

LAPORAN PRAKTIKUM PRAKTIK PEMROGRAMAN

OLEH: HELGA ARYA PRAYOGA (24051130022)



MODUL 7

TOPIK:

EXCEPTION HANDLING DAN METHOD



TABLE OF CONTENTS

Week #7	1
A. Penjelasan Tugas Praktikum.....	3
B. Langkah-langkah dan Screenshot	3
C. Kendala yang Dialami	14
D. Kesimpulan.....	15

A. Penjelasan Tugas Praktikum

1. Penjelasan exception handling dan fungsinya.
2. Bentuk umum dari exception handling dan syarat untuk membuatnya.
3. Program exception handling sederhana.
4. Penjelasan method rekursif.
5. Penjelasan macam – macam tipe method.
6. Program method rekursif.
7. Penjelasan akses modifier.
8. Method tanpa nilai kembalian.
9. Input keyboard dengan pemanggilan method dengan nilai kembalian.
10. Paduan Antara Fungsi Dan Prosedur dalam 1 kelas.
11. Program menghitung fungsi sigma.
12. Program menghitung luas bangun.

B. Langkah-langkah dan Screenshot

1. Apa yang anda ketahui tentang exception handling? Dan apa fungsinya?

Jawab :

Exception handling adalah mekanisme dalam pemrograman yang digunakan untuk menangani kondisi kesalahan atau pengecualian yang terjadi selama eksekusi program. Fungsinya adalah untuk menjaga agar program tidak berhenti secara tiba-tiba dan memberikan cara untuk menangani kesalahan seperti memberikan pesan kesalahan yang informatif kepada pengguna.

2. Jelaskan bentuk umum dari exception handling! Dan sebutkan syarat yang harus dipenuhi untuk membuat sebuah exception handling.

Jawab :

Bentuk umum dari exception handling di Java adalah menggunakan blok try, catch, dan finally. Try adalah blok kode yang mungkin menghasilkan pengecualian. Catch adalah blok kode yang menangkap dan menangani pengecualian yang terjadi di blok try. Finally adalah blok kode yang selalu dijalankan setelah blok try dan catch, terlepas dari apakah pengecualian terjadi atau tidak. Syarat yang harus dipenuhi untuk membuat sebuah exception handling adalah kode yang berpotensi menghasilkan pengecualian harus ditempatkan dalam blok try. Blok catch juga harus sesuai dengan jenis pengecualian yang ingin ditangani.

3. Tuliskan sebuah contoh program exception handling sederhana!
Dengan menggunakan 3 blok yaitu try-catch-finally.

Kode Program:

```
public class ExceptionSederhana {
    public static void main(String[] args) {
        try {
            int[] data = {1, 2, 3};
            System.out.println(data[2]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Terjadi kesalahan: " +
e.getMessage());
        } finally {
            System.out.println("Blok finally dieksekusi.");
        }
    }
}
```

Screenshot:



The screenshot shows an IDE with two tabs: 'faktorial5.java' and 'ExceptionSederhana.java'. The 'ExceptionSederhana.java' tab is active, displaying the following code:

```
1 public class ExceptionSederhana {
2     public static void main(String[] args) {
3         try {
4             int[] data = {1, 2, 3};
5             System.out.println(data[2]);
6         } catch (ArrayIndexOutOfBoundsException e) {
7             System.out.println("Terjadi kesalahan: " + e.getMessage());
8         } finally {
9             System.out.println(x:"Blok finally dieksekusi.");
10        }
11    }
12 }
```

Below the code editor, the 'TERMINAL' tab is active, showing the execution output:

```
PS E:\Praktik Pemrograman 1\Pertemuan7> java ExceptionSederhana.java
3
Blok finally dieksekusi.
PS E:\Praktik Pemrograman 1\Pertemuan7> 
```

4. Apa sebenarnya method itu? Method ada yang dinamakan dengan method rekursif, apa method rekursif itu? Jelaskan!

Jawab :

Method adalah sekumpulan instruksi yang dikelompokkan untuk melakukan tugas tertentu dan dapat dipanggil dari bagian lain dari program.

Method rekursif adalah method yang memanggil dirinya sendiri untuk menyelesaikan suatu tugas. Metode ini biasanya digunakan untuk

masalah yang dapat dibagi menjadi sub-masalah yang lebih kecil, seperti perhitungan faktorial atau urutan Fibonacci.

