	<b>FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA</b>			
	<b>LAB SHEET PEMROGRAMAN 2</b>			
	Semester 2	EXCEPTIONS AND ERRORS		4 x 50 mnt
	No. LST/EKA/PTI208/13	Revisi : 02	Mei 2010	Hal 1 dari 5

## A. Kompetensi

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu menangani exception dan error dalam pemrograman java.

## B. Dasar Teori

Ada 3 kategori error

### 1. Syntax error

Terjadi karena ada aturan bahasa pemrograman yang tidak ditaati, error ini akan di deteksi oleh kompiler. Error jenis ini mudah untuk dicari dan diperbaiki karena kompiler akan memberitahukan dimana error terjadi dan mengapa error bisa terjadi.

### 2. Logic error

Terjadi ketika program bekerja tidak sesuai dengan tujuan dari pembuatan program itu. Error jenis ini ditangani dengan *debugging*.

### 3. Runtime error

Terjadi ketika program dijalankan. Misal user memasukkan input yang salah, program mengakses elemen array diluar indexnya, dan kesalahan external lain.

Exception adalah runtime error. Jika sebuah program tidak menyediakan kode untuk menangani exception maka program akan berhenti dengan tidak lazim, dan bisa mengakibatkan kesalahan fatal.

Misal sebuah program untuk transfer uang antar rekening, jika terjadi error ketika uang sudah dikurangi dari rekening asal tetapi uang tersebut belum ditambahkan ke rekening tujuan, maka nasabah akan kehilangan uangnya.

Exception dalam java adalah instance dari kelas turunan Throwable. Kelas Throwable memiliki dua kelas turunan yaitu Exception dan Error.

Error yang diberikan oleh JVM yang direpresentasikan dalam kelas Error tidak bisa di-*handle*. Error ini terjadi karena kesalahan sistem internal dan jarang terjadi.

Exception direpresentasikan dalam kelas Exception terjadi karena kesalahan program dan kondisi external lain. Exception bisa di-*handle* di dalam program.

## Try Catch

Bentuk exception handling dalam java

```
try {
// blok kode yang dimonitor errornya
}
catch (ExceptionType1 exOb) {
// penanganan exception untuk ExceptionType1
}
catch (ExceptionType2 exOb) {
// penanganan exception untuk ExceptionType2
```

Dibuat oleh : Herman DS	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------------	---	------------------



FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA

LAB SHEET PEMROGRAMAN 2

Semester 2	EXCEPTIONS AND ERRORS	4 x 50 mnt
No. LST/EKA/PTI208/13	Revisi : 02	Mei 2010
		Hal 2 dari 5

```
}  
// ...
```

Kode yang dimonitor errornya dimasukkan dalam blok *try* jika terjadi exception maka blok *try* akan melempar (throw) exception objek. Jika exception objek memiliki tipe yang sama dengan parameter blok *catch* maka kode di dalam blok *catch* akan dijalankan. Kita bisa membuat lebih dari satu blok *catch* untuk tipe exception yg berbeda-beda

**Contoh exception handling**

```
public class TestException {  
    public static void main(String[] args) {  
        try {  
            System.out.println(sum(new int[] {1, 2, 3, 4, 5}));  
        }  
        catch (Exception ex) {  
            ex.printStackTrace();  
            System.out.println("\n" + ex.getMessage());  
            System.out.println("\n" + ex.toString());  
  
            System.out.println("\nTrace Info Obtained from getStackTrace");  
            StackTraceElement[] traceElements = ex.getStackTrace();  
            for (int i = 0; i < traceElements.length; i++) {  
                System.out.print("method " + traceElements[i].getMethodName());  
                System.out.print("(" + traceElements[i].getClassName() + "):");  
                System.out.println(traceElements[i].getLineNumber() + " ");  
            }  
        }  
    }  
  
    private static int sum(int[] list) {  
        int result = 0;  
        for (int i = 0; i <= list.length; i++)  
            result += list[i];  
        return result;  
    }  
}
```

**Finally**

Ketika sebuah exception dilempar/terjadi maka sebuah method akan berjalan tidak sebagai mana mestinya. Tergantung bagaimana method itu dibuat, sebuah method dapat berhenti secara prematur. Misal ada sebuah method yang membuka dan memodifikasi sebuah file kemudian menutup file tersebut. Kita tentu tidak menginginkan langkah menutup file di bypass oleh *exception handling*. Disinilah manfaatnya blok *finally*.

Kode yang ada di dalam blok *finally* akan dieksekusi setelah blok *try-catch* dijalankan.

**Contoh blok finally**

```
public class FinallyDemo {  
    public static void main(String[] args) {  
        java.io.PrintWriter output = null;
```

Dibuat oleh :  
Herman DS

Dilarang memperbanyak sebagian atau seluruh isi dokumen  
tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta

Diperiksa oleh :



```
try {
    // Create a file
    output = new java.io.PrintWriter("text.txt");

    // Write formatted output to the file
    output.println("Welcome to Java");
}
catch (java.io.IOException ex) {
    ex.printStackTrace();
}
finally {
    // Close the file
    if (output != null) output.close();
}
}
```

### Throws

Jika sebuah method dapat menyebabkan sebuah exception yang tidak bisa dihandle oleh method itu sendiri, maka method tersebut harus mendefinisikan sifat ini agar method yang memanggilmnya dapat melindungi diri dari exception yang terjadi. Kita bisa melakukan ini dengan menggunakan kata kunci *throws* pada deklarasi method. Kata kunci *throws* akan diikuti daftar tipe exception yang mungkin dilempar oleh sebuah method. Kata kunci ini perlu dipakai untuk semua exception kecuali exception dari kelas *Exception* dan *RuntimeException* termasuk turunannya.


