y) ? x : y</code>, which evaluates to 1. The terminal output on the right shows the command <code>java -cp /tmp/V1c2oRbG1p Ekspresi</code> and the corresponding output: <code>x = 1</code>, <code>y = 2</code>, and <code>hasil ekspresi = (x<y)?x:y = 1</code>.</p>
	<pre>// Ardiansyah // 13020220091 // 25-05-2024 public class Ekspresi { public static void main(String[] args) { // TODO Auto-generated method stub /* KAMUS */ int x = 1; int y = 2; /* ALGORITMA */ System.out.print("x = " + x + "\n"); System.out.print("y = " + y + "\n"); System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y)); } }</pre>

	<pre> /*Gunakan dalam kurung "(statemen dan kondisi)" untuk menyatakan satu kesatuan pernyataan*/ } } </pre>
	<p>Program ini bertujuan untuk mendemonstrasikan penggunaan operator kondisional (ternary operator) dalam bahasa pemrograman Java.</p> <p>Dalam kelas "Ekspresi", metode utama ("main method") digunakan sebagai titik masuk utama untuk eksekusi program Java. Di dalam metode ini, dua variabel bertipe data int, yaitu x dan y, dideklarasikan dan diinisialisasi dengan nilai 1 dan 2 secara berturut-turut.</p> <p>Nilai dari kedua variabel tersebut kemudian dicetak ke konsol menggunakan perintah System.out.print(). Setelah itu, operator kondisional (ternary operator) digunakan untuk memeriksa apakah nilai x kurang dari nilai y. Jika kondisi tersebut benar, maka nilai x akan dicetak; jika tidak, nilai y yang akan dicetak.</p> <p>Penjelasan mengenai penggunaan operator kondisional tersebut diberikan sebagai berikut:</p> <ul style="list-style-type: none"> - `(x < y)` adalah ekspresi kondisional yang memeriksa apakah nilai x kurang dari nilai y. - `(x < y) ? x : y` adalah bentuk operator kondisional yang mengembalikan nilai x jika ekspresi benar, dan nilai y jika ekspresi salah. - `System.out.print("hasil ekspresi = (x<y)?x:y = " + ...)` mencetak hasil ekspresi kondisional ke konsol. <p>Tujuan dari program ini adalah untuk memberikan contoh penggunaan operator kondisional dalam Java, yang digunakan untuk membuat keputusan berdasarkan kondisi yang diberikan. Dengan demikian, program ini membantu dalam memahami konsep operator kondisional dalam pemrograman Java.</p>

9

Main.java	Output
<pre> 1 // Ardiansyah 2 // 13020220091 3 // 25-05-2024 4 public class Ekspresi1 { 5 public static void main(String[] args) { 6 // TODO Auto-generated method stub 7 /* KAMUS */ 8 int x = 1; 9 int y = 2; 10 float fx; 11 float fy; 12 /* ALGORITMA */ 13 System.out.print("x/y (format integer) = " + x / y); 14 System.out.print("\nx/y (format float) = " + (float) x / y); 15 /* supaya hasilnya tidak nol */ 16 fx = x; 17 fy = y; 18 System.out.print("\nx/y (format integer) = " + fx / fy); 19 System.out.print("\nx/y (format float) = " + fx / fy); 20 /* casting */ </pre>	<pre> java -cp /tmp/V1c2oRbG1p Ekspresi1 x/y (format integer) = 0 x/y (format float) = 0.5 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) = 3.3333333 </pre>

```

// Ardiansyah
// 13020220091
// 25-05-2024

public class Ekspresi1 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        /* KAMUS */

        int x = 1;
        int y = 2;
        float fx;
        float fy;

        /* ALGORITMA */

        System.out.print("x/y (format integer) = " + x / y);
        System.out.print("\nx/y (format float) = " + (float) x / y);
        /* supaya hasilnya tidak nol */

        fx = x;
        fy = y;

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

        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);

        x = 10;
        y = 3;

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

```

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

Program ini bertujuan untuk mendemonstrasikan operasi pembagian integer dan penggunaan casting dalam bahasa pemrograman Java.

