JURNAL
KONSTRUKSI PERANGKAT LUNAK

PERTEMUAN 15

Disusun Oleh :

Muhammad Abdul Aziz

2211104026

SE0601

Asisten Praktikum :

Naufal El Kamil Aditya Pratama Rahman

Imelda

Dosen Pengampu :

Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING

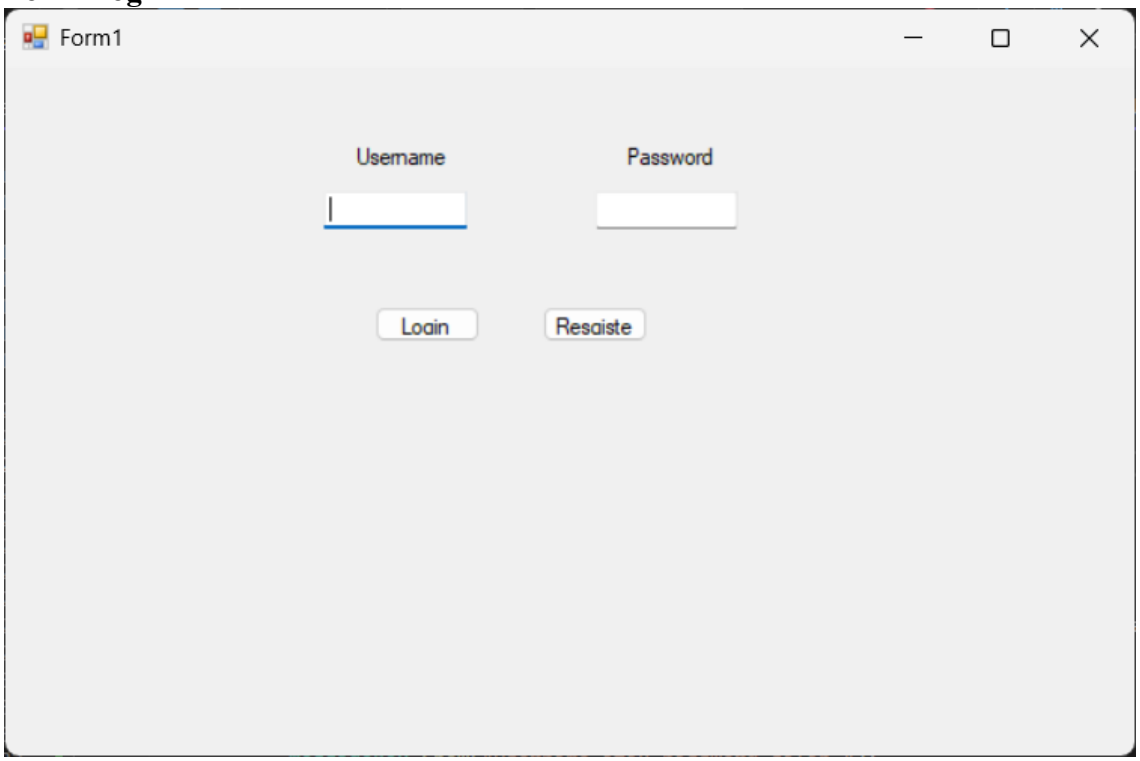
FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

