

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
Modul 5
Latihan Exception Handling



Oleh :
Riska Vany Dewi
2311103111
S1SI-07-C

PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
UNIVERSITAS TELKOM PURWOKERTO

2024



I. GUIDED

Kode:

Buku.java

```
package Perpustakaan;

/**
 * @author Riska Vany Dewi
 * 2311103111
 * S1SI07C
 */
public class Buku {
    private String judul;
    private String pengarang;
    private String isbn;

    public Buku(String judul, String pengarang, String isbn) throws ValidasiInputException {
        if(judul.isEmpty()){
            throw new ValidasiInputException("Judul kosong");
        }

        if(pengarang.isEmpty()){
            throw new ValidasiInputException("Pengarang kosong");
        }

        if(!isbn.matches("\\d{13}")){
            throw new ValidasiInputException("ISBN harus 13 angka");
        }
    }
}
```

```
this.judul = judul;

this.pengarang = pengarang;
this.isbn = isbn;
}

public String getJudul(){
    return judul;
}

public String getPengarang(){
    return pengarang;
}

public String getIsbn(){
    return isbn;
}
}
```

Penjelasan:

Kelas `Buku` dalam paket `Perpustakaan` adalah representasi dari objek buku yang memiliki tiga atribut: `judul`, `pengarang`, dan `isbn`. Konstruktor kelas ini menerima tiga parameter dan melakukan validasi untuk memastikan bahwa `judul` dan `pengarang` tidak kosong, serta bahwa `isbn` terdiri dari 13 digit angka. Jika salah satu dari validasi ini gagal, sebuah pengecualian `ValidasiInputException` akan dilemparkan dengan pesan yang sesuai. Kelas ini juga menyediakan metode getter untuk masing-masing atribut, yaitu `getJudul()`, `getPengarang()`, dan `getIsbn()`, yang memungkinkan pengambilan nilai dari atribut-atribut tersebut setelah objek `Buku` dibuat.

ValidasiInputException.java

```
package Perpustakaan;

/**
 * @author Riska Vany Dewi
 */
```

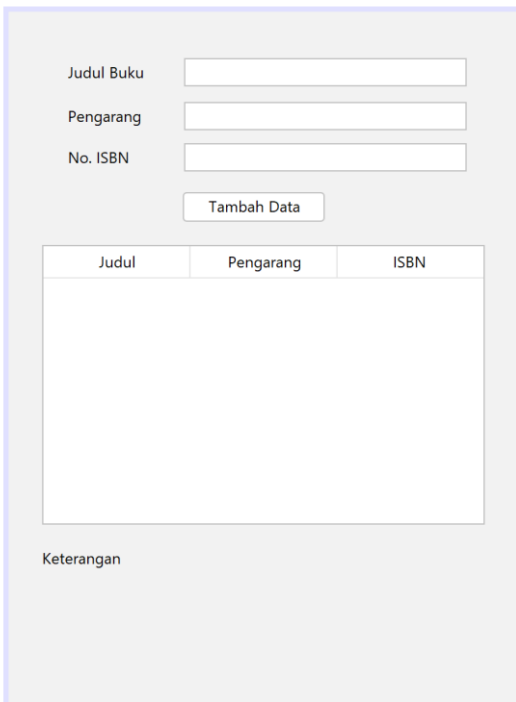


```
* 2311103111
* S1SI07C
*/
public class ValidasiInputException extends Exception {
    public ValidasiInputException(String message){
        super(message);
    }
}
```

Penjelasan:

Kelas `ValidasiInputException` dalam paket `Perpustakaan` adalah subclass dari `Exception` yang digunakan untuk menangani kesalahan validasi input dalam program. Kelas ini memiliki konstruktor yang menerima sebuah parameter `message`, yang merupakan pesan kesalahan yang akan ditampilkan kepada pengguna. Pesan ini kemudian diteruskan ke konstruktor superclass (`Exception`) menggunakan `super(message)`, sehingga saat pengecualian ini dilemparkan, pesan yang sesuai dapat diakses dan ditampilkan. Dengan adanya kelas ini, program dapat memberikan umpan balik yang jelas dan spesifik ketika data yang dimasukkan tidak memenuhi syarat yang ditentukan.

InputData.java



Judul	Pengarang	ISBN
-------	-----------	------

Kode tambahan di source tombolActionPerformed InputData.java :

```
private void tombolActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try{  
        String judul = txtJudul.getText();  
        String pengarang = txtPengarang.getText();  
        String isbn = txtIsbn.getText();  
  
        Buku buku = new Buku(judul, pengarang, isbn);  
  
        DefaultTableModel data = (DefaultTableModel)tableBuku.getModel();  
  
        data.addRow(new Object[] {buku.getJudul(), buku.getPengarang(), buku.getIsbn()});  
  
        txtJudul.setText("");  
        txtPengarang.setText("");  
        txtIsbn.setText("");  
        lblMessage.setText("Data berhasil ditambahkan!");  
        lblMessage.setForeground(Color.green);  
    } catch (ValidasiInputException e) {  
        lblMessage.setText("Error:" + e.getMessage());  
        lblMessage.setForeground(Color.red);  
    }  
}
```

Penjelasan:

Metode `tombolActionPerformed` digunakan untuk menangani peristiwa ketika tombol ditekan dalam antarmuka pengguna Java. Di dalam metode ini, program mengambil input dari tiga field teks (judul, pengarang, dan ISBN) dan mencoba membuat objek `Buku` menggunakan nilai-nilai tersebut. Jika validasi berhasil, data buku ditambahkan ke tabel yang



ditampilkan dalam UI. Setelah itu, field input dikosongkan, dan pesan sukses ditampilkan. Jika terjadi kesalahan validasi, pesan kesalahan akan ditampilkan kepada pengguna.

