

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 5
EXCEPTION HANDLING



Oleh :

Ardzaricho Revansyah

2311103137

S1SI-07-C

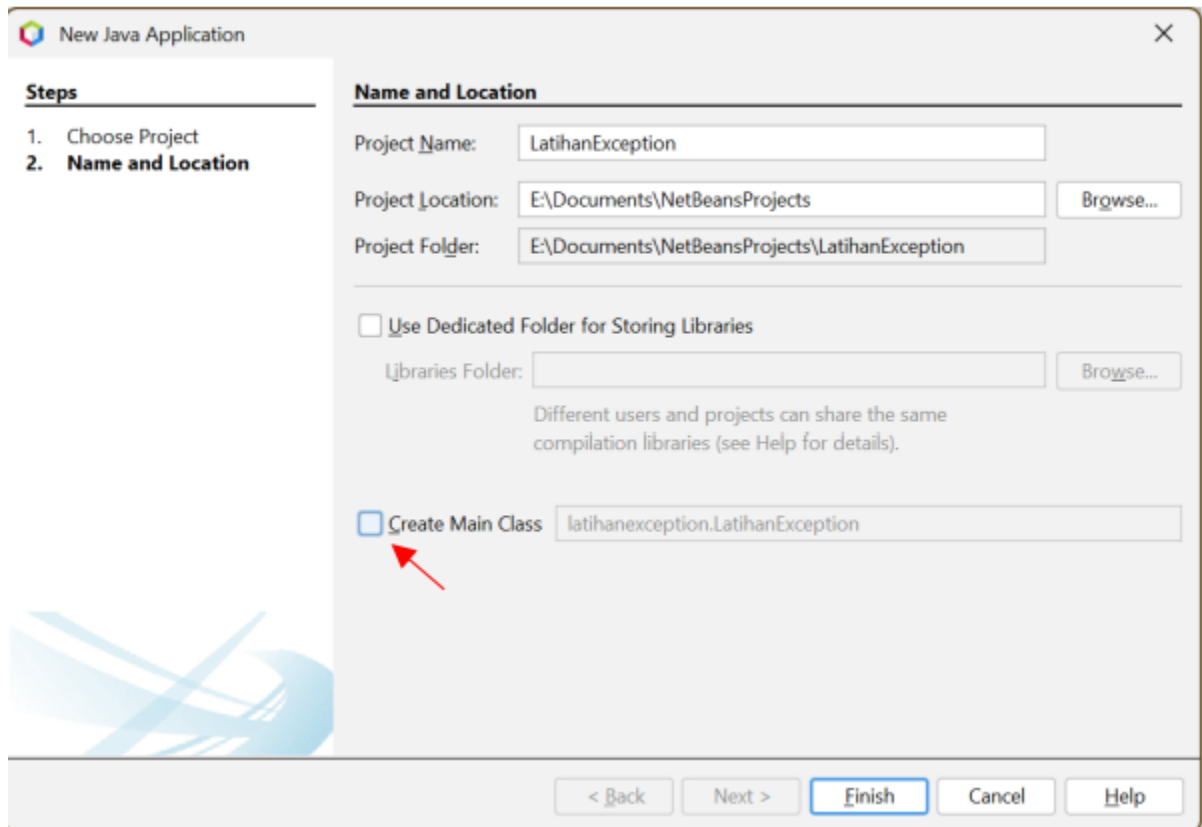
PROGRAM STUDI SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
UNIVERSITAS TELKOM PURWOKERTO
2024

A. Studi Kasus

Sistem Informasi Perpustakaan – Input Buku

B. GUIDED

1. Buat project baru pada Netbeans. Pada saat membuat project, hilangkan centang pada Create Main Class



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

Buku.java

```
package Perpustakaan;
```

```
/**
```

```
 * @author Ardzaricho Revansyah
```

```
 * 2311103137
```

```
* S1SI-07-c
*/
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;
    }
}
```

Penjelasan : Kodangan di atas mendefinisikan kelas Buku dalam paket Perpustakaan, yang merepresentasikan data buku dengan atribut privat judul, pengarang, dan isbn. Kelas ini menggunakan konstruktor untuk menginisialisasi atributnya dan menyediakan tiga metode

getter (getJudul, getPengarang, getIsbn) untuk mengakses nilai atribut tersebut dari luar kelas. Dengan pendekatan ini, prinsip enkapsulasi dalam pemrograman berorientasi objek terjaga. Komentar dokumentasi di awal kode memberikan informasi tentang identitas pembuat program.

