

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 5
LATIHAN EXCEPTION HANDLING



Oleh :
Hafid Ramadhan
2311103104
SI07C

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

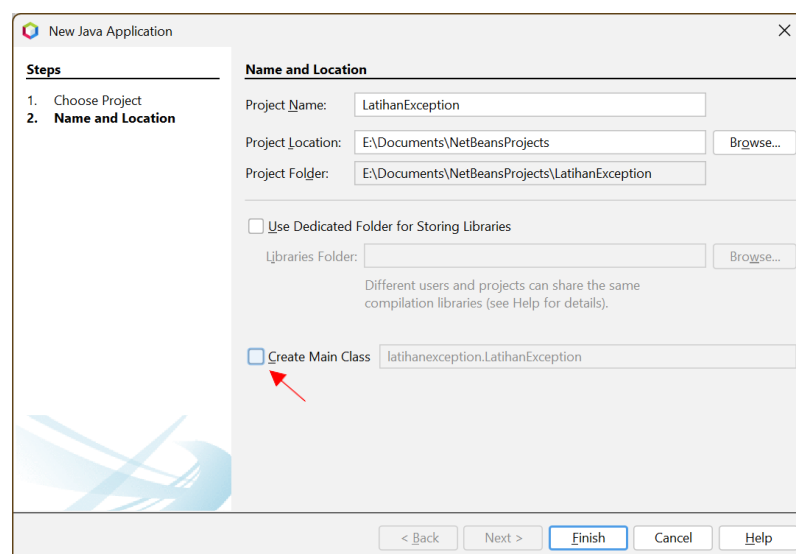
LATIHAN EXCEPTION HANDLING

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;  
    }  
}
```

```

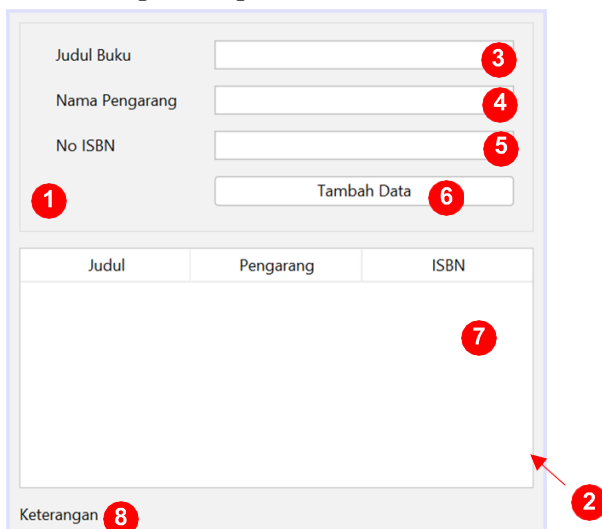
public String getJudul() {
    return judul;
}

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	JTextField	txtJudul
4	JTextField	txtPengarang
5	JTextField	txtIsbn
6	JButton	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) {

```

