PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK MODUL V EXCEPTION HANDLING



Disusun oleh:

ZOE SATRIAN 2311103141 S1SI-07-C

PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
TELKOM UNIVERSITY PURWOKERTO
PURWOKERTO
2024

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;
     public class Buku {
      private String judul;
      private String pengarang;
      private String isbn;
      public Buku(String judul, String pengarang, String isbn){
n Beror a Wijayanto, S.Pd., M.T
```

Pemrogram

```
this.pengarang = pengarang;
this.isbn = isbn;

public String getJudul() { return judul;
}

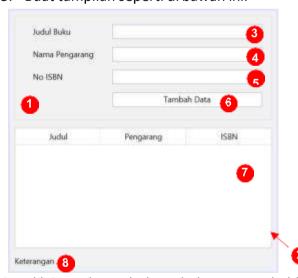
public String getPengarang() { return pengarang;
}

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	JTextField	txtJudul		
4	JTextField	txtPengarang		
5	JTextField	txtlsbn		
6	JButton	btnSubmit		
7	JTable	tblBuku		
8	JLabel	IblKeterangan		
8	JLabel	lblKeterangan		

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

```
String judul = txtJudul.getText();

String pengarang = txtPengarang.getText();

String isbn = txtIsbn.getText();
```

```
// Membuat model tabel
javax.swing.table.DefaultTableModel data =
(javax.swing.table.DefaultTableModel) tblBuku.getModel();
  data.addRow(new Object[]{judul, pengarang, isbn});

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

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

9. Ubah kode pada Class buku

```
public Buku(String judul, String pengarang, String isbn) throws
ValidationException {
```

```
if (judul.isEmpty()) {
throw new ValidationException("Judul buku tidak boleh kosong!");
}
if (pengarang.isEmpty()) {
throw new ValidationException("Pengarang tidak boleh kosong!");
}
if (!isbn.matches("\\d{13}")) {
throw new ValidationException("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;
}
```

```
try {
      String judul = txtJudul.getText();
      String pengarang = txtPengarang.getText();
      String isbn = txtIsbn.getText();
      // Membuat model tabel
      javax.swing.table.DefaultTableModel data =
(javax.swing.table.DefaultTableModel) tblBuku.getModel();
      data.addRow(new Object[]{judul, pengarang, isbn});
      // Reset form input
      txtJudul.setText("");
      txtPengarang.setText("");
      txtlsbn.setText("");
      lblKeterangan.setForeground(Color.GREEN);
      lblKeterangan.setText("Buku berhasil ditambahkan!");
      lblKeterangan.setForeground(Color.GREEN);
lblKeterangan.setText("Buku berhasil ditambahkan!");
    } catch (Exception e) {
    lblKeterangan.setForeground(Color.RED);
    lblKeterangan.setText("Error: " + e.getMessage());
```



A. 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)

Tambahan code

Buku.java

```
this.tahunTerbit = tahunTerbit;

this.hargaBuku = hargaBuku;

this.kategori = kategori;

public String getTahunTerbit() {

return tahunTerbit;
```

```
public String gethargaBuku() {
    return hargaBuku;
}

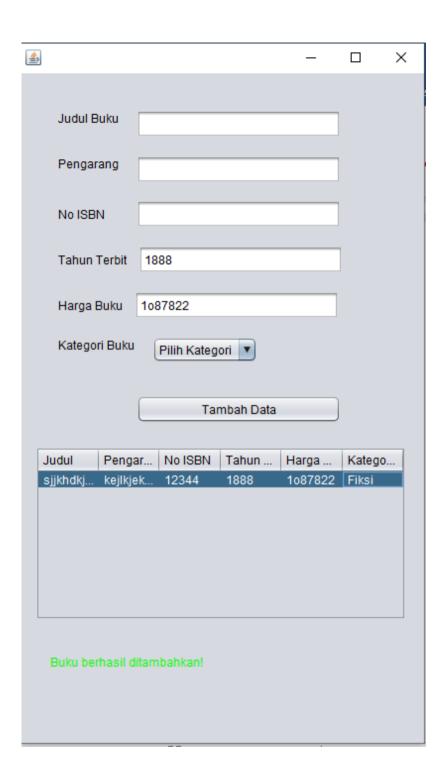
public String getkategori() {
    return kategori;
}
```

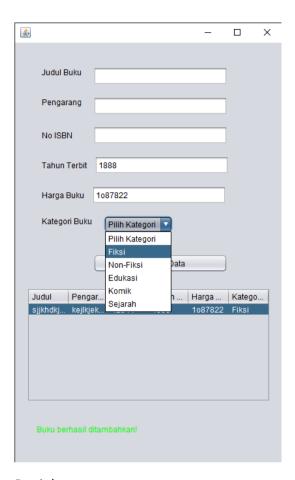
Tambahan code pada UIFrame, pada bagian "private void btnSubmitActionPerformed"

Dan "private void comboBoxActionPerformed"

```
String tahunTerbit = txtTahunTerbit.getText();
           String hargaBuku = txtHargaBuku.getText();
      String kategori = comboBox.getSelectedItem().toString();
    // Membuat model tabel
           javax.swing.table.DefaultTableModel
                                                              data
        (javax.swing.table.DefaultTableModel) tblBuku.getModel();
      data.addRow(new Object[]{judul, pengarang, isbn, tahunTerbit, hargaBuku,
kategori});
    private void comboBoxActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String selectedItem = comboBox.getSelectedItem().toString();
        comboBox.setModel(new
                                      javax.swing.DefaultComboBoxModel<>(new
        String[]{
         "Pilih Kategori", "Fiksi", "Non-Fiksi", "Edukasi", "Komik", "Sejarah"
      }));
  }
```

Output:





Penjelasan:

Pada kategori buku element yang digunakan yaitu combo box, dengan terdapat 4 value kategori. Yaitu Fiksi, Non-Fiksi, Edukasi, Komik dan Sejarah.