

**LAPORAN PRAKTIKUM**  
**BAHASA PEMROGRAMAN 2**  
**MODUL 2**

Dosen pengampu : Yulyanto, S.Kom., M.TI.



Disusun oleh :

Nama : Muhammad Rizal Nurfirdaus  
NIM : 20230810088  
Jadwal : Rabu, 14:40 – 16:15  
Kelas : TINFO-2023-04

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS KUNINGAN**

**2025**

## DAFTAR ISI

DAFTAR ISI.....	ii
BAB I.....	1
PRETEST.....	1
BAB II.....	2
PRAKTIKUM.....	2
BAB III .....	4
POSTTEST .....	4
BAB IV .....	8
TUGAS .....	8
BAB V .....	11
KESIMPULAN.....	11

## **BAB I**

### **PRETEST**

1. Tuliskan Script YesNoOption Message. Pada event JOptionPane Kerjakan pada selembar kertas dengan waktu 5 menit dan dikumpulkan melalui Asisten lab.

```
import javax.swing.JOptionPane;

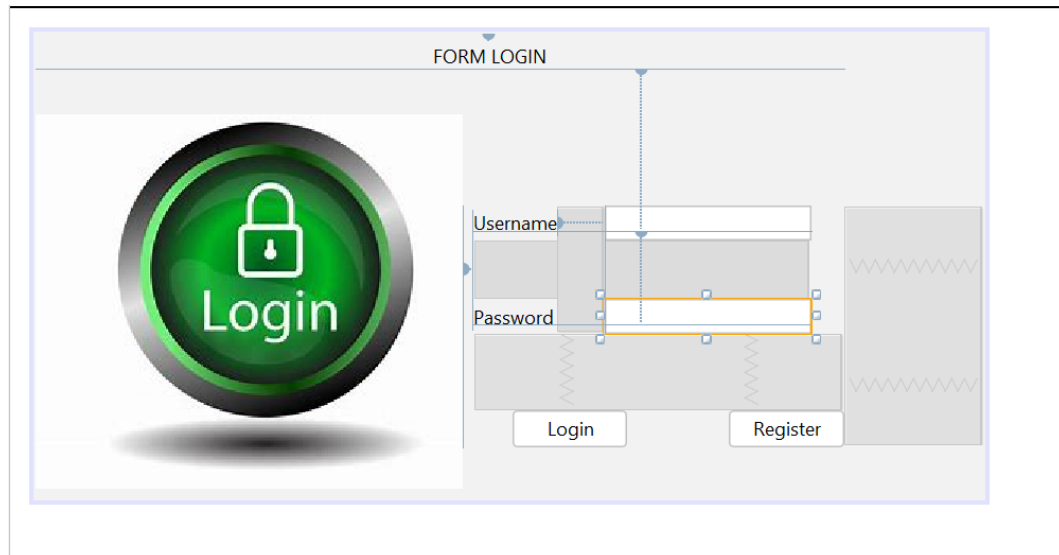
public class YesNoOptionExample {
    public static void main(String[] args) {
        int pilihan = JOptionPane.showConfirmDialog(
            null,
            "Apakah Anda yakin ingin keluar?",
            "Konfirmasi",
            JOptionPane.YES_NO_OPTION
        );

        if (pilihan == JOptionPane.YES_OPTION) {
            System.out.println("Anda memilih YES.");
            // Tambahkan aksi jika pengguna memilih YES, misalnya keluar aplikasi
            System.exit(0);
        } else if (pilihan == JOptionPane.NO_OPTION) {
            System.out.println("Anda memilih NO.");
            // Tambahkan aksi jika pengguna memilih NO
        }
    }
}
```

## BAB II

### PRAKTIKUM

1. Buatlah User Interface dengan Menggunakan komponen SWING dengan sebagai berikut :



Lalu memasukkan kode sebagai berikut :

