MAKALAH, PERCOBAAN, LATIHAN, DAN TUGAS
MODUL PRAKTIKUM 9

Disusun sebagai salah satu tugas
mata kuliah PBO I

Patricia Joanne
140810160065

Dikumpulkan tanggal
7 November 2017

PROGRAM STUDI S-1 TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN
2017

Tugas Pendahuluan

Buatlah review 1 halaman mengenai penanganan exception dengan cara melempar exception dan berikan 1 contoh program.

Exception adalah suatu kondisi abnormal yang terjadi pada saat menjalankan program karena dalam Java segala sesuatu merupakan objek, maka exception juga direpresentasikan dalam sebuah objek yang menjelaskan tentang exception tersebut.

Terdapat dua penanganan exception yaitu menangani sendiri exception tersebut atau meneruskannya ke luar dengan cara membuat objek tentang exception tersebut dan melemparkannya (throw) keluar agar ditangani oleh kode yang memanggil method (method yang di dalamnya terdapat exception) tersebut.

Ada lima keyword yang digunakan oleh Java untuk menangani exception yaitu try, catch, finally, throw dan throws.

Contoh program sederhana:

```
import java.util.Scanner;

public class Main {
    public static Scanner g = new Scanner(System.in);
    public static void main(String[] args){
        int angka;
        try {
            System.out.print("Masukkan angka = ");
            angka = g.nextInt();
        }
        catch(Exception e){
            System.out.println("Input yang anda masukkan bukan angka!");
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Main.java

E:\DOCS\task.bbr\Programming\Java>java Main
Masukkan angka = w
Input yang anda masukkan bukan angka!
```

Praktikum 9

Exception Class

Percobaan 1: Jalankan program di bawah ini Bagaimana output program? Jelaskan!

```
public class Percobaan1 {
    public static void main(String[] args){
        int a[] = new int[5];
        a[5] = 100;
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Percobaan1.java

E:\DOCS\task.bbr\Programming\Java>java Percobaan1
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
        at Percobaan1.main(Percobaan1.java:4)
```

Percobaan 2: Memahami cara menangkap Exception dengan tipe ArrayIndexOutOfBounds-Exception

```
public class Percobaan2 {
    public static void main(String[] args){
        int a[] = new int[5];
        try {
            a[5] = 100 ;
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Indeks Array melebihi batas");
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Percobaan2.java

E:\DOCS\task.bbr\Programming\Java>java Percobaan2
Indeks Array melebihi batas
```

Percobaan 3: Jalankan percobaan 3, bagaimana output program? Perbaiki dengan Percobaan32 untuk menangani exception.

```java
public class Percobaan3 {
    public static void main(String[] args){
        int bil = 10;
        System.out.println(bil/0);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Percobaan3.java

E:\DOCS\task.bbr\Programming\Java>java Percobaan3
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Percobaan3.main(Percobaan3.java:4)
```

```java
public class Percobaan32 {
    public static void main(String[] args){
        int bil = 10;
        try {
            System.out.println(bil/0);
        }
        catch(ArithmeticException e){
            System.out.println("Tidak boleh membagi bilangan dengan 0");
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Percobaan32.java

E:\DOCS\task.bbr\Programming\Java>java Percobaan32
Tidak boleh membagi bilangan dengan 0
```

Percobaan 4: Memahami try bertingkat

```java
public class Percobaan4 {
    public static void main(String[] args){
        int bil = 10;
        try {
            System.out.println(bil/0);
        }
        catch(ArithmeticException e){
            System.out.println("Terjadi exception karena tidak boleh membagi bilangan dengan 0");
        }
        catch(Exception e){
            System.out.println("Terdapat Error");
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Percobaan4.java

E:\DOCS\task.bbr\Programming\Java>java Percobaan4
Terjadi exception karena tidak boleh membagi bilangan dengan 0
```

Percobaan 5: Bandingkan output dua program di bawah ini! Jelaskan!

```java
public class Percobaan5 {
    public static void main(String[] args){
        int bil = 10;
        String b[] = {"a","b","c"};
        try {
            System.out.println(bil/0);
            System.out.println(b[3]);
        }
        catch(ArithmeticException e){
            System.out.println("Error Aritmetik");
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Error Kapasitas Array Melebihi Batas");
        }
        catch(Exception e){
            System.out.println("Terdapat Error");
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Percobaan5.java

E:\DOCS\task.bbr\Programming\Java>java Percobaan5
Error Aritmetik
```

```java
public class Percobaan52 {
    public static void main(String[] args){
        int bil = 10;
        String b[] = {"a","b","c"};
        try {
            System.out.println(b[3]);
            System.out.println(bil/0);
        }
        catch(ArithmeticException e){
            System.out.println("Error Aritmetik");
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Error Kapasitas Array Melebihi Batas");
        }
        catch(Exception e){
            System.out.println("Terdapat Error");
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Percobaan52.java

E:\DOCS\task.bbr\Programming\Java>java Percobaan52
Error Kapasitas Array Melebihi Batas
```

