

**TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK
MODUL 15**



**Disusun Oleh :
KHOLIFAH DINA / 2211104004
SE-06-01**

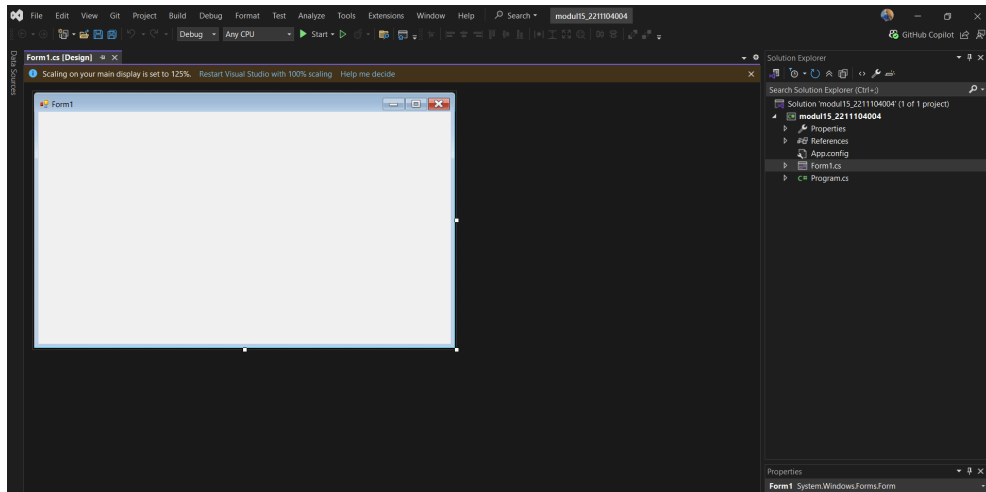
**Asisten Praktikum:
Naufal El Kamil Aditya Pratama Rahman
Imelda**

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

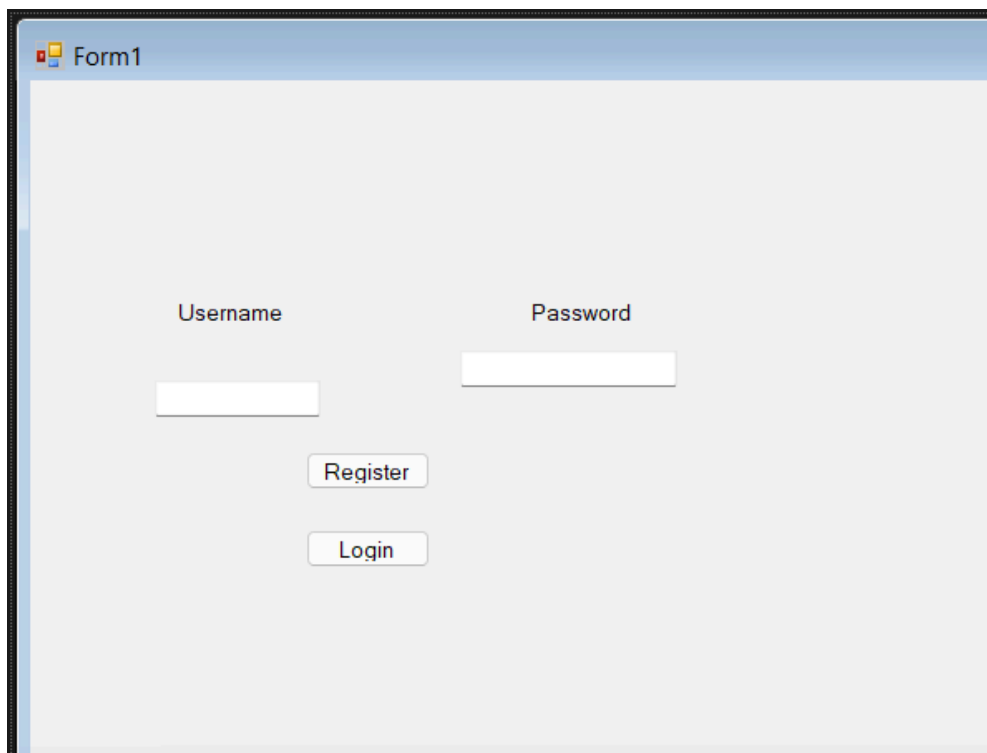
**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025
TUGAS JURNAL**

1. MEMBUAT PROJECT WEB API

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



Form :

A screenshot of the 'Form1' window. The window has a title bar with the text 'Form1'. The form contains two input fields labeled 'Username' and 'Password'. Below these fields are two buttons: 'Register' and 'Login'.