Bentuk penggunaan kata kunci *throws*

```
type method-name(parameter-list) throws exception-list
{
    // body of method
}
```

Disini exception-list bisa lebih dari satu tipe exception yang dipisahkan dengan tanda koma.

Tipe exception yang membutuhkan kata kunci *throws* disebut dengan *checked exception*. Sedangkan exception yg tidak membutuhkan kata kunci *throws* disebut *unchecked exception*

Exception	Maksudnya
ArithmeticException	Arithmetic error, such as divide-by-zero.
ArrayIndexOutOfBoundsException	Array index is out-of-bounds.
ArrayStoreException	Assignment to an array element of an incompatible type.
ClassCastException	Invalid cast.
IllegalArgumentException	Illegal argument used to invoke a method.
IllegalMonitorStateException.	Illegal monitor operation, such as waiting on an

	<b>FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LAB SHEET PEMROGRAMAN 2</b>		
	Semester 2	EXCEPTIONS AND ERRORS	4 x 50 mnt
	No. LST/EKA/PTI208/13	Revisi : 02	Mei 2010
Hal 4 dari 5			

	unlocked thread
IllegalStateException	Environment or application is in incorrect state.
IllegalThreadStateException	Requested operation not compatible with current thread state.
IndexOutOfBoundsException	Some type of index is out-of-bounds.
NegativeArraySizeException	Array created with a negative size
NullPointerException	Invalid use of a null reference.
NumberFormatException	Invalid conversion of a string to a numeric format.
SecurityException	Attempt to violate security.
StringIndexOutOfBoundsException	Attempt to index outside the bounds of a string.
UnsupportedOperationException	An unsupported operation was encountered.

Tabel 1. Unchecked Exception

Exception	Maksudnya
ClassNotFoundException	Class not found.
CloneNotSupportedException	Attempt to clone an object that does not implement the <b>Cloneable</b> interface.
IllegalAccessException	Access to a class is denied.
InstantiationException	Attempt to create an object of an abstract class or interface.
InterruptedException	One thread has been interrupted by another thread.
NoSuchFieldException	A requested field does not exist.
NoSuchMethodException	A requested method does not exist.

Tabel 2. Checked Exception

### C. Alat/ Bahan

1. Lab. Sheet Pemrograman Java no 13
2. PC / Laptop with OS installed
3. JDK 1.5 or latest
4. J-Creator or text editor

### D. Langkah Kerja


1. Baca dan pahami dasar teori di atas.
2. Lakukan kompilasi dan eksekusi terhadap contoh-contoh source code atau program yang ada di dasar teori dan LAMPIRAN.
3. Kerjakan tugas individu di bawah.

### E. Tugas Individu

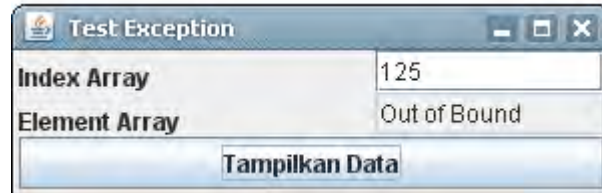
Buatlah aplikasi Java GUI yang memenuhi kriteria sebagai berikut.

- ✓ Buatlah array yang berisi 100 bilangan integer acak.

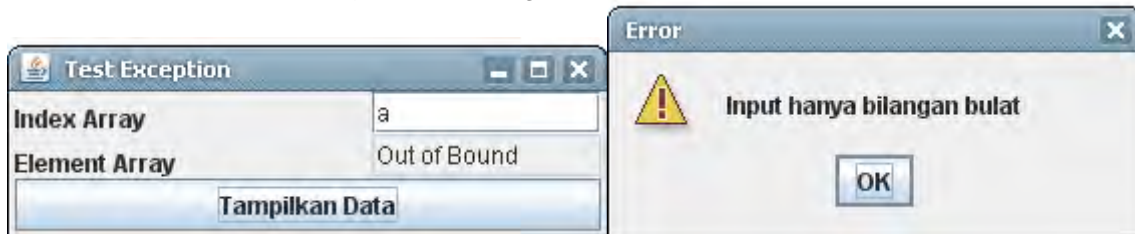
Dibuat oleh : Herman DS	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------------	---	------------------

	<b>FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LAB SHEET PEMROGRAMAN 2</b>		
	Semester 2	EXCEPTIONS AND ERRORS	4 x 50 mnt
	No. LST/EKA/PTI208/13	Revisi : 02	Mei 2010
			Hal 5 dari 5

- ✓ Buatlah sebuah text field untuk memasukkan index dari array tersebut, dan sebuah text field lain untuk menampilkan elemen pada index yang telah ditentukan.
- ✓ Buat tombol **tampilkan data** untuk menampilkan elemen array pada index yang telah ditentukan. Jika index yang ditentukan *out of bound* maka tampilkan "Out of bound" (lihat gambar)



- ✓ Jika input tidak sah (misal input bilangan desimal atau angka atau tanda baca) maka tampilkan jendela peringatan (lihat gambar)



## F. Lampiran

- **CH 80** (Introduction to Computer Science using Java, Java 5.0 version, January 2006, Bradley Kjell, Central Connecticut State University <http://chortle.ccsu.edu/CS151/cs151java.html>)
- **Java™ Swing, 2nd Edition**, Brian Cole, Robert Eckstein, James Elliott, Marc Loy, David Wood

Dibuat oleh : Herman DS	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------------	--	------------------