

**JURNAL
KONSTRUKSI PERANGKAT LUNAK**

PERTEMUAN 15



**Disusun Oleh :
Muhammad Abdul Aziz
2211104026
SE0601**

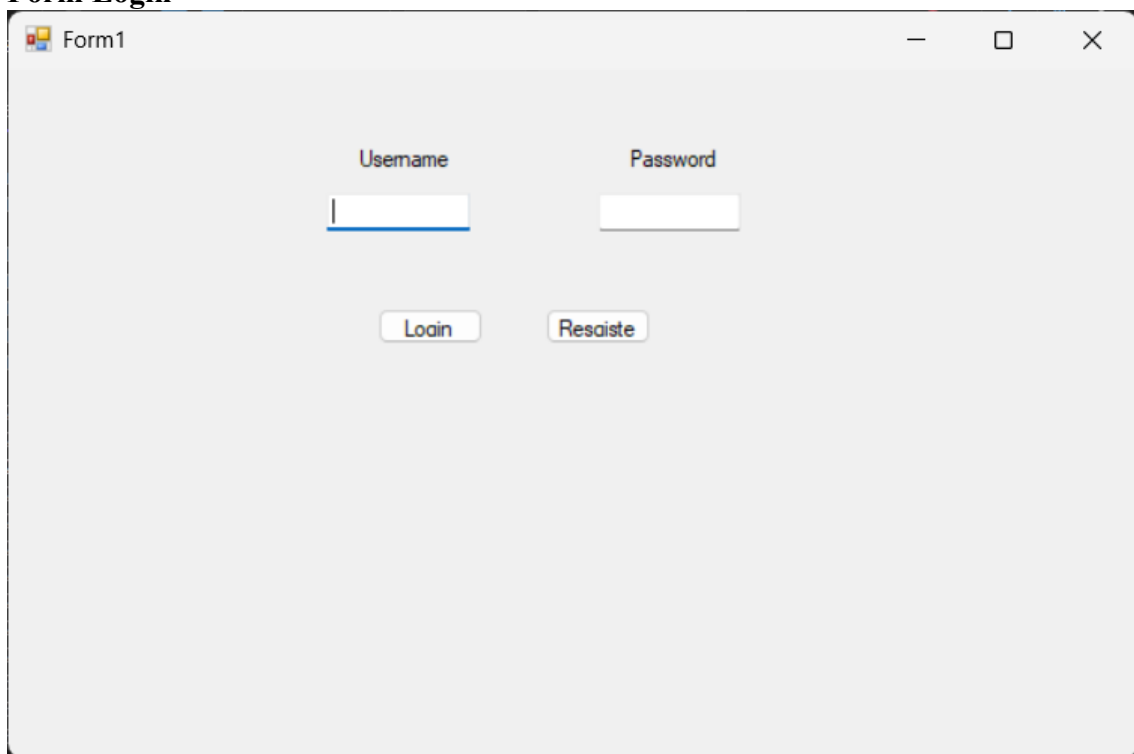
**Asisten Praktikum :
Naufal El Kamil Aditya Pratama Rahman
Imelda**