Percobaan 6: Penggunaan finally

```java
public class ExcepTest {
    public static void main(String args[]){
        int a[] = new int[2];
        try {
            System.out.println("Access element three :" + a[3]);
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Exception thrown :" + e);
        }
        finally {
            a[0] = 6;
            System.out.println("First element value: " +a[0]);
            System.out.println("The finally statement is executed");
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac ExcepTest.java

E:\DOCS\task.bbr\Programming\Java>java ExcepTest
Exception thrown :java.lang.ArrayIndexOutOfBoundsException: 3
First element value: 6
The finally statement is executed
```

Percobaan 7: Method yang melempar unchecked exception

```java
public class Percobaan7 {
    public static void method1(){
        throw new ArrayIndexOutOfBoundsException("Melebihi Kapasitas");
    }
    public static void main(String[] args) {
        try {
            method1();
        }
        catch(ArrayIndexOutOfBoundsException ex){
            System.out.println(ex.getMessage());
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Percobaan7.java

E:\DOCS\task.bbr\Programming\Java>java Percobaan7
Melebihi Kapasitas
```

Percobaan 8: Method yang melempar checked exception

```java
import java.io.FileNotFoundException;

public class Percobaan8 {
    public static void method1() throws FileNotFoundException {
        throw new FileNotFoundException("File Tidak Ada");
    }
    public static void main(String[] args){
        try {
            method1();
        }
        catch (FileNotFoundException ex){
            System.out.println(ex.getMessage());
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Percobaan8.java

E:\DOCS\task.bbr\Programming\Java>java Percobaan8
File Tidak Ada
```

Percobaan 9: Memahami mengenai mekanisme exception

```java
public class Exercise1 {
    static void f1(){
        System.out.print("1");
        try {
            System.out.print("2");
            f2();
            System.out.print("3");
        }
        catch(Exception e){
            System.out.print("4");
        }
        finally {
            System.out.print("5");
        }
        System.out.println("6");
    }
```

```
    static void f2 () throws Exception {
        if (true) throw new Exception();
    }
    public static void main(String s[]){
        f1();
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Exercise1.java

E:\DOCS\task.bbr\Programming\Java>java Exercise1
12456
```

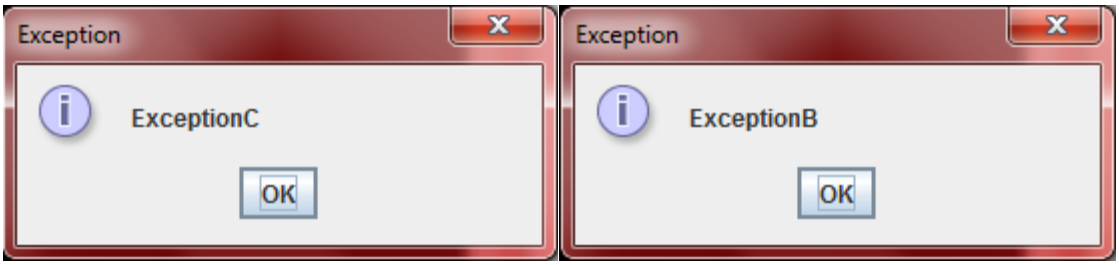
Percobaan 10: Menggunakan konsep Inheritance untuk membuat superclass dan subclass exception. Program menangani exception dengan menangkap subclass exception dengan superclass.

```
import javax.swing.*;
class ExceptionA extends Exception {}
class ExceptionB extends ExceptionA {}
class ExceptionC extends ExceptionB {}

public class Demo {
    public static void main(String args[]){
        try {
            throw new ExceptionC();
        }
        catch(ExceptionA a){
            JOptionPane.showMessageDialog(
                null, a.toString(), "Exception", JOptionPane.INFORMATION_MESSAGE);
        }
        try {
            throw new ExceptionB();
        }
        catch(ExceptionA b){
            JOptionPane.showMessageDialog(
                null, b.toString(), "Exception",JOptionPane.INFORMATION_MESSAGE);
        }
        System.exit(0);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Demo.java

E:\DOCS\task.bbr\Programming\Java>java Demo
```