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

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

Source Code:

- **Program.cs**

```

15_2211104004 modul15_NIM.Program
1  using System;
2  using System.Windows.Forms;
3
4  namespace modul15_NIM
5  {
6      0 references
7      static class Program
8      {
9          [STAThread]
10         0 references
11         static void Main()
12         {
13             Application.EnableVisualStyles();
14             Application.SetCompatibleTextRenderingDefault(false);
15             Application.Run(new Form1());
16         }
17     }
18 }

```

- User.cs

```

modul15_2211104004 modul15_NIM.User
1  namespace modul15_NIM
2  {
3      0 references
4      public class User
5      {
6          0 references
7          public string Username { get; set; }
8          2 references
9          public string PasswordHash { get; set; }
10     }
11 }

```

- Helper.cs (SHA256 hashing)

```

1  using System.Security.Cryptography;
2  using System.Text;
3
4  namespace modul15_NIM
5  {
6      0 references
7      public static class Helper
8      {
9          2 references
10         public static string HashPassword(string password)
11         {
12             using (SHA256 sha256 = SHA256.Create())
13             {
14                 byte[] bytes = Encoding.UTF8.GetBytes(password);
15                 byte[] hashBytes = sha256.ComputeHash(bytes);
16                 StringBuilder builder = new StringBuilder();
17                 foreach (byte b in hashBytes)
18                     builder.Append(b.ToString("x2"));
19                 return builder.ToString();
20             }
21         }
22     }
23 }

```

- UserStorage.cs (Simpan file JSON)

```

modul15_2211104004 modul15_NIM.UserStorage
1  using System.Collections.Generic;
2  using System.IO;
3  using System.Xml;
4  using Newtonsoft.Json;
5
6  namespace modul15_NIM
7  {
8      3 references
9      public static class UserStorage
10     {
11         private static string filePath = "users.json";
12
13         3 references
14         public static List<User> LoadUsers()
15         {
16             if (!File.Exists(filePath))
17                 return new List<User>();
18
19             string json = File.ReadAllText(filePath);
20             return JsonConvert.DeserializeObject<List<User>>(json);
21
22         1 reference
23         public static void SaveUser(User user)
24         {
25             var users = LoadUsers();
26             users.Add(user);
27             string json = JsonConvert.SerializeObject(users, Newtonsoft.Json.Formatting.Indented);
28             File.WriteAllText(filePath, json);
29
30         1 reference
31         public static User FindUser(string username)
32         {
33             var users = LoadUsers();
34             return users.Find(u => u.Username == username);
35
36         1 reference
37         public static bool UsernameExists(string username)
38         {
39             var users = LoadUsers();
40             return users.Exists(u => u.Username == username);
41         }
42     }
43 }

```

- Form1.cs (Gabungan Login + Register)

```

1  using System;
2  using System.Text.RegularExpressions;
3  using System.Windows.Forms;
4
5  namespace modul15_NIM
6  {
7      public partial class Form1 : Form
8      {
9          1 reference
10         public Form1()
11         {
12             InitializeComponent();
13         }
14
15         // Tombol Register
16         1 reference
17         private void btnRegister_Click(object sender, EventArgs e)
18         {
19             string username = txtUsername.Text.Trim();
20             string password = txtPassword.Text;
21
22             // Validasi panjang
23             if (username.Length < 8 || username.Length > 20 || password.Length < 8 || password.Length > 20)
24             {
25                 lblOutput.Text = "Username/Password harus 8-20 karakter.";
26                 return;
27             }
28
29             // Validasi hanya alfabet + angka (ASCII)
30             if (!Regex.IsMatch(username, @"[a-zA-Z0-9]+$"))
31             {
32                 lblOutput.Text = "Username hanya boleh huruf dan angka (ASCII).";
33                 return;
34             }
35
36             // Validasi password rules
37             if (!Regex.IsMatch(password, @"^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9])(?=.*[!@#$%^&*]).+$"))
38             {
39                 lblOutput.Text = "Password harus ada huruf besar, angka, dan simbol (!@#$%^&*).";
40                 return;
41             }
42
43             if (password.ToLower().Contains(username.ToLower()))
44             {
45                 lblOutput.Text = "Password tidak boleh mengandung username.";
46                 return;
47             }
48
49             if (UserStorage.UsernameExists(username))
50             {
51                 lblOutput.Text = "Username sudah terdaftar.";
52                 return;
53             }
54
55             string hashed = Helper.HashPassword(password);
56             UserStorage.SaveUser(new User { Username = username, PasswordHash = hashed });
57
58             lblOutput.Text = "Registrasi berhasil!";
59         }
60
61         // Tombol Login
62         0 references
63         private void btnLogin_Click(object sender, EventArgs e)
64         {
65             string username = txtUsername.Text.Trim();
66             string password = txtPassword.Text;
67
68             var user = UserStorage.FindUser(username);
69             if (user == null)
70             {
71                 lblOutput.Text = "Username tidak ditemukan.";
72                 return;
73             }
74
75             string hashed = Helper.HashPassword(password);
76             if (user.PasswordHash == hashed)
77             {
78                 lblOutput.Text = "Login berhasil!";
79             }
80             else
81             {
82                 lblOutput.Text = "Password salah.";
83             }
84         }
85
86         1 reference
87         private void txtUsername_TextChanged(object sender, EventArgs e)
88         {
89         }
90     }

