

Laporan Praktikum Bahasa Pemrograman 1

Modul 4



Nama : Asep haryana saputra

NIM : 20230810043

Kelas : TINFC-2023-04

**Teknik Informatika
Fakultas Ilmu Komputer
Universitas Kuningan**

Pretest

1. Apa itu Swing Control?

Swing Control adalah komponen dari *Java Swing*, yang merupakan bagian dari Java Foundation Classes (JFC) untuk membangun antarmuka pengguna grafis (GUI). Swing menyediakan elemen UI seperti tombol, label, *text field*, dan panel. Swing bersifat platform-independen, artinya tampilan komponen dapat disesuaikan tanpa bergantung pada sistem operasi.

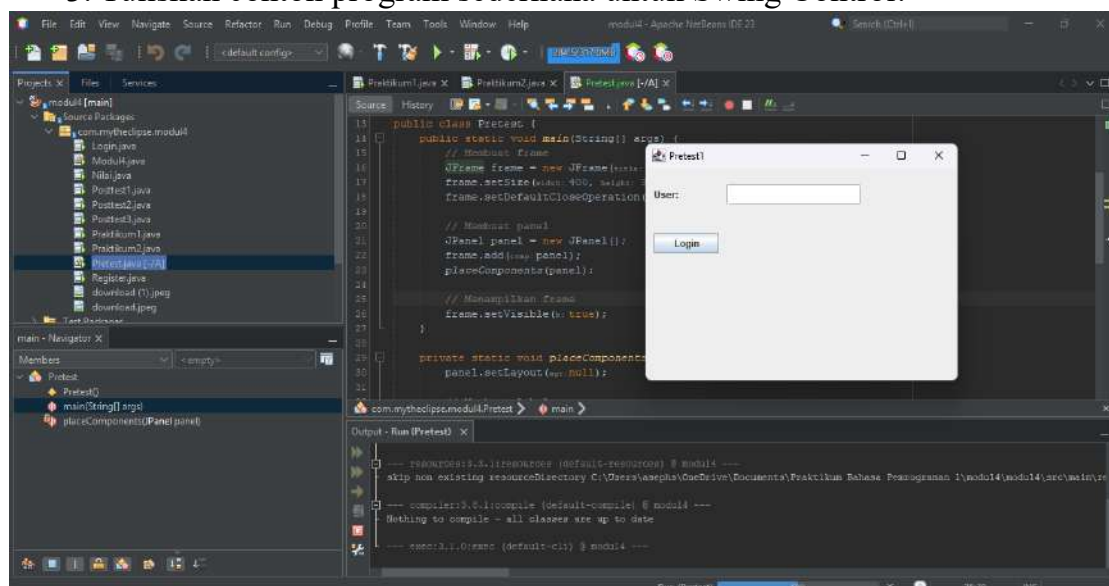
2. Apa fungsi dari Swing Control?

Fungsi utama dari Swing Control adalah memungkinkan pengembang membuat aplikasi GUI yang interaktif dengan menyediakan berbagai komponen antarmuka yang dapat digunakan, seperti:

- Menampilkan informasi (misalnya, dengan JLabel atau JTable).
- Mengumpulkan input pengguna (misalnya, dengan JButton, JTextField, atau JComboBox).
- Memungkinkan pengguna berinteraksi dengan aplikasi (misalnya, dengan JMenuBar atau JToolBar).

Swing Control juga mendukung berbagai pengaturan tata letak dan gaya kustom untuk memberikan pengalaman pengguna yang optimal.

3. Tuliskan contoh program sederhana untuk Swing Control!



Kode:

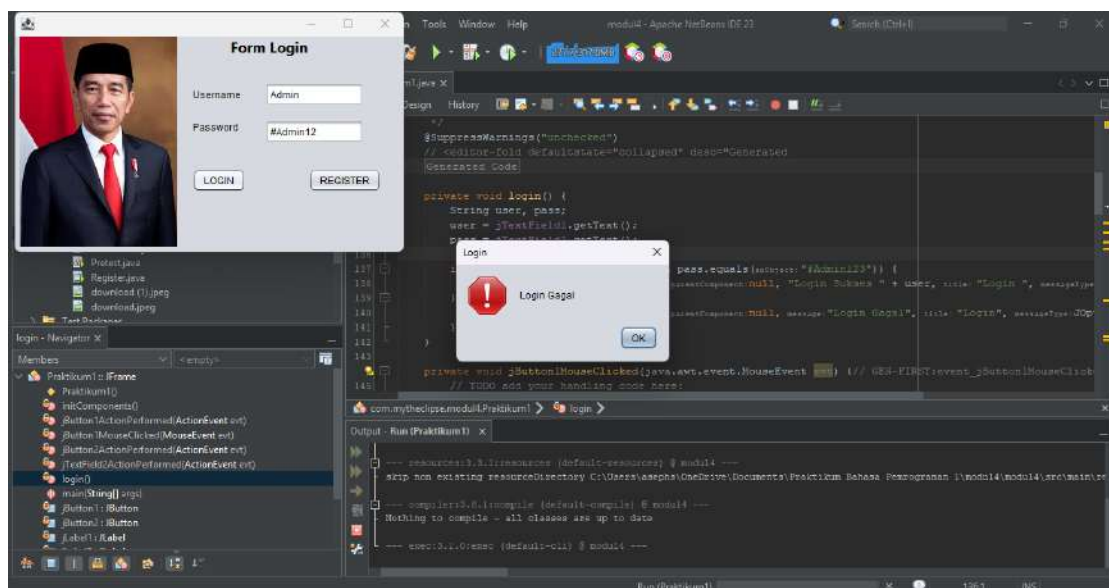
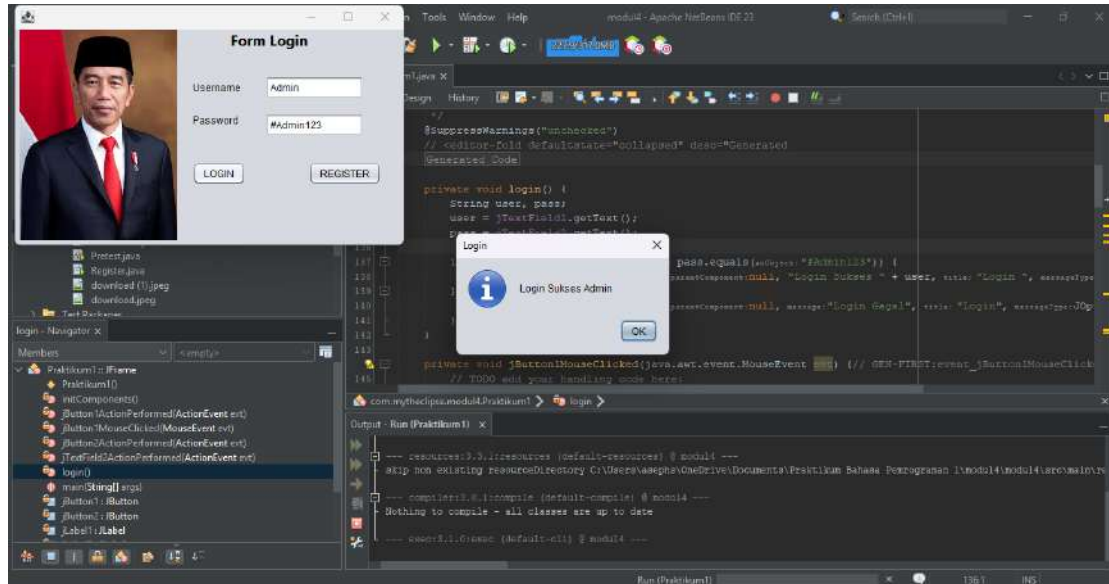
<https://github.com/MythEclipse/Praktikum-Bahasa-Pemrograman-1/blob/main/modul4/modul4/src/main/java/com/mytheclipse/modul4/Pretest.java>

Penjelasan:

1. Program membuat jendela utama menggunakan JFrame.
2. Komponen seperti JLabel, JTextField, dan JButton ditambahkan ke dalam JPanel.
3. JButton memiliki *event listener* untuk menangkap aksi pengguna saat tombol ditekan.
4. Pesan muncul menggunakan JOptionPane ketika tombol ditekan.

Praktikum

1.



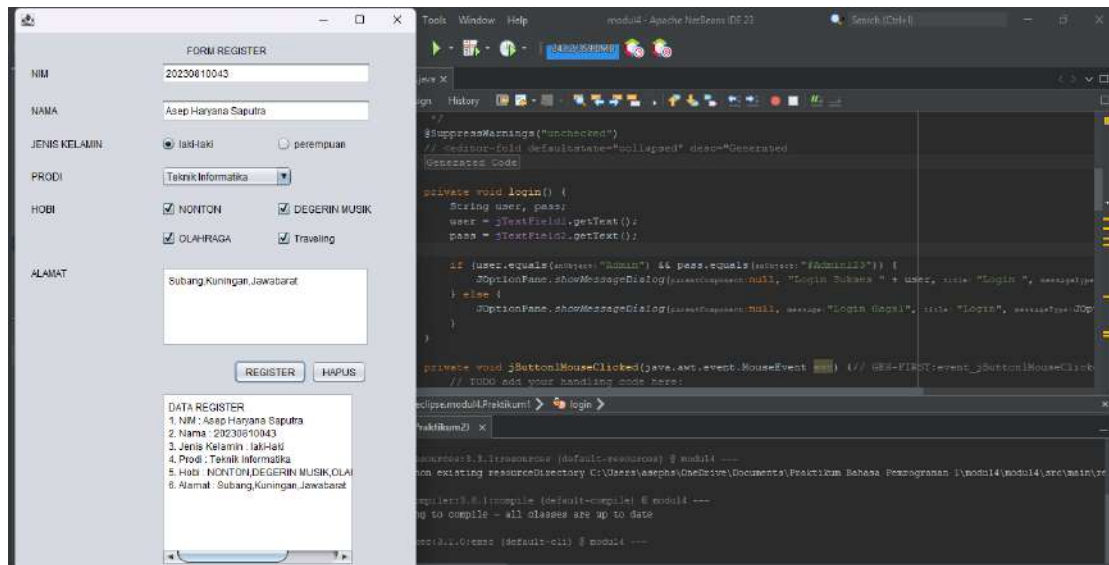
Kode:

<https://github.com/MythEclipse/Praktikum-Bahasa-Pemrograman-1/blob/main/modul4/modul4/src/main/java/com/mytheclipse/modul4/Praktikum1.java>

Penjelasan:

Metode **login** ini merupakan implementasi sederhana dari proses autentikasi pengguna. Ini mengecek kredensial pengguna dan memberikan umpan balik melalui dialog. Jika kredensial yang dimasukkan benar, pengguna akan mendapatkan pesan sukses; jika tidak, mereka akan mendapatkan pesan gagal. Kode ini adalah contoh umum dari logika login yang sering digunakan dalam aplikasi desktop berbasis Java menggunakan Swing.

