

**TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK**

MODUL XV



**Disusun Oleh :
Atika Aji Hadiyani
2211104003
SE-0601**

Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs.

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
TELKOM UNIVERSITY PURWOKERTO**

TUGAS JURNAL XV

1. MEMBUAT PROJECT MODUL

Buka IDE misalnya dengan Visual Studio

- A. Copy salah satu folder tugas jurnal yang dimiliki sebelumnya (dari modul 2 sampai modul 13), kemudian rename folder hasil copy-paste tersebut dengan modul15_NIM (coba pilih tugas pendahuluan yang paling sederhana)
- B. Misalnya menggunakan Visual Studio, bukalah project/folder yang di-copy sebelumnya.

2. PENGEMBANGAN DENGAN SECURE CODING PRACTICES

Buatlah aplikasi desktop dengan fitur:

1. Registrasi user dengan input username dan password
2. Penyimpanan data user pada file json
3. Login user

Dengan mengikuti Secure Coding Practices yang memenuhi faktor-faktor berikut:

A. Input Validation (wajib mengimplementasikan salah satu, diizinkan lebih)

a. Validasi range data

Range data input harus dibatasi dan ditetapkan. Contoh:

- Hanya boleh huruf alfabet ASCII
- Harus mengandung angka

b. Validasi panjang data

- Panjang atau ukuran data harus dibatasi dan ditetapkan. Contoh:
- Minimal 8 karakter
- Maksimal 20 karakter

c. Handling data invalid

Data yang tidak valid harus ditolak atau dihandle dengan jelas (jangan dibiarkan menjadi runtime error yang tidak dihandle). Contoh:

- Jika terdapat aturan minimal 8 karakter input, input di bawah 8 karakter harus ditolak atau dihandle dengan spesifik
- Jika terdapat aturan hanya boleh huruf alfabet ASCII, input dengan karakter selain alfabet ASCII harus ditolak atau dihandle dengan spesifik

B. Password Management (wajib mengimplementasikan salah satu, diizinkan lebih

a. Password hashing

Ketika sistem menyimpan password, password harus dienkripsi atau dihash.

Contoh:

- Sistem mengenkripsi password dengan metode hash SHA256
- Sistem mengubah password dengan konsisten supaya tidak sama persis dengan inputan user

b. Password rules

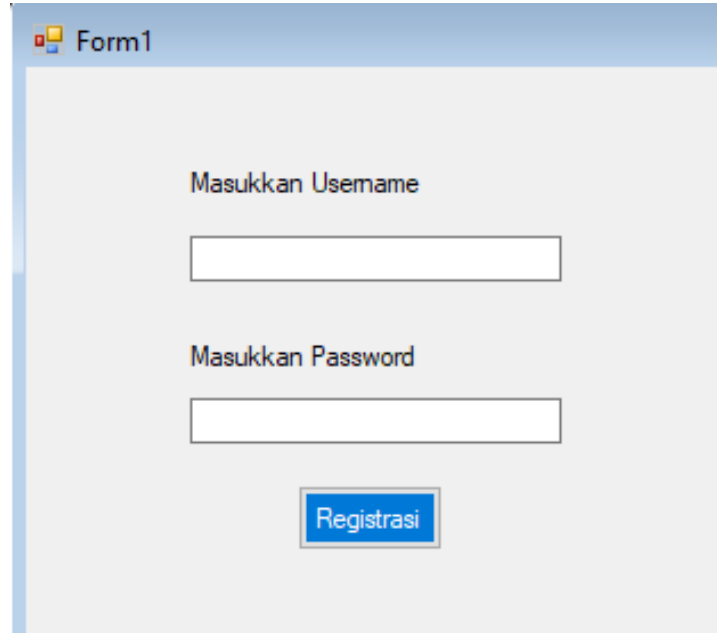
Ketika sistem harus menerima inputan password, harus ada aturan keamanan untuk password tersebut. Contoh:

- Password harus mengandung minimal 1 karakter unik (!@#\$\$%^&*)
- Password tidak boleh mengandung kata dari username

Jawab:

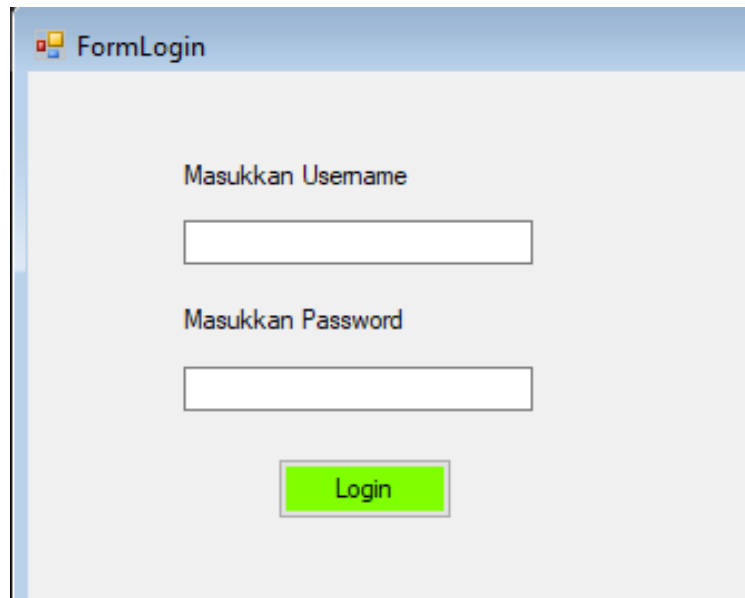
Source Code

a. Desain FormRegister.cs



The image shows a screenshot of a Windows application window titled "Form1". Inside the window, there is a registration form. It consists of two text input fields. The first field is labeled "Masukkan Username" and the second field is labeled "Masukkan Password". Below these fields is a blue button with the text "Registrasi" in white. The form has a light gray background and a blue border.

b. Desain FormLogin.cs



The image shows a screenshot of a Windows application window titled "FormLogin". The window has a light gray background and a blue title bar. Inside the window, there are two text labels: "Masukkan Usemame" (note the typo) and "Masukkan Password". Below each label is a white rectangular text box. At the bottom center of the window is a green button with the text "Login".

