LAPORAN PROYEK 1 PERTEMUAN 5:   
KASUS COLLECTION FRAMEWORK

A blue and orange logo

AI-generated content may be incorrect.

Disusun oleh:

|  |  |
| --- | --- |
| Nama | : Zaidan Arkaan Azharuddya Asshauqi |
| NIM | : 241524030 |
| Kelas | : D4-1A |

Daftar isi

[KASUS DEFENSIVE PROGRAMMING 3](#_Toc193403565)

[CASE 1: Exceptions Aren't Always Errors 3](#_Toc193403566)

[Soal: 3](#_Toc193403567)

[Jawaban: 4](#_Toc193403568)

[Pemahaman: 5](#_Toc193403569)

[CASE 2: Placing Exception Handlers 6](#_Toc193403570)

[Soal: 6](#_Toc193403571)

[Jawaban: 7](#_Toc193403572)

[Pemahaman: 8](#_Toc193403573)

[CASE 3: Throwing Exceptions 9](#_Toc193403574)

[Soal: 9](#_Toc193403575)

[Jawaban: 10](#_Toc193403576)

[Pemahaman: 11](#_Toc193403577)

[KESIMPULAN 11](#_Toc193403578)

# KASUS DEFENSIVE PROGRAMMING

## CASE 1: Exceptions Aren't Always Errors

### Soal:

File CountLetters.java contains a program that reads a word from the user and prints the number of   
occurrences of each letter in the word. Save it to your directory and study it, then compile and run it   
to see how it works. In reading the code, note that the word is converted to all upper case first, then   
each letter is translated to a number in the range 0..25 (by subtracting 'A') for use as an index. No test   
is done to ensure that the characters are in fact letters.

1. Run CountLetters and enter a phrase, that is, more than one word with spaces or other punctuation   
in between. It should throw an ArrayIndexOutOfBoundsException, because a non-letter will   
generate an index that is not between 0 and 25. It might be desirable to allow non-letter   
characters, but not count them. Of course, you could explicitly test the value of the character to   
see if it is between 'A’ and 'Z'. However, an alternative is to go ahead and use the translated   
character as an index, and catch an ArrayIndexOutOfBoundsException if it occurs. Since you   
want don't want to do anything when a non-letter occurs, the handler will be empty. Modify this   
method to do this as follows:

- Put the body of the first for loop in a try.

- Add a catch that catches the exception, but don't do anything with it. Compile and run your program.

2. Now modify the body of the catch so that it prints a useful message (e.g., "Not a letter") followed   
by the exception. Compile and run the program. Although it's useful to print the exception for   
debugging, when you're trying to smoothly handle a condition that you don't consider erroneous   
you often don't want to. In your print statement, replace the exception with the character that   
created the out of bounds index. Run the program again; much nicer!

A screenshot of a computer code

AI-generated content may be incorrect.

### Jawaban:

Program tersebut bertujuan untuk menghitung setiap huruf yang ada dalam text yang diinput oleh user, setelah dijalankan, jika user menginput karakter selain huruf seperti spasi ‘ ‘ dan karakter spesial ‘!@#$%^&\*()\_<>,.” Akan menyebabkan error dan program crash. Untuk itu kita dapat menambahkan exception handlers yaitu try and catch.

#### 1. Menambahkan for loop in try and catch tetapi tidak menambahkan apapun didalam bodynya

import java.util.Scanner;

public class CountLetters{

    public static void main(String[] args){

        int[] counts = new int[26];

        Scanner scan = new Scanner(System.in);

        //get word from user

        System.out.print("Enter a single word (letters only, please): ");

        String word = scan.nextLine();

        //convert to all upper case

        word = word.toUpperCase();

        //count frequency of each letter in string

        // Menambahkan for loop in a try dan menambahkan catch block yang tidak melakukan apa-apa

        for (int i=0; i < word.length(); i++)

            try {

                counts[word.charAt(i)-'A']++;

            }

            catch (ArrayIndexOutOfBoundsException e) {

            }

        //print frequencies

        System.out.println();

        for (int i=0; i < counts.length; i++)

            if (counts [i] != 0)

                System.out.println((char)(i +'A') + ": " + counts[i]);

        }

}

Setelah menambahkan for loop in try dan catch tanpa menambahkan apapun didalam bodynya, program akan dapat berjalan terus ketika kita menginput tanda spasi ‘ ‘ atau karakter spesial.

A screen shot of a computer

AI-generated content may be incorrect.

