

NAMA:Faishal Arif Setiawan

NIM:2311104066

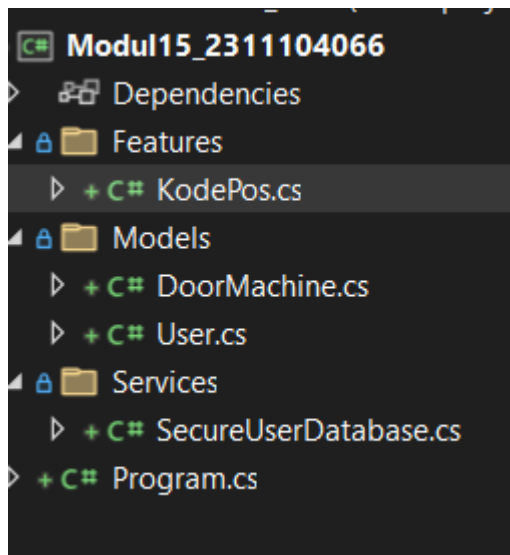
MODUL 15

I.Link Github

https://github.com/faishalstwn/KPL_FaishalArifSetiawan_2311104066_SE_07_02

II.source code

Struktur folder:



Kodepos.cs

```

1  using System;
2  using System.Collections.Generic;
3
4  namespace Modul15_2311104066.Features
5  {
6