**LAPORAN PRAKTIKUM**

49

**ALGORITMA DAN PEMROGRAMAN LANJUT**

**Pertemuan Ke – 12**



**DISUSUN OLEH :**

**HELDA LUDYA SAFITRI**

**175410186**

**TEKNIK INFORMATIKA**

**STMIK AKAKOM**

**YOGYAKARTA**

**2018**

**PERTEMUAN KE-12**

**EXCEPTION**

1. **TUJUAN**
2. Mahasiswa mampu menangani eksepso sehingga tidak terjadi running error.
3. Mahasiswa dapat menggunakan perintah try catch dan finally.
4. Mahasiswa menggunakan perintah throw.
5. **DASAR TEORI**

Exception adalah event yang terjadi ketika program menemui kesalahan pada saat instruksi program dijalankan.

Exception terdiri dari dua macam kelompok, yaitu :

* Exception yang merupakan subclass RunTimeException.
* Exception yang bukan subclass RunTimeException.

Dalam bahasa Java, ketika terjadi kesalahan, otomatis akan dilemparkan sebuah objek yang disebut exception, yang kemudian dapat diproses lebih lanjut oleh fungsi-fungsi yang siap menangani kesalahan tersebut. Proses pelemparan exception tersebut sering dikenal dengan istilah throwing exception, sedangkan proses penerimaan exception yang bersangkutan dikenal dengan istilah catch exception.

1. **PEMBAHASAN PRAKTIK**

**Praktik 1**

import java.util.Scanner;

public class TestError{

public static void main(String[] arg){

Scanner reader = new Scanner(System.in);

System.out.println("Please enter the miles you have driven.");

double miles = reader.nextDouble();

System.out.println("Please enter the gallon of gas you have used.");

double gallon = reader.nextDouble();

double mpg = miles/gallon;

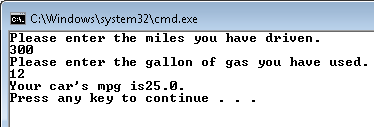
System.out.println("Your car's mpg is" +mpg+".");

}

}

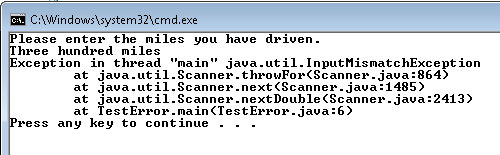
Script di atas merupakan program untuk mengecek error tanpa exception. Dengan menggunakan fungsi scanner , reader sebagai scanner pada program diatas, user diminta untuk memasukkan nilai miles dan gallon yang bertipe data double kemudian program akan menghitung nilai mpg dengan rumus nilai miles dibagi dengan nilai gallon yang kemudian akan dijalankan dengan 2 cara di bawah ini :

Output 1 :



Terlihat output diatas menampilkan nilai yang diinputkan pada program dan program terlihat tidak error.

Output 2 :



Output diatas terlihat menghasilkan keterangan error, hal tersebut dikarenakan input yang dimasukkan oleh user tidak sesuai dengan tipe data yang ada pada program. Pada program tiap data yang digunakan adalah double tetapi yang diinputkan oleh user adalah tipe data string sehingga program tidak dapat dijalankan.

**Praktik 2**

import java.util.Scanner;

public class TestError2 {

public static void main(String[] arg) {

Scanner reader = new Scanner(System.in);

try {

System.out.println("Please enter the miles you have driven.");

double miles = reader.nextDouble();

System.out.println("Please enter the gallon of gas you have used.");

double gallon = reader.nextDouble();

double mpg = miles/gallon;

System.out.println("Your car's mpg is"+mpg+".");

}catch (Exception e){

System.out.println("You must enter a number.");

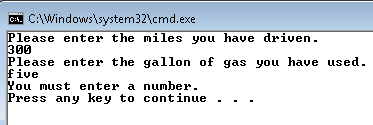
}

}

}

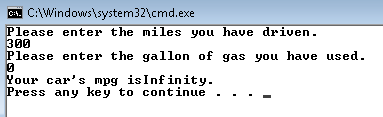
Tujuan program diatas sama dengan program sebelumnya, praktikan hanya memodifikasi program sebelumnya dengan menambahkan try dan catch. Pada try diisikan syntax penginputan nilai dari miles dan gallon hingga syntax perihitungan nilai mpg. Kemudian isi catch adalah perintah S.O.P yang berisi “You must enter a number”. Kemudian program dijalankan dengan cara dibawah ini :

Output 1 :



Terlihat output diatas ketika praktikan memasukkan nilai five ke gallon maka tidak lagi muncul error exception input, tetapi error yang terdeteksi pada try maka akan dialihkan ke catch dengan menjalankan perintah yang ada di dalam catch yakni S.O.P yang tertera pada gambar.

