

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL V
LATIHAN EXCEPTION HANDLING**



Oleh :
Marsya Valeria
2311103105
S1SI-07-C

**PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
UNIVERSITAS TELKOM PURWOKERTO
2024**

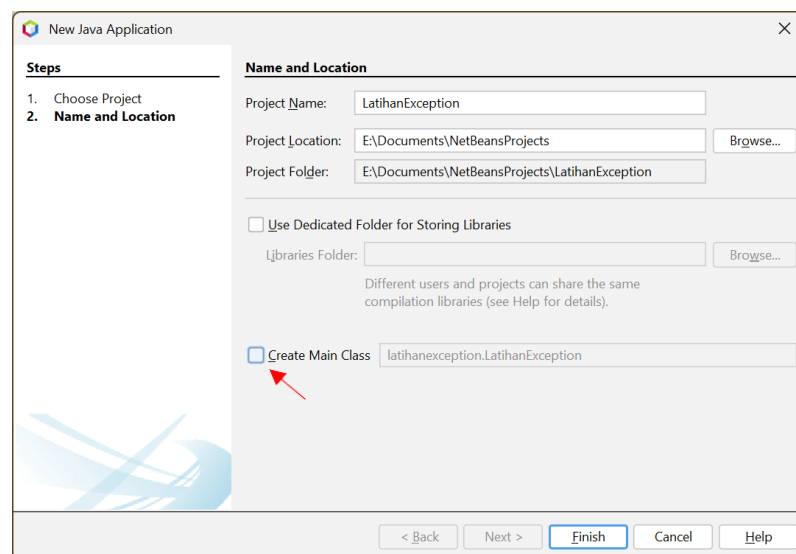
LATIHAN EXCEPTION HANDLING

A. Studi Kasus

Sistem Informasi Perpustakaan – Input Buku

B. Guided

1. Buat project baru pada Netbeans. Pada saat membuat project, hilangkan centang pada *Create Main Class*



2. Tambahkan package baru dengan nama **perpustakaan** dengan cara klik kanan pada *Source Packages* ☐ *New* ☐ *Java Package...*
3. Pada *package* perpustakaan, buat Class baru dengan nama **Buku**

Buku.Java

```
public class Buku { private String
    judul;
    private String pengarang; private String
    isbn;

    public Buku(String judul, String pengarang, String isbn){ this.judul = judul;
        this.pengarang = pengarang;
        this.isbn = isbn;
    }
}
```

```

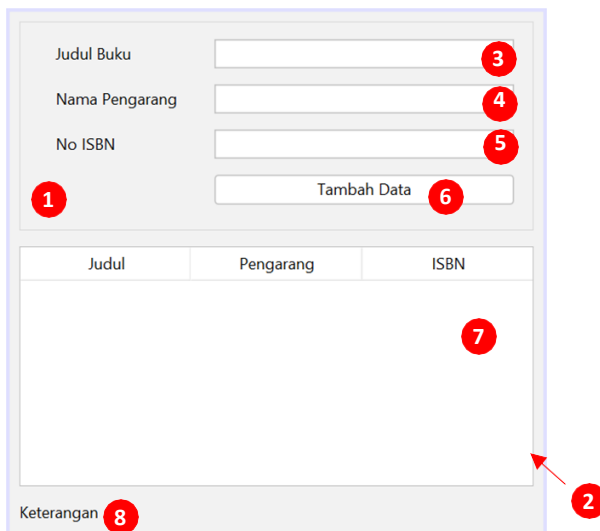
public String getJudul() { return judul;
}

public String getPengarang() { return
    pengarang;
}

public String getIsbn() { return isbn;
}
}

```

4. Tambahkan tampilan program untuk input data dengan cara klik kanan pada *package* perpustakaan □ New □ JFrame Form... , beri nama **InputBuku**
5. Buat tampilan seperti di bawah ini:



No.	Component	Variable Name
1	JPanel	
2	JScrollPane	
3	TextField	txtJudul
4	TextField	txtPengarang
5	TextField	txtIsbn
6	Button	btnSubmit
7	JTable	tblBuku
8	JLabel	lblKeterangan

