

# EKSEPSI (EXCEPTIONS)



# Eksepsi

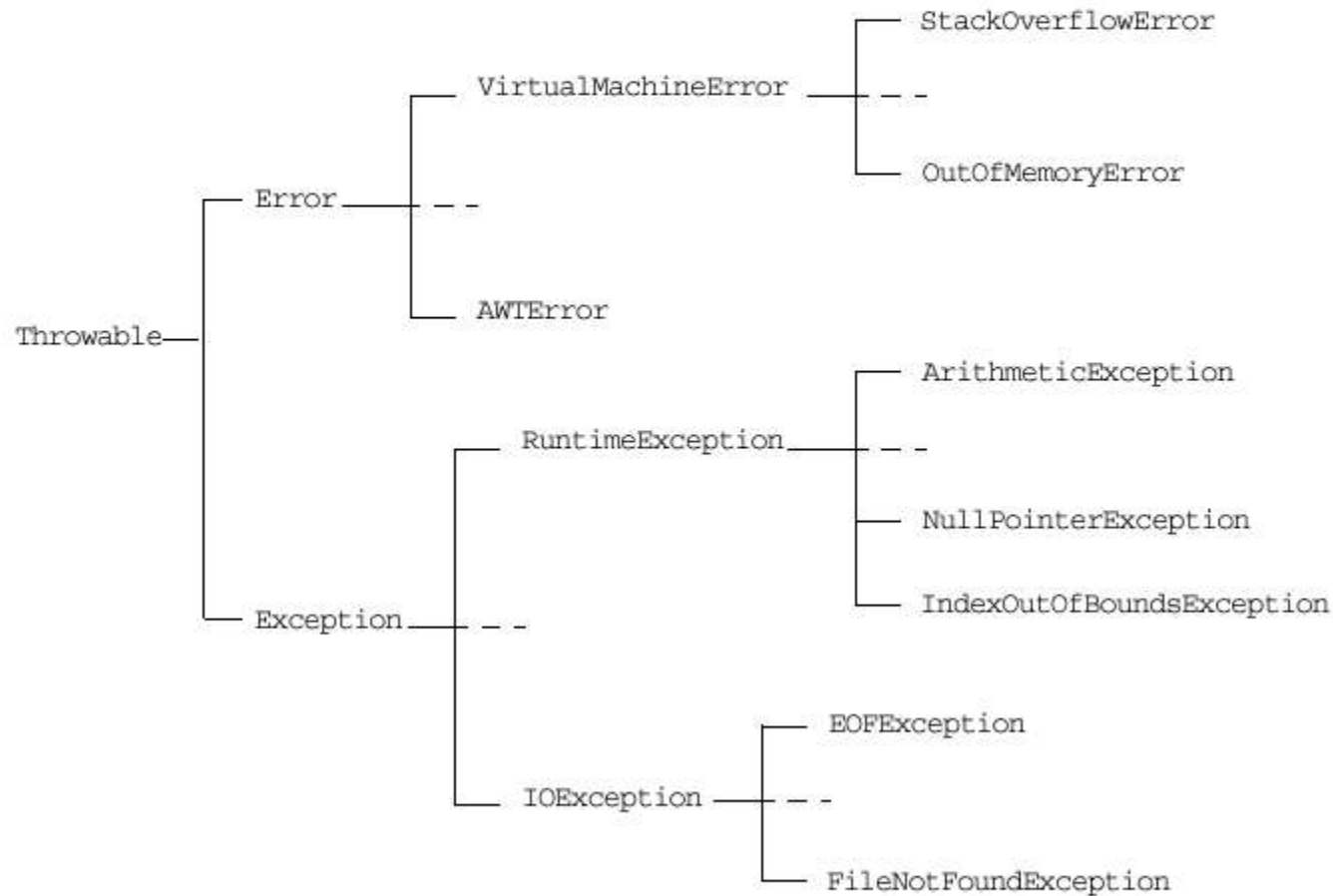
- ♦ Saat Anda membuat program, bagaimana cara Anda mengatasi error waktu compiling/running?
  - Identifikasi masalah dalam error amat terbantu jika error message-nya jelas
- ♦ Eksepsi (exception)
  - Mekanisme yang digunakan oleh bhs pemrograman untuk mendeskripsikan apa yang harus dilakukan jika sesuatu yang tidak diharapkan terjadi (misal error karena berbagai hal).