5. Ada berapa macam tipe method di java? Sebutkan dan berikan contoh! Maksimal 5 baris kode untuk contoh setiap tipe method! Serta panggil method tersebut dalam main programnya!

Kode Program:

```
public class TipeMethod {
    // Method Statis
    public static void greet() {
        System.out.println("Hello, World!");
    }

    // Method Non - Statis
    public void display() {
        System.out.println("Display method");
    }

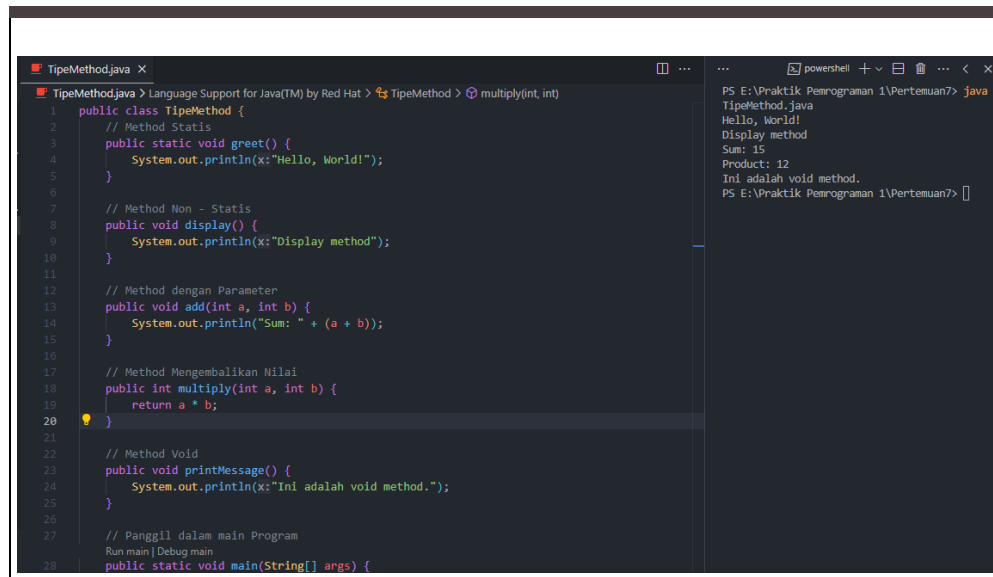
    // Method dengan Parameter
    public void add(int a, int b) {
        System.out.println("Sum: " + (a + b));
    }

    // Method Mengembalikan Nilai
    public int multiply(int a, int b) {
        return a * b;
    }

    // Method Void
    public void printMessage() {
        System.out.println("Ini adalah void method.");
    }

    // Panggil dalam main Program
    public static void main(String[] args) {
        greet();
        TipeMethod example = new TipeMethod();
        example.display();
        example.add(5, 10);
        System.out.println("Product: " + example.multiply(3,
4));
        example.printMessage();
    }
}
```

Screenshot:



```
1 public class TipeMethod {
2     // Method Statis
3     public static void greet() {
4         System.out.println(x:"Hello, World!");
5     }
6
7     // Method Non - Statis
8     public void display() {
9         System.out.println(x:"Display method");
10    }
11
12    // Method dengan Parameter
13    public void add(int a, int b) {
14        System.out.println("Sum: " + (a + b));
15    }
16
17    // Method Mengembalikan Nilai
18    public int multiply(int a, int b) {
19        return a * b;
20    }
21
22    // Method Void
23    public void printMessage() {
24        System.out.println(x:"Ini adalah void method.");
25    }
26
27    // Panggil dalam main Program
28    public static void main(String[] args) {
29        // ...
30    }
31 }
```

PS E:\Praktik Pemrograman 1\Pertemuan7> java TipeMethod.java
Hello, World!
Display method
Sum: 15
Product: 12
Ini adalah void method.
PS E:\Praktik Pemrograman 1\Pertemuan7>