Dalam kelas "Ekspresi1", terdapat metode utama ("main method") yang merupakan titik masuk untuk eksekusi program Java. Pada bagian ini, dua variabel bertipe data int, yaitu x dan y, dideklarasikan dan diinisialisasi dengan nilai awal 1 dan 2. Selain itu, terdapat dua variabel bertipe data float, yaitu fx dan fy.

Proses pembagian antara nilai variabel x dan y dilakukan dengan menggunakan operator pembagian ("/"). Hasil dari operasi ini dicetak ke konsol menggunakan perintah System.out.print(). Pada langkah-langkah tertentu, dilakukan casting nilai variabel x menjadi float sebelum melakukan pembagian, sehingga hasilnya akan berupa float.

Pada langkah selanjutnya, nilai dari variabel x dan y diubah menjadi 10 dan 3. Kemudian, proses pembagian kembali dilakukan dan hasilnya dicetak ke konsol.

Tujuan dari program ini adalah untuk memberikan contoh penggunaan operasi pembagian integer dan casting dalam Java, serta untuk menunjukkan perbedaan hasil pembagian antara operasi dengan operand integer dan float. Dengan demikian, program ini membantu untuk memahami cara kerja operasi pembagian dan penggunaan casting dalam bahasa pemrograman Java.

10

Main.java	Output
<pre>1 // Ardiansyah 2 // 13020220091 3 // 25-05-2024 4 public class PrintHello { 5 6 public static void main(String[] args) { 7 // TODO Auto-generated method stub 8 /* menuliskan hello ke layar */ 9 System.out.print("Hello"); 10 /* menuliskan hello dan ganti baris*/ 11 System.out.print("\nHello "); 12 /* menuliskan hello dan ganti baris*/ 13 System.out.println("World"); 14 System.out.println("Welcome"); 15 } 16 } 17</pre>	<pre>java -cp /tmp/V1c2oRbGlp PrintHello Hello Hello World Welcome </pre>

```
// Ardiansyah
// 13020220091
// 25-05-2024


public class PrintHello {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* menuliskan hello ke layar */
        System.out.print("Hello");
        /* menuliskan hello dan ganti baris*/
        System.out.print("\nHello ");
        /* menuliskan hello dan ganti baris*/
        System.out.println("World");
        System.out.println("Welcome");
    }
}
```

Program ini memiliki tujuan sederhana untuk mencetak beberapa teks ke layar. Berikut adalah penjelasan setiap bagian dari program tersebut:

Deklarasi kelas "PrintHello" mengawali program. Kemudian, metode utama ("main method") dari kelas ditentukan sebagai titik masuk untuk eksekusi program Java.

Komentar "// TODO Auto-generated method stub" di bawah metode utama adalah hasil dari IDE (Integrated Development Environment) seperti Eclipse. Komentar ini

	<p>menunjukkan bahwa kode di bawahnya adalah stub atau kode sementara yang dihasilkan secara otomatis.</p> <p>Selanjutnya, terdapat beberapa perintah cetak menggunakan <code>System.out.print()</code> dan <code>System.out.println()</code>. Perintah pertama mencetak teks "Hello" tanpa mengganti baris, sedangkan perintah kedua mencetak teks "Hello" dengan mengganti baris setelahnya. Perintah ketiga mencetak teks "World" dengan penambahan baris baru setelahnya menggunakan <code>"println"</code>. Sedangkan perintah terakhir mencetak teks "Welcome" dengan penambahan baris baru setelahnya.</p> <p>Tujuan dari program ini adalah untuk menunjukkan cara menggunakan metode <code>"print"</code> dan <code>"println"</code> untuk mencetak teks ke layar dalam bahasa pemrograman Java. Program ini juga memperlihatkan perbedaan antara <code>"print"</code> dan <code>"println"</code> dalam hal penambahan baris baru setelah mencetak teks, sehingga memberikan pemahaman yang lebih baik tentang fungsi-fungsi cetak dalam Java.</p>
11	 <pre> Main.java 1 // Ardiansyah 2 // 13020220091 3 // 25-05-2024 4 /* Efek dari operator ++ */ 5 public class Incr { 6 public static void main(String[] args) { 7 // Kamus 8 int i, j; 9 // Program 10 i = 3; 11 j = i++; 12 System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j); 13 } 14 } 15 Output java -cp /tmp/V1c2oRbG1p Incr Nilai i : 5 Nilai j : 3 </pre>
	<pre> // Ardiansyah // 13020220091 // 25-05-2024 /* Efek dari operator ++ */ public class Incr { public static void main(String[] args) { // Kamus int i, j; // Program i = 3; </pre>

