

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL X
MATERI

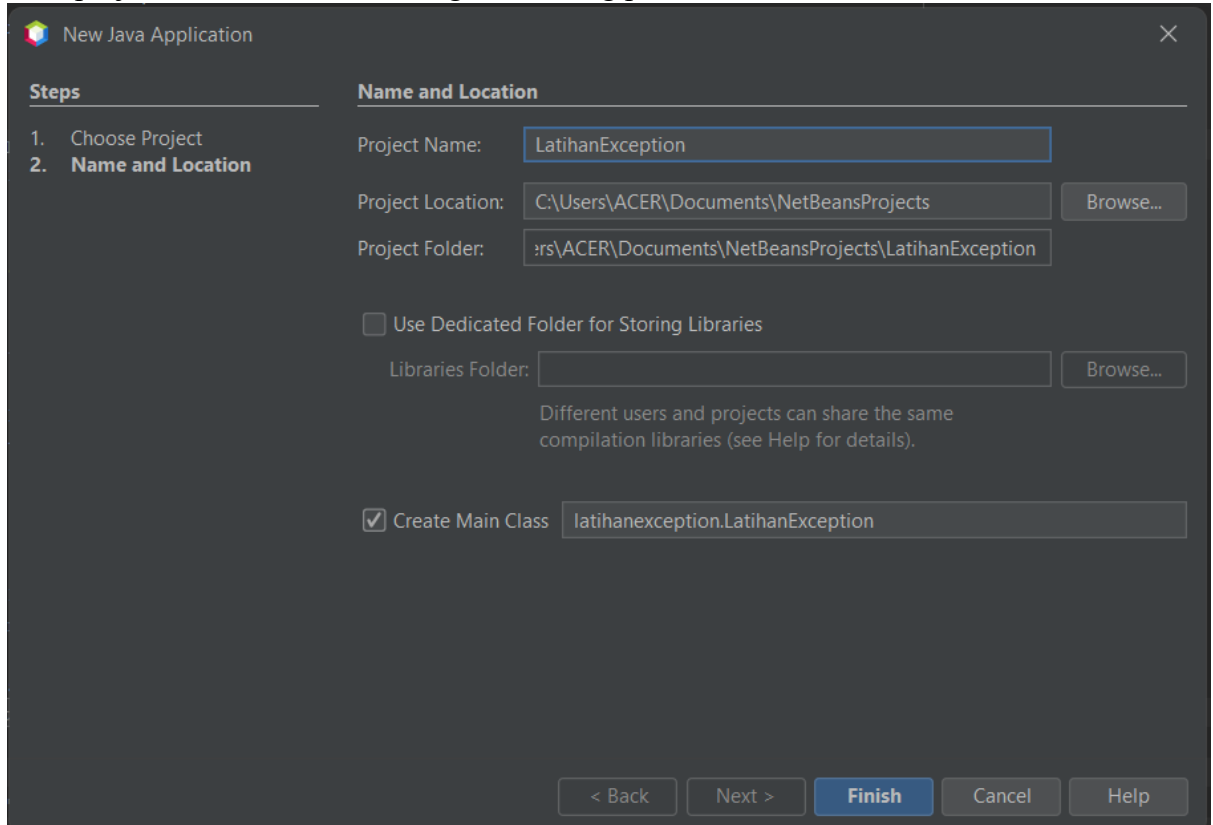


Oleh :
Muhammad Raflan Kemal
2311103132
07C

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

I. GUIDED

1. Buat project baru, setelah itu hilangkan centang pada main class



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class

< Back Next > **Finish** Cancel Help

2. Buat kelas Buku, InputData dan ValidInputException

Buku.java

```
package perpustakaan;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */
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 Tidak Boleh Kosong");
        }

        if(pengarang.isEmpty()){
```

```

        throw new ValidasiInputException("Pengarang Tidak Boleh Kosong");
    }

    if(isbn.matches("\\d{13}")){
        throw new ValidasiInputException("ISBN Harus 13 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;
}
}

