





JAWABAN

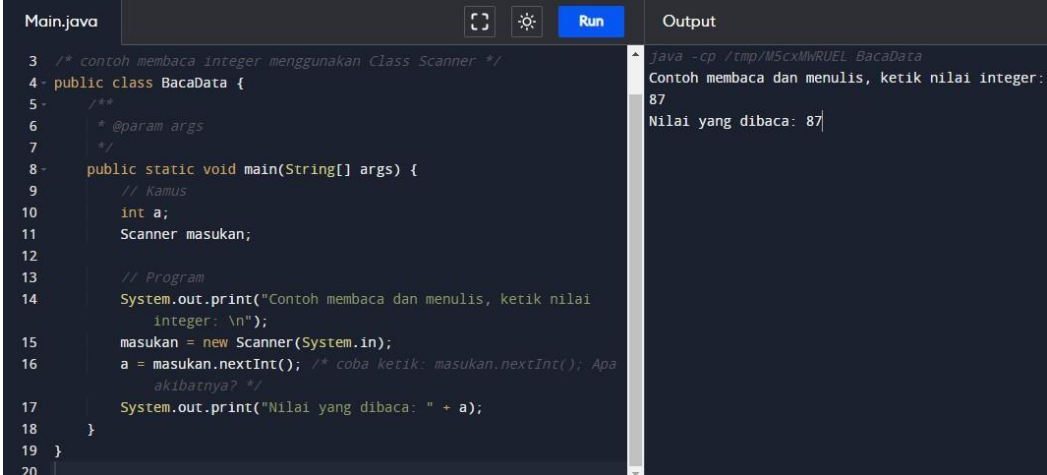
1. Tugas Praktek : Praktek Program Java (terlampir)

1	<div><div>Main.java</div><div><div></div><pre>1- public class Asgdll { 2- /** 3- * @param args 4- */ 5- public static void main(String[] args) { 6- // Kamus 7- float f = 20.0f; 8- double fll; 9- 10- // Algoritma 11- fll = 10.0f; 12- System.out.println("f : " + f + "\nfll: " + fll); 13- 14- // Menambahkan pencetakan nilai fll 15- System.out.println("fll: " + fll); 16- } 17- } 18</pre></div><div>Output</div><div>java -cp /tmp/MScxMwRUEL As f : 20.0 fll: 10.0 fll: 10.0</div></div>
	<pre>public class Asgdll { /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/ /** * @param args */ public static void main(String[] args) { // Kamus float f = 20.0f; double fll; // Algoritma fll = 10.0f; System.out.println("f : " + f + "\nfll: " + fll); // Menambahkan pencetakan nilai fll System.out.println("fll: " + fll); } }</pre>
	<p>Program ini adalah contoh sederhana dari sebuah program Java 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"><code>`public class Asgdll { }`</code>: Ini adalah definisi kelas Java yang diberi nama <code>`Asgdll`</code>. Kelas ini adalah kelas utama dari program dan akan menjadi titik awal eksekusi program.<code>`public static void main(String[] args) { }`</code>: Ini adalah metode utama (<code>`main method`</code>) dari kelas <code>`Asgdll`</code>. Metode ini akan dieksekusi pertama kali saat program dijalankan. Parameter <code>`args`</code> adalah array string yang dapat digunakan untuk menerima argumen dari baris perintah (command line arguments), tetapi dalam contoh ini tidak digunakan.

	<p>3. <code>float f = 20.0f;</code>: Mendeklarasikan sebuah variabel <code>f</code> dengan tipe data <code>float</code> dan memberikannya nilai awal 20.0. Suffix <code>f</code> menandakan bahwa nilai tersebut adalah float.</p> <p>4. <code>double fl;</code>: Mendeklarasikan sebuah variabel <code>fl</code> dengan tipe data <code>double</code> tanpa memberikan nilai awal. Secara default, nilai variabel <code>fl</code> akan menjadi <code>0.0</code>.</p> <p>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 bisa 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>.</p> <p>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>, kemudian mencetaknya ke konsol.</p> <p>7. <code>System.out.println("fl: " + fl);</code>: Mencetak nilai variabel <code>fl</code> ke layar. Ini menunjukkan bagaimana nilai variabel dapat dicetak lebih dari satu kali dalam program.</p> <p>Tujuan dari program ini adalah untuk menunjukkan bagaimana variabel dideklarasikan, diinisialisasi, dan dicetak dalam bahasa pemrograman Java.</p>
2	 <p>The screenshot shows an IDE with a file named <code>Main.java</code>. The code defines a class <code>Asign</code> with a <code>main</code> method. The output window shows the execution results: <code>hello</code> and <code>Ini nilai i : 5</code>.</p> <pre> 1 public class Asign { 2 /** 3 * @param args 4 */ 5 public static void main(String[] args) { 6 // Kamus 7 int i; 8 9 // Program 10 System.out.print("hello\n"); 11 i = 5; 12 System.out.println("Ini nilai i : " + i); 13 } 14 } 15 </pre> <p>Output:</p> <pre> java -cp /tmp/MScxMwRUE... hello Ini nilai i : 5 </pre>
	<pre> public class Asign { /** * @param args */ /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/ /** */ public static void main(String[] args) { // Kamus int i; // Program System.out.print("hello\n"); i = 5; } } </pre>

	<pre> System.out.println("Ini nilai i : " + i); } } </pre>
	<p>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.</p> <ol style="list-style-type: none"> 1. <code>public class Asign</code>: Ini adalah deklarasi kelas yang dinamai "Asign". Dalam Java, nama kelas harus sama dengan nama file yang menyimpan program, dan kata kunci "public" menunjukkan bahwa kelas ini dapat diakses dari luar paket. 2. <code>public static void main(String[] args)</code>: Ini adalah metode utama (main method) dari kelas. Setiap program Java harus memiliki metode ini sebagai titik masuk untuk eksekusi program. Penjelasan singkat dari parameter "args" adalah sebagai berikut: <ul style="list-style-type: none"> - "public": Kata kunci akses yang menunjukkan metode ini dapat diakses dari luar kelas. - "static": Metode ini dapat dipanggil tanpa membuat instance objek dari kelas. - "void": Metode ini tidak mengembalikan nilai apapun. - "main": Nama metode utama. - "String[] args": Parameter yang diterima oleh metode main, yang biasanya digunakan untuk menerima argumen baris perintah. 3. <code>int i</code>: Mendeklarasikan variabel i dengan tipe data int (integer). 4. <code>System.out.print("hello\n")</code>: Mencetak string "hello" ke konsol. Perintah ini menggunakan kelas System dan metode out dari kelas tersebut, yang digunakan untuk mencetak teks ke arah output standar. 5. <code>i = 5</code>: Menginisialisasi nilai 5 ke variabel i. 6. <code>System.out.println("Ini nilai i : " + i)</code>: Mencetak nilai variabel i ke konsol. Di sini, "+ i" digunakan untuk menggabungkan (concatenate) nilai variabel i dengan string "Ini nilai i : ". Metode println digunakan untuk mencetak string dan menambahkan karakter newline (\n) di akhirnya, sehingga output berikutnya akan dicetak di baris baru.
3	<div> <div> Main.java <div> <div></div> <div></div> <div>Run</div> </div> </div> <div> <pre> 13 long k1 = 10000; 14 char c = 65; /* inisialisasi karakter dengan integer */ 15 char c1 = 'Z'; /* inisialisasi karakter dengan karakter */ 16 double x = 50.2f; 17 float y = 50.2f; 18 // Algoritma 19 // penulisan karakter sebagai karakter 20 System.out.println("Karakter = " + c); 21 System.out.println("Karakter = " + c1); 22 // penulisan karakter sebagai integer 23 System.out.println("Karakter = " + (int) c); 24 System.out.println("Karakter = " + (int) c1); 25 System.out.println("Bilangan integer (short) = " + ks); 26 System.out.println("\t(int) = " + k1); 27 System.out.println("\t(long)= " + k1); 28 System.out.println("Bilangan Real x = " + x); 29 System.out.println("Bilangan Real y = " + y); 30 } 31 } 32 </pre> </div> <div> Output <pre> java -cp /tmp/M5cxIMwRUEL ASIGN1 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> </div> </div>

