

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERBASIS ORIENTASI OBJEK  
MODUL 5**



**Nama : Levina Maheswari M.L  
NIM : 2311103110  
Kelas : S1SI07C**

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

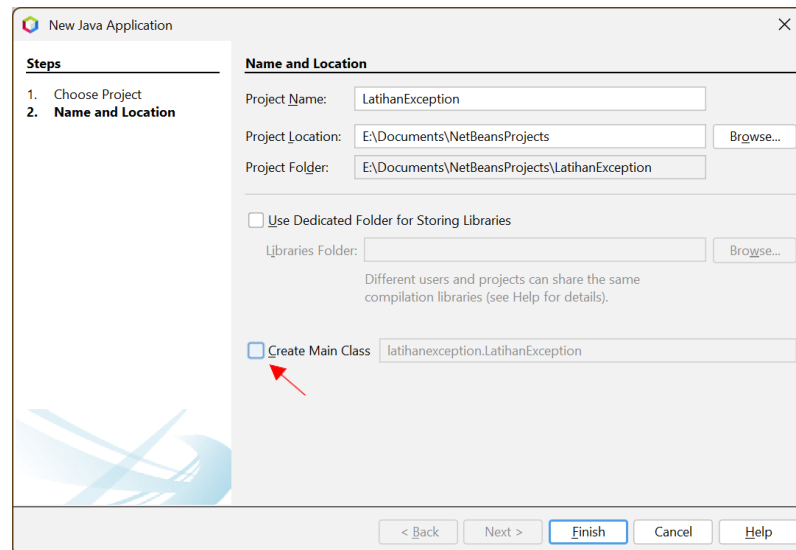
# LATIHAN EXCEPTION HANDLING

## 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*



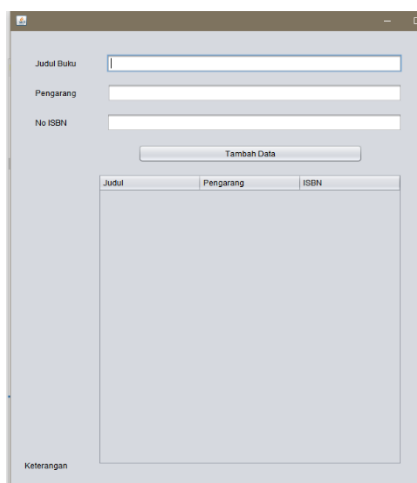
2. Tambahkan package baru dengan nama **perpustakaan** dengan cara klik kanan pada *Source Packages* ☐ *New* ☐ *Java Package...*
3. Pada *package* perpustakaan, buat Class baru dengan nama **Buku**

## Buku.Java

```
package Perpustakaan;
/**
 *
 * @author Levina Maheswari
 * 2311103110/S1SI07C
 */
public class Buku {
    private String judul;
    private String pengarang;
    private String 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	judul
4	TextField	pengarang
5	TextField	isbn
6	Button	tombol
7	JTable	tableBuku
8	JLabel	keterangan

```
private void tombolActionPerformed(java.awt.event.ActionEvent evt) {  
    String judul1 = judulBuku.getText();  
    String pengarang1 = pengarang.getText();  
    String isbn1 = isbn.getText();  
  
    Buku buku = new Buku(judul1, pengarang1, isbn1);  
    DefaultTableModel data = (DefaultTableModel)bukutable.getModel();  
    data.addRow(new Object[]{ buku.getJudul(), buku.getPengarang(), buku.getIsbn()});  
  
    judulBuku.setText("");  
    pengarang.setText("");  
    isbn.setText("");  
}
```

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

```
package Perpustakaan;  
/**  
 * @author Levina Maheswari  
 * 2311103110/S1SI07C  
 */  
public class ValidasiInputException extends Exception {  
    public ValidasiInputException(String Message) {  
        super(Message);  
    }  
}
```

## 9. Ubah kode pada Class buku

Buku.java

```
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("Judul Kosong");
    }
    this.judul = judul;
    this.pengarang = pengarang;
    this.isbn = isbn;
}
```

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

```
private void tombolActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String judul1 = judulBuku.getText();
        String pengarang1 = pengarang.getText();
        String isbn1 = isbn.getText();