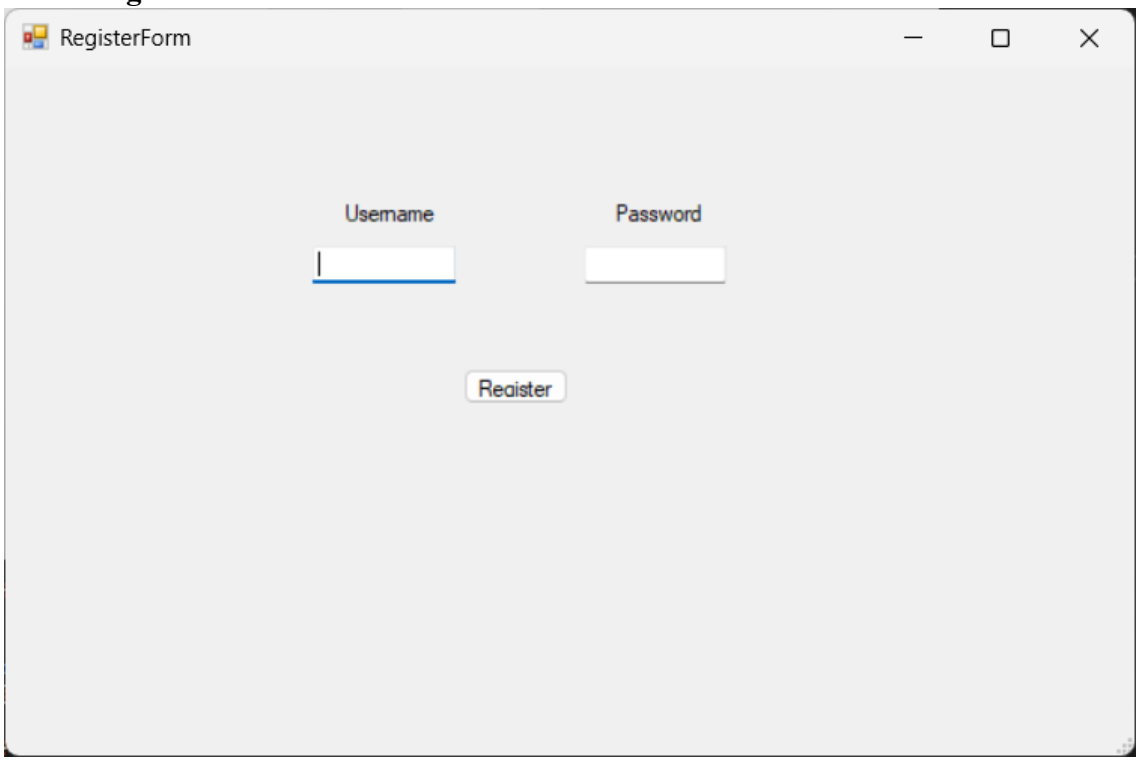
2025

1. **Hasil Run**

Form Login



Form Register



2. **Penjelasan singkat dari kode implementasi yang dibuat (beserta screenshot dari potongan source code yang dijelaskan).**

Source Code

Form1.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_2211104026
{
    3 references
    public partial class Form1: Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void btnLogin_Click(object sender, EventArgs e)
        {
            string username = txtUsername.Text.Trim();
            string password = txtPassword.Text;
            string hashed = Helper.ComputeSha256Hash(password);

            var users = UserStorage.LoadUsers();

            var user = users.FirstOrDefault(u => u.Username == username && u.PasswordHash == hashed);

            if (user != null)
                MessageBox.Show("Login sukses!");
            else
                MessageBox.Show("Username atau password salah.");
        }

        1 reference
        private void btnRegister_Click(object sender, EventArgs e)
        {
            new RegisterForm().ShowDialog();
        }

        1 reference
        private void txtUsername_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

Register.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_2211104026
{
    3 references
    public partial class RegisterForm: Form
    {
        1 reference
        public RegisterForm()
        {
            InitializeComponent();
        }

        1 reference
        private void btnSubmitReg_Click(object sender, EventArgs e)
        {
            string username = txtUsernameReg.Text.Trim();
            string password = txtPasswordReg.Text;

            if (!Helper.IsValidInput(username, password, out string error))
            {
                MessageBox.Show(error);
                return;
            }

            var users = UserStorage.LoadUsers();

            if (users.Any(u => u.Username == username))
            {
                MessageBox.Show("Username sudah terdaftar.");
                return;
            }

            string hashed = Helper.ComputeSha256Hash(password);
            users.Add(new User { Username = username, PasswordHash = hashed });
            UserStorage.SaveUsers(users);

            MessageBox.Show("Registrasi berhasil!");
            this.Close();
        }
    }
}
```

User.cs

