

LAPORAN MODUL 15 KPL

Nama : Kafka Putra Riyadi

NIM : 2311104041

Kelas : SE 07-02

Source Code Class Program.cs:

```
using JMmodul15_2311104041;
using System;

class Program
{
    static void Main(string[] args)
    {
        UserManager userManager = new UserManager();

        while (true)
        {
            Console.WriteLine("=== Menu ===");
            Console.WriteLine("1. Registrasi");
            Console.WriteLine("2. Login");
            Console.WriteLine("3. Keluar");
            Console.Write("Pilih opsi: ");
            string option = Console.ReadLine();

            if (option == "1")
            {
                Console.Write("Masukkan username: ");
                string username = Console.ReadLine();

                Console.Write("Masukkan password: ");
                string password = Console.ReadLine();

                if (userManager.Register(username, password, out string errorMsg))
                    Console.WriteLine("Registrasi berhasil!");
                else
                    Console.WriteLine($"Registrasi gagal: {errorMsg}");
            }
            else if (option == "2")
            {
                Console.Write("Masukkan username: ");
                string username = Console.ReadLine();
```

```

Console.Write("Masukkan password: ");
string password = Console.ReadLine();

if (userManager.Login(username, password))
{
    Console.WriteLine("Login berhasil!");

    CovidConfig config = new CovidConfig();
    config.LoadConfig();
    config.UbahSatuan();

    Console.Write($"Berapa suhu badan anda saat ini? Dalam nilai {config.satuan_suhu}:
");
    if (!double.TryParse(Console.ReadLine(), out double suhu))
    {
        Console.WriteLine("Input suhu tidak valid.");
        continue;
    }

    Console.Write("Berapa hari yang lalu (perkiraan) anda terakhir memiliki gejala demam?
");
    if (!int.TryParse(Console.ReadLine(), out int hariDemam))
    {
        Console.WriteLine("Input hari demam tidak valid.");
        continue;
    }

    bool suhuNormal = false;
    if (config.satuan_suhu == "celcius")
        suhuNormal = suhu >= 36.5 && suhu <= 37.5;
    else
        suhuNormal = suhu >= 97.7 && suhu <= 99.5;

    Console.WriteLine(suhuNormal && hariDemam < config.batas_hari_deman
        ? config.pesan_diterima
        : config.pesan_ditolak);
}
else
{
    Console.WriteLine("Login gagal: Username atau password salah.");
}
}
else if (option == "3")
{
    Console.WriteLine("Terima kasih.");
    break;
}

```

```

    }
    else
    {
        Console.WriteLine("Opsi tidak valid.");
    }
}
}
}
}

```

Penjelasan :

File Program.cs adalah file utama tempat logika program dijalankan. Program akan memuat konfigurasi dari CovidConfig, lalu meminta input suhu tubuh dan hari terakhir mengalami demam dari pengguna. Berdasarkan konfigurasi tersebut, program memeriksa apakah suhu tubuh dalam batas normal dan apakah hari demam berada di bawah batas. Selain itu, file ini mengatur menu interaktif yang memungkinkan pengguna melakukan registrasi, login, dan pengecekan suhu — semuanya dikelola oleh class Manager.

Source Code Class User.cs :

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace JMmodul15_2311104041
{
    public class User
    {
        public string Username { get; set; }
        public string PasswordHash { get; set; }
    }
}

```

Penjelasan :

File User.cs mendefinisikan class User yang merepresentasikan akun pengguna. Setiap objek User menyimpan dua properti penting: Username dan PasswordHash.

Password disimpan dalam bentuk hash (SHA256) untuk keamanan sesuai dengan prinsip Secure Coding Practices. File ini juga menyediakan metode statis `HashPassword(string password)` untuk melakukan hashing terhadap input password yang akan disimpan atau dibandingkan saat login.

Source Code Class UserManager.cs :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace JMmodul15_2311104041
{
    using System.Text.Json;
    using System.Security.Cryptography;
    using System.Text;
    using System.Text.RegularExpressions;

    public class UserManager
    {
        private const string filePath = "user_data.json";
        private List<User> users = new List<User>();

        public UserManager()
        {
            LoadUsers();
        }

        public bool Register(string username, string password, out string errorMsg)
        {
            errorMsg = "";

            // Validasi panjang username
            if (username.Length < 4 || username.Length > 20 || !Regex.IsMatch(username, @"^[a-zA-Z]+$"))
            {
                errorMsg = "Username harus terdiri dari 4-20 huruf alfabet ASCII saja.";
                return false;
            }

            // Validasi panjang password
            if (password.Length < 8 || password.Length > 20)
```

```

    {
        errorMsg = "Password harus 8-20 karakter.";
        return false;
    }

    // Password tidak boleh mengandung username
    if (password.ToLower().Contains(username.ToLower()))
    {
        errorMsg = "Password tidak boleh mengandung username.";
        return false;
    }

    // Password harus mengandung angka dan karakter unik
    if (!Regex.IsMatch(password, @"^\d") || !Regex.IsMatch(password, @"[!@#$$%^&*]"))
    {
        errorMsg = "Password harus mengandung setidaknya 1 angka dan 1 karakter unik (!@#$$%^&*).";
        return false;
    }

    if (users.Any(u => u.Username == username))
    {
        errorMsg = "Username sudah digunakan.";
        return false;
    }

    string hash = ComputeSha256Hash(password);
    users.Add(new User { Username = username, PasswordHash = hash });
    SaveUsers();
    return true;
}

public bool Login(string username, string password)
{
    string hash = ComputeSha256Hash(password);
    return users.Any(u => u.Username == username && u.PasswordHash == hash);
}

private void LoadUsers()
{
    if (File.Exists(filePath))
    {
        string json = File.ReadAllText(filePath);
        users = JsonSerializer.Deserialize<List<User>>(json) ?? new List<User>();
    }
}