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("");
}

```

Penjelasan : Kodingan di atas adalah metode yang dipanggil saat tombol "Submit" ditekan dalam sebuah aplikasi berbasis GUI (Graphical User Interface) Java. Metode ini mengambil data yang dimasukkan pengguna pada form (judul, pengarang, dan ISBN) menggunakan metode `getText()` dari objek `txtJudul`, `txtPengarang`, dan `txtIsbn`. Data tersebut kemudian digunakan untuk membuat sebuah objek `Buku` baru dengan memanfaatkan konstruktor kelas `Buku`. Setelah itu, data dari objek `Buku` ditambahkan ke tabel GUI (dengan nama `tblBuku`) melalui model tabel `DefaultTableModel` menggunakan metode `addRow`. Terakhir, metode ini mereset form input dengan mengosongkan teks pada `txtJudul`, `txtPengarang`, dan `txtIsbn`, sehingga form siap digunakan kembali.

7. Jalankan project lalu coba tambahkan data, maka akan muncul data pada tabel. Namun ketika form kosong lalu klik tambah data, maka akan ada data kosong yang masuk pada tabel. Hal ini bisa diantisipasi dengan membuat user defined exception
8. Buat sebuah class baru dengan nama `ValidasiInputException`

`ValidasiInputException.java`

```

package Perpustakaan;

/**
 * @author Ardzaricho Revansyah
 * 2311103137
 * S1SI-07-c
 */
public class ValidasiInputException extends Exception {
    public ValidasiInputException(String message) {
        super(message);
    }
}

```

Penjelasan : Kodingan di atas mendefinisikan sebuah kelas bernama `ValidasiInputException` dalam paket `Perpustakaan`. Kelas ini merupakan kelas custom exception yang dibuat dengan mewarisi kelas `Exception` bawaan Java. Custom exception ini bertujuan untuk menangani situasi atau kondisi khusus yang terjadi saat validasi input pada aplikasi gagal. Konstruktor kelas `ValidasiInputException` menerima sebuah parameter `String` bernama `message`, yang merupakan pesan error yang dapat digunakan untuk memberikan informasi spesifik terkait kesalahan. Konstruktor ini memanggil konstruktor superclass (`Exception`) untuk menginisialisasi exception dengan pesan tersebut. Adanya komentar dokumentasi di awal kode menunjukkan bahwa kelas ini dibuat oleh *Ardzaricho Revansyah* dengan identitas mahasiswa *2311103137* dari kelas *SISI-07-c*.

9. Ubah kode pada Class buku

Buku.java

```
package Perpustakaan;
```

```
/**
```

```
 * @author Ardzaricho Revansyah
```

```
 * 2311103137
```

```
 * SISI-07-c
```

```
 */
```

```
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;
}
}

```

Penjelasan : Pada kodingan di atas, penambahan dilakukan pada konstruktor kelas Buku untuk menyertakan validasi input menggunakan custom exception `ValidasiInputException`. Validasi ini memastikan bahwa data yang diberikan untuk atribut judul, pengarang, dan isbn sesuai dengan aturan yang ditetapkan. Jika judul atau pengarang kosong, akan dilemparkan exception dengan pesan kesalahan yang sesuai. Selain itu, validasi untuk isbn dilakukan menggunakan metode `matches`, yang memastikan bahwa nilai ISBN terdiri dari tepat 13 digit angka. Jika tidak memenuhi kriteria ini, exception dengan pesan "ISBN harus 13 digit angka!" akan dilemparkan. Penambahan ini meningkatkan keandalan program dengan memastikan data yang dimasukkan valid sebelum diinisialisasi ke dalam objek Buku.

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());  
    }  
}
```

Penjelasan : Penambahan pada kodingan di atas adalah penerapan mekanisme **exception handling** menggunakan blok try-catch untuk menangani validasi input yang dilakukan oleh konstruktor kelas Buku. Dalam blok try, jika data input valid, objek Buku berhasil

dibuat, ditambahkan ke tabel, dan form input direset. Selain itu, ditambahkan pesan keberhasilan pada label lblKeterangan, dengan teks berwarna hijau menggunakan metode setForeground. Namun, jika terjadi kesalahan validasi (misalnya, salah satu input kosong atau ISBN tidak sesuai format), exception `ValidasiInputException` akan ditangkap oleh blok catch. Dalam kasus ini, label lblKeterangan akan menampilkan pesan kesalahan yang sesuai dengan warna teks merah, memberikan umpan balik langsung kepada pengguna mengenai jenis kesalahan input. Penambahan ini meningkatkan pengalaman pengguna dengan memberikan informasi yang jelas tentang status operasi.

