

**TUGAS JURNAL  
KONSTRUKSI PERANGKAT LUNAK  
MODUL 15**



**Disusun Oleh :**

**Arinka Anendya Chazanah / 2211104013**

**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**

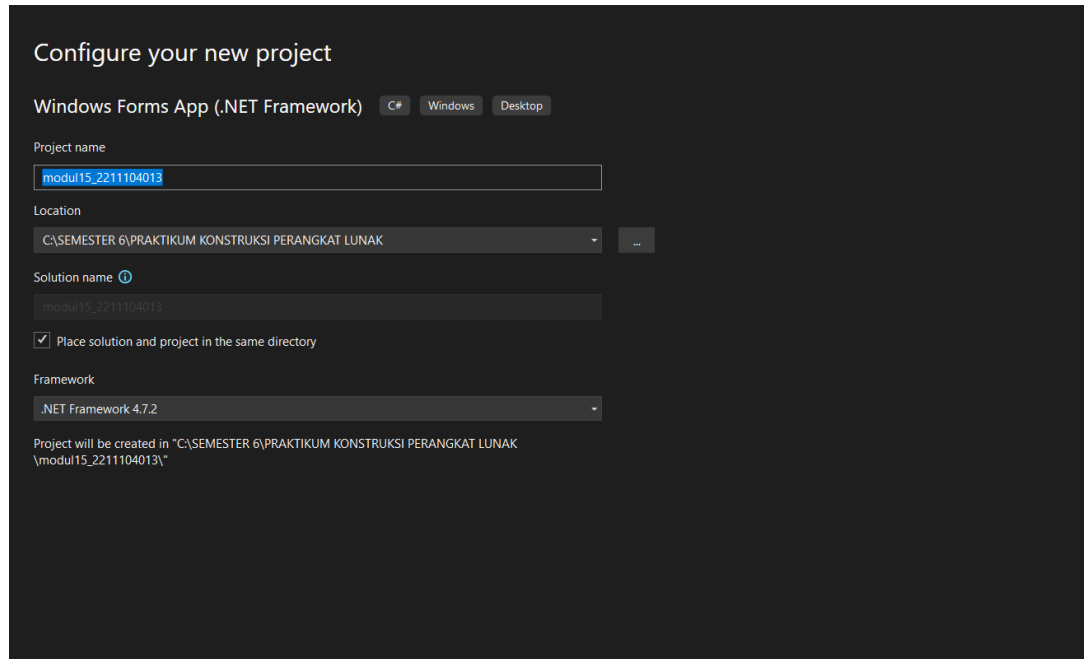
**TELKOM UNIVERSITY PURWOKERTO**

**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



## 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

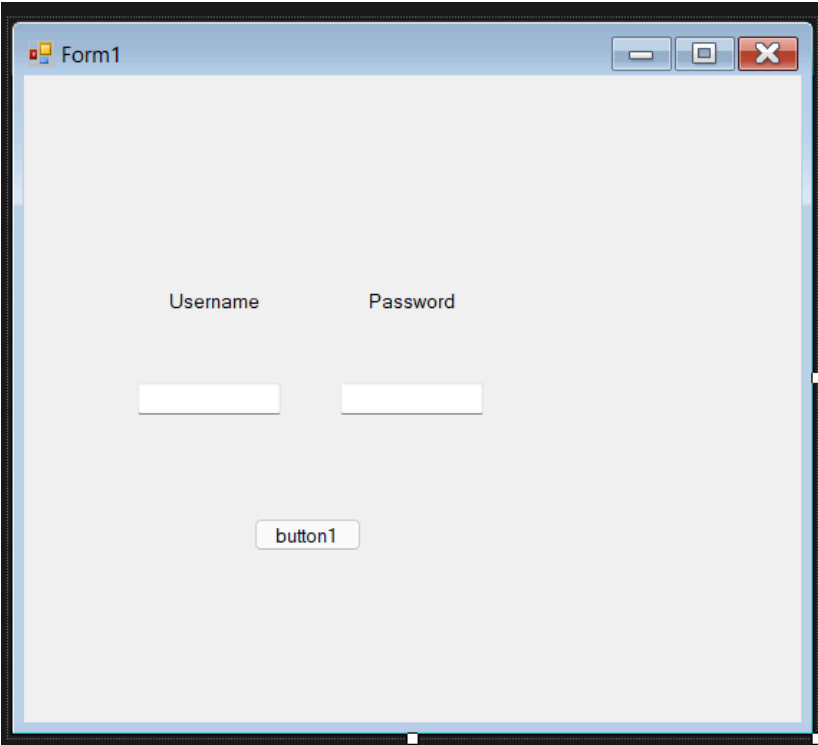
Ketika sistem harus menerima inputan password, harus ada aturan keamanan untuk password tersebut. Contoh:

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

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

**Source Code:**

**a. Membuat Form Registrasi**



The screenshot shows a standard Windows application window titled "Form1". Inside the window, there is a registration form with two text input fields. The first field is labeled "Username" and the second field is labeled "Password". Below these two fields, there is a single button labeled "button1". The form is centered within the window.

