

**LAPORAN JURNAL
MODUL 15**



Disusun Oleh :

Zaenarif Putra 'Ainurdin – 2311104049

Kelas :

SE-07-02

Dosen :

Yudha Islami Sulistya

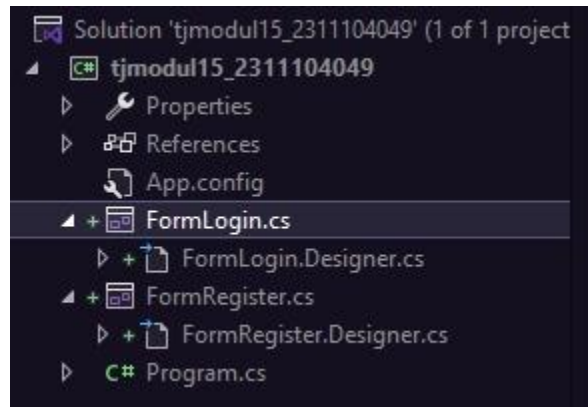
**PROGRAM STUDI SOFTWARE ENGINEERING
DIREKTORAT KAMPUS PURWOKERTO
TELKOM UNIVERSITY
PURWOKERTO
2025**

I. Link Github

https://github.com/zaenarifputra/KPL_Zaenarif-Putra-Ainurdin_2311104049_S1_SE-07-02/tree/6bbc6b49cc3c7cd1f4e58c0c7ce91cf9f04920ae/15_Review_Tugas_Besar/TJ

II. Alur Pengerjaan

1. Pada modul 15 berikut saya refactor dari modul 3, yang dimana di rename menjadi “tjmodul15_231110409” yang kemudian juga untuk file formnya terdapat 2 yaitu FormLogin, dan FormRegister sehingga hasilnya seperti berikut :



2. Setelah rename semuanya kita menambahkan folder Helpers dan juga filenya yang diberi nama “SecurityHelper.cs” berikut syntax dan juga penjelasan sederhananya :

```
1 using System.Security.Cryptography;
2 using System.Text;
3
4 namespace tjmodul15_231110409.Helpers
5 {
6     2 references
7     public static class SecurityHelper
8     {
9         2 references
10        public static string HashPassword(string password)
11        {
12            using (SHA256 sha = SHA256.Create())
13            {
14                byte[] bytes = sha.ComputeHash(Encoding.UTF8.GetBytes(password));
15                StringBuilder builder = new StringBuilder();
16                foreach (byte b in bytes)
17                    builder.Append(b.ToString("x2"));
18                return builder.ToString();
19            }
20        }
21    }
```

Class SecurityHelper adalah class statis dalam namespace tjmodul15_231110409.Helpers yang menyediakan method HashPassword, berfungsi untuk mengubah input password menjadi string hash menggunakan algoritma SHA256 dari library System.Security.Cryptography; di dalamnya, password yang diketik pengguna akan dikonversi menjadi array byte dengan Encoding.UTF8, kemudian diproses oleh sha.ComputeHash() untuk menghasilkan hash byte array, lalu setiap byte-nya diubah ke format string heksadesimal dua digit menggunakan ToString("x2") dan dirangkai dengan StringBuilder menjadi string akhir sepanjang 64 karakter, yang kemudian dikembalikan sebagai representasi hash password yang aman dan tidak dapat dibalik, sehingga cocok