```

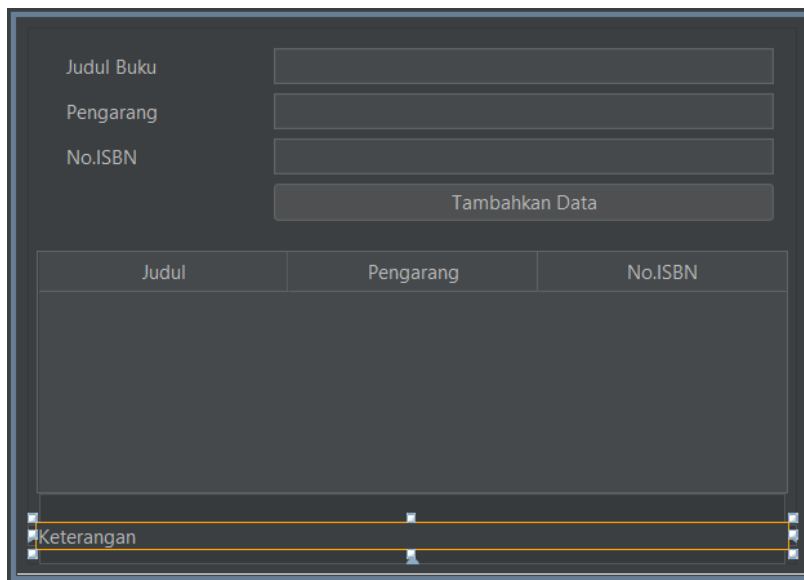
    lblKeterangan.setForeground(Color.RED);
    lblKeterangan.setText("Error: " + e.getMessage());
}

```

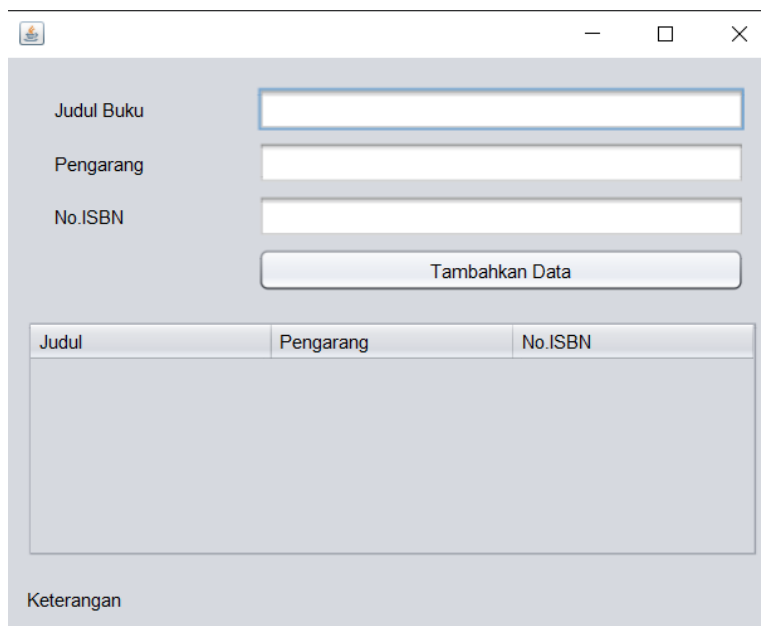
11. Jalankan project, coba tambahkan data baru. Ketika form kosong maka akan muncul peringatan pada bagian bawah aplikasi

➔ Penyelesaian:

Disini saya membuat tampilan terlebih dahulu, sebagai berikut:



- Tampilan halamannya:



C. Unguided

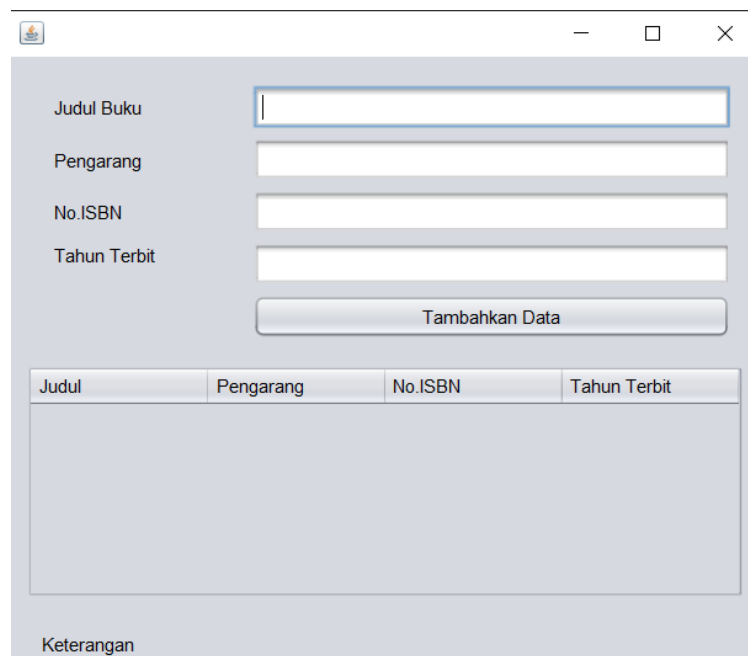
Tambahkan inputan berikut pada aplikasi:

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)

- **Penyelesaian:**

1. Tahun Terbit: Validasi untuk angka empat digit.

Langkah awal yang perlu dilakukan adalah menambahkan label dan input untuk "Tahun Terbit" pada desain program (menggunakan variabel textField: txtTahun). Selanjutnya, tambahkan kolom "Tahun Terbit" pada tabel agar data yang diinput dapat ditampilkan, seperti ilustrasi pada gambar berikut:



ValidasiInputException.java:

```
package latihanp10hafidramadhan;

/**
 * @author Hafid Ramadhan 2311103104
 */
public class ValidasiInputException extends Exception {

    public ValidasiInputException(String message) {
        super(message);
    }
}
```

Fungsi dari Kelas ini digunakan untuk Exception Handling pada project kali ini.

Buku.Java

```
package latihanp10hafidramadhan;

/**
 * @author Hafid Ramadhan 2311103104
 */
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 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 (tahun.isEmpty()) {
            throw new ValidasiInputException("Silahkan isi Tahun Terbit
            terlebih dahulu");
        }

        if (!tahun.matches("\\d+")) {
            throw new ValidasiInputException("tahun terbit only angka");
        }
        if (!tahun.matches("\\d{4}")) {
            throw new ValidasiInputException("Tahun Terbit wajib isi 4
            angka");
        }

        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;  
    }  
}
```

- **Penjelasan :**

Lakukan modifikasi pada kelas Buku dengan menambahkan atribut baru, yaitu tahun, yang diinisialisasi melalui konstruktor. Sertakan juga validasi dengan menggunakan exception handling untuk memastikan bahwa input dari pengguna sesuai dengan ketentuan yang ditetapkan. Selain itu, tambahkan metode getter getTahun() yang berfungsi untuk mengembalikan nilai atribut tahun. Metode ini akan digunakan dalam kelas InputData untuk menyisipkan serta menampilkan data Tahun Terbit pada tabel saat program dijalankan.

Tampilan.java

```
package latihanp10hafidramadhan;  
  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.swing.table.DefaultTableModel;  
  
/**  
 *  
 * @author Hafid Ramadhan 2311103104  
 */  
public class Babas extends javax.swing.JFrame {  
  
    /**  
     * Creates new form Babas  
     */  
    public Babas() {  
        initComponents();  
    }  
  
    /**  
     * This method is called from within the constructor to initialize the form.  
     * WARNING: Do NOT modify this code. The content of this method is always  
     * regenerated by the Form Editor.  
     */  
    @SuppressWarnings("unchecked")  
    // <editor-fold defaultstate="collapsed" desc="Generated Code">  
    private void initComponents() {  
  
        jPanel1 = new javax.swing.JPanel();  
        jLabel1 = new javax.swing.JLabel();  
        jLabel2 = new javax.swing.JLabel();  
        jLabel3 = new javax.swing.JLabel();  
        txtJudul = new javax.swing.JTextField();  
        txtPengarang = new javax.swing.JTextField();  
        Judul = new javax.swing.JScrollPane();  
        tabelBuku = new javax.swing.JTable();  
        txtIsbn = new javax.swing.JTextField();  
        jButton1 = new javax.swing.JButton();  
        jLabel4 = new javax.swing.JLabel();  
        txtTahun = new javax.swing.JTextField();  

```

