

Nama : Kelvianto Pratama Harum

NIM : 200210500016

MK : Pemrograman Lanjut

Dosen : Muhammad Fajar B, S.Pd., M.Cs.

Pertemuan 1

Java Fundamental

A. Contoh 1.1 – Pengenalan Java

Source code:

```
1 package com.example.JavaFundamental;
2
3 public class App {
4     public static void main(String[] args) {
5         // Contoh 1.1 dari Modul Pemrograman Lanjut
6         System.out.println("Hallo Covid-19");
7     }
8 }
```

Output:

```
> Task :app:run
Hallo Covid-19

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **App.java** tergabung pada package **com.example.JavaFundamental**

Pada baris **3**, dideklarasikan Class bernama **App**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **8**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **App**.

Pada baris **4**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **4** hingga baris **7**.

Pada baris **6**, digunakan untuk menampilkan string “Hallo Covid-19”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**,

member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

B. Contoh 1.2 – Tipe Data

Source code:

```
1 package com.example.JavaFundamental;
2
3 public class Contohl2 {
4     public static void main(String args[]){
5         byte myNumbyte = 100;
6         short myNumshort = 5000;
7         int myNumint = 100000;
8         long myNumlong = 150000000000L;
9         float myNumfloat = 5.75f;
10        double myNumdouble = 19.99d;
11
12        System.out.println(myNumbyte);
13        System.out.println(myNumshort);
14        System.out.println(myNumint);
15        System.out.println(myNumlong);
16        System.out.println(myNumfloat);
17        System.out.println(myNumdouble);
18
19    }
20 }
21
```

Output:

```
> Task :app:run
100
5000
100000
150000000000
5.75
19.99

BUILD SUCCESSFUL in 728ms
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Contoh12.java** bergabung pada package **com.example.JavaFundamental**

Pada baris **3**, dideklarasikan Class bernama **Contoh12**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **20**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Contoh12**.

Pada baris **4**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **4** hingga baris **19**.

Pada baris **5**, dideklarasikan variable **myNumbyte** dengan tipe data **byte**, tipe data ini berukuran **1 byte** atau **8 bit** dan secara default akan menggunakan tanda (**signed**) sehingga dapat menampung data dengan nilai dari -128 hingga 127. Variabel ini didefinisikan dengan nilai **100**.

Pada baris **6**, dideklarasikan variable **myNumshort** dengan tipe data **short**, tipe data ini berukuran **2 bytes** atau **16 bit** dan secara default akan menggunakan tanda (**signed**) sehingga dapat menampung data dengan nilai dari -32.768 hingga 32.767. Variabel ini didefinisikan dengan nilai **5000**.

Pada baris **7**, dideklarasikan variable **myNumint** dengan tipe data **int**, tipe data ini berukuran **4 bytes** atau **32 bit** dan secara default akan menggunakan tanda (**signed**) sehingga dapat menampung data dengan nilai dari -2.147.483.648 hingga 2.147.483.647. Variabel ini didefinisikan dengan nilai **100000**.

Pada baris **8**, dideklarasikan variable **myNumlong** dengan tipe data **long**, tipe data ini berukuran **8 bytes** atau **64 bit** dan secara default akan menggunakan tanda (**signed**) sehingga dapat menampung data dengan nilai dari -9.223.372.036.854.775.808 hingga 9.223.372.036.854.775.807. Variabel ini didefinisikan dengan nilai **15000000000L**. Huruf “L” pada akhir nilai menandakan ke compiler bahwa nilai ini adalah data integer long (64 bit).

Pada baris **9**, dideklarasikan variable **myNumfloat** dengan tipe data **float**, tipe data ini berukuran **4 bytes** atau **32 bit** dan secara default akan menggunakan tanda (**signed**) sehingga dapat menampung data dengan nilai pecahan dengan akurasi hingga 7 desimal (angka dibelakang koma), baik negatif maupun positif. Variabel ini didefinisikan dengan nilai **5.75f**. Huruf “f” pada akhir nilai menandakan ke compiler bahwa nilai adalah data floating point (32 bit).

