



PEMOGRAMAN JAVA

Yoannita

- **Tipe Data (ii) : Array**
- **Exception Handling**

"Education is when you read the fine print. Experience is what you get if you don't." (Pete Seeger)



Array

- Misalkan dibutuhkan 100 variabel bertipe int untuk suatu perhitungan
- Array satu dimensi merupakan kumpulan dari variabel yang memiliki tipe data sama
- Pendeklarasian :
 - Tipe-data nama-array[];
 - Tipe-data[] nama-array;
- Contoh :
 - `int arr1[];` // atau :
 - `int[] arr2;`



Array

```
int x [] = new int [3] ;
```

x[0]	x[1]	x[2]
------	------	------

```
x[0] = 20;
```

```
x[1] = 10;
```

```
x[2] = 0;
```

20	10	0
----	----	---

```
Nama-array = new tipe=data[ukuran-array]
```

```
int[] arr;
```

```
arr = new int [10];
```



Array

```
class singleArray {  
    public static void main (String[] args) {  
        int[] x = new int[3];  
        x[0] = 20;  
        x[1] = 10;  
        x[2] = 0;  
        System.out.println ("Nilai x[0] : " + x[0]);  
        System.out.println ("Nilai x[1] : " + x[1]);  
        System.out.println ("Nilai x[2] : " + x[2]);  
    }  
}
```




Array: Error(i)

- Error yang menyebutkan **Java.lang.ArrayIndexOutOfBoundsException**

```
int [] x;
```

```
x = new int[3];
```

```
x = [10] = 100; -----> ERROR!
```

- **Mengapa?**



Array: Error(ii)

- Deklarasi array baru :

BENAR → `int[] x = new int [3];` atau
`int[] x = {0,1,2}`

ERROR -----> { `int [] x = new int [3];`
`x = {0,1,2}`



Panjang Dimensi Array

...

```
int x = {0,1,2}
```

```
System.out.print (x.length);
```

...

```
// akan mencetak angka 3
```

```
String y = "hallo";
```

```
System.out.println(y.length());
```




Exception Handling

Penanganan eksepsi diperlukan untuk alasan berikut :

1. User umumnya tidak mengerti dan tidak peduli bagaimana cara suatu program bekerja atau dieksekusi sehingga menampilkan pesan kesalahan yang tidak dimengerti. Anda mungkin ingin mengganti pesan kesalahan dengan pesan yang lebih mudah dimengerti.

Contohnya apabila user mengetikkan huruf untuk harga barang, maka akan muncul pesan error berikut: *...java.lang.NumberFormatException: For input string:...*, dan anda dapat menggantinya dengan *"harga barang harus berupa angka"*




Exception Handling (ii)

2. Anda ingin program memperbaiki kesalahan akibat eksepsi dan terus melanjutkan eksekusi seolah-olah tidak pernah terjadi eksepsi.
3. Anda ingin program terus berjalan dan bukannya dihentikan pada saat eksepsi terjadi.



Try... Catch... Finally...

- Exception dilemparkan selama eksekusi dari blok *try* dapat ditangkap dan ditangani dalam blok *catch*.
- Kode dalam blok *finally* selalu dieksekusi.



Try... Catch... Finally...