```
lblMessage = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setText("Judul Buku");

jLabel2.setText("Pengarang");

jLabel3.setText("No.ISBN");

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

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

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

    },
    new String [] {
        "Judul", "Pengarang", "No.ISBN", "Tahun Terbit"
    }
));
Judul.setViewportView(tabelBuku);

jButton1.setText("Tambahkan Data");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jLabel4.setText("Tahun Terbit");

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

javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(24, 24, 24)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1)
                .addComponent(jLabel2)
                .addComponent(jLabel3)
                .addComponent(jLabel4)
                .addComponent(txtJudul)
                .addComponent(txtPengarang)
                .addComponent(txtTahun)
                .addComponent(jButton1)
            )
            .addGap(24, 24, 24)
        )
);
```

```

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
        .addGroup(jPanell1Layout.createSequentialGroup()
            .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 120, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 9,
Short.MAX_VALUE)
            .addComponent(txtJudul,
javax.swing.GroupLayout.PREFERRED_SIZE, 310,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanell1Layout.createSequentialGroup())

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
        .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 91, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE, 64, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE, 76, javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G, false)
        .addComponent(jButton1,
javax.swing.GroupLayout.DEFAULT_SIZE, 310, Short.MAX_VALUE)
        .addComponent(txtPengarang,
javax.swing.GroupLayout.DEFAULT_SIZE, 310, Short.MAX_VALUE)
        .addComponent(txtIsbn)
        .addComponent(txtTahun)))
        .addContainerGap(15, Short.MAX_VALUE))
    );
    jPanell1Layout.setVerticalGroup(

jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanell1Layout.createSequentialGroup()
            .addContainerGap(13, Short.MAX_VALUE)

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G, false)
            .addComponent(txtJudul)
            .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
NE)
            .addComponent(jLabel2)
            .addComponent(txtPengarang,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
NE)

```

```

        .addComponent(txtIsbn, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel3))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanellLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
        .addComponent(jLabel4)
        .addComponent(txtTahun, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton1)
        .addGap(18, 18, 18)
        .addComponent(Judul, javax.swing.GroupLayout.PREFERRED_SIZE, 151,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(16, 16, 16))
    );

Judul.getAccessibleContext().setAccessibleName("");
Judul.getAccessibleContext().setAccessibleDescription("");

lblMessage.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()
            .addContainerGap()
            .addComponent(jPanell, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .addGap(19, 19, 19)
            .addComponent(lblMessage, javax.swing.GroupLayout.PREFERRED_SIZE,
396, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanell, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
67, Short.MAX_VALUE)
            .addComponent(lblMessage)
            .addContainerGap())
    );

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

private void txtPengarangActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void txtJudulActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton1ActionPerformed(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)tabelBuku.getModel();
    data.addRow(new Object[] {buku.getJudul(),
buku.getPengarang(),buku.getIsbn(),buku.getTahun()});

    txtJudul.setText("");
    txtPengarang.setText("");
    txtIsbn.setText("");
} catch (ValidasiInputException e) {
    lblMessage.setText("error:"+ e.getMessage());
}
}

private void txtTahunActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<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(Babas.class.getName()).log(java.util.logging.Level
1.SEVERE, null, ex);
    } catch (InstantiationException ex) {

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

```

```

java.util.logging.Logger.getLogger(Babas.class.getName()).log(java.util.logging.Level
1.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Babas().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JScrollPane Judul;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel1;
private javax.swing.JLabel lblMessage;
private javax.swing.JTable tabelBuku;
private javax.swing.JTextField txtIsbn;
private javax.swing.JTextField txtJudul;
private javax.swing.JTextField txtPengarang;
private javax.swing.JTextField txtTahun;
// End of variables declaration
}