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

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

1. Hasil Run

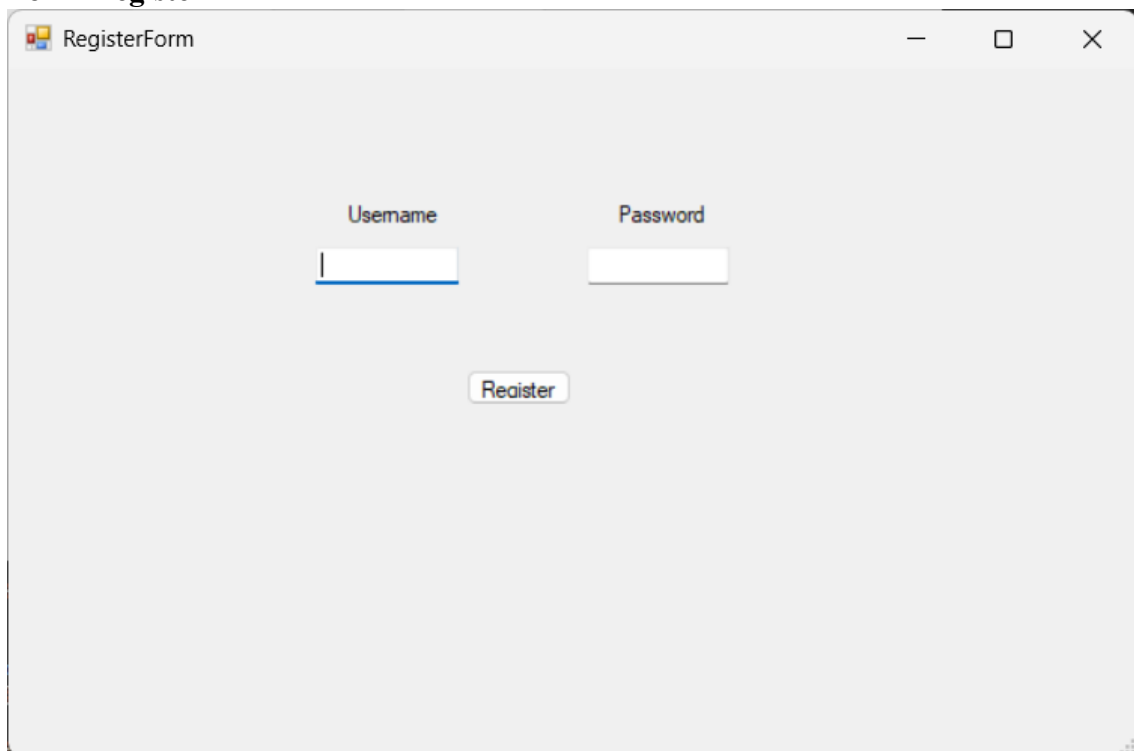
Form Login



The screenshot shows a Windows application window titled "Form1". Inside the window, there are two input fields: "Username" and "Password". Below these fields are two buttons: "Login" and "Resiste". The "Username" field has a blue underline, and the "Password" field is empty.

| Username | Password |
|----------|----------|
| | |

Form Register



The screenshot shows a Windows application window titled "RegisterForm". Inside the window, there are two input fields: "Username" and "Password". Below these fields is a single button: "Register". The "Username" field has a blue underline, and the "Password" field is empty.

| Username | Password |
|----------|----------|
| | |

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

Source Code

Form1.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace modul15_2211184026
12 {
13     3 references
14     public partial class Form1: Form
15     {
16         1 reference
17         public Form1()
18         {
19             InitializeComponent();
20
21         1 reference
22         private void btnLogin_Click(object sender, EventArgs e)
23         {
24             string username = txtUsername.Text.Trim();
25             string password = txtPassword.Text;
26             string hashed = Helper.ComputeSha256Hash(password);
27
28             var users = UserStorage.LoadUsers();
29
30             var user = users.FirstOrDefault(u => u.Username == username && u.PasswordHash == hashed);
31
32             if (user != null)
33                 MessageBox.Show("Login sukses!");
34             else
35                 MessageBox.Show("Username atau password salah.");
36         }
37
38         1 reference
39         private void btnRegister_Click(object sender, EventArgs e)
40         {
41             new RegisterForm().ShowDialog();
42         }
43
44         1 reference
45         private void txtUsername_TextChanged(object sender, EventArgs e)
46         {
47         }
48     }
49 }
```

Register.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace modul15_2211184026
12 {
13     3 references
14     public partial class RegisterForm: Form
15     {
16         1 reference
17         public RegisterForm()
18         {
19             InitializeComponent();
20         }
21
22         1 reference
23         private void btnSubmitReg_Click(object sender, EventArgs e)
24         {
25             string username = txtUsernameReg.Text.Trim();
26             string password = txtPasswordReg.Text;
27
28             if (!Helper.IsValidInput(username, password, out string error))
29             {
30                 MessageBox.Show(error);
31                 return;
32             }
33
34             var users = UserStorage.LoadUsers();
35
36             if (users.Any(u => u.Username == username))
37             {
38                 MessageBox.Show("Username sudah terdaftar.");
39                 return;
40             }
41
42             string hashed = Helper.ComputeSha256Hash(password);
43             users.Add(new User { Username = username, PasswordHash = hashed });
44             UserStorage.SaveUsers(users);
45
46             MessageBox.Show("Registrasi berhasil!");
47             this.Close();
48         }
49     }
50 }
```

User.cs

```
1  public class User
2  {
3      3 references
4      public string Username { get; set; }
5      2 references
6      public string PasswordHash { get; set; }
7  }
```

Helper.cs

```
1 using System.Linq;
2 using System.Security.Cryptography;
3 using System.Text;
4
5 3 references
6 public static class Helper
7 {
8     2 references
9     public static string ComputeSha256Hash(string rawData)
10    {
11        using (SHA256 sha256 = SHA256.Create())
12        {
13            byte[] bytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(rawData));
14            StringBuilder builder = new StringBuilder();
15            foreach (var b in bytes)
16                builder.Append(b.ToString("x2"));
17            return builder.ToString();
18        }
19    }
20
21    1 reference
22    public static bool IsValidInput(string username, string password, out string error)
23    {
24        error = "";
25
26        // Validasi panjang
27        if (username.Length < 8 || username.Length > 20)
28        {
29            error = "Username harus 8-20 karakter.";
30            return false;
31        }
32
33        if (password.Length < 8 || password.Length > 20)
34        {
35            error = "Password harus 8-20 karakter.";
36            return false;
37        }
38
39        // Validasi karakter ASCII
40        if (!username.All(c => c >= 32 && c <= 126))
41        {
42            error = "Username hanya boleh karakter ASCII.";
43            return false;
44        }
45
46        // Password mengandung angka
47        if (!password.Any(char.IsDigit))
48        {
49            error = "Password harus mengandung angka.";
50            return false;
51        }
52
53        // Password harus punya karakter unik
54        if (!password.Any(ch => "!@#%*&".Contains(ch)))
55        {
56            error = "Password harus mengandung minimal 1 karakter unik (!@#%*&).";
57            return false;
58        }
59
60        // Password tidak boleh mengandung username
61        if (password.ToLower().Contains(username.ToLower()))
62        {
63            error = "Password tidak boleh mengandung username.";
64            return false;
65        }
66
67        return true;
68    }
69 }
```

