

LAPORAN PRAKTIKUM PEKAN 7
PEMOGRAMAN BERORIENTASI OBJEK



Oleh :

KEVIN ANDHIKA

NIM 2311532005

MATA KULIAH PEMOGRAMAN BERORIENTASI OBJEK

DOSEN PENGAMPU : NURFIAH, S.ST., M.KOM.

FAKULTAS TEKNOLOGI INFORMASI

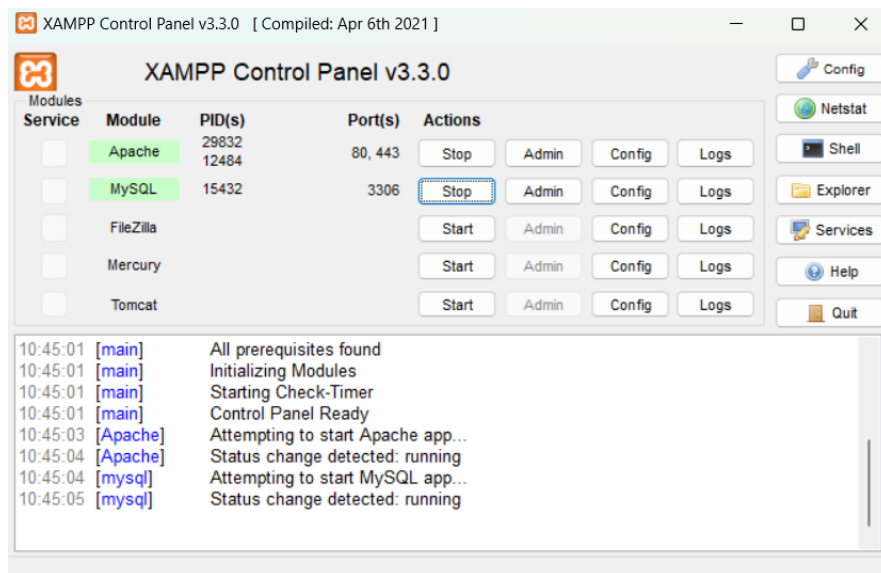
DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, NOVEMBER 2024

PEMBAHASAN

1. Buka phpmyadmin melalui xampp, jangan lupa mencentang kolom Apache dan MySQL untuk menghubungkan database



2. Buka class **User** yang ada didalam package **Model** lalu tambahkan syntax berikut

```
*User.java x ValidationEx... ValidationU... LoginService...
1 package Model;
2
3 public class User {
4     String id, nama, username, password;
5
6     public User(String uname, String pass) {
7         this.username = uname;
8         this.password = pass;
9     }
10
11     public String getId() {
12         return id;
13     }
14     public void setId(String id) {
15         this.id = id;
16     }
17
18     public String getNama() {
19         return nama;
20     }
21     public void setNama(String nama) {
22         this.nama = nama;
23     }
24
25     public String getUsername() {
26         return username;
27     }
28     public void setUsername(String username) {
29         this.username = username;
30     }
31
32     public String getPassword() {
33         return password;
34     }
35     public void setPassword(String password) {
36         this.password = password;
37     }
38
39     public void sayHai (String message) {
40         System.out.println(message);
41     }
42 }
```

3. Untuk membuat sebuah exception pada program, buat package baru yang bernama **Error** dan di dalamnya dibuat class **ValidationException** dengan extends **Throwable** dengan kode syntax sebagai berikut

```
*User.java  ValidationEx...  ValidationU...  LoginServi
1 package Error;
2
3 public class ValidationException extends Throwable {
4     public ValidationException(String message) {
5         super(message);
6     }
7 }
8
```

4. Kemudian buat lagi package baru yang bernama **Util** dan di dalamnya dibuat class **ValidationUtil** yang berguna untuk memverifikasi proses login saat terjadi suatu error dengan memberitahu apa yang error dengan menggunakan syntax berikut

```
*User.java  ValidationEx...  ValidationU...  LoginService...  LoginFrame.java
1 package util;
2
3 import Error.ValidationException;
4
5
6 public class ValidationUtil {
7     public static void validate(User user) throws ValidationException, NullPointerException{
8         if(user.getUsername() == null) {
9             throw new NullPointerException("Username is null");
10        }else if(user.getUsername().isBlank()) {
11            throw new NullPointerException("Username is empty");
12        }else if(user.getPassword() == null) {
13            throw new NullPointerException("Password is null");
14        }else if(user.getPassword().isBlank()) {
15            throw new NullPointerException("Password is empty");
16        }
17    }
18 }
```