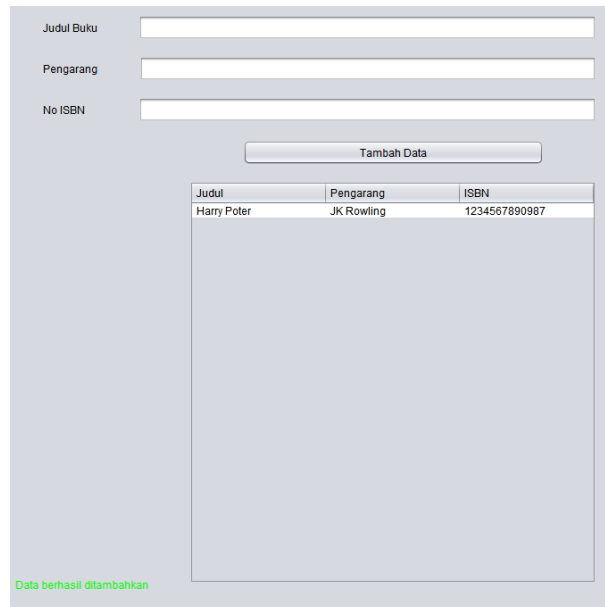
        Buku buku = new Buku(judul1, pengarang1, isbn1);
        DefaultTableModel data = (DefaultTableModel)bukutable.getModel();
        data.addRow(new Object[]{buku.getJudul(), buku.getPengarang(), buku.getIsbn()});

        judulBuku.setText("");
        pengarang.setText("");
        isbn.setText("");
        lblMessage.setText("Data berhasil ditambahkan");
        lblMessage.setForeground(Color.green);

    } catch (ValidasiInputException e){
        lblMessage.setText("Error :"+e.getMessage());
    }
}
```

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

**Data berhasil ditambahkan :**

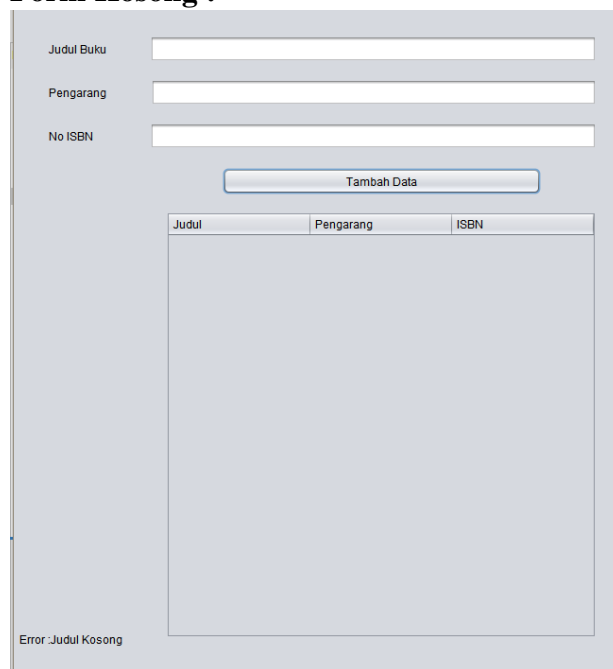


The screenshot shows a web application interface for adding book data. It features three input fields labeled 'Judul Buku', 'Pengarang', and 'No ISBN'. Below these fields is a 'Tambah Data' button. A table below the button displays the added data:

Judul	Pengarang	ISBN
Harry Potter	JK Rowling	1234567890987

A green message at the bottom left states 'Data berhasil ditambahkan'.

**Form Kosong :**



The screenshot shows the same application interface as above, but the input fields are empty. The 'Tambah Data' button is highlighted with a blue border. The table below the button is empty. An error message at the bottom left states 'Error :Judul Kosong'.

Kode tersebut merupakan aplikasi GUI yang digunakan untuk mengelola data buku. Class Buku menggambarkan objek buku dengan atribut seperti judul, pengarang, dan ISBN, serta memvalidasi input melalui constructor-nya. Jika input tidak valid, custom exception `ValidasiInputException` akan di jalankan, yang dirancang sebagai class khusus untuk menangani kesalahan input. Dalam metode `tombolActionPerformed`, data yang diinput dari form GUI diproses, divalidasi, dan jika valid, objek Buku ditambahkan ke tabel. Jika terjadi kesalahan, pesan error akan ditampilkan.

## 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)

### Buku.java