```

**Hasil Run
Register :**

Form1

Username Password

Kholifahdina12

Register

Login

Registrasi berhasil!

Login:

Form1

Username Password

DinaDina1234

Register

Login

Registrasi berhasil!

4. Penjelasan Program

Program ini merupakan aplikasi desktop berbasis Windows Forms (WinForms) yang dikembangkan menggunakan bahasa C# dengan Visual Studio, dirancang untuk memenuhi tugas modul 15 terkait secure coding practices. Aplikasi ini menggabungkan fitur registrasi dan login dalam satu form utama (Form1), sehingga memudahkan pengguna untuk mendaftarkan akun sekaligus melakukan proses login

tanpa perlu form terpisah. Pada antarmuka Form1, terdapat dua input utama berupa TextBox yaitu txtUsername untuk mengisi username, dan txtPassword untuk mengisi password. Password input dikonfigurasi menggunakan properti UseSystemPasswordChar = true agar karakter yang diketikkan tersamarkan (bintang). Selain itu, terdapat dua tombol utama yaitu btnRegister untuk proses registrasi, dan btnLogin untuk proses login. Label lblOutput digunakan sebagai tempat menampilkan pesan validasi maupun hasil dari proses login dan registrasi. Fungsi registrasi mengimplementasikan input validation secara ketat sesuai prinsip secure coding. Validasi dilakukan pada panjang username dan password, yang harus berada dalam rentang 8 hingga 20 karakter. Selain itu, username hanya boleh berisi huruf dan angka (alfabet ASCII), sedangkan password diwajibkan mengandung setidaknya satu huruf besar, satu angka, dan satu karakter spesial (!@#\$%^&*). Password juga tidak boleh mengandung bagian dari username. Apabila validasi gagal, aplikasi akan memberikan pesan error secara spesifik kepada pengguna. Jika validasi berhasil, password yang dimasukkan akan di-hash menggunakan algoritma SHA256 yang diimplementasikan pada file Helper.cs. Hasil hash ini kemudian disimpan bersama username ke dalam file users.json menggunakan mekanisme serialisasi JSON dari library Newtonsoft.Json. Semua proses penyimpanan data dikelola oleh class UserStorage.cs yang menangani proses baca tulis file JSON secara aman. Pada saat proses login, aplikasi akan memeriksa apakah username yang dimasukkan sudah terdaftar. Jika ditemukan, password yang dimasukkan akan di-hash kembali dan dibandingkan dengan hash yang tersimpan. Jika hasil hash cocok, login dinyatakan berhasil, sebaliknya akan ditampilkan pesan "Password salah" atau "Username tidak ditemukan" sesuai kondisi yang terjadi. Selain implementasi fitur utama, aplikasi ini juga memenuhi aspek software profiling dimana saat dijalankan dapat diamati performanya dari sisi penggunaan CPU dan memory, baik pada saat idle maupun saat pengguna melakukan input dan proses login atau register. Dengan seluruh mekanisme validasi input, manajemen password menggunakan hashing, penyimpanan data dengan file JSON, serta pengelolaan pesan error yang terarah, program ini telah menerapkan konsep secure coding sederhana namun efektif sesuai dengan ketentuan praktikum modul 15.