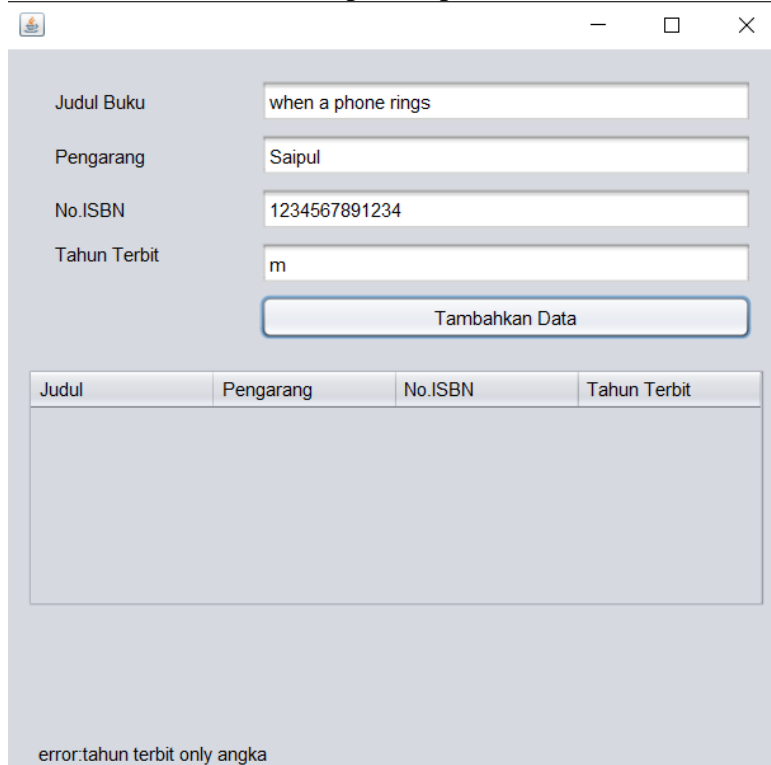
```

• Penjelasan:

Pada variabel tombol btnSubmit, dilakukan penambahan variabel tahun yang nilainya diambil dari teks input pengguna melalui metode getText. Nilai tahun ini kemudian dimasukkan ke dalam objek buku yang telah dibuat sebelumnya. Selanjutnya, kode buku.getTahun() ditambahkan ke dalam DefaultTableModel, sehingga data input, termasuk tahun, dapat dimasukkan dan ditampilkan pada tabel. Karena proses ini berada dalam blok Exception Handling, validasi dilakukan terlebih dahulu terhadap input pengguna. Jika input tidak valid, seperti kosong, mengandung huruf, atau tidak berupa angka dengan 4 digit, akan muncul pesan kesalahan pada label keterangan. Sebaliknya, jika input valid, data akan ditambahkan ke tabel tanpa hambatan. bertujuan untuk memastikan bahwa data yang dimasukkan oleh pengguna sesuai dengan format yang diharapkan, menghindari error dalam program, dan menjaga keakuratan informasi pada tabel. Dengan menggunakan DefaultTableModel, data dapat ditampilkan secara dinamis dan langsung diperbarui di antarmuka aplikasi. Pesan kesalahan juga membantu memberikan umpan balik yang jelas kepada pengguna untuk memperbaiki input mereka.

Hasil Program :

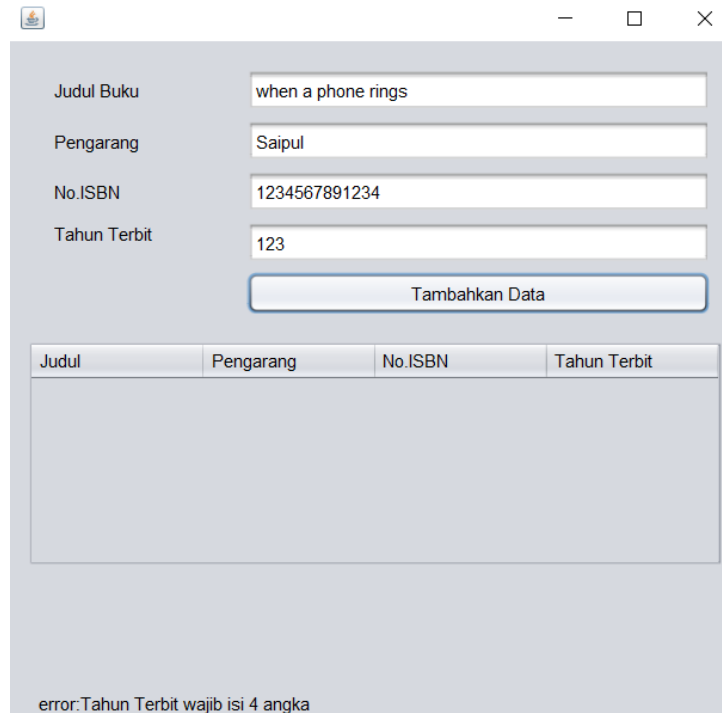
- Tampilan saat ketika memasukkan huruf pada inputan Tahun terbit



The screenshot shows a web form for adding book data. The fields are: Judul Buku (when a phone rings), Pengarang (Saipul), No.ISBN (1234567891234), and Tahun Terbit (m). A 'Tambahkan Data' button is below the fields. Below the button is a table with headers: Judul, Pengarang, No.ISBN, and Tahun Terbit. The table body is empty. At the bottom, an error message reads: 'error:tahun terbit only angka'.

Judul	Pengarang	No.ISBN	Tahun Terbit
-------	-----------	---------	--------------

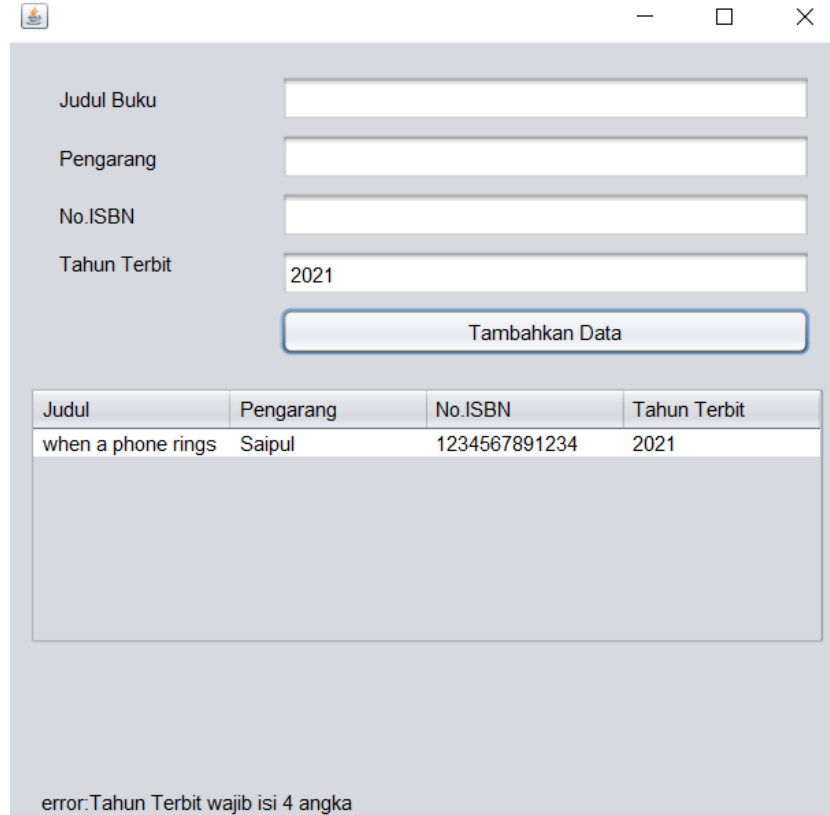