//Kode Login

```
private void login(){
    String user, pass;

    String IUser= jTextField1.getText(); //Mengambil data dari TextFiled Input
    IUser

    String Ipass= jPasswordField1.getText(); //Mengambil data dari
    PAsswordFiled IPassword

    user = IUser;
    pass = Ipass;

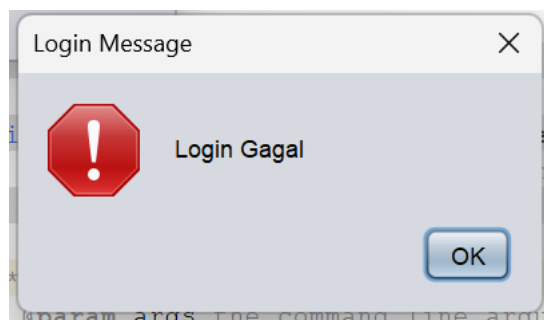
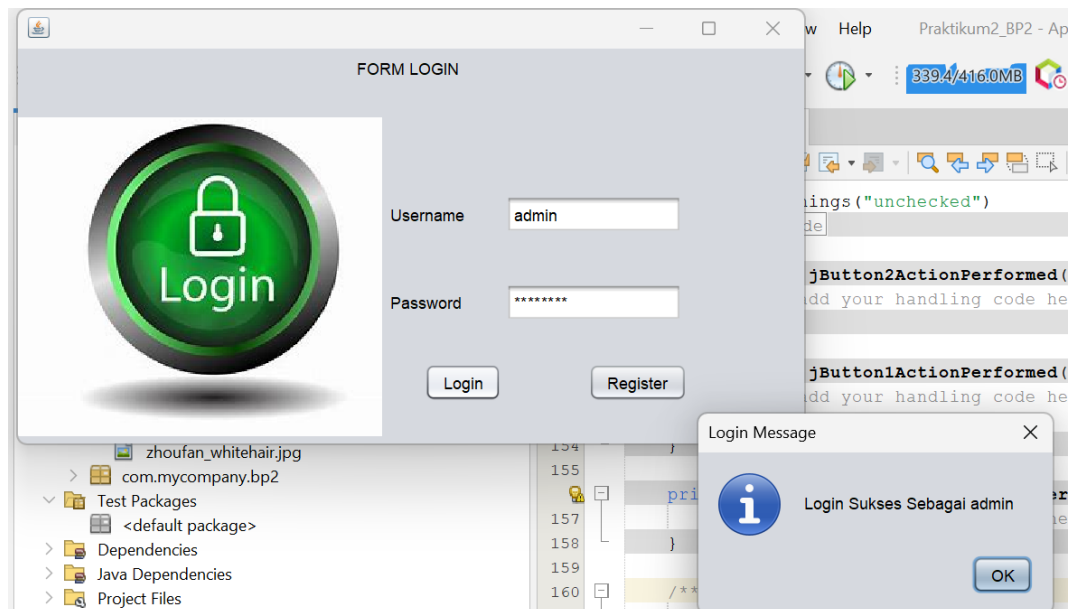
    //Kondisi Login
    if(user.equals("admin")&&pass.equals("admin123"))
    {
        JOptionPane.showMessageDialog(null, "Login Sukses Sebagai " +user,
            "Login Message",JOptionPane.INFORMATION_MESSAGE);
    }
    else
```

```

    {
        JOptionPane.showMessageDialog(null, "Login Gagal",
            "Login Message",JOptionPane.ERROR_MESSAGE);
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    login();
}

```



Analisis : Kode login ini memiliki struktur yang sederhana dan langsung pada inti fungsi. Program mengambil input dari jTextField1 dan jPasswordField1, lalu membandingkan apakah username bernilai "admin" dan password "admin123". Jika keduanya sesuai, maka login dinyatakan berhasil, dan pesan sukses ditampilkan melalui JOptionPane. Sebaliknya, jika tidak sesuai, maka akan muncul pesan login gagal.

### **BAB III**

#### **POSTTEST**

1. Lengkapi Kondisi login di atas, sehingga memiliki kondisi login sebagai berikut :
  - a. Jika Username Tidak Diisi, maka akan memberikan Message “Username Harus Diisi”.
  - b. Jika Password Tidak Diisi, maka akan memberikan Message “Password Harus Diisi”.
  - c. Jika Username Salah, maka akan memberikan Message “Username Harus Salah”.
  - d. Jika Password Salah, maka akan memberikan Message “Password Harus Salah”.

