

Praktikum Day #4

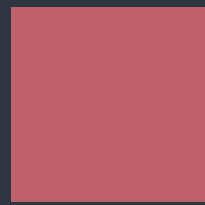
Exception Handling

Lab Assistant :
Fikri & Jo



Overview

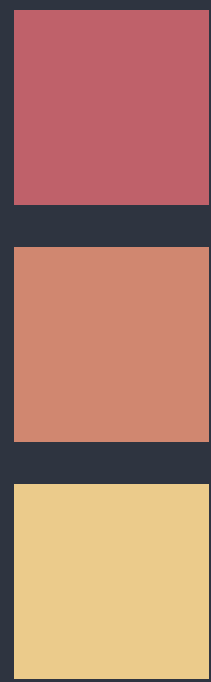
- Exception
- Jenis Error
- Try Catch Finally
- Throw and Throws



Apa itu Exception?



SEBUAH **EVENT** YANG AKAN
MENGGANGGU ALUR PROSES
PROGRAM NORMAL DARI SEBUAH
PROGRAM



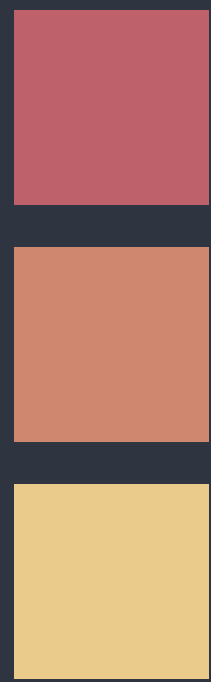
**BILA TIDAK DITANGANI MAKA
MENGAKIBATKAN PROGRAM
TERMINATE (TERPAKSA BERHENTI)**




3 Jenis Error



1. *Syntax* Error



Melanggar aturan
sintaks bahasa
pemrograman,
ditemukan saat
kompilasi oleh
kompiler



```
public class SyntaxErrorExample {  
    public static void main(String[] args) {  
        int x = 5  
        int y = 10;  
        System.out.println("Jumlah: " + (x + y))  
    }  
}
```



2. Logic Error

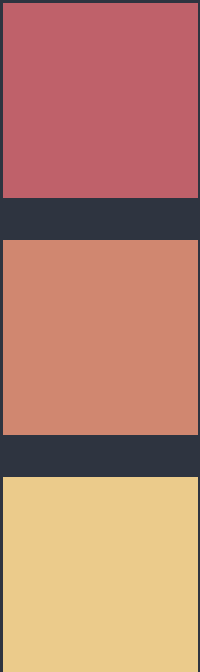


Kesalahan logika,
menghasilkan
output/performa
yang menyimpang

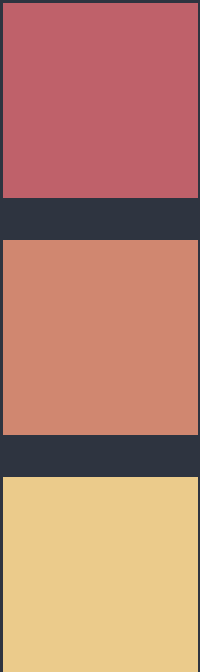
```
public class LogicErrorExample {  
    public static void main(String[] args) {  
        int[] numbers = {10, 20, 30, 40};  
        int sum = 0;  
  
        // Logic error: menghitung jumlah dengan salah  
        for (int i = 0; i <= numbers.length; i++) {  
            sum += numbers[i];  
        }  
  
        // Logic error: rumus rata-rata salah (seharusnya dibagi jumlah elemen)  
        double average = sum / 2;  
  
        System.out.println("Rata-rata: " + average);  
    }  
}
```



3. Runtime Error



Operasi yang salah
saat eksekusi
program



```
public class ArrayIndexExceptionExample {  
    public static void main(String[] args) {  
        int[] numbers = {1, 2, 3, 4, 5};  
  
        // Akan menyebabkan ArrayIndexOutOfBoundsException  
        System.out.println("Elemen ke-10: " + numbers[9]);  
    }  
}
```