	<pre> /* Deskripsi : */ /* Program ini berisi contoh sederhana untuk mendefinisikan */ /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024 */ /* variabel-variabel bilangan bulat (short int, int, long int), */ /* karakter, bilangan riil, */ public class ASIGNi { /** * @param args */ 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); // 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. Penjelasan tiap bagian dari program tersebut adalah sebagai berikut:</p> <ol style="list-style-type: none"> 1. public class ASIGNi: Ini adalah deklarasi kelas yang dinamai "ASIGNi". Dalam Java, nama kelas harus sama dengan nama file yang menyimpan program. 2. public static void main(String[] args): Ini adalah metode utama (main method) dari kelas. Setiap program Java harus memiliki metode ini sebagai titik masuk untuk eksekusi program. Penjelasan singkat dari parameter "args" adalah sebagai berikut: <ul style="list-style-type: none"> - "public": Kata kunci akses yang menunjukkan metode ini dapat diakses dari luar kelas. - "static": Metode ini dapat dipanggil tanpa membuat instance objek dari kelas. - "void": Metode ini tidak mengembalikan nilai apapun. - "main": Nama metode utama. - "String[] args": Parameter yang diterima oleh metode main, yang biasanya digunakan untuk menerima argumen baris perintah.

	<p>3. short ks = 1; int ki = 1; long kl = 10000;; Mendeklarasikan variabel-variabel dengan tipe data bilangan bulat (short, int, long) dan menginisialisasi nilainya.</p> <p>4. char c = 65; char c1 = 'Z'; Mendeklarasikan variabel karakter (char) dan menginisialisasi nilainya. Di sini, karakter 'Z' direpresentasikan oleh nilai ASCII-nya, yaitu 65.</p> <p>5. double x = 50.2f; float y = 50.2f;; Mendeklarasikan variabel-variabel dengan tipe data bilangan riil (double dan float) dan menginisialisasi nilainya.</p> <p>6. System.out.println("Karakter = " + c); Mencetak nilai variabel karakter c ke konsol. Di sini, "+" digunakan untuk menggabungkan (concatenate) nilai variabel dengan string yang diberikan.</p> <p>7. System.out.println("Bilangan integer (short) = " + ks); Mencetak nilai variabel integer ks ke konsol.</p> <p>8. System.out.println("\t(int) = " + ki); Mencetak nilai variabel integer ki ke konsol dengan memberikan indentasi menggunakan karakter escape sequence "\t".</p> <p>9. System.out.println("Bilangan Real x = " + x); Mencetak nilai variabel riil x ke konsol.</p> <p>10. System.out.println("Bilangan Real y = " + y); Mencetak nilai variabel riil y ke konsol.</p>
4	 <p>The screenshot shows a Java IDE with a file named 'Main.java'. The code defines a class 'BacaData' with a 'main' method that uses 'Scanner' to read an integer from the user. The output window shows the program running, prompting the user to enter an integer, and displaying the entered value '87'.</p> <pre> 3 /* contoh membaca integer menggunakan Class Scanner */ 4- public class BacaData { 5- /** 6 * @param args 7 */ 8- public static void main(String[] args) { 9 // Kamus 10 int a; 11 Scanner masukan; 12 13 // Program 14 System.out.print("Contoh membaca dan menulis, ketik nilai integer: \n"); 15 masukan = new Scanner(System.in); 16 a = masukan.nextInt(); /* coba ketik: masukan.nextInt(); Apa akibatnya? */ 17 System.out.print("Nilai yang dibaca: " + a); 18 } 19 } 20 </pre> <p>Output:</p> <pre> java -cp /tmp/M5cxMwRUEl BacaData Contoh membaca dan menulis, ketik nilai integer: 87 Nilai yang dibaca: 87 </pre>
	<pre> import java.util.Scanner; /* contoh membaca integer menggunakan Class Scanner */ public class BacaData { /** /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/ * @param args */ public static void main(String[] args) { // Kamus int a; </pre>

	<pre> 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); } } </pre>
	<p>Program ini bertujuan untuk mendemonstrasikan penggunaan kelas Scanner untuk membaca input dari pengguna dan kemudian mencetak nilai yang dibaca ke konsol. Penjelasan tiap bagian dari program tersebut adalah sebagai berikut:</p> <ol style="list-style-type: none"> 1. <code>import java.util.Scanner;</code>: Ini adalah pernyataan import yang mengimpor kelas Scanner dari paket <code>java.util</code>. Ini memungkinkan kita untuk menggunakan kelas Scanner dalam program kita. 2. <code>public class BacaData</code>: Ini adalah deklarasi kelas yang dinamai "BacaData". 3. <code>public static void main(String[] args)</code>: Ini adalah metode utama (main method) dari kelas. Setiap program Java harus memiliki metode ini sebagai titik masuk untuk eksekusi program. Penjelasan singkat dari parameter "args" adalah sebagai berikut: <ul style="list-style-type: none"> - "public": Kata kunci akses yang menunjukkan metode ini dapat diakses dari luar kelas. - "static": Metode ini dapat dipanggil tanpa membuat instance objek dari kelas. - "void": Metode ini tidak mengembalikan nilai apapun. - "main": Nama metode utama. - "String[] args": Parameter yang diterima oleh metode main, yang biasanya digunakan untuk menerima argumen baris perintah. 4. <code>int a;</code>: Mendeklarasikan variabel a dengan tipe data <code>int</code> (integer). 5. <code>Scanner masukan;</code>: Mendeklarasikan variabel masukan dengan tipe data Scanner. Ini adalah objek yang digunakan untuk membaca input dari pengguna. 6. <code>System.out.print("Contoh membaca dan menulis, ketik nilai integer: \n");</code>: Mencetak pesan ke konsol yang meminta pengguna untuk memasukkan nilai integer. 7. <code>masukan = new Scanner(System.in);</code>: Membuat objek Scanner baru yang terhubung ke aliran input standar (<code>System.in</code>). Ini memungkinkan program untuk membaca input yang dimasukkan oleh pengguna dari keyboard. 8. <code>a = masukan.nextInt();</code>: Membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel a menggunakan metode <code>nextInt()</code> dari objek Scanner. 9. <code>System.out.print("Nilai yang dibaca: " + a);</code>: Mencetak nilai yang dibaca dari pengguna ke konsol.

