MODUL VII

EXCEPTION HANDLING DAN *GRAPHICAL USER INTERFACE*

1. Tujuan

a. Mahasiswa melakukan improvisasi kode program yang melibatkan mekanisme exception dan assertion.
b. Mahasiswa mampu menulis method yang mengandung exception.
c. Mahasiswa mampu mengimplementasikan try-catch untuk permasalahan exception.
d. Menggunakan assertion untuk meningkatkan pendeteksian bug selama pengembangan program.
e. Mahasiswa mampu mendefinisikan array pada pemrograman objek secara 1D dan 2D

2. Latihan praktikum

Buat file .java di editor masing-masing, dan lakukan latihan pemrograman yang ditunjukkan setiap nomor.

a. Pengambilan exception

Sebuah exception didefinisikan sebagai kondisi error yang dapat terjadi selama eksekusi program. Program di bawah ini merupakan contoh ilustrasi mekanisme exception-handling. Definisi pertama dari program ini adalah program tanpa exception

```
import java.util.*;
class InputUmurMain {
    public static void main(String[] args) {
        GregorianCalendar today;
        int age, thisYear, bornYr;
        String answer;
        Scanner scanner = new Scanner(System.in);
        InputUmur input = new InputUmur( );
        age = input.getAge("Sudah berpakah umurmu? ");
        today = new GregorianCalendar( );
        thisYear = today.get(Calendar.YEAR);
        bornYr = thisYear - age;



        System.out.print("Apakah kamu sudah ultah tahun ini? (Y atau T)");
        answer = scanner.next();
        if (answer.equals("T") || answer.equals("t") ) {
            bornYr--;
        }
        System.oucvbdgfdt.println("\nkamu lahir tahun  " + bornYr);
    }
}
```

```
import java.util     .*;
class InputUmur {
private static final String DEFAULT_MESSAGE = "Umurmu: ";
private Scanner scanner;
    public InputUmur( ) {
      scanner = new Scanner(System.in);
    }
    public int getUmur() {
      return getUmur(DEFAULT_MESSAGE);
    }
    public int getUmur(String prompt) {
      System.out.print(prompt);
      int umur = scanner.nextInt();
      return umur;
    }
}
```

Untk mencapai mekanisme exception yang tepat, ketika sebuah method menjalankan sebuah exception maka method tersebut disebut sebagai *exception thrower* yang terdiri dari dua jenis yaotu: *catcher* dan *propagator*. *Exception catcher* melakukan pencocokan blok *catch* untuk exception yang dieksekusi, sedangkan *exception propagator* tidak terjadi demikian.

```
import javax.swing.*;
class InputUmur2 {
    private static final String DEFAULT_MESSAGE = "Umur kamu:";
    private static final int DEFAULT_LOWER_BOUND = 0;
    private static final int DEFAULT_UPPER_BOUND = 99;
    private int lowerBound;
    private int upperBound;
    private Scanner scanner;
    public InputUmur2( ) {
        init(DEFAULT_LOWER_BOUND, DEFAULT_UPPER_BOUND);
    }
    public InputUmur2(int low, int high) throws IllegalArgumentException {
        if (low > high) {
            throw new IllegalArgumentException(
            " (" + low + ") " +
            "lebih besar daripada (" + high + ")");
        } else {
            init(low, high);
        }
    }
    public int getAge() throws Exception {
        return getAge(DEFAULT_MESSAGE);
    }
    public int getAge(String prompt) throws Exception {
        int age;
        while (true) {
            System.out.print(prompt);
        try {
            age = scanner.nextInt();
        if (age < lowerBound || age > upperBound) {
            throw new Exception("Input out of bound");
        }
            return age;
        } catch (InputMismatchException e) {
            scanner.next();
            System.out.println("Input salah.\n" +
```

```
            "Input harus dalam bentuk bilangan");
        }
        }
    }
    private void init(int low, int high) {
        lowerBound = low;
        upperBound = high;
        scanner = new Scanner(System.in);
    }
}
```

b. Assertions

Assertion pada Java merupakan fitur bahasa pemrograman yang digunakan untuk mendeteksi error logika di dalam sebuah program. Eksekusi pada assertion memiliki cara tersendiri untuk eksekusi yaitu **`java –ea <main class>`**.

```
    class AkunBank {
        private double saldo;
        public AkunBank(double saldoMasuk) {
            saldo = saldoMasuk;
        }
        public void deposit(double jumlah) {
            double saldoLama = saldo;
            saldo -= jumlah;
            assert saldo > saldoLama;
        }
        public void pembatalan(double jumlah) {
            double saldoLama = saldo;
            saldo -= jumlah;
            assert saldo < saldoLama;
        }
        public double getSaldo( ) {
            return saldo;
        }
    }
```

```
import javax.swing.*;
class AkunBankMain {
  public static void main(String[] args) {
      AkunBank acct = new AkunBank(200);
      acct.deposito(25);
      System.out.println("Saldo saat ini adalah " + acct.getSaldo());
  }
}
```