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

Program ini memiliki tujuan untuk mengilustrasikan efek dari operator ++ pada variabel. Operator ++ digunakan untuk menambahkan nilai variabel dengan 1. Terdapat dua bentuk operator ++, yaitu pre-increment (++i) dan post-increment (i++).

Berikut adalah penjelasan dari program tersebut:

1. Pada bagian keyword yang digunakan, terdapat beberapa keyword seperti `public`, `class`, `static`, `void`, `main`, dan `String[] args`. Ini adalah bagian penting dalam deklarasi kelas dan metode utama yang akan dieksekusi saat program dijalankan.

2. Variabel yang digunakan dalam program ini adalah `i` dan `j`, keduanya bertipe int untuk menyimpan nilai integer.

3. Tujuan utama dari program ini adalah untuk menunjukkan perbedaan antara pre-increment (++i) dan post-increment (i++). Pada baris `j = i++`, nilai dari `i` disalin ke `j` terlebih dahulu, kemudian `i` ditambahkan 1. Sedangkan pada baris `System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j);`, pre-increment (++i) digunakan, sehingga nilai `i` akan diincrement terlebih dahulu sebelum digunakan dalam operasi penjumlahan dengan 1, sedangkan nilai `j` masih menyimpan nilai `i` sebelum diincrement.

Dengan demikian, program ini memberikan pemahaman yang jelas tentang penggunaan operator ++ dan perbedaan antara pre-increment dan post-increment dalam Java.

12

Main.java		Run	Output
1	// Ardiansyah		java -cp /tmp/V1c2oRbGlp Oper1
2	// 13020220091		n = 10
3	// 25-05-2024		x = 1
4	public class Oper1 {		y = 2
5	public static void main(String[] args) {		n & 8 = 8
6	// KAMUS		x & ~8 = 1
7	int n = 10; /* 1010 */		y << 2 = 8
8	int x = 1; /* 1 */		y >> 3 = 0
9	int y = 2; /* 10 */		
10	// ALGORITMA		
11	System.out.println("n = " + n);		
12	System.out.println("x = " + x);		
13	System.out.println("y = " + y);		
14	System.out.println("n & 8 = " + (n & 8)); /* 1010 AND 1000		
15	*/		
16	System.out.println("x & ~8 = " + (x & ~8)); /* 1 AND 0111 */		
17	System.out.println("y << 2 = " + (y << 2)); /* 10 ==> 1000 = 8		
	*/		
	System.out.println("y >> 3 = " + (y >> 3)); /* 10 ==> 0000 = 0		
	*/		
	}		
	}		

```
// Ardiansyah
// 13020220091
// 25-05-2024

public class Oper1 {
    public static void main(String[] args) {
        // KAMUS
        int n = 10; /* 1010 */
        int x = 1; /* 1 */
        int y = 2; /* 10 */

        // ALGORITMA
        System.out.println("n = " + n);
        System.out.println("x = " + x);
        System.out.println("y = " + y);
        System.out.println("n & 8 = " + (n & 8)); /* 1010 AND 1000 */
        System.out.println("x & ~8 = " + (x & ~8)); /* 1 AND 0111 */
        System.out.println("y << 2 = " + (y << 2)); /* 10 ==> 1000 = 8
        */

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

    }
}
```

Program ini bertujuan untuk mengilustrasikan penggunaan beberapa operator terhadap bit pada variabel bertipe data integer. Operator-operator bit tersebut antara lain AND (&), OR (|), NOT (~), dan LEFT SHIFT (<<), RIGHT SHIFT (>>).