Pada baris **10**, dideklarasikan variable **myNumdouble** dengan tipe data **double**, tipe data ini berukuran **8 bytes** atau **64 bit** dan secara default akan menggunakan tanda (**signed**) sehingga dapat menampung data dengan nilai pecahan dengan akurasi hingga 15 desimal (angka dibelakang koma), baik negatif maupun positif. Variabel ini

didefinisikan dengan nilai **19.99d**. Huruf “d” pada akhir nilai menandakan ke compiler bahwa nilai ini adalah data double (64 bit).

Pada baris **12**, digunakan untuk menampilkan nilai pada variabel **myNumbyte** yaitu **100**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **13**, digunakan untuk menampilkan nilai pada variabel **myNumshort** yaitu **5000**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **14**, digunakan untuk menampilkan nilai pada variabel **myNumint** yaitu **100000**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **15**, digunakan untuk menampilkan nilai pada variabel **myNumlong** yaitu **15000000000**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **16**, digunakan untuk menampilkan nilai pada variabel **myNumfloat** yaitu **5.75**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **17**, digunakan untuk menampilkan nilai pada variabel **myNumdouble** yaitu **19.99d**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

C. Contoh 1.3 – Operasi Matematika

Source code:

```
1 package com.example.JavaFundamental;
2
3 public class Contoh13 {
4     public static void main(String args[]) {
5         int a,b,c;
6         a = 10;
7         b = 20;
8         c = a+b;
9
10        System.out.println(c);
11    }
12 }
13
```

Output:

```
> Task :app:run
30

BUILD SUCCESSFUL in 613ms
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Contoh13.java** bergabung pada package **com.example.JavaFundamental**

Pada baris **3**, dideklarasikan Class bernama **Contoh13**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **12**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Contoh13**.

Pada baris **4**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **4** hingga baris **11**.

Pada baris **5**, dideklarasikan variabel **a, b dan c** dengan tipe data **int (signed, 4 bytes/32 bit)**.

Pada baris **6**, variabel **a** didefinisikan dengan nilai **10**.

Pada baris **7**, variabel **b** didefinisikan dengan nilai **20**.

Pada baris **8**, variabel **c** didefinisikan dengan hasil penjumlahan dari variabel **a dan b (a+b)** yaitu **30**.

Pada baris **10**, digunakan untuk menampilkan nilai pada variabel **c** yaitu **30**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

D. Contoh 1.4 – Seleksi Kondisi

Source code:

```
1  package com.example.JavaFundamental;
2
3  public class Contoh14 {
4      public static void main(String args[]){
5          int waktu = 22;
6          if (waktu < 10){
7              System.out.println("Selamat pagi.");
8          } else if (waktu < 20){
9              System.out.println("Selamat siang,");
10         } else {
11             System.out.println("Selamat malam.");
12         }
13     }
14 }
15
```

Output:

```
> Task :app:run
Selamat malam.

BUILD SUCCESSFUL in 566ms
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Contoh14.java** tergabung pada package **com.example.JavaFundamental**

Pada baris **3**, dideklarasikan Class bernama **Contoh14**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **14**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Contoh14**.

Pada baris **4**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **4** hingga baris **13**.

Pada baris **5**, dideklarasikan variabel **waktu** dengan tipe data **int (signed, 4 bytes/32 bit)** dan nilai **22**.

Pada baris **6**, diberikan seleksi kondisi dengan **if-else statement**, statement ini memiliki scope dari baris **6** hingga **12** dengan 3 bagian, yaitu **if**, **else if**, dan **else**. Jika salah satu bagian ini terpenuhi, maka bagian selanjutnya tidak akan dieksekusi. Seleksi kondisi pada **if-else statement** berjalan sekuensial, yang berarti setiap bagian diperiksa dari bagian yang pertama kali dideklarasikan, lalu lanjut ke bagian kedua, dan seterusnya. Pada bagian ini diperiksa kondisi jika variabel **waktu** lebih kecil dari nilai **10**, jika iya maka program didalam bagian **if-else statement** ini akan dijalankan. Jika tidak, maka program akan memeriksa bagian **if-else statement** berikutnya. Bagian ini memiliki scope dari baris **6** hingga baris **8** dimana penutup blok (tanda ‘}’) berakhir.