```
public class User
{
    3 references
    public string Username { get; set; }
    2 references
    public string PasswordHash { get; set; }
}
```

Helper.cs

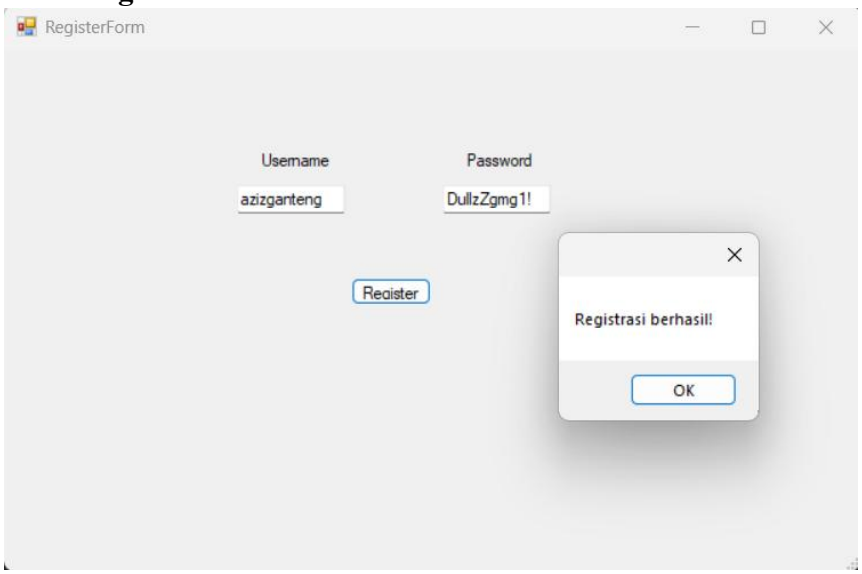
```
using System.Linq;
using System.Security.Cryptography;
using System.Text;

3 references
public static class Helper
{
    2 references
    public static string ComputeSha256Hash(string rawData)
    {
        using (SHA256 sha256 = SHA256.Create())
        {
            byte[] bytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(rawData));
            StringBuilder builder = new StringBuilder();
            foreach (var b in bytes)
                builder.Append(b.ToString("x2"));
            return builder.ToString();
        }
    }

    1 reference
    public static bool IsValidInput(string username, string password, out string error)
    {
        error = "";

        // Validasi panjang
        if (username.Length < 8 || username.Length > 20)
        {
            error = "Username harus 8-20 karakter.";
            return false;
        }

        if (password.Length < 8 || password.Length > 20)
        {
            error = "Password harus 8-20 karakter.";
            return false;
        }

        // Validasi karakter ASCII
        if (!username.All(c => c >= 32 && c <= 126))
        {
            error = "Username hanya boleh karakter ASCII.";
            return false;
        }

        // Password mengandung angka
        if (!password.Any(char.IsDigit))
        {
            error = "Password harus mengandung angka.";
            return false;
        }

        // Password harus punya karakter unik
        if (!password.Any(ch => "!@#$%^&*".Contains(ch)))
        {
            error = "Password harus mengandung minimal 1 karakter unik (!@#$%^&*).";
            return false;
        }

        // Password tidak boleh mengandung username
        if (password.ToLower().Contains(username.ToLower()))
        {
            error = "Password tidak boleh mengandung username.";
            return false;
        }

        return true;
    }
}
```

UserStorage.cs

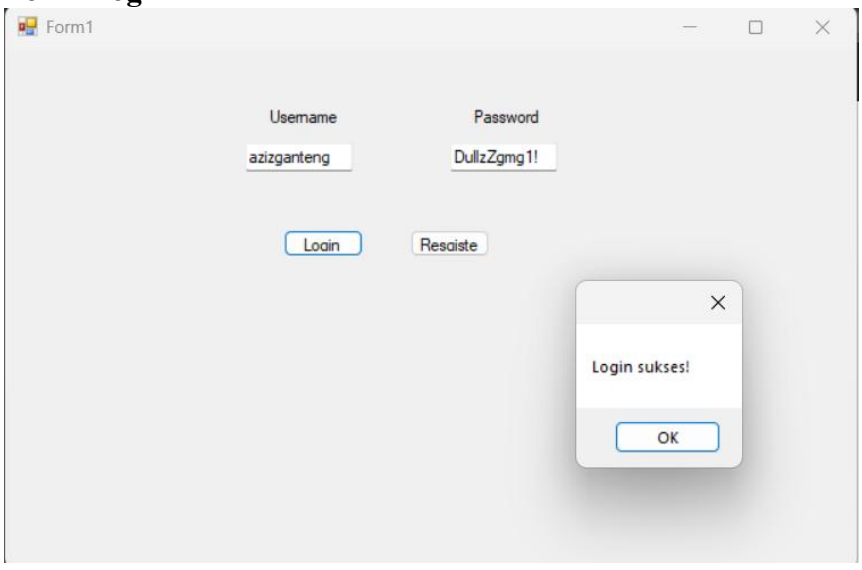
```
using Newtonsoft.Json;
using System.Collections.Generic;
using System.IO;