```

Kelas Buku dalam program ini berfungsi untuk mewakili data sebuah buku, mencakup informasi seperti judul, pengarang, dan nomor ISBN. Saat sebuah objek Buku dibuat, program memeriksa apakah judul dan pengarang telah diisi, serta memastikan ISBN terdiri dari tepat 13 angka. Jika ada data yang tidak sesuai, program akan memberikan pesan kesalahan menggunakan `ValidasiInputException`, sehingga pengguna dapat memperbaikinya. Selain itu, kelas ini menyediakan metode sederhana untuk mengambil data judul, pengarang, dan ISBN, membuatnya lebih mudah digunakan dalam aplikasi. Desain ini membantu menjaga kualitas data buku dan memberikan pengalaman yang lebih aman bagi pengguna.

InputData.java

```

package perpustakaan;

import java.awt.Color;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */

```

```

public class InputData extends javax.swing.JFrame {

    /**
     * Creates new form InputData
     */
    public InputData() {
        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();
        txtIsbn = new javax.swing.JTextField();
        btnSubmit = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        tableBuku = new javax.swing.JTable();
        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);
            }
        });

        txtIsbn.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtIsbnActionPerformed(evt);
        }
    });

    btnSubmit.setText("Tambah Data");
    btnSubmit.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnSubmitActionPerformed(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(29, 29, 29)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(jLabel1,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel2,
                    javax.swing.GroupLayout.DEFAULT_SIZE, 64, Short.MAX_VALUE)
                .addComponent(jLabel3,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(btnSubmit,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 235,
                    javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(txtJudul,
                        javax.swing.GroupLayout.DEFAULT_SIZE, 235, Short.MAX_VALUE)
                    .addComponent(txtPengarang)

```

```

        .addComponent(txtIsbn)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
NG)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(11, 11, 11)

        .addGroup(jPanel1Layout.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))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(jPanel1Layout.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(9, 9, 9)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel3)
            .addComponent(txtIsbn,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(btnSubmit)
            .addContainerGap(13, Short.MAX_VALUE))
    );

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

```

```

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

lblMessage.setText("Keterangan");

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

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(jPanel1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(layout.createParallelGroup()
            .addComponent(lblMessage,
javax.swing.GroupLayout.PREFERRED_SIZE, 303,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
Short.MAX_VALUE))
        );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createParallelGroup()
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(lblMessage)
            .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
Short.MAX_VALUE))
        );

```

```

    );

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

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

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

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

        if (judul.isEmpty()) {
            throw new ValidasiInputException("Judul buku tidak boleh kosong.");
        }
        if (pengarang.isEmpty()) {
            throw new ValidasiInputException("Nama pengarang tidak boleh kosong.");
        }
        if (isbn.isEmpty()) {
            throw new ValidasiInputException("ISBN tidak boleh kosong.");
        }

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

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

        txtJudul.setText("");
        txtPengarang.setText("");
        txtIsbn.setText("");

        lblMessage.setText("Buku berhasil ditambahkan.");
    } catch (ValidasiInputException e) {
        lblMessage.setText("Error: " + e.getMessage());
    } catch (Exception e) {
        lblMessage.setText("Error tak terduga: " + e.getMessage());
    } lblMessage.setForeground(Color.red);
    }
}

```



```

/**
 * @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\(InputData.class.getName\(\)\).log\(java.util.loggi
ng.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

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

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

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

    /* Create and display the form */
    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.JPanel jPanel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JLabel lblMessage;
    private javax.swing.JTable tableBuku;
    private javax.swing.JTextField txtIsbn;
    private javax.swing.JTextField txtJudul;
    private javax.swing.JTextField txtPengarang;
    // End of variables declaration
}