- Tampilan saat ketika memasukkan angka >3 pada inputan Tahun terbit



The screenshot shows the same web form as above, but with the 'Tahun Terbit' field containing the value '123'. The 'Tambahkan Data' button is still present. The error message at the bottom now reads: 'error:Tahun Terbit wajib isi 4 angka'.

Judul	Pengarang	No.ISBN	Tahun Terbit
-------	-----------	---------	--------------

- Tampilan Inputan Yang benar



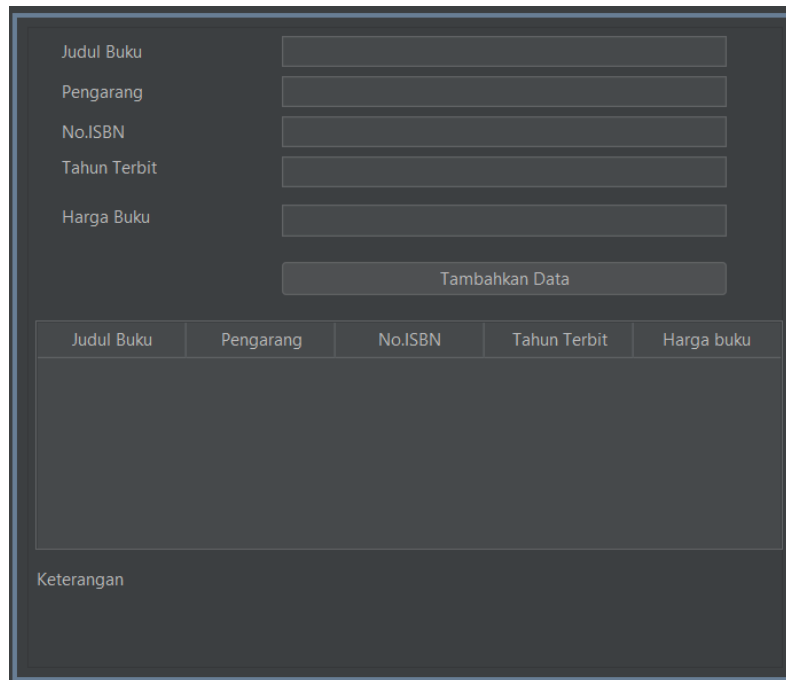
The screenshot shows a web application window with a title bar containing a small icon and standard window controls (minimize, maximize, close). The form contains the following elements:

- Input fields for "Judul Buku", "Pengarang", "No.ISBN", and "Tahun Terbit".
- The "Tahun Terbit" field contains the value "2021".
- A button labeled "Tambahkan Data" is positioned below the input fields.
- A table below the button displays the entered data:

Judul	Pengarang	No.ISBN	Tahun Terbit
when a phone rings	Saipul	1234567891234	2021
- An error message at the bottom reads: "error:Tahun Terbit wajib isi 4 angka".

2. Harga Buku: Validasi untuk angka positif.

Penyelesaian:



The screenshot shows a web application window with a title bar. The form contains the following elements:

- Input fields for "Judul Buku", "Pengarang", "No.ISBN", "Tahun Terbit", and "Harga Buku".
- A button labeled "Tambahkan Data" is positioned below the input fields.
- A table below the button displays the entered data:

Judul Buku	Pengarang	No.ISBN	Tahun Terbit	Harga buku
- A label "Keterangan" is located below the table.

Langkah pertama yang harus dilakukan adalah menambahkan label dan field input untuk harga pada desain program, menggunakan variabel text field yang diberi nama txtHarga. Selanjutnya, tambahkan kolom baru dengan nama "Harga" pada tabel agar data yang dimasukkan dapat ditampilkan dengan jelas.

Buku.Java