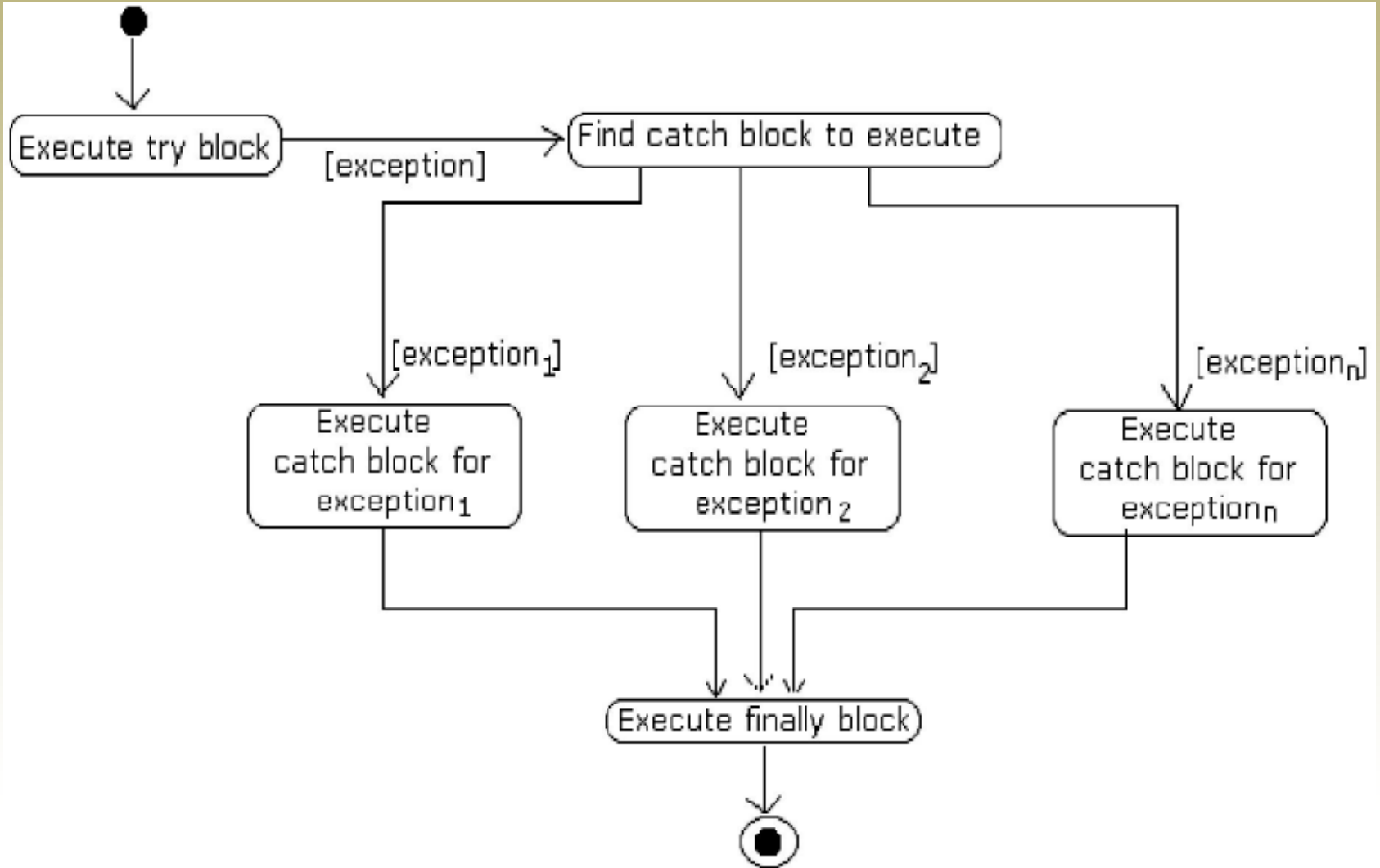
- Aspek kunci tentang sintak dari konstruksi try-catch-finally:
 - Notasi blok bersifat perintah
 - Setiap blok *try*, terdapat satu atau lebih blok *catch*, tetapi hanya satu blok *finally*.
 - Blok *catch* dan blok *finally* harus selalu muncul dalam konjungsi dengan blok *try*, dan diatas urutan
 - Blok *try* harus diikuti oleh **paling sedikit** satu blok *catch* ATAU satu blok *finally*, atau keduanya.
 - Setiap blok *catch* mendefinisikan sebuah penanganan exception. Header dari blok *catch* harus membawa satu argumen, dimana exception pada blok tersebut akan ditangani.
 - Exception harus menjadi class pelempar atau satu dari subclassesnya.



