

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 5
LATIHAN EXCEPTION



Oleh :

Bunga Ramadhani S

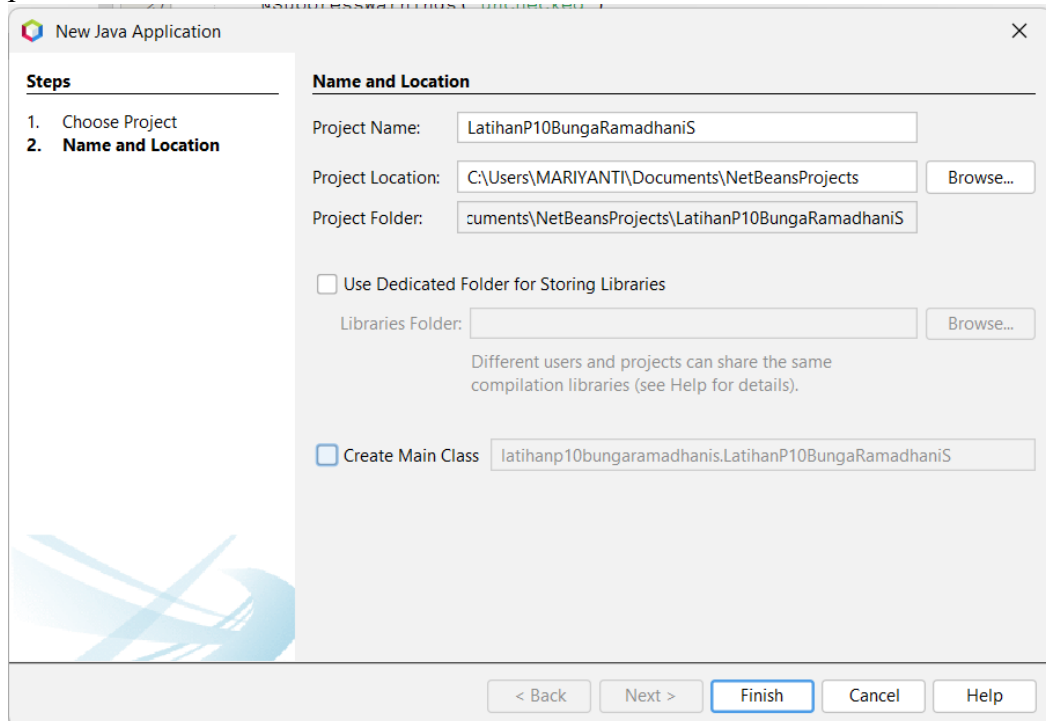
2311103128

S1SI-07-C

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

```
/**
 *
 * @author Bunga Ramadhani S
 * 2311103128
 * S1SI07C
 */

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:

The screenshot shows a Java Swing window titled "InputBuku". It has a light gray background. At the top, there are three text input fields. The first is labeled "Judul Buku", the second "Pengarang", and the third "ISBN". Below these fields is a button labeled "Tambah Data". At the bottom of the window, there is a table with three columns: "Judul", "Pengarang", and "ISBN". Below the table is a label "Keterangan".

6. Klik 2x pada tombol Tambah Data, tambahkan kode berikut

```

private void submitActionPerformed(java.awt.event.ActionEvent evt) {
    String Judul = judul.getText();
    String Pengarang = pengarang.getText();
    String ISBN = isbn.getText();

    // Membuat objek Buku baru
    Buku buku = new Buku(Judul, Pengarang, ISBN);
}

```

```

        // Menambah data buku ke tabel
        DefaultTableModel data = (DefaultTableModel)
tableBuku.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 Kosong");
        }
        if (pengarang.isEmpty()) {
            throw new ValidasiInputException("Pengarang Kosong");
        }
        if (!isbn.matches("\\d{13}")) {
            throw new ValidasiInputException("ISBN harus terdiri dari 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;
    }
}

```

10. Klik 2x pada tombol Tambah Data, ubah kode berikut

```

private void submitActionPerformed(java.awt.event.ActionEvent evt){
    try {
        String Judul = judul.getText();
        String Pengarang = pengarang.getText();
        String ISBN = isbn.getText();

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

        // Menambah data buku ke tabel
        DefaultTableModel data = (DefaultTableModel)
tableBuku.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

II. UNGUIDED

Tambahkan inputan berikut pada aplikasi:

1. Tahun Terbit: Validasi untuk angka empat digit.

Buku.java

```
/**
 *
 * @author Bunga Ramadhani S
 * 2311103128
 * S1SI07C
 */
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 ValidasiInput{
        if (judul.isEmpty()){
            throw new ValidasiInput ("Judul Kosong");
        }
        if (pengarang.isEmpty()){
            throw new ValidasiInput ("Pengarang Kosong");
        }
        if (!isbn.matches("\\d{13}")){
            throw new ValidasiInput ("ISBN harus terdiri dari 13 angka");
        }
        if (!tahunTerbit.matches("\\d{4}")){
            throw new ValidasiInput ("Tahun Terbit harus terdiri dari 4
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:

Menambahkan atribut baru yaitu tahunTerbit, lalu bersama dengan validasinya dalam konstruktor, ditambahkan ke dalam kode ini. Tahun Terbit harus memiliki 4 digit tepat. Kalau tidak, pengecualian input valid akan dikirim dengan pesan, "Tahun Terbit harus terdiri dari 4 angka" untuk mendapatkan nilai tahunTerbit, metode getter getTahunTerbit() juga ditambahkan. Dengan penambahan ini, data tentang tahun terbit buku dapat disimpan dan divalidasi.

InputData.java

```

        private void submitActionPerformed(java.awt.event.ActionEvent evt)
        {
            try {
                String Judul = judul.getText();
                String Pengarang = pengarang.getText();
                String ISBN = isbn.getText();
                String TahunTerbit = tahunTerbit.getText();

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

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

                judul.setText("");
                pengarang.setText("");
                isbn.setText("");
                tahunTerbit.setText("");

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

```

Penjelasan:

Penambahan pada kode ini termasuk mengambil input dari tahunTerbit, yang digunakan untuk mengaktifkan atribut tahunTerbit saat membuat objek Buku. Selain itu, data tahun terbit ditambahkan ke tabel dengan menggunakan model TabelDefaultModel. Setelah data dimasukkan dengan benar, input tahunTerbit direset bersama dengan input lainnya. Dengan penambahan ini, aplikasi sekarang dapat memantau semua informasi tentang tahun terbit buku.

Output:

Judul Buku: Khodam GGS

Pengarang: Jaka Tingkir

ISBN: 2311103128888

Tahun Terbit: 19

Tambah Data

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

Error: Tahun Terbit harus terdiri dari 4 angka

Penjelasan:

Selain menambah kode program, harus menambah kolom pada desain form untuk Tahun Terbit. Hasil outputnya seperti diatas, dimana kolom yang tadinya ada 3 bertambah menjadi 4.

2. Harga Buku: Validasi untuk angka positif.

Buku.java

```
/**
 *
 * @author Bunga Ramadhani S
 * 2311103128
 * S1SI07C
 */
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) throws ValidasiInput{
    if (judul.isEmpty()){
        throw new ValidasiInput ("Judul Kosong");
    }
    if (pengarang.isEmpty()){
        throw new ValidasiInput ("Pengarang Kosong");
    }
    if (!isbn.matches("\\d{13}")){
        throw new ValidasiInput ("ISBN harus terdiri dari 13 angka");
    }
    if (!tahunTerbit.matches("\\d{4}")){
        throw new ValidasiInput ("Tahun Terbit harus terdiri dari 4
angka");
    }
    try {
        long hargaNumber = Long.parseLong(harga);
        if (hargaNumber < 0) {
            throw new ValidasiInput("Harga tidak boleh angka negatif!");
        }
    } catch (NumberFormatException e) {
        throw new ValidasiInput("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:

Ada atribut harga baru dan validasi yang ditambahkan ke kode ini. Validasi menjamin input harga adalah angka positif. Akan mengirimkan pengecualian `ValidasiInput` dengan pesan yang relevan jika harga bernilai negatif atau tidak valid. Untuk mendapatkan nilai atribut harga, metode getter `getHarga()` juga ditambahkan. Aplikasi dapat mengontrol informasi harga buku dengan validasi yang sesuai dengan penambahan ini.

`InputData.java`

```

    private void submitActionPerformed(java.awt.event.ActionEvent evt)
    {
        try {
            String Judul = judul.getText();
            String Pengarang = pengarang.getText();
            String ISBN = isbn.getText();
            String TahunTerbit = tahunTerbit.getText();
            String Harga = harga.getText();

            Buku buku = new Buku(Judul, Pengarang, ISBN, TahunTerbit,
            Harga);

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

            judul.setText("");
            pengarang.setText("");
            isbn.setText("");
            tahunTerbit.setText("");
            harga.setText("");

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

```

Penjelasan:

Penambahan pada kode ini termasuk mengambil input harga untuk atribut harga saat membuat objek `Buku` dan memasukkan data harga ke tabel dengan menggunakan model `TabelDefaultModel`. Setelah data dimasukkan, input harga direset bersama dengan input lainnya. Aplikasi sekarang dapat mengelola semua informasi harga buku, dengan validasi yang sudah dilakukan pada konstruktor kelas `buku`.