# Eksepsi

- ◊ Java menyediakan 2 kategori eksepsi:
  - Checked exception
  - Unchecked exception
- ◊ Checked exception:
  - Eksepsi yang diharapkan diantisipasi terjadinya oleh programmer. Biasanya terjadi karena faktor eksternal. Contoh:
    - ◊ File yang akan dibuka tidak ditemukan
    - ◊ Koneksi jaringan gagal
- ◊ Unchecked exception:
  - Eksepsi yang terjadi dari suatu kondisi yang disebabkan oleh adanya bugs (disebut *runtime exceptions*), atau situasi yang cukup rumit untuk ditangani oleh program.
  - Disebut unchecked karena kita tidak diharapkan untuk mengeceknya sebelum hal itu terjadi. Contoh:
    - ◊ Usaha untuk mengakses data di luar suatu indeks array
    - ◊ Error karena kehabisan memori



# Exception Categories



# Eksepsi

- ❖ Class Exception dalam Java mendefinisikan error ringan yang dihadapi oleh program.
- ❖ Class Error dalam Java mendefinisikan error serius yang dihadapi program.
- ❖ Class RuntimeException digunakan untuk menangani eksepsi yang muncul sebagai akibat adanya bugs

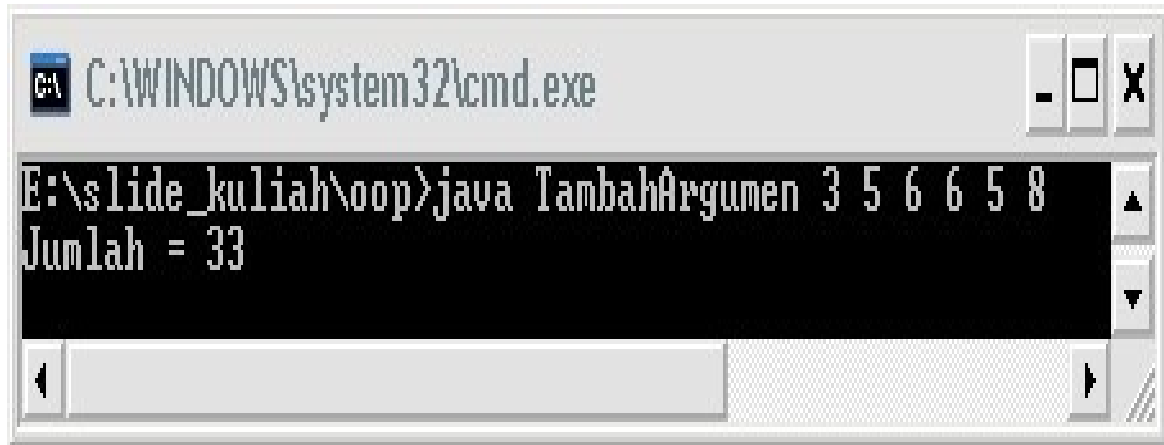
# Contoh Exception

```
1 public class HelloWorld {
2     public static void main (String[] args) {
3         int i = 0;
4
5         String greetings [] = {
6             "Hello world!",
7             "No, I mean it!",
8             "HELLO WORLD!!"
9         };
10
11         while (i < 4) {
12             System.out.println (greetings[i]);
13             i++;
14         }
15     }
16 }
```



# Contoh Exception

```
public class TambahArgumen {  
    public static void main (String[] args) {  
        int jumlah = 0;  
        for (int i = 0; i < args.length; i++) {  
            jumlah += Integer.parseInt(args[i]);  
        }  
        System.out.println("Jumlah = " +  
jumlah);  
    }  
}
```

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The command prompt shows the command 'E:\slide\_kuliah\oop>java TambahArgumen 3 5 6 6 5 8' and the output 'Jumlah = 33'. The window has standard Windows window controls (minimize, maximize, close) in the top right corner and a scrollbar at the bottom.

```
C:\WINDOWS\system32\cmd.exe  
E:\slide_kuliah\oop>java TambahArgumen 3 5 6 6 5 8  
Jumlah = 33
```

# Statemen try dan catch

- Java menyediakan mekanisme untuk menentukan eksepsi mana yang terjadi dan bagaimana mengatasinya. Program berikut merupakan modifikasi program sebelumnya dengan menggunakan statemen try-catch

```
public class TambahArgumen2 {  
    public static void main (String[] args) {  
        try {  
            int jumlah = 0;  
            for (int i = 0; i < args.length; i++) {  
                jumlah += Integer.parseInt(args[i]);  
            }  
            System.out.println("Jumlah = " + jumlah);  
        } catch (NumberFormatException nfe) {  
            System.err.println("Salah satu argumen bukan integer");  
        }  
    }  
}
```



```
C:\WINDOWS\system32\cmd.exe  
E:\slide_kuliah\oop>java TambahArgumen2 3 5 6 6.5 8  
Salah satu argumen bukan integer
```