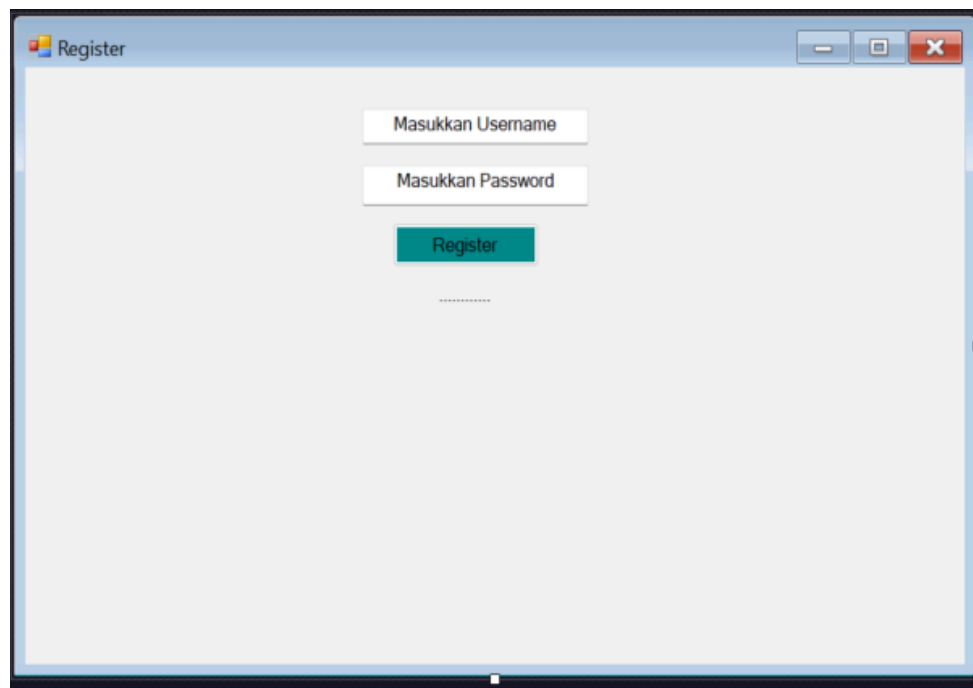
untuk menyimpan password pengguna dalam format terenkripsi tanpa menyimpan data asli.

3. Berikutnya membuat folder yang diberi nama Models dan untuk filenya juga diberi nama User.cs berikut syntax dan juga penjelasan sederhanaanya :

```
C# tjmodul15_2311104049 tjmodul15_2311104049.M
1 namespace tjmodul15_2311104049.Models
2 {
3     6 references
4     public class User
5     {
6         3 references
7         public string Username { get; set; }
8         2 references
9         public string Password { get; set; }
10    }
```

Class User dalam namespace tjmodul15_2311104049.Models adalah class model sederhana yang merepresentasikan data user, terdiri dari dua properti publik yaitu Username dan Password, yang masing-masing bertipe string dan menggunakan properti auto-implemented (get; set;) sehingga data dapat dibaca dan ditulis secara langsung, di mana class ini biasa digunakan sebagai struktur data saat melakukan deserialisasi atau serialisasi JSON (misalnya ketika menyimpan dan membaca file users.json), dan berfungsi sebagai blueprint atau cetakan objek user yang berisi informasi login pengguna dalam aplikasi.

