

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 5
LATIHAN EXCEPTION HANDLING**



Oleh :
Rizqi Dwi Saputra
2311103133
S1SI – 07 – C

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

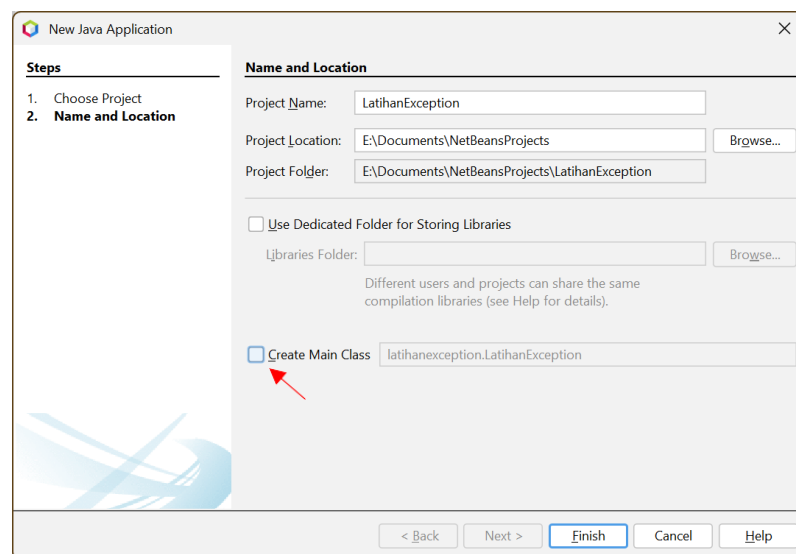
LATIHAN EXCEPTION HANDLING

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;
    }
}
```

```

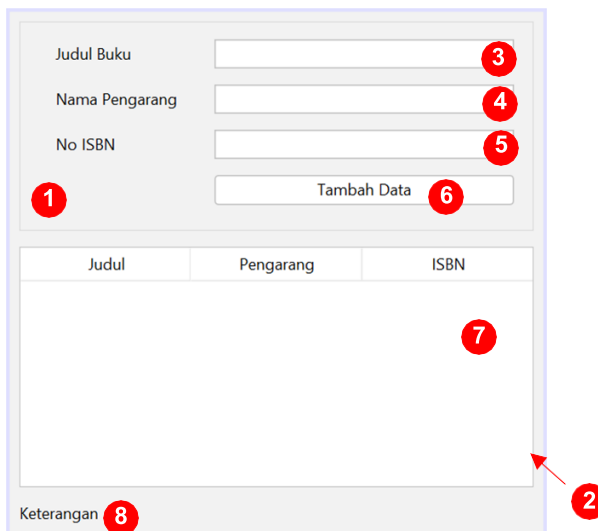
public String getJudul() {
    return judul;
}

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)
tblBuku.getModel();

```

```
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

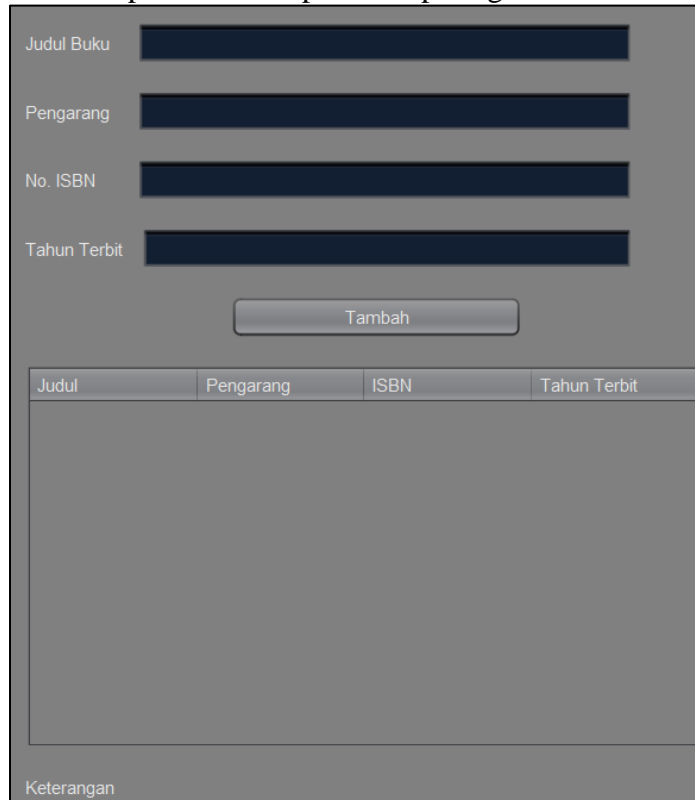
C. Unguided

Tambahkan inputan berikut pada aplikasi:

1. Tahun Terbit: Validasi untuk angka empat digit.

Jawab:

Langkah pertama yang harus dilakukan adalah menambahkan label dan inputan Tahun Terbit pada design program (variabel textField: txtTahun), kemudian tambahkan kolom “Tahun Terbit” pada tabel untuk menampilkan data inputan. Seperti gambar berikut ini:



ValidasiInputExc.java:

```
package unguided_1;
//@author Rizqi Dwi Saputra_2311103133

public class ValidasiInputExc extends Exception {

    public ValidasiInputExc(String message){
        super(message);
    }

}
```

Kelas ini digunakan sebagai Exception Handling pada project kali ini.

Buku.java

```
package unguided_1;
//@author Rizqi Dwi Saputra_2311103133

public class Buku {
```

```
private String judul;
private String pengarang;
private String isbn;
private String tahun;

public Buku(String judul, String pengarang, String isbn, String tahun,
throws ValidasiInputExc {

    if(judul.isEmpty()){
        throw new ValidasiInputExc("Silahkan isi Judul buku");
    }

    if(pengarang.isEmpty()){
        throw new ValidasiInputExc("Silahkan isi Nama Pengarang");
    }

    if(!isbn.matches("\\d{13}")){
        throw new ValidasiInputExc("Nomor ISBN minimal 13 angka");
    }

    if(tahun.isEmpty()){
        throw new ValidasiInputExc("Silahkan isi Tahun Terbit");
    }

    if(!tahun.matches("\\d+")){
        throw new ValidasiInputExc("Tahun Terbit tidak boleh pakai huruf
gusy");
    }

    if(!tahun.matches("\\d{4}")){
        throw new ValidasiInputExc("Tahun Terbit harus 4 angka guys");
    }

    this.judul = judul;
    this.pengarang = pengarang;
    this.isbn = isbn;
    this.tahun = tahun;
}

public String getJudul(){
    return judul;
}

public String getPengarang(){
    return pengarang;
}

public String getIsbn(){
    return isbn;
}

public String getTahun(){
    return tahun;
}
}
```

Modifikasi kode pada kelas Buku, yakni buat dulu variabel tahun kemudian inisialisasi variabel

tersebut pada konstruktor. Kemudian gunakan exception handling untuk memvalidasi inputan user, jika user tidak menginput tahun pada Text Field tahun maka muncul pesan “Silahkan isi Tahun Terbit”, jika user menginput huruf pada Text Field tahun maka muncul pesan “Tahun Terbit tidak boleh pakai huruf gusy”, jika user menginput kurang dari 4 angka pada Text Field tahun maka label keterangan akan muncul pesan “Tahun Terbit harus 4 angka guys”. Terakhir buat fungsi getTahun (getter) yang akan digunakan pada kelas InputData untuk menambahkan dan menampilkan data Tahun Terbit pada tabel ketika project dijalankan.

InputData.java:

```
package unguided_1;

import java.awt.Color;
import javax.swing.table.DefaultTableModel;
//@author Rizqi Dwi Saputra_2311103133

public class InputData extends javax.swing.JFrame {

    public InputData() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel3 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        txtJudul = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        txtPengarang = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        txtIsbn = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        txtTahun = new javax.swing.JTextField();
        btnSubmit = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        tableBuku = new javax.swing.JTable();
        lblKet = new javax.swing.JLabel();

        jLabel3.setText("jLabel3");

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setText("Judul Buku");

        txtJudul.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtJudulActionPerformed(evt);
            }
        });

        jLabel2.setText("Pengarang");

        jLabel4.setText("No. ISBN");

        jLabel5.setText("Tahun Terbit");
    }
}
```

```

btnSubmit.setText("Tambah");
btnSubmit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSubmitActionPerformed(evt);
    }
});

tableBuku.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Judul", "Pengarang", "ISBN", "Tahun Terbit"
    }
));
jScrollPane1.setViewportViewView(tableBuku);

lblKet.setText("Keterangan");

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createSequentialGroup()
                        .addGap(31, 31, 31)

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
                            .addGroup(layout.createSequentialGroup()
                                .addGroup(layout.createSequentialGroup()
                                    .addComponent(jLabel5)

                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                                    .addComponent(txtTahun,
javax.swing.GroupLayout.DEFAULT_SIZE, 330, Short.MAX_VALUE))
                                .addGroup(layout.createSequentialGroup()
                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addGroup(layout.createSequentialGroup()
                                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                                .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                                .addComponent(jLabel1,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
                                            .addComponent(jLabel4))
                                        .addGap(18, 18, 18)