Percobaan 11: Membuat exception sendiri

```
class Salah extends Exception {
    public Salah(){}
    public Salah(String pesan){
        super(pesan);
    }
}
```

```
public class TesSalah {
    public static void main(String [] arg) throws Salah {
        Salah s = new Salah("Salah disengaja ha..ha..");
        int i = 0;
        if (i==0) throw s;
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac TesSalah.java

E:\DOCS\task.bbr\Programming\Java>java TesSalah
Exception in thread "main" Salah: Salah disengaja ha..ha..
        at TesSalah.main(TesSalah.java:3)
```

Percobaan 12: Membuat exception sendiri

```
class counterException extends Exception {
    String complaint;
    public counterException(String c){
        this.complaint = c;
    }
    public String toString(){
        return "counter Exception "+complaint;
    }
}
```

```
class counter {
    int n = 0;
    public int zero(){
        return n=0;
    }
    public int up(){
        return ++n;
    }
    public int down() throws counterException {
        if(n <= 0) throw new counterException(n+" countdown failed.");
        return --n;
    }
}
```

```
public class Example1 {
    public static void main(String args[]){
        counter aCounter = new counter();
        aCounter.zero();
        aCounter.up();
        try {
            aCounter.down();
        }
        catch(counterException ce){
            System.out.println("" +ce);
        }
        try {
            aCounter.down();
        }
        catch(counterException ce){
            System.out.println("" +ce);
        }
        finally {
        System.out.println("Finally");
        }
    }
}
```

```
public class Example2 {
    public static void main(String args[]) throws Exception {
        counter aCounter = new counter();
        aCounter.zero();
        aCounter.up();
        aCounter.down();
        aCounter.down();
        System.out.println("Completed");
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Example1.java

E:\DOCS\task.bbr\Programming\Java>java Example1
counter Exception 0 countdown failed.
Finally

E:\DOCS\task.bbr\Programming\Java>javac Example2.java

E:\DOCS\task.bbr\Programming\Java>java Example2
Exception in thread "main" counter Exception 0 countdown failed.
        at counter.down(counter.java:10)
        at Example2.main(Example2.java:7)
```

Latihan:

1. Semua exception yang berasal dari `java.lang.RuntimeException` adalah unchecked exceptions, sedangkan exception lainnya yang tidak berasal dari `java.lang.RuntimeException` adalah checked exceptions. Jelaskan mengenai unchecked exceptions dan checked exceptions, berikan contoh!

   a. Checked Exception adalah eksepsi yang diantisipasi oleh programmer untuk dihandle dalam program dan terjadi dikarenakan oleh kondisi luar yang siap muncul saat program berjalan. Misalnya membuka file yang tidak ada atau gangguan jaringan. Yang termasuk checked exception adalah class `java.lang.Throwable` dan semua subclassnya, kecuali class dan subclass dari `java.lang.Error` dan `java.lang.RuntimeError`.

   Contoh:

```java
import java.io.*;

class Checked {
    public static void main(String args[]){
        FileInputStream fis = null;
        try {
            fis = new FileInputStream("myfile.txt");
        }
        catch(FileNotFoundException fnfe){
            System.out.println("The specified file is not present at the given path");
        }
        int k;
        try {
            while((k = fis.read() ) != -1){
            System.out.print((char)k);
        }
        fis.close();
```

```
        catch(IOException ioe){
            System.out.println("I/O error occurred: "+ioe);
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Checked.java

E:\DOCS\task.bbr\Programming\Java>java Checked
The specified file is not present at the given path
Exception in thread "main" java.lang.NullPointerException
        at Checked.main(Checked.java:14)
```

b. Unchecked Exception bisa muncul dari kondisi yang merepresentasikan adanya bug atau situasi yang secara umum dianggap terlalu sulit bagi program untuk menghandlenya. Disebut sebagai unchecked karena kita tidak perlu mengeceknya atau melakukan sesuatu jika kondisi ini terjadi. Eksepsi yang muncul dari kategori situasi yang merepresentasikan bug ini disebut sebagai runtime exception. Misalnya mengakses array melebihi size yang dimilikinya. Yang termasuk unchecked exception adalah `java.lang.Error` dan subclassnya serta `java.lang.RuntimeException` dan subclassnya.