4. Berikutnya menuju pada form yang 1 terdapat formRegister berikut gui tampilan dan juga penjelasan syntaxnya :



```
using System;
using System.Collections.Generic;
using System.IO;
using System.Text.RegularExpressions;
using System.Windows.Forms;
using Newtonsoft.Json;
using tjmodul15_2311104049.Helpers;
using tjmodul15_2311104049.Models;

namespace tjmodul15_2311104049
{
    public partial class FormRegister : Form
    {
        private readonly string userFile =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "users.json");

        public FormRegister()
        {
            InitializeComponent();
        }

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

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

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

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

            if (!IsValidUsername(username))
            {
                MessageBox.Show("Username hanya boleh huruf (3-20 karakter).", "Validasi Gagal", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }

            if (!IsValidPassword(password, username))
            {
                MessageBox.Show("Password harus 8-20 karakter, mengandung angka &
```

```
simbol, dan tidak mengandung username.", "Validasi Gagal", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
    return;
}

string hashedPassword = SecurityHelper.HashPassword(password);

List<User> users = new List<User>();

try
{
    if (File.Exists(userFile))
    {
        string json = File.ReadAllText(userFile);
        if (json.Trim().StartsWith("["))
        {
            users = JsonConvert.DeserializeObject<List<User>>(json);
        }
        else
        {
            MessageBox.Show("Format users.json salah. Harus berupa array JSON.",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show("Gagal membaca users.json: " + ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}

if (users.Exists(u => u.Username.Equals(username,
StringComparison.OrdinalIgnoreCase)))
{
    MessageBox.Show("Username sudah terdaftar.", "Registrasi Gagal",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}

users.Add(new User { Username = username, Password = hashedPassword });
File.WriteAllText(userFile, JsonConvert.SerializeObject(users,
Formatting.Indented));

MessageBox.Show("Registrasi berhasil!", "Sukses", MessageBoxButtons.OK,
MessageBoxIcon.Information);
this.Close(); // Menutup form setelah registrasi sukses (opsional)
}

private bool IsValidUsername(string username)
```

```

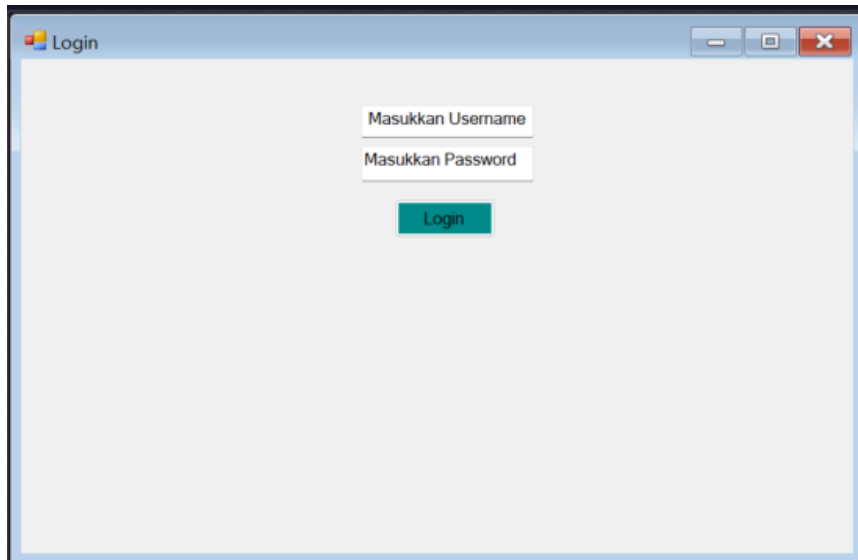
    {
        return Regex.IsMatch(username, @"^[a-zA-Z]{3,20}$");
    }

    private bool IsValidPassword(string password, string username)
    {
        return password.Length >= 8 &&
            password.Length <= 20 &&
            Regex.IsMatch(password, @"[0-9]") &&
            Regex.IsMatch(password, @"[!@#%$^&*]") &&
            !password.ToLower().Contains(username.ToLower());
    }
}

```

Class FormRegister dalam namespace tjmodul15_2311104049 adalah form Windows Forms yang berfungsi sebagai tampilan dan logika registrasi user, di mana file users.json dijadikan media penyimpanan data dan didefinisikan melalui properti userFile dengan path dinamis berdasarkan lokasi aplikasi, method btnRegister_Click menangani proses ketika tombol daftar diklik, mulai dari mengambil input username dan password dari TextBox, melakukan validasi menggunakan dua method lokal IsValidUsername dan IsValidPassword (dengan syarat panjang, karakter simbol, angka, dan tidak mengandung username), kemudian melakukan hashing password melalui SecurityHelper.HashPassword, membaca data JSON jika sudah ada dengan JsonConvert.DeserializeObject, mengecek apakah username sudah digunakan, lalu menambahkan data user baru ke list dan menyimpannya kembali ke file menggunakan JsonConvert.SerializeObject, serta menampilkan notifikasi berhasil menggunakan MessageBox, sedangkan event seperti FormRegister_Load, txtUsername_TextChanged, dan txtPassword_TextChanged tersedia untuk menangani interaksi GUI meskipun belum diisi logika tambahan.