#### 2. Modifikasi try catch dengan menambahkan pesan “Not a letter: (hurufnya)”

import java.util.Scanner;

public class CountLetters{

    public static void main(String[] args){

        int[] counts = new int[26];

        Scanner scan = new Scanner(System.in);

        //get word from user

        System.out.print("Enter a single word (letters only, please): ");

        String word = scan.nextLine();

        //convert to all upper case

        word = word.toUpperCase();

        //count frequency of each letter in string

        // Menambahkan for loop in a try dan menambahkan catch block yang tidak melakukan apa-apa

        for (int i=0; i < word.length(); i++)

            try {

                counts[word.charAt(i)-'A']++;

            }

            catch (ArrayIndexOutOfBoundsException e) {

                System.out.println("Not a letter: " + word.charAt(i));

            }

        //print frequencies

        System.out.println();

        for (int i=0; i < counts.length; i++)

            if (counts [i] != 0)

                System.out.println((char)(i +'A') + ": " + counts[i]);

        }

}

Setelah menambahkan pesan “Not a letter: (letternya)”, sekarang kita dapat melihat exception for debugging.

A screen shot of a computer

AI-generated content may be incorrect.

### Pemahaman:

Exception adalah mekanisme yang digunakan untuk menangani kondisi tak terduga seperti mencegah program crash, untuk masalah program crash ini yang disebabkan oleh error, error dalam kasus diatas masih dapat ditangani sehingga kita menggunakan try catch exceptions.

## CASE 2: Placing Exception Handlers

### Soal:

File ParseInts.java contains a program that does the following:

* Prompts for and reads in a line of input
* Uses a second Scanner to take the input line one token at a time and parses an integer from each token as it is extracted.
* Sums the integers.
* Prints the sum.

Save ParseInts to your directory and compile and run it. If you give it the input

10 20 30 40

it should print

The sum of the integers on the line is 100.

Try some other inputs as well. Now try a line that contains both integers and other values, e.g.,

We have 2 dogs and 1 cat.

You should get a NumberFormatException when it tries to call *Integer.parseInt* on "We", which is not an integer. One way around this is to put the loop that reads inside a try and catch the NumberFormatException but not do anything with it. This way if it's not an integer it doesn't cause an error; it goes to the exception handler, which does nothing. Do this as follows:

* Modify the program to add a try statement that encompasses the entire while loop. The try and opening { should go before the while, and the catch after the loop body. Catch a NumberFormatException and have an empty body for the catch.
* Compile and run the program and enter a line with mixed integers and other values. You should find that it stops summing at the first non-integer, so the line above will produce a sum of 0, and the line "1 fish 2 fish" will produce a sum of 1. This is because the entire loop is inside the try, so when an exception is thrown the loop is terminated. To make it continue, move the try and catch inside the loop. Now when an exception is thrown, the next statement is the next iteration of the loop, so the entire line is processed. The dogs-and-cats input should now give a sum of 3, as should the fish input.

A screenshot of a computer code

AI-generated content may be incorrect.

### Jawaban:

Program tersebut bertujuan untuk menghitung angka dari text yang diinput oleh user, lalu menjumlahkan angka yang ada pada text tersebut. Untuk inputan normal seperti angka saja, program akan berjalan normal dan penjumlahan nya pun berjalan lancer, tetapi ketika user menginput karakter yang bukan integer, akan menyebabkan error NumberFormatException.

#### 1. Menambahkan Exception handlers diluar loop

import java.util.Scanner;

public class ParseInts{

    public static void main(String[] args){

        int val, sum=0;

        Scanner scan = new Scanner(System.in);

        String line;

        System.out.println("Enter a line of text");

        Scanner scanLine = new Scanner(scan.nextLine());

        try {

        while (scanLine.hasNext()){

                val = Integer.parseInt(scanLine.next());

                sum += val;

}

        } catch (NumberFormatException e) {

        }

        System.out.println("The sum of the integers on this line is " + sum);

    }

}

Output:

A black background with white text

AI-generated content may be incorrect.

Mengapa jumlah dari sum integer adalah 0? Padahal di line tersebut ada angka 123 dan 123, seharusnya sum integer adalah 246. Ini disebabkan karena cara kerja try catch ini adalah ketika suatu kode dalam blok try mengalami error (exception) maka program akan langsung melompat ke blok catch, tanpa mengeksekusi kode setelah error dalam try, oleh karena itu ketika membaca ada huruf di loop tersebut, kode dalam blok try yaitu loopnya akan berhenti dan lansung menyetak jumlah dari sum yang belum diisi apa apa.

#### 2. Menambahkan Exception handlers didalam loop

import java.util.Scanner;

public class ParseInts{

    public static void main(String[] args){

        int val, sum=0;

        Scanner scan = new Scanner(System.in);

        String line;

        System.out.println("Enter a line of text");

        Scanner scanLine = new Scanner(scan.nextLine());

        while (scanLine.hasNext()){

            try {

                val = Integer.parseInt(scanLine.next());

                sum += val;

            } catch (NumberFormatException e) {

            }

        }

        System.out.println("The sum of the integers on this line is " + sum);

    }

}

Output:

A black background with white text

AI-generated content may be incorrect.

