LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 3

NAMA : M TAQIYYUDDIN F

NIM : 105219039

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

1. Pendahuluan

a. Inti Sari

Pada praktikum di modul ini, dibuat suatu program tentang pemahaman enkapsulasi dan pengolahan file pada class, serta penggunaan method try dan catch di java. Enkapsulasi adalah suatu metode dalam pengolahan objek atau class yang menyimpan atau melindungi data (baik method, atribut, dst) di suatu class dalam hal pengaksesan class lainnya. Cara class lain mengakses data tersebut menggunakan suatu method yang telah disediakan oleh class tersebut. Operasi file adalah suatu pengolahan data yang memanipulasi isi pada file yang ingin dimodifikasi. Teknik yang digunakan pada praktikum ini adalah membaca maupun menulis file, dengan menggunakan method try dan catch yang menghandle proses pengolahan file tersebut.

b. Rumusan Masalah

- Bagaimana proses enkapsulasi bekerja ?
- Bagaimana pengolahan operasi file dilakukan ?
- Apa kegunaan dari method try dan catch ?

2. Pembahasan

```
public class DataDiri {
    private String nama;
    private String nim;
    private int tinggiBadan;
    private double beratBadan;
```

Gambar 2.1. Deklarasi Class Beserta Attribute-nya

Pada percobaan ini, data para mahasiswa akan disimpan untuk dimaksukkan kedalam suatu file. Data tersebut didapat dari objek yang dibuat berdasarkan class DataDiri yang berisi identitas dari tiap mahasiswa tersebut. Di class ini, dimiliki atribut berupa nama, nim, tinggiBadan, dan beratBadan, dimana penggunaannya menyimpan suatu data berdasarkan nama masing-masing atribut disetiap objek mahasiswa.

```java
public DataDiri(String nama, String nim, int tinggiBadan, double beratBadan) {
    this.nama = nama;
    this.nim = nim;
    this.tinggiBadan = tinggiBadan;
    this.beratBadan = beratBadan;
}
```

Gambar 2.2. Pembuatan Constructor Pada Class

Pada percobaan ini, dibuat suatu constructor sebagai method pembuatan class tersebut, dimana dideklarasikan semua nilai atribut bersamaan dibuatnya objek tersebut dengan nilai dari argument yang diberikan pada constructor tersebut.

```java
public String getNama() {
    return this.nama;
}

public String getNim() {
    return this.nim;
}

public int getTinggiBadan() {
    return this.tinggiBadan;
}

public double getBeratBadan() {
    return this.beratBadan;
}
```

Gambar 2.3. Method Getter Untuk Mengambil Nilai Atribut

Pada percobaan ini, digunakan method untuk mendapatkan nilai atribut pada objek pada method masing-masing, seperti getNama() untuk mendapatkan nilai atribut nama, getNim() untuk mendapatkan nilai nim, dan seterusnya.

```java
public double BMI() {
    return (this.tinggiBadan + this.beratBadan) / 2.0;
}
```

Gambar 2.4. Method Untuk Mendapatkan Nilai BMI

Pada percobaan ini, digunakan method untuk mendapatkan nilai bmi pada tiap objek mahasiswa, dimana menggunakan nilai atribut tinggi badan ditambah dengan nilai atribut berat badan, lalu nilai penjumlahan yang dibagi dua tersebut diberikan ke pemanggil method ini.

```
import java.io.*;
import java.util.*;
```

Gambar 2.5. Package Operasi File dan Proses Input

Pada percobaan ini, digunakan package java.io untuk memproses Operasi file, baik itu membaca maupun menulis file. Sedangkan java.util terdapat suatu method penanganan sistem seperti menerima input dari sistem.

```
public static void main(String[] args) {
    DataDiri[] mahasiswa = new DataDiri[2];
    Scanner input = new Scanner(System.in);

    String nama;
    String nim;
    int tinggiBadan;
    double beratBadan;
```