```
package latihanp10Hafidramadhan;

/**
 *
 * @author Hafid Ramadhan 2311103104
 *
 */
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 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 (tahun.isEmpty()) {
            throw new ValidasiInputException("Silahkan isi Tahun Terbit");
        }

        if (!tahun.matches("\\d+")) {
            throw new ValidasiInputException("Tahun Terbit only angka");
        }
        if (!tahun.matches("\\d{4}")) {
            throw new ValidasiInputException("Tahun Terbit wajib isi 4 angka ");
        }
        if (harga.isEmpty()) {
            throw new ValidasiInputException("di isi terlebih dahulu Harga Buku");
        }
        int hargaInt = Integer.parseInt(harga);
        if (hargaInt <= 0) {
            throw new ValidasiInputException("Masukkan Harga bukan utang anda!!");
        }
        if (!harga.matches("\\d+")) {
            throw new ValidasiInputException("Harga wajib angka");
        }

        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;
}
}
```

Penjelasan:

Pada kelas Buku, terdapat beberapa atribut seperti judul, pengarang, isbn, tahun, dan harga, yang semuanya bertipe data String dan digunakan untuk menyimpan informasi tentang buku. Atribut-atribut ini diinisialisasi melalui konstruktor, yang juga melakukan proses validasi input untuk memastikan data yang dimasukkan sesuai dengan aturan yang ditetapkan. Pada atribut judul, validasi dilakukan untuk memastikan bahwa nilai tidak kosong. Jika kosong, akan muncul pesan “judul kosong”. Hal serupa diterapkan pada atribut pengarang, dengan pesan kesalahan “Pengarang kosong”. Untuk atribut isbn, dilakukan validasi bahwa nilai harus berupa angka 13 digit, jika tidak, akan muncul pesan “Isbn harus 13 angka”. Atribut tahun divalidasi agar tidak kosong, hanya berupa angka, dan terdiri dari tepat 4 digit. Apabila tidak memenuhi, pesan seperti “Silahkan isi Tahun Terbit”, “Tahun Terbit only angka”, atau “Tahun Terbit wajib isi 4 angka” akan ditampilkan.

Sedangkan untuk atribut harga, validasi memastikan bahwa input tidak kosong, berupa angka, dan lebih besar dari nol. Jika input kosong, akan muncul pesan “di isi terlebih dahulu Harga Buku”. Input yang tidak berupa angka akan menghasilkan pesan “Harga wajib angka”, dan nilai negatif akan memunculkan pesan “Masukkan Harga bukan utang anda!!”. Nilai harga dikonversi dari String ke Integer menggunakan Integer.parseInt untuk melakukan validasi terhadap nilainya. Selain itu, untuk setiap atribut, disediakan metode getter seperti getJudul, getPengarang, getIsbn, getTahun, dan getHarga, yang memungkinkan pengambilan nilai atribut tersebut untuk digunakan pada kelas lain, seperti menampilkan informasi dalam tabel pada saat program dijalankan.

Tampilan.java

```
private void jButton1ActionPerformed(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) tabelBuku.getModel();
        data.addRow(new Object[]{buku.getJudul(), buku.getPengarang(),
            buku.getIsbn(), buku.getTahun(), formatHarga});

        txtJudul.setText("");
        txtPengarang.setText("");
        txtIsbn.setText("");
        txtTahun.setText("");
        lblMessage.setText(" Data sukses ditambahkan");
        lblMessage.setForeground(Color.GREEN);
    } catch (ValidasiInputException e) {
        lblMessage.setText("Gagal : " + e.getMessage());
        lblMessage.setForeground(Color.RED);
    }
}
```

- **Penjelasan:**

Pada kode tombol Submit, nilai harga diambil dari inputan pengguna menggunakan metode `getText`, lalu digunakan untuk menginisialisasi sebuah objek dari kelas `Buku`. Untuk meningkatkan tampilan, dibuat variabel tambahan bernama `formatHarga`, yang secara otomatis menambahkan prefix “Rp.” di depan nilai harga yang dimasukkan pengguna. Variabel ini kemudian digunakan pada objek `DefaultTableModel` untuk menampilkan data, termasuk harga dengan format yang lebih menarik, di dalam tabel. Kode ini berada di dalam blok `Exception Handling`, sehingga validasi dilakukan terlebih dahulu untuk memastikan inputan sesuai. Jika inputan tidak valid, pesan error akan ditampilkan pada sebuah label keterangan untuk memberi tahu pengguna tentang kesalahan. Sebaliknya, jika semua input valid, data akan berhasil disimpan dan ditampilkan di tabel dengan format harga yang lebih informatif.

- Hasil Program:

➔ Ketika menginputkan inputan minus :

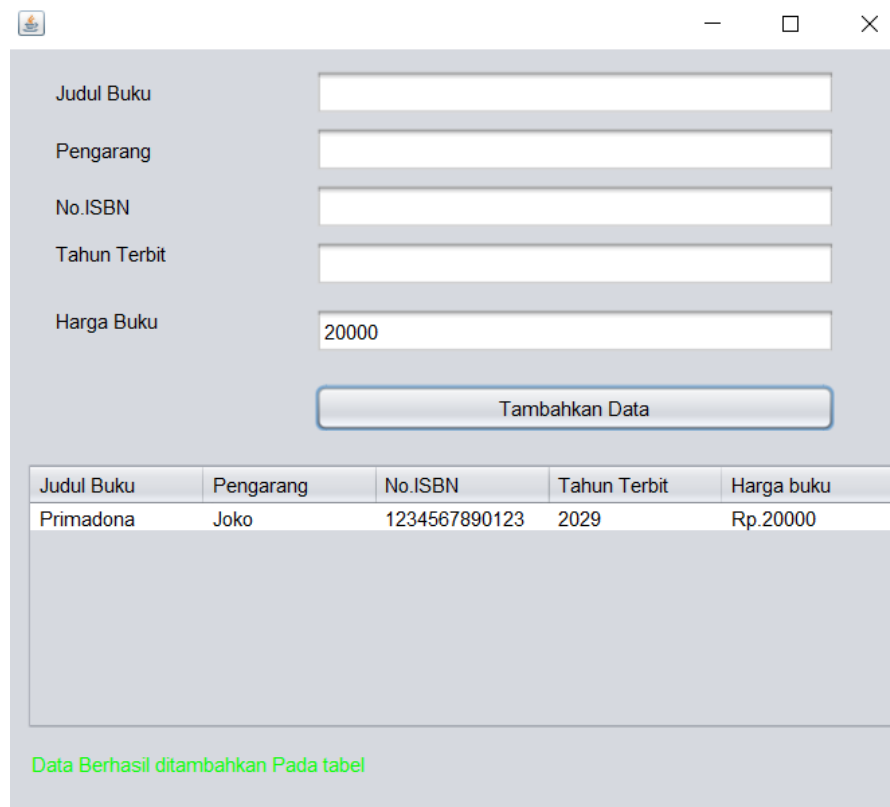


The screenshot shows a web application window with a form for adding book data. The form fields are: Judul Buku (duisahf), Pengarang (dbqwdbu), No.ISBN (1234567890123), Tahun Terbit (1234), and Harga Buku (-2000). Below the form is a button labeled 'Tambahkan Data'. Underneath the button is a table with the following data:

Judul Buku	Pengarang	No.ISBN	Tahun Terbit	Harga buku
Primadona	Joko	1234567890123	2029	Rp.20000

Below the table, a red error message is displayed: "Gagal : Masukkan Harga bukan utang anda!!".

➔ Hasil Program Yang benar :



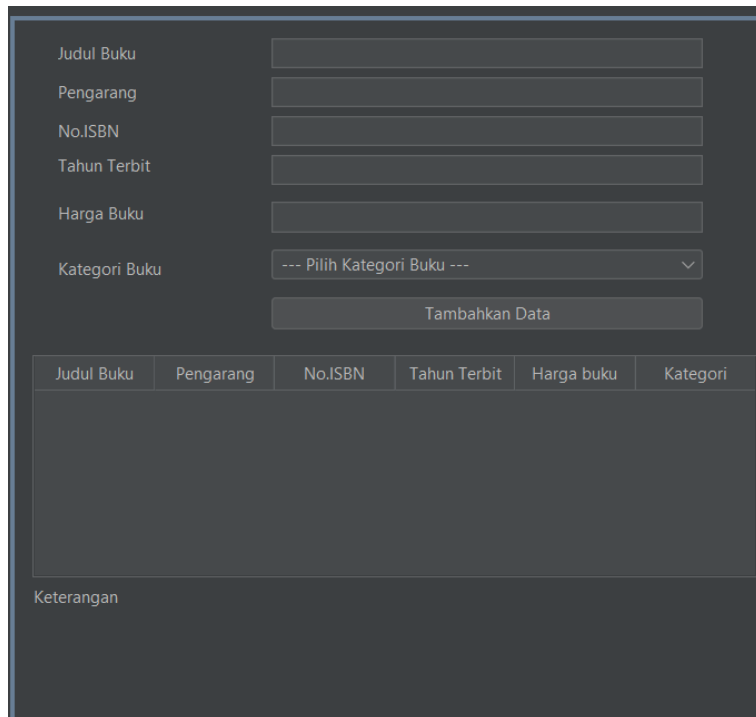
The screenshot shows the same web application window, but the 'Harga Buku' field now contains '20000' instead of '-20000'. The 'Tambahkan Data' button is still present. Below the button, the table from the previous screenshot is shown, but it now includes the new entry:

Judul Buku	Pengarang	No.ISBN	Tahun Terbit	Harga buku
Primadona	Joko	1234567890123	2029	Rp.20000

Below the table, a green success message is displayed: "Data Berhasil ditambahkan Pada tabel".

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

JAWAB:



Langkah pertama yang perlu dilakukan adalah menambahkan label dan komponen input untuk kategori (menggunakan JComboBox) pada desain program (dengan variabel ComboBox: txtKategori). Selanjutnya, tambahkan kolom "Kategori" pada tabel untuk menampilkan data yang dimasukkan melalui input tersebut.

Buku.java