**FormRegister.cs:**

```

1  using System;
2  using System.IO;
3  using System.Security.Cryptography;
4  using System.Text;
5  using System.Text.RegularExpressions;
6  using System.Windows.Forms;
7  using System.Xml;
8  using Newtonsoft.Json;
9
10 namespace modul15_NIM
11 {
12     2 references
13     public partial class FormRegister : Form
14     {
15         0 references
16         public FormRegister()
17         {
18             InitializeComponent();
19         }
20
21         1 reference
22         private void btnRegister_Click(object sender, EventArgs e)
23         {
24             string username = txtUsername.Text.Trim();
25             string password = txtPassword.Text;
26
27             // Validasi panjang username & password
28             if (username.Length < 8 || username.Length > 20 || password.Length < 8 || password.Length > 20)
29             {
30                 lblOutput.Text = "Username/Password harus 8-20 karakter.";
31                 return;
32             }
33
34             // Validasi username hanya huruf dan angka
35             if (!Regex.IsMatch(username, @"^[a-zA-Z0-9]+$"))
36             {
37                 lblOutput.Text = "Username hanya boleh huruf dan angka.";
38                 return;
39             }
40
41             // Password harus mengandung angka dan karakter unik
42             if (!Regex.IsMatch(password, @"[0-9]") || !Regex.IsMatch(password, @"[!@#%*&*]"))
43             {
44                 lblOutput.Text = "Password harus mengandung angka dan karakter unik (!@#%*&*).";
45                 return;
46             }
47
48             // Password tidak boleh mengandung username
49             if (password.ToLower().Contains(username.ToLower()))
50             {
51                 lblOutput.Text = "Password tidak boleh mengandung username.";
52                 return;
53             }
54         }
55     }
56 }

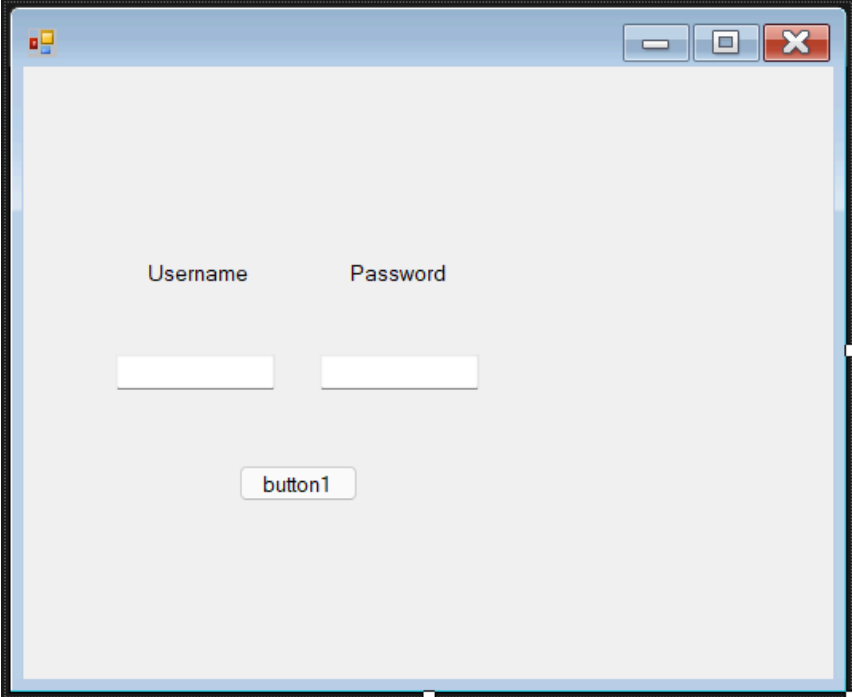
```

```

51
52     // Hash password
53     string hashedPassword = HashPassword(password);
54
55     // Simpan ke file JSON
56     User newUser = new User { Username = username, Password = hashedPassword };
57     SaveUser(newUser);
58
59     lblOutput.Text = "Registrasi berhasil!";
60 }
61
62 1 reference
63 private string HashPassword(string password)
64 {
65     using (SHA256 sha256 = SHA256.Create())
66     {
67         byte[] bytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(password));
68         StringBuilder sb = new StringBuilder();
69         foreach (byte b in bytes)
70         {
71             sb.Append(b.ToString("x2"));
72         }
73         return sb.ToString();
74     }
75 }
76
77 1 reference
78 private void SaveUser(User user)
79 {
80     string path = "users.json";
81     UserList list = new UserList();
82
83     if (File.Exists(path))
84     {
85         string json = File.ReadAllText(path);
86         list = JsonConvert.DeserializeObject<UserList>(json) ?? new UserList();
87     }
88
89     list.Users.Add(user);
90     File.WriteAllText(path, JsonConvert.SerializeObject(list, Formatting.Indented));
91 }
92 }