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



Judul Buku: Laut Bercerita

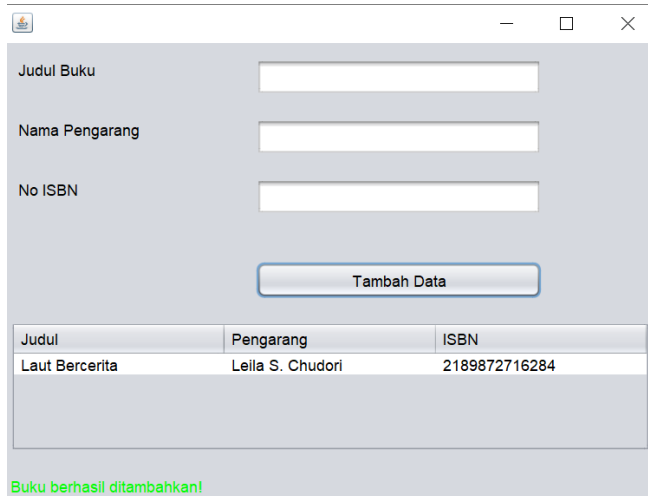
Nama Pengarang: Leila S. Chudori

No ISBN: 218987271628

Tambah Data

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

Error: ISBN harus 13 digit angka!



Judul Buku:

Nama Pengarang:

No ISBN:

Tambah Data

Judul	Pengarang	ISBN
Laut Bercerita	Leila S. Chudori	2189872716284

Buku berhasil ditambahkan!

C. UNGUIDED

Tambahkan inputan berikut pada aplikasi:

1. Tahun Terbit: Validasi untuk angka empat digit.

Buku.java

```
package Perpustakaan;

/**
 * @author Ardzaricho Revansyah
 * 2311103137
 * S1SI-07-c
 */
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 {

        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!");
    }

    if (!tahunTerbit.matches("\\d{4}")) {
        throw new ValidasiInputException("Tahun Terbit harus 4 digit angka!");
    }

    this.judul = judul;

    this.pengarang = pengarang;

    this.isbn = isbn;

    this.tahunTerbit = tahunTerbit;

    }

    public String getJudul() {
        return judul;
    }

    public String getPengarang() {
        return pengarang;
    }
}
```

```

public String getIsbn() {

    return isbn;

}

public String getTahunTerbit() {

    return tahunTerbit;

}

}

```

Penjelasan : Pada kodingan di atas, penambahan dilakukan dengan menambahkan atribut baru bernama tahunTerbit beserta validasinya dalam konstruktor. Validasi memastikan bahwa tahunTerbit harus terdiri dari 4 digit angka menggunakan metode matches. Jika tidak sesuai, akan dilemparkan ValidasiInputException dengan pesan kesalahan yang relevan. Selain itu, sebuah metode getter getTahunTerbit() ditambahkan untuk memungkinkan pengambilan nilai tahunTerbit. Penambahan ini memperluas informasi tentang buku dengan menyertakan tahun terbitnya, sekaligus menjaga validitas data yang dimasukkan.

InputBuku.java

```

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

    try {

        String judul = txtJudul.getText();

        String pengarang = txtPengarang.getText();

        String isbn = txtIsbn.getText();

        String tahunTerbit = txtThnTerbit.getText();

        String harga = txtHarga.getText();

        String kategori = (String) cbKategori.getSelectedItem();
    }
}