Output 2 :



Terlihat output di atas digunakan jenis inputan yang berbeda pada gallon yaitu 0, ketika dijalakan perhitungan mpg maka akan menghasilkan infinity, hal tersebut dikarenakan pembagi dari rumus mpg adalah 0.

**Praktik 3**

import java.util.Scanner;

public class TestError3 {

public static void main(String[] arg) {

Scanner reader = new Scanner(System.in);

try {

System.out.println("Please enter miles you have driven.");

double miles = reader.nextDouble();

System.out.println("Please enter the gallon of gas you have used.");

double gallon = reader.nextDouble();

if(gallon==0)

throw new ArithmeticException();

double mpg = miles/gallon;

System.out.println("Your car's mpg is"+mpg+".");

}catch (Exception e){

System.out.println("You must enter a number.");

System.out.println(e.getClass());

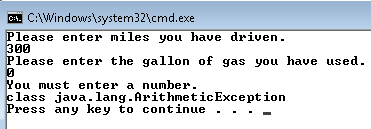
}

}

}

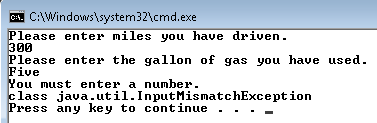
Program diatas masih sama dengan program sebelumnya . namun pada program diatas throw nya terdapat didalam if . masih menggunakan try catch dan throw berada ditengahnya. Lalu dibawah sendiri terdapat tambahan perintah e.getClass . Perintah throw dilakukan jika nilai gallon bernilai 0 dan throw terdapat pada ArithmeticException dimana terdapat kesalahan pada perhitungan. Dengan penambahan Exception e pada catch yang berisikan S.O.P “You must enter a number” dan menampilkan class dari exceptionnya. Berikut adalah hasil dari program diatas.

Output 1 :



Terlihat pada output diatas praktikan memberikan input 0 pada gallon yang membuat program menjalankan throw pada exception sehingga akan dikerjakan perintah apda catch yang merupakan penampilan S.O.P berupa “You must enter a number” dan kelas dari exception yakni AritmeticException.

Output 2 :



Terlihat pada output diatas, gallon diberi nilai five yang mana merupakan tipe data string dan tidak sesuai dengan yang program mau sehingga akan dijalankan perintah catch yang akan menampilkan “You must enter a number” dan kelas exceptionnya yaitu InputMismatchException dimana error tersebut diperuntukkan untuk ketidaksesuaian input data pada program.

**Praktik 4**

import java.util.Scanner;

public class TestError4 {

public static void main(String[] arg) {

Scanner reader = new Scanner(System.in);

try {

System.out.println("Please enter the miles you have driven.");

double miles = reader.nextDouble();

System.out.println("Please enter the gallon of gas you have used.");

double gallon = reader.nextDouble();

if(gallon==0)

throw new ArithmeticException();

double mpg = miles/gallon;

System.out.println("Your car's mpg is"+mpg+".");

}catch (ArithmeticException e){

System.out.println("You can not divide by zero");

}catch (Exception e){

System.out.println("You must enter a number.");

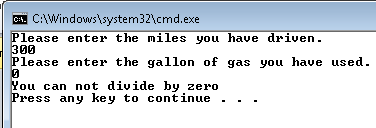
}

}

}

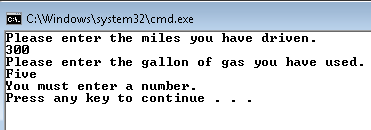
Program diatas menggunakan multi catch , program diatas masih menggunakan program sebelumnya yang dimodifikasi jadi jika pada program sebelumnya hanya terdapat catch 1 . di program ini setelah catch pertama , dibawahnya masih terdapat catch lagi . Isi dari catch pertama adalah perintah yang dilakukan jika program mendeteksi AritmeticException maka akan dijalankan S.O.P “You cannot divide by zero”, dan catch kedua akan dijalankan ketika pada try ditemukan exception selain AritmeticException yang berisi S.O.P “You must enter a number”.

