

VERSION 1.0  
FEBRUARI, 2021



# PEMOGRAMAN BERORIENTASI OBJEK

JAVA EXCEPTION. MODUL 5

TIM PENYUSUN:

- GALIH WASIS WICAKSONO S.KOM, .M.CS.
- FARLI NAHRUL JAVIER
- MUHAMMAD NUR ICHSAN

PRESENTED BY: LAB. INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH MALANG

## PEMOGRAMAN BERORIENTASI OBJEK

### CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mampu memahami dan menerapkan :

1. Java Exception Class
2. Exception handler dengan statement try catch
3. Memahami exception stack
4. Exception handler dengan statement throws

### SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mampu mengimplementasikan :

1. Membuat Class Exception sendiri
2. Multiple catch
3. Melempar exception dengan statement throw
4. Menggunakan finally

### KEBUTUHAN HARDWARE & SOFTWARE

1. Compiler java (JDK), JRE.
2. Editor Java (Intelij IDEA, NetBeans, Gel, Eclipse, Jcreator, dll).

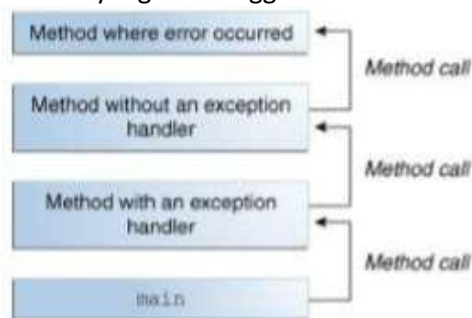
### MATERI POKOK

#### 1. Java Exception

Exception adalah event, yang terjadi ketika program dieksekusi, yang mengakibatkan terganggunya alur program secara normal.

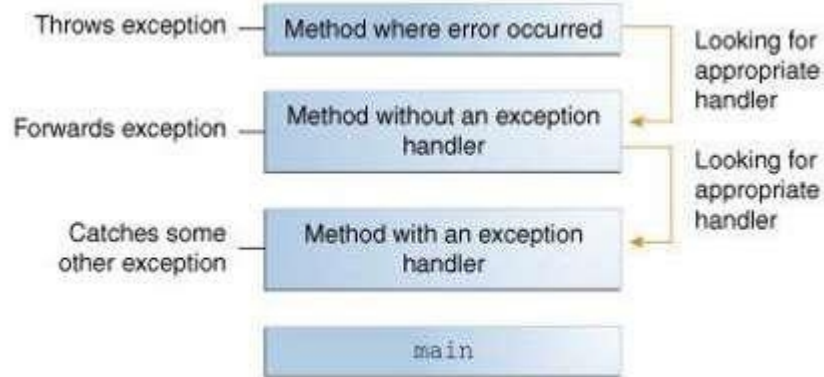
Ketika error terjadi di suatu method, maka method tersebut akan membuat object yang kemudian dikirim ke runtime system. Object ini, yang disebut sebagai exception object, mengandung informasi tentang error yang terjadi, termasuk tipe dan kondisi program pada saat error terjadi. Proses pembuatan exception object dan kemudian mengirimnya ke runtime system ini disebut sebagai “throwing an exception” (melemparkan exception).

Setelah method melemparkan exception, runtime system akan berusaha untuk mencari solusi untuk menangani masalah yang ada. Solusi yang ada bisa jadi lebih dari satu, oleh karena itu runtime system akan memilih berdasarkan suatu urutan yang berdasar pada method tempat terjadinya error dan atau method lain yang memanggil method tersebut.



Runtime system mencari method yang memiliki blok untuk menangani error yang terjadi berdasarkan call stack. Pencarian dimulai dari tempat terjadinya error ,kemudian dirunut berdasarkan call stack secara reverse order. Jika ditemukan blok untuk menangani error yang bersangkutan, maka system akan menyerahkan exception ke handler.

Penyerahan exception ini disebut sebagai catch the exception (menangkap exception). Jika system tidak menemukan handler yang sesuai di call stack, maka runtime system akan menghentikan jalannya program.



**a. Checked exceptions**

Merupakan exception yang disebabkan oleh kesalahan pemakai program atau hal lain yang dapat diprediksi oleh pemrogram. Contohnya, jika ingin membuka file tapi file tidak ditemukan, maka exception akan muncul.

**b. Runtime exception**

Ini adalah exception yang muncul dimana kemunculannya tidak bisa dihindari oleh pemrogram.

**c. Errors**

Ini sebenarnya bukan exception, namun merupakan masalah yang muncul diluar kendali pemakai dan pemrogram. Error secara umum akan dibiarkan saja, sebab tidak ada yang bisa dilakukan untuk mengatasinya. Sebagai contoh, jika stack overflow muncul, maka error akan muncul.

