PRAKTIKUM

-> EXCEPTION HANDLING

1. Contoh menggunakan throw (exception handling)

```
class Mahasiswa {
      private String nim;
      private String nama;

      public void setNIM(String inputNIM) {
            try {
                  nim = inputNIM;
                  if (inputNIM == null) {
                        throw new NullPointerException();
                  }
            } catch (NullPointerException npe) {
                  System.out.println("KESALAHAN: " + "NIM tidak boleh null");
            }
      }

      public String getNIM() {
            return nim;
      }

      public void setNama(String inputNama) {
try {
nama = inputNama;
if (nama == null) {
throw new NullPointerException();
}
} catch (NullPointerException npe) {
System.out.println("KESALAHAN: " + "Nama mahasiswa tidak boleh
null");
}
}

      public String getNama() {
            return nama;
      }
}

class DemoThrow {
      public static void main(String[] args) {
            Mahasiswa mhs = new Mahasiswa();
            mhs.setNIM(null);
            mhs.setNama("Nadilla");
            System.out.println("\nNIM : " + mhs.getNIM());
            System.out.println("Nama : " + mhs.getNama());
      }
}
```

2. Contoh menggunakan throws (exception handling)

```
//Read an array of bytes from the keyboard
import java.io.*;
class ReadBytes {
      public static void main (String args []) throws IOException {
            byte data [] = new byte [10];
            System.out.println("Enter Some Characters : ");
            System.in.read(data); // read an array bytes from key board
            System.out.print("You Entered : ");
            for(int i=0; i <data.length;i++)
                  System.out.print((char) data[i]);
      }
}
```

3. Contoh menggunakan finally (exception handling)

```
class DemoFinally {
      public static void cobaEksepsi(int pembilang, int penyebut) {
            try {
                  int hasil = pembilang / penyebut;
                  System.out.println("Hasil bagi: " + hasil);
                  int[] Arr = { 1, 2, 3, 4, 5 }; // array dengan 5 elemen
                  Arr[10] = 23; // mengakses indeks ke-10
            } catch (ArithmeticException eksepsi1) {
                  System.out.println("Terdapat pembagian dengan 0");
            } catch (ArrayIndexOutOfBoundsException eksepsi2) {
                  System.out.println("Indeks di luar rentang");
            } finally {
                  System.out.println("Ini adalah statemen dalam blok finally");
            }
      }

      public static void main(String[] args) {
            cobaEksepsi(4, 0); // menimbulkanArithmeticException
            System.out.println();
            cobaEksepsi(12, 4); // menimbulkan ArrayIndexOutOfBoundsException
      }
}
```

-> STREAM

4. Stream menggunakan byte stream

```
//Read an array of bytes from the keyboard
import java.io.*;

class ReadBytes {
      public static void main(String args[]) throws IOException {
            byte data[] = new byte[10];
            System.out.println("Enter Some Characters : ");
            System.in.read(data); // read an array bytes from key board
            System.out.print("You Entered : ");
            for (int i = 0; i < data.length; i++)
                  System.out.print((char) data[i]);
      }
}
```

5. Stream menggunakan character stream

```
import java.io.*;

class bacaKarakter {
      public static void main(String args[]) {
            char c;
            try {
                  BufferedReader br = new BufferedReader(new InputStreamReader(
                              System.in));
                  System.out.println("Masukan karakter, dan akhiri dengan q");
                  do {
                        c = (char) br.read();
                        System.out.println("Terbaca karakter : " + c);
                  } while (c != 'q');
            } catch (IOException e) {
                  System.out.println("KESALAHAN IO");
            }
            System.exit(0);
      }
}
```

-> AKSES FILE

6. Menulis ke dalam file (akses file)

```
import java.io.PrintWriter;
import java.io.FileOutputStream;
import java.io.FileNotFoundException;

public class TextFileOutputDemo {
      public static void main(String[] args) {
            PrintWriter outputStream = null;
            try {
                  outputStream = new PrintWriter(new
FileOutputStream("stuff.txt"));
            } catch (FileNotFoundException e) {
                  System.out.println("Error opening the file stuff.txt.");
                  System.exit(0);
            }
            System.out.println("Writing to file.");
            outputStream.println("Saya sedang menulis.");
            outputStream.println("Menulis kedalam file.");
            outputStream.close();
            System.out.println("End of program.");
      }
}
```

7. Akses dan Menulis ke dalam Suatu File (akses file)

```
import java.util.Scanner;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.io.FileOutputStream;

public class HasNextLineDemo2 {
      public static void main(String[] args) throws FileNotFoundException {
            Scanner inputStream = null;
            PrintWriter outputStream = null;
            inputStream = new Scanner(new FileInputStream("/home/stuff.txt"));
            // outputStream = new PrintWriter(new 
FileOutputStream("numbered.txt"));
            String line = null;
            // int count = 0;
            while (inputStream.hasNextLine()) {
                  line = inputStream.nextLine();
                  // count++;
                  System.out.println(line);
                  // outputStream.println(count + " " + line);
            }
            inputStream.close();
            // outputStream.close( );
      }
}
```