Output :

Exception in thread "main"

**java.lang.ArrayIndexOutOfBoundsException: Index 9 out of bounds
for length 5**



Contoh Exception



ArrayIndexOutOfBoundsException

Terjadi saat mengakses indeks array yang tidak tersedia

```
public class ArrayIndexExample {  
    public static void main(String[] args) {  
        int[] numbers = {10, 20, 30, 40, 50};  
  
        try {  
            System.out.println("Mengakses indeks ke-5: " + numbers[5]);  
        } catch (ArrayIndexOutOfBoundsException e) {  
            System.out.println("Error: Indeks array melebihi batas!");  
            System.out.println("Pesan error: " + e.getMessage());  
        }  
    }  
}
```

Error: Indeks array melebihi batas!

Pesan error: Index 5 out of bounds for length 5

NumberFormatException

exception yang terjadi jika kita melakukan passing parameter non-number ke method Integer.parseInt()

```
public class EmptyStringExample {  
    public static void main(String[] args) {  
        String emptyString = "";  
  
        try {  
            int number = Integer.parseInt(emptyString);  
            System.out.println("Angka yang dikonversi: " + number);  
        } catch (NumberFormatException e) {  
            System.out.println("Error: String kosong tidak bisa dikonversi ke angka");  
            System.out.println("Pesan error: " + e.getMessage());  
        }  
    }  
}
```

Error: String kosong tidak bisa dikonversi ke angka
Pesan error: For input string: ""



InputMismatchException

Terjadi saat input Scanner tidak sesuai

```
import java.util.Scanner;
import java.util.InputMismatchException;

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

        try {
            System.out.print("Masukkan umur Anda (angka): ");
            int age = scanner.nextInt();
            System.out.println("Umur Anda: " + age);
        } catch (InputMismatchException e) {
            System.out.println("Error: Input harus berupa angka!");
        } finally {
            scanner.close();
        }
    }
}
```

Error: Input harus berupa angka!



ArithmeticException


Terjadi saat operasi aritmatika tidak valid

```
public class ArithmeticExample {  
    public static void main(String[] args) {  
        int numerator = 10;  
        int denominator = 0;  
  
        try {  
            int result = numerator / denominator;  
            System.out.println("Hasil pembagian: " + result);  
        } catch (ArithmeticException e) {  
            System.out.println("Error: Pembagian dengan nol tidak diperbolehkan!");  
            System.out.println("Pesan error: " + e.getMessage());  
        }  
    }  
}
```

```
Error: Pembagian dengan nol tidak diperbolehkan!  
Pesan error: / by zero
```



**Contoh sederhana
alur try catch finally**



```
public class SederhanaTryCatchFinally {
    public static void main(String[] args) {
        System.out.println("Program dimulai");

        try {
            System.out.println("\nMasuk ke blok try");
            int hasil = 10 / 0; // Akan menyebabkan ArithmeticException
            System.out.println("Ini tidak akan dieksekusi");
        }
        catch (ArithmeticException e) {
            System.out.println("Masuk ke blok catch");
            System.out.println("Terjadi error: " + e.getMessage());
        }
        finally {
            System.out.println("Masuk ke blok finally (selalu dieksekusi)");
        }

        System.out.println("\nProgram melanjutkan eksekusi");
    }
}
```