Berikut adalah penjelasan dari program tersebut:

1. Pada bagian keyword yang digunakan, terdapat beberapa keyword seperti `public`, `class`, `static`, `void`, `main`, dan `String[] args`. Ini adalah bagian penting dalam deklarasi kelas dan metode utama yang akan dieksekusi saat program dijalankan.
2. Variabel yang digunakan dalam program ini adalah `n`, `x`, dan `y`, yang semuanya bertipe int untuk menyimpan nilai integer.
3. Tujuan utama dari program ini adalah untuk menunjukkan hasil operasi operator bit seperti AND, OR, NOT, LEFT SHIFT, dan RIGHT SHIFT pada variabel bertipe data integer. Operator AND (&) digunakan untuk menghasilkan bit 1 pada posisi yang sama jika kedua bit yang dibandingkan adalah 1, sedangkan operator OR (|) menghasilkan bit 1 jika salah satu atau kedua bit yang dibandingkan adalah 1. Operator NOT (~) digunakan untuk membalikkan setiap bit pada operand. Operator LEFT SHIFT (<<) digunakan untuk menggeser bit ke kiri sebanyak n posisi, sedangkan operator RIGHT SHIFT (>>) menggeser bit ke kanan sebanyak n posisi.

Dengan demikian, program ini memberikan pemahaman yang jelas tentang penggunaan operator bit dalam Java dan bagaimana operator tersebut memanipulasi bit-bit dalam variabel bertipe data integer.

13

Main.java

Run

Output

```
1 // Ardiansyah
2 // 13020220091
3 // 25-05-2024
4 /* pemakaian beberapa operator terhadap RELATIONAL DAN bit */
5- public class Oper2 {
6-     public static void main(String[] args) {
7-         // KAMUS
8-         char i, j;
9-         // ALGORITMA
10         i = 3; /* 00000011 dalam biner */
11         j = 4; /* 00000100 dalam biner */
12         System.out.println("i = " + (int) i);
13         System.out.println("j = " + (int) j);
14         System.out.println("i & j = " + (i & j)); /* 0: 00000000
            dalam biner */
15         System.out.println("i | j = " + (i | j)); /* 7: 00000111
            biner */
16         System.out.println("i ^ j = " + (i ^ j)); /* 7: 00000111
            biner Ingat!!! operator "^" pada bahasa java bukan
            sebagai pangkat */
```

```
java -cp /tmp/V1c2oRbGlp Oper2
i = 3
j = 4
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
|
```

```
// Ardiansyah
// 13020220091
```

```
// 25-05-2024
/* pemakaian beberapa operator terhadap RELATIONAL DAN bit */
public class Oper2 {
    public static void main(String[] args) {
        // KAMUS
        char i, j;
        // ALGORITMA
        i = 3; /* 00000011 dalam biner */
        j = 4; /* 00000100 dalam biner */
        System.out.println("i = " + (int) i);
        System.out.println("j = " + (int) j);
        System.out.println("i & j = " + (i & j)); /* 0: 00000000 dalam biner */
        System.out.println("i | j = " + (i | j)); /* 7: 00000111 biner */
        System.out.println("i ^ j = " + (i ^ j)); /* 7: 00000111 biner
Ingat!!! operator "^" pada bahasa java bukan sebagai pangkat */
        System.out.println(Math.pow(i, j)); /* Class Math memiliki method pow(a,b) untuk pemangkatan */
        System.out.println("~i = " + ~i); /* -4: 11111100 biner */
    }
}
```

Program ini bertujuan untuk mengilustrasikan penggunaan beberapa operator terhadap tipe data karakter (char) dan operasi bitwise di Java. Operator-operator yang digunakan antara lain AND (&), OR (|), XOR (^), dan NOT (~).