Pada baris **7**, digunakan untuk menampilkan nilai string **“Selamat pagi.”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **8**, adalah bagian kedua dari **if-else statement** yang dideklarasikan pada baris **6**. Pada bagian ini diperiksa kondisi jika variabel **waktu** lebih kecil dari nilai **20**, jika iya maka program didalam bagian **if-else statement** ini akan dijalankan. Jika tidak, maka program akan memeriksa bagian **if-else statement** berikutnya. Bagian ini memiliki scope dari baris **8** hingga baris **10** dimana penutup blok (tanda ‘}’) berakhir.

Pada baris **9**, digunakan untuk menampilkan nilai string **“Selamat siang.”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **10**, adalah bagian ketiga dari **if-else statement** yang dideklarasikan pada baris **6**. Bagian ini dieksekusi jika kedua bagian **if-else statement** sebelumnya bernilai boolean **false (salah)**, atau pada dasarnya adalah blok kode **default** pada **switch-case statement** jika tidak ada kondisi yang terpenuhi. Bagian ini memiliki scope dari baris **10** hingga baris **12** dimana penutup blok (tanda ‘}’) berakhir.

Pada baris **11**, digunakan untuk menampilkan nilai string **“Selamat malam.”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

E. Contoh 1.5 – For Loop

Source code:

```
1 package com.example.JavaFundamental;
2
3 public class Contoh15 {
4     public static void main(String args[]){
5         for (int i = 0; i < 5; i++){
6             System.out.println(i);
7         }
8     }
9 }
10
```

Output:

```
> Task :app:run
0
1
2
3
4

BUILD SUCCESSFUL in 908ms
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Contoh15.java** tergabung pada package **com.example.JavaFundamental**

Pada baris **3**, dideklarasikan Class bernama **Contoh15**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **9**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Contoh15**.

Pada baris **4**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **4** hingga baris **8**.

Pada baris **5**, dideklarasikan pengulangan **for-loop statement** dengan scope dari baris **5** hingga baris **7**, nilai awal variabel **i** dengan tipe data **int** (4 bytes/32 bit) dan nilai **0**, kondisi **jika nilai variabel i kurang dari 5**, dan setiap pengulangan **nilai variabel i ditambah dengan 1 (i++)**. Sehingga, blok ini akan dieksekusi sebanyak **5 kali**, yaitu pada saat nilai **variabel i sama dengan 0, 1, 2, 3, 4**.

Pada baris **6**, digunakan untuk menampilkan nilai variabel **i**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**,

member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

F. Praktikum 1.1 – Dasar Pemrograman Java

Source code:

```
1 package com.example.JavaFundamental;
2 import java.util.Scanner;
3
4 public class Praktikum11 {
5     public static float volumeBalok(float panjang, float lebar, float tinggi){
6         float volume = panjang * lebar * tinggi;
7         return volume;
8     }
9     public static void main(String args[]){
10         float panjang, lebar, tinggi;
11         Scanner console = new Scanner(System.in);
12
13         System.out.print("Masukkan panjang balok : ");
14         panjang = console.nextFloat();
15
16         System.out.print("Masukkan lebar balok : ");
17         lebar = console.nextFloat();
18
19         System.out.print("Masukkan tinggi balok : ");
20         tinggi = console.nextFloat();
21
22         if(panjang == 0 || lebar == 0 || tinggi == 0){
23             System.out.println("Input keliru! Nilai input tidak boleh nol!");
24         } else {
25             System.out.print("Volume balok adalah: ");
26             System.out.println(volumeBalok(panjang, lebar, tinggi));
27         }
28     }
29 }
```

Output:

```
> Task :app:run
Masukkan panjang balok : 11,5
Masukkan lebar balok : 7,5
Masukkan tinggi balok : 5
Volume balok adalah: 431.25

BUILD SUCCESSFUL in 14s
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Praktikum11.java** tergabung pada package **com.example.JavaFundamental**

Pada baris **2**, class dari package **java.util.Scanner** dimasukkan (import) kedalam project **Praktikum11.java**. Semua fungsi/method yang ada pada class ini dapat diakses dalam file ini.

Pada baris **3**, dideklarasikan Class bernama **Praktikum11**, class ini mencakup semua kode yang ada pada baris **4** hingga baris **29**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Praktikum11**.

Pada baris **5**, dideklarasikan method **volumeBalok** dengan 3 parameter yaitu **panjang**, **lebar**, dan **tinggi** (semua dengan tipe data **float (signed, 4 bytes/32 bit)**). Method ini memiliki scope dari baris **5** hingga baris **8**.