```
package latihanp10Hafidramadhan;

/**
 * @author Hafid Ramadhan 2311103104
 */
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 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 (tahun.isEmpty()) {
            throw new ValidasiInputException("Silahkan isi Tahun Terbit");
        }

        if (!tahun.matches("\\d+")) {
            throw new ValidasiInputException("Tahun Terbit only angka");
        }
        if (!tahun.matches("\\d{4}")) {
            throw new ValidasiInputException("Tahun Terbit wajib isi 4 angka ");
        }
        if (harga.isEmpty()) {
            throw new ValidasiInputException("di isi terlebih dahulu Harga Buku");
        }
        int hargaInt = Integer.parseInt(harga);
        if (hargaInt <= 0) {
            throw new ValidasiInputException("Masukkan Harga bukan utang anda!!");
        }
        if (!harga.matches("\\d+")) {
            throw new ValidasiInputException("Harga wajib angka");
        }
        if (kategori == "--- Masukkan Kategori Buku ---") {
            throw new ValidasiInputException("Masukkan Kategori Dulu jangan buru-
buru");
        }
        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;
    }
}

```

Untuk implementasi variabel kategori dalam kelas Buku, pertama-tama, buatlah variabel kategori yang akan diinisialisasi dalam konstruktor. Kategori buku ini akan disertakan sebagai parameter saat pembuatan objek Buku. Dalam hal ini, kategori buku bisa memiliki nilai default, yaitu "--- Masukkan Kategori Buku ---", yang menandakan bahwa kategori belum dipilih. Kemudian, jika nilai kategori masih dalam keadaan default tersebut, sistem harus menampilkan pesan validasi yang mengingatkan pengguna untuk memilih kategori, dengan pesan "Masukkan kategori dulu jangan buru-buru!!". Selain itu, buatlah fungsi getter `getKategori` dalam kelas Buku yang berfungsi untuk mengakses kategori yang telah dipilih. Fungsi ini sangat berguna, terutama dalam kelas lain seperti `InputData`, yang akan menggunakan metode ini untuk menambahkan dan menampilkan data kategori buku ke dalam tabel saat proyek dijalankan. Dengan cara ini, kategori buku yang dipilih bisa dikelola dengan baik dan memastikan bahwa input pengguna valid sebelum data ditampilkan.

Tampilan.java

```
private void jButton1ActionPerformed(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)txtKategori.getSelectedItem();