Masukkan Usemame

Masukkan Password

Login

c. File FormRegister.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace modul15_2211104003
{
    public partial class FormRegister: Form
    {
        public FormRegister()
        {
            InitializeComponent();
        }

        private void FormRegister_Load(object sender, EventArgs e)
        {
        }

        private void btnRegister_Click(object sender, EventArgs e)
        {
            string username = txtUsername.Text;
            string password = txtPassword.Text;

            if (!AuthHelper.IsValidInput(username, password, out string error))
            {
                MessageBox.Show(error, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }

            string hashed = AuthHelper.HashPassword(password);
            AuthHelper.SaveUser(new User { Username = username, PasswordHash = hashed });

            MessageBox.Show("Registrasi berhasil!", "Info", MessageBoxButtons.OK, MessageBoxIcon.Information);
            this.Hide();
            FormLogin loginForm = new FormLogin();
            loginForm.Show();
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }
    }
}
```

d. File FormLogin.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace modul15_2211104003
{
    public partial class FormLogin: Form
    {
        public FormLogin()
        {
            InitializeComponent();
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
            string username = txtUsernameLogin.Text;
            string password = txtPasswordLogin.Text;

            if (AuthHelper.CheckLogin(username, password))
            {
                MessageBox.Show("Login berhasil!", "Sukses", MessageBoxButtons.OK, MessageBoxIcon.Information);
                // TODO: buka form utama aplikasi
            }
            else
            {
                MessageBox.Show("Username atau password salah!", "Gagal", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void FormLogin_Load(object sender, EventArgs e)
        {
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }

        private void label1_Click_1(object sender, EventArgs e)
        {
        }
    }
}
```

e. File User.cs

```
namespace modul15_2211104003
{
    public class User
    {
        public string Username { get; set; }
        public string PasswordHash { get; set; }
    }
}
```

Screenshoot Output

a. Registrasi

The image displays three sequential screenshots of a registration form titled "Form1".

First Screenshot: The form contains two input fields: "Masukkan Username" and "Masukkan Password". Both fields are empty. A blue "Registrasi" button is located at the bottom.

Second Screenshot: The "Masukkan Username" field now contains the text "Atika". The "Masukkan Password" field contains ten black dots, indicating a masked password. The "Registrasi" button remains at the bottom.

Third Screenshot: This screenshot shows the same form with the username "Atika" and masked password. An "Info" dialog box is overlaid on the bottom half of the form. The dialog box has a title bar "Info" and a close button. It contains an information icon (a blue circle with a white 'i') and the text "Registrasi berhasil!". At the bottom of the dialog box is an "OK" button.

b. Login

The image displays three sequential screenshots of a web application's login interface, titled "FormLogin".

First Screenshot: The login form is shown with empty input fields. The labels "Masukkan Ustream" and "Masukkan Password" are positioned above the respective text boxes. A green "Login" button is located at the bottom.

Second Screenshot: The user "Atika" has entered the username field, and the password field is filled with masked characters (dots). The "Login" button remains visible.

Third Screenshot: After clicking the "Login" button, a modal dialog box titled "Sukses" appears. It contains an information icon (i) and the message "Login berhasil!". An "OK" button is at the bottom of the dialog.

Penjelasan

Tiga file utama, yaitu *FormRegister.cs*, *FormLogin.cs*, dan *User.cs*, saling berperan dalam membangun fitur login dan registrasi dasar pada aplikasi Windows Forms. File *FormRegister.cs* berfungsi untuk proses pendaftaran akun baru. Ketika pengguna menekan tombol Register, sistem akan melakukan validasi terhadap data yang dimasukkan. Validasi ini mencakup pengecekan apakah username hanya terdiri dari huruf alfabet, apakah password memiliki panjang yang sesuai (minimal 8 dan maksimal 20 karakter), mengandung angka, serta menyertakan karakter khusus seperti `!@#$$%^&*`. Selain itu, sistem juga memverifikasi apakah password yang digunakan tidak terlalu mirip dengan username. Jika ditemukan kesalahan, sistem akan memberikan pesan yang spesifik untuk memudahkan pengguna memperbaiki input-nya. Bila semua data lolos validasi, password akan diubah menjadi bentuk hash menggunakan algoritma SHA256 demi keamanan—sehingga password tidak disimpan dalam bentuk aslinya—dan disimpan ke dalam sistem.

Di sisi lain, *FormLogin.cs* bertugas memproses autentikasi pengguna yang telah terdaftar. Saat login, sistem membandingkan input username dan password dengan data yang sudah tersimpan. Karena password disimpan dalam bentuk hash, input password akan di-hash terlebih dahulu agar dapat dibandingkan dengan benar. Jika hasilnya sesuai, sistem akan menampilkan notifikasi bahwa login berhasil. Sebaliknya, bila tidak cocok, sistem akan menampilkan pesan kesalahan yang menjelaskan masalahnya. Adapun file *User.cs* berfungsi sebagai representasi struktur data pengguna, yang menyimpan informasi username dan password yang telah melalui proses hashing.