```

```

private void SaveUsers()
{
    string json = JsonSerializer.Serialize(users, new JsonSerializerOptions { WriteIndented =
true });
    File.WriteAllText(filePath, json);
}

private string ComputeSha256Hash(string rawData)
{
    using (SHA256 sha256 = SHA256.Create())
    {
        byte[] bytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(rawData));
        return Convert.ToHexString(bytes);
    }
}
}
}

```

Penjelasan :

File UserManager.cs mendefinisikan class UserManager yang bertanggung jawab mengelola seluruh proses autentikasi pengguna, yaitu registrasi dan login. Di dalamnya, UserManager menyimpan daftar pengguna (List<User>) yang dimuat dari atau disimpan ke file JSON user_data.json. Proses registrasi mencakup validasi panjang dan karakter username, serta validasi password agar memenuhi aturan keamanan (panjang minimal, karakter unik, dan tidak mengandung username). Password yang valid kemudian di-hash menggunakan algoritma SHA256 sebelum disimpan. Saat login, UserManager mencocokkan username dan password yang telah di-hash dengan data yang tersimpan. File ini menerapkan prinsip Secure Coding Practices seperti validasi input, penanganan data tidak valid, dan pengelolaan password secara aman. Fungsinya sangat penting sebagai lapisan penghubung antara interaksi user di Program.cs dan objek data User.

Source Code Class CovidConfig.cs

```

using System;

```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.Json;
using System.IO;

namespace JMmodul15_2311104041
{
    public class CovidConfig
    {
        public string satuan_suhu { get; set; } = "celcius";
        public int batas_hari_deman { get; set; } = 14;
        public string pesan_ditolak { get; set; } = "Anda tidak diperbolehkan masuk ke dalam gedung ini";
        public string pesan_diterima { get; set; } = "Anda dipersilahkan untuk masuk ke dalam gedung ini";

        private const string configPath = "covid_config.json";

        public void LoadConfig()
        {
            if (File.Exists(configPath))
            {
                string jsonString = File.ReadAllText(configPath);
                var config = JsonSerializer.Deserialize<CovidConfig>(jsonString);
                if (config != null)
                {
                    this.satuan_suhu = config.satuan_suhu;
                    this.batas_hari_deman = config.batas_hari_deman;
                    this.pesan_ditolak = config.pesan_ditolak;
                    this.pesan_diterima = config.pesan_diterima;
                }
            }
        }

        public void UbahSatuan()
        {
            if (satuan_suhu == "celcius")
                satuan_suhu = "fahrenheit";
            else
                satuan_suhu = "celcius";
        }
    }
}

```

Penjelasan :

File CovidConfig.cs berisi class CovidConfig yang bertugas memuat konfigurasi protokol kesehatan seperti satuan suhu, batas hari demam, serta pesan diterima atau ditolak. Data ini dibaca dari file JSON eksternal bernama covid_config.json menggunakan System.Text.Json. Jika file konfigurasi ada, maka nilai-nilai properti akan diperbarui sesuai isi file. Selain itu, terdapat metode UbahSatuan() yang mengubah satuan suhu dari Celcius ke Fahrenheit atau sebaliknya.

Outputannya:

```
msi Menu app
1. Registrasi
2. Login
3. Keluar
Pilih opsi: 2
Masukkan username: Rafha
Masukkan password: Brcles928786*
Login berhasil!
Berapa suhu badan anda saat ini? Dalam nilai fahrenheit: 97
Berapa hari yang lalu (perkiraan) anda terakhir memiliki gejala demam? 10
Anda tidak diperbolehkan masuk ke dalam gedung ini
msi Menu app
1. Registrasi
2. Login
3. Keluar
Pilih opsi: 3
Terima kasih.

D:\VPL_RAFHA PUTRA RIYADI_2311104041\SC-07-02\15_Review_Tugas Besar\JurnalModul15\Modul15_2311104041\bin\Debug\net8.0\Modul15_2311104041.exe (process 21716) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops
Press any key to close this window . . .
```