Output 1 :



Terlihat pada gambar di atas nilai gallon diisikan dengan 0 maka program akan menjalankan perintah pada if dan catch yang pertama yaitu “You cannot divide by zero”.

Output 2 :



Terlihat pada gambar di atas nilai gallon diisikan dengan five maka program akan menjalankan perintah pada catch yang kedua yaitu “You must enter a number”.

**Praktik 5**

import java.util.Scanner;

public class TestError5 {

public static void main(String[] arg) {

Scanner reader = new Scanner(System.in);

try {

System.out.println("Please enter the miles you have driven.");

double miles = reader.nextDouble();

System.out.println("Please enter the gallon of gas you have used.");

double gallon = reader.nextDouble();

if(gallon == 0)

throw new DivideByZeroException1();

if(gallon<0 || miles<0)

throw new NegativeNumberException1();

double mpg = miles/gallon;

System.out.println("Your car's mpg is"+mpg+".");

}catch (DivideByZeroException1 e) {

System.out.println("You can not divide by zero");

}catch (NegativeNumberException1 e) {

System.out.println("You can not use negatif numbers");

}

catch (Exception e){

System.out.println("You must enter a number.");

System.out.println(e.getClass());

}

}

}

class DivideByZeroException1 extends Exception{

DivideByZeroException1() {

}

DivideByZeroException1(String message) {

super(message);

}

}

class NegativeNumberException1 extends Exception {

NegativeNumberException1(){

}

NegativeNumberException1(String message) {

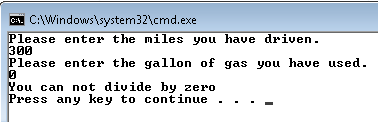
super(message);

}

}

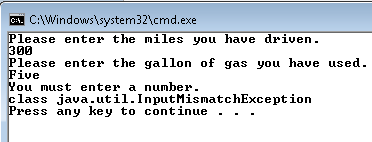
Script di atas merupakan syntax program pembuatan Exception class. Terlihat terdapat 2 if, if yang pertama berisikan jika nilai gallon = 0 maka akan dijalankan throw DivideByZeroException1. If kedua berisikan jika nilai gallon atau miles bernilai < 0 maka akan dijalankan throw NegativeNumberException1. Class exception yang dibuat oleh praktikan adalah DivideByZeroException1 dan NegativeNumberException1 dimana pada masing-masing kelas akan menjalankan methodnya masing-masing.

Output 1:



Terlihat pada output diatas nilai gallon adalah 0 sehingga akan dijalankan catch dengan exception DivideByZeroException1 yang berisi perintah S.O.P “You cannot divide by zero”.

Output 2 :



Terlihat pada output diatas nilai gallon adalah five sehingga akan dijalankan catch dengan exception e yang berisi perintah S.O.P “You must enter a number” dan kelas exceptionnya.

**Praktik 6**

import java.util.Scanner;

public class TestError6 {

public static void main(String[] arg) {

Scanner reader = new Scanner(System.in);

try {

System.out.println("Please enter the miles you have driven.");

double miles = reader.nextDouble();

System.out.println("Please enter the gallon of gas you have used.");

double gallon = reader.nextDouble();

if(gallon==0)

throw new DivideByZeroException1();

if (gallon<0 || miles<0)

throw new NegativeNumberException1();

double mpg = miles/gallon;

System.out.println("Your car's mpg is"+mpg+".");

}catch (DivideByZeroException1 e) {

System.out.println("You can not divide by zero");

}catch (NegativeNumberException1 e) {

System.out.println("You can not use negatif numbers");

}

catch(Exception e){

System.out.println("You must enter a number.");

System.out.println(e.getClass());

}finally {

System.out.println("Thank you for using the MPC System");

}

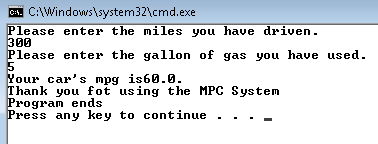
System.out.println("Program ends");

}

}

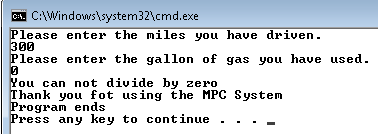
Pada program diatas menggunakan finally, masih sama dengan program sebelumnya namun pada program ini setelah catch berakhir dibawahnya ditambah perintah finally yang isinya “Thank you for using the MPC system” , lalu dibawahnya terdapat kalimat program end. Berikut adalah hasil dari program diatas :