5. Berikutnya di form 2 diberi formLogin berikut gui tampilan dan juga penjelasan syntaxnya:



```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Windows.Forms;
using Newtonsoft.Json;
using tjmodul15_2311104049.Helpers;
using tjmodul15_2311104049.Models;

namespace tjmodul15_2311104049
{
    public partial class FormLogin : Form
    {
        private string userFile = "users.json";

        public FormLogin()
        {
            InitializeComponent();
        }

        private void txtUsername_TextChanged(object sender, EventArgs e)
        {
            // Optional: validasi langsung saat mengetik
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
            string username = txtUsername.Text.Trim();
            string inputPassword = txtPassword.Text;

            if (string.IsNullOrEmpty(username) || string.IsNullOrEmpty(inputPassword))
            {
                MessageBox.Show("Username dan Password wajib diisi.", "Peringatan",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }

            string hashedPassword = SecurityHelper.HashPassword(inputPassword);

            if (!File.Exists(userFile))
            {
                MessageBox.Show("Belum ada user terdaftar.", "Informasi",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
                return;
            }

            List<User> users;
```

```
try
{
    string json = File.ReadAllText(userFile);
    users = JsonConvert.DeserializeObject<List<User>>(json);
}
catch (Exception ex)
{
    MessageBox.Show("Gagal membaca file user: " + ex.Message, "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}

var user = users.FirstOrDefault(u =>
    u.Username.Equals(username, StringComparison.OrdinalIgnoreCase) &&
    u.Password == hashedPassword);

if (user != null)
{
    MessageBox.Show("Login berhasil!", "Sukses", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
    // Bisa lanjut ke form utama jika ada
    // this.Hide(); new MainForm().Show();
}
else
{
    MessageBox.Show("Login gagal! Username atau password salah.", "Gagal",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

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

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

Class FormLogin dalam namespace tjmodul15_2311104049 adalah form Windows Forms yang menangani proses login pengguna, dengan menggunakan file users.json sebagai penyimpanan data user yang dibaca melalui File.ReadAllText dan didekode menggunakan JsonConvert.DeserializeObject menjadi list User, kemudian saat tombol login (btnLogin_Click) ditekan, input username dan password dari textbox akan diambil dan dicek apakah kosong, lalu password diubah menjadi hash menggunakan SecurityHelper.HashPassword agar bisa dibandingkan dengan data yang sudah tersimpan, dilanjutkan pencocokan data user melalui FirstOrDefault untuk mencari kecocokan username dan password hash secara case-insensitive, jika cocok maka ditampilkan pesan

login berhasil, dan jika tidak maka muncul notifikasi gagal login, sementara method tambahan seperti txtUsername_TextChanged, txtPassword_TextChanged, dan FormLogin_Load disiapkan untuk menangani interaksi GUI atau logika tambahan yang bisa dikembangkan lebih lanjut.

6. Berikutnya pembuatan file json yang diberi nama “users.json” berikut syntax dan juga penjelasan sederhananya :

