# Modul Praktikum 11 :
# Exception Handling

## 1. Tujuan Praktikum

- Memahami konsep exception handling dalam pemrograman.
- Mampu menangani error menggunakan mekanisme try-catch-finally.
- Mampu membuat exception handling sendiri (custom exception).
- Mengerti pentingnya penggunaan exception untuk membuat program yang robust.

## 2. Prasyarat

- Menguasai dasar pemrograman bahasa yang digunakan (Java/Python/C++/...).
- Memahami struktur kontrol (if, while, dll).

## 3. Materi Singkat

### Apa itu Exception?

Exception adalah kejadian yang terjadi saat program berjalan yang menyebabkan program berhenti jika tidak ditangani dengan benar, misalnya pembagian dengan nol, akses indeks array di luar batas, atau kesalahan input.

Cara Menangani Exception

- Try: blok kode yang mungkin menimbulkan error.
- Catch: blok kode yang menangani error.
- Finally: blok kode yang tetap dijalankan, baik terjadi error atau tidak.
- Throw / Raise: melempar exception secara manual.

## 4. Studi Kasus & Latihan

### 4.1 Contoh Dasar Try-Catch (Java)

```
public class ExceptionDemo {
    public static void main(String[] args) {
        try {
            int result = 10 / 0;  // Error: division by zero
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Terjadi error: " + e.getMessage());
        } finally {
            System.out.println("Blok finally selalu dijalankan.");
        }
    }
}
```

### Latihan 1

Buat program yang meminta input angka dari user, kemudian melakukan pembagian dengan angka yang dimasukkan user tersebut. Tangani error apabila user memasukkan angka nol.

### Jawaban Latihan 1 :

```
import java.util.Scanner;

public class LatihanExceptionHandling {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        try {
            System.out.print("Masukkan angka pembagi: ");
            int divisor = scanner.nextInt();
            int result = 100 / divisor;
            System.out.println("Hasil pembagian 100 / " + divisor + " = " + result);
        } catch (ArithmeticException e) {
            System.out.println("Error: Pembagian dengan nol tidak diperbolehkan!");
        } catch (Exception e) {
            System.out.println("Error: Input tidak valid!");
        } finally {
            scanner.close();
            System.out.println("Program selesai.");
        }
    }
}
```

## 4.2 Multiple Catch (Java)

```
try {
    int[] arr = new int[3];
    arr[5] = 10;  // Error: indeks di luar batas
} catch (ArithmeticException e) {
    System.out.println("Error aritmatika: " + e.getMessage());
} catch (ArrayIndexOutOfBoundsException e) {
    System.out.println("Error indeks array: " + e.getMessage());
}
```

### Latihan 2

Buat program yang meminta input angka dari user, memasukkannya ke array pada indeks tertentu. Tangani error indeks yang salah dan error pembagian dengan nol jika ada.

## Jawaban Latihan 2 :

```
import java.util.Scanner;

public class Latihan2ExceptionHandling {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[] arr = new int[5];  // Array dengan 5 elemen indeks 0-4

        try {
            System.out.print("Masukkan indeks array (0-4): ");
            int index = scanner.nextInt();

            System.out.print("Masukkan angka untuk dibagi 100: ");
            int divisor = scanner.nextInt();

            int result = 100 / divisor;
            arr[index] = result;

            System.out.println("Hasil pembagian 100 / " + divisor + " disimpan di arr[" + index + "]
= " + arr[index]);
        } catch (ArithmeticException e) {
            System.out.println("Error: Pembagian dengan nol tidak diperbolehkan!");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: Indeks array di luar batas! Gunakan indeks antara 0 sampai
4.");
        } catch (Exception e) {
            System.out.println("Error: Input tidak valid!");
        } finally {
            scanner.close();
            System.out.println("Program selesai.");
        }
    }
}
```

## 4.3 Membuat Custom Exception (Java)

```
class MyException extends Exception {
    public MyException(String message) {
        super(message);
    }
}

public class TestCustomException {
    public static void checkValue(int num) throws MyException {
        if (num < 1) {
            throw new MyException("Nilai harus lebih dari 0");
        }
        System.out.println("Nilai valid: " + num);
    }

    public static void main(String[] args) {
        try {
            checkValue(0);
        } catch (MyException e) {
            System.out.println("Custom exception: " + e.getMessage());
        }
    }
}
```

## Latihan 3

Buat custom exception untuk validasi umur minimal 17 tahun saat mendaftar.

## Jawaban Latiha 3:

```
// Definisi Custom Exception
class AgeValidationException extends Exception {
    public AgeValidationException(String message) {
        super(message);
    }
}

public class ValidasiUmur {
    public static void validateAge(int age) throws AgeValidationException {
        if (age < 17) {
            throw new AgeValidationException("Umur harus minimal 17 tahun untuk
mendaftar.");
        }
    }

    public static void main(String[] args) {
        java.util.Scanner scanner = new java.util.Scanner(System.in);
        try {
            System.out.print("Masukkan umur Anda: ");
            int umur = scanner.nextInt();

            validateAge(umur);
            System.out.println("Pendaftaran berhasil. Umur Anda: " + umur);
        } catch (AgeValidationException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("Input tidak valid!");
        } finally {
            scanner.close();
            System.out.println("Program selesai.");
        }
    }
}
```

## 5. Tugas Praktikum

1. Buat program kalkulator sederhana dengan operasi tambah, kurang, kali, bagi.
2. Tangani exception untuk input yang bukan angka.
3. Tangani exception pembagian dengan nol.
4. Buat custom exception untuk validasi operasi yang tidak tersedia.

## Jawaban Tugas :

```
import java.util.Scanner;

// Custom Exception untuk operasi yang tidak valid
class InvalidOperationException extends Exception {
    public InvalidOperationException(String message) {
        super(message);
    }
}

public class Kalkulator {
```

```
    public static double kalkulasi(double a, double b, String operasi) throws
InvalidOperationException {
        switch (operasi.toLowerCase()) {
            case "tambah":
            case "+":
                return a + b;
            case "kurang":
            case "-":
                return a - b;
            case "kali":
            case "*":
                return a * b;
            case "bagi":
            case "/":
                if (b == 0) {
                    throw new ArithmeticException("Pembagian dengan nol tidak diperbolehkan.");
                }
                return a / b;
            default:
                throw new InvalidOperationException("Operasi '" + operasi + "' tidak tersedia.");
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Masukkan angka pertama: ");
            double angka1 = Double.parseDouble(scanner.nextLine());

            System.out.print("Masukkan operasi (tambah, kurang, kali, bagi): ");
            String operasi = scanner.nextLine();

            System.out.print("Masukkan angka kedua: ");
            double angka2 = Double.parseDouble(scanner.nextLine());

            double hasil = kalkulasi(angka1, angka2, operasi);
            System.out.println("Hasil: " + hasil);

        } catch (NumberFormatException e) {
            System.out.println("Error: Input harus berupa angka.");
        } catch (ArithmeticException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (InvalidOperationException e) {
            System.out.println("Error: " + e.getMessage());
        } finally {
            scanner.close();
            System.out.println("Program selesai.");
        }
    }
}
```