   Contoh:

```
class Unchecked {
    public static void main(String args[]){
        int num1 = 10;
        int num2 = 0;
        int res=num1/num2;
        System.out.println(res);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Unchecked.java

E:\DOCS\task.bbr\Programming\Java>java Unchecked
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Unchecked.main(Unchecked.java:5)
```

2. Buatlah contoh program untuk menangani exception dengan cara menangkap exception seperti di bawah ini:

- ArithmeticException

```java
class Arithmetic {
    public static void main(String args[]){
        int num1 = 10;
        int num2 = 0;
        int res = num1/num2;
        System.out.println(res);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Arithmetic.java

E:\DOCS\task.bbr\Programming\Java>java Arithmetic
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Arithmetic.main(Arithmetic.java:5)
```

- ArrayStoreException

```java
public class ArrayStore {
    public static void main(String[] args){
        Object[] s = new Integer[4];
        s[0] = 4.4;
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac ArrayStore.java

E:\DOCS\task.bbr\Programming\Java>java ArrayStore
Exception in thread "main" java.lang.ArrayStoreException: java.lang.Double
        at ArrayStore.main(ArrayStore.java:4)
```

- ClassCastException

```java
public class ClassCast {
    public static void main(String[] args){
        Object obj = new Integer(100);
        System.out.println((String) obj);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac ClassCast.java

E:\DOCS\task.bbr\Programming\Java>java ClassCast
Exception in thread "main" java.lang.ClassCastException: java.lang.Integer cannot be cast to java.la
ng.String
        at ClassCast.main(ClassCast.java:4)
```

- ArrayIndexOutOfBoundsException

```java
class AIOOB {
    public static void main(String[] args){
        char[] matrix = new char[] {'H', 'e', 'l', 'l', 'o'};
        System.out.println(matrix);

        for(int i = 0; i <= matrix.length; ++i){
            System.out.println(matrix[i]);
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac AIOOB.java

E:\DOCS\task.bbr\Programming\Java>java AIOOB
Hello
H
e
l
l
o
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
        at AIOOB.main(AIOOB.java:7)
```

- StringIndexOutOfBoundsException

```java
public class SIOOB {
    public static void main(String[] args){
        String str = "LAPRAK OOP LALALA";
        System.out.println("Length: " +str.length());
        char ch = str.charAt(50);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac SIOOB.java

E:\DOCS\task.bbr\Programming\Java>java SIOOB
Length: 17
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 50
        at java.lang.String.charAt(String.java:658)
        at SIOOB.main(SIOOB.java:5)
```

- NegativeArraySizeException

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

public class NegArraySize {
    public static void main(String args[]) throws IOException {
        int c[] = new int[-2];
        Scanner in = new Scanner(new InputStreamReader(System.in));
        int b=in.nextInt();
        int a[]=new int[b];
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac NegArraySize.java

E:\DOCS\task.bbr\Programming\Java>java NegArraySize
Exception in thread "main" java.lang.NegativeArraySizeException
        at NegArraySize.main(NegArraySize.java:6)
```

- NoSuchElementException

```java
import java.util.HashSet;
import java.util.Hashtable;
import java.util.Set;
import java.util.StringTokenizer;

public class NoSuch {
    public static void main(String[] args){
        Set sampleSet = new HashSet();
        Hashtable sampleTable = new Hashtable();
        StringTokenizer tokenizer = new StringTokenizer("", ":,");
        sampleSet.iterator().next();
        sampleTable.elements().nextElement();
        tokenizer.nextToken();
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac NoSuch.java

E:\DOCS\task.bbr\Programming\Java>java NoSuch
Exception in thread "main" java.util.NoSuchElementException
        at java.util.HashMap$HashIterator.nextNode(HashMap.java:1439)
        at java.util.HashMap$KeyIterator.next(HashMap.java:1461)
        at NoSuch.main(NoSuch.java:11)
```

- NullPointerException

```java
public class NullPointer {
    public static void main(String[] args){
        NullPointer t = initT();
        t.foo("Hi");
    }
    private static NullPointer initT(){
        return null;
    }
    public void foo(String s){
        System.out.println(s.toLowerCase());
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac NullPointer.java

E:\DOCS\task.bbr\Programming\Java>java NullPointer
Exception in thread "main" java.lang.NullPointerException
        at NullPointer.main(NullPointer.java:4)
```

- NumberFormatException

```java
class NumFormat {
   public static void main(String args[]){
    int num = Integer.parseInt("XYZ");
    System.out.println(num);
   }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac NumFormat.java

E:\DOCS\task.bbr\Programming\Java>java NumFormat
Exception in thread "main" java.lang.NumberFormatException: For input string: "XYZ"
        at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
        at java.lang.Integer.parseInt(Integer.java:580)
        at java.lang.Integer.parseInt(Integer.java:615)
        at NumFormat.main(NumFormat.java:3)
```

3. Terdapat dua cara untuk menangani Exception yaitu dengan menangkap Exception dan melempar Exception. Lakukan penanganan exception dengan **melempar Exception menggunakan throw**. Berilah penjelasan (apakah program termasuk unchecked exceptions atau checked exceptions)!