5

Main.java	Output
<pre> 18 19 // Algoritma 20 System.out.print("hello\n"); 21 System.out.print("baca 1 karakter : "); 22 // perintah baca karakter cc 23 cc = dataIn.readLine().charAt(0); 24 System.out.print("baca 1 bilangan : "); 25 // perintah baca bil 26 bil = Integer.parseInt(dataInput.readLine()); 27 /* String kar = JOptionPane.showInputDialog("Karakter 1 : "); 28 * System.out.println(kar); */ 29 // JOptionPane.showMessageDialog(null,"hello"); 30 System.out.print(cc + "\n" + bil + "\n"); 31 System.out.print("bye \n"); 32 /* Tambahkan program membaca/input data 33 menggunakan Class Scanner, Class Console dan Class 34 JOptionPane */ 35 } 36 } </pre>	<pre> Java -cp /tmp/GnVBA03cFg Bacakar hello baca 1 karakter : a baca 1 bilangan : 9 a 9 bye </pre>

```

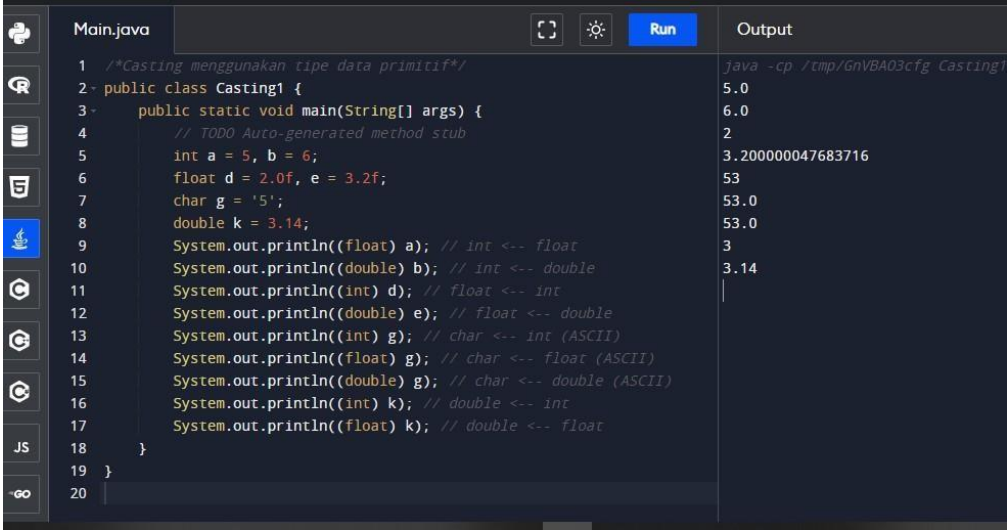
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Bacakar {
    /**
     *Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/
    * @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));

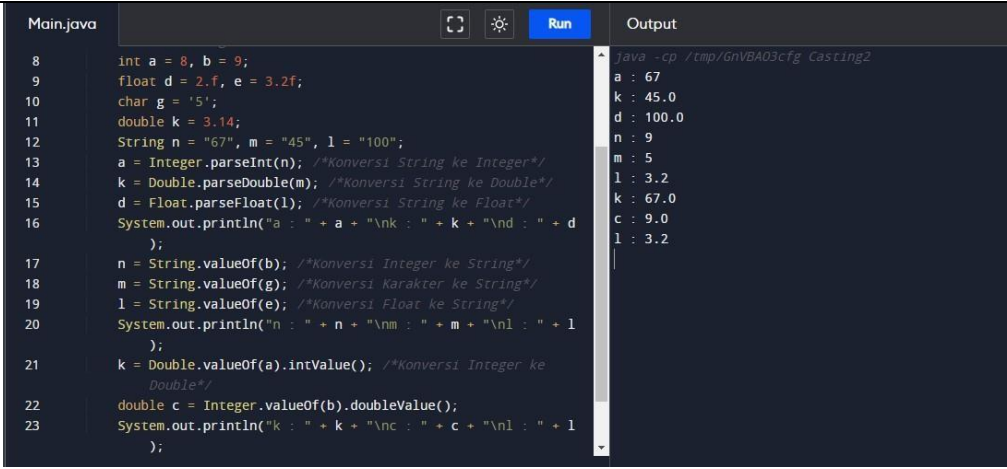
        // Algoritma
        System.out.print("hello\n");
        System.out.print("baca 1 karakter : ");
        // perintah baca karakter cc
        cc = dataIn.readLine().charAt(0);
        System.out.print("baca 1 bilangan : ");
        // perintah baca bil
        bil = Integer.parseInt(dataInput.readLine());
        /* 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

```

	<pre> JOptionPane */ } } </pre>
	<p>Program ini bertujuan untuk membaca input karakter dan bilangan dari pengguna dan kemudian mencetaknya ke konsol.</p> <p>Penjelasan tiap bagian dari program tersebut adalah sebagai berikut:</p> <ol style="list-style-type: none"> 1. <code>import java.io.BufferedReader;</code> Ini adalah pernyataan import yang mengimpor kelas <code>BufferedReader</code> dari paket <code>java.io</code>. Ini digunakan untuk membaca input karakter dari sumber input. 2. <code>import java.io.IOException;</code> Ini adalah pernyataan import yang mengimpor kelas <code>IOException</code> dari paket <code>java.io</code>. Ini digunakan untuk menangani pengecualian yang mungkin terjadi saat operasi masukan/keluaran (I/O). 3. <code>import java.io.InputStreamReader;</code> Ini adalah pernyataan import yang mengimpor kelas <code>InputStreamReader</code> dari paket <code>java.io</code>. Ini digunakan untuk membaca karakter dari aliran input (<code>InputStream</code>). 4. <code>public class Bacakar:</code> Ini adalah deklarasi kelas yang dinamai "Bacakar". 5. <code>public static void main(String[] args) throws IOException:</code> Ini adalah metode utama (main method) dari kelas. Metode ini melemparkan <code>IOException</code> karena operasi pembacaan input dapat menyebabkan pengecualian. Penjelasan singkat dari parameter "args" adalah sama seperti sebelumnya. 6. <code>char cc; int bil;</code> Mendeklarasikan variabel <code>cc</code> dan <code>bil</code> dengan tipe data <code>char</code> dan <code>int</code>. Variabel ini akan digunakan untuk menyimpan input karakter dan bilangan yang dimasukkan oleh pengguna. 7. <code>InputStreamReader isr = new InputStreamReader(System.in);</code> Membuat objek <code>InputStreamReader</code> yang terhubung ke aliran input standar (<code>System.in</code>). Ini akan digunakan bersama dengan <code>BufferedReader</code> untuk membaca karakter dari konsol. 8. <code>BufferedReader dataIn = new BufferedReader(isr);</code> Membuat objek <code>BufferedReader</code> yang menggunakan <code>InputStreamReader isr</code>. Ini digunakan untuk membaca input karakter dari pengguna. 9. <code>BufferedReader dataInput = new BufferedReader(new InputStreamReader(System.in));</code> Cara lain untuk membuat objek <code>BufferedReader</code> yang langsung terhubung ke aliran input standar (<code>System.in</code>). 10. <code>System.out.print("hello\n");</code> Mencetak pesan "hello" ke konsol. 11. <code>cc = dataIn.readLine().charAt(0);</code> Membaca satu baris input dari pengguna menggunakan <code>BufferedReader</code> dan kemudian mengambil karakter pertama dari baris tersebut.