```
[
  {
    "Username": "admin",
    "Password": "e99a18c428cb38f708663b87029e4abed8134aa"
  }
]
```

pada syntax berikut diberikan semua username dan password yang dimana password sudah dijadikan sebuah hash agar tidak bisa dilihat dan tidak bisa diketahui agar lebih aman.

7. Berikutnya setelah semua konfigurasi yang dibutuhkan sudah dibuat berikut akhir untuk menjalankan programnya melalui program.cs berikut syntax dan juga penjelasan sederhananya :

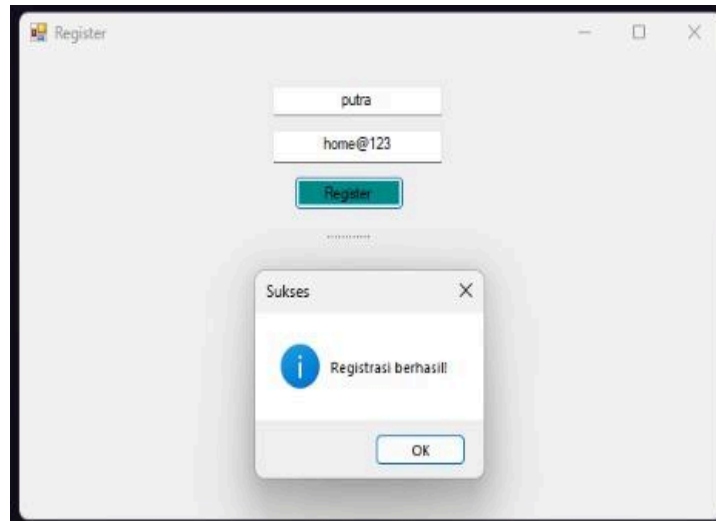
```
C# tjmodul15_2311104049 tjmodul15_2311104049.Prog
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using System.Windows.Forms;
6
7 namespace tjmodul15_2311104049
8 {
9     0 references
10     static class Program
11     {
12         /// <summary>
13         /// The main entry point for the application.
14         /// </summary>
15         [STAThread]
16         0 references
17         static void Main()
18         {
19             Application.EnableVisualStyles();
20             Application.SetCompatibleTextRenderingDefault(false);
21             Application.Run(new FormLogin());
22         }
23     }
```

Class Program dalam namespace tjmodul15_2311104049 adalah titik awal (entry point) dari aplikasi Windows Forms, ditandai dengan method Main yang diberi atribut [STAThread] untuk memastikan aplikasi berjalan dalam mode single-threaded apartment yang dibutuhkan oleh elemen GUI seperti clipboard dan dialog box, kemudian Application.EnableVisualStyles() digunakan untuk menerapkan tema visual Windows yang modern pada kontrol aplikasi, Application.SetCompatibleTextRenderingDefault(false) mengatur