Pada baris **6**, dideklarasikan variabel **volume** dengan tipe data **float (4 bytes/32 bit)** dan nilai hasil dari operasi perkalian variabel **panjang**, **lebar** dan **tinggi**.

Pada baris **7**, nilai dari variabel **volume** digunakan sebagai nilai balik atau **return value** dari method **volumeBalok**.

Pada baris **9**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **9** hingga baris **28**.

Pada baris **10**, dideklarasikan variabel **panjang**, **lebar** dan **tinggi** dengan tipe data **float (signed, 4 bytes/32 bit)**.

Pada baris **11**, dideklarasikan object **console** dengan class **Scanner** dan parameter **System.in**. Baris ini mendeklarasikan object **console**, yang akan digunakan untuk menangkap **console user-input** (data yang masuk melalui **System.in**).

Pada baris **13**, digunakan untuk menampilkan nilai string “**Masukkan panjang balok:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **14**, variabel **panjang** didefinisikan dengan menangkap **console user-input** yang dikonversi (**casting**) menjadi nilai **float**.

Pada baris **16**, digunakan untuk menampilkan nilai string “**Masukkan lebar balok:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **17**, variabel **lebar** didefinisikan dengan menangkap **console user-input** yang dikonversi (**casting**) menjadi nilai **float**.

Pada baris **19**, digunakan untuk menampilkan nilai string “**Masukkan tinggi balok:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **20**, variabel **tinggi** didefinisikan dengan menangkap **console user-input** yang dikonversi (**casting**) menjadi nilai **float**.

Pada baris **22**, dideklarasikan percabangan **if-else statement** dengan kondisi **jika nilai variabel panjang sama dengan 0** atau **jika nilai variabel lebar sama dengan 0** atau **jika nilai variabel tinggi sama dengan 0**. Jika kondisi ini menghasilkan nilai boolean **true (benar)**, maka kode dalam blok baris **22** hingga baris **24** akan dieksekusi, jika tidak maka kode dalam blok baris **24** hingga baris **27** akan dieksekusi.

Pada baris **23**, digunakan untuk menampilkan nilai string **“Input keliru! Nilai input tidak boleh nol!”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **25**, digunakan untuk menampilkan nilai string **“Volume balok adalah: ”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **26**, digunakan untuk menampilkan hasil eksekusi dari pemanggilan fungsi **volumeBalok** dengan argumen variabel **panjang**, **lebar**, dan **tinggi**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

G. Soal Tantangan

Buatlah sebuah program sederhana yang dapat digunakan untuk menghitung volume 3 jenis bangun ruang (Balok, Kubus, dan Kerucut). Pada program tersebut user harus memilih terlebih dahulu bangun ruang apa yang volumenya ingin diketahui.

Source code:

```
1 package com.example.JavaFundamental;
2 import java.util.Scanner;
3
4 public class Tantangan {
5     public static float volumeBalok(float panjang, float lebar, float tinggi){
6         float volume = panjang * lebar * tinggi;
7         return volume;
8     }
9
10    public static float volumeKubus(float sisi){
11        float volume = sisi * sisi * sisi;
12        return volume;
13    }
14
15    public static double volumeKerucut(float jarijari, float tinggi){
16        final double PI = 3.141592653d;
17        double volume = (PI * jarijari * jarijari * tinggi) / 3;
18        return volume;
19    }
20    public static void main(String args[]){
21        Scanner console = new Scanner(System.in);
22        while(true){
23            System.out.println("=====MENGHITUNG VOLUME BANGUN RUANG=====");
24            System.out.println("Pilih jenis bangun ruang: ");
25            System.out.println("1. Balok");
26            System.out.println("2. Kubus");
27            System.out.println("3. Kerucut");
28
29            System.out.println("4. Keluar dari aplikasi ini");
30            System.out.println("=====");
31            System.out.print("Masukkan pilihan: ");
32            int pilihan = console.nextInt();
33
34            if(pilihan == 1){
35                float panjang, lebar, tinggi;
36                System.out.print("Masukkan panjang Balok: ");
37                panjang = console.nextFloat();
38
39                System.out.print("Masukkan lebar Balok: ");
40                lebar = console.nextFloat();
41
42                System.out.print("Masukkan tinggi Balok: ");
43                tinggi = console.nextFloat();
44
45                System.out.print("Volume balok adalah: ");
46                System.out.println(volumeBalok(panjang,lebar,tinggi));
47                System.out.println();
48            } else if (pilihan == 2){
49                float sisi;
50                System.out.print("Masukkan sisi kubus: ");
51                sisi = console.nextFloat();
52
53                System.out.print("Volume kubus adalah: ");
54                System.out.println(volumeKubus(sisi));
55                System.out.println();
56            }
57        }
58    }
59 }
```

```

55     } else if (pilihan == 3){
56         float jarijari, tinggi;
57         System.out.print("Masukkan jari-jari Kerucut: ");
58         jarijari = console.nextFloat();
59
60         System.out.print("Masukkan tinggi Kerucut: ");
61         tinggi = console.nextFloat();
62
63         System.out.print("Volume Kerucut adalah: ");
64         System.out.println(volumeKerucut(jarijari, tinggi));
65         System.out.println();
66     } else if (pilihan == 4){
67         System.out.println("Terima kasih telah menggunakan aplikasi ini.");
68         break;
69     } else {
70         System.out.println("Pilihan tidak valid!");
71     }
72 }
73
74 }
75 }