Output:



The screenshot shows a web application window with a light blue background. At the top, there are three input fields labeled "Judul Buku", "Pengarang", and "No. ISBN". Below these fields is a button labeled "Tambah Data". Under the button is a table with three columns: "Judul", "Pengarang", and "ISBN". The table contains one row of data: "Dilan 1990", "Pidi Baiq", and "1234567890987". At the bottom of the window, a green message "Data berhasil ditambahkan!" is displayed.

Judul	Pengarang	ISBN
Dilan 1990	Pidi Baiq	1234567890987

Data berhasil ditambahkan!



The screenshot shows the same web application window as the previous one, but with different data. The "Judul Buku" field is filled with "Burung Terbang di Kelam Malam", the "Pengarang" field is empty, and the "No. ISBN" field is filled with "0987654321234". The "Tambah Data" button is still present. The table below it contains the same data as the previous screenshot. At the bottom of the window, a red error message "Error: Pengarang kosong" is displayed.

Judul	Pengarang	ISBN
Dilan 1990	Pidi Baiq	1234567890987

Error: Pengarang kosong

II. UNGUIDED

Tambahkan inputan berikut pada aplikasi:

1. Tahun Terbit: Validasi untuk angka empat digit.
2. Harga Buku: Validasi untuk angka positif.
3. Kategori Buku: Validasi untuk input tidak kosong. (dalam bentuk Dropdown)

Buku.java

```
package Perpustakaan;

/**
 * @author Riska Vany Dewi
 * 2311103111
 * S1SI07C
 */
public class Buku {
    private String judul;
    private String pengarang;
    private String isbn;
    private String tahunTerbit;
    private double hargaBuku;
    private String kategori;

    public Buku(String judul, String pengarang, String isbn, String tahunTerbit,
        String hargaBuku, String kategori) throws ValidasiInputException {
        if(judul.isEmpty()){
            throw new ValidasiInputException("Judul kosong");
        }

        if(pengarang.isEmpty()){
            throw new ValidasiInputException("Pengarang kosong");
        }
    }
}
```

```
if(!isbn.matches("\\d{13}")){
    throw new ValidasiInputException("ISBN harus 13 angka");
}

if (tahunTerbit.length() != 4 || !tahunTerbit.matches("\\d{4}")) {
    throw new ValidasiInputException("Tahun terbit harus 4 digit");
}

double harga = Double.parseDouble(hargaBuku);
if (harga <= 0) {
    throw new ValidasiInputException("Harga buku harus angka positif");
}

if (kategori == null || kategori.isEmpty()) {
    throw new ValidasiInputException("Kategori buku tidak boleh kosong");
}

this.judul = judul;
this.pengarang = pengarang;
this.isbn = isbn;
this.tahunTerbit = tahunTerbit;
this.hargaBuku = Double.parseDouble(hargaBuku);
this.kategori = kategori;
}

public String getJudul(){
    return judul;
}
```



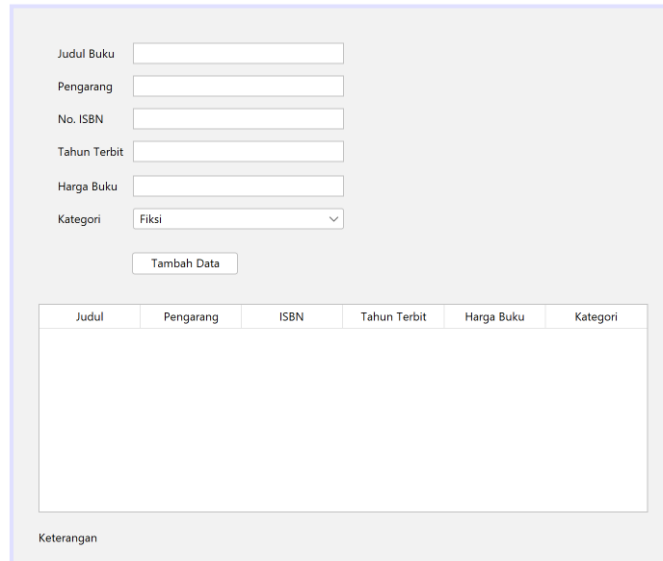
```
public String getPengarang(){  
    return pengarang;  
}  
  
public String getIsbn(){  
    return isbn;  
}  
  
public String getTahunTerbit(){  
    return tahunTerbit;  
}  
  
public double getHargaBuku(){  
    return hargaBuku;  
}  
  
public String getKategori(){  
    return kategori;  
}  
}
```