```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addComponent(txtJudul)
    .addComponent(txtPengarang)
    .addComponent(txtIsbn,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
    .addComponent(lblKet))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    .addComponent(btnSubmit, javax.swing.GroupLayout.PREFERRED_SIZE,
196, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(128, 128, 128))
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup())
    .addGap(32, 32, 32)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel1)
    .addComponent(txtJudul,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel2)
    .addComponent(txtPengarang,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel4)
    .addComponent(txtIsbn,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel5)
    .addComponent(txtTahun,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)
    .addComponent(btnSubmit)
    .addGap(18, 18, 18)
    .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 258,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addComponent(lblKet)
```

```

        .addContainerGap(19, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void txtJudulActionPerformed(java.awt.event.ActionEvent evt) {

}

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

    try{
        String judul = txtJudul.getText();
        String pengarang = txtPengarang.getText();
        String isbn = txtIsbn.getText();
        String tahun = txtTahun.getText();

        Buku buku = new Buku(judul, pengarang, isbn, tahun);

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

        txtJudul.setText("");
        txtPengarang.setText("");
        txtIsbn.setText("");
        txtTahun.setText("");
        lblKet.setText("Sukses! Data berhasil ditambahkan");
        lblKet.setForeground(Color.GREEN);
    }catch(ValidasiInputExc e){
        lblKet.setText("Error : " + e.getMessage());
        lblKet.setForeground(Color.RED);
    }

}

public static void main(String args[]) {

    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(InputData.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);

```

```

        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(InputData.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(InputData.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(InputData.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
    }
    //</editor-fold>

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new InputData().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btnSubmit;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel lblKet;
private javax.swing.JTable tableBuku;
private javax.swing.JTextField txtIsbn;
private javax.swing.JTextField txtJudul;
private javax.swing.JTextField txtPengarang;
private javax.swing.JTextField txtTahun;
// End of variables declaration
}

```

Modifikasi kode pada variabel button (btnSubmit) yaitu menambahkan variabel tahun yang diisi dengan nilai text yang diinput (getText). Kemudian tambahkan variabel tahun yang sudah dibuat kedalam objek buku. Terakhir, tambahkan kode buku.getTahun() pada DefaultTableModel agar data inputan (termasuk tahun) akan masuk dan ditampilkan kedalam tabel. Tetapi, karena kode tersebut ada didalam Exception Handling, maka akan ada validasi terlebih dahulu pada inputan, jika inputan salah (tidak sesuai) maka muncul pesan error pada label keterangan jika tidak maka data berhasil ditambahkan kedalam tabel.

Hasil Project:

Judul Buku

Pengarang

No. ISBN

Tahun Terbit

Judul	Pengarang	ISBN	Tahun Terbit

Error : Tahun Terbit tidak boleh pakai huruf gusy

Judul Buku	<input type="text" value="Kok Putra Ganteng"/>
Pengarang	<input type="text" value="Tegar"/>
No. ISBN	<input type="text" value="1019182837465"/>
Tahun Terbit	<input type="text" value="201"/>

Judul	Pengarang	ISBN	Tahun Terbit
-------	-----------	------	--------------

Error : Tahun Terbit harus 4 angka guys

Judul	Pengarang	ISBN	Tahun Terbit
Kok Putra Ganteng	Tegar	1019182837465	2023

Sukses! Data berhasil ditambahkan

2. Harga Buku: Validasi untuk angka positif.

Jawab:

Langkah pertama yang harus dilakukan adalah menambahkan label dan inputan Harga pada design program (variabel textField: txtHarga), kemudian tambahkan kolom “Harga” pada tabel untuk menampilkan data inputan. Seperti gambar berikut ini:



Buku.java:

```
package unguided_2;
//@author Rizqi Dwi Saputra_2311103133

public class Buku {

    private String judul;
    private String pengarang;
    private String isbn;
    private String tahun;
    private String harga;

    public Buku(String judul, String pengarang, String isbn, String tahun,
String harga) throws ValidasiInputExc {

        if(judul.isEmpty()){
            throw new ValidasiInputExc("Silahkan isi Judul buku");
        }

        if(pengarang.isEmpty()){
            throw new ValidasiInputExc("Silahkan isi Nama Pengarang");
        }
    }
}
```



```

        if(!isbn.matches("\\d{13}")){
            throw new ValidasiInputExc("Nomor ISBN minimal 13 angka");
        }

        if(tahun.isEmpty()){
            throw new ValidasiInputExc("Silahkan isi Tahun Terbit");
        }

        if(!tahun.matches("\\d+")){
            throw new ValidasiInputExc("Tahun Terbit tidak boleh pakai huruf
gusy");
        }

        if(!tahun.matches("\\d{4}")){
            throw new ValidasiInputExc("Tahun Terbit harus 4 angka guys");
        }

        if(harga.isEmpty()){
            throw new ValidasiInputExc("Silahkan isi Harga Buku");
        }

        if(!harga.matches("\\d+")){
            throw new ValidasiInputExc("Harga kok pakai Huruf?!");
        }

        int hargaInt = Integer.parseInt(harga);
        if(hargaInt <= 0){
            throw new ValidasiInputExc("Harga kok Negatif??!!");
        }

        this.judul = judul;
        this.pengarang = pengarang;
        this.isbn = isbn;
        this.tahun = tahun;
        this.harga = harga;
    }

    public String getJudul(){
        return judul;
    }

    public String getPengarang(){
        return pengarang;
    }

    public String getIsbn(){
        return isbn;
    }

    public String getTahun(){
        return tahun;
    }

    public String getHarga(){
        return harga;
    }

```

```
}
```

Membuat variabel harga pada kelas Buku, kemudian inisialisasi variabel harga pada konstruktor. Kemudian buat pesan validasi jika inputan harga kosong maka muncul pesan “Silahkan isi Harga Buku”, jika inputan harga berupa huruf maka muncul pesan “Harga kok pakai Huruf?!”. Untuk mengkonversi String menjadi Integer maka tambahkan lagi variabel bertipe data Integer yang akan mengkonversi variabel Harga (string) menjadi Integer dengan kode Integer.parseInt. Jika inputan harga berupa angka negatif maka muncul pesan “Harga kok Negatif?!!”. Terakhir buat fungsi getter getHarga yang akan digunakan pada kelas InputData untuk menambahkan dan menampilkan data Harga Buku pada tabel ketika project dijalankan.

InputData.java

```
private void btnSubmitActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try{  
        String judul = txtJudul.getText();  
        String pengarang = txtPengarang.getText();  
        String isbn = txtIsbn.getText();  
        String tahun = txtTahun.getText();  
        String harga = txtHarga.getText();  
  
        Buku buku = new Buku(judul, pengarang, isbn, tahun, harga);  
  
        String formatHarga = "Rp." + buku.getHarga();  
  
        DefaultTableModel data = (DefaultTableModel)tableBuku.getModel();  
        data.addRow(new Object[] {buku.getJudul(), buku.getPengarang(),  
buku.getIsbn(), buku.getTahun(), formatHarga});  
  
        txtJudul.setText("");  
        txtPengarang.setText("");  
        txtIsbn.setText("");  
        txtTahun.setText("");  
        lblKet.setText("Sukses! Data berhasil ditambahkan");  
        lblKet.setForeground(Color.GREEN);  
    }catch(ValidasiInputExc e){  
        lblKet.setText("Error : " + e.getMessage());  
        lblKet.setForeground(Color.RED);  
    }  
  