```

**b. Membuat Form Login**



A screenshot of a Windows-style application window titled "Form Login". The window has a standard title bar with minimize, maximize, and close buttons. The main content area is light gray and contains the following elements:

- Two labels, "Username" and "Password", positioned above their respective input fields.
- Two white rectangular text input fields, one for the username and one for the password.
- A single button labeled "button1" centered below the input fields.

**FormLogin.cs:**

```

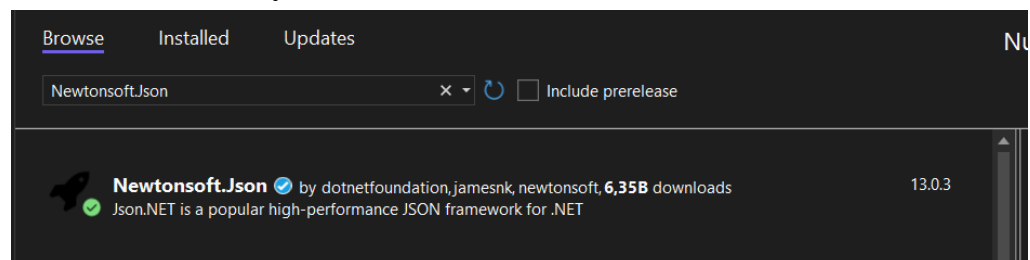
ul15_2211104013 modul15_NIM.FormLogin
1 using System;
2 using System.IO;
3 using System.Security.Cryptography;
4 using System.Text;
5 using System.Windows.Forms;
6 using Newtonsoft.Json;
7
8 namespace modul15_NIM
9 {
10     2 references
11     public partial class FormLogin : Form
12     {
13         0 references
14         public FormLogin()
15         {
16             InitializeComponent();
17         }
18
19         0 references
20         private void btnLogin_Click(object sender, EventArgs e)
21         {
22             string username = txtUsername.Text.Trim();
23             string password = txtPassword.Text;
24             string hashedPassword = HashPassword(password);
25
26             string path = "users.json";
27             if (!File.Exists(path))
28             {
29                 lblOutput.Text = "Data user belum ada!";
30                 return;
31             }
32
33             string json = File.ReadAllText(path);
34             Userlist list = JsonConvert.DeserializeObject<Userlist>(json);
35
36             foreach (User user in list.Users)
37             {
38                 if (user.Username == username && user.Password == hashedPassword)
39                 {
40                     lblOutput.Text = "Login Berhasil!";
41                     return;
42                 }
43             }
44
45             lblOutput.Text = "Username atau Password salah!";
46         }
47
48         1 reference
49         private string HashPassword(string password)
50         {
51             using (SHA256 sha256 = SHA256.Create())
52             {
53                 byte[] bytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(password));
54
55                 byte[] bytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(password));
56                 StringBuilder sb = new StringBuilder();
57                 foreach (byte b in bytes)
58                 {
59                     sb.Append(b.ToString("x2"));
60                 }
61                 return sb.ToString();
62             }
63         }
64
65         private void FormLogin_Load(object sender, EventArgs e)
66         {
67         }
68     }
69 }