```

Kelas InputData dirancang untuk mempermudah pengguna dalam menginput dan mengelola data buku. Aplikasi ini memiliki form sederhana yang terdiri dari tiga input utama, yaitu Judul Buku, Pengarang, dan ISBN, serta tombol Tambah Data untuk menambahkan data ke tabel. Ketika tombol ditekan, sistem akan memvalidasi input untuk memastikan semua data terisi dan ISBN memiliki 13 angka. Jika validasi gagal, pesan kesalahan akan ditampilkan di label khusus, sedangkan data yang valid akan langsung ditambahkan ke tabel, dan input form akan otomatis dikosongkan untuk entri berikutnya. Dengan umpan balik yang jelas melalui pesan kesalahan atau keberhasilan berwarna, aplikasi ini menawarkan solusi praktis untuk mengelola daftar buku secara efisien, baik untuk perpustakaan kecil maupun penggunaan pribadi.

ValidasiInputException.java

```

package perpustakaan;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */
public class ValidasiInputException extends Exception {
    public ValidasiInputException(String message){
        super (message);
    }

    @Override
    public String toString() {
        return "Terjadi kesalahan pada validasi input: " + getMessage();
    }
}

```

```
}
```

Kelas `ValidasiInputException` adalah kelas khusus yang dirancang untuk menangani kesalahan validasi input dalam aplikasi. Kelas ini merupakan turunan dari kelas `Exception` bawaan Java, sehingga memungkinkan kita untuk melempar dan menangkap kesalahan saat data yang dimasukkan tidak sesuai dengan aturan. Ketika kesalahan terjadi, kita dapat memberikan pesan yang spesifik melalui konstruktor, dan pesan ini akan diteruskan ke mekanisme penanganan kesalahan bawaan. Selain itu, kelas ini juga mengubah cara pesan ditampilkan dengan mengoverride metode `toString()`, sehingga pesan yang dihasilkan lebih jelas, menyebutkan bahwa kesalahan terkait dengan validasi input, diikuti oleh detailnya. Hal ini membuat kelas ini sangat berguna untuk memberikan umpan balik yang informatif dan membantu dalam perbaikan kesalahan.

Judul Buku

Pengarang

No ISBN

Tambah Data

Judul	Pengarang	ISBN

Keterangan

— □ ×

Judul Buku

Menabur Api Menuai Abu

Pengarang

Rafcuy

No ISBN

Tambah Data

Judul	Pengarang	ISBN
-------	-----------	------

Error: ISBN tidak boleh kosong.

— □ ×

Judul Buku

Pengarang

No ISBN

Tambah Data

Judul	Pengarang	ISBN
Menanam Api Menuai Abu	Rafcuy	236482928

Buku berhasil ditambahkan.

II. UNGUIDED

1. Tahun Terbit : Validasi untuk angka empat digit
Buku.java

```
package perpustakaan;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */
public class Buku {
    private String judul;
    private String pengarang;
    private String isbn;
    private String tahunTerbit;

    public Buku(String judul, String pengarang, String isbn, String tahunTerbit)
throws ValidasiInputException {
        validasiInput(judul, pengarang, isbn, tahunTerbit);
        this.judul = judul;
        this.pengarang = pengarang;
        this.isbn = isbn;
        this.tahunTerbit = tahunTerbit;
    }

    private void validasiInput(String judul, String pengarang, String isbn, String
tahunTerbit) throws ValidasiInputException {
        if (judul == null || judul.trim().isEmpty()) {
            throw new ValidasiInputException("Judul tidak boleh kosong");
        }
        if (pengarang == null || pengarang.trim().isEmpty()) {
            throw new ValidasiInputException("Pengarang tidak boleh kosong");
        }
        if (isbn == null || isbn.trim().isEmpty()) {
            throw new ValidasiInputException("ISBN tidak boleh kosong");
        }
        if (tahunTerbit == null || tahunTerbit.trim().isEmpty()) {
            throw new ValidasiInputException("Tahun terbit tidak boleh kosong");
        }

        try {
            int tahun = Integer.parseInt(tahunTerbit);
            if (tahun < 1000 || tahun > 9999) {
                throw new ValidasiInputException("Tahun terbit tidak valid");
            }
        }
    }
}
```

```

        } catch (NumberFormatException e) {
            throw new ValidasiInputException("Tahun terbit harus berupa angka");
        }
    }

    public String getJudul() {
        return judul;
    }

    public String getPengarang() {
        return pengarang;
    }

    public String getIsbn() {
        return isbn;
    }

    public String getTahunTerbit() {
        return tahunTerbit;
    }
}

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