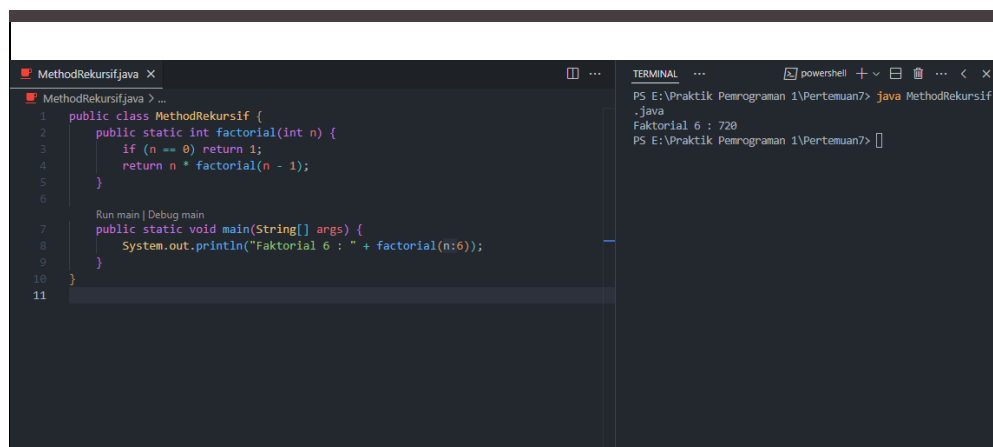
6. Berikan contoh sebuah method rekursif!

Kode Program:

```
public class MethodRekursif {
    public static int factorial(int n) {
        if (n == 0) return 1;
        return n * factorial(n - 1);
    }

    public static void main(String[] args) {
        System.out.println("Faktorial 6 : " + factorial(6));
    }
}
```

Screenshot:



```
1 public class MethodRekursif {
2     public static int factorial(int n) {
3         if (n == 0) return 1;
4         return n * factorial(n - 1);
5     }
6
7     public static void main(String[] args) {
8         System.out.println("Faktorial 6 : " + factorial(6));
9     }
10 }
11
```

PS E:\Praktik Pemrograman 1\Pertemuan7> java MethodRekursif .java
Faktorial 6 : 720
PS E:\Praktik Pemrograman 1\Pertemuan7>

7. Java adalah bahasa pemrograman bersifat objek oriented (OOP) karena semua variabel, class, fungsi/method diakses dalam bentuk objek, dan dalam setiap objek tersebut terdapat akses modifier yang nilai defaultnya adalah public. Apa akses modifier itu? Sebutkan macam akses modifier, jelaskan serta berikan contohnya!

Jawab :

Akses modifier adalah kata kunci yang digunakan untuk menentukan tingkat aksesibilitas dari variabel, metode, atau kelas dalam pemrograman Java. Ada empat jenis akses modifier dalam java.

Kode Program:

```
public class AksesModifier {
    public static void main(String[] args) {
        AksesModifier modifier = new AksesModifier();
        modifier.publicMethod();
        modifier.privateMethod();
        modifier.defaultMethod();
        modifier.protectedMethod();
    }

    // Dapat diakses dari mana saja.
    public void publicMethod() {
        System.out.println("Public method");
    }

    // Hanya dapat diakses dalam kelas yang sama.
    private void privateMethod() {
        System.out.println("Private method");
    }

    // Dapat diakses dalam kelas yang sama dan oleh subclass.
    protected void protectedMethod() {
        System.out.println("Protected method");
    }

    // Hanya dapat diakses dalam paket yang sama.
    void defaultMethod() {
        System.out.println("Default method");
    }
}
```

Screenshot:

```
1 public class AksesModifier {
2     public static void main(String[] args) {
3         AksesModifier modifier = new AksesModifier();
4         modifier.publicMethod();
5         modifier.privateMethod();
6         modifier.defaultMethod();
7         modifier.protectedMethod();
8     }
9
10    // Dapat diakses dari mana saja.
11    public void publicMethod() {
12        System.out.println(x:"Public method");
13    }
14
15    // Hanya dapat diakses dalam kelas yang sama.
16    private void privateMethod() {
17        System.out.println(x:"Private method");
18    }
19
20    // Dapat diakses dalam kelas yang sama dan oleh subclass.
21    protected void protectedMethod() {
22        System.out.println(x:"Protected method");
23    }
24
25    // Hanya dapat diakses dalam paket yang sama.
26    void defaultMethod() {
27        System.out.println(x:"Default method");
28    }
29 }
```