UserStorage.cs

```
1 using Newtonsoft.Json;
2 using System.Collections.Generic;
3 using System.IO;
4
5 3 references
6 public static class UserStorage
7 {
8     private static string path = "users.json";
9
10 2 references
11 public static List<User> LoadUsers()
12 {
13     if (!File.Exists(path)) return new List<User>();
14     string json = File.ReadAllText(path);
15     return JsonConvert.DeserializeObject<List<User>>(json) ?? new List<User>();
16 }
17
18 1 reference
19 public static void SaveUsers(List<User> users)
20 {
21     string json = JsonConvert.SerializeObject(users, Formatting.Indented);
22     File.WriteAllText(path, json);
23 }
```

Output

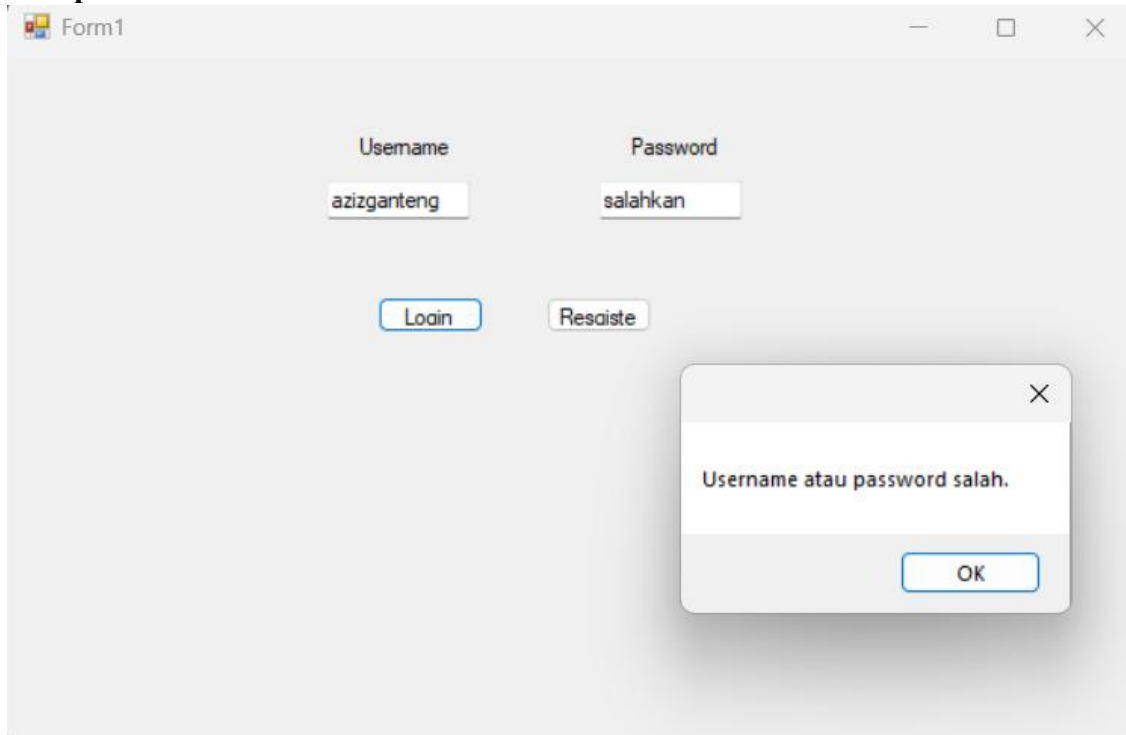
FormRegister

The screenshot shows a Windows application window titled "RegisterForm". It contains two text input fields: "Username" with the text "azizganteng" and "Password" with the text "DullzZgmng1!". Below these fields is a "Register" button. A small modal dialog box is open in the foreground, displaying the message "Registrasi berhasil!" (Registration successful!) and an "OK" button.

Form Login

The screenshot shows a Windows application window titled "Form1". It contains two text input fields: "Username" with the text "azizganteng" and "Password" with the text "DullzZgmng1!". Below these fields are two buttons: "Login" and "Resoiste" (likely a typo for "Register"). A small modal dialog box is open in the foreground, displaying the message "Login sukses!" (Login successful!) and an "OK" button.

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.