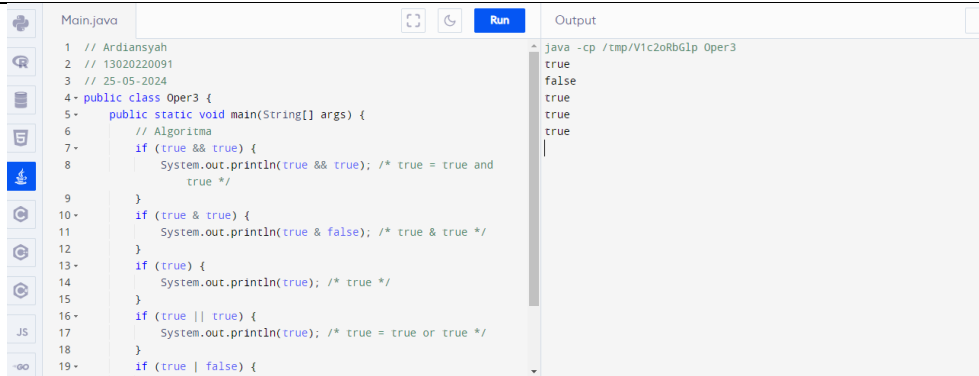
Berikut adalah penjelasan dari program tersebut:

1. Keyword yang digunakan:

- `public`: Mendefinisikan aksesibilitas kelas.
- `class`: Mendefinisikan sebuah kelas.
- `static`: Mendeklarasikan metode atau variabel sebagai statis yang berarti dapat diakses tanpa membuat instance dari kelas tersebut.
- `void`: Menunjukkan bahwa metode `main` tidak mengembalikan nilai apa pun.

	<ul style="list-style-type: none"> - <code>`main`</code>: Metode utama yang akan dieksekusi saat program dijalankan. - <code>`String[] args`</code>: Parameter dari metode <code>`main`</code>, yang merupakan array dari argumen baris perintah. <p>2. Variabel yang digunakan:</p> <ul style="list-style-type: none"> - <code>`i`, `j`</code>: Variabel bertipe <code>char</code> yang digunakan untuk menyimpan nilai karakter. <p>3. Tujuan program:</p> <p>Program ini bertujuan untuk menunjukkan penggunaan beberapa operator terhadap tipe data karakter dan operasi bitwise.</p> <ul style="list-style-type: none"> - Operator AND (<code>&</code>) digunakan untuk melakukan operasi bitwise AND antara dua bilangan biner. - Operator OR (<code> </code>) digunakan untuk melakukan operasi bitwise OR antara dua bilangan biner. - Operator XOR (<code>^</code>) digunakan untuk melakukan operasi bitwise XOR antara dua bilangan biner. - Operator NOT (<code>~</code>) digunakan untuk melakukan operasi bitwise NOT pada sebuah bilangan biner, yaitu membalik semua bitnya. - <code>`Math.pow(a, b)`</code>: Metode statis dari kelas <code>Math</code> yang digunakan untuk melakukan pemangkatan, di mana parameter pertama adalah basis dan parameter kedua adalah eksponen. Namun, penggunaan ini dalam program tidak sesuai dengan tujuan program yang sebenarnya, karena ini adalah operasi pemangkatan dan bukan operasi bitwise. <p>4. Output:</p> <ul style="list-style-type: none"> - Program mencetak nilai dari variabel <code>`i`</code> dan <code>`j`</code>. - Program kemudian mencetak hasil operasi bitwise AND, OR, XOR, dan NOT antara <code>`i`</code> dan <code>`j`</code>. - Terakhir, program mencetak hasil dari operasi pemangkatan menggunakan metode <code>`Math.pow`</code>, yang sebenarnya tidak sesuai dengan tujuan program.
--	---

14



The screenshot shows an IDE with a file named 'Main.java'. The code defines a class 'Oper3' with a 'main' method. Inside the 'main' method, there are four conditional blocks: 1. 'if (true && true)' which prints 'true'. 2. 'if (true & true)' which prints 'false'. 3. 'if (true)' which prints 'true'. 4. 'if (true || true)' which prints 'true'. The 'Output' window on the right shows the results of these prints: 'true', 'false', 'true', and 'true' on separate lines.

```
1 // Ardiansyah
2 // 13020220091
3 // 25-05-2024
4 public class Oper3 {
5     public static void main(String[] args) {
6         // Algoritma
7         if (true && true) {
8             System.out.println(true && true); /* true = true and
9             true */
10        }
11        if (true & true) {
12            System.out.println(true & false); /* true & true */
13        }
14        if (true) {
15            System.out.println(true); /* true */
16        }
17        if (true || true) {
18            System.out.println(true); /* true = true or true */
19        }
20        if (true | false) {
```

```
java -cp /tmp/Vlc2oRbGlp Oper3
true
false
true
true
```

```
// Ardiansyah
// 13020220091
// 25-05-2024

public class Oper3 {
    public static void main(String[] args) {
        // Algoritma
        if (true && true) {
            System.out.println(true && true); /* true = true and true
        */
        }
        if (true & true) {
            System.out.println(true & false); /* true & true */
        }
        if (true) {
            System.out.println(true); /* true */
        }
        if (true || true) {
            System.out.println(true); /* true = true or true */
        }
        if (true | false) {
            System.out.println(true | false); /* true|false */
        }
    }
}
```