```

```

// Membuat objek Buku baru
Buku buku = new Buku(judul, pengarang, isbn, tahunTerbit, harga, kategori);

// Menambah data buku ke tabel
DefaultTableModel data = (DefaultTableModel)
tblBuku.getModel();

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

// Reset form input
txtJudul.setText("");
txtPengarang.setText("");
txtIsbn.setText("");
txtThnTerbit.setText("");
txtHarga.setText("");
cbKategori.setSelectedIndex(0);

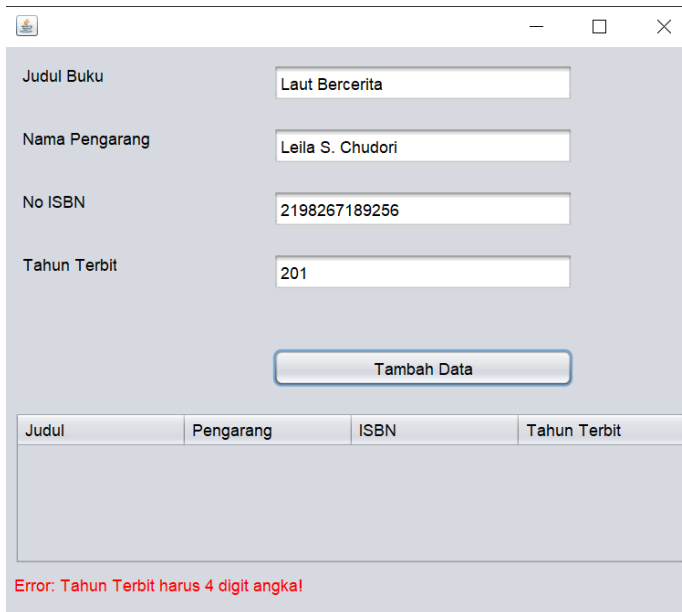
lblKeterangan.setForeground(Color.GREEN);
lblKeterangan.setText("Buku berhasil ditambahkan!");
} catch (ValidasiInputException e) {
    lblKeterangan.setForeground(Color.RED);
    lblKeterangan.setText("Error: " + e.getMessage());
}
}

```

Penjelasan : Pada kodingan di atas, penambahan mencakup pengolahan dua input baru, yaitu harga dan kategori. Input harga diambil dari txtHarga, sedangkan kategori dipilih dari cbKategori (komponen combo box). Data ini kemudian digunakan untuk membuat objek Buku yang sekarang mencakup atribut tambahan harga dan kategori. Data buku lengkap kemudian ditambahkan ke tabel menggunakan addRow, dengan kolom tambahan untuk harga

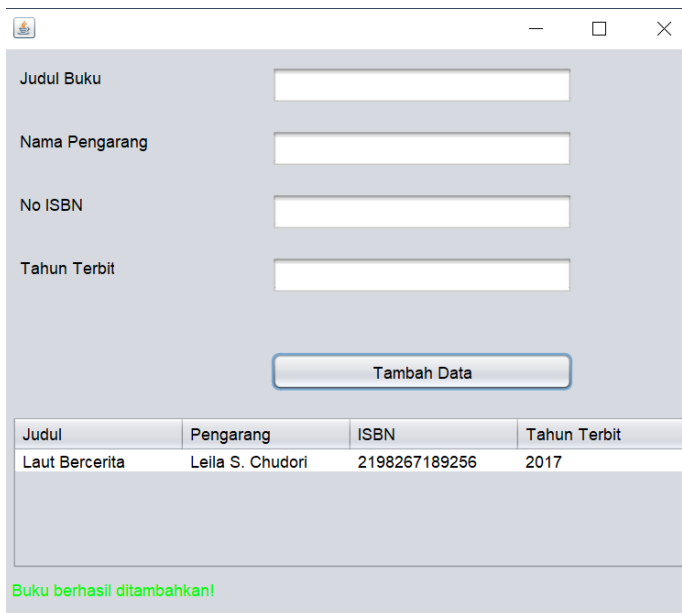
dan kategori. Selain itu, form input diperbarui untuk mereset txtHarga dan mengatur combo box cbKategori kembali ke indeks pertama. Penambahan ini memungkinkan aplikasi mengelola informasi buku yang lebih lengkap.

Output



The screenshot shows a window titled "Output" with a light gray background. It contains four input fields with labels: "Judul Buku" (containing "Laut Bercerita"), "Nama Pengarang" (containing "Leila S. Chudori"), "No ISBN" (containing "2198267189256"), and "Tahun Terbit" (containing "201"). Below these fields is a button labeled "Tambah Data". Under the button is a table with four columns: "Judul", "Pengarang", "ISBN", and "Tahun Terbit". The table is currently empty. At the bottom of the window, a red error message reads: "Error: Tahun Terbit harus 4 digit angka!"

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



The screenshot shows the same window as above, but now the input fields are empty. The "Tambah Data" button is still present. The table below it now contains one row of data: "Laut Bercerita", "Leila S. Chudori", "2198267189256", and "2017". At the bottom of the window, a green success message reads: "Buku berhasil ditambahkan!"

