

Laporan Jurnal 15

1. AuthService.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Text.Json;

public class AuthService
{
    private const string FilePath = "users.json";
    private List<User> _users;

    public AuthService()
    {
        _users = LoadUsers();
    }

    private List<User> LoadUsers()
    {
        if (!File.Exists(FilePath))
        {
            return new List<User>();
        }
        string jsonString = File.ReadAllText(FilePath);
        return JsonSerializer.Deserialize<List<User>>(jsonString);
    }

    private void SaveUsers()
    {
        string jsonString = JsonSerializer.Serialize(_users, new
        JsonSerializerOptions { WriteIndented = true });
        File.WriteAllText(FilePath, jsonString);
    }

    // B. Password Management (i): Password hashing menggunakan SHA256
    private string HashPassword(string password)
    {
        using (SHA256 sha256Hash = SHA256.Create())
        {
            byte[] bytes =
            sha256Hash.ComputeHash(Encoding.UTF8.GetBytes(password));
            StringBuilder builder = new StringBuilder();
            for (int i = 0; i < bytes.Length; i++)
            {
                builder.Append(bytes[i].ToString("x2"));
            }
            return builder.ToString();
        }
    }

    public string Register(string username, string password)
```

```

{
    // A. Input Validation (ii): Validasi panjang data
    if (username.Length < 5 || username.Length > 20)
    {
        return "Username harus memiliki 5-20 karakter.";
    }

    // A. Input Validation (ii): Validasi panjang data
    if (password.Length < 8)
    {
        return "Password minimal harus 8 karakter.";
    }

    // A. Input Validation (i & iii): Validasi range data dan
    Handling data invalid
    // Cek apakah username hanya berisi huruf dan angka
    if (!username.All(char.IsLetterOrDigit))
    {
        return "Username hanya boleh berisi huruf dan angka.";
    }

    // B. Password Management (ii): Password rules
    if (password.ToLower().Contains(username.ToLower()))
    {
        return "Password tidak boleh mengandung username.";
    }

    // Cek apakah username sudah ada
    if (_users.Any(u => u.Username.Equals(username,
StringComparison.OrdinalIgnoreCase)))
    {
        return "Username sudah digunakan.";
    }

    // Memanggil metode hashing password
    string hashedPassword = HashPassword(password);

    _users.Add(new User { Username = username, HashedPassword =
hashedPassword });
    SaveUsers();

    return "Registrasi berhasil!";
}

public bool Login(string username, string password)
{
    string hashedPassword = HashPassword(password);
    User user = _users.FirstOrDefault(u =>
u.Username.Equals(username, StringComparison.OrdinalIgnoreCase) &&
u.HashedPassword == hashedPassword);
    return user != null;
}
}

```

Penjelasan:

Kelas AuthService berfungsi sebagai pusat logika untuk semua proses otentikasi pengguna, yaitu registrasi dan login. Kelas ini dirancang dengan fokus pada keamanan.

Poin-Poin Utama:

1. Manajemen Data: Secara otomatis memuat data pengguna dari file users.json saat aplikasi dimulai dan menyimpan kembali setiap ada perubahan, sehingga data bersifat permanen.
2. Keamanan Password: Password pengguna tidak pernah disimpan secara langsung. Sebaliknya, password diubah menjadi *hash* yang aman menggunakan algoritma SHA256. Saat login, password yang dimasukkan pengguna akan di-hash kembali dan dibandingkan dengan hash yang tersimpan. Ini adalah praktik keamanan standar untuk melindungi password.
3. Validasi Input: Sebelum mendaftarkan pengguna baru, kelas ini melakukan beberapa validasi penting:
 - o Memeriksa panjang minimal/maksimal username dan password.
 - o Memastikan username hanya berisi huruf dan angka.
 - o Mencegah penggunaan password yang mengandung username.
 - o Mencegah pendaftaran username yang sudah ada.
4. Fungsi Inti: Menyediakan dua metode utama:
 - o Register: Untuk membuat akun pengguna baru setelah semua validasi terpenuhi.
 - o Login: Untuk memverifikasi apakah kombinasi username dan password yang dimasukkan benar.

2. From1.cs

