

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
MODUL 3 ENCAPSULATION AND FILE(S)



NAMA : Nadia Ramadhani

NIM : 105219018

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN ILMU KOMPUTER
UNIVERSITAS PERTAMINA
2021

1. Pendahuluan

a. Intisari

Pada praktikum kali ini membuat program untuk memasukkan nama, nim, tinggi dan berat badan mahasiswa dan data tersebut disimpan ke dalam sebuah file txt. Selain itu, program tersebut dapat menghitung dan menampilkan BMI dari data yang telah dimasukkan.

b. Rumusan Masalah

1. Apakah enkapsulasi dapat digunakan untuk menghitung suatu nilai?
2. Bagaimana cara menampilkan isi file txt yang telah dibuat sebelumnya pada output?

2. Pembahasan

Pertama-tama class Mahasiswa dibuat terlebih dahulu dengan *private* modifier variable, kemudian membuat method agar variable tersebut dapat mengakses data yang nanti akan diberikan. Kemudian pada main class, file txt bmi dibentuk dengan menggunakan *Printwriter*, lalu isi dari file tersebut dicetak. *Try* dan *catch* digunakan untuk mengecek apakah terdapat error. Jika ada *error*, maka program akan berjalan ke *catch* dan *printStackTrace* akan memberi tahu letak exception atau *error* tersebut.

```
1  public class Mahasiswa{
2      private String nama;
3      private String nim;
4      private double tb;
5      private double bb;
6      private double bmi;
7
8      public void setMahasiswa(String nama, String nim, double tb, double bb) {
9          this.nama = nama;
10         this.nim = nim;
11         this.tb = tb;
12         this.bb = bb;
13         this.bmi = (tb + bb)/2;
14     }
15
16     public String getName() {
17         return nama;
18     }
19
20     /*public void setName(String nama){
21         this.nama = nama;
22     }*/
23
24     public String getNim(){
25         return nim;
26     }
27
28     /*public void setNim(String nim){
29         this.nim = nim;
30     }*/
31
32     public double getTB(){
```

```
33         return tb;
34     }
35
36     /*public void setTB(double tb){
37         this.tb = tb;
38     }*/
39
40     public double getBB(){
41         return bb;
42     }
43
44     /*public void setBB(double bb){
45         this.bb = bb;
46     }*/
47
48     /* public double setbmi(double bmi){
49         return this.bmi = (tb + bb)/2;
50     }*/
51
52     public double getbmi(){
53         return bmi;
54     }
55
56 }
```

Untuk membaca file txt ke dalam command, pertama-tama masukkan file tersebut dengan menggunakan `File inputFile = new File ("filename.txt")`, kemudian buat sebuah objek menggunakan class `Scanner`. Setelah itu, gunakan method `Scanner`, pada praktikum kali ini digunakan `while`, karena selama objek dalam class `Scanner` tersebut masih memiliki kalimat, program akan menampilkan isi outputnya, jika tidak maka class tersebut akan ditutup. `Try` dan `catch` juga digunakan untuk mengecek jika terdapat *error*.

```

1  import java.io.*;
2  //import java.util.Scanner;
3  import java.util.Scanner;
4
5  public class Main {
6      Run | Debug
7      public static void main(String[] args){
8          Mahasiswa data1 = new Mahasiswa();
9          data1.setMahasiswa("A", "001", 178, 66);
10
11          Mahasiswa data2 = new Mahasiswa();
12          data2.setMahasiswa("B", "002", 180, 45);
13
14          try {
15              PrintWriter outputFile = new PrintWriter("bmi.txt");
16              outputFile.println("Nama = " + data1.getNama());
17              outputFile.println("NIM = " + data1.getNim());
18              outputFile.println("Tinggi Badan = " + data1.getTB());
19              outputFile.println("Berat Badan = " + data1.getBB());
20              outputFile.println("BMI = " + data1.getBmi());
21              outputFile.println("-----");
22              outputFile.println("Nama = " + data2.getNama());
23              outputFile.println("NIM = " + data2.getNim());
24              outputFile.println("Tinggi Badan = " + data2.getTB());
25              outputFile.println("Berat Badan = " + data2.getBB());
26              outputFile.println("BMI = " + data2.getBmi());
27              outputFile.close();
28          } catch (FileNotFoundException e){
29              e.printStackTrace();
30          }
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

Nama = A
NIM = 001
Tinggi Badan = 178.0
Berat Badan = 66.0
BMI = 122.0
-----
Nama = B
NIM = 002
Tinggi Badan = 180.0
Berat Badan = 45.0
BMI = 112.5

[Done] exited with code=0 in 1.522 seconds

```

```

Nama = A
NIM = 001
Tinggi Badan = 178.0
Berat Badan = 66.0
BMI = 122.0
-----
Nama = B
NIM = 002
Tinggi Badan = 180.0
Berat Badan = 45.0
BMI = 112.5

```

```

32      try {
33          File inputFile = new File("bmi.txt");
34          Scanner in = new Scanner(inputFile);
35          while (in.hasNextLine()){
36              String kalimat = in.nextLine();
37              System.out.println(kalimat);
38          }
39          in.close();
40      } catch (FileNotFoundException e){
41          e.printStackTrace();
42      }
43  }
44  }
45

```

3. Kesimpulan

- a. Enkapsulasi adalah menyembunyikan data dengan modifier *private* hanya bisa diakses dari dalam kelas tersebut agar data menjadi lebih aman. Untuk mengakses data yang di *private* tersebut, digunakan *setter* dan *getter* dengan modifier *public*. Method *setter* menggunakan *void* karena hanya digunakan untuk menerima data, sedangkan method *getter* menggunakan *return* untuk mengembalikan data yang diterima. Untuk membuat file txt dapat menggunakan *Printwriter* dan menggunakan *try* dan *catch* untuk mengecek apakah terdapat error.
- b. Enkapsulasi dapat digunakan untuk menghitung suatu nilai yang telah diberikan, dengan cara menuliskan *this.nilai* pada variable *void* dan menuliskan *getter* / *returnnya* pada variable *public* sesuai dengan tipe datanya, tidak perlu menggunakan *setter*. Dengan menggunakan method *Scanner*, isi sebuah file txt dapat ditampilkan ke dalam output program.