```

Class Buku dalam package perpustakaan adalah representasi objek buku yang memiliki atribut judul, pengarang, isbn, dan tahunTerbit. Konstruktor dari class ini melakukan validasi terhadap input untuk memastikan bahwa data tidak kosong, ISBN terisi, dan tahun terbit berupa angka dengan format yang valid (4 digit). Jika validasi gagal, maka akan dilemparkan exception ValidasiInputException. Selain itu, class ini menyediakan getter untuk mengakses atributnya. Class ini dirancang untuk memastikan data buku yang dimasukkan ke dalam aplikasi perpustakaan memiliki kualitas dan format yang benar, mencegah error akibat input tidak valid.

InputData.java

```

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

        String Judul = txtJudul.getText().trim();
        String Pengarang = txtPengarang.getText().trim();
        String Isbn = txtIsbn.getText().trim();
        String TahunTerbit = txtTahunTerbit.getText().trim();
    }
}

```

```

        if (Judul.isEmpty() || Pengarang.isEmpty() || Isbn.isEmpty() ||
TahunTerbit.isEmpty()) {
            lblMessage.setForeground(Color.RED);
            lblMessage.setText("Semua field harus diisi!");
            return;
        }

        Buku buku = new Buku(Judul, Pengarang, Isbn, TahunTerbit);

        DefaultTableModel model = (DefaultTableModel) Tabel.getModel();
        model.addRow(new Object[] {buku.getJudul(), buku.getPengarang(),
buku.getIsbn(), buku.getTahunTerbit()});

        txtJudul.setText("");
        txtPengarang.setText("");
        txtIsbn.setText("");
        txttahunTerbit.setText("");

        lblMessage.setForeground(Color.BLACK);
        lblMessage.setText("Buku berhasil ditambahkan.");

    } catch (ValidasiInputException e) {
        lblMessage.setForeground(Color.RED);
        lblMessage.setText("Error: " + e.getMessage());
    } catch (Exception e) {
        lblMessage.setForeground(Color.RED);
        lblMessage.setText("Error tak terduga: " + e.getMessage());
    }
}

```

Kode ini mengambil input tahun terbit dari txtTahunTerbit.getText() dan menggunakannya untuk membuat objek Buku. Data seperti judul, pengarang, ISBN, dan tahun terbit kemudian ditambahkan ke tabel menggunakan DefaultTableModel, dengan kolom tahun terbit ikut ditampilkan. Setelah data berhasil ditambahkan, semua kolom input otomatis dikosongkan agar pengguna bisa langsung memasukkan data baru. Label lblKeterangan memberikan umpan balik berupa pesan sukses berwarna hijau atau pesan kesalahan berwarna merah jika ada masalah, sehingga pengguna mendapat informasi jelas tentang status data yang dimasukkan.

Judul

Ashel Cantik

Pengarang

Kemal Ganteng

No ISBN

9821028371

Tahun Terbit

11

Tambah Data

Judul	Pengarang	ISBN	Tahun Terbit

Error: Tahun terbit tidak valid

Judul

Pengarang

No ISBN

Tahun Terbit

Tambah Data

Judul	Pengarang	ISBN	Tahun Terbit
Ashel Cantik	Kemal Ganteng	9821028371	1955

Buku berhasil ditambahkan.