Judul	Pengarang	ISBN	Tahun Terbit
Laut Bercerita	Leila S. Chudori	2198267189256	2017

2. Harga Buku: Validasi untuk angka positif.

Buku.java

```
package Perpustakaan;

/**
 * @author Ardzaricho Revansyah
 * 2311103137
 * S1SI-07-c
 */
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 {

        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!");

}

if (!tahunTerbit.matches("\\d{4}")) {

    throw new ValidasiInputException("Tahun Terbit harus 4 digit angka!");

}

try {

    long hargaNumber = Long.parseLong(harga);

    if (hargaNumber < 0) {

        throw new ValidasiInputException("Harga tidak boleh angka negatif!");

    }

} catch (NumberFormatException e) {

    throw new ValidasiInputException("Input harus berupa angka!");

}


this.judul = judul;

this.pengarang = pengarang;

this.isbn = isbn;

this.tahunTerbit = tahunTerbit;

this.harga = harga;

}
```



```
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;  
}  
}
```

Penjelasan : Penambahan pada kodingan di atas meliputi atribut baru harga, beserta validasinya di dalam konstruktor. Validasi harga memastikan bahwa input berupa angka

positif dengan menggunakan metode `Long.parseLong`. Jika input tidak berupa angka atau merupakan angka negatif, akan dilemparkan `ValidasiInputException` dengan pesan yang sesuai. Selain itu, sebuah metode getter `getHarga()` ditambahkan untuk memungkinkan pengambilan nilai harga. Penambahan ini memungkinkan aplikasi untuk menangani informasi harga buku dengan validasi yang ketat untuk memastikan data yang dimasukkan valid.

InputBuku.java

```
private void BtnSubmitActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try {  
  
        String judul = txtJudul.getText();  
  
        String pengarang = txtPengarang.getText();  
  
        String isbn = txtIsbn.getText();  
  
        String tahunTerbit = txtTahunTerbit.getText();  
  
        String harga = txtHarga.getText();  
  
  
        // Membuat objek Buku baru  
  
        Buku buku = new Buku(judul, pengarang, isbn, tahunTerbit, harga);  
  
  
        // Menambah data buku ke tabel  
  
        DefaultTableModel data = (DefaultTableModel)  
tblBuku.getModel();  
  
        data.addRow(new Object[]{buku.getJudul(),  
buku.getPengarang(), buku.getIsbn(),
```

```

        buku.getTahunTerbit(), buku.getHarga()));

// Reset form input

txtJudul.setText("");

txtPengarang.setText("");

txtIsbn.setText("");

txtTahunTerbit.setText("");

txtHarga.setText("");

lblKeterangan.setForeground(Color.GREEN);

lblKeterangan.setText("Buku berhasil ditambahkan!");

    } catch (ValidasiInputException e) {

        lblKeterangan.setForeground(Color.RED);

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

    }

}

```

Penjelasan : Penambahan pada kodingan di atas mencakup pengolahan input baru, yaitu harga, yang diambil dari txtHarga. Data ini kemudian digunakan untuk membuat objek Buku, yang kini mencakup atribut harga. Selain itu, tabel tblBuku diperbarui dengan menambahkan kolom baru untuk menampilkan nilai harga. Form input juga diperluas untuk mereset nilai pada txtHarga setelah data berhasil ditambahkan. Penambahan ini memungkinkan aplikasi untuk mengelola informasi harga buku dan memberikan umpan balik langsung kepada pengguna tentang keberhasilan atau kesalahan validasi input.

Output

A Java Swing window titled "Perpustakaan" with a standard Mac OS X title bar (red, yellow, green buttons). The window contains a form with five text input fields and a "Tambah Data" button. The fields are labeled "Judul Buku", "Nama Pengarang", "No ISBN", "Tahun Terbit", and "Harga Buku". The inputs contain "Laut Bercerita", "Leila S. Chudori", "2198721627812", "2017", and "-200000" respectively. Below the inputs is a table with five columns: "Judul", "Pengarang", "ISBN", "Tahun Terbit", and "Harga". The table is currently empty. At the bottom, a red error message reads "Error: Harga tidak boleh angka negatif!".

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

Error: Harga tidak boleh angka negatif!