```
package Perpustakaan;
/**
 * @author Levina Maheswari
 * 2311103110/S1SI07C
 */
public class Buku {
    private String judul;
    private String pengarang;
    private String isbn;
    private String tahunTerbit;
    private String 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("Judul Kosong");
    }
    if(!tahunTerbit.matches("\\d{4}")){
        throw new ValidasiInputException("Tahun Terbit Kosong");
    }
    if(hargaBuku.isEmpty()){
        throw new ValidasiInputException("Harga Buku Kosong");
    }
    if(kategori.isEmpty()|| kategori == null){
        throw new ValidasiInputException("Kategori Kosong");
    }
    this.judul = judul;
    this.pengarang = pengarang;
    this.isbn = isbn;
    this.tahunTerbit = tahunTerbit;
    this.hargaBuku = 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 String gethargaBuku(){
    return hargaBuku;
}
public String getkategori(){
    return kategori;
}
}
```

### inputData.java

```
private void tombolActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String judul1 = judulBuku.getText();
        String pengarang1 = pengarang.getText();
        String isbn1 = isbn.getText();
        String tahunTerbit1 = tahunTerbit.getText();
        String hargaBuku1 = hargaBuku.getText();
        String kategori1 = (String) kategori.getSelectedItem();

        Buku buku = new Buku(judul1, pengarang1, isbn1, tahunTerbit1, hargaBuku1, kategori1);
        DefaultTableModel data = (DefaultTableModel)bukutable.getModel();
        data.addRow(new Object[]{ buku.getJudul(),
            buku.getPengarang(),buku.getIsbn(),buku.gettahunTerbit(),buku.gethargaBuku(), buku.getkategori()});

        judulBuku.setText("");
        pengarang.setText("");
        isbn.setText("");
        tahunTerbit.setText("");
        hargaBuku.setText("");
        kategori.setSelectedItem("");
        lblMessage.setText("Data berhasil ditambahkan");
        lblMessage.setForeground(Color.green);

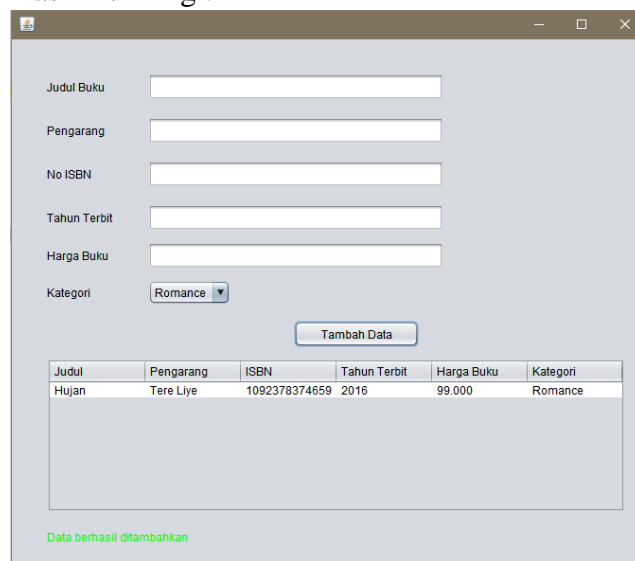
    } catch (ValidasiInputException e){
        lblMessage.setText("Error :"+e.getMessage());
    }
}
```



## ValidasiInputException.java

```
package Perpustakaan;
/**
 * @author Levina Maheswari
 * 2311103110/S1SI07C
 */
public class ValidasiInputException extends Exception {
    public ValidasiInputException(String Message) {
        super(Message);
    }
}
```

### Hasil Running :



Judul	Pengarang	ISBN	Tahun Terbit	Harga Buku	Kategori
Hujan	Tere Liye	1092378374659	2016	99.000	Romance

Data berhasil ditambahkan

### Jelaskan :

Kode tersebut merupakan aplikasi GUI yang digunakan untuk mengelola data buku. Class Buku menggambarkan objek buku dengan atribut seperti judul, pengarang, ISBN, tahun terbit, harga, dan kategori, serta memvalidasi input melalui constructor-nya. Jika input tidak valid, custom exception `ValidasiInputException` akan di jalankan, yang dirancang sebagai class khusus untuk menangani kesalahan input. Dalam metode `tombolActionPerformed`, data yang diinput dari form GUI diproses, divalidasi, dan jika valid, objek Buku ditambahkan ke tabel. Jika terjadi kesalahan, pesan error akan ditampilkan.