2. Harga Buku : Validasi untuk angka positif Buku.java

```
package perpustakaan;
```

```
/**
```

```
 *
```



```

* @author Muhammad Raflan Kemal
* 2311103132
* 07C
*/
public class Buku {
    private String judul;
    private String pengarang;
    private String isbn;
    private String tahunTerbit;
    private String harga;

    public Buku(String judul, String pengarang, String isbn, String tahunTerbit,
String harga) throws ValidasiInputException {
        validasiInput(judul, pengarang, isbn, tahunTerbit, harga);
        this.judul = judul;
        this.pengarang = pengarang;
        this.isbn = isbn;
        this.tahunTerbit = tahunTerbit;
        this.harga = harga;
    }

    private void validasiInput(String judul, String pengarang, String isbn, String
tahunTerbit, String harga) throws ValidasiInputException {
        if (judul == null || judul.trim().isEmpty()) {
            throw new ValidasiInputException("Judul tidak boleh kosong");
        }
        if (pengarang == null || pengarang.trim().isEmpty()) {
            throw new ValidasiInputException("Pengarang tidak boleh kosong");
        }
        if (isbn == null || isbn.trim().isEmpty()) {
            throw new ValidasiInputException("ISBN tidak boleh kosong");
        }
        if (tahunTerbit == null || tahunTerbit.trim().isEmpty()) {
            throw new ValidasiInputException("Tahun terbit tidak boleh kosong");
        }

        try {
            int tahun = Integer.parseInt(tahunTerbit);
            if (tahun < 1000 || tahun > 9999) {
                throw new ValidasiInputException("Tahun terbit tidak valid");
            }
        } catch (NumberFormatException e) {
            throw new ValidasiInputException("Tahun terbit harus berupa angka");
        }

        try {

```

```

        long hargaNumber = Long.parseLong(harga);
        if (hargaNumber < 0) {
            throw new ValidasiInputException("Harga tidak boleh angka negatif!");
        }
    } catch (NumberFormatException e) {
        throw new ValidasiInputException("Harga harus berupa angka!");
    }
}

public String getJudul() {
    return judul;
}

public String getPengarang() {
    return pengarang;
}

public String getIsbn() {
    return isbn;
}

public String getTahunTerbit() {
    return tahunTerbit;
}

public String getHarga() {
    return harga;
}
}

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

```

Kode ini menambahkan atribut harga ke kelas *Buku* untuk menyimpan informasi harga buku. Validasi memastikan input harga berupa angka positif menggunakan `Long.parseLong(harga)`. Jika harga negatif atau tidak bisa dikonversi menjadi angka, akan dilemparkan `ValidasiInputException` dengan pesan yang sesuai. Atribut harga kemudian diinisialisasi dan dapat diakses melalui metode `getHarga()`.

inputData.java

```

private void btnSubmitActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String judul = txtJudul.getText().trim();
    }
}

```

```

String pengarang = txtPengarang.getText().trim();

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

String tahunTerbit = txttahunTerbit.getText().trim();

String harga = txtHarga.getText().trim();


    if (judul.isEmpty() || pengarang.isEmpty() || isbn.isEmpty() ||
tahunTerbit.isEmpty() || harga.isEmpty()) {

        lblMessage.setForeground(Color.RED);

        lblMessage.setText("Semua field harus diisi!");

        return;

    }


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


    DefaultTableModel model = (DefaultTableModel) Tabel.getModel();

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


    txtJudul.setText("");

    txtPengarang.setText("");

    txtIsbn.setText("");

    txttahunTerbit.setText("");

    txtHarga.setText("");


    lblMessage.setForeground(Color.BLACK);

    lblMessage.setText("Buku berhasil ditambahkan.");
} catch (ValidasiInputException e) {

    lblMessage.setForeground(Color.RED);

    lblMessage.setText("Error: " + e.getMessage());

} catch (Exception e) {

    lblMessage.setForeground(Color.RED);

```

```

        lblMessage.setText("Error tak terduga: " + e.getMessage());
    }
}

```

Kode ini mengambil input harga melalui `txtHarga.getText()` dan menyimpannya dalam objek *Buku*. Data buku ditambahkan ke tabel menggunakan `DefaultTableModel`, lalu kolom input direset untuk entri baru. Jika berhasil, label `lblKeterangan` menampilkan pesan hijau "Buku berhasil ditambahkan!". Jika terjadi kesalahan input, seperti format harga yang salah, pengecualian ditangani, dan pesan kesalahan merah ditampilkan di label tersebut.