Script program :

```
private void login(){  
    String user = IUser.getText().trim();  
    String pass = new String(IPassword.getPassword()).trim(); // JPasswordField  
  
    // Validasi untuk username tidak diisi  
    if (user.isEmpty()) {  
        JOptionPane.showMessageDialog(null, "Username harus diisi", "Login  
Message", JOptionPane.WARNING_MESSAGE);  
        return;  
    }  
  
    // Validasi untuk password tidak diisi  
    if (pass.isEmpty()) {  
        JOptionPane.showMessageDialog(null, "Password harus diisi", "Login  
Message", JOptionPane.WARNING_MESSAGE);  
        return;  
    }  
  
    // Cek Username  
    if (!user.equals("admin")) {
```

```

JOptionPane.showMessageDialog(null, "Username Salah", "Login
Message", JOptionPane.ERROR_MESSAGE);

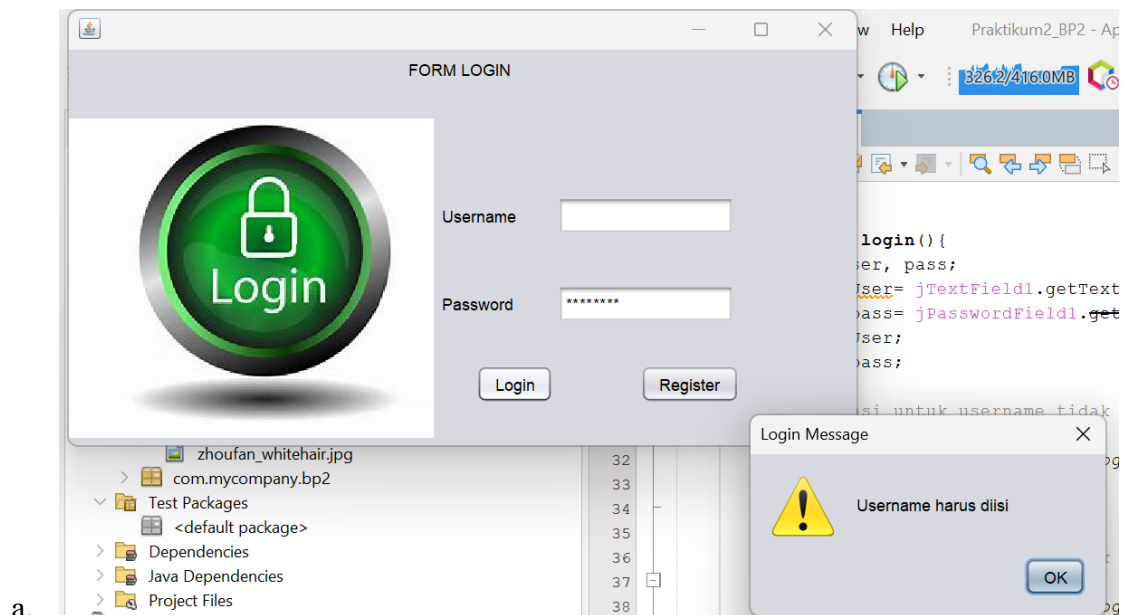