```
using System;
using System.Windows.Forms;

namespace modul15_2311104064
{
    public partial class Form1 : Form
    {
        private AuthService _authService;

        public Form1()
        {
            InitializeComponent();
            _authService = new AuthService();
        }

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

            string message = _authService.Register(username, password);

            lblStatus.Text = message;
        }

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

```

        bool isLoggedIn = _authService.Login(username, password);
        if (isLoggedIn)
        {
            lblStatus.Text = "Login berhasil! Selamat datang, " +
username + ".";
            MessageBox.Show("Login berhasil!", "Sukses",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            lblStatus.Text = "Login gagal. Username atau password
salah.";
            MessageBox.Show("Login gagal!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

Penjelasan:

Kesimpulan Kode Form1.cs

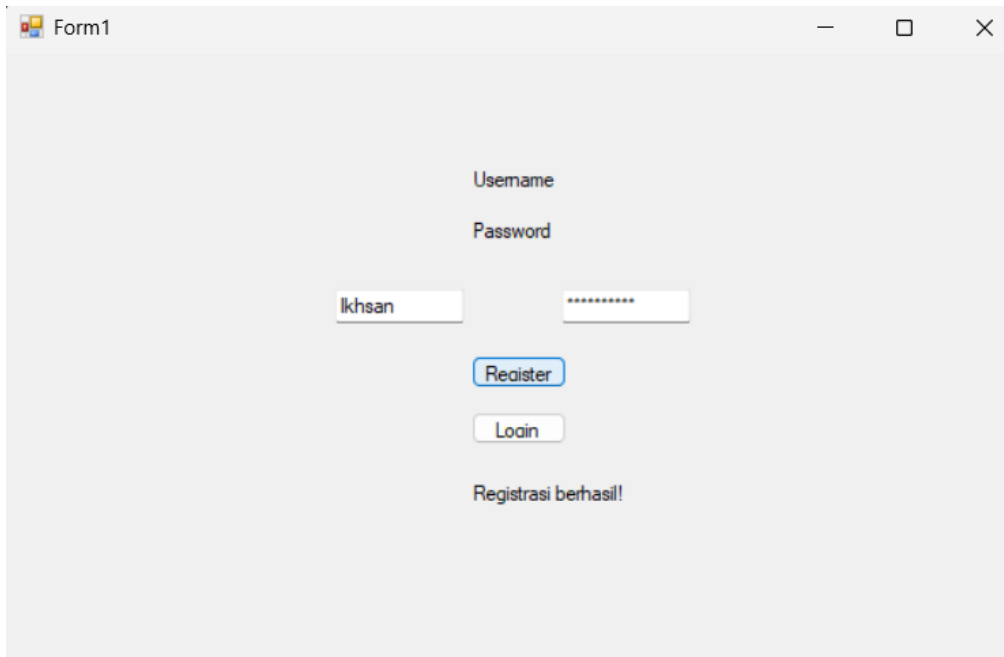
Kode ini berfungsi sebagai lapisan antarmuka pengguna (UI) atau "front-end" dari aplikasi registrasi dan login Anda. Tugas utamanya adalah menghubungkan apa yang dilihat dan dilakukan oleh pengguna di form dengan logika pemrosesan yang ada di kelas AuthService.

Poin-Poin Utama:

1. Mengelola Interaksi Pengguna: Kode ini menangani aksi pengguna, yaitu ketika tombol "Register" (btnRegister) atau "Login" (btnLogin) diklik.
2. Mendelegasikan Tugas: Form1 tidak melakukan validasi atau proses otentikasi sendiri. Sebaliknya, ia mendelegasikan tugas tersebut ke kelas AuthService.
 - Saat btnRegister_Click dijalankan, ia mengambil input username dan password dari txtUsername dan txtPassword, lalu menyerahkannya ke _authService.Register().
 - Saat btnLogin_Click dijalankan, ia melakukan hal yang sama dan menyerahkannya ke _authService.Login().