2.



Kode:

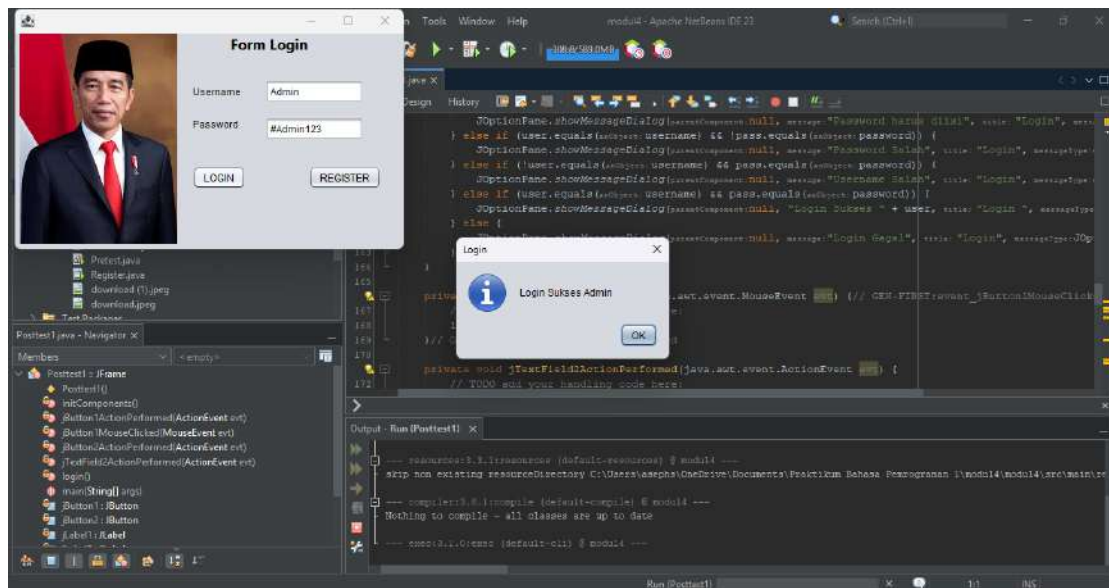
<https://github.com/MythEclipse/Praktikum-Bahasa-Pemrograman-1/blob/main/modul4/modul4/src/main/java/com/mytheclipse/modul4/Praktikum2.java>