Program ini bertujuan untuk mengilustrasikan perbedaan antara operator logika AND ('&&') dan operator bitwise AND ('&'), serta antara operator logika OR ('||') dan operator bitwise OR ('|') dalam bahasa pemrograman Java.

Berikut adalah penjelasan dari program tersebut:

1. Dalam bagian keyword yang digunakan, terdapat beberapa keyword seperti ``public``, ``class``, ``static``, ``void``, ``main``, dan ``String[] args``. Ini adalah bagian penting dalam deklarasi kelas dan metode utama yang akan dieksekusi saat program dijalankan.

2. Tujuan utama dari program ini adalah untuk menunjukkan perbedaan antara operator logika (``&&``, ``||``) dan operator bitwise (``&``, ``|``) dalam bahasa Java.

- Operator logika digunakan untuk mengevaluasi ekspresi logika dan menghasilkan nilai boolean (``true`` atau ``false``), sedangkan operator bitwise digunakan untuk melakukan operasi bitwise pada bilangan biner.


- Dalam ekspresi yang menggunakan operator logika (``&&``, ``||``), jika ekspresi pertama sudah cukup untuk menentukan hasil keseluruhan, maka ekspresi kedua tidak akan dievaluasi.

- Dalam ekspresi yang menggunakan operator bitwise (``&``, ``|``), kedua operand akan dievaluasi terlepas dari hasil operand pertama.

3. Output dari program adalah hasil dari beberapa ekspresi yang menggunakan operator logika dan operator bitwise. Hasil dari setiap ekspresi dicetak, yang mencakup nilai ``true`` atau ``false``.


Dengan demikian, program ini memberikan pemahaman yang jelas tentang perbedaan antara operator logika dan operator bitwise dalam Java serta cara mereka mempengaruhi evaluasi ekspresi logika.

15

Main.java	Run	Output
<pre>1 // Ardiansyah 2 // 13020220091 3 // 25-05-2024 4 /* Operator terner */ 5 public class Oper4 { 6 public static void main(String[] args) { 7 // KAMUS 8 int i = 0; /* perhatikan int i,j=0 bukan seperti ini */ 9 int j = 0; 10 char c = 8; 11 char d = 10; 12 int e = (((int)c > (int)d) ? c : d); 13 int k = ((i > j) ? i : j); 14 // ALGORITMA 15 System.out.print("Nilai e = " + e); 16 System.out.print("\nNilai k = " + k); 17 i = 2; 18 j = 3; 19 k = ((i++ > j++) ? i : j); 20 System.out.print("\nNilai k = " + k); 21 } 22 }</pre>		<pre>java -cp /tmp/V1c2oRbG1p Oper4 Nilai e = 10 Nilai k = 0 Nilai k = 4</pre>

```
// Ardiansyah
// 13020220091
// 25-05-2024
/* Operator terner */
public class Oper4 {
    public static void main(String[] args) {
        // KAMUS

        int i = 0; /* perhatikan int i,j=0 bukan seperti ini */
        int j = 0;
        char c = 8;
        char d = 10;
        int e = (((int)c > (int)d) ? c : d);
        int k = ((i > j) ? i : j);
        // ALGORITMA
        System.out.print("Nilai e = " + e);
        System.out.print("\nNilai k = " + k);
        i = 2;
        j = 3;
        k = ((i++ > j++) ? i : j);
        System.out.print("\nNilai k = " + k);
    }
}
```