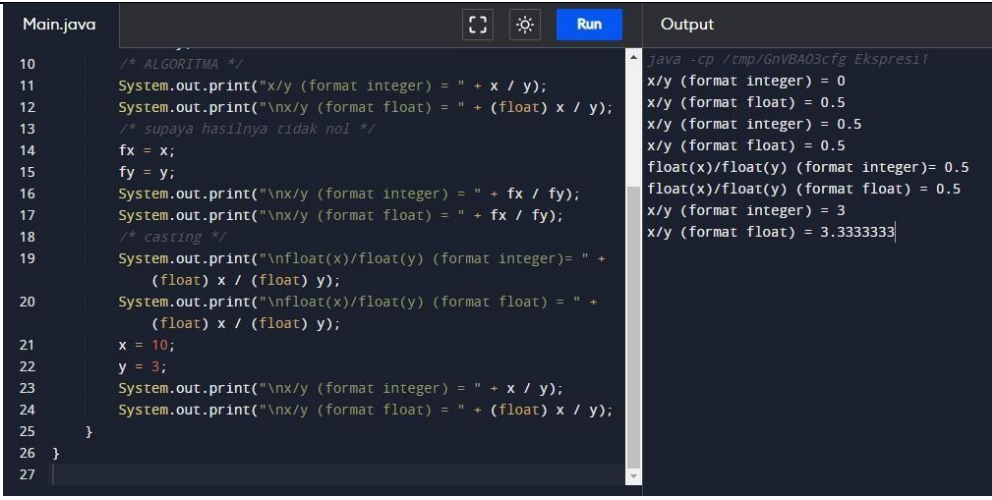
	<p>12. <code>bil = Integer.parseInt(dataInput.readLine());</code>: Membaca satu baris input dari pengguna menggunakan <code>BufferedReader</code>, kemudian mengonversi baris tersebut menjadi bilangan bulat menggunakan metode <code>parseInt</code> dari kelas <code>Integer</code>.</p> <p>13. <code>System.out.print(cc + "\n" + bil + "\n");</code>: Mencetak karakter dan bilangan yang dimasukkan oleh pengguna ke konsol.</p> <p>14. <code>System.out.print("bye \n");</code>: Mencetak pesan "bye" ke konsol.</p> <p>15. Tambahkan program membaca/input data menggunakan Class <code>Scanner</code>, Class <code>Console</code> dan Class <code>JOptionPane</code>: Ini adalah komentar yang menunjukkan bahwa program dapat diperluas dengan menambahkan kemampuan membaca input menggunakan kelas <code>Scanner</code>, <code>Console</code>, atau <code>JOptionPane</code>.</p>
6	 <pre> 1 /*Casting menggunakan tipe data primitif*/ 2 public class Casting1 { 3 public static void main(String[] args) { 4 // TODO Auto-generated method stub 5 int a = 5, b = 6; 6 float d = 2.0f, e = 3.2f; 7 char g = '5'; 8 double k = 3.14; 9 System.out.println((float) a); // int <-- float 10 System.out.println((double) b); // int <-- double 11 System.out.println((int) d); // float <-- int 12 System.out.println((double) e); // float <-- double 13 System.out.println((int) g); // char <-- int (ASCII) 14 System.out.println((float) g); // char <-- float (ASCII) 15 System.out.println((double) g); // char <-- double (ASCII) 16 System.out.println((int) k); // double <-- int 17 System.out.println((float) k); // double <-- float 18 } 19 } 20 </pre> <p>Output:</p> <pre> java -cp /tmp/GnVBA03cfg Casting1 5.0 6.0 2 3.200000047683716 53 53.0 53.0 3 3.14 </pre> <pre> /*Casting menggunakan tipe data primitif*/ /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_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 } } </pre>
	<p>Penjelasan tiap bagian dari program tersebut adalah sebagai berikut:</p>

	<ol style="list-style-type: none"> 1. <code>public class Casting1</code>: Ini adalah deklarasi kelas yang dinamai "Casting1". 2. <code>public static void main(String[] args)</code>: Ini adalah metode utama (main method) dari kelas. Seperti yang disebutkan sebelumnya, metode ini adalah titik masuk untuk eksekusi program Java. 3. <code>int a = 5, b = 6</code>:: Mendeklarasikan dua variabel bertipe data int, yaitu a dan b, dan memberi nilai awal 5 dan 6. 4. <code>float d = 2.0f, e = 3.2f</code>:: Mendeklarasikan dua variabel bertipe data float, yaitu d dan e, dan memberi nilai awal 2.0 dan 3.2. Harus diperhatikan bahwa untuk literal float, diperlukan sufiks 'f' atau 'F' di akhir nilai. 5. <code>char g = '5'</code>:: Mendeklarasikan variabel bertipe data char, yaitu g, dan memberi nilai awal '5'. Perlu diingat bahwa nilai karakter '5' akan diinterpretasikan sebagai nilai ASCII dari karakter '5', yaitu 53. 6. <code>double k = 3.14</code>:: Mendeklarasikan variabel bertipe data double, yaitu k, dan memberi nilai awal 3.14. 7. <code>System.out.println((float) a)</code>:: Melakukan proses casting nilai variabel a dari tipe int menjadi tipe float dan mencetaknya ke konsol. 8. <code>System.out.println((double) b)</code>:: Melakukan proses casting nilai variabel b dari tipe int menjadi tipe double dan mencetaknya ke konsol. 9. <code>System.out.println((int) d)</code>:: Melakukan proses casting nilai variabel d dari tipe float menjadi tipe int dan mencetaknya ke konsol. Proses ini akan membuang bagian desimal dari nilai float. 10. <code>System.out.println((double) e)</code>:: Melakukan proses casting nilai variabel e dari tipe float menjadi tipe double dan mencetaknya ke konsol. 11. <code>System.out.println((int) g)</code>:: Melakukan proses casting nilai variabel g dari tipe char (nilai ASCII) menjadi tipe int dan mencetaknya ke konsol. 12. <code>System.out.println((float) g)</code>:: Melakukan proses casting nilai variabel g dari tipe char (nilai ASCII) menjadi tipe float dan mencetaknya ke konsol. 13. <code>System.out.println((double) g)</code>:: Melakukan proses casting nilai variabel g dari tipe char (nilai ASCII) menjadi tipe double dan mencetaknya ke konsol. 14. <code>System.out.println((int) k)</code>:: Melakukan proses casting nilai variabel k dari tipe double menjadi tipe int dan mencetaknya ke konsol. Proses ini akan membuang bagian desimal dari nilai double. 15. <code>System.out.println((float) k)</code>:: Melakukan proses casting nilai variabel k dari tipe double menjadi tipe float dan mencetaknya ke konsol.
--	---