6. Klik 2x pada tombol Tambah Data, tambahkan kode berikut

```

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

    // Membuat objek Buku baru
    Buku buku = new Buku(judul, pengarang, isbn);

    // Menambah data buku ke tabel DefaultTableModel data =
    (DefaultTableModel)

```

```
data.addRow(new Object[]{buku.getJudul(), buku.getPengarang(), buku.getIsbn()});

// Reset form input
txtJudul.setText("");
txtPengarang.setText("");
txtIsbn.setText("");
}
```

7. Jalankan project lalu coba tambahkan data, maka akan muncul data pada tabel. Namun ketika form kosong lalu klik tambah data, maka akan ada data kosong yang masuk pada tabel. Hal ini bisa diantisipasi dengan membuat *user defined exception*.
8. Buat sebuah class baru dengan nama **ValidasiInputException**

ValidasiInputException.java

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

9. Ubah kode pada Class buku

Buku.java

```
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 buku tidak boleh
kosong!");
        }
        if (pengarang.isEmpty()) {
            throw new ValidasiInputException("Pengarang tidak boleh
kosong!");
        }
        if (!isbn.matches("\\d{13}")) {
            throw new ValidasiInputException("ISBN harus 13 digit 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;
    }
}

```

10. Klik 2x pada tombol Tambah Data, ubah kode berikut

```

private void btnSubmitActionPerformed(java.awt.event.ActionEvent evt)
{
    try {

        String judul = txtJudul.getText();
        String pengarang = txtPengarang.getText(); String isbn =
        txtIsbn.getText();

        // Membuat objek Buku baru
        Buku buku = new Buku(judul, pengarang, isbn);

        // Menambah data buku ke tabel DefaultTableModel data =
        (DefaultTableModel)
tblBuku.getModel();

        data.addRow(new Object[]{buku.getJudul(), buku.getPengarang(),
buku.getIsbn()});

        // Reset form input txtJudul.setText("");
        txtPengarang.setText(""); txtIsbn.setText("");

        lblKeterangan.setForeground(Color.GREEN); lblKeterangan.setText("Buku
        berhasil ditambahkan!");

    } catch (ValidasiInputException e) {

```

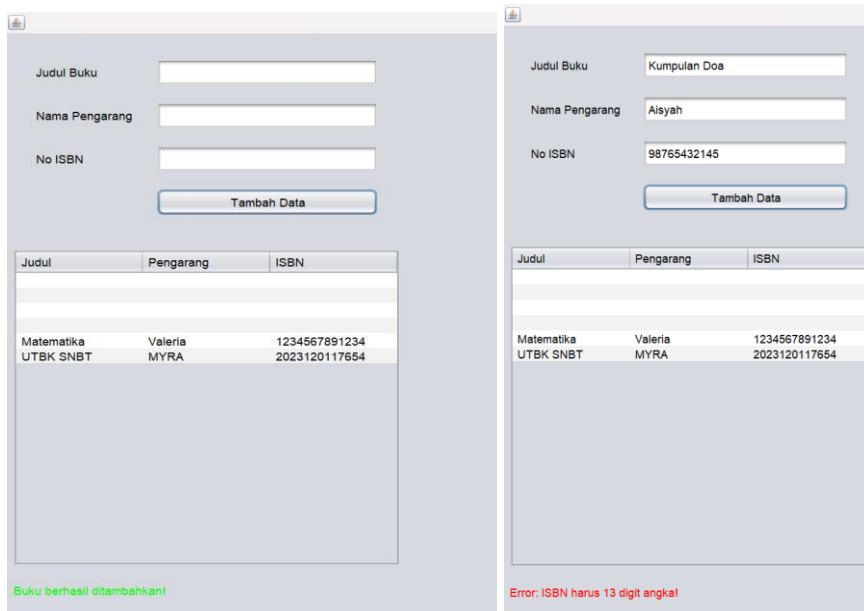
```

        lblKeterangan.setForeground(Color.RED); lblKeterangan.setText("Error: "
        + e.getMessage());
    }
}

```

11. Jalankan project, coba tambahkan data baru. Ketika form kosong maka akan muncul peringatan pada bagian bawah aplikasi

Hasil Run :



The application interface consists of three input fields: 'Judul Buku', 'Nama Pengarang', and 'No ISBN', each with a corresponding text box. Below these fields is a 'Tambah Data' button. Underneath the button is a table with three columns: 'Judul', 'Pengarang', and 'ISBN'. The table contains two rows of data: 'Matematika' by 'Valeria' with ISBN '1234567891234', and 'UTBK SNBT' by 'MYRA' with ISBN '2023120117654'. At the bottom of the application window, there is a status bar. In the left screenshot, it displays a green message: 'Buku berhasil ditambahkan!'. In the right screenshot, it displays a red error message: 'Error: ISBN harus 13 digit angka!'.