PS E:\Praktik Pemrograman 1\Pertemuan7> java AksesModifier.java

Public method
Private method
Default method
Protected method

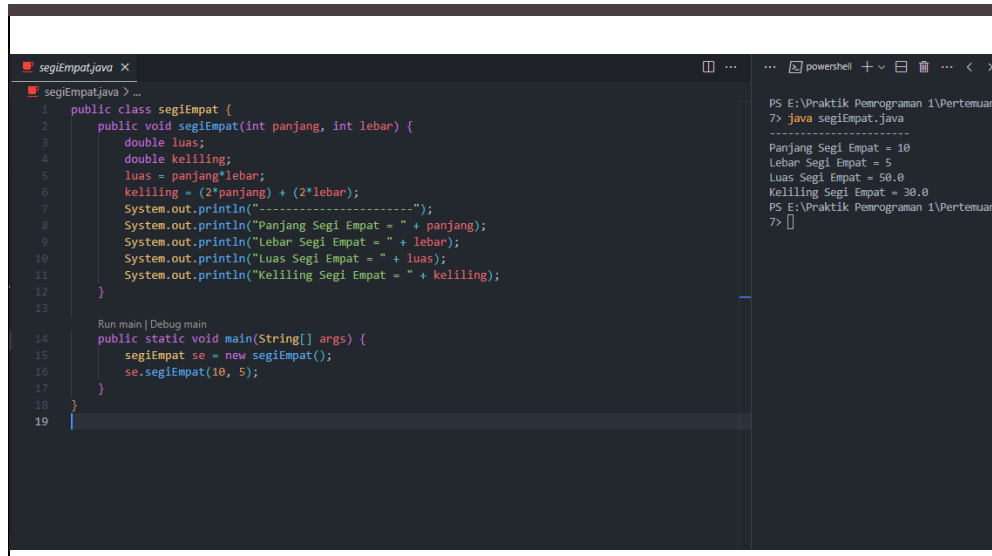
8. Method Tanpa Nilai Kembalian (Prosedur)

Kode Program:

```
public class segiEmpat {
    public void segiEmpat(int panjang, int lebar) {
        double luas;
        double keliling;
        luas = panjang*lebar;
        keliling = (2*panjang) + (2*lebar);
        System.out.println("-----");
        System.out.println("Panjang Segi Empat = " +
panjang);
        System.out.println("Lebar Segi Empat = " + lebar);
        System.out.println("Luas Segi Empat = " + luas);
        System.out.println("Keliling Segi Empat = " +
keliling);
    }

    public static void main(String[] args) {
        segiEmpat se = new segiEmpat();
        se.segiEmpat(10, 5);
    }
}
```


Screenshot:



```
segiEmpat.java X
segiEmpat.java >...
1 public class segiEmpat {
2     public void segiEmpat(int panjang, int lebar) {
3         double luas;
4         double keliling;
5         luas = panjang*lebar;
6         keliling = (2*panjang) + (2*lebar);
7         System.out.println("-----");
8         System.out.println("Panjang Segi Empat = " + panjang);
9         System.out.println("Lebar Segi Empat = " + lebar);
10        System.out.println("Luas Segi Empat = " + luas);
11        System.out.println("Keliling Segi Empat = " + keliling);
12    }
13
14    Run main | Debug main
15    public static void main(String[] args) {
16        segiEmpat se = new segiEmpat();
17        se.segiEmpat(10, 5);
18    }
19
PS E:\Praktik Pemrograman 1\Pertemuan
7> java segiEmpat.java
-----
Panjang Segi Empat = 10
Lebar Segi Empat = 5
Luas Segi Empat = 50.0
Keliling Segi Empat = 30.0
PS E:\Praktik Pemrograman 1\Pertemuan
7> 
```

9. Input Keyboard Dengan Pemanggilan Method dengan nilai kembalian (Fungsi)

Kode Program:

```
import java.io.*;
public class inputConsole {
    public static String readString() {
        BufferedReader bfr = new BufferedReader(new
InputStreamReader(System.in));
        String string = " ";
        try {
            string = bfr.readLine();
        } catch (IOException ex) {
            System.out.println(ex);
        } return string;
    }

    public static int readInt() {
        return Integer.parseInt (readString());
    }

    public static double readDouble() {
        return Double.parseDouble(readString());
    }
}
```

Screenshot:

```
inputConsole.java X
inputConsole.java > ...
1  import java.io.*;
2  public class inputConsole {
3      public static String readString() {
4          BufferedReader bfr = new BufferedReader(new InputStreamReader(System.in));
5          String string = "";
6          try {
7              string = bfr.readLine();
8          } catch (IOException ex) {
9              System.out.println(ex);
10             } return string;
11         }
12
13         public static int readInt() {
14             return Integer.parseInt(readString());
15         }
16
17         public static double readDouble() {
18             return Double.parseDouble(readString());
19         }
20     }
21 }
```