Try... Catch... Finally...

```
try {  
    //tulis pernyataan yang dapat mengakibatkan exception  
    //dalam blok ini  
}  
catch( <exceptionType1> <varName1> ){  
    //tulis aksi apa dari program Anda yang dijalankan jika ada  
    //exception tipe tertentu terjadi  
}  
...  
catch( <exceptionTypen> <varNamen> ){  
    //tulis aksi apa dari program Anda yang dijalankan jika ada  
    //exception tipe tertentu terjadi  
}  
finally{  
    //tambahkan kode terakhir di sini  
}
```

Try... Catch... Finally...



Gambar 1: Alur kejadian blok try-catch-finally



Exception Handling : Contoh

- Contoh : terdapat kode yang mencetak argumen kedua ketika menjalankan kode menggunakan argumen command-line.
- Perkirakan, tidak ada pengecekan dalam kode Anda untuk angka dari argumen dan hanya mengakses argumen kedua `args[1]`, maka akan didapatkan exception berikut :
 - Exception in thread "main"
 - `java.lang.ArrayIndexOutOfBoundsException: 1`
 - `at ExceptionExample.main(ExceptionExample.java:5)`



Exception Example(i)

```
public class ExceptionExample
{
    public static void main( String[] args ){

        try {
            System.out.println( args[1] );

            // mencetak argumen command line

        } catch( ArrayIndexOutOfBoundsException exp ) {

            System.out.println("Exception caught!");

            /* kesalahan akan dicetak apabila tidak ada argumen command line
               kedua yang diberikan */

        }

    }
}
```


Exception Example(ii)

```
class DemoEksepsi {
    public static void main(String[] args){
        try {
            int[] arr = new int[1];
            System.out.println(arr[1]); /* ingat bahwa index array dimulai dari nol
            shg baris ini akan memunculkan pesan kesalahan karena arr yg
            didefinisikan hanya memuat satu index(arr[0]) */
            System.out.println("baris ini tidak akan pernah dieksekusi");
        } catch (ArrayIndexOutOfBoundsException e)
        {
            System.out.println("terjadi eksepsi karena index diluar kapasitas
            array");
            System.out.println("Kesalahan yang ditangkap :" + e);
        } finally{ /* finally akan selalu dicetak */
            System.out.println("percobaan selesai");
        }

        System.out.println("Good bye");
    }
}
```



Exception Example(iii)

- `import java.io.*;`

```
public class IfElse{
    public static void main(String[] args) throws IOException{
        try{
            int n;
            BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
            n = Integer.parseInt(in.readLine());
            if (n % 2 == 0)
            {
                System.out.println("Given number is Even.");
            }
            else
            {
                System.out.println("Given number is Odd.");
            }
        }
        catch(NumberFormatException e){
            System.out.println(e.getMessage() + " is not a numeric value.");
            System.exit(0);
        }
    }
}
```

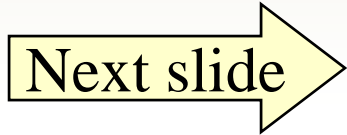
Latihan (i)

- Diberikan kode berikut:

```
public class TestExceptions{
    public static void main( String[] args ){
        for( int i=0; true; i++ ){
            System.out.println("args["+i+"]="+
                args[i]);
        }
    }
}
```

- Compile dan jalankan program TestExceptions. Keluarannya akan tampak seperti ini:

```
javac TestExceptions one two three
args[0]=one
args[1]=two
args[2]=three
Exception in thread "main"
java.lang.ArrayIndexOutOfBoundsException: 3
at TestExceptions.main(1.java:4)
```



Next slide



Latihan (i)

- Ubah program TestExceptions untuk menangani exception, keluaran program setelah ditangkap exception-nya akan seperti ini:

```
javac TestExceptions one two three
```

```
args[0]=one
```

```
args[1]=two
```

```
args[2]=three
```

```
Exception caught:
```

```
java.lang.ArrayIndexOutOfBoundsException: 3
```

```
Quiting...
```