	<p>Program ini mengilustrasikan penggunaan operator ternary ('?:') dalam bahasa pemrograman Java. Operator ternary memungkinkan kita untuk menulis kondisi if-else dalam satu baris kode, meningkatkan keterbacaan dan kekompakan kode.</p> <p>Dalam program ini, variabel `e` diinisialisasi dengan hasil ekspresi ternary yang mengevaluasi nilai ASCII dari karakter `c` dan `d`. Selain itu, variabel `k` diinisialisasi dengan hasil ekspresi ternary yang membandingkan nilai dari variabel `i` dan `j`. Setelah itu, nilai dari variabel `i` dan `j` ditingkatkan (increment). Hasil dari evaluasi ekspresi ternary untuk variabel `e` dan `k` dicetak sebagai output program.</p>
16	 <p>The screenshot shows a Java IDE with a file named 'Main.java'. The code defines a class 'Operator' with a 'main' method. It includes comments for the author 'Ardiansyah', ID '13020220091', and date '25-05-2024'. The code declares variables for 'Kamus' (Boolean, int, float) and 'Algoritma' (System.out.println). It sets 'Bool1' to true and 'Bool2' to false, then performs logical operations: 'TF = Bool1 && Bool2' (AND), 'TF = Bool1 Bool2' (OR), 'TF = !Bool1' (NOT), and 'TF = Bool1 ^ Bool2' (XOR). The output window shows the command 'java -cp ./tmp/V1c2oRb6lp Operator' and the message 'Silahkan baca teksnya dan tambahkan perintah program di bawah ini untuk menampilkan output program'.</p>
	<pre>// Ardiansyah // 13020220091 // 25-05-2024 /* Contoh pengoperasian variabel bertype dasar */ public class Operator { public static void main(String[] args) { // Kamus boolean Bool1, Bool2, TF; int i, j, hsl; float x, y, res; // Algoritma System.out.println("Silahkan baca teksnya dan tambahkan perintah program di bawah ini untuk menampilkan output program"); Bool1 = true; Bool2 = false; TF = Bool1 && Bool2; /* Boolean AND */ TF = Bool1 Bool2; /* Boolean OR */ TF = !Bool1; /* NOT */ TF = Bool1 ^ Bool2; /* XOR */ } }</pre>

```
TF = Bool1 ^ Bool2; /* XOR */
```

```
// Operasi numerik
```

```
i = 5;
```

```
j = 2;
```

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

```
hsl = i - j;
```

```
hsl = i / j;
```

```
hsl = i * j;
```

```
hsl = i / j; /* pembagian bulat */
```

```
hsl = i % j; /* sisa modulo */
```

```
// Operasi numerik
```

```
x = 5;
```

```
y = 5;
```

```
res = x + y;
```

```
res = x - y;
```

```
res = x / y;
```

```
res = x * y;
```

```
// Operasi relasional numerik
```

```
TF = (i == j);
```

```
TF = (i != j);
```

```
TF = (i < j);
```

```
TF = (i > j);
```

```
TF = (i <= j);
```

```
TF = (i >= j);
```

```
// Operasi relasional numerik
```

```
TF = (x != y);
```

```
TF = (x < y);
```

```
TF = (x > y);
```

```
TF = (x <= y);
```

```
TF = (x >= y);
```

```
}
```

```
}
```

	<p>Program ini merupakan contoh penggunaan variabel berjenis data dasar (primitive) dan operator-operator numerik serta relasional dalam bahasa pemrograman Java.</p> <p>Dalam program ini, beberapa tipe data dasar seperti `boolean`, `int`, dan `float` dideklarasikan dan beberapa operasi numerik dan relasional dilakukan. Misalnya, terdapat operasi penjumlahan, pengurangan, perkalian, pembagian, serta operasi modulo. Operasi-relasional numerik juga dilakukan menggunakan operator seperti `==`, `!=`, `<`, `>`, `<=`, dan `>=`.</p> <p>Meskipun program ini tidak mencetak output secara langsung ke konsol karena tidak ada perintah `System.out.println()` yang diaktifkan, namun hasil dari operasi tersebut dapat ditugaskan ke variabel boolean dan dapat dicetak dengan menambahkan perintah `System.out.println()`.</p>