# Statemen try dan catch

- ◇ Bagaimana jika program dimodifikasi agar dapat menampilkan pesan bahwa argumen tertentu bukan integer, namun tetap menjumlahkan argument-argumen yang bertipe integer dan menampilkan hasilnya.
- ◇ Contoh :

Java TambahArgumen3 1 3.4 5 7.2

Akan menghasilkan output

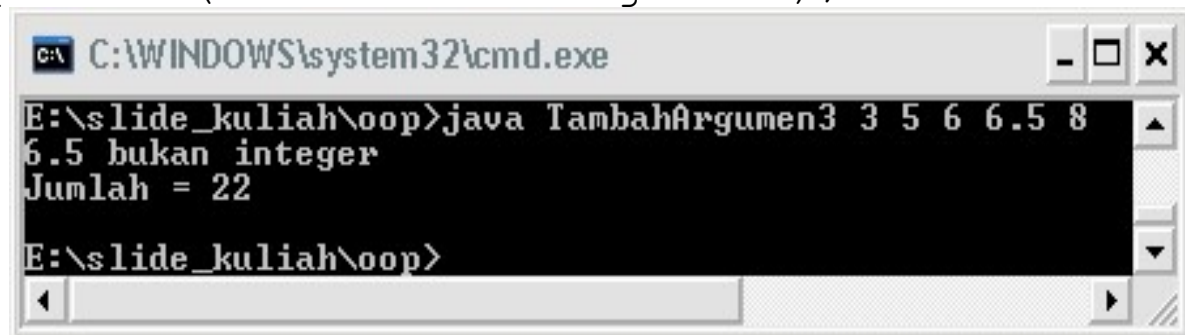
3.4 bukan integer

7.2 bukan integer

Jumlah = 6

# Statemen try dan catch

```
public class TambahArgumen3 {  
    public static void main (String[] args) {  
        int jumlah = 0;  
        for (int i = 0; i < args.length; i++) {  
            try {  
                jumlah += Integer.parseInt(args[i]);  
            } catch (NumberFormatException nfe) {  
                System.err.println(args[i]+" bukan integer");  
            }  
        }  
        System.out.println("Jumlah = " + jumlah);  
    }  
}
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The command prompt displays the following text:

```
E:\slide_kuliah\oop>java TambahArgumen3 3 5 6 6.5 8  
6.5 bukan integer  
Jumlah = 22  
  
E:\slide_kuliah\oop>
```

# Eksepsi buatan sendiri

- ◊ Kita dapat membuat eksepsi sendiri yang merupakan turunan dari class `Exception` dalam Java. Contoh:

```
public class OverdraftException extends Exception {  
    private final double deficit;  
    public OverdraftException(String msg, double  
deficit) {  
        super(msg);  
        this.deficit = deficit;  
    }  
    public double getDeficit() {  
        return deficit;  
    }  
}
```

- ◊ Jelaskan apa makna eksepsi di atas !

# Eksepsi buatan sendiri

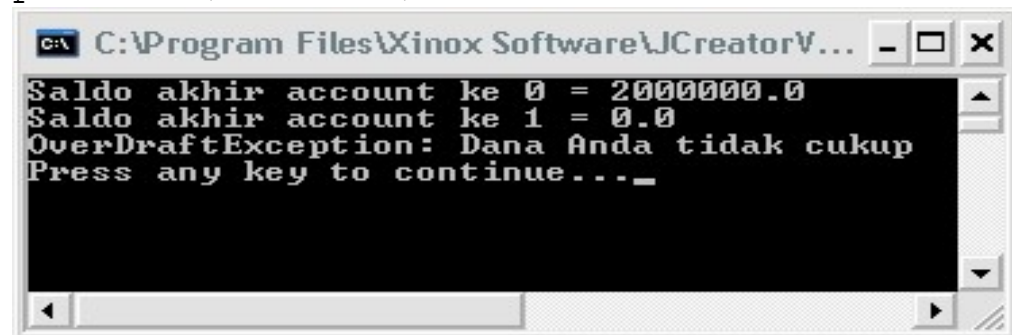
- ♦ Selanjutnya, class tersebut di atas dapat dimanfaatkan oleh method withdraw dalam class Account sbb:

```
public class Account {  
    protected double balance ;  
    protected Account(double initBalance) {  
        balance = initBalance;  
    }  
    public double getBalance() {  
        return balance;  
    }  
    public void deposit(double amt){  
        balance = balance +amt;  
    }  
    public void withdraw(double amt) throws OverdraftException{  
        if (amt<=balance) {  
            balance = balance - amt;  
        } else {  
            throw new OverdraftException("Dana Anda tidak cukup",  
amt-balance);  
        }  
    }  
}
```

# Eksepsi buatan sendiri

- ♦ Class main untuk mengeksekusinya adalah sbb:

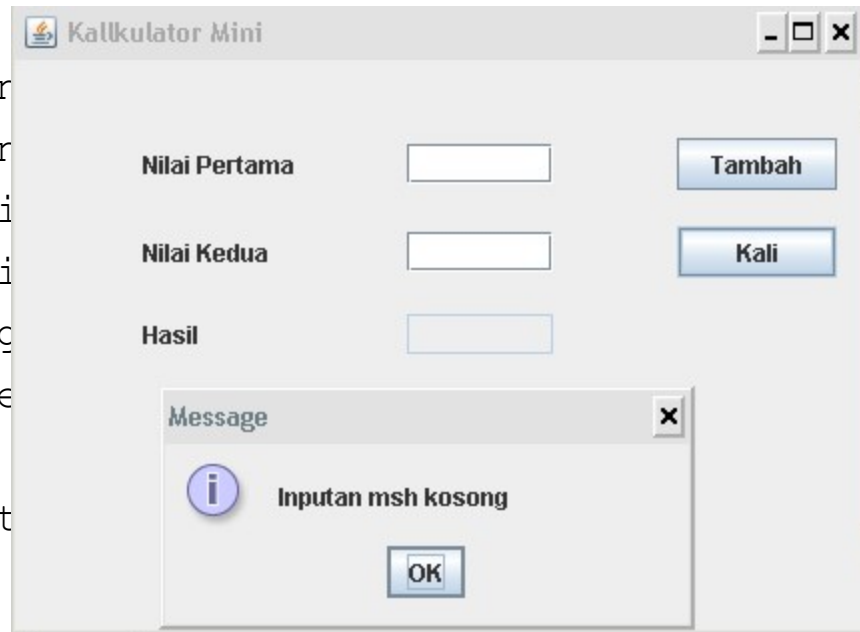
```
class tesBank2 {  
    public static void main (String[] args) {  
        int i;  
        Account [] accounts = new Account[100];  
        accounts[0] = new Account(2000000);  
        accounts[1] = new Account(0);  
        for (i = 0; i<2; i++) {  
            double b = accounts[i].getBalance();  
            System.out.println("Saldo akhir account ke " + i + " =  
" +b);  
            try {  
                accounts[i].withdraw(1000000);  
            } catch (OverdraftException ode) {  
                System.err.println("" +ode);  
            }  
        }  
    }  
}
```



```
C:\Program Files\Xinox Software\JCreatorV...  
Saldo akhir account ke 0 = 2000000.0  
Saldo akhir account ke 1 = 0.0  
OverDraftException: Dana Anda tidak cukup  
Press any key to continue...
```

# Contoh Try & Catch

```
private void btnTambahMouseClicked(java.awt.event.MouseEvent evt)
{
    try{
        int nil1=Ir
        int nil2=Ir
        int nilA=ni
        String hasi
        hasil=Integ
        nilHasil.se
    }
    catch (Except
    {
        //System.out.println("Data smlr diinput...");
        JOptionPane.showMessageDialog(null, e);
    }
}
```



```
public class TestException
{
    public static void bagi(int a,int b) throws Exception
    {
        if(b==0)
        {
            throw new Exception("penyebut tidak boleh 0");
        }else
        {
            int hasil=a/b;
            System.out.println(hasil);
        }
    }
    public static void main(String[] a)
    {
        try{
            bagi(10,0);
        }catch(Exception e )
        {
            System.out.println(e);
        }
    }
}
```

```
D:\JavaMaga lang>javac TestException.java
D:\JavaMaga lang>java TestException
java.lang.Exception: penyebut tidak boleh 0
D:\JavaMaga lang>
```

```
public class TestError
{
    public static void main(String[] a)
    {
        try{
            int[] angka={10,20};
            System.out.println(angka[1]);
            int hasil=10/5;
            System.out.println(hasil);
        }catch(ArithmeticException e1)
        { System.out.println("ada error pembagian: "+e1);
        }
        catch(ArrayIndexOutOfBoundsException e2)
        { System.out.println("ada error array : "+e2);
        }
        catch(Exception e)
        { System.out.println("ada error : "+e);
        }
        finally
        { System.out.println("blok yg selalu jalan,
            entah ada exception atau tidak");
        }
    }
}
```

```
D:\JavaMagelang>javac TestError.java
D:\JavaMagelang>java TestError
20
2
blok yg selalu jalan, entah ada exception atau tidak
D:\JavaMagelang>
```



# Rehat Sejenak

- WOW