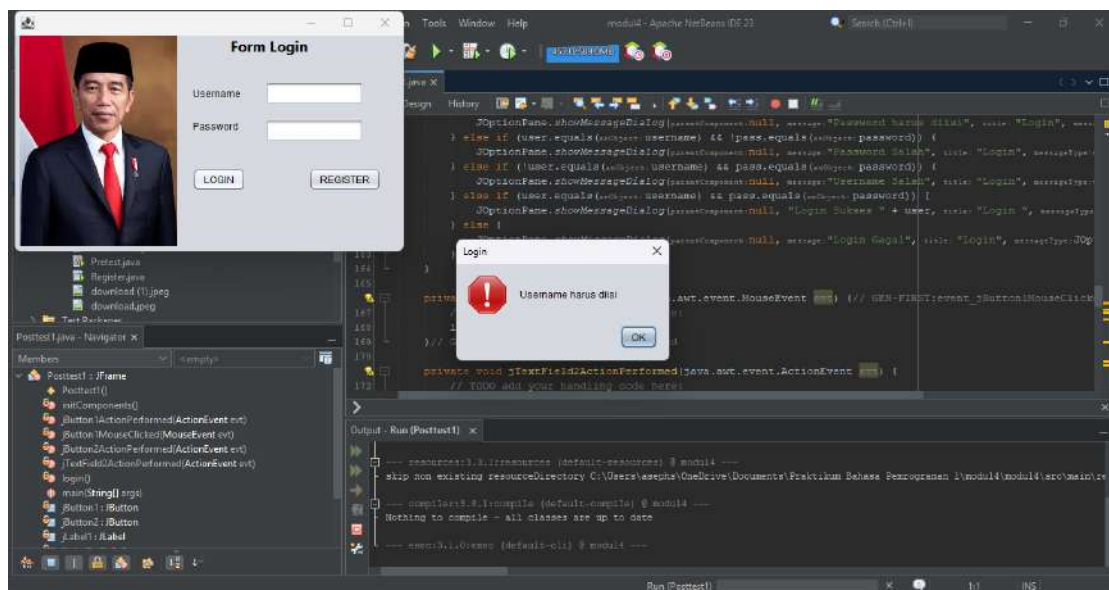
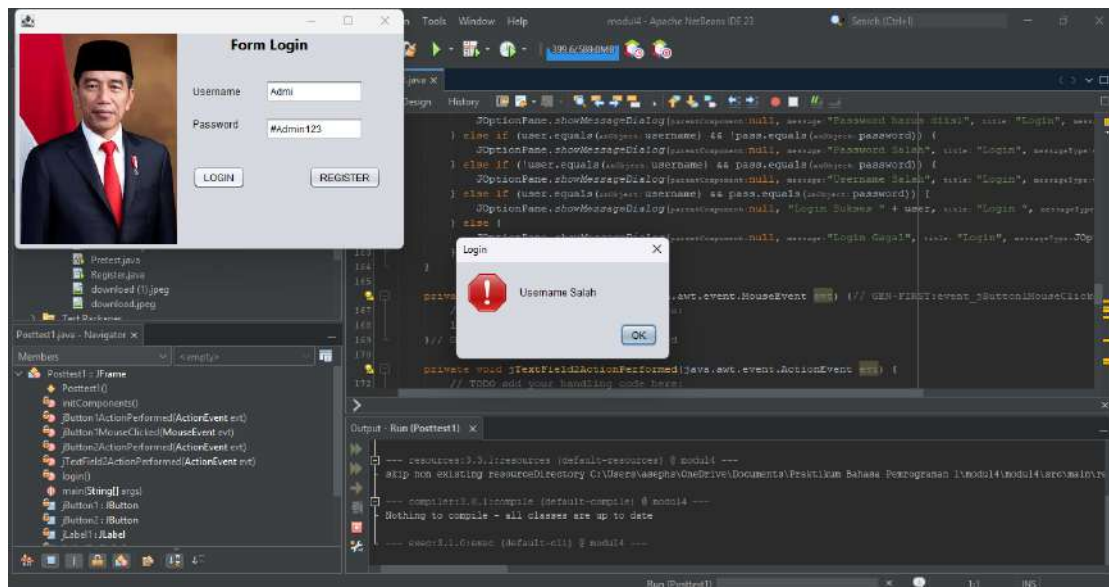
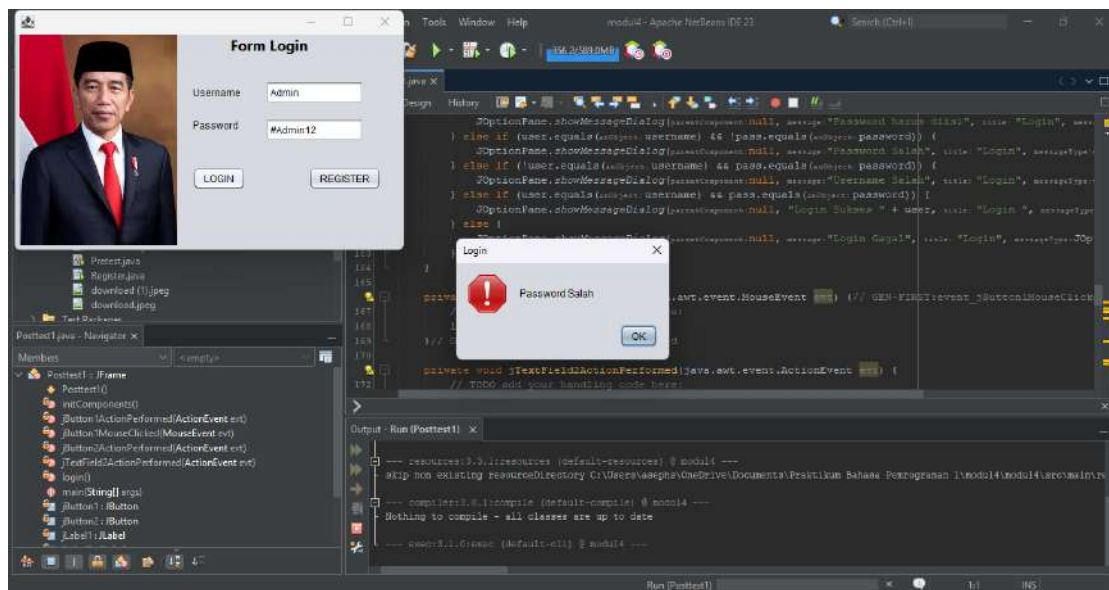
Penjelasan:

Metode **simpan()** ini adalah contoh dari pengumpulan dan pengolahan data yang diambil dari berbagai komponen GUI dalam aplikasi Java. Metode ini mengumpulkan informasi seperti NIM, nama, jenis kelamin, program studi, hobi, dan alamat, dan kemudian menampilkannya dalam format yang mudah dibaca. Ini adalah praktik umum dalam aplikasi berbasis GUI untuk memproses input dari pengguna dan memberikan umpan balik.

PostTest

1.