3. Menampilkan Umpan Balik (Feedback): Setelah mendapatkan hasil dari AuthService, Form1 bertanggung jawab untuk memberikan umpan balik kepada pengguna.
 - Hasil dari registrasi (apakah berhasil atau ada pesan error validasi) ditampilkan di label status (lblStatus).
 - Hasil dari login (berhasil atau gagal) ditampilkan di lblStatus dan juga dalam bentuk kotak pesan pop-up (MessageBox) untuk memberikan notifikasi yang lebih jelas.
4. Pemisahan Tanggung Jawab (Separation of Concerns): Kode ini menunjukkan praktik yang baik di mana logika tampilan (apa yang terlihat

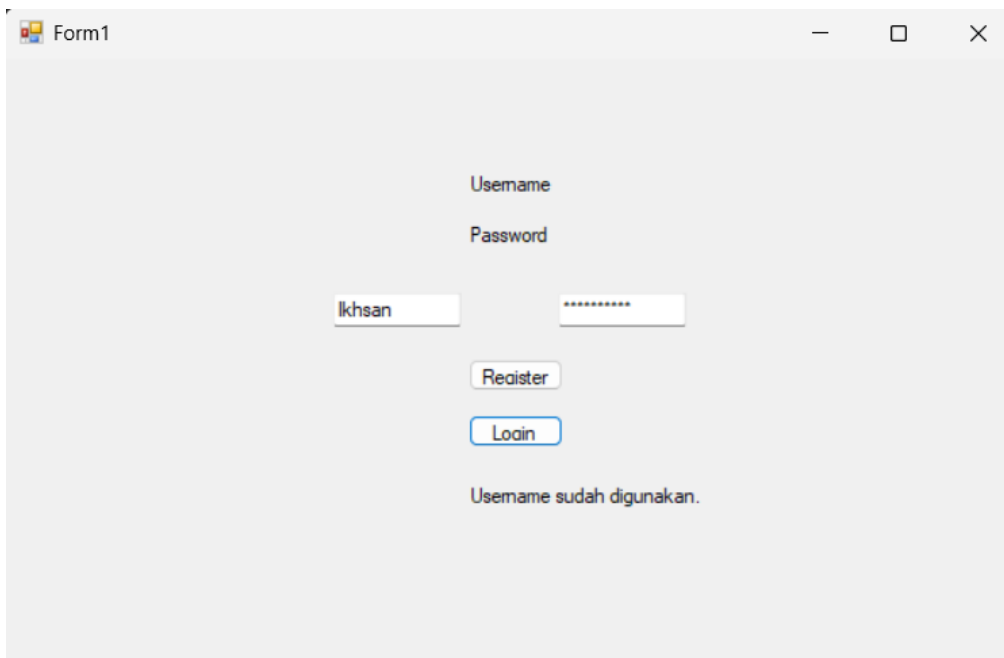
di form) dipisahkan dari logika bisnis (bagaimana cara mendaftar dan login). Form1 hanya peduli tentang tampilan dan interaksi, sementara AuthService peduli tentang aturan, keamanan, dan pengelolaan data.

3. Hasil Run



A screenshot of a Windows application window titled "Form1". The window has a light gray background and standard Windows window controls (minimize, maximize, close) in the top right corner. The form contains the following elements:

- Labels "Username" and "Password" are positioned above their respective input fields.
- The "Username" input field contains the text "Ikhsan".
- The "Password" input field contains a series of dots, indicating a masked password.
- Below the input fields are two buttons: "Register" (highlighted with a blue border) and "Login" (with a gray border).
- At the bottom of the form, the text "Registrasi berhasil!" (Registration successful!) is displayed.



A screenshot of the same "Form1" application window. The layout is identical to the previous screenshot, but the state is different:

- The "Register" button now has a gray border, and the "Login" button is highlighted with a blue border.
- At the bottom of the form, the text "Username sudah digunakan." (Username already used.) is displayed, indicating an error.