The same Java Swing window as above, but now the "Tambah Data" button is disabled. The table below the inputs now contains one row of data: "Laut Bercerita", "Leila S. Chudori", "2198721627812", "2017", and "200000". At the bottom, a green success message reads "Buku berhasil ditambahkan!".

Judul	Pengarang	ISBN	Tahun Terbit	Harga
Laut Bercerita	Leila S. Chudori	2198721627812	2017	200000

Buku berhasil ditambahkan!

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

Buku.java

```
package Perpustakaan;
```

```
/**
```

```
 * @author Ardzaricho Revansyah
```

```
 * 2311103137
```

```
 * S1SI-07-c
```

```
 */
```

```
public class Buku {
```

```
    private String judul;
```

```
    private String pengarang;
```

```
    private String isbn;
```

```
    private String tahunTerbit;
```

```
    private String harga;
```

```
    private String kategori;
```

```
    public Buku(String judul, String pengarang, String isbn, String tahunTerbit,
```

```
    String harga, String kategori) 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!");
```

```
}

if (!tahunTerbit.matches("\\d{4}")) {

    throw new ValidasiInputException("Tahun Terbit harus 4 digit angka!");

}

try {

    long hargaNumber = Long.parseLong(harga);

    if (hargaNumber < 0) {

        throw new ValidasiInputException("Harga tidak boleh angka negatif!");

    }

} catch (NumberFormatException e) {

    throw new ValidasiInputException("Input harus berupa angka!");

}

if (kategori == null || kategori.isEmpty()) {

    throw new ValidasiInputException("Kategori harus dipilih!");

}


this.judul = judul;

this.pengarang = pengarang;

this.isbn = isbn;

this.tahunTerbit = tahunTerbit;

this.harga = harga;

this.kategori = kategori;
```

```
}
```

```
public String getJudul() {  
    return judul;  
}
```

```
public String getPengarang() {  
    return pengarang;  
}
```

```
public String getIsbn() {  
    return isbn;  
}
```

```
public String getTahunTerbit() {  
    return tahunTerbit;  
}
```

```
public String getHarga() {  
    return harga;  
}
```

```
public String getKategori() {
```

```
        return kategori;
    }
}
```

Penjelasan : Penambahan pada kodingan di atas meliputi atribut kategori, yang menyimpan informasi kategori buku, dan validasinya dalam konstruktor. Validasi untuk kategori memastikan bahwa nilai yang dimasukkan tidak kosong atau null, melemparkan `ValidasiInputException` jika tidak valid. Selain itu, sebuah metode getter `getKategori()` ditambahkan untuk memungkinkan pengambilan nilai kategori. Penambahan ini memperluas fungsionalitas kelas `Buku` dengan memungkinkan pengelolaan informasi kategori buku, serta memastikan bahwa data yang dimasukkan sudah valid.

InputBuku.java

```
private void BtnSubmitActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String judul = txtJudul.getText();
        String pengarang = txtPengarang.getText();
        String isbn = txtIsbn.getText();
        String tahunTerbit = txtTahunTerbit.getText();
        String harga = txtHarga.getText();
        String kategori = (String) cbxKategori.getSelectedItem();

        // Membuat objek Buku baru
        Buku buku = new Buku(judul, pengarang, isbn, tahunTerbit, harga, kategori);

        // Menambah data buku ke tabel
        DefaultTableModel data = (DefaultTableModel)
```



```

tblBuku.getModel();

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

// Reset form input

txtJudul.setText("");
txtPengarang.setText("");
txtIsbn.setText("");
txtTahunTerbit.setText("");
txtHarga.setText("");
cbxKategori.setSelectedIndex(0);
lblKeterangan.setForeground(Color.GREEN);
lblKeterangan.setText("Buku berhasil ditambahkan!");

} catch (ValidasiInputException e) {

    lblKeterangan.setForeground(Color.RED);

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

}

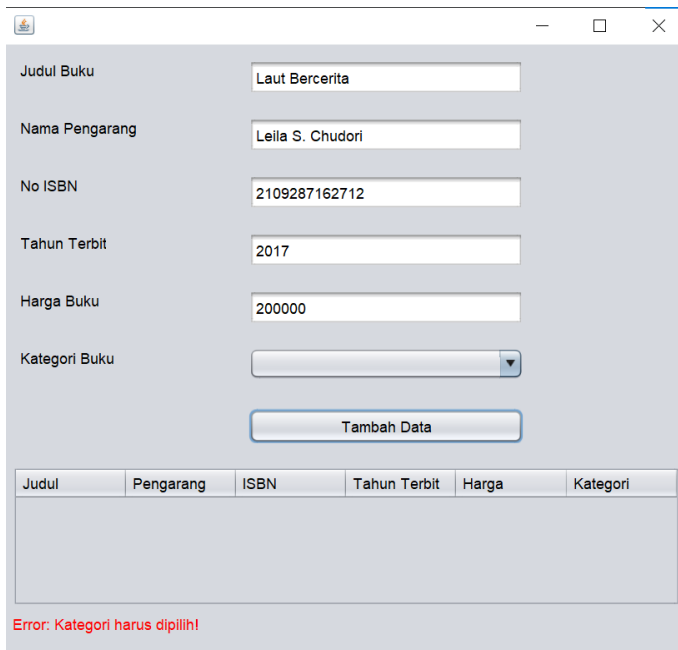
}