Output

Program dimulai

Masuk ke blok try

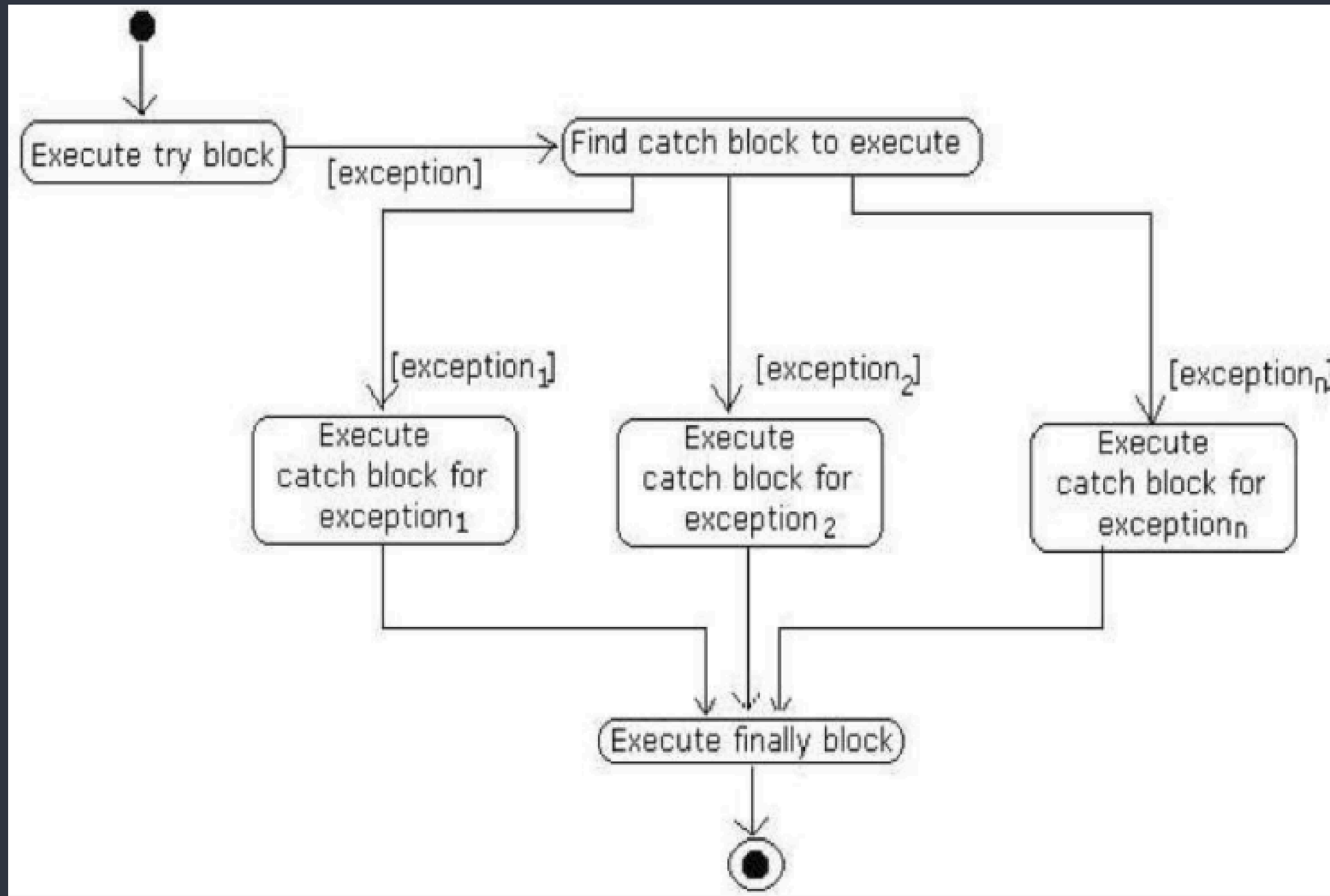
Masuk ke blok catch

Terjadi error: / by zero

Masuk ke blok finally (selalu dieksekusi)

Program melanjutkan eksekusi

Alur Program



Finally dan Throw



Finally adalah blok yang selalu dikerjakan apapun yang terjadi didalam blok Exception tersebut. Keyword finally ini hanya ditulis sekali.

Throw adalah keyword yang berfungsi untuk melemparkan kesalahan dengan sengaja, maksudnya adalah agar blok catch dikerjakan/dieksekusi oleh program.