10. Paduan Antara Fungsi Dan Prosedur dalam 1 kelas

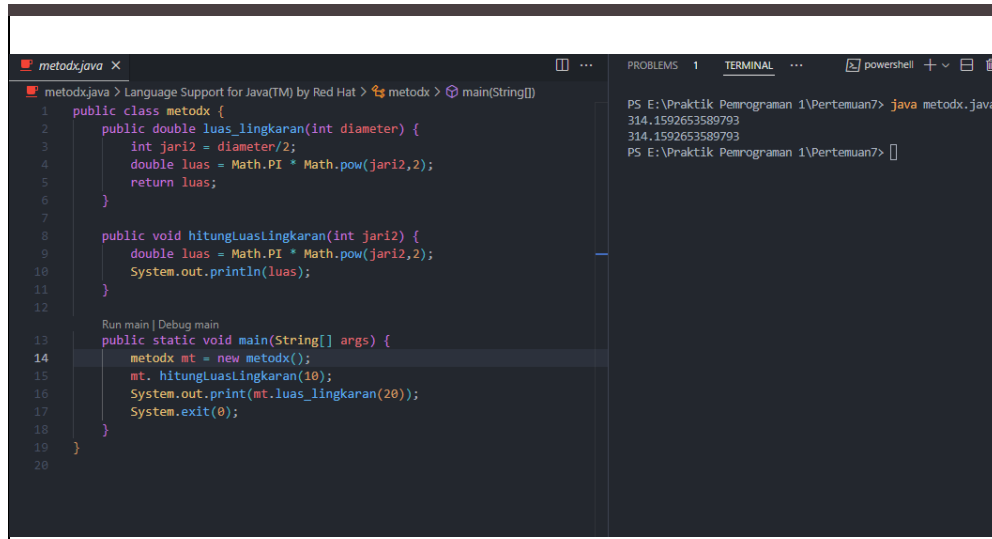
Kode Program:

```
public class metodx {
    public double luas_lingkaran(int diameter) {
        int jari2 = diameter/2;
        double luas = Math.PI * Math.pow(jari2,2);
        return luas;
    }

    public void hitungLuasLingkaran(int jari2) {
        double luas = Math.PI * Math.pow(jari2,2);
        System.out.println(luas);
    }

    public static void main(String[] args) {
        metodx mt = new metodx();
        mt.hitungLuasLingkaran(10);
        System.out.print(mt.luas_lingkaran(20));
        System.exit(0);
    }
}
```

Screenshot:



```
metodx.java x
public class metodx {
    public double luas_lingkaran(int diameter) {
        int jari2 = diameter/2;
        double luas = Math.PI * Math.pow(jari2,2);
        return luas;
    }

    public void hitungLuasLingkaran(int jari2) {
        double luas = Math.PI * Math.pow(jari2,2);
        System.out.println(luas);
    }

    Run main | Debug main
    public static void main(String[] args) {
        metodx mt = new metodx();
        mt.hitungLuasLingkaran(10);
        System.out.print(mt.luas_lingkaran(20));
        System.exit(0);
    }
}

PROBLEMS 1 TERMINAL ... powershell + v
PS E:\Praktik Pemrograman 1\Pertemuan7> java metodx.java
314.1592653589793
314.1592653589793
PS E:\Praktik Pemrograman 1\Pertemuan7>
```

11. Program menghitung fungsi sigma

Kode Program:

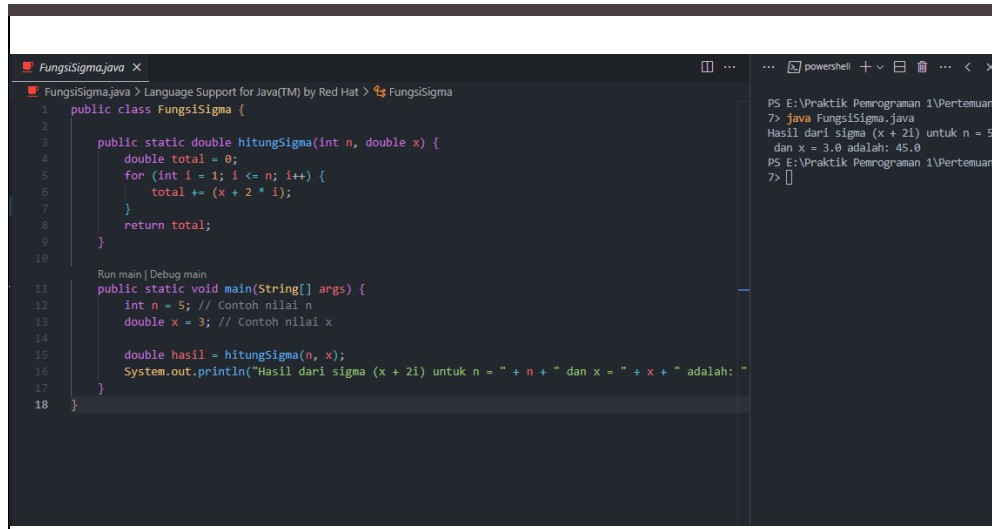
```
public class FungsiSigma {

    public static double hitungSigma(int n, double x) {
        double total = 0;
        for (int i = 1; i <= n; i++) {
            total += (x + 2 * i);
        }
        return total;
    }

    public static void main(String[] args) {
        int n = 5; // Contoh nilai n
        double x = 3; // Contoh nilai x

        double hasil = hitungSigma(n, x);
        System.out.println("Hasil dari sigma (x + 2i) untuk n
= " + n + " dan x = " + x + " adalah: " + hasil);
    }
}
```

Screenshot:



```
1 public class FungsiSigma {
2
3     public static double hitungSigma(int n, double x) {
4         double total = 0;
5         for (int i = 1; i <= n; i++) {
6             total += (x + 2 * i);
7         }
8         return total;
9     }
10
11     Run main | Debug main
12     public static void main(String[] args) {
13         int n = 5; // Contoh nilai n
14         double x = 3; // Contoh nilai x
15
16         double hasil = hitungSigma(n, x);
17         System.out.println("Hasil dari sigma (x + 2i) untuk n = " + n + " dan x = " + x + " adalah: "
18     }
```

PS E:\Praktik Pemrograman 1\Pertemuan 1> java FungsiSigma.java
Hasil dari sigma (x + 2i) untuk n = 5 dan x = 3.0 adalah: 45.0
PS E:\Praktik Pemrograman 1\Pertemuan 1>