}
```

Menambahkan variable harga pada kode Button Submit yang diisi dengan inputan harga (getText) kemudian menginisialisasikan variabel tersebut kedalam objek buku. Untuk membuat tampilannya lebih menarik, maka saya membuat variabel baru bernama formatHarga yang nantinya akan memunculkan kata “Rp.” Sebelum harga inputan. Terakhir, panggil variabel formatHarga pada DefaultTableModel agar inputan (termasuk harga) akan masuk dan ditampilkan kedalam tabel. Tetapi, karena kode tersebut ada didalam Exception Handling, maka akan ada validasi terlebih dahulu pada inputan, jika inputan salah (tidak sesuai) maka muncul pesan error pada label keterangan jika tidak maka data berhasil ditambahkan kedalam tabel.

Hasil Project:

Judul Buku	<input type="text" value="Kok Putra Ganteng"/>
Pengarang	<input type="text" value="Rhojay"/>
No. ISBN	<input type="text" value="1122334455660"/>
Tahun Terbit	<input type="text" value="2024"/>
Harga Buku	<input type="text" value="aa"/>
<input type="button" value="Tambah"/>	

Judul	Pengarang	ISBN	Tahun Terbit	Harga
-------	-----------	------	--------------	-------

Error : Harga kok pakai Huruf?!

Judul Buku	<input type="text" value="Kok Putra Ganteng"/>
Pengarang	<input type="text" value="Rhojay"/>
No. ISBN	<input type="text" value="1122334455660"/>
Tahun Terbit	<input type="text" value="2024"/>
Harga Buku	<input type="text" value="-2000"/>

Judul	Pengarang	ISBN	Tahun Terbit	Harga

Error : Harga kok Negatif??!!

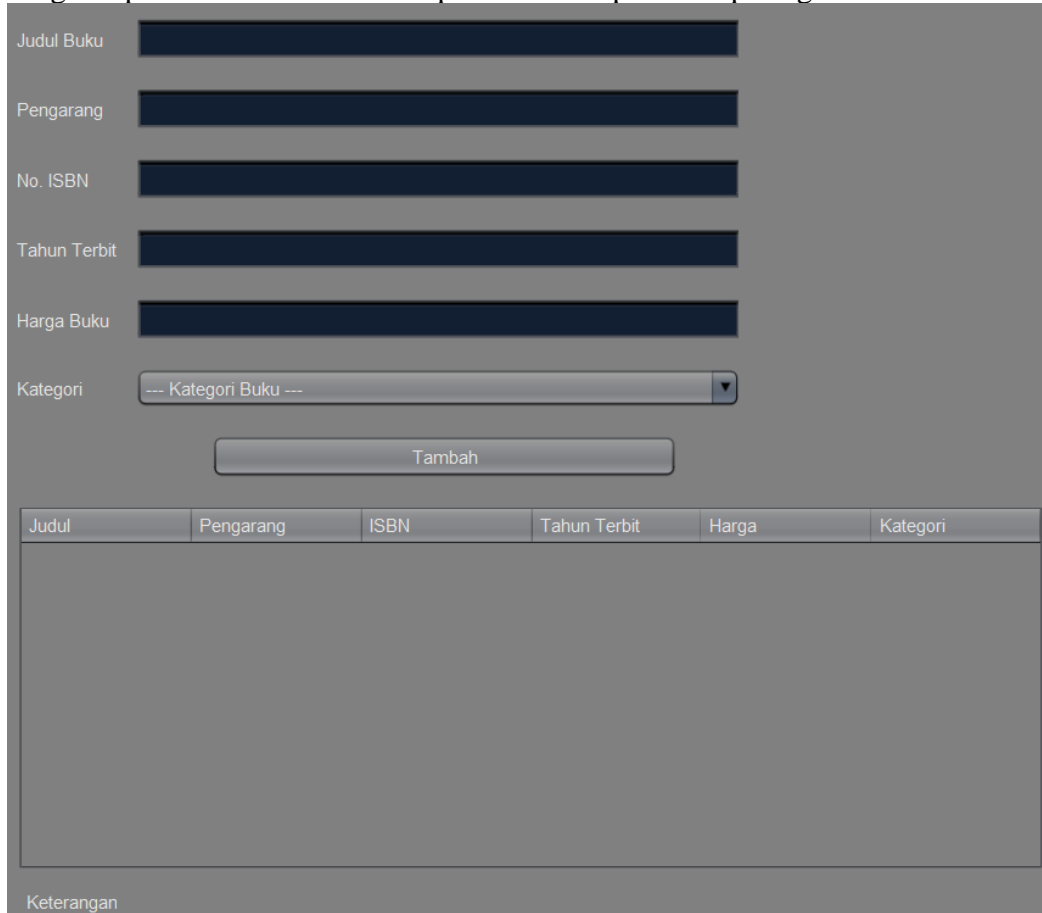
Judul	Pengarang	ISBN	Tahun Terbit	Harga
Kok Putra Ganteng	Rhojay	1122334455660	2024	Rp.200000

Sukses! Data berhasil ditambahkan

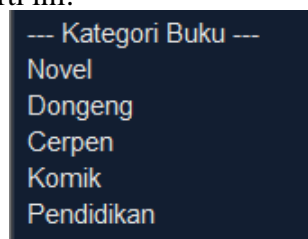
3. Kategori Buku: Validasi untuk input tidak kosong. (dalam bentuk Dropdown)

Jawab:

Langkah pertama yang harus dilakukan adalah menambahkan label dan inputan Kategori (JCombo Box) pada design program (variabel Combo Box: cbxKategori), kemudian tambahkan kolom “Kategori” pada tabel untuk menampilkan data inputan. Seperti gambar berikut ini:



Saya mengisi nilai JCombo Box seperti ini:



--- Kategori Buku --- berada dipaling atas sebagai pilihan default yang nantinya akan digunakan sebagai validasi Exception.

Buku.java:

```
package unguided_3;
//@author Rizqi Dwi Saputra_2311103133

public class Buku {

    private String judul;
    private String pengarang;
    private String isbn;
```

```

private String tahun;
private String harga;
private String kategori;

public Buku(String judul, String pengarang, String isbn, String tahun,
String harga, String kategori) throws ValidasiInputExc {

    if(judul.isEmpty()){
        throw new ValidasiInputExc("Silahkan isi Judul buku");
    }

    if(pengarang.isEmpty()){
        throw new ValidasiInputExc("Silahkan isi Nama Pengarang");
    }

    if(!isbn.matches("\\d{13}")){
        throw new ValidasiInputExc("Nomor ISBN minimal 13 angka");
    }

    if(tahun.isEmpty()){
        throw new ValidasiInputExc("Silahkan isi Tahun Terbit");
    }

    if(!tahun.matches("\\d+")){
        throw new ValidasiInputExc("Tahun Terbit tidak boleh pakai huruf
gusy");
    }

    if(!tahun.matches("\\d{4}")){
        throw new ValidasiInputExc("Tahun Terbit harus 4 angka guys");
    }

    if(harga.isEmpty()){
        throw new ValidasiInputExc("Silahkan isi Harga Buku");
    }

    if(!harga.matches("\\d+")){
        throw new ValidasiInputExc("Harga kok pakai Huruf?!");
    }

    int hargaInt = Integer.parseInt(harga);
    if(hargaInt <= 0){
        throw new ValidasiInputExc("Harga kok Negatif??!!");
    }

    if(kategori == "--- Kategori Buku ---"){
        throw new ValidasiInputExc("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 String getTahun() {
        return tahun;
    }

    public String getHarga() {
        return harga;
    }

    public String getKategori() {
        return kategori;
    }
}

```

Membuat variabel kategori pada kelas Buku, kemudian inisialisasi variabel kategori pada konstruktor. Kemudian buat pesan validasi jika inputan kategori adalah default yaitu “--- Kategori Buku ---” maka muncul pesan “Kategori tidak boleh KOSONG!!!”. Terakhir buat fungsi getter getKategori yang akan digunakan pada kelas InputData untuk menambahkan dan menampilkan data Kategori Buku pada tabel ketika project dijalankan.

InputData.java:

```

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

    try{
        String judul = txtJudul.getText();
        String pengarang = txtPengarang.getText();
        String isbn = txtIsbn.getText();
        String tahun = txtTahun.getText();
        String harga = txtHarga.getText();
        String kategori = (String)cbxKategori.getSelectedItem();

        Buku buku = new Buku(judul, pengarang, isbn, tahun, harga,
        kategori);

        String formatHarga = "Rp." + buku.getHarga();