Penjelasan :

Kode program di atas adalah kode program bahasa java yang di gunakan untuk membuat sistem informasi perpustakaan. Pada program ini kita dapat memasukkan berbagai informasi terkait buku, seperti Judul Buku, Nama Pengarang, No ISBN. Sebelum data dimasukkan ke tabel program akan mengecek apakah semua data yang dimasukkan sudah benar. Seperti No ISBN apakah berupa angka 13 digit. Jika No ISBN tidak 13 digit maka program akan menampilkan peringatan bahwa ISBN harus 13 digit angka. Setelah semuanya benar baru data buku akan dimasukkan ke tabel.

C. 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)

Kode :

ValidasiInputException.java

```
package perpustakaan;

/**
 *
 * @author MARSYA VALERIA
 * 2311103105
 * 07C
 */
public class ValidasiInputException extends Exception {
    public ValidasiInputException(String message){
        super(message);
    }
}
```

Buku.java

```
package perpustakaan;

/**
 *
 * @author MARSYA VALERIA
 * 2311103105
 * 07C
 */
public class Buku {
    private String judul;
    private String pengarang;
    private String isbn;
    private int tahun;
    private double harga;
    private String kategori;

    public Buku(String judul, String pengarang, String isbn, int tahun, double harga,
        String kategori) throws ValidasiInputException {
        if (judul.isEmpty()) {
            throw new ValidasiInputException("Judul buku tidak boleh kosong!");
        }
    }
}
```

```
}  
if (pengarang.isEmpty()) {  
    throw new ValidasiInputException("Pengarang tidak boleh kosong!");  
}  
if (!isbn.matches("\\d{13}")) {  
    throw new ValidasiInputException("ISBN harus 13 digit angka!");  
}  
if (tahun < 1000 || tahun > 9999) {  
    throw new ValidasiInputException("Tahun harus 4 digit!");  
}  
if (harga <= 0) {  
    throw new ValidasiInputException("Harga harus angka positif!");  
}  
if (kategori == null || kategori.isEmpty()) {  
    throw new ValidasiInputException("Kategori tidak boleh kosong!");  
}  
  
this.judul = judul;  
this.pengarang = pengarang;  
this.isbn = isbn;  
this.tahun = tahun;  
this.harga = harga;  
this.kategori = kategori;  
}  
  
public String getJudul() {  
    return judul;  
}  
  
public String getPengarang() {  
    return pengarang;  
}  
  
public String getIsbn() {  
    return isbn;  
}  
  
public int getTahun() {  
    return tahun;  
}  
  
public double getHarga() {  
    return harga;  
}
```



```
public String getKategori() {  
    return kategori;  
}  
}
```