12. Program menghitung luas bangun

Kode Program:

```
import javax.swing.JOptionPane;

public class LuasBangun {
    public static void main(String[] args) {
        int pilihan =
Integer.parseInt(JOptionPane.showInputDialog("Pilih bangun
yang ingin dihitung luasnya:\n1. Persegi\n2. Lingkaran\n3.
Segitiga"));

        switch (pilihan) {
            case 1:
                int sisi =
Integer.parseInt(JOptionPane.showInputDialog("Masukkan
panjang sisi persegi:"));
                double luasPersegi = hitungLuasPersegi(sisi);
                JOptionPane.showMessageDialog(null, "Luas
persegi adalah: " + luasPersegi);
                break;
            case 2:
                int diameter =
Integer.parseInt(JOptionPane.showInputDialog("Masukkan
diameter lingkaran:"));
                double luasLingkaran =
hitungLuasLingkaran(diameter);
                JOptionPane.showMessageDialog(null, "Luas
lingkaran adalah: " + luasLingkaran);
                break;
            case 3:
```

```

        int alas =
Integer.parseInt(JOptionPane.showInputDialog("Masukkan
panjang alas segitiga:"));
        int tinggi =
Integer.parseInt(JOptionPane.showInputDialog("Masukkan tinggi
segitiga:"));
        double luasSegitiga =
hitungLuasSegitiga(alas, tinggi);
        JOptionPane.showMessageDialog(null, "Luas
segitiga adalah: " + luasSegitiga);
        break;
        default:
            JOptionPane.showMessageDialog(null, "Pilihan
tidak valid");
    }
}

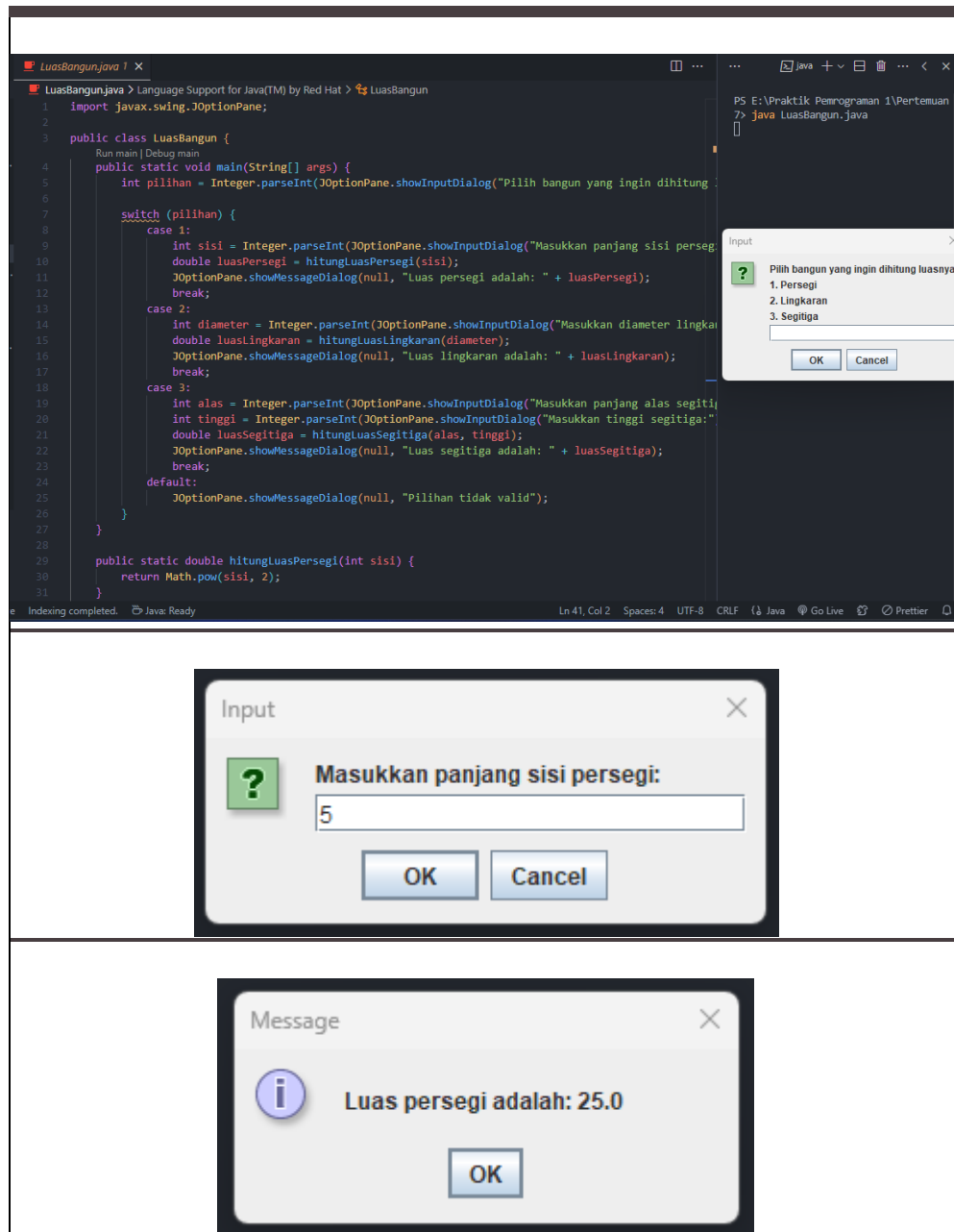
public static double hitungLuasPersegi(int sisi) {
    return Math.pow(sisi, 2);
}

public static double hitungLuasLingkaran(int diameter) {
    int jariJari = diameter / 2;
    return Math.PI * Math.pow(jariJari, 2);
}

public static double hitungLuasSegitiga(int alas, int
tinggi) {
    return (alas * tinggi) / 2;
}
}

```

Screenshot:



C. Kendala yang Dialami

Tidak ada kendala dalam membuat program

D. Kesimpulan

Penjelasan exception handling dan fungsinya. Bentuk umum dari exception handling dan syarat untuk membuatnya. Program exception handling sederhana. Penjelasan method rekursif. Penjelasan macam – macam tipe method. Program method rekursif. Penjelasan akses modifier. Method tanpa nilai kembalian. Input keyboard dengan pemanggilan method dengan nilai kembalian. Paduan Antara Fungsi Dan Prosedur dalam 1 kelas. Program menghitung fungsi sigma. Program menghitung luas bangun.