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

        txtJudul.setText("");
        txtPengarang.setText("");
        txtIsbn.setText("");
        txtTahun.setText("");
        lblKet.setText("Sukses! Data berhasil ditambahkan");
    }
}

```

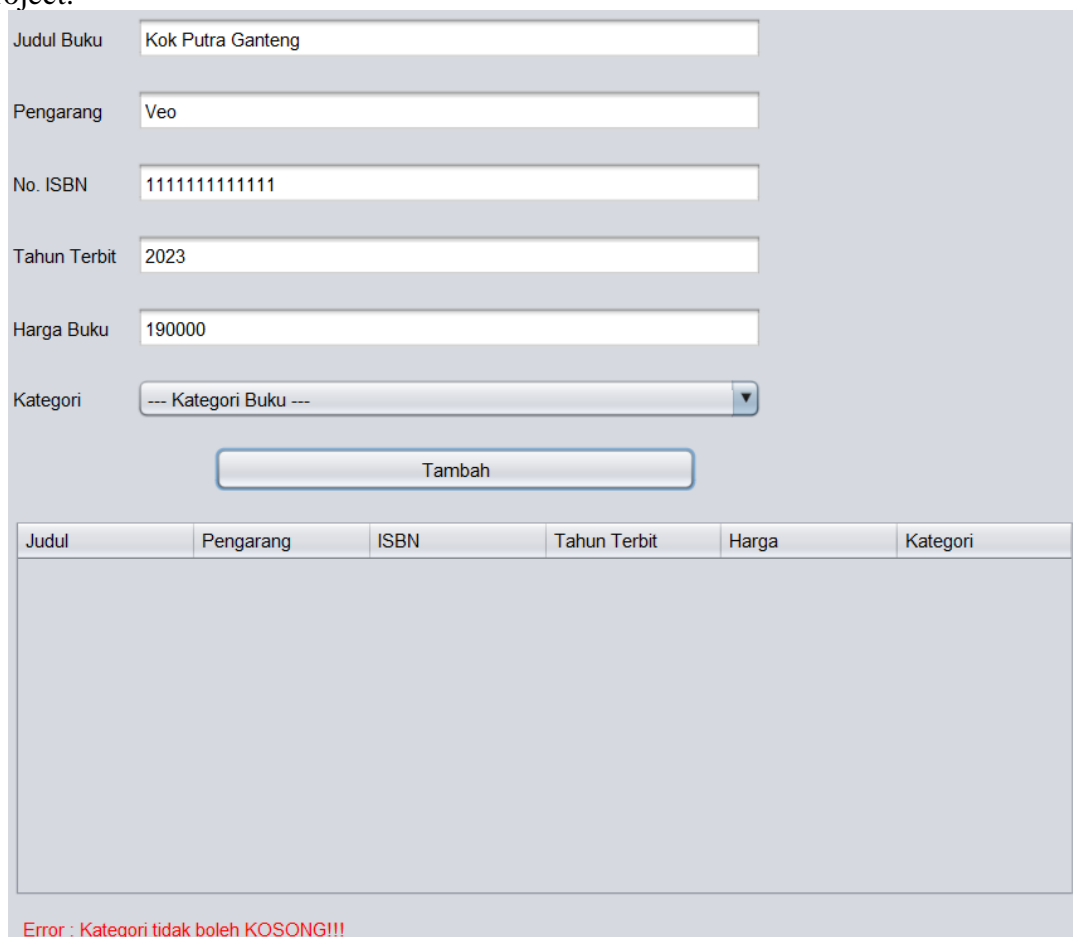
```

        lblKet.setForeground(Color.GREEN);
    } catch (ValidasiInputExc e) {
        lblKet.setText("Error : " + e.getMessage());
        lblKet.setForeground(Color.RED);
    }
}

```

Menambahkan variable kategori pada kode Button Submit yang dikonversi ke String dengan inputan item yang dipilih (getSelectedItem) kemudian menginisialisasikan variabel tersebut kedalam objek buku. Terakhir, panggil fungsi getKategori pada DefaultTableModel agar inputan (termasuk kategori) akan masuk dan ditampilkan kedalam tabel. Tetapi, karena kode tersebut ada didalam Exception Handling, maka akan ada validasi terlebih dahulu pada inputan, jika inputan salah (tidak sesuai) maka muncul pesan error pada label keterangan jika tidak maka data berhasil ditambahkan kedalam tabel.

Hasil Project:



The screenshot shows a Java Swing application window with a light gray background. It contains several input fields for book information:

- Judul Buku:** Text field containing "Kok Putra Ganteng".
- Pengarang:** Text field containing "Veo".
- No. ISBN:** Text field containing "111111111111".
- Tahun Terbit:** Text field containing "2023".
- Harga Buku:** Text field containing "190000".
- Kategori:** A dropdown menu currently showing "--- Kategori Buku ---".

Below the input fields is a button labeled "Tambah". Underneath the button is a table with the following columns: Judul, Pengarang, ISBN, Tahun Terbit, Harga, and Kategori. The table is currently empty. At the bottom of the window, there is a red error message: "Error : Kategori tidak boleh KOSONG!!!".

Judul	Pengarang	ISBN	Tahun Terbit	Harga	Kategori
Kok Putra Ganteng	Veo	1111111111111	2023	Rp.190000	Komik

Sukses! Data berhasil ditambahkan