Gambar 2.6. Initialisasi Array Objek, Method Input, Serta Variabel Pada Main

Pada percobaan ini, dibuat kumpulan objek mahasiswa berdasarkan class DataDiri, dalam bentuk array yang menampung dua objek. Lalu, dibuat suatu objek berdasarkan objek scanner dari package java.util bernama input, untuk menggunakan berbagai method menerima input data dari sistem. Terakhir, digunakan suatu variabel nama dan nim bertipe data string, tinggiBadan bertipe data int, serta beratBadan bertipe data double untuk menyimpan data yang sedang diinput.

```
for(int i = 0; i < mahasiswa.length; i++) {
    System.out.println("Mahasiwa ke-" + (i+1));
    System.out.print("Nama\t\t: ");
    nama = input.nextLine();
    System.out.print("NIM\t\t: ");
    nim = input.nextLine();
    System.out.print("Tinggi Badan\t: ");
    tinggiBadan = input.nextInt();
    input.nextLine();
    System.out.print("Berat Badan\t: ");
    beratBadan = input.nextDouble();
    input.nextLine();
    mahasiswa[i] = new DataDiri(nama, nim, tinggiBadan, beratBadan);
    System.out.println("========================");
}
input.close();
```

Gambar 2.7. Proses Input Data Pada Objek

Pada percobaan ini, ditampilkan output untuk memberi informasi mahasiswa keberapa data ini ditulis, serta ditahap input, nilai apakah yang diminta secara berurut dari nama, nim, tinggi badan, serta berat badan. Disetiap tahap, digunakan suatu variabel untuk menerima inputan tersebut, menggunakan method dari objek input yaitu nextLine() untuk menerima input berupa string, nextInt() dan nextDouble() untuk menerima input masing-masing berupa int dan double. Dalam proses input dengan tipe data numerik, setelah proses input data, digunakan method nextLine() untuk memindahkan cursor

untuk menerima proses input selanjutnya. Dari variabel penerima input tersebut, digunakan untuk nilai argument saat pembuatu objek mahasiswa.

```
try {
    PrintWriter fileInput = new PrintWriter("bmi.txt");
    for(int i = 0; i < mahasiswa.length; i++) {
        fileInput.println("Mahasiwa ke-" + (i+1));
        fileInput.println("Nama\t\t: " + mahasiswa[i].getNama());
        fileInput.println("NIM\t\t: " + mahasiswa[i].getNim());
        fileInput.println("Tinggi Badan\t: " + mahasiswa[i].getTinggiBadan());
        fileInput.println("Berat Badan\t: " + mahasiswa[i].getBeratBadan());
        fileInput.println("BMI\t\t: " + mahasiswa[i].BMI());
        fileInput.println("");
    }
    fileInput.close();
} catch(FileNotFoundException e) {
    e.printStackTrace();
}
```

Gambar 2.8. Menulis Data Objek Kedalam File

Pada percobaan ini, digunakan class PrintWriter pada package java.io yang digunakan oleh objek fileInput. Untuk memproses penulisan data pada file bmi.txt. Setelah itu, digunakan proses perulangan yang membaca semua objek pada array, untuk mendapatkan nilai dari atribut tiap objek yang akan ditulis kedalam file. Nilai tersebut diambil menggunakan method getter dari masing-masing atribut, beserta method BMI yang menerima nilai bmi dengan kalkulasi pada method tersebut. Selanjutnya, proses pembacaan keseluruhan data objek diakhiri dan proses penulisan data pada file dihentikan dengan method close(). Serangkaian proses tersebut ditampung di teknik try, jika terdapat exception dimana file tersebut tidak ditemukan pada folder source code tersebut. Maka akan memasuki proses catch dimana dengan argument FileNotFoundException (digunakan singkatan e) penanganan exception tersebut dijalankan method printStackTrace() untuk mengetahui line berapa proses tidak dapat dijalankan.

```
try {
    File fileOutput = new File("bmi.txt");
    Scanner readFile = new Scanner(fileOutput);
    System.out.println("\nHasil Output Dari Baca File :");
    while(readFile.hasNextLine()) {
        System.out.println(readFile.nextLine());
    }
    readFile.close();
} catch(FileNotFoundException e) {
    e.printStackTrace();
}
```