```java
import java.io.*;

public class ReadFile {
    public static void main(String[] args) throws FileNotFoundException, IOException {
        File file = new File("Data.txt");
        BufferedReader fileReader;
        fileReader = new BufferedReader(new FileReader(file));

        while(true){
            String line = fileReader.readLine();
            if(line == null) break;
            System.out.println(line);
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac ReadFile.java

E:\DOCS\task.bbr\Programming\Java>java ReadFile
Exception in thread "main" java.io.FileNotFoundException: Data.txt (The system cannot find the file
specified)
        at java.io.FileInputStream.open0(Native Method)
        at java.io.FileInputStream.open(FileInputStream.java:195)
        at java.io.FileInputStream.<init>(FileInputStream.java:138)
        at java.io.FileReader.<init>(FileReader.java:72)
        at ReadFile.main(ReadFile.java:7)
```

Keyword *throw* bisa digunakan untuk checked dan unchecked exception, namun untuk contoh di atas *throw* yang digunakan adalah untuk checked exception karena exceptionnya FileNotFoundException yang bukan merupakan subclass dari `java.lang.RuntimeException`.

Tugas:

1. Terdapat dua cara untuk menangani Exception yaitu dengan menangkap Exception dan melempar Exception. Lakukan penanganan exception dengan menangkap Exception menggunakan blok try-catch. Berilah penjelasan (apakah program termasuk unchecked exceptions atau checked exceptions)!

```java
import java.io.*;

public class ReadFile {
    public static void method1() throws FileNotFoundException, IOException {
        File file = new File("Data.txt");
        BufferedReader fileReader;
        fileReader = new BufferedReader(new FileReader(file));
```

```
        while(true){
            String line = fileReader.readLine();
            if (line == null)
                break;
            System.out.println(line);
        }
    }
    public static void main(String[] args) throws FileNotFoundException, IOException {
        try {
            method1();
        }
        catch (FileNotFoundException ex){
            System.out.println(ex.getMessage());
        }
    }
}
```

Keyword *try-catch* bisa digunakan untuk checked dan unchecked exception, namun untuk contoh di atas *try-catch* yang digunakan adalah untuk checked exception karena exceptionnya masih bisa dihandle.

2. Buatlah sebuah class **Stack, FullStackException dan EmptyStackException**. Class Stack ini menggambar Stack yang menerapkan konsep LIFO (Last In First Out). Konsep LIFO ini, data yang terakhir masuk akan keluar pertama kali.

```
class FullStackException extends RuntimeException {
    public FullStackException() {}
    public FullStackException(String msg){
        super(msg);
    }
}

class EmptyStackException extends RuntimeException {
    public EmptyStackException() {}
    public EmptyStackException(String msg){
        super(msg);
    }
}
```

```
public class Stack {
    private int size;
    private int top;
    Object[] elemen;

    public Stack(){
        this.size = 5;
        elemen = new Object[this.size];
        this.top = 0;
    }
    public Stack(int s){
        size = s;
        elemen = new Object[s];
        this.top = 0;
    }
    public int getSize(){
        return this.size;
    }
    public int getTop(){
        return this.top;
    }
```

```
    public void push(Object o){
        try {
            if (this.top == this.size){
                throw new FullStackException("Stack Full");
            }
            elemen[top] = o;
            top++;
        }
        catch (FullStackException ex){
            System.out.println("FullStackException" + ex.getMessage());
        }
    }
    public Object pop(){
        try {
            if(this.top == 0){
                throw new EmptyStackException("Stack Kosong");
            }
        }
        catch (EmptyStackException e){
            System.out.println("EmptyStackException: " + e.getMessage());
        }
        finally {
            Object temp = elemen[top];
            top--;
            return temp;
        }
    }
}
```

```java
public class EmptyStack {
    public static void main(String[] args){
        Stack stack = new Stack();
        stack.push("1");
        stack.push("2");
        stack.pop();
        stack.pop();
        stack.pop();
    }
}
```

```java
public class FullStack {
    public static void main(String[] args){
        Stack stack = new Stack();
        stack.push("1");
        stack.push("2");
        stack.push("3");
        stack.push("4");
        stack.push("5");
        stack.push("6");
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac FullStack.java

E:\DOCS\task.bbr\Programming\Java>java FullStack
Patricia Joanne 140810160065
FullStackException: Stack Penuh

E:\DOCS\task.bbr\Programming\Java>javac EmptyStack.java

E:\DOCS\task.bbr\Programming\Java>java EmptyStack
Patricia Joanne 140810160065
EmptyStackException: Stack Kosong
```