---

## MATERI PRAKTIKUM

### 1. Exception handler

Exception handler menggunakan kombinasi try dan catch keywords. Kode program yang dapat menimbulkan exception diletakkan di dalam blok try . Sintaks penggunaan try-catch adalah sebagaimana berikut :

```
try
{
//Protected code
}catch(ExceptionName e1)
{
//Blok Catch
}
```

Statement catch disertai dengan deklarasi tipe exception yang ingin ditangani. Jika exception terjadi di bagian protected code, maka blok catch akan melakukan pemeriksaan. Jika tipe exceptionnya ada di dalam blok catch, maka exception yang bersangkutan akan ditangani.

Contoh: Berikut adalah array dua dimensi. Kemudian program akan mengakses elemen ke 3 dari array tersebut, sehingga mengakibatkan exception

```
public class ExcepTest {
    public static void main(String[] args)
    { try {
        int a[] = new int[2];
        System.out.println("Access element three :" + a[3]);

        } catch (ArrayIndexOutOfBoundsException e)
        { System.out.println("Exception thrown :" + e);
        }
        System.out.println("Out of the block");
    }
}
```

Program diatas akan menghasilkan:

Exception thrown :java.lang.ArrayIndexOutOfBoundsException: 3 Out of the block

## 2. Statement throws

Jika method tidak dapat menangani exception, maka method tersebut harus mendeklarasikan jenis exception yang mungkin dilemparkan menggunakan kata throws. Kata throws digunakan pada bagian deklarasi method.

Berikut adalah sebuah method yang melemparkan RemoteException

```
import java.io.*;
public class className {

    public void deposit(double amount) throws RemoteException {
        // Method implementation
        throw new RemoteException();
    }
    // Remainder of class definition
}
```

Pada suatu method dapat didefinisikan lebih dari satu macam keluaran exception yang dipisahkan dengan tanda koma (,). Sebagai contoh: method berikut kemungkinan melemparkan RemoteException dan InsufficientFundsException:

```
import java.io.*;
public class className {

    public void withdraw(double amount) throws RemoteException,
        InsufficientFundsException {
        // Method implementation
    }
    // Remainder of class definition
}
```

## 3. Finally

Keyword ini merupakan keyword yang menunjukkan bahwa block program tersebut akan selalu dieksekusi meskipun adanya kesalahan yang muncul atau pun tidak ada.

**Syntax :**

```
try {
    // Protected code
```

```

} catch (ExceptionType1 e1) {
    // Catch block
} catch (ExceptionType2 e2) {
    // Catch block
} catch (ExceptionType3 e3) {
    // Catch block
}finally {
    // The finally block always executes.
}

```

**Contoh**

```

public class ExcepTest {

    public static void main(String args[])
    { int a[] = new int[2];
      try {
          System.out.println("Access element three :" + a[3]);
      } catch (ArrayIndexOutOfBoundsException e)
          { System.out.println("Exception thrown :" + e);
        }finally {
            a[0] = 6;
            System.out.println("First element value: " + a[0]);
            System.out.println("The finally statement is executed");
        }
    }
}

```

**LEMBAR KERJA**

RekapNilai	Main
- nilai : int - matkul : String - nama : String	
+getNilai() +setNilai() +getMatkul() +setMatkul() +getNama() +setNama()	+main()

- Berdasarkan class diagram diatas,buatlah program java untuk melakukan rekapitulasi nilai menggunakan enkapsulasi dan berikan Exception Handling untuk semua kemungkinan error yang ada dalam program pada saat enkapsulasi misal tidak boleh memasukkan nilai null pada setter.
- Pilih salah satu dari soal berikut:
  - Kembangkan program dari kegiatan 1,berikan inputan dari user untuk memasukkan data pada properti nilai,matkul dan nama.gunakan perulangan tanpa batas untuk input data tersebut, berikan exception handling misal akan berhenti ketika user menghendaki, atau inputan melebihi batas array.
  - Kembangkan program dari kegiatan 1 menggunakan file system dari **java** untuk membaca dan menulis data RekapNilai dari class RekapNilai kedalam sebuah file.kemudian tampilkan semua data rekapan yang telah kamu masukkan,berikan Exception Handling untuk semua kemungkinan error yang ada dalam program. (gunakan semua exception handler termasuk **finally**)
- Jelaskan kepada asisten semua exception handling yang kamu berikan kepada syntax kamu.

FEBRUARI,  
2021

PEMOGRAMAN BERORIENTASI OBJEK

4

---

**RUBRIK PENILAIAN**

KRITERIA	NILAI (%)
Kegiatan 1	25
Kegiatan 2 A   Kegiatan 2 B	30   50
Kegiatan 3	25
<b>Total</b>	<b>100</b>