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

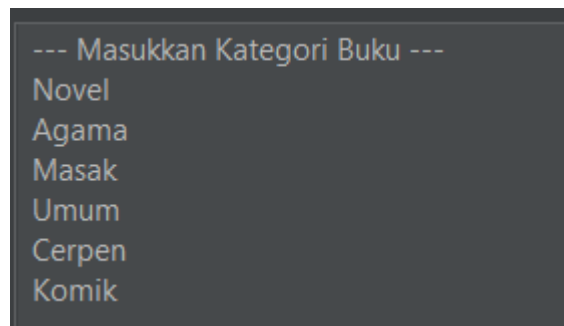
        String formatHarga = "Rp." + buku.getHarga();

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

        txtJudul.setText("");
        txtPengarang.setText("");
        txtIsbn.setText("");
        txtTahun.setText("");
        lblMessage.setText(" Data sukses ditambahkan");
        lblMessage.setForeground(Color.GREEN);
    } catch (ValidasiInputException e) {
        lblMessage.setText("Gagal : " + e.getMessage());
        lblMessage.setForeground(Color.RED);
    }
}
```

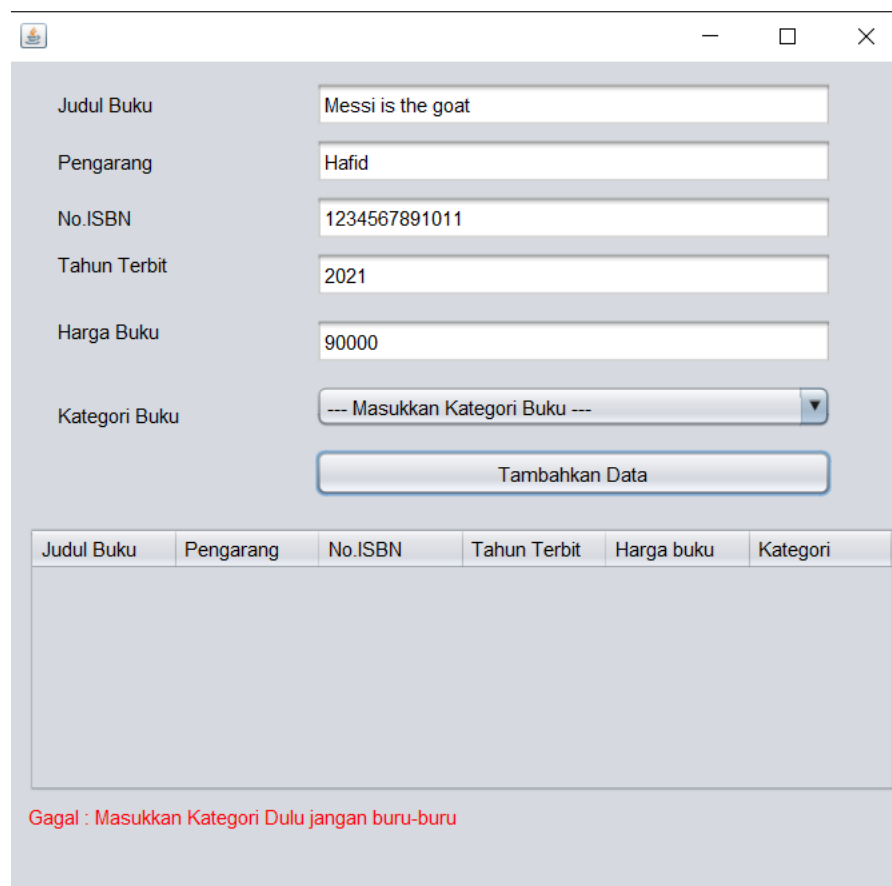
Pada kode tombol Submit, ditambahkan variabel kategori yang nilainya diperoleh berdasarkan item yang dipilih oleh pengguna menggunakan metode `getSelectedItem`. Variabel ini kemudian dikonversi menjadi tipe `String` dan diinisialisasikan ke dalam objek dari kelas `Buku`. Selanjutnya, fungsi `getKategori` dari objek `Buku` dipanggil pada `DefaultTableModel`, sehingga inputan kategori, bersama data lainnya, dapat ditampilkan di tabel. Karena kode ini berada di dalam blok `Exception Handling`, proses validasi dilakukan untuk memastikan semua input sesuai dengan aturan yang ditentukan. Jika input tidak valid—seperti salah memilih atau tidak mengisi kategori—pesan error akan ditampilkan pada label keterangan untuk memberi tahu pengguna tentang kesalahan tersebut. Sebaliknya, jika semua input valid, data, termasuk kategori yang dipilih, akan berhasil ditambahkan ke dalam tabel dengan format yang sesuai

Disini saya menambahkan mengisi nilai JComboBox seperti dibawah ini:



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

- Tampilan Saat Belum menginputkan kategori Buku



Judul Buku: Messi is the goat

Pengarang: Hafid

No.ISBN: 1234567891011

Tahun Terbit: 2021

Harga Buku: 90000

Kategori Buku: --- Masukkan Kategori Buku ---

Tambahkan Data

Judul Buku	Pengarang	No.ISBN	Tahun Terbit	Harga buku	Kategori
------------	-----------	---------	--------------	------------	----------

Gagal : Masukkan Kategori Dulu jangan buru-buru

- Tampilan Saat sudah memilih kategori

Judul Buku

Pengarang

No.ISBN

Tahun Terbit

Harga Buku

20000

Kategori Buku

Umum

Tambahkan Data

Judul Buku	Pengarang	No.ISBN	Tahun Terbit	Harga buku	Kategori
messi is goat	Hafid	12345678910...	2025	Rp.20000	Umum

Data sukses ditambahkan