

**PRAKTIKUM PBO LANJUT  
PENANGANAN EXCEPTION**

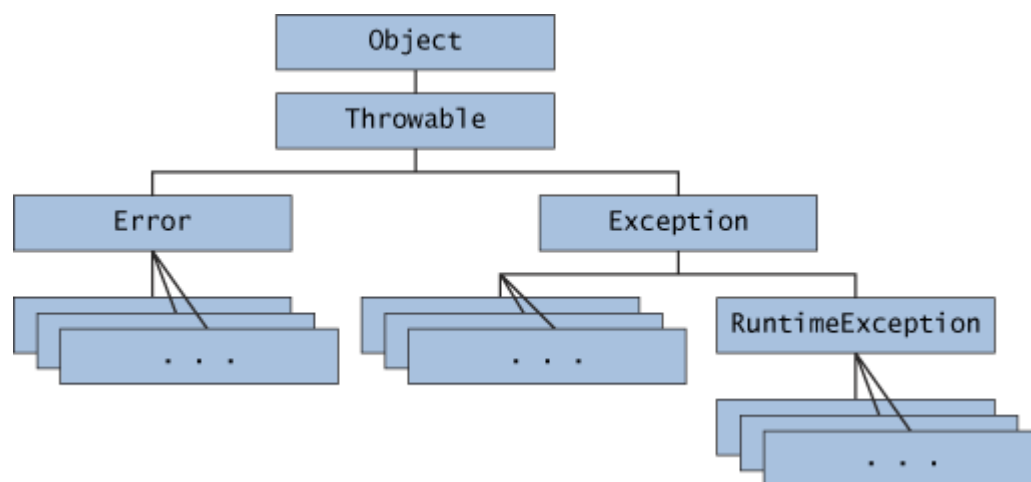
**A. TUJUAN PRAKTIKUM**

Mahasiswa mampu memahami konsep penanganan suatu exception.

**B. EXCEPTION**

Exception merupakan suatu event/kejadian saat eksekusi program dimana hasil eksekusi program tidak sesuai dengan skenario normal. Misalnya, ada program untuk membuka suatu file namun file tersebut ternyata tidak ada.

Diagram kelas untuk exception dapat diperhatikan di bawah ini. Dari diagram kelas di bawah, semua exception dianggap sebagai suatu kelas. Kelas Throwable mengharuskan semua kelas turunannya diimplementasikan sebagai exception.



Mekanisme penanganan exception adalah dengan menggunakan blok:

```
try{
    ..... // beberapa pernyataan
} catch (IOException var){
    ..... // pernyataan untuk menangani IO error
} catch (ArithmeticException var){
    ..... // pernyataan untuk menangani arithmaetic error
}
finally {
    ..... // pernyataan yang harus dikerjakan baik ada error maupun tidak ada
error
}
```

Suatu kelas dapat melemparkan suatu exception jika terjadi suatu kesalahan melalui methodnya.

```

.....
.....
public void aMethod( ) throws Exception{
    if (right)
        .....// do an action
    else
        throw new Exception("error");
}
.....
.....

```

Keuntungan dari penggunaan exception adalah memudahkan penanganan error dibanding dengan menggunakan penanganan kesalahan secara manual.

### C. PRAKTIKUM

1. Buatlah kelas LatihanException. Buatlah antar muka seperti gambar dibawah. Kemudian tambahkan potongan program di bawah untuk event ketika tombol “OK” di tekan dan tombol “Cancel” di tekan.



```

.....
cancelButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {
        System.exit(0);
    }
});
okButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {

        int age = Integer.parseInt(inTextF.getText());
        outLabel.setText("Umur anda adalah"+inTextF.getText());
        JOptionPane.showMessageDialog(null,"Terima kasih anda
telah
        menggunakan program ini");
    }
});
.....

```

Kemudian buat kelas testing dan tambahkan potongan program dibawah. Apakah yang terjadi jika anda memasukkan :

- Angka
- Huruf
- Angka negative

**Jelaskan apa yang terjadi untuk setiap masukan !**

```
public class Main {

    public static void main(String[] args) {
        InputAge2 frame = new InputAge2();
        frame.setVisible(true);
    }
}
```

2. Perbaiki program pada event untuk tombol “OK” di atas sesuai bawah ini. Compile dan jalankan untuk 3 macam masukan diatas ! Amati dan jelaskan untuk 3 masukan !

```
.....
okButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {

        try {
            int age = Integer.parseInt(inTextF.getText());
            outLabel.setText("Umur anda adalah "+inTextF.getText());
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, ""+
                inTextF.getText()+" is invalid\n"
                +"Please enter digits only");
            inTextF.setText("");
        }
        JOptionPane.showMessageDialog(null, "Terima kasih anda telah
            menggunakan program ini");
    }
});
.....
```

3. Modifikasi program event untuk tombol “OK” seperti di bawah ini! Tambahkan metode setAge dibawah metode yang mengandung event listener. Kelas ini dilengkapi dengan Throw. Jelaskan perbedaan dengan program sebelumnya untuk 3 macam masukan seperti nomor diatas.

```
.....
okButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {
        try {
            int age = Integer.parseInt(inTextF.getText());
            setAge(age);
            outLabel.setText("Umur anda adalah "+inTextF.getText());
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, ""+
                inTextF.getText()+" is invalid\n"
                +"Please enter digits only");
            inTextF.setText("");
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, ex.getMessage());
            inTextF.setText("");
        }
        JOptionPane.showMessageDialog(null, "Terima kasih anda telah
            menggunakan program ini");
    }
});
.....
```

```

public void setAge(int age) throws Exception
{
    if (age > 0)
        outLabel.setText(String.valueOf(age));
    else {
        throw new Exception("Input must be greater than 0");
    }
}

```

4. Modifikasi metode setAge() seperti di bawah ini! Tambahkan class NegativeValueException. Jelaskan perbedaan dengan program sebelumnya.

```

.....
public void setAge(int age) throws Exception
{
    if (age > 0)
        outLabel.setText(String.valueOf(age));
    else {
        throw new NegativeValueException("Input must be greater than
0");
    }
}
.....

class NegativeValueException extends Exception {
    public NegativeValueException() {
    }
    public NegativeValueException(String message) {
        super(message);
    }
}

```

## D. TUGAS

1. Berdasar program perpustakaan yang pernah anda buat pada modul sebelumnya. tambahkan pada event saat menyimpan data pada form GUI “Tambah Koleksi” yang berisi perintah :
  - a. Perbaiki pembuatan obyek Koleksi menggunakan method **set**.
  - b. Pada saat **set** data tersebut, cek apakah data volume, seri dan halaman berupa nilai int menggunakan exception.

## E. LAPORAN

1. Penjelasan untuk 4 soal praktikum (@15%)
2. Kode program, analisa setiap potongan program dan capture GUI untuk tugas (40%, @20)

*@@@ Do better than your best !! @@@*