Penjelasan:

Pada kode kelas `Buku` yang terbaru, terdapat penambahan atribut baru, yaitu `tahunTerbit`, `hargaBuku`, dan `kategori`, yang memungkinkan objek `Buku` menyimpan lebih banyak informasi. Konstruktor juga diperbarui untuk menerima parameter tambahan ini, dan validasi input ditambahkan untuk memastikan bahwa `tahunTerbit` adalah 4 digit, `hargaBuku` adalah angka positif, dan `kategori` tidak boleh kosong. Selain itu, harga buku sekarang dikonversi dari `String` ke `double` dan disimpan dalam atribut `hargaBuku`. Metode getter baru untuk `tahunTerbit`, `hargaBuku`, dan `kategori` juga ditambahkan untuk memungkinkan akses ke nilai-nilai ini setelah objek dibuat.

InputData.java





Kode tambahan di source tombolActionPerformed InputData.java:

```
private void tombolActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String judul = txtJudul.getText();
        String pengarang = txtPengarang.getText();
        String isbn = txtIsbn.getText();

        String tahunTerbit = tahunTerbit1.getText();
        String hargaBuku = hargaBuku1.getText();
        String kategori = (String) kategori1.getSelectedItem();

        Buku buku = new Buku(judul, pengarang, isbn, tahunTerbit, hargaBuku, kategori);

        DefaultTableModel data = (DefaultTableModel)tableBuku.getModel();
        data.addRow(new Object[]{buku.getJudul(), buku.getPengarang(), buku.getIsbn(),
            buku.getTahunTerbit(), buku.getHargaBuku(), buku.getKategori()});

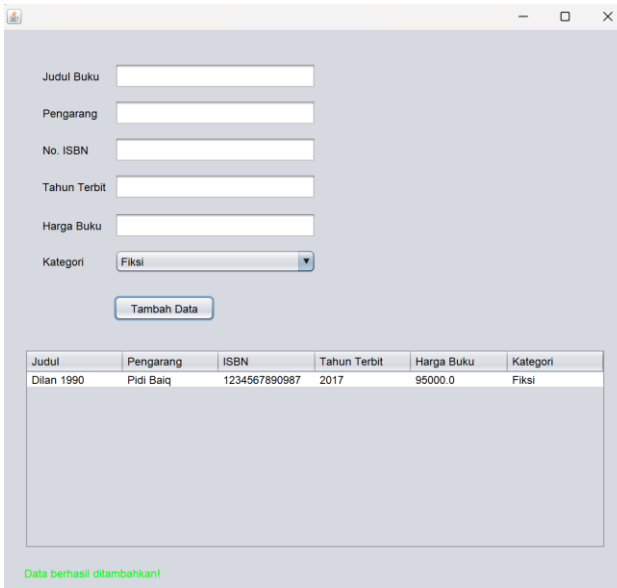
        txtJudul.setText("");
        txtPengarang.setText("");
    }
}
```

```
txtIsbn.setText("");  
tahunTerbit1.setText("");  
hargaBuku1.setText("");  
kategori1.setSelectedIndex(0);  
lblMessage.setText("Data berhasil ditambahkan!");  
lblMessage.setForeground(Color.green);  
} catch (ValidasiInputException e) {  
    lblMessage.setText("Error:" + e.getMessage());  
    lblMessage.setForeground(Color.red);  
}  
}
```

Penjelasan:

Pada metode `tombolActionPerformed`, kode ini diperbarui untuk menangani input tambahan terkait buku, termasuk `tahunTerbit`, `hargaBuku`, dan `kategori`. Pertama, nilai-nilai diambil dari field teks dan dropdown yang sesuai. Kemudian, objek `Buku` dibuat dengan memanfaatkan konstruktor yang baru, yang kini menerima parameter tambahan untuk validasi yang lebih ketat terhadap data yang dimasukkan. Setelah objek berhasil dibuat, data buku ditambahkan ke model tabel (`DefaultTableModel`), dan semua field input direset untuk persiapan entri berikutnya. Pesan sukses ditampilkan untuk memberi tahu pengguna bahwa data telah berhasil ditambahkan. Jika ada kesalahan validasi, pesan kesalahan ditampilkan kepada pengguna dengan warna merah.

Output:



Judul	Pengarang	ISBN	Tahun Terbit	Harga Buku	Kategori
Dilan 1990	Pidi Baiq	1234567890987	2017	95000.0	Fiksi

Data berhasil ditambahkan!

Judul Buku

Dilan 1995

Pengarang

Pidi Baiq

No. ISBN

1234567890123

Tahun Terbit

18

Harga Buku

97000

Kategori

Fiksi

Tambah Data

Judul	Pengarang	ISBN	Tahun Terbit	Harga Buku	Kategori
Dilan 1990	Pidi Baiq	1234567890987	2017	95000.0	Fiksi

Error:Tahun terbit harus 4 digit