5. Buat lagi package baru yang bernama **Service** dengan class **LoginService** di dalamnya, sertakan syntax ini di dalam class loginservice tersebut

```
*User.java  ValidationEx...  ValidationU...  LoginService...  LoginFrame.java
1 package service;
2
3 import java.sql.Connection;
10
11 public class LoginService {
12     public boolean authenticate(User user) {
13         String query = "SELECT * FROM user WHERE username = ? AND password = ?";
14
15         try (Connection conn = Database.koneksi();
16             PreparedStatement statement = conn.prepareStatement(query)) {
17             statement.setString(1, user.getUsername());
18             statement.setString(2, user.getPassword());
19
20             ResultSet resultSet = statement.executeQuery();
21             return resultSet.next();
22         } catch (SQLException e) {
23             e.printStackTrace();
24         }
25         return false;
26     }
27 }
```

6. Setelah itu, buka kembali class JFrame **LoginFrame** yang ada pada package **UI**, pada bagian tombol login tambahkan validasi dengan cara memanggil method yang dibuat sebelumnya

```
public void actionPerformed(ActionEvent e) {
    String userValue = tfUsername.getText();
    String passValue = tfPassword.getText();

    User user = new User(userValue, passValue);
    try {
        ValidationUtil.validate(user);
        LoginService loginSer = new LoginService();
        if(loginSer.authenticate(user)) {
            System.out.println("Login Successfull!!");
            JOptionPane.showMessageDialog(null, "Login Successfull!!");
            new MainFrame().setVisible(true);
            dispose();
        } else {
            System.out.println("Invalid Username or Password");
            JOptionPane.showMessageDialog(null, "Login Gagal, Invalid Username or Password.");
        }
    } catch (ValidationException | NullPointerException exception) {
        System.out.println("Data Tidak Valid : " + exception.getMessage());
        JOptionPane.showMessageDialog(null, "Login Gagal: " + exception.getMessage());
    } finally {
        System.out.println("Selalu Dieksekusi");
    }
}
```

HASIL PRAKTIKUM

* **Catatan** : Sebelum melakukan pengujian, tabel user pada database harus berisi minimal satu buah user *

Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

```
SELECT * FROM `user`
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

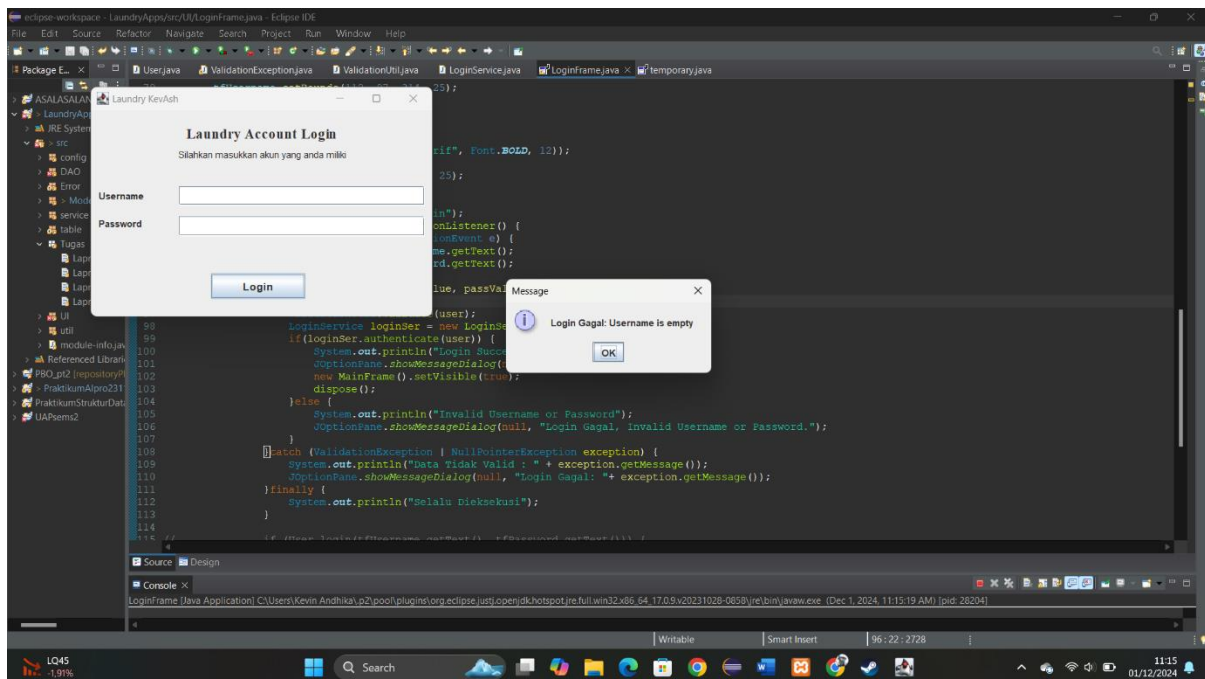
☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

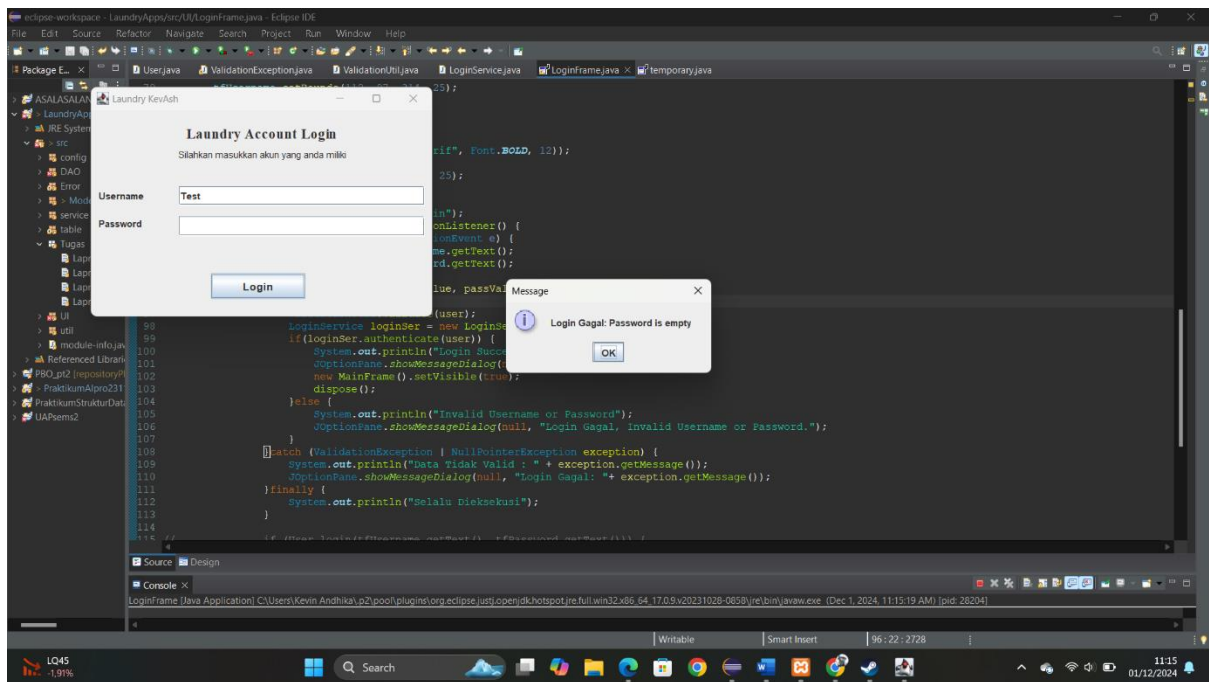
		id	name	username	password
<input type="checkbox"/>	Edit	1	Kevin Andhika	Flan	Kevin25TXC