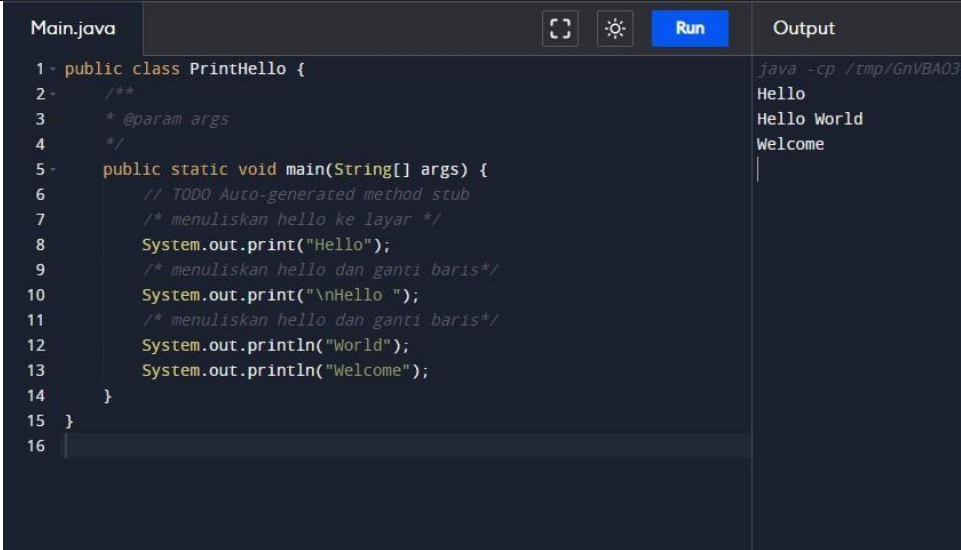
	<p>Tujuan dari program ini adalah untuk memperlihatkan bagaimana proses casting dilakukan antara tipe data primitif dalam Java, serta dampaknya terhadap nilai variabel yang dicasting.</p>
7	 <pre> Main.java 8 int a = 8, b = 9; 9 float d = 2.f, e = 3.2f; 10 char g = '5'; 11 double k = 3.14; 12 String 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); 17 n = String.valueOf(b); /*Konversi Integer ke String*/ 18 m = String.valueOf(g); /*Konversi Karakter ke String*/ 19 l = String.valueOf(e); /*Konversi Float ke String*/ 20 System.out.println("n : " + n + "\nm : " + m + "\nl : " + l 21); 21 k = Double.valueOf(a).intValue(); /*Konversi Integer ke 22 Double*/ 22 double c = Integer.valueOf(b).doubleValue(); 23 System.out.println("k : " + k + "\nc : " + c + "\nl : " + l 24); </pre> <pre> Output java -cp /tmp/GnVBA03cfg 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> /*Casting menggunakan tipe data Class*/ public class Casting2 { /** *Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/ * @param args */ public static void main(String[] args) { // 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); } } </pre>
	<p>Program ini bertujuan untuk mendemonstrasikan proses casting (pengubahan tipe data) antara tipe data primitif dan tipe data kelas (Class) dalam bahasa pemrograman Java.</p> <p>Penjelasan tiap bagian dari program tersebut adalah sebagai berikut:</p> <ol style="list-style-type: none"> 1. public class Casting2: Ini adalah deklarasi kelas yang dinamai "Casting2".

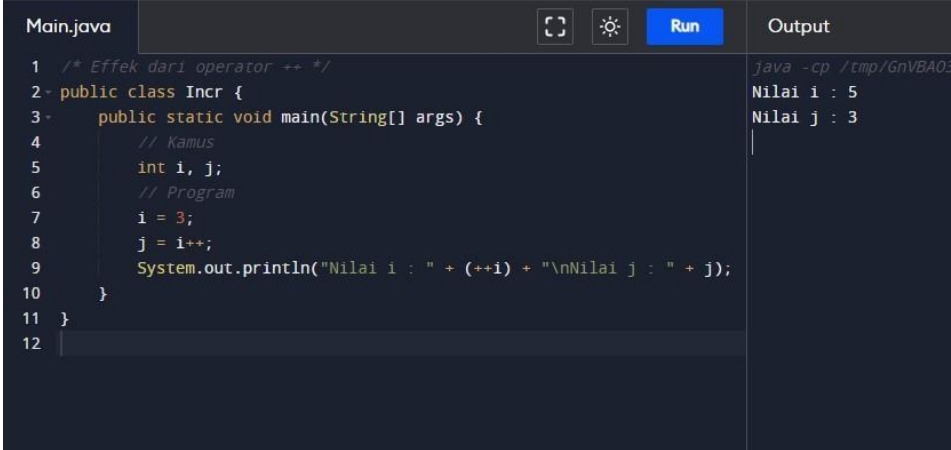
	<p>2. <code>public static void main(String[] args):</code> Ini adalah metode utama (main method) dari kelas. Metode ini adalah titik masuk untuk eksekusi program Java.</p> <p>3. <code>int a = 8, b = 9;:</code> Mendeklarasikan dua variabel bertipe data <code>int</code>, yaitu <code>a</code> dan <code>b</code>, dan memberi nilai awal 8 dan 9.</p> <p>4. <code>float d = 2.f, e = 3.2f;:</code> Mendeklarasikan dua variabel bertipe data <code>float</code>, yaitu <code>d</code> dan <code>e</code>, dan memberi nilai awal 2.0 dan 3.2. Harus diperhatikan bahwa untuk literal <code>float</code>, diperlukan sufiks <code>'f'</code> atau <code>'F'</code> di akhir nilai.</p> <p>5. <code>char g = '5';:</code> Mendeklarasikan variabel bertipe data <code>char</code>, yaitu <code>g</code>, dan memberi nilai awal <code>'5'</code>. Perlu diingat bahwa nilai karakter <code>'5'</code> akan diinterpretasikan sebagai nilai ASCII dari karakter <code>'5'</code>, yaitu 53.</p> <p>6. <code>double k = 3.14;:</code> Mendeklarasikan variabel bertipe data <code>double</code>, yaitu <code>k</code>, dan memberi nilai awal 3.14.</p> <p>7. <code>String n = "67", m = "45", l = "100";:</code> Mendeklarasikan tiga variabel bertipe data <code>String</code>, yaitu <code>n</code>, <code>m</code>, dan <code>l</code>, dan memberi nilai awal <code>"67"</code>, <code>"45"</code>, dan <code>"100"</code>.</p> <p>8. <code>a = Integer.parseInt(n);:</code> Melakukan proses casting dari tipe data <code>String</code> (<code>n</code>) ke tipe data <code>int</code> (<code>a</code>) menggunakan metode <code>parseInt()</code> dari kelas <code>Integer</code>.</p> <p>9. <code>k = Double.parseDouble(m);:</code> Melakukan proses casting dari tipe data <code>String</code> (<code>m</code>) ke tipe data <code>double</code> (<code>k</code>) menggunakan metode <code>parseDouble()</code> dari kelas <code>Double</code>.</p> <p>10. <code>d = Float.parseFloat(l);:</code> Melakukan proses casting dari tipe data <code>String</code> (<code>l</code>) ke tipe data <code>float</code> (<code>d</code>) menggunakan metode <code>parseFloat()</code> dari kelas <code>Float</code>.</p> <p>11. <code>System.out.println("a : " + a + "\nk : " + k + "\nd : " + d);:</code> Mencetak nilai variabel <code>a</code>, <code>k</code>, dan <code>d</code> ke konsol.</p> <p>12. <code>n = String.valueOf(b);:</code> Melakukan proses casting dari tipe data <code>int</code> (<code>b</code>) ke tipe data <code>String</code> (<code>n</code>) menggunakan metode <code>valueOf()</code> dari kelas <code>String</code>.</p> <p>13. <code>m = String.valueOf(g);:</code> Melakukan proses casting dari tipe data <code>char</code> (<code>g</code>) ke tipe data <code>String</code> (<code>m</code>) menggunakan metode <code>valueOf()</code> dari kelas <code>String</code>.</p> <p>14. <code>l = String.valueOf(e);:</code> Melakukan proses casting dari tipe data <code>float</code> (<code>e</code>) ke tipe data <code>String</code> (<code>l</code>) menggunakan metode <code>valueOf()</code> dari kelas <code>String</code>.</p> <p>15. <code>System.out.println("n : " + n + "\nm : " + m + "\nl : " + l);:</code> Mencetak nilai variabel <code>n</code>, <code>m</code>, dan <code>l</code> ke konsol.</p> <p>16. <code>k = Double.valueOf(a).intValue();:</code> Melakukan proses casting dari tipe data <code>int</code> (<code>a</code>) ke tipe data <code>double</code> (<code>k</code>) menggunakan metode <code>valueOf()</code> dari kelas <code>Double</code>, kemudian diikuti dengan pemanggilan metode <code>intValue()</code> untuk mengonversi kembali menjadi <code>int</code>.</p>
--	---