Setelah menaruh exception didalam loop, loop tersebut akan terus mengecek hingga tidak ada input lagi. Oleh karena itu sum akan bertambah berdasarkan angka yang diinput dan mengabaikan karakter selain integer dari input.

### Pemahaman:

Untuk menangani exception handlers, perlu penempatan try catch yang tepat untuk menangani error tersebut, seperti menaruh exception handlers dalam loop dikarenakan kita ingin mengecek setiap index dari scanLine.hasNext oleh karena itu kita dapat menaruh exception handlers dalam loop, jika kita menaruhnya diluar loop maka loop akan berhenti ketika index pertamanya langsung error.

## CASE 3: Throwing Exceptions

### Soal:

File Factorials.java contains a program that calls the factorial method of the MathUtils class to compute the factorials of integers entered by the user. Save these files to your directory and study the code in both, then compile and run Factorials to see how it works. Try several positive integers, then try a negative number. You should find that it works for small positive integers (values < 17), but that it returns a large negative value for larger integers and that it always returns 1 for negative integers.

1. Returning 1 as the factorial of any negative integer is not correct—mathematically, the factorial function is not defined for negative integers. To correct this, you could modify your factorial method to check if the argument is negative, but then what? The method must return a value, and even if it prints an error message, whatever value is returned could be misconstrued. Instead it should throw an exception indicating that something went wrong so it could not complete its calculation. You could define your own exception class, but there is already an exception appropriate for this situation—IllegalArgumentException, which extends RuntimeException. Modify your program as follows:

• Modify the header of the factorial method to indicate that factorial can throw an IllegalArgumentException.

• Modify the body of factorial to check the value of the argument and, if it is negative, throw an IllegalArgumentException. Note that what you pass to throw is actually an instance of the IllegalArgumentException class, and that the constructor takes a String parameter. Use this parameter to be specific about what the problem is.

• Compile and run your Factorials program after making these changes. Now when you enter a negative number an exception will be thrown, terminating the program. The program ends because the exception is not caught, so it is thrown by the main method, causing a runtime error.

• Modify the main method in your Factorials class to catch the exception thrown by factorial and print an appropriate message, but then continue with the loop. Think carefully about where you will need to put the try and catch.

1. Returning a negative number for values over 16 also is not correct. The problem is arithmetic overflow—the factorial is bigger than can be represented by an int. This can also be thought of as an IllegalArgumentException—this factorial method is only defined for arguments up to 16. Modify your code in factorial to check for an argument over 16 as well as for a negative argument. You should throw an IllegalArgumentException in either case, but pass different messages to the constructor so that the problem is clear.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

### Jawaban:

Header dari file Factorials.java

public class MathUtils{

    //-------------------------------------------------------------

    // Returns the factorial of the argument given

    //-------------------------------------------------------------

    public static int factorial(int n) throws IllegalArgumentException{

        if (n < 0) {

            throw new IllegalArgumentException("Factorial is not defined for negative numbers: " + n);

        }

        else if(n > 16) {

            throw new IllegalArgumentException("Input too large: " + n);

        }

        int fac = 1;

        for (int i=n; i>0; i--)

            fac \*= i;

        return fac;

    }

}

Body dari file Factorials.java

public class MathUtils{

    //-------------------------------------------------------------

    // Returns the factorial of the argument given

    //-------------------------------------------------------------

    public static int factorial(int n) throws IllegalArgumentException{

        if (n < 0) {

            throw new IllegalArgumentException("Factorial is not defined for negative numbers: " + n);

        }

        else if(n > 16) {

            throw new IllegalArgumentException("Input too large: " + n);

        }

        int fac = 1;

        for (int i=n; i>0; i--)

            fac \*= i;

        return fac;

    }

}

Kita dapat menggunakan throw new () untuk menangani error secara lebih spesifik serta meghentikan eksekusi kode yang tidak valid. Sehigga program dapat berjalan dengan lancar. Untuk penggunaan throw new () ini juga dapat memaksa pengguna untuk benar-benar menghandle error, ketika melebihi nilai dari int ketika input n lebih dari 16 pun program akan terus berjalan.

### Pemahaman:

Dengan menggunakan throw new() kita dapat menghandle error secara lebih spesifik.

## KESIMPULAN

Untuk menangani error, kita dapat menggunakan exceptions handlers, exceptions handlers juga dapat berfungsi ketika menangani hal yang tidak error. Penempatan exceptions handlers seperti try catch harus berada didalam for, untuk mengecek setiap kali loop itu berjalan. Kita juga dapat menggunakan Throwing Exceptions untuk menangani error yang lebih spesifik dan memberikan pesan yang jelas.