agar rendering teks menggunakan sistem GDI+ modern dibanding GDI lama untuk kompatibilitas visual yang lebih baik, dan akhirnya Application.Run(new FormLogin()) akan memulai dan menampilkan form pertama yaitu FormLogin, yang menjadi tampilan utama saat aplikasi dijalankan. Kemudian jika ingin mengganti tampilan menjadi FormRegister maka menjadi seperti berikut FormRegister())

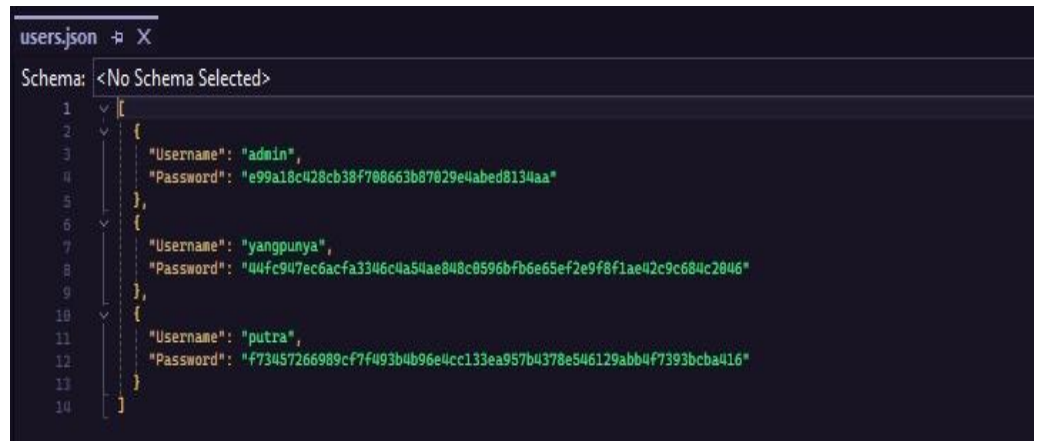
III. Hasil Running

1. Form Register

a. Hasil Running

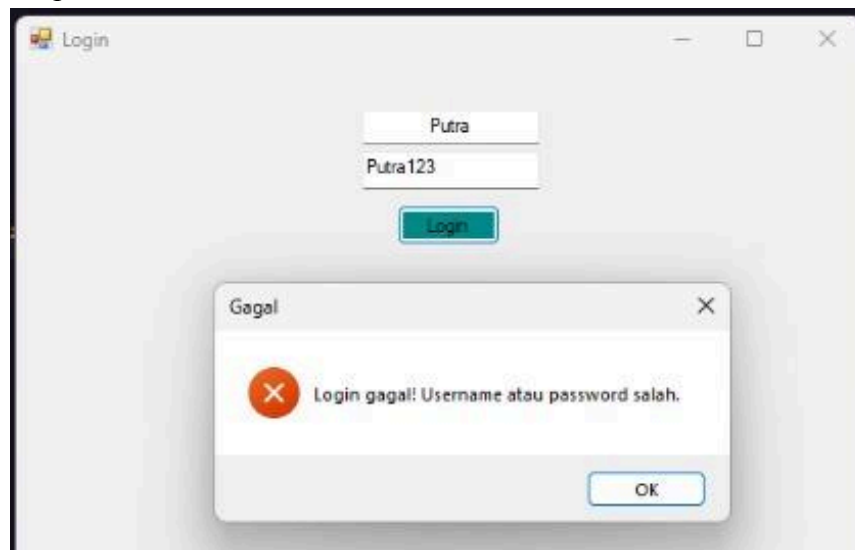


b. Ouput Update Register pada users.json

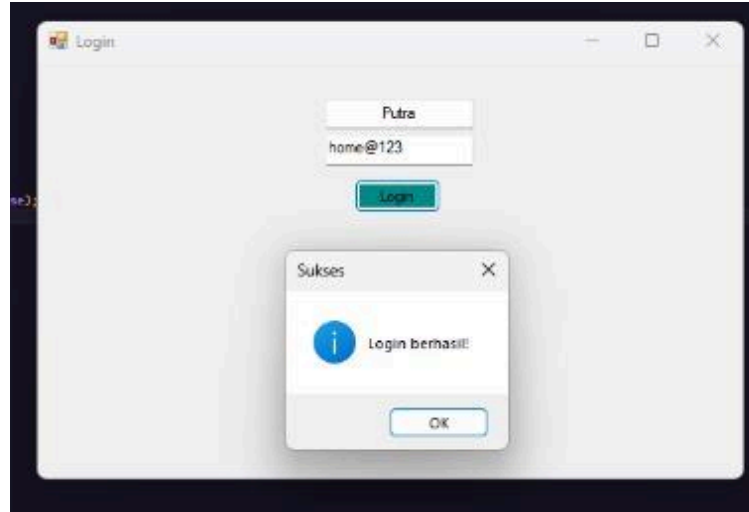


2. Form Login

a. Output Salah Password atau Username



b. Output Berhasil Login



Kesimpulan:

Program aplikasi ini merupakan sistem login dan registrasi berbasis Windows Forms dengan menerapkan prinsip **secure coding**, di mana data pengguna disimpan dalam file JSON dan password di-**hash** menggunakan algoritma SHA256 agar tidak tersimpan dalam bentuk asli, serta dilakukan validasi ketat terhadap input username dan password. Form *'FormRegister'* menangani pendaftaran user baru dengan pengecekan format dan duplikasi, sementara *'FormLogin'* memverifikasi kredensial pengguna dengan mencocokkan hash password. Class *'User'* berfungsi sebagai model data, *'SecurityHelper'* menangani proses enkripsi password, dan class *'Program'* menjadi titik masuk utama aplikasi yang menjalankan *'FormLogin'* sebagai tampilan pertama. Dengan struktur ini, aplikasi sederhana namun aman berhasil dibangun menggunakan pendekatan modular, terorganisir, dan sesuai praktik pengembangan perangkat lunak yang baik.