return;
}

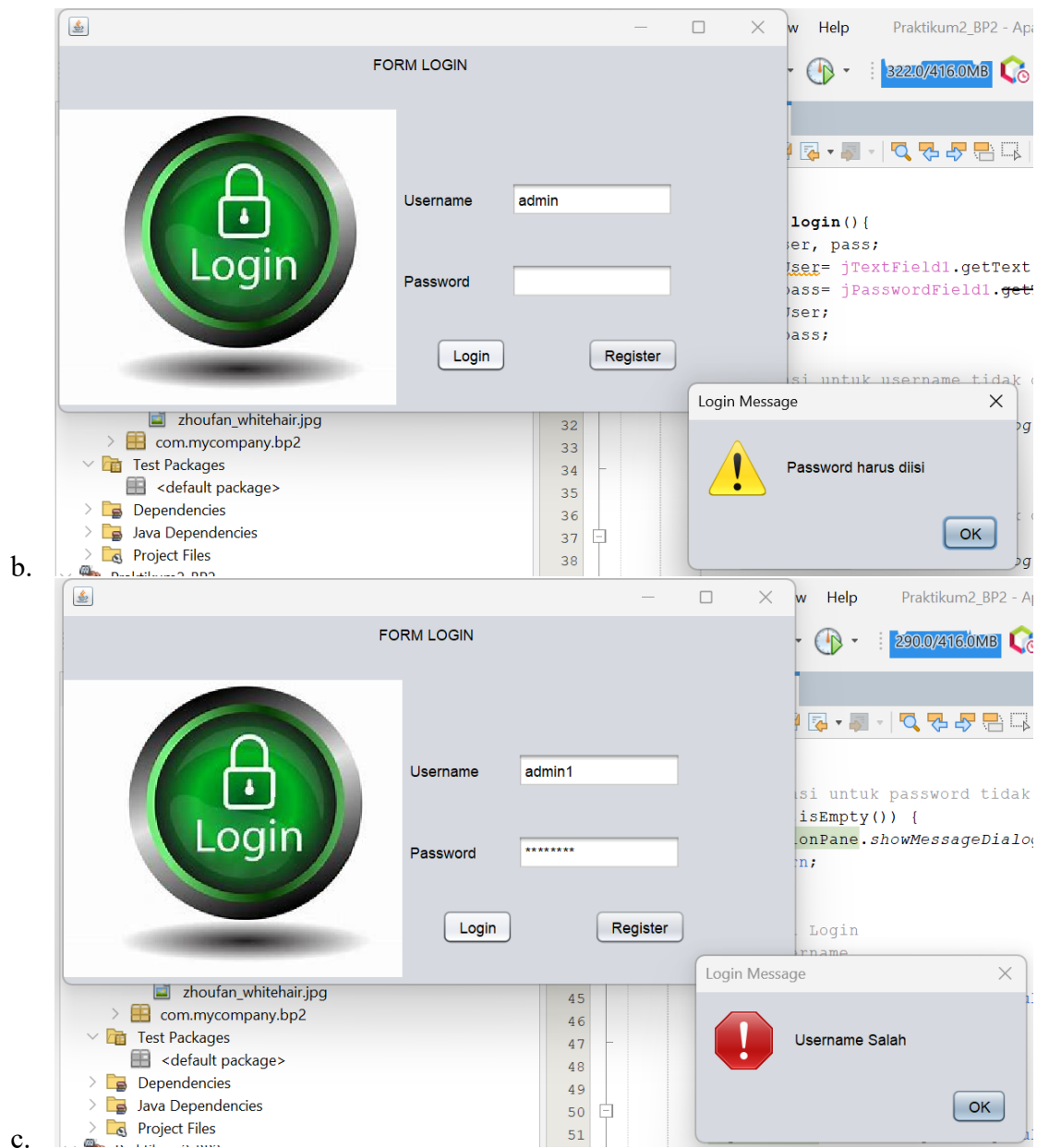
// Cek Password
if (!pass.equals("admin123")) {
    JOptionPane.showMessageDialog(null, "Password Salah", "Login Message",
JOptionPane.ERROR_MESSAGE);

return;
}

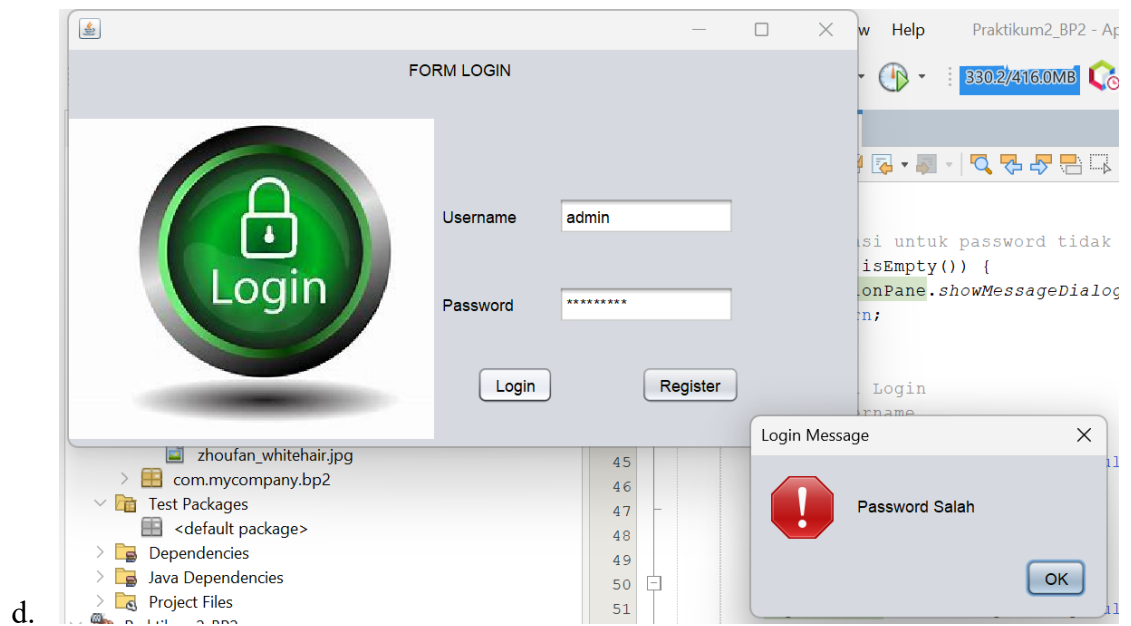
// Jika semua benar
JOptionPane.showMessageDialog(null, "Login Sukses Sebagai " + user,
    "Login Message", JOptionPane.INFORMATION_MESSAGE);
}