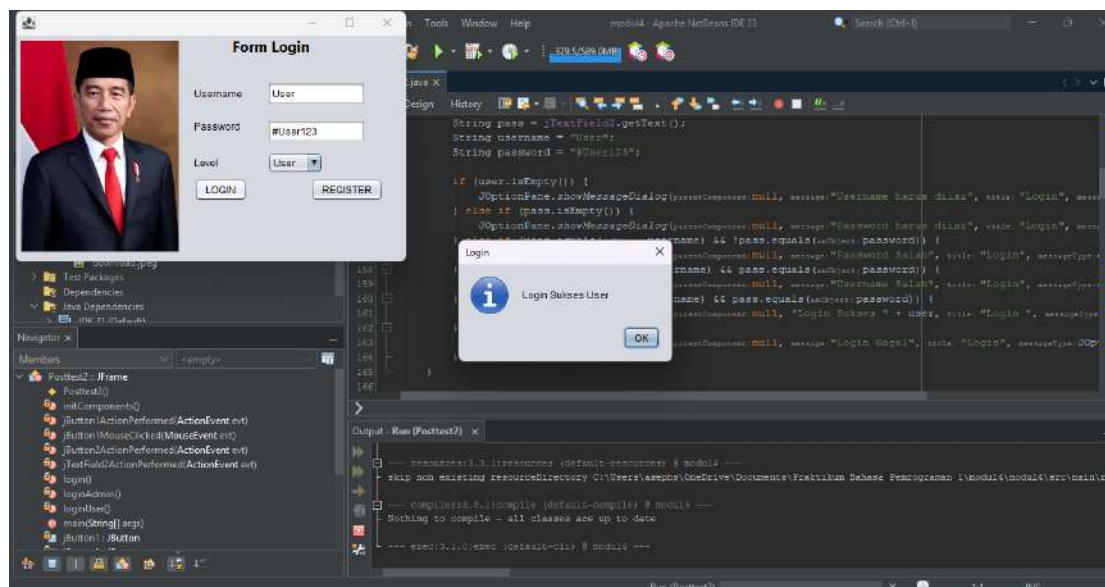
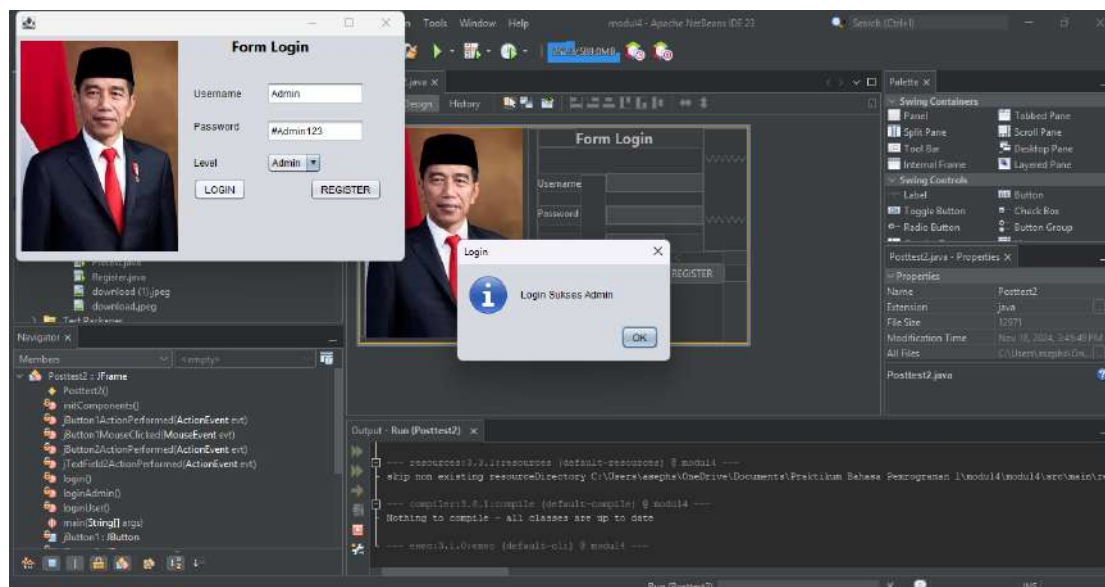
Kode:

<https://github.com/MythEclipse/Praktikum-Bahasa-Pemrograman-1/blob/main/modul4/modul4/src/main/java/com/mytheclipse/modul4/Posttest1.java>

Penjelasan:

Metode **login()** ini adalah implementasi dari proses autentikasi pengguna yang lebih lengkap dibandingkan dengan contoh sebelumnya. Metode ini tidak hanya memeriksa apakah username dan password benar, tetapi juga memberikan umpan balik yang jelas kepada pengguna jika ada yang salah, seperti jika username atau password tidak diisi atau salah. Ini adalah praktik yang baik dalam pengembangan aplikasi untuk memastikan bahwa pengguna mendapatkan informasi yang tepat tentang status login mereka.

2.