```

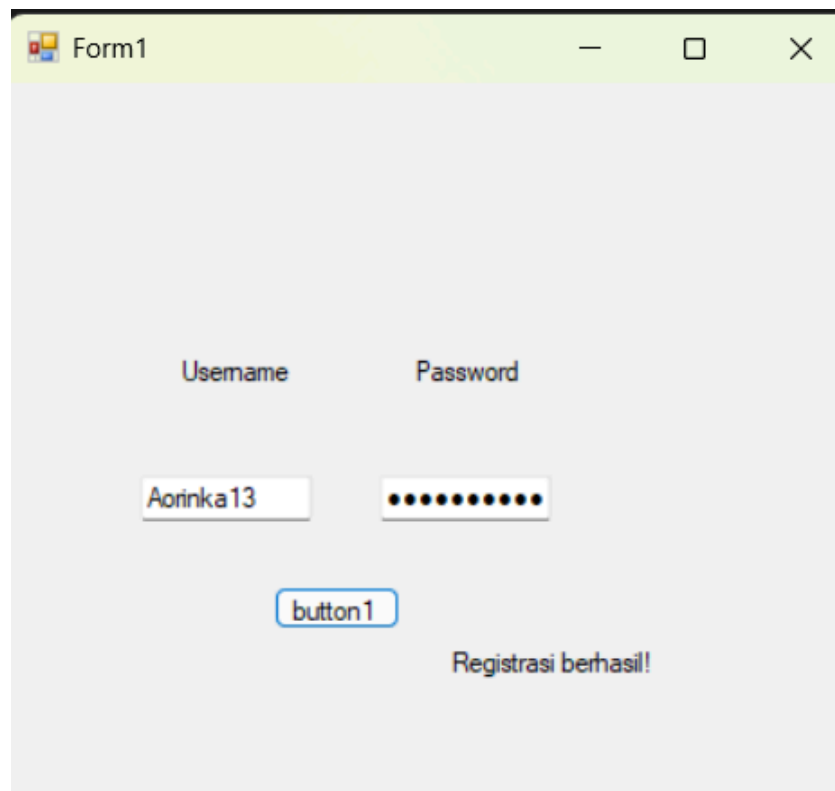
User.cs:

```
C# modul15_2211104013 modul15_NIM.User
1 using System.Collections.Generic;
2
3 namespace modul15_NIM
4 {
5     6 references
6     public class User
7     {
8         2 references
9         public string Username { get; set; }
10        2 references
11        public string Password { get; set; }
12    }
13
14    6 references
15    public class UserList
16    {
17        2 references
18        public List<User> Users { get; set; } = new List<User>();
19    }
20 }
```

### c. Tambahkan Library JSON



### d. Hasil Uji Coba Run:



#### 4. PENJELASAN PORGRAM

Program yang dibangun merupakan sebuah aplikasi desktop berbasis Windows Forms dengan bahasa pemrograman C#. Aplikasi ini memiliki fitur utama yaitu registrasi user dengan menerapkan prinsip-prinsip secure coding practices agar keamanan data user tetap terjaga. Dalam proses registrasi, user diminta untuk memasukkan username dan password melalui tampilan form yang telah dirancang dengan desain modern, simpel, dan user-friendly. Form registrasi dirancang secara dinamis menggunakan kode, sehingga tampilan memiliki warna latar belakang abu-abu terang dengan panel input berwarna putih yang membuat kontras visual lebih menarik. Label diberikan dengan ukuran font yang lebih besar dan tebal, sementara tombol registrasi dibuat dengan warna biru modern agar lebih menonjol. Sebelum data disimpan, program melakukan serangkaian validasi input. Validasi panjang data memastikan bahwa username dan password yang dimasukkan memiliki panjang minimal 8 karakter dan maksimal 20 karakter. Selain itu, username dibatasi hanya boleh berisi huruf dan angka menggunakan validasi regular expression. Validasi password juga dibuat lebih ketat, yaitu wajib mengandung setidaknya satu angka dan satu karakter spesial dari kumpulan simbol `!@#$%^&*`. Untuk mencegah kelemahan keamanan, program juga memastikan bahwa password tidak boleh mengandung bagian dari username yang dimasukkan. Apabila ada kesalahan pada input, sistem akan menampilkan pesan kesalahan melalui kotak dialog (MessageBox) yang jelas dan mudah dipahami oleh user. Setelah seluruh validasi berhasil dilewati, password yang dimasukkan oleh user akan diolah menggunakan algoritma hash SHA256. Dengan metode hashing ini, password tidak akan pernah disimpan dalam bentuk teks asli (plain-text), sehingga meningkatkan keamanan penyimpanan data. Data user yang berisi username dan password yang telah di-hash kemudian disimpan ke dalam file JSON (users.json) dengan bantuan library Newtonsoft.Json. File JSON ini menyimpan data dalam format array, memungkinkan banyak user untuk diregistrasikan dan disimpan dengan rapi. Secara keseluruhan, program ini tidak hanya berfungsi untuk registrasi user, tetapi juga sudah menerapkan beberapa prinsip penting dari secure coding, seperti input validation, invalid handling, password hashing, dan password policy enforcement. Selain itu, desain antar muka aplikasi dibuat sederhana namun elegan sehingga memudahkan user dalam melakukan registrasi dengan nyaman dan aman. Program ini dapat dengan mudah dikembangkan lebih lanjut untuk menambahkan fitur seperti login user, manajemen data user, maupun pengembangan keamanan tambahan.