	<p>17. <code>double c = Integer.valueOf(b).doubleValue();</code> Melakukan proses casting dari tipe data <code>int</code> (<code>b</code>) ke tipe data <code>double</code> (<code>c</code>) menggunakan metode <code>valueOf()</code> dari kelas <code>Integer</code>, kemudian diikuti dengan pemanggilan metode <code>doubleValue()</code> untuk mengonversi kembali menjadi <code>double</code>.</p> <p>Tujuan dari program ini adalah untuk memperlihatkan bagaimana proses casting dilakukan antara tipe data primitif dan tipe data kelas (Class) dalam Java, serta dampaknya terhadap nilai variabel yang dicasting.</p>
8	<div> <div> Main.java <div> <div></div> <div></div> <div>Run</div> </div> <div>Output</div> </div> <div> <pre> 1 /* pemakaian operator kondisional */ 2- public class Ekspresi { 3- /** 4- * @param args 5- */ 6- public static void main(String[] args) { 7- // TODO Auto-generated method stub 8- /* KAMUS */ 9- int x = 1; 10 int y = 2; 11 /* ALGORITMA */ 12 System.out.print("x = " + x + "\n"); 13 System.out.print("y = " + y + "\n"); 14 System.out.print("hasil ekspresi = (x<y)?x:y = " + 15 ((x < y) ? x : y)); 16- /*Gunakan dalam kurung "(statemen dan kondisi)" untuk 17 menyatakan satu kesatuan pernyataan*/ 18 } 19 } 20 </pre> <div> <pre> java -cp /tmp/GnVBA03cfg Ekspresi x = 1 y = 2 hasil ekspresi = (x<y)?x:y = 1 </pre> </div> </div> </div>
	<pre> /* pemakaian operator kondisional */ public class Ekspresi { /** /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/ * @param args */ 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)); /*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>Penjelasan tiap bagian dari program tersebut adalah sebagai berikut:</p> <ol style="list-style-type: none"> 1. <code>public class Ekspresi</code>: Ini adalah deklarasi kelas yang dinamai "Ekspresi". 2. <code>public static void main(String[] args)</code>: Ini adalah metode utama (main method) dari kelas. Metode ini adalah titik masuk untuk eksekusi program Java.

	<p>3. <code>int x = 1; int y = 2;</code> Mendeklarasikan dua variabel bertipe data int, yaitu x dan y, dan memberi nilai awal 1 dan 2.</p> <p>4. <code>System.out.print("x = " + x + "\n");</code> Mencetak nilai variabel x ke konsol.</p> <p>5. <code>System.out.print("y = " + y + "\n");</code> Mencetak nilai variabel y ke konsol.</p> <p>6. <code>System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y));</code> Menggunakan operator kondisional (ternary operator) untuk memeriksa apakah nilai x kurang dari nilai y. Jika benar, maka nilai x akan dicetak, jika tidak, nilai y yang akan dicetak.</p> <ul style="list-style-type: none"> - <code>(x < y)</code> adalah ekspresi kondisional. - <code>(x < y) ? x : y</code> adalah bentuk operator kondisional yang mengembalikan nilai x jika ekspresi benar, dan nilai y jika ekspresi salah. - <code>System.out.print("hasil ekspresi = (x<y)?x:y = " + ...)</code> 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.</p>
9	 <p>The screenshot shows a Java IDE with a file named 'Main.java'. The code includes comments in Indonesian and Java code for integer division, floating-point division, and casting. The output window shows the results of these operations.</p> <pre> Main.java 10 /* ALGORITMA */ 11 System.out.print("x/y (format integer) = " + x / y); 12 System.out.print("\nx/y (format float) = " + (float) x / y); 13 /* supaya hasilnya tidak nol */ 14 fx = x; 15 fy = y; 16 System.out.print("\nx/y (format integer) = " + fx / fy); 17 System.out.print("\nx/y (format float) = " + fx / fy); 18 /* casting */ 19 System.out.print("\nfloat(x)/float(y) (format integer)= " + 20 (float) x / (float) y); 21 System.out.print("\nfloat(x)/float(y) (format float) = " + 22 (float) x / (float) y); 23 x = 10; 24 y = 3; 25 System.out.print("\nx/y (format integer) = " + x / y); 26 System.out.print("\nx/y (format float) = " + (float) x / y); 27 } Output java -cp /tmp/GnVBA03cfg 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>
	<pre> /* pembagian integer, casting */ /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_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 */ </pre>

	<pre> 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); } } </pre>
	<p>Program ini bertujuan untuk mendemonstrasikan operasi pembagian integer dan penggunaan casting dalam bahasa pemrograman Java.</p> <p>Penjelasan tiap bagian dari program tersebut adalah sebagai berikut:</p> <ol style="list-style-type: none"> 1. <code>public class Ekspresi1</code>: Ini adalah deklarasi kelas yang dinamai "Ekspresi1". 2. <code>public static void main(String[] args)</code>: Ini adalah metode utama (main method) dari kelas. Metode ini adalah titik masuk untuk eksekusi program Java. 3. <code>int x = 1; int y = 2;</code>: Mendeklarasikan dua variabel bertipe data int, yaitu x dan y, dan memberi nilai awal 1 dan 2. 4. <code>float fx; float fy;</code>: Mendeklarasikan dua variabel bertipe data float, yaitu fx dan fy. 5. <code>System.out.print("x/y (format integer) = " + x / y);</code>: Mencetak hasil pembagian x dengan y ke konsol. Karena kedua operand (x dan y) adalah integer, maka hasilnya akan berupa integer juga. 6. <code>System.out.print("\nx/y (format float) = " + (float) x / y);</code>: Casting nilai x menjadi float sebelum melakukan pembagian, sehingga hasilnya akan berupa float. 7. <code>fx = x; fy = y;</code>: Mengkopi nilai variabel x dan y ke variabel fx dan fy yang bertipe float. 8. <code>System.out.print("\nx/y (format integer) = " + fx / fy);</code>: Mencetak hasil pembagian fx dengan fy ke konsol. Karena kedua operand (fx dan fy) adalah float, maka hasilnya akan berupa float. 9. <code>System.out.print("\nx/y (format float) = " + fx / fy);</code>: Mencetak hasil pembagian fx dengan fy ke konsol. Karena kedua operand (fx dan fy) adalah float, maka hasilnya akan berupa float. 10. <code>System.out.print("\nfloat(x)/float(y) (format integer)= " + (float) x / (float) y);</code>: Casting nilai x dan y menjadi float sebelum melakukan pembagian, sehingga hasilnya akan berupa float.