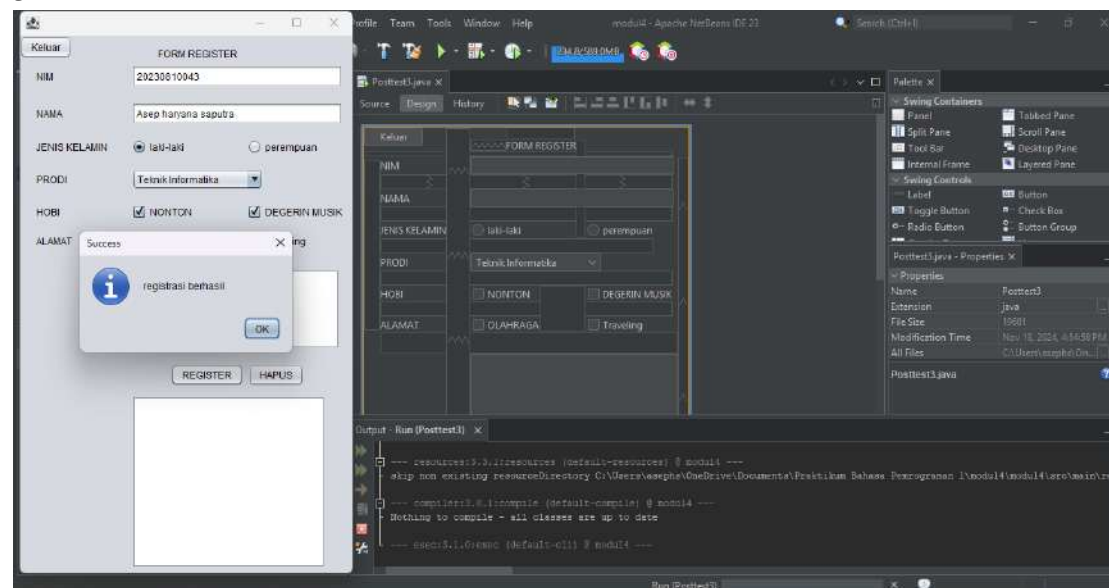
Kode:

<https://github.com/MythEclipse/Praktikum-Bahasa-Pemrograman-1/blob/main/modul4/modul4/src/main/java/com/mytheclipse/modul4/Posttest2.java>

Penjelasan:

"User" dan "Admin". Terdapat tiga metode utama dalam kode ini: **loginUser()**, **loginAdmin()**, dan **login()**. Metode **loginUser()** bertugas untuk memproses login pengguna biasa dengan memeriksa kredensial yang dimasukkan melalui dua **JTextField**. Jika username atau password kosong, atau jika salah satu dari keduanya tidak sesuai dengan nilai yang telah ditentukan, pengguna akan diberikan umpan balik yang sesuai melalui dialog pesan. Demikian pula, metode **loginAdmin()** melakukan fungsi yang sama tetapi untuk pengguna dengan akses Admin, menggunakan kredensial yang berbeda. Metode **login()** berfungsi sebagai pengontrol utama yang menentukan jenis pengguna yang mencoba untuk login berdasarkan pilihan yang diambil dari **JComboBox**. Jika pengguna memilih "Admin", maka metode **loginAdmin()** akan dipanggil, sementara jika memilih "User", maka metode **loginUser()** yang akan dijalankan.

3.



Kode:

<https://github.com/MythEclipse/Praktikum-Bahasa-Pemrograman-1/blob/main/modul4/modul4/src/main/java/com/mytheclipse/modul4/Posttest3.java>

Penjelasan:

Metode **hapus()** bertugas untuk mengatur ulang semua komponen input di antarmuka pengguna ke keadaan awal. Ini dilakukan dengan mengosongkan teks pada **JTextField**, mengatur radio button dan checkbox ke posisi tidak terpilih, serta mengatur pilihan pada **JComboBox** ke indeks pertama. Dengan cara ini, pengguna dapat memulai proses pengisian data baru tanpa terpengaruh oleh informasi sebelumnya.

Metode **simpan()** berfungsi untuk mengumpulkan data dari berbagai komponen antarmuka pengguna. Data yang diambil mencakup NIM dan nama dari **JTextField**, jenis kelamin dari dua radio button, program studi dari **JComboBox**, hobi dari beberapa checkbox, dan alamat dari **JTextArea**. Setelah mengumpulkan semua informasi, metode ini menyusun string yang berisi data registrasi dan menampilkannya di **JTextArea** kedua.