2. Tugas Kasus : Buat Flowchat dan Class Diagram dari kasus di

bawah ini kemudian terjemahkan ke dalam program menggunakan

Bahasa Java.

```
// Ardiansyah
// 13020220091
// 25-05-2024
import java.util.Scanner;

public class KonversiWaktu {
    private int totalDetik;

    public KonversiWaktu() {
        // Constructor
    }

    public void setTotalDetik(int detik) {
        totalDetik = detik;
    }
}
```

```
public int hitungDetikSekarang() {
    int detikSekarang = totalDetik % 60;
    System.out.println("2. Hitung detikSekarang = " + detikSekarang);
    return detikSekarang;
}

public int hitungTotalMenit() {
    int totalMenit = totalDetik / 60;
    System.out.println("3. Hitung totalMenit = " + totalMenit);
    return totalMenit;
}

public int hitungMenitSekarang() {
    int menitSekarang = hitungTotalMenit() % 60;
    System.out.println("4. Hitung menitSekarang = " + menitSekarang);
    return menitSekarang;
}

public int hitungTotalJam() {
    int totalJam = hitungTotalMenit() / 60;
    System.out.println("5. Hitung totalJam = " + totalJam);
    return totalJam;
}

public int hitungJamSekarang() {
    int jamSekarang = hitungTotalJam() % 24;
    System.out.println("6. Hitung jamSekarang = " + jamSekarang);
    return jamSekarang;
}

public void tampilWaktu() {
    System.out.println("7. Tampil waktu (Jam:Menit:Detik)");
    System.out.println("Waktu sekarang: " + hitungJamSekarang() + ":" +
hitungMenitSekarang() + ":" + hitungDetikSekarang());
}
```

```

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Masukkan total detik: ");
        int totalDetik = scanner.nextInt();

        KonversiWaktu waktu = new KonversiWaktu();
        waktu.setTotalDetik(totalDetik);
        waktu.tampilWaktu();

        scanner.close();
    }
}

```

```

1 // Ardiansyah
2 // 13020220891
3
4 import java.util.Scanner;
5
6 public class KonversiWaktu {
7     private int totalDetik;
8
9     public KonversiWaktu() {
10         // Constructor
11     }
12
13     public void setTotalDetik(int detik) {
14         totalDetik = detik;
15     }
16
17     public int hitungDetikSekarang() {
18         int detikSekarang = totalDetik % 60;
19         System.out.println("2. Hitung detikSekarang = " + detikSekarang);
20         return detikSekarang;
21     }
22
23     public int hitungTotalMenit() {
24         int totalMenit = totalDetik / 60;
25
26         System.out.println("3. Hitung totalMenit = " + totalMenit);
27     }
28
29     public int hitungTotalJam() {
30         int totalJam = totalMenit / 60;
31         System.out.println("5. Hitung totalJam = " + totalJam);
32     }
33
34     public void tampilWaktu() {
35         System.out.println("6. Hitung jamSekarang = " + totalJam);
36         System.out.println("3. Hitung totalMenit = " + totalMenit);
37         System.out.println("4. Hitung menitSekarang = " + hitungDetikSekarang());
38         System.out.println("2. Hitung detikSekarang = " + hitungDetikSekarang());
39         System.out.println("Waktu sekarang: " + totalJam + ":" + totalMenit + ":" + hitungDetikSekarang());
40     }
41 }

```

Masukkan total detik: 7, Tampil waktu (Jam:Menit:Detik)

```

3. Hitung totalMenit = 20572560
5. Hitung totalJam = 342876
6. Hitung jamSekarang = 12
3. Hitung totalMenit = 20572560
4. Hitung menitSekarang = 0
2. Hitung detikSekarang = 45
Waktu sekarang: 12:0:45

```

KonversiWaktu
- totalDetik : int
+ KonversiWaktu : ()
+ setTotalDetik(detik: int): void
+ hitungDetikSekarang(): int
+ hitungTotalMenit(): int
+ hitungMenitSekarang(): int
+ hitungTotalJam(): int
+ hitungJamSekarang(): int
+ tampilWaktu(): void