InputBuku

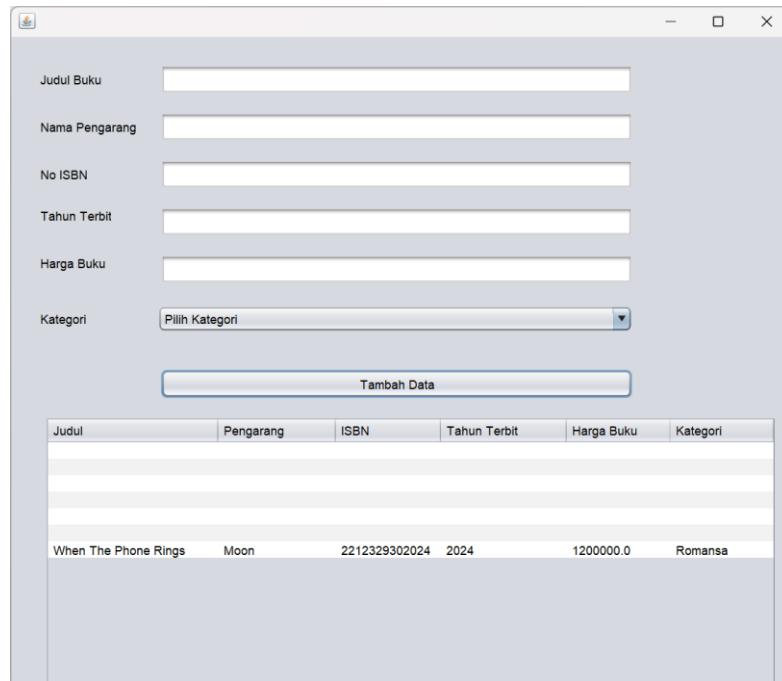
```
private void btnSubmitActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try {  
        String judul = txtJudul.getText();  
        String pengarang = txtPengarang.getText();  
        String isbn = txtIsbn.getText();  
        int tahun = Integer.parseInt(TahunTerbit.getText());  
        double harga = Double.parseDouble(Harga.getText());  
        String kategori = (String) Kategori.getSelectedItem();  
  
        System.out.println("Data yang akan ditambahkan:");  
        System.out.println("Judul: " + judul);  
        System.out.println("Pengarang: " + pengarang);  
        System.out.println("ISBN: " + isbn);  
        System.out.println("Tahun Terbit: " + tahun);  
        System.out.println("Harga: " + harga);  
        System.out.println("Kategori: " + kategori);  
  
        // Validasi kategori  
        if ("Pilih Kategori".equals(kategori)) {  
            throw new ValidasiInputException("Kategori buku harus dipilih!");  
        }  
  
        // Membuat objek Buku baru dengan validasi input  
        Buku buku = new Buku(judul, pengarang, isbn, tahun, harga, kategori);  
  
        // Menambah data buku ke tabel  
        DefaultTableModel data = (DefaultTableModel) tblBuku.getModel();  
        data.addRow(new Object[]{  
            buku.getJudul(),  
            buku.getPengarang(),  
            buku.getIsbn(),  
            buku.getTahun(),  
            buku.getHarga(),  
            buku.getKategori()});  
    }  
}
```

```
System.out.println("Data berhasil ditambahkan ke tabel.");
```

```
// Reset form input
txtJudul.setText("");
txtPengarang.setText("");
txtIsbn.setText("");
TahunTerbit.setText("");
Harga.setText("");
Kategori.setSelectedIndex(0); // Reset dropdown ke default

lblKeterangan.setForeground(Color.GREEN);
lblKeterangan.setText("Buku berhasil ditambahkan!");
} catch (ValidasiInputException e) {
    System.out.println("Validasi Error: " + e.getMessage());
    lblKeterangan.setForeground(Color.RED);
    lblKeterangan.setText("Error: " + e.getMessage());
} catch (NumberFormatException e) {
    System.out.println("Number Format Error: " + e.getMessage());
    lblKeterangan.setForeground(Color.RED);
    lblKeterangan.setText("Error: Format input salah!");
}
```

Hasil Run :



Judul	Pengarang	ISBN	Tahun Terbit	Harga Buku	Kategori
When The Phone Rings	Moon	2212329302024	2024	1200000.0	Romansa

Penjelasan :

Kode program di atas adalah kode program bahasa java yang di gunakan untuk membuat sistem informasi perpustakaan. Pada program ini kita dapat memasukkan berbagai informasi terkait buku, seperti Judul Buku, Nama Pengarang, No ISBN, Tahun Terbit, Harga Buku, dan Kategori. Sebelum data dimasukkan, program akan mengecek apakah semua informasi yang dimasukkan sudah benar. Seperti, No ISBN harus berupa angka 13 digit, Tahun Terbit adalah 4 digit, Harga Buku harus berupa angka positif, dan Kategori harus dipilih. Jika ada data yang salah atau kosong, program akan memberikan pesan peringatan supaya pengguna bisa memperbaikinya. Setelah data sudah benar atau sudah diperbaiki, maka semua informasi buku tersebut akan masuk ke tabel.