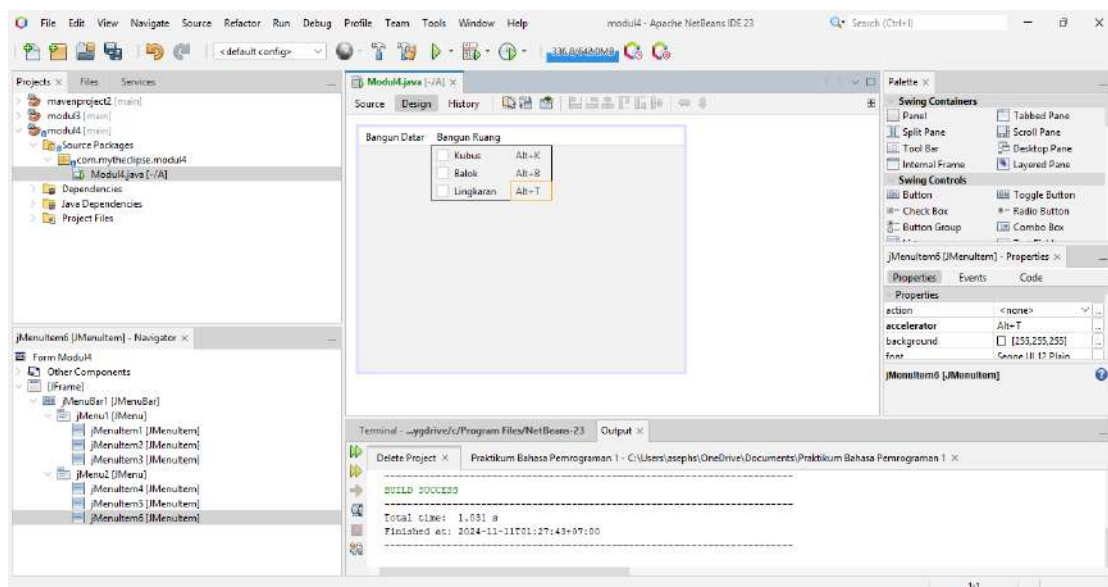
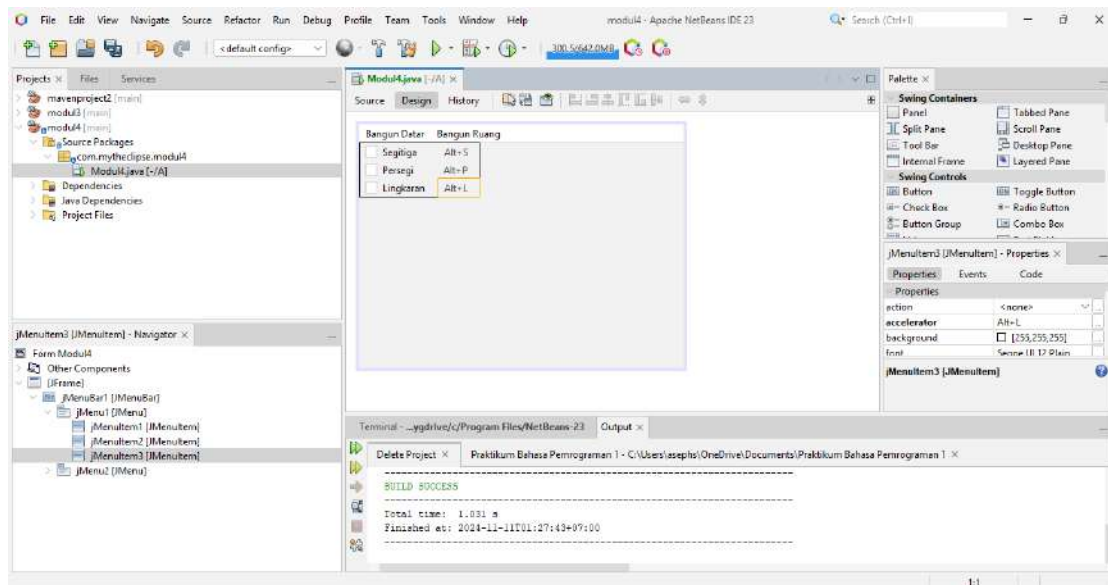
Penggunaan **StringBuffer** untuk hobi memungkinkan penggabungan beberapa pilihan hobi menjadi satu string dengan format yang rapi.

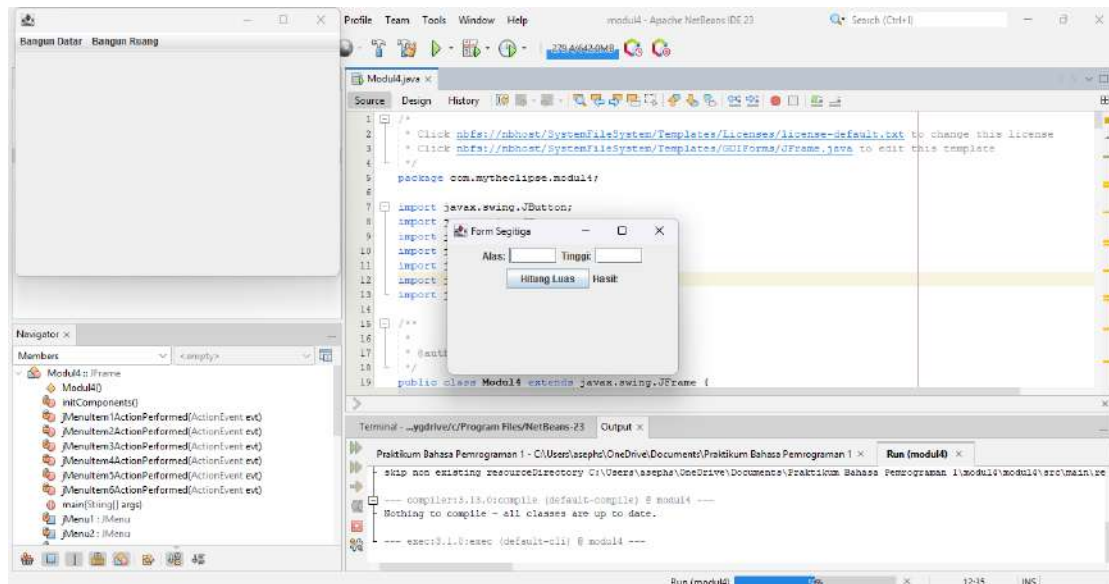
Metode **REGISTER()** adalah inti dari proses registrasi. Metode ini juga mengumpulkan data dengan cara yang sama seperti **simpan()**, tetapi menambahkan logika untuk memeriksa

kelengkapan data. Jika semua data penting (NIM, nama, jenis kelamin, program studi, hobi, dan alamat) tidak diisi, maka aplikasi akan menampilkan pesan kesalahan melalui **JOptionPane**, menginformasikan pengguna bahwa registrasi gagal karena data tidak lengkap. Jika semua data sudah diisi, pengguna akan diberi tahu bahwa registrasi berhasil, dan metode **simpan()** akan dipanggil untuk menyimpan data. Setelah itu, semua komponen input akan diatur ulang ke keadaan kosong, sehingga siap untuk registrasi berikutnya.