Contoh

Output :

```
Transaksi gagal: Jumlah pembayaran tidak valid: -100  
Log transaksi disimpan ke database
```

```
public class ThrowFinallyCombo {  
    public static void main(String[] args) {  
        try {  
            processPayment(-100);  
        }  
        catch (IllegalArgumentException e) {  
            System.out.println("Transaksi gagal: " + e.getMessage());  
        }  
        finally {  
            System.out.println("Log transaksi disimpan ke database");  
        }  
    }  
  
    public static void processPayment(int amount) {  
        if (amount <= 0) {  
            throw new IllegalArgumentException(  
                "Jumlah pembayaran tidak valid: " + amount  
            );  
        }  
        System.out.println("Memproses pembayaran: Rp" + amount);  
    }  
}
```



Apa itu **Throws**???

Throws adalah kata kunci di Java yang digunakan untuk memberi tahu bahwa suatu method mungkin menyebabkan **error (exception)**, tetapi method tersebut **tidak menangani errornya sendiri**.

- Artinya, error akan "**dilempar**" ke kode yang memanggil method tersebut.
- Biasanya digunakan untuk **checked exceptions** (error yang wajib ditangani).

Contoh :

Bayangkan kamu punya **method** untuk membaca file, tapi **file-nya** belum tentu ada.

```
import java.io.*;

public class BacaFile {

    // Method ini memberi tahu: "Aku bisa error FileNotFoundException!"
    public static void baca(String namaFile) throws FileNotFoundException {
        File file = new File(namaFile);
        Scanner scanner = new Scanner(file); // Mungkin error jika file tidak ada

        while (scanner.hasNextLine()) {
            System.out.println(scanner.nextLine());
        }
        scanner.close();
    }

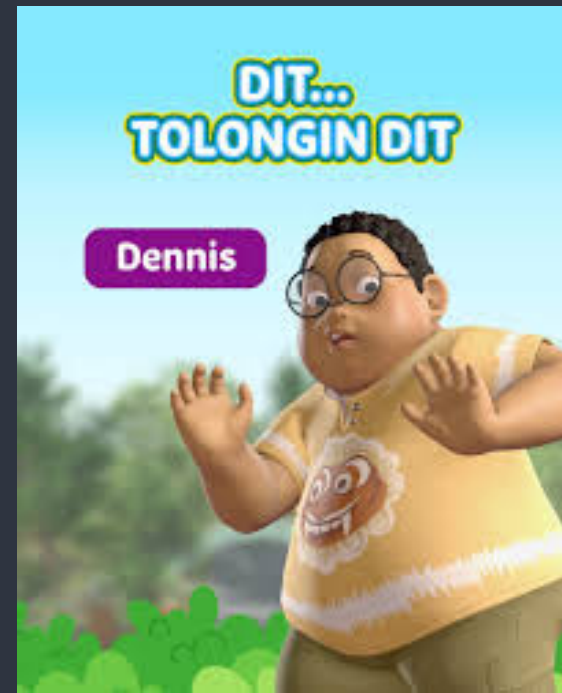
    public static void main(String[] args) {
        try {
            baca("data.txt"); // Panggil method yang mungkin error
        }
        catch (FileNotFoundException e) {
            System.out.println("File tidak ditemukan!"); // Tangkap errornya di sini
        }
    }
}
```

Bahasa Bayi

Method **baca()** punya **throws** **FileNotFoundException** →
Artinya:

"Hee rek aku bisa gagal kalau filenya tidak ada, tapi aku gak mau tanggung jawab menangani error-nya!"

Saat dipanggil di **main()**, wajib pakai **try-catch** karena **FileNotFoundException** adalah **checked exception**.
Mudahnya **Adit** sebagai **main method** harus membantu method yang ia panggil (**Denis**).



Perbedaan `throw` vs `throws`

`throw`

`throws`

Fungsi Melempar error secara manual

Mendeklarasikan bahwa method bisa error

Posisi Di dalam body method

Di deklarasi method (sebelum `{ }`)

Contoh `throw new Exception("Error!");`

`void bacaFile() throws IOException`

TUGAS

KERJAKAN SOAL ETS
TAHUN LALU DAN
QUIZ 1 PAK BUDI
TAHUN LALU





[https://linktr.ee/
swayanghimatika](https://linktr.ee/swayanghimatika)



@swayanghimatika

*** /swayanghimatika**