```

Output:

```

> Task :app:run
=====MENGHITUNG VOLUME BANGUN RUANG=====
Pilih jenis bangun ruang:
1. Balok
2. Kubus
3. Kerucut
4. Keluar dari aplikasi ini
=====
Masukkan pilihan: 1
Masukkan panjang Balok: 9,5
Masukkan lebar Balok: 4,7
Masukkan tinggi Balok: 2
Volume balok adalah: 89.299995

```

```

=====MENGHITUNG VOLUME BANGUN RUANG=====
Pilih jenis bangun ruang:
1. Balok
2. Kubus
3. Kerucut
4. Keluar dari aplikasi ini
=====
Masukkan pilihan: 2
Masukkan sisi kubus: 7,5
Volume kubus adalah: 421.875

```

```

=====MENGHITUNG VOLUME BANGUN RUANG=====
Pilih jenis bangun ruang:
1. Balok
2. Kubus
3. Kerucut
4. Keluar dari aplikasi ini
=====
Masukkan pilihan: 3
Masukkan jari-jari Kerucut: 14
Masukkan tinggi Kerucut: 7,5
Volume Kerucut adalah: 1539.38039997

```

```

=====MENGHITUNG VOLUME BANGUN RUANG=====
Pilih jenis bangun ruang:
1. Balok
2. Kubus
3. Kerucut
4. Keluar dari aplikasi ini
=====
Masukkan pilihan: 5
Pilihan tidak valid!
=====MENGHITUNG VOLUME BANGUN RUANG=====
Pilih jenis bangun ruang:
1. Balok
2. Kubus
3. Kerucut
4. Keluar dari aplikasi ini
=====
Masukkan pilihan: 4
Terima kasih telah menggunakan aplikasi ini.

BUILD SUCCESSFUL in 50s
2 actionable tasks: 1 executed, 1 up-to-date

```

Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Tantangan.java** tergabung pada package **com.example.JavaFundamental**

Pada baris **2**, class dari package **java.util.Scanner** dimasukkan (import) kedalam project **Tantangan.java**. Semua fungsi/method yang ada pada class ini dapat diakses dalam file ini.

Pada baris **3**, dideklarasikan Class bernama **Tantangan**, class ini mencakup semua kode yang ada pada baris **4** hingga baris **75**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Tantangan**.

Pada baris **5**, dideklarasikan method **volumeBalok** dengan 3 parameter yaitu **panjang**, **lebar**, dan **tinggi** (semua dengan tipe data **float (signed, 4 bytes/32 bit)**). Method ini memiliki scope dari baris **5** hingga baris **8**.

Pada baris **6**, dideklarasikan variabel **volume** dengan tipe data **float (4 bytes/32 bit)** dan nilai hasil dari operasi perkalian variabel **panjang**, **lebar** dan **tinggi**.

Pada baris **7**, nilai dari variabel **volume** digunakan sebagai nilai balik atau **return value** dari method **volumeBalok**.

Pada baris **10**, dideklarasikan method **volumeKubus** dengan 1 parameter yaitu **sisi** dengan tipe data **float (signed, 4 bytes/32 bit)**. Method ini memiliki scope dari baris **10** hingga baris **13**.