Tugas

1





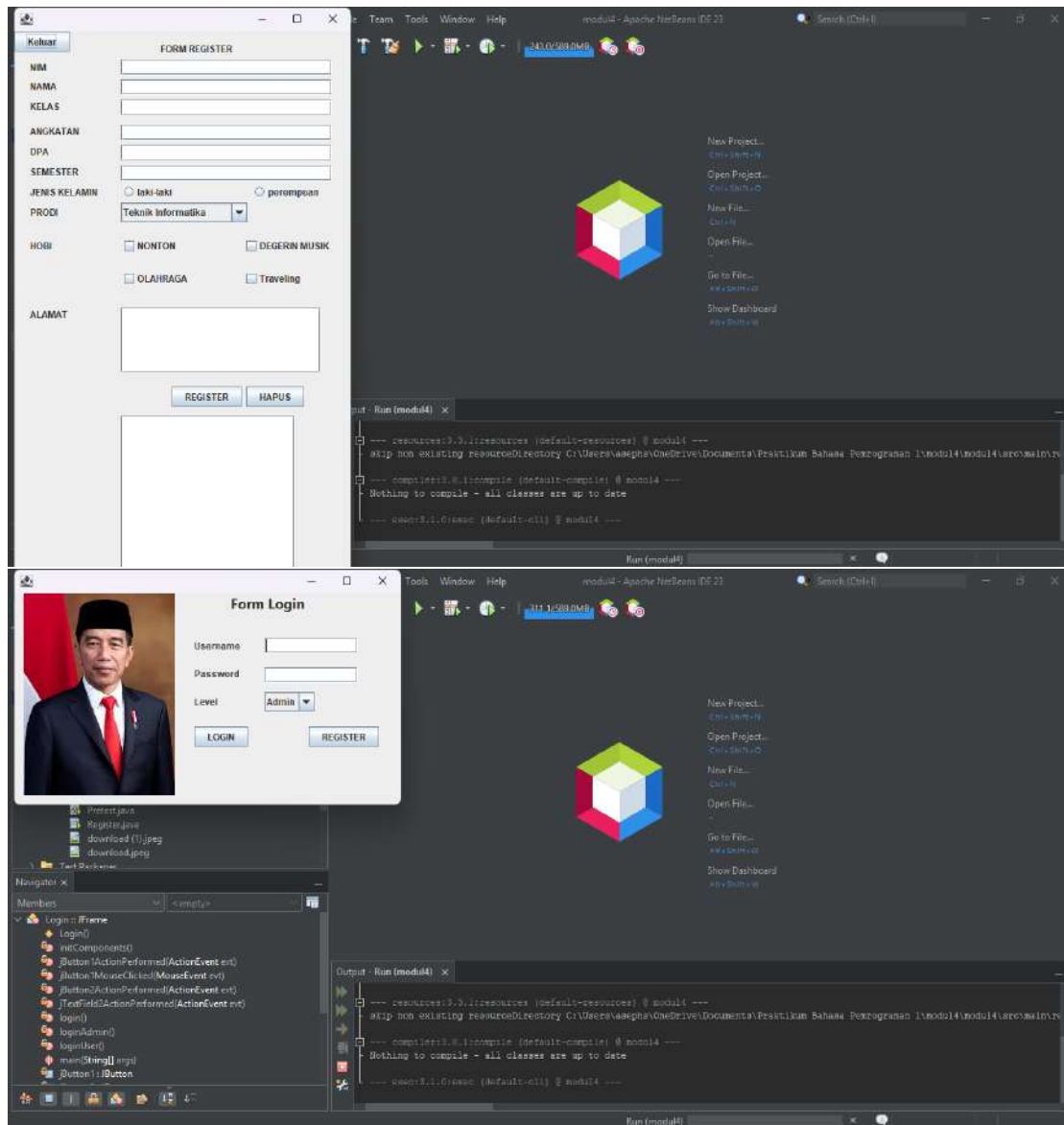
Kode:

<https://github.com/MythEclipse/Praktikum-Bahasa-Pemrograman-1/tree/main/modul4/modul4/src/main/java/com/mytheclipse/modul4>

Penjelasan:

Saat program dijalankan, antarmuka utama aplikasi ditampilkan dengan dua menu: "Bangun Datar" dan "Bangun Ruang". Pengguna dapat memilih item menu yang diinginkan, seperti Segitiga, Persegi, Lingkaran, Kubus, atau Balok. Ketika item menu dipilih, program memanggil ActionListener terkait yang membuka jendela/form baru untuk perhitungan geometris sesuai pilihan.

2.



Kode:

<https://github.com/MythEclipse/Praktikum-Bahasa-Pemrograman-1/blob/main/modul4/modul4/src/main/java/com/mytheclipse/modul4/Login.java>

<https://github.com/MythEclipse/Praktikum-Bahasa-Pemrograman-1/blob/main/modul4/modul4/src/main/java/com/mytheclipse/modul4/Register.java>

<https://github.com/MythEclipse/Praktikum-Bahasa-Pemrograman-1/blob/main/modul4/modul4/src/main/java/com/mytheclipse/modul4/Nilai.java>