Gambar 2.9. Membaca Data Dari File

Pada percobaan ini, digunakan Teknik try dan catch yang sama, dengan proses yang diantisipasi exceptionnya adalah serangkaian proses dari pembuatan objek fileOutput dengan class File yang berfungsi membaca isi file dari bmi.txt, objek readFile untuk menerima inputan dari pembacaan isi file tersebut, beserta menulis output dari setiap line pada file yang diakhiri dengan menutup proses membaca isi file tersebut.

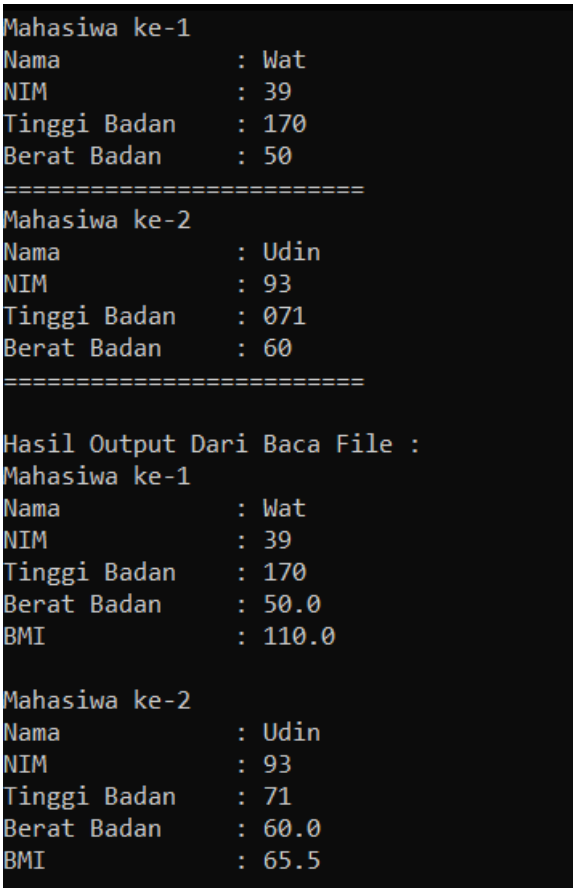
3. Kesimpulan

- Proses enkapsulasi pada praktikum ini digunakan class DataDiri untuk menyimpan data dari tiap objek mahasiswa tersebut, dengan attribute dari class tersebut. Objek dapat diberikan suatu nilai pada saat pembuatan / constructor objek, dari parameter yang telah diberikan oleh variabel-variabel yang menerima input dari sistem. Untuk mendapatkan data yang disimpan tersebut, digunakan metode getter untuk mendapatkan nilai setiap attribute yang dimiliki oleh objek tersebut

- Pengolahan operasi file tersebut dijalankan menggunakan class PrintWriter untuk menulis data kedalam suatu file dan File untuk mendapatkan data pada suatu file yang didapat dari penggunaan package java.io. untuk proses menulis data, digunakan method println() untuk menulis data kedalam file per baris, sedangkan proses membaca file juga digunakan class Scanner untuk mendapatkan data dari file yang telah dibaca, lalu digunakan method hasNextLine() untuk pengecekan keseluruhan file dan nextLine() untuk memberikan output file per barisnya, yang masing-masing operasi file diakhiri dengan method close() untuk menghentikan proses operasi file tersebut.

- Teknik try and catch digunakan untuk mengantisipasi exception pada suatu program, dimana pecahan dari kumpulan line code dirangkup dalam method try, exception yang diterima pada argument di method catch akan dicek jika code pada try tersebut tidak jalan, akan dipindah pada proses method catch. Sedangkan apabila tidak terdapat expection yang terjadi, maka proses pada try dapat berjalan seperti umumnya.

4. Tes Akhir



Gambar 4.1. Hasil Output Dari Program Tersebut

Folder package terdapat pada file terkompresi.