	<p>11. <code>System.out.print("\nfloat(x)/float(y) (format float) = " + (float) x / (float) y);</code>; Casting nilai x dan y menjadi float sebelum melakukan pembagian, sehingga hasilnya akan berupa float.</p> <p>12. <code>x = 10; y = 3;</code>; Mengubah nilai variabel x dan y.</p> <p>13. <code>System.out.print("\nx/y (format integer) = " + x / y);</code>; Mencetak hasil pembagian x dengan y ke konsol. Karena kedua operand (x dan y) adalah integer, maka hasilnya akan berupa integer.</p> <p>14. <code>System.out.print("\nx/y (format float) = " + (float) x / y);</code>; Casting nilai x menjadi float sebelum melakukan pembagian, sehingga hasilnya akan berupa float.</p> <p>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.</p>
10	 <pre> Main.java 1 public class PrintHello { 2 /** 3 * @param args 4 */ 5 public static void main(String[] args) { 6 // TODO Auto-generated method stub 7 /* menuliskan hello ke layar */ 8 System.out.print("Hello"); 9 /* menuliskan hello dan ganti baris*/ 10 System.out.print("\nHello "); 11 /* menuliskan hello dan ganti baris*/ 12 System.out.println("World"); 13 System.out.println("Welcome"); 14 } 15 } 16 Output java -cp /tmp/GnVBA03 Hello Hello World Welcome </pre>
	<pre> public class PrintHello { /** /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/ * @param args */ 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"); } } </pre>
	<p>Program ini memiliki tujuan sederhana untuk mencetak beberapa teks ke layar. Berikut penjelasan setiap bagian dari program tersebut:</p>

	<ol style="list-style-type: none"> 1. <code>public class PrintHello</code>: Ini adalah deklarasi kelas yang dinamai "PrintHello". 2. <code>public static void main(String[] args)</code>: Ini adalah metode utama (main method) dari kelas. Metode ini adalah titik masuk untuk eksekusi program Java. 3. <code>// TODO Auto-generated method stub</code>: Ini adalah komentar yang dihasilkan secara otomatis oleh IDE (Integrated Development Environment) seperti Eclipse. Ini menandakan bahwa kode di bawahnya adalah stub atau kode sementara yang dihasilkan otomatis. 4. <code>System.out.print("Hello");</code>: Ini adalah perintah untuk mencetak teks "Hello" ke layar tanpa mengganti baris setelahnya. 5. <code>System.out.print("\nHello ");</code>: Ini adalah perintah untuk mencetak teks "Hello" ke layar dengan mengganti baris setelahnya. 6. <code>System.out.println("World");</code>: Ini adalah perintah untuk mencetak teks "World" ke layar dengan mengganti baris setelahnya. Perbedaannya dengan perintah sebelumnya adalah bahwa ini menggunakan "println" yang secara otomatis akan menambahkan baris baru setelah mencetak teks. 7. <code>System.out.println("Welcome");</code>: Ini adalah perintah untuk mencetak teks "Welcome" ke layar dengan mengganti baris setelahnya. <p>Tujuan dari program ini adalah untuk menunjukkan cara menggunakan metode "print" dan "println" untuk mencetak teks ke layar dalam bahasa pemrograman Java. Program ini juga memperlihatkan perbedaan antara "print" dan "println" dalam hal penambahan baris baru setelah mencetak teks.</p>
11	
	<pre> /* Efek dari operator ++ */ /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/ public class Incr { public static void main(String[] args) { // Kamus int i, j; // Program i = 3; j = i++; System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j); } } </pre>

	<pre> j = i++; System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j); } } </pre>
	<p>Program ini bertujuan 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++).</p> <p>Berikut adalah penjelasan dari program tersebut:</p> <ol style="list-style-type: none"> Keyword yang digunakan: <ul style="list-style-type: none"> - `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. - `main`: Metode utama yang akan dieksekusi saat program dijalankan. - `String[] args`: Parameter dari metode `main`, yang merupakan array dari argumen baris perintah. Variabel yang digunakan: <ul style="list-style-type: none"> - `i`: Variabel bertipe int yang digunakan untuk menyimpan nilai integer. - `j`: Variabel bertipe int yang juga digunakan untuk menyimpan nilai integer. Tujuan program: <p>Program ini bertujuan 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. 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.</p>
12	<div> <div> Main.java <div> <div></div> <div></div> <div>Run</div> </div> <div>Output</div> </div> <pre> 1 /* pemakaian beberapa operator terhadap bit */ 2- public class Oper1 { 3- public static void main(String[] args) { 4 // KAMUS 5 int n = 10; /* 1010 */ 6 int x = 1; /* 1 */ 7 int y = 2; /* 10 */ 8 // ALGORITMA 9 System.out.println("n = " + n); 10 System.out.println("x = " + x); 11 System.out.println("y = " + y); 12 System.out.println("n & 8 = " + (n & 8)); /* 1010 AND 1000 */ 13 System.out.println("x & ~8 = " + (x & ~8)); /* 1 AND 0111 */ 14 System.out.println("y << 2 = " + (y << 2)); /* 10 ==> 1000 = 8 */ 15 System.out.println("y >> 3 = " + (y >> 3)); /* 10 ==> 0000 = 0 */ 16 } 17 } </pre> <pre> java -cp /tmp/GnV n = 10 x = 1 y = 2 n & 8 = 8 x & ~8 = 1 y << 2 = 8 y >> 3 = 0 </pre> </div>
	<pre> /* pemakaian beberapa operator terhadap bit */ /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/ public class Oper1 { public static void main(String[] args) { </pre>

	<pre>// 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 */ } }</pre>
	<p>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 (>>).</p> <p>Berikut adalah penjelasan dari program tersebut:</p> <ol style="list-style-type: none"> 1. Keyword yang digunakan: <ul style="list-style-type: none"> - `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. - `main`: Metode utama yang akan dieksekusi saat program dijalankan. - `String[] args`: Parameter dari metode `main`, yang merupakan array dari argumen baris perintah. 2. Variabel yang digunakan: <ul style="list-style-type: none"> - `n`, `x`, `y`: Variabel bertipe int yang digunakan untuk menyimpan nilai integer. 3. Tujuan program: <p>Program ini bertujuan 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.</p>

13

Main.java	Output
<pre> 4 // KAMUS 5 char i, j; 6 // ALGORITMA 7 i = 3; /* 00000011 dalam biner */ 8 j = 4; /* 00000100 dalam biner */ 9 System.out.println("i = " + (int) i); 10 System.out.println("j = " + (int) j); 11 System.out.println("i & j = " + (i & j)); /* 0: 00000000 dalam biner */ 12 System.out.println("i j = " + (i j)); /* 7: 00000111 biner */ 13 System.out.println("i ^ j = " + (i ^ j)); /* 7: 00000111 biner Ingat!!! operator "^" pada bahasa java bukan sebagai pangkat */ 14 System.out.println(Math.pow(i, j)); /* Class Math memiliki method pow(a,b) untuk pemangkatan */ 15 System.out.println("~i = " + ~i); /* -4: 11111100 biner */ 16 } 17 } 18 </pre>	<pre> java -cp /tmp/GnVBA03cfg i = 3 j = 4 i & j = 0 i j = 7 i ^ j = 7 81.0 ~i = -4 </pre>

```

/* pemakaian beberapa operator terhadap RELATIONAL DAN bit */
/**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/
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 */
    }
}

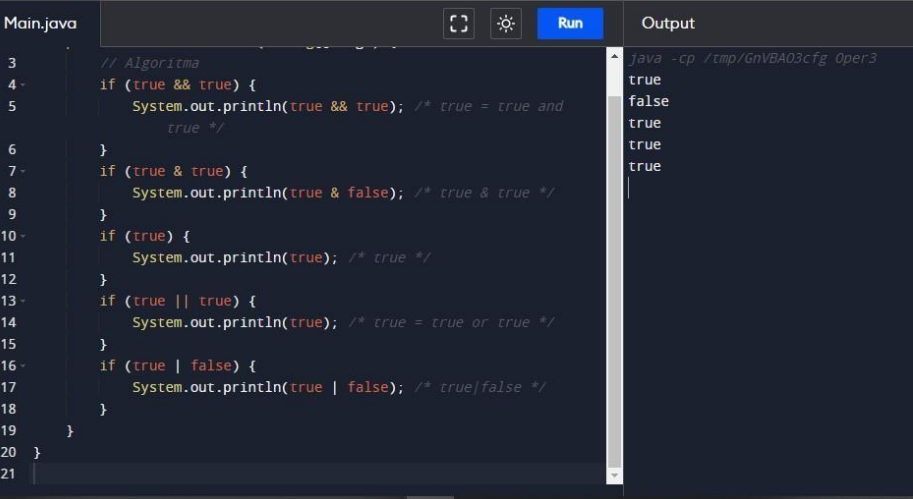
```

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.
- `main`: Metode utama yang akan dieksekusi saat program dijalankan.