```

Penjelasan : Pada kodingan di atas, penambahan terletak pada pengambilan nilai kategori yang kini diambil dari komponen cbxKategori (combo box) dengan menggunakan metode

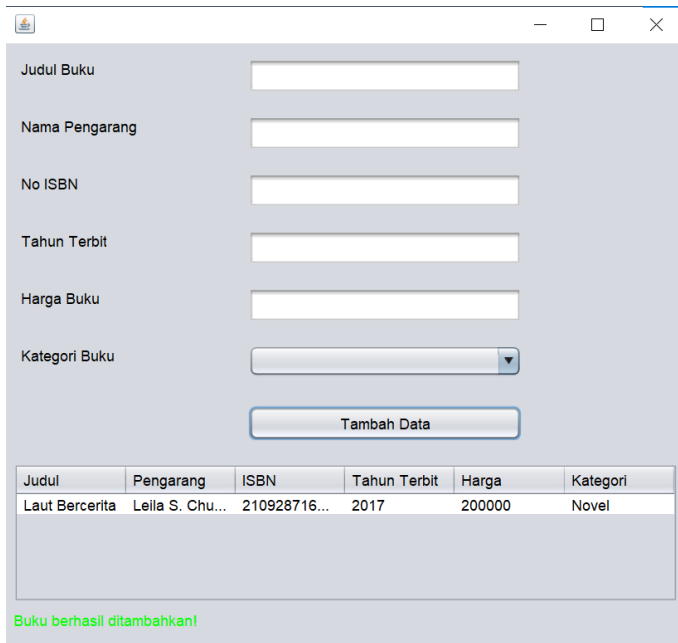
getSelectedItem()). Nilai kategori yang dipilih kemudian dimasukkan ke dalam objek Buku yang baru dibuat. Selanjutnya, data buku, termasuk kategori, ditambahkan ke tabel tblBuku. Setelah itu, form input direset, termasuk mengatur combo box cbxKategori kembali ke indeks pertama. Penambahan ini memungkinkan aplikasi untuk menangani informasi kategori buku yang dipilih oleh pengguna.

Output



A screenshot of a Java Swing window titled "Buku" (Book). The window contains a form with the following fields: "Judul Buku" (Book Title) with value "Laut Bercerita", "Nama Pengarang" (Author Name) with value "Leila S. Chudori", "No ISBN" with value "2109287162712", "Tahun Terbit" (Year Published) with value "2017", "Harga Buku" (Book Price) with value "200000", and "Kategori Buku" (Book Category) which is a dropdown menu currently showing an empty selection. Below the form is a "Tambah Data" (Add Data) button. Underneath the button is a table with 6 columns: "Judul", "Pengarang", "ISBN", "Tahun Terbit", "Harga", and "Kategori". The table is currently empty. At the bottom of the window, a red error message reads: "Error: Kategori harus dipilih!" (Error: Category must be selected!).

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



A screenshot of the same Java Swing window after the "Tambah Data" button has been clicked. The form fields are now empty. The "Kategori Buku" dropdown menu now shows "Novel" as the selected option. The "Tambah Data" button remains visible. The table below the button now contains one row of data: "Laut Bercerita", "Leila S. Chu...", "210928716...", "2017", "200000", and "Novel". At the bottom of the window, a green success message reads: "Buku berhasil ditambahkan!" (Book successfully added!).

Judul	Pengarang	ISBN	Tahun Terbit	Harga	Kategori
Laut Bercerita	Leila S. Chu...	210928716...	2017	200000	Novel