```









Analisis : Pada kode login ini, proses validasi dilakukan secara berurutan dan sederhana. Pertama, sistem memeriksa apakah field username dan password diisi. Jika salah satu kosong, maka akan muncul peringatan untuk mengisi field tersebut. Setelah itu, username divalidasi dengan membandingkan input dengan nilai tetap "admin". Jika tidak cocok, langsung muncul pesan "Username Salah". Hal yang sama dilakukan untuk password, yang harus sama dengan "admin123". Jika kedua input benar, maka akan ditampilkan pesan sukses login.

## BAB IV

### TUGAS

1. Lengkapi Form Login di atas dengan Menambahkan Level untuk Login menggunakan JComboBox. Adapun untuk level terdiri dari 2 level, yaitu ; Admin dan User dengan Tampilan sebagai berikut :



Script Program :

```
private void login(){
    String user = jTextField1.getText().trim();
    String pass = new String(jPasswordField1.getPassword()).trim();
    String level = jComboBox1.getSelectedItem().toString();

    // Validasi untuk input kosong
    if (user.isEmpty()) {
        JOptionPane.showMessageDialog(null, "Username harus diisi", "Login
Message", JOptionPane.WARNING_MESSAGE);
        return;
    }

    if (pass.isEmpty()) {
        JOptionPane.showMessageDialog(null, "Password harus diisi", "Login
Message", JOptionPane.WARNING_MESSAGE);
        return;
    }

    if (level.equals("Pilih Level")) {
        JOptionPane.showMessageDialog(null, "Level harus dipilih", "Login
Message", JOptionPane.WARNING_MESSAGE);
        return;
    }

    // Validasi Login
    if (user.equals("admin") && pass.equals("admin123") &&
level.equals("Admin")) {
        JOptionPane.showMessageDialog(null, "Login sukses sebagai Admin",
"Login Message", JOptionPane.INFORMATION_MESSAGE);
    } else if (user.equals("user") && pass.equals("user123") &&
level.equals("User")) {
```

```

JOptionPane.showMessageDialog(null, "Login sukses sebagai User",
"Login Message", JOptionPane.INFORMATION_MESSAGE);
} else if (!(user.equals("admin") || user.equals("user"))) {
JOptionPane.showMessageDialog(null, "Username Salah", "Login
Message", JOptionPane.ERROR_MESSAGE);
} else if ((user.equals("admin") && !pass.equals("admin123")) ||
(user.equals("user") && !pass.equals("user123"))) {
JOptionPane.showMessageDialog(null, "Password Salah", "Login
Message", JOptionPane.ERROR_MESSAGE);
} else {
JOptionPane.showMessageDialog(null, "Level Salah", "Login Message",
JOptionPane.ERROR_MESSAGE);
}
}
}

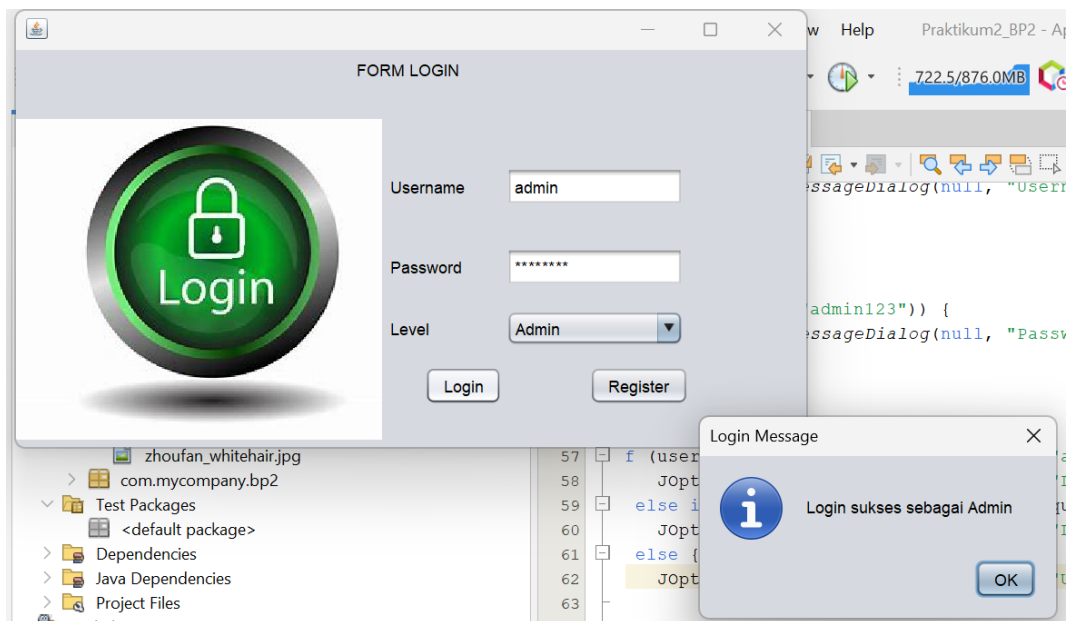
```