<input type="checkbox"/>	Edit	2	Test User	test	12345

☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

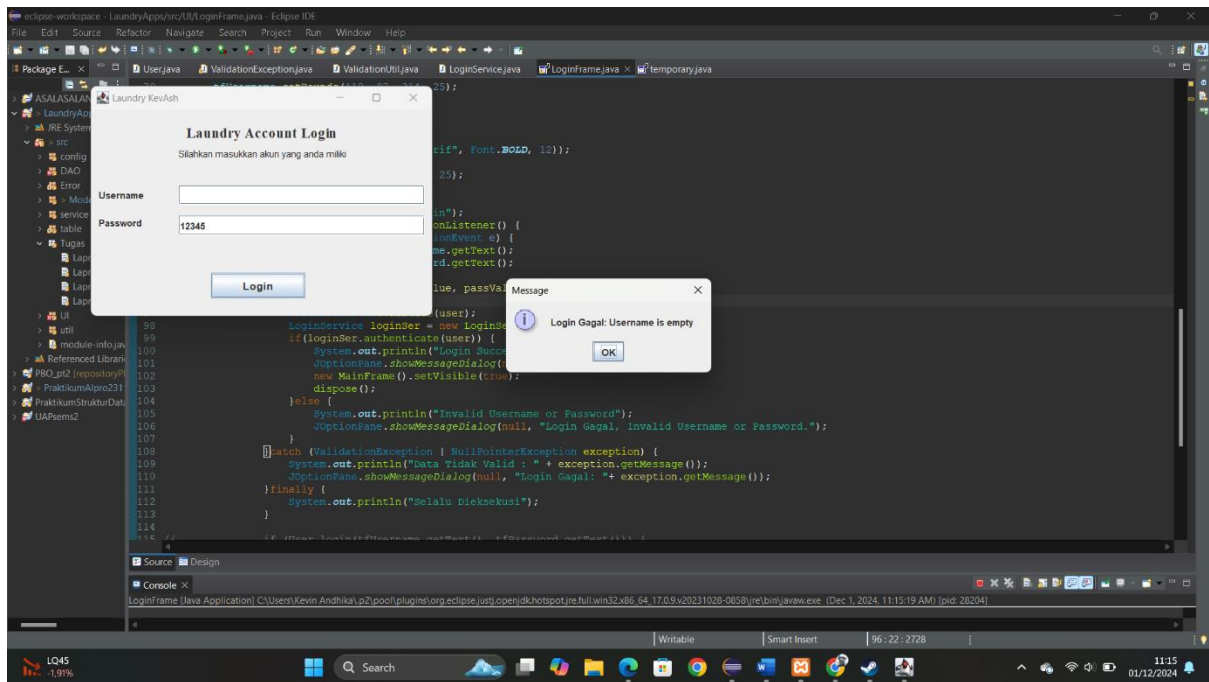
CASE 1 : Username & Password kosong



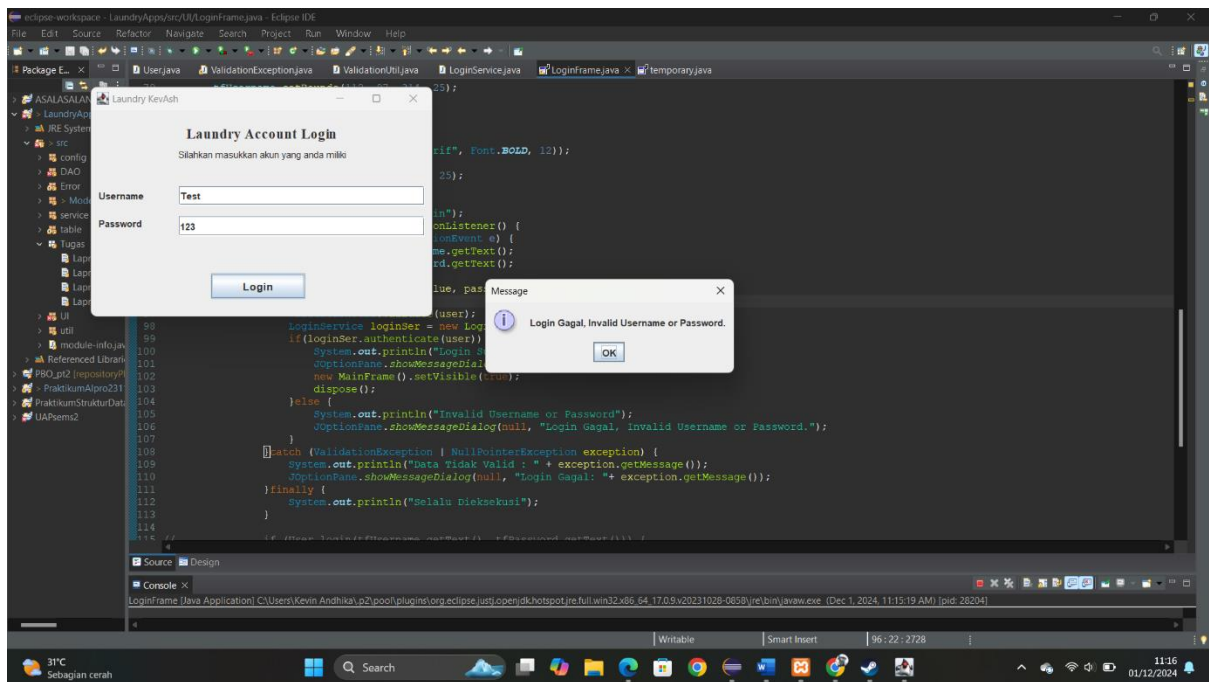
CASE 2 : Password kosong



CASE 3 : Username kosong



CASE 4 : Username & Password salah



CASE 5 : Username & Password benar