Output 1 :



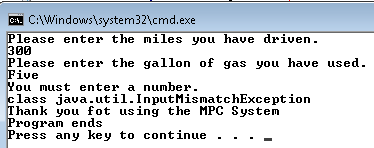
Terlihat pada output diatas program menjalankan pekerjaannya dengan baik karena nilai yang dinputkan oleh user juga memang sudah tepat sehingga program dapat berjalan dengan seharusnya. Dan terlihat terdapat “Thank you for using the mpc System” dan “Program ends”.

Output 2 :



Pada output diatas nilai gallon = 0 yang secara otomatis akan menjalankan perintah pada if pertama yaitu throw ke exception DivideByZeroException1 yang kemudian dikerjakan oleh catch exception tersebut yang berisikan perintah S.O.P “You can not divide by zero” lalu dijalankan perintah dalam finally dan S.O.P terakhir.

Output 3 :



Pada output diatas nilai gallon = five, karena nilai yang diinputkan tidak sesuai dengan tipe data pada program maka terjedi error exception yang kemudian progam akan menjalankan perintah pada catch Exception e dimana berisi S.O.P “You must enter a number” dan S.O.P kelas Exception yang terjadi. Tak lupa perintah dari Finally dan “Progam Ends” dijalankan.

**Praktik 7**

import java.util.Scanner;

public class TestError4 {

public static void main(String[] arg) {

Scanner reader = new Scanner(System.in);

try {

System.out.println("Please enter the miles you have driven.");

double miles = reader.nextDouble();

System.out.println("Please enter the gallon of gas you have used.");

double gallon = reader.nextDouble();

if(gallon==0)

throw new ArithmeticException();

double mpg = miles/gallon;

System.out.println("Your car's mpg is"+mpg+".");

}catch (Exception e){

System.out.println("You must enter a number");

}catch (ArithmeticException e){

System.out.println("You can not divide by zero.");

}

}

Script diatas merupakan syntax progam yang sama pada syntax program pada praktik 4. Tetapi urutan multi catchnya dirubah atau dibalik. Oleh karena itu, apabila program dijalankan akan terjadi error akibat terjadi perubahan pada penulisan catch.

1. **LATIHAN**

import java.util.Scanner;

public class lattihan {

public static void main(String[] args) {

Scanner reader = new Scanner(System.in);

String[] helda = {"helda", "cantik", "sekali", "Kendal"};

try {

System.out.println("Masukkan index yang akan ditampilkan: ");

int index = reader.nextInt();

System.out.println(helda[index]);

}catch (IndexOutOfBoundsException e){

System.out.println(" Woy! Terjadi kesalahan pada nilai index");

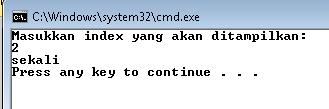
}

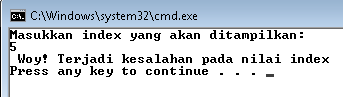
}

}

Program diatas merupakan bentuk penggunaan exception indexOutofBoundsException, dimana variable helda dideklarasikan dengan tipe data String dengan data yang akan dimasukkan oleh user sesuai index data, apabila index yang dimasukkan tidak sesuai dengan jumlah data yang ada maka akan ditampilkan kesalahan oleh S.O.P setelah catch.

Output :





1. **TUGAS**

import java.util.Scanner;

public class tugas

{

public static void tablePerkalian(int x, int y){

int array[][] = new int [x][y];

int jum;

for(int k=0;k<=x;k++)

{

System.out.print(k+"\t");

}

System.out.println(" ");

for(int i=0;i<x;i++)

{

System.out.print((i+1)+"\t");

array[i][0]=i+1;

for(int j=0;j<y;j++)

{

array[0][j]=j+1;

jum=array[i][0]\*array[0][j];

System.out.print(jum+"\t");

}

System.out.println("\t");

}

}

public static void main(String[]arg)

{

Scanner input = new Scanner(System.in);

int baris,kolom;

try{

System.out.print("Masukkan jumlah Baris : ");

baris = input.nextInt();

