LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK MODUL EXCEPTION HANDLING MATERI



Oleh :
MUHAMMAD RAFI AWALLAISAL
2311103134
S1SI07C

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

I. 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 @Muhammad Rafi Awallaisal 2311103134/S1SI07C
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 kosong ");
    }
    if(pengarang.isEmpty()){
       throw new ValidasiInputException ("pengarang 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;
  }
}
```

- 4. ambahkan 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:



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 bolehkosong!");
}
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 ();
        Buku buku = new Buku (judul, pengarang, isbn);

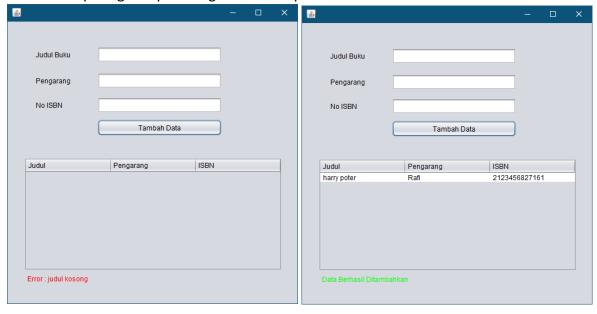
        DefaultTableModel data= (DefaultTableModel)tableBuku.getModel();
        data.addRow (new Object []{buku.getJudul(),buku.getPengarang(),buku.getIsbn()});

        txtJudul.setText("");
```

```
txtPengarang.setText("");
txtISBN.setText("");
lblMessage.setText("Data Berhasil Ditambahkan");
lblMessage.setForeground(Color.GREEN);

}catch(ValidasiInputException e){
   lblMessage.setText("Error: "+ e.getMessage());
   lblMessage.setForeground(Color.red);
}
```

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



II. 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)

Penambahan inputan pada class Buku.java

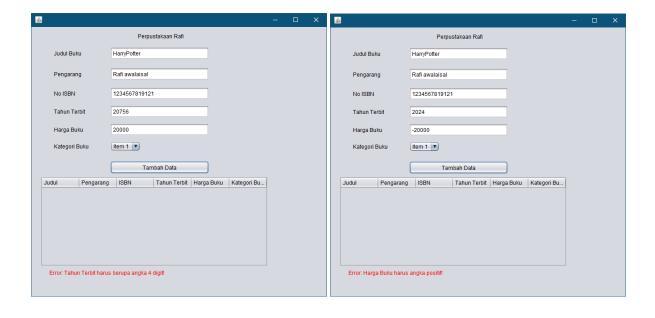
```
package perpustakaan;
//Author @Muhammad Rafi Awallaisal 2311103134/S1SI07C
public class Buku {
   private String judul;
   private String pengarang;
   private String isbn;
   private String tahunterbit;
   private String hargabuku;
   private String kategoribuku;
   public Buku (String judul, String pengarang, String isbn, String
tahunterbit, String hargabuku, String kategoribuku )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");
       this.judul = judul;
       this.pengarang = pengarang;
       this.isbn = isbn;
   }
   public String getJudul (){
       return judul;
   public String getPengarang(){
       return pengarang;
   public String getIsbn (){
       return isbn;
    public String getTahunterbit (){
       return tahunterbit;
    }
```

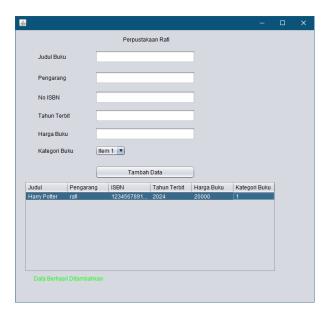
```
public String getKategoriBuku (){
    return kategoribuku;
}
    public String getHargabuku (){
    return hargabuku;
}
```

Pada source frame

```
private void btnSubmitActionPerformed(java.awt.event.ActionEvent evt)
{
       try{
        // Mengambil input
       String judul = txtJudul.getText();
        String pengarang = txtPengarang.getText();
        String isbn = txtISBN.getText();
        String tahunterbit = txtTahunTerbit.getText();
        String hargabuku = txtHargaBuku.getText();
        String kategoribuku = (String) txtKategoriBuku.getSelectedItem();
        // Validasi input
        if (judul.isEmpty() || pengarang.isEmpty() || isbn.isEmpty() ||
tahunterbit.isEmpty() || hargabuku.isEmpty()) {
            throw new ValidasiInputException("Semua field wajib diisi!");
        }
        // Validasi Tahun Terbit: harus angka 4 digit
        if (!tahunterbit.matches("\\d{4}")) {
            throw new ValidasiInputException("Tahun Terbit harus berupa angka
4 digit!");
        }
        // Validasi Harga Buku: harus angka positif
        double harga;
        try {
            harga = Double.parseDouble(hargabuku);
            if (harga <= 0) {</pre>
                throw new ValidasiInputException("Harga Buku harus angka
positif!");
        } catch (NumberFormatException ex) {
            throw new ValidasiInputException("Harga Buku harus berupa
angka!");
        // Validasi Kategori Buku: tidak boleh kosong
```

```
if (kategoribuku == null || kategoribuku.isEmpty()) {
            throw new ValidasiInputException("Pilih salah satu Kategori
Buku!");
        // Membuat objek Buku
        Buku buku = new Buku(judul, pengarang, isbn, tahunterbit, hargabuku,
kategoribuku);
        // Menambahkan data ke tabel
        DefaultTableModel data = (DefaultTableModel) tableBuku.getModel();
        data.addRow(new Object[]{buku.getJudul(), buku.getPengarang(),
buku.getIsbn(), buku.getTahunterbit(), buku.getHargabuku(),
buku.getKategoriBuku()});
        // Reset input
        txtJudul.setText("");
        txtPengarang.setText("");
        txtISBN.setText("");
        txtTahunTerbit.setText("");
        txtHargaBuku.setText("");
        txtKategoriBuku.setSelectedIndex(0);
        lblMessage.setText("Data Berhasil Ditambahkan");
        lblMessage.setForeground(Color.GREEN);
    } catch (ValidasiInputException e) {
        lblMessage.setText("Error: " + e.getMessage());
        lblMessage.setForeground(Color.RED);
       }
    }
```





Penjelasan:

Penambahan Konstruktor Buku Baru: Jika kelas Buku belum mendukung atribut tambahan (tahunTerbit, harga, kategori).

Validasi ini memastikan bahwa input tahun terbit hanya berupa angka dengan panjang empat digit (contoh: 2023).

Jika input tidak valid, akan muncul pesan error.

Menambahkan Properti Baru: Jika kelas Buku hanya memiliki atribut judul, pengarang, dan isbn, maka ditambahkan properti tambahan:

Tahun Terbit (String tahun Terbit)

Harga (double harga)

Kategori (String kategori)