Output:



—

□

×

Judul Buku

abc

Pengarang

def

ISBN

1234567891011

Tahun Terbit

2011

Harga

-2000

Tambah Data

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

Error: Harga tidak boleh angka negatif!

Judul Buku

Pengarang

ISBN

Tahun Terbit

Harga

Tambah Data

Judul	Pengarang	ISBN	Tahun Terbit	Harga
abc	def	1234567891011	2020	7900

Buku berhasil ditambahkan!

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

```
/**
 *
 * @author Bunga Ramadhani S
 * 2311103128
 * S1SI07C
 */
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 ValidasiInput{
        if (judul.isEmpty()){
            throw new ValidasiInput ("Judul Kosong");
        }
        if (pengarang.isEmpty()){
            throw new ValidasiInput ("Pengarang Kosong");
        }
        if (!isbn.matches("\\d{13}")){
            throw new ValidasiInput ("ISBN harus terdiri dari 13 angka");
        }
        if (!tahunTerbit.matches("\\d{4}")){
            throw new ValidasiInput ("Tahun Terbit harus terdiri dari 4
angka");
        }
        try {
            long hargaNumber = Long.parseLong(harga);
            if (hargaNumber < 0) {
                throw new ValidasiInput("Harga tidak boleh angka negatif!");
            }
        } catch (NumberFormatException e) {
            throw new ValidasiInput("Input harus berupa angka!");
        }
        if (kategori == null || kategori.isEmpty()) {
            throw new ValidasiInput("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 kode ini adalah atribut baru kategori beserta validasinya. Validasi memastikan bahwa input kategori tidak kosong atau invalid. Jika kategori tidak dipilih, maka pengecualian `ValidInputException` akan dilempar dengan pesan "Kategori harus dipilih!". Selain itu, metode getter `getKategori()` ditambahkan untuk mengakses nilai atribut kategori. Penambahan ini memungkinkan aplikasi untuk menyimpan dan memvalidasi kategori buku yang dimasukkan, menambah detail informasi buku yang dikelola.

InputData.java

```

private void submitActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String Judul = judul.getText();
        String Pengarang = pengarang.getText();
        String ISBN = isbn.getText();
        String TahunTerbit = tahunterbit.getText();
        String Harga = harga.getText();
        String Kategori = (String) kategori.getSelectedItem();

        Buku buku = new Buku(Judul, Pengarang, ISBN, TahunTerbit,
            Harga, Kategori);

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

        judul.setText("");
        pengarang.setText("");
        isbn.setText("");
        tahunterbit.setText("");
        harga.setText("");
        kategori.setSelectedItem(0);

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

```

```

    }
}

```

Penjelasan:

Salah satu penambahan pada kode ini adalah mengambil nilai dari combo box cbKategori untuk menghasilkan kategori yang telah dipilih pengguna, yang kemudian digunakan untuk membuat objek buku. Setelah objek buku dibuat, data kategori juga ditambahkan ke tabel. Selain itu, setelah data dimasukkan, nilai input cbKategori direset ke indeks pertama, yang dapat tetap kosong atau default. Dengan penambahan ini, aplikasi dapat mengintegrasikan kategori buku ke dalam proses pengelolaan data dan memberikan umpan balik yang relevan kepada pengguna.

Output:

The screenshot shows a Java Swing application window with a light gray background. At the top, there's a title bar with a small icon and standard window controls. Below the title bar, the form is organized into a vertical stack of input fields, each with a label to its left: 'Judul Buku', 'Pengarang', 'ISBN', 'Tahun Terbit', 'Harga', and 'Kategori'. The 'Kategori' field is a dropdown menu currently showing 'Biografi'. Below these fields is a button labeled 'Tambah Data'. Underneath the button is a table with six columns: 'Judul', 'Pengarang', 'ISBN', 'Tahun Terbit', 'Harga', and 'Kategori'. The table contains one row of data: 'bunga', 'ramadhani', '2311103128...', '2005', '999000', and 'Biografi'. At the very bottom of the window, there is a green text message: 'Buku berhasil ditambahkan!'.

Judul	Pengarang	ISBN	Tahun Terbit	Harga	Kategori
bunga	ramadhani	2311103128...	2005	999000	Biografi

Buku berhasil ditambahkan!