3 references
public static class UserStorage
{
    private static string path = "users.json";

    2 references
    public static List<User> LoadUsers()
    {
        if (!File.Exists(path)) return new List<User>();
        string json = File.ReadAllText(path);
        return JsonConvert.DeserializeObject<List<User>>(json) ?? new List<User>();
    }

    1 reference
    public static void SaveUsers(List<User> users)
    {
        string json = JsonConvert.SerializeObject(users, Formatting.Indented);
        File.WriteAllText(path, json);
    }
}
```

Output

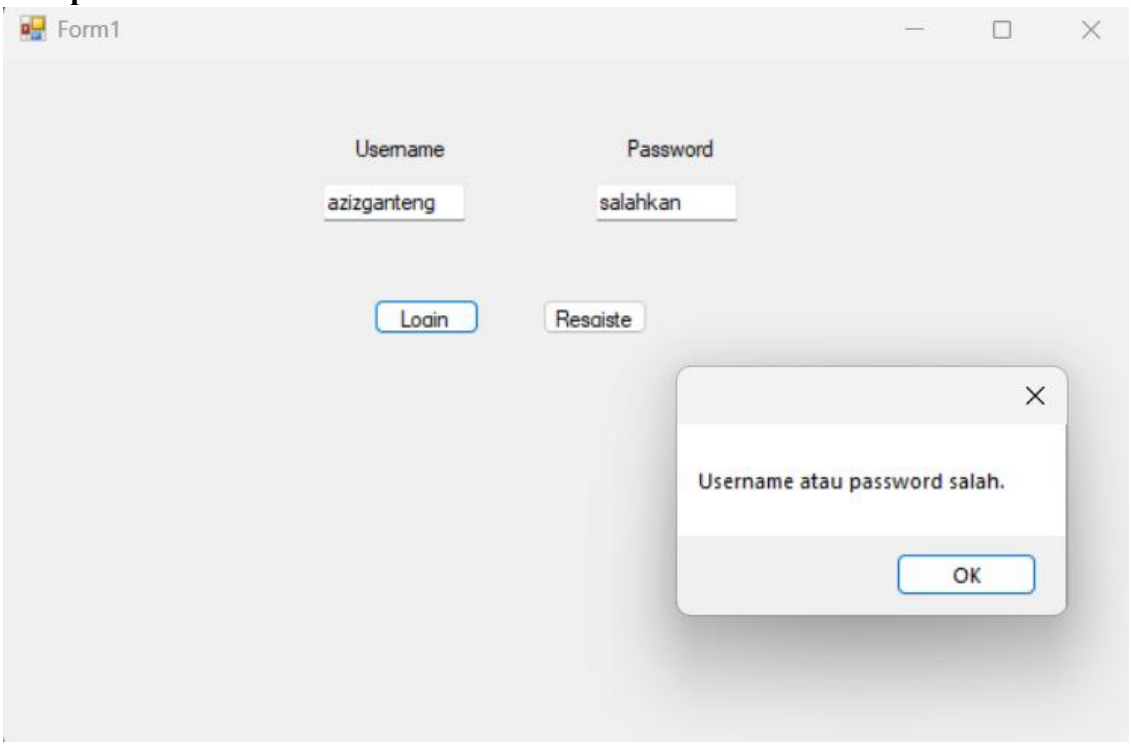
FormRegister



Form Login



Jika password salah



Penjelasan

Proyek ini merupakan aplikasi desktop sederhana yang dikembangkan menggunakan C# dan Windows Forms, dengan tujuan utama untuk mengimplementasikan Secure Coding Practices dalam fitur registrasi dan login pengguna. Aplikasi ini memungkinkan pengguna untuk melakukan registrasi dengan username dan password yang tervalidasi berdasarkan aturan keamanan, seperti panjang karakter 8–20, hanya karakter ASCII, wajib mengandung angka dan simbol khusus, serta password tidak boleh mengandung username. Password pengguna tidak disimpan secara langsung, melainkan di-*hash* menggunakan algoritma SHA-256 sebelum disimpan ke dalam file JSON. Proyek ini juga menangani input tidak valid secara eksplisit untuk mencegah kesalahan saat runtime. Dengan pendekatan ini, aplikasi tidak hanya berfungsi sesuai kebutuhan dasar autentikasi, tetapi juga mengikuti prinsip keamanan perangkat lunak yang baik.