Pada baris **11**, dideklarasikan variabel **volume** dengan tipe data **float (4 bytes/32 bit)** dan nilai hasil dari operasi perkalian variabel **sisi** sebanyak 3 kali.

Pada baris **12**, nilai dari variabel **volume** digunakan sebagai nilai balik atau **return value** dari method **volumeKubus**.

Pada baris **15**, dideklarasikan method **volumeKerucut** dengan 2 parameter yaitu **jarijari** dan **tinggi** (semua dengan tipe data **float (signed, 4 bytes/32 bit)**). Method ini memiliki scope dari baris **15** hingga baris **19**.

Pada baris **16**, dideklarasikan sebuah konstanta (constant/final) **PI** dengan tipe data **double (signed, 8 bytes/64 bit)** dengan nilai **3.141592653d**. Huruf “d” pada akhir nilai menandakan ke compiler bahwa nilai ini adalah data double (64 bit).

Pada baris **17**, dideklarasikan variabel **volume** dengan tipe data **double (8 bytes/64 bit)** dan nilai hasil dari operasi perkalian konstanta **PI** dengan variabel **jarijari** sebanyak 2 kali, dan variabel **tinggi** (**PI * jarijari * jarijari * tinggi**) lalu dibagi dengan **3**.

Pada baris **18**, nilai dari variabel **volume** digunakan sebagai nilai balik atau **return value** dari method **volumeKerucut**.

Pada baris **20**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **20** hingga baris **74**.

Pada baris **21**, dideklarasikan object **console** dengan class **Scanner** dan parameter **System.in**. Baris ini mendeklarasikan object **console**, yang akan digunakan untuk menangkap **console user-input** (data yang masuk melalui **System.in**).

Pada baris **22**, dideklarasikan pengulangan **while-loop statement** dengan kondisi **true**. Sehingga blok kode dalam statement ini akan terus berjalan hingga ada interupsi program atau kode **break**. Scope kode ini mulai dari baris **23** hingga baris **72**.

Pada baris **23**, digunakan untuk menampilkan nilai string “=====MENGHITUNG VOLUME BANGUN RUANG=====”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **24**, digunakan untuk menampilkan nilai string “**Pilih jenis bangun ruang:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **25**, digunakan untuk menampilkan nilai string **“1. Balok”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **26**, digunakan untuk menampilkan nilai string **“2. Kubus”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **27**, digunakan untuk menampilkan nilai string **“3. Kerucut”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **28**, digunakan untuk menampilkan nilai string **“4. Keluar dari aplikasi ini”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **29**, digunakan untuk menampilkan nilai string **“=====”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **30**, digunakan untuk menampilkan nilai string **“Masukkan pilihan: ”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **31**, dideklarasikan variabel **pilihan** dengan tipe data **int (signed, 4 bytes/32 bit)** dengan nilai dari **console user-input** yang dikonversi (**casting**) ke tipe data **int**.

Pada baris **33**, dideklarasikan percabangan **if-else statement** dengan kondisi **jika variabel pilihan sama dengan nilai 1**. Jika kondisi ini terpenuhi, maka kode dalam blok baris **33** hingga baris **47**. Jika tidak maka, kode dalam blok tersebut tidak akan dieksekusi dan lanjut ke kode setelah baris penutup.

Pada baris **34**, dideklarasikan variabel **panjang**, **lebar**, dan **tinggi** dengan tipe data **float (signed, 4 bytes/32 bit)**.

Pada baris **35**, digunakan untuk menampilkan nilai string “**Masukkan panjang balok:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **36**, variabel **panjang** didefinisikan dengan menangkap **console user-input** yang dikonversi (**casting**) menjadi nilai **float**.

Pada baris **38**, digunakan untuk menampilkan nilai string “**Masukkan lebar balok:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **39**, variabel **lebar** didefinisikan dengan menangkap **console user-input** yang dikonversi (**casting**) menjadi nilai **float**.

Pada baris **41**, digunakan untuk menampilkan nilai string “**Masukkan tinggi balok:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **42**, variabel **tinggi** didefinisikan dengan menangkap **console user-input** yang dikonversi (**casting**) menjadi nilai **float**.