The screenshot shows a Java Swing window with a light gray background. At the top, there are five input fields with labels to their left: 'Judul' (containing 'kak ini ga selese error terus'), 'Pengarang' (containing 'kumpulin seadanya ya'), 'No ISBN' (containing '123456'), 'Tahun Terbit' (containing '1234'), and 'Harga' (containing '1234'). Below these fields is a button labeled 'Tambah Data'. Under the button is a table with five columns: 'Judul', 'Pengarang', 'ISBN', 'Tahun Terbit', and 'Harga'. The table has several empty rows. At the bottom left of the window is a label 'Keterangan'.

Judul	Pengarang	ISBN	Tahun Terbit	Harga

```
run:
Exception in thread "AWT-EventQueue-0" java.lang.RuntimeException: Uncompilable code - exception ValidationException is never thrown in body of corresponding try statement
at InputBuku.submitActionPerformed(InputBuku.java:251)
at InputBuku.submitActionPerformed(InputBuku.java:22)
at java.desktop/javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:1972)
at java.desktop/javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2313)
at java.desktop/javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:405)
at java.desktop/javax.swing.DefaultButtonModel.setPressed(DefaultButtonModel.java:262)
at java.desktop/javax.swing.plaf.basic.BasicButtonListener.mouseReleased(BasicButtonListener.java:278)
at java.desktop/java.awt.Component.processMouseEvent(Component.java:6626)
at java.desktop/javax.swing.JComponent.processMouseEvent(JComponent.java:3389)
at java.desktop/java.awt.Component.processEvent(Component.java:6391)
at java.desktop/java.awt.Container.processEvent(Container.java:2266)
at java.desktop/java.awt.Component.dispatchEventImpl(Component.java:5001)
at java.desktop/java.awt.Container.dispatchEventImpl(Container.java:2324)
at java.desktop/java.awt.Component.dispatchEvent(Component.java:4633)
at java.desktop/java.awt.LightweightDispatcher.retargetMouseEvent(Container.java:4548)
at java.desktop/java.awt.LightweightDispatcher.processMouseEvent(Container.java:4515)
at java.desktop/java.awt.LightweightDispatcher.dispatchEvent(Container.java:4516)
at java.desktop/java.awt.Container.dispatchEventImpl(Container.java:2310)
at java.desktop/java.awt.Window.dispatchEventImpl(Window.java:2780)
at java.desktop/java.awt.Component.dispatchEvent(Component.java:4633)
at java.desktop/java.awt.EventQueue.dispatchEventImpl(EventQueue.java:773)
at java.desktop/java.awt.EventQueue$4.run(EventQueue.java:728)
at java.desktop/java.awt.EventQueue$4.run(EventQueue.java:716)
at java.base/java.security.AccessController.doPrivileged(AccessController.java:399)
at java.base/java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:86)
at java.desktop/java.awt.EventQueue$5.run(EventQueue.java:745)
at java.desktop/java.awt.EventQueue$5.run(EventQueue.java:744)
at java.base/java.security.AccessController.doPrivileged(AccessController.java:399)
at java.base/java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:86)
at java.desktop/java.awt.EventQueue.dispatchEvent(EventQueue.java:743)
at java.desktop/java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:203)
at java.desktop/java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:124)
at java.desktop/java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:112)
at java.desktop/java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:109)
at java.desktop/java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101)
at java.desktop/java.awt.EventDispatchThread.run(EventDispatchThread.java:90)
```