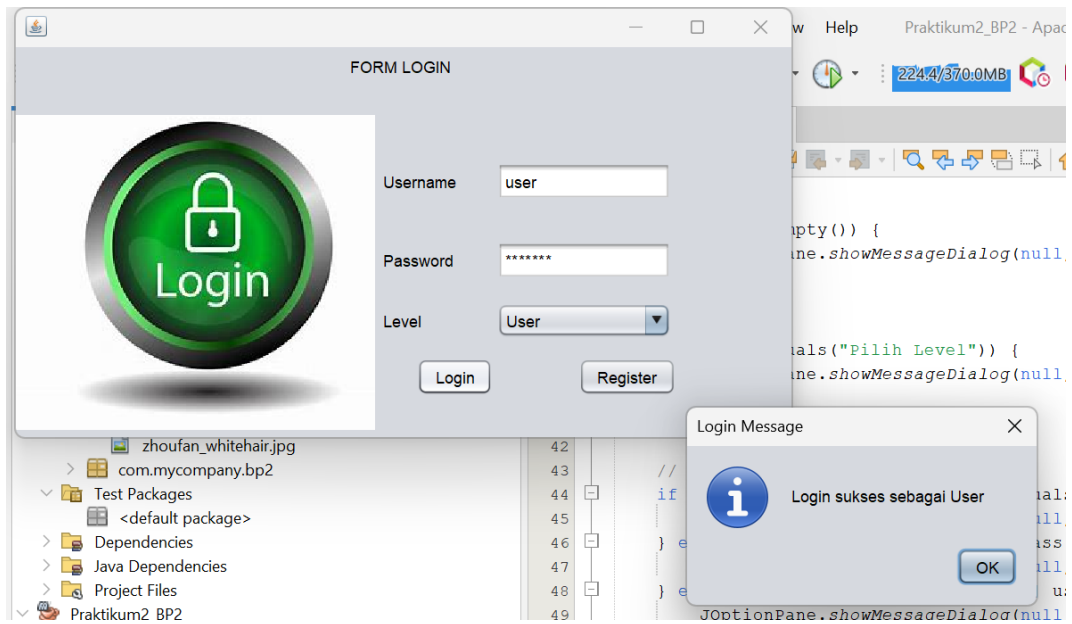
```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
login();
}

```

Masukkan di Tombol Login script diatas





Analisis : Program Login di atas dengan Menambahkan Level untuk Login menggunakan JComboBox. Adapun untuk level terdiri dari 2 level, yaitu ; Admin dan User dengan Validasi pertama dilakukan untuk memastikan bahwa input seperti username, password, dan level tidak kosong, agar pengguna langsung diberi peringatan sebelum melanjutkan proses login. Setelah input dipastikan lengkap, barulah dilakukan pengecekan kecocokan antara username, password, dan level secara menyeluruh dalam satu blok logika yang terpusat. Jika kombinasi tersebut benar, maka login dinyatakan berhasil sesuai dengan level yang dipilih. Namun jika terjadi kesalahan, program akan menampilkan pesan kesalahan yang lebih spesifik seperti "Username Salah" atau "Password Salah", sehingga pengguna tahu di bagian mana letak kesalahannya. Pendekatan ini membuat proses login lebih user-friendly dan logika program lebih rapi.

## **BAB V**

### **KESIMPULAN**

Dari rangkaian pembahasan kode login yang telah dijelaskan, dapat disimpulkan bahwa proses login dalam aplikasi Java GUI memerlukan logika validasi yang tertata dengan baik agar dapat memberikan pengalaman pengguna yang baik dan aman. Dimulai dari contoh sederhana yang hanya mengecek kombinasi username dan password secara langsung, kita melihat pentingnya penggunaan metode seperti `trim()` untuk menghindari kesalahan input akibat spasi, serta penggunaan `getPassword()` dari `JPasswordField` untuk keamanan yang lebih baik dibanding `getText()`. Validasi awal seperti memastikan input tidak kosong juga menjadi langkah penting agar sistem tidak langsung mengevaluasi data yang belum lengkap.

Selanjutnya, logika login yang lebih kompleks diperkenalkan, termasuk validasi berdasarkan level (seperti Admin atau User). Di sini, penggabungan pengecekan username, password, dan level login harus dilakukan secara hati-hati agar tidak terjadi tumpang tindih logika, seperti yang sempat terjadi dalam kode awal yang kurang tepat. Dengan perbaikan logika dan penempatan validasi yang sistematis, sistem login menjadi lebih akurat dalam memberikan pesan kesalahan dan memastikan kombinasi input sesuai peran yang diizinkan.

Secara keseluruhan, penerapan login di aplikasi GUI tidak hanya soal mencocokkan data, tetapi juga menyangkut pengalaman pengguna, keamanan input, serta struktur logika yang efisien. Dari contoh kode yang sudah diperbaiki, kita dapat memahami pentingnya memberikan feedback yang jelas kepada pengguna, menjaga urutan validasi, dan menghindari pengecekan yang redundan agar program berjalan optimal dan mudah dikembangkan ke tahap selanjutnya seperti pengelolaan data pengguna atau akses halaman berbeda sesuai level login.