c. Array

a) Array dasar

```
    import java.util.*;
    class arrayDasar {
    public static void main (String[] args) {
            Scanner scanner = new Scanner (System.in);
            double[] rainfall = new double[12];
            double annualAverage,
            sum,
```

```
            difference;
            sum = 0.0;
            for (int i = 0; i < 12; i++) {
            System.out.print("Curah hujan untuk bulan: " + (i+1) + ": ");
            rainfall[i] = scanner.nextDouble();
            sum += rainfall[i];
            }
            annualAverage = sum / 12.0;
            System.out.format       ("Rata-rata curah hujan:%5.2f\n\n",
            annualAverage);
            for (int i = 0; i < 12; i++) {
            System.out.format("%3d", i+1); //month #

             System.out.format("%15.2f", rainfall[i]);

              difference = Math.abs( rainfall[i] - annualAverage );
             System.out.format("%15.2f\n", difference);
              }
          }
    }
```

b) Array object

```
    import java.util.*;
    class objectPersonArray {
    public static void main (String[] args) {
           Person[] person; //declare the person array
           person = new Person[5]; //and then create it
           //----------- buat Array dengan nama Person -------------------//
           String name, inpStr;
           int age;
           char gender;
           for (int i = 0; i < person.length; i++) {
           //pembacaan niai data
           System.out.print("Enter name: ");
           name = scanner.next();
           System.out.print("Enter age: ");
           age = scanner.nextInt();
           System.out.print("Enter gender: ");
           inpStr = scanner.next();
           gender = inpStr.charAt(0);
           //buat Person baru dan masukkan nilai
           person[i] = new Person( );
           person[i].setName ( name );
           person[i].setAge ( age );
           person[i].setGender( gender );
           }
           //-------------- Hitung umur rata-rata --------------//
           float sum = 0, averageAge;
           for (int i = 0; i < person.length; i++) {
           sum += person[i].getAge();
           }
          averageAge = sum / (float) person.length;
          System.out.println("Average age: " + averageAge);
          System.out.println("\n");
          Person youngest, oldest;
          youngest = oldest = person[0];
          for (int i = 1; i < person.length; i++) {
          if (person[i].getAge() < youngest.getAge()) {

          youngest = person[i];
```

```
          }
          else if (person[i].getAge() > oldest.getAge()) {
          oldest = person[i];
          }
      }
        System.out.println("Oldest : " + oldest.getName()
        + " is " + oldest.getAge() + " years old.");
        System.out.println("Youngest: " + youngest.getName()
         + " is " + youngest.getAge() + " years old.");
         System.out.print("Name to search: ");
         String searchName = scanner.next();
         int i = 0;
        while (i < person.length && //still more persons to search
         !person[i].getName().equals(searchName)) {
         i++;
         }
         if (i   == person.length) {
        System.out.println( searchName + " was not in the array" );
          } else {
          System.out.println("Found " + searchName + " at position " + i);
           }
           }
}
```

c) Array 2D

```
class array2D {
public static void main (String[] args) {
       double[][] payScaleTable = { {10.50, 12.00, 14.50, 16.75, 18.00},
                                    {20.50, 22.25, 24.00, 26.25, 28.00},
                                    {34.00, 36.50, 38.00, 40.35, 43.00},
                                    {50.00, 60.00, 70.00, 80.00, 99.99} };
        double sum = 0.0, average;
       for (int j = 0; j < 5; j++) {
       sum += payScaleTable[2][j];
       }
       average = sum / 5;
       System.out.println(" Average of Level 2 Employees: " + average );
       System.out.println("\n");

       double difference;
       for (int i = 0; i < 4; i++) {
       difference = payScaleTable[i][4] - payScaleTable[i][0];
       System.out.println("Pay difference at Grade Level " +
        i + " is " + difference);
        }
      System.out.println("\n");
      for (int i = 0; i < payScaleTable.length; i++) {
          for (int j = 0; j < payScaleTable[i].length; j++) {
             System.out.print( payScaleTable[i][j] + " " );
          }
          System.out.println("");
     }
    System.out.println("\n");
    for (int i = 0; i < payScaleTable.length; i++) {
          for (int j = 0; j < payScaleTable[i].length; j++) {
             payScaleTable[i][j] += 1.50;
             System.out.print(payScaleTable[i][j] + " ");
           }
         System.out.println("");
    }
  }
}
```

3. Tugas Praktikum

1) Diketahui pada class InputUmur2, method getAge menjalankan class exception dengan dapat memanfaatkan getMessage untuk menerima pesan error. Namun pesan error yang diberikan tersebut kurang informatif kepada client. Solusi yang dapat dipakai adalah dengan menggunakan subclass pada class exception. Subclass yang terdapat di class exception pada program yang dicontohkan mengandung tiga buah informasi: *lower bound* (batas bawah angka), *upper bound* (batas atas angka), dan nilai yang diinputkan oleh pengguna program. Program *subclass* dan *main class*nya ditunjukkan di bawah ini

```
class InputUmurException extends Exception {
    private static final String DEFAULT_MESSAGE = "Input out of bounds";
    private int lowerBound;
    private int upperBound;
    private int value;

    public InputUmurException(int low, int high, int input) {
        this(DEFAULT_MESSAGE, low, high, input);
    }
    public InputUmurException(String msg, int low, int high, int input) {
    super(msg);
    if (low > high) {
        throw new IllegalArgumentException();
    }
        lowerBound = low;
        upperBound = high;
        value = input;
    }
    public int lowerBound() {
        return lowerBound;
    }
    public int upperBound() {
        return upperBound;
    }
    public int value() {
        return value;
    }
}
```

Soal: buatlah class main untuk memfungsionalkan (mengaktifkan program sesuai dengan fungsinya) class InputUmurException dengan cara melibatkan method getAge dari class baru untuk class InputUmur Exception!