System.out.print("Masukkan jumlah Kolom : ");

kolom = input.nextInt();

if (baris>10 || kolom>10)

throw new ArithmeticException();

tablePerkalian(baris,kolom);

}catch (ArithmeticException e) {

System.out.println("Jumlah baris dan kolom tidak boleh lebih dari 10");

}catch (Exception e){

System.out.println("Harus memasukkan angka.");

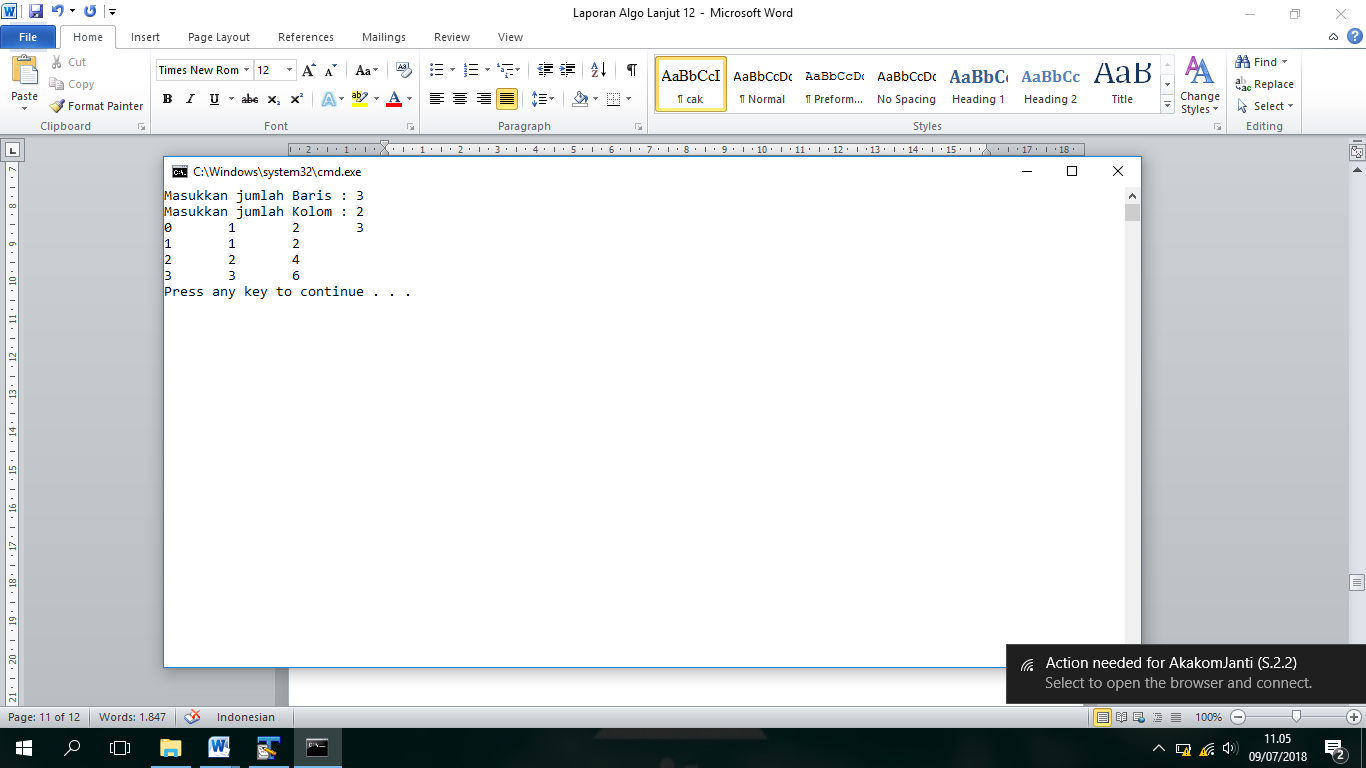
}

}

}

Program diatas akan menampilkan table perkalian dengan variable x dan y, bertipe data integer, menggunakan perulangan for pada array yang akan ditampilkan, dengan syarat jumlah baris dan kolom tidak boleh lebih dari 10.

Output :



1. **KESIMPULAN**

Berdasarkan pembahasan praktik diatas dapat disimpulkan bahwa Exception adalah event yang terjadi ketika program nemui kesalahan pada saat instruksi program diajalankan. Exception yang merupakan subclass RunTimeException , Exception yang bukan subclass RunTimeException.

1. **LISTING**

Terlampir.