	<ul style="list-style-type: none"> - <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</code>, <code>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	 <pre> Main.java 3 // Algoritma 4 if (true && true) { 5 System.out.println(true && true); /* true = true and true */ 6 } 7 if (true & true) { 8 System.out.println(true & false); /* true & true */ 9 } 10 if (true) { 11 System.out.println(true); /* true */ 12 } 13 if (true true) { 14 System.out.println(true); /* true = true or true */ 15 } 16 if (true false) { 17 System.out.println(true false); /* true false */ 18 } 19 } 20 } 21 Output java -cp ./tmp/GnVBA03cfg Oper3 true false true true true </pre>
	<pre> /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/ public class Oper3 { public static void main(String[] args) { // Algoritma if (true && true) { System.out.println(true && true); /* true = true and true */ } } } </pre>

	<pre> 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 */ } } } </pre>
	<p>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.</p> <p>Berikut adalah penjelasan dari program tersebut:</p> <ol style="list-style-type: none"> Keyword yang digunakan: <ul style="list-style-type: none"> - `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. - `main`: Metode utama yang akan dieksekusi saat program dijalankan. - `String[] args`: Parameter dari metode `main`, yang merupakan array dari argumen baris perintah. Tujuan program: <p>Program ini bertujuan untuk menunjukkan perbedaan antara operator logika (`&`, ` `) dan operator bitwise (`&`, ` `) dalam bahasa Java.</p> <ul style="list-style-type: none"> - 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. Output: <ul style="list-style-type: none"> - Program mencetak hasil dari beberapa ekspresi yang menggunakan operator logika dan operator bitwise. - Hasil dari setiap ekspresi dicetak, yang mencakup nilai `true` atau `false`.

15

Main.java



Run

Output

```

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

```

```

java -cp /tmp/GnVBA03cfg 0
Nilai e = 10
Nilai k = 0
Nilai k = 4

```

```

/* Operator terner */
/**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/
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);
    }
}

```

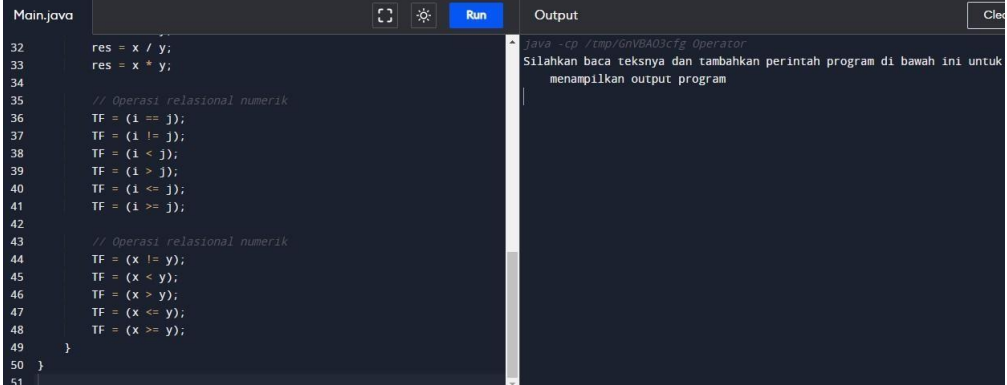
Program ini bertujuan untuk mengilustrasikan penggunaan operator ternary (^?:) dalam bahasa pemrograman Java.

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.
- `main`: Metode utama yang akan dieksekusi saat program dijalankan.
- `String[] args`: Parameter dari metode `main`, yang merupakan array dari argumen baris perintah.

2. Tujuan program:

	<p>Program ini bertujuan untuk menunjukkan cara menggunakan operator ternary (`?:`) dalam Java. Operator ternary memungkinkan kita untuk menulis kondisi if-else dalam satu baris kode.</p> <p>3. Proses eksekusi:</p> <ul style="list-style-type: none"> - Variabel `i` dan `j` diinisialisasi dengan nilai 0. - Variabel `c` dan `d` diinisialisasi dengan nilai karakter ASCII. - Variabel `e` diinisialisasi dengan hasil ekspresi ternary. Jika nilai ASCII dari `c` lebih besar dari nilai ASCII `d`, maka nilai `e` akan menjadi nilai ASCII `c`, jika tidak, maka nilai `e` akan menjadi nilai ASCII `d`. - Variabel `k` diinisialisasi dengan hasil ekspresi ternary. Jika nilai `i` lebih besar dari nilai `j`, maka nilai `k` akan menjadi nilai `i`, jika tidak, maka nilai `k` akan menjadi nilai `j`. - Variabel `i` dan `j` ditingkatkan nilainya (increment) dengan menggunakan operator `++`. - Hasil dari variabel `e` dan `k` dicetak. <p>4. Output:</p> <ul style="list-style-type: none"> - Program mencetak nilai dari variabel `e` dan `k` setelah evaluasi ekspresi ternary.
16	
	<pre> /* Contoh pengoperasian variabel bertipe dasar */ /**Ahmad_Dzaki_Ubaidillah_Nim_13020220092_20_Februari_2024*/ 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 */ // Operasi numerik </pre>

	<pre> 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); } } </pre>
	<p>Program ini bertujuan untuk menunjukkan contoh pengoperasian variabel berjenis data dasar (primitive) dan operator-operator numerik dan relasional dalam bahasa pemrograman Java.</p> <p>Berikut adalah penjelasan dari program tersebut:</p> <ol style="list-style-type: none"> 1. Keyword yang digunakan: <ul style="list-style-type: none"> - <code>public</code>: Mendefinisikan aksesibilitas kelas. - <code>class</code>: Mendefinisikan sebuah kelas. - <code>static</code>: Mendeklarasikan metode atau variabel sebagai statis yang berarti dapat diakses tanpa membuat instance dari kelas tersebut. - <code>void</code>: Menunjukkan bahwa metode <code>main</code> tidak mengembalikan nilai apa pun. - <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. 2. Tujuan program:

Program ini bertujuan untuk memberikan contoh penggunaan operator-operator numerik dan relasional dalam bahasa Java serta operasi pada variabel-variabel berjenis data dasar.

3. Proses eksekusi:

- Variabel-variabel yang dideklarasikan meliputi beberapa tipe data dasar, seperti ``boolean``, ``int``, ``float``.
- Setelah deklarasi variabel, beberapa operasi numerik dan relasional dilakukan. Misalnya, penjumlahan, pengurangan, perkalian, pembagian, dan operasi modulo.
- Operasi-relasional numerik juga dilakukan dengan menggunakan operator seperti ``==``, ``!=``, ``<``, ``>``, ``<=``, ``>=``.
- Setelah itu, hasil dari operasi tersebut dapat ditugaskan ke variabel boolean.

4. Output:

- Program ini tidak mencetak output secara langsung ke konsol karena tidak ada perintah ``System.out.println()`` yang diaktifkan. Oleh karena itu, untuk melihat hasil operasi, kita dapat menambahkan perintah ``System.out.println()`` untuk mencetak nilai variabel atau hasil operasi ke konsol.