Pada baris **44**, digunakan untuk menampilkan nilai string “**Volume balok adalah:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **45**, digunakan untuk menampilkan hasil eksekusi dari pemanggilan fungsi **volumeBalok** dengan argumen variabel **panjang**, **lebar**, dan **tinggi**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **46**, digunakan untuk menampilkan baris kosong pada console. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **47**, dideklarasikan **else-if statement** dengan kondisi **jika variabel pilihan sama dengan nilai 2**. Baris ini hanya akan dieksekusi jika **if-else statement** sebelumnya (baris **33**) bernilai **false**. Jika kondisi ini terpenuhi, maka kode dalam blok

baris **47** hingga baris **55** akan dieksekusi. Jika tidak, maka kode dalam blok tersebut tidak akan dieksekusi dan lanjut ke kode setelah baris penutup.

Pada baris **48**, dideklarasikan variabel **sisi** dengan tipe data **float (signed, 4 bytes/32 bit)**.

Pada baris **49**, digunakan untuk menampilkan nilai string “**Masukkan sisi kubus:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **50**, variabel **sisi** didefinisikan dengan menangkap **console user-input** yang dikonversi (**casting**) menjadi nilai **float**.

Pada baris **52**, digunakan untuk menampilkan nilai string “**Volume kubus adalah:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **53**, digunakan untuk menampilkan hasil eksekusi dari pemanggilan fungsi **volumeKubus** dengan argumen variabel **sisi**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **54**, digunakan untuk menampilkan baris kosong pada console. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **55**, dideklarasikan **else-if statement** dengan kondisi **jika variabel pilihan sama dengan nilai 3**. Baris ini hanya akan dieksekusi jika **if-else statement** sebelumnya (baris **33**) **dan** jika **else-if statement** sebelumnya (baris **47**) bernilai **false**. Jika kondisi ini terpenuhi, maka kode dalam blok baris **55** hingga baris **66** akan dieksekusi. Jika tidak, maka kode dalam blok tersebut tidak akan dieksekusi dan lanjut ke kode setelah baris penutup.

Pada baris **56**, dideklarasikan variabel **jarijari** dan **tinggi** dengan tipe data **float (signed, 4 bytes/32 bit)**.

Pada baris **57**, digunakan untuk menampilkan nilai string “**Masukkan jari-jari Kerucut:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada

pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **58**, variabel **jarijari** didefinisikan dengan menangkap **console user-input** yang dikonversi (**casting**) menjadi nilai **float**.

Pada baris **60**, digunakan untuk menampilkan nilai string “**Masukkan tinggi Kerucut:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **61**, variabel **tinggi** didefinisikan dengan menangkap **console user-input** yang dikonversi (**casting**) menjadi nilai **float**.

Pada baris **63**, digunakan untuk menampilkan nilai string “**Volume Kerucut adalah:** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **print()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **64**, digunakan untuk menampilkan hasil eksekusi dari pemanggilan fungsi **volumeKerucut** dengan argumen variabel **jarijari** dan **tinggi**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **65**, digunakan untuk menampilkan baris kosong pada console. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **66**, dideklarasikan **else-if statement** dengan kondisi **jika variabel pilihan sama dengan nilai 3**. Baris ini hanya akan dieksekusi jika **if-else statement** sebelumnya (baris **33**) **dan** jika **else-if statement** sebelumnya (baris **47**) **dan** jika **else-if statement** sebelumnya (baris **55**) bernilai **false**. Jika kondisi ini terpenuhi, maka kode dalam blok baris **66** hingga baris **69** akan dieksekusi. Jika tidak, maka kode dalam blok tersebut tidak akan dieksekusi dan lanjut ke kode setelah baris penutup.

Pada baris **67**, digunakan untuk menampilkan nilai string “**Terima kasih telah menggunakan aplikasi ini.** ”. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **68**, keyword **break** digunakan untuk menghentikan **while-loop statement** (baris **22**).

Pada baris **69**, dideklarasikan **else statement** dimana bagian ini akan dieksekusi jika semua **if-else statement** sebelumnya bernilai **false**. Hal ini sama dengan **default** pada **switch-case statement**. Bagian ini mulai dari baris **69** hingga baris **71**.

Pada baris **70**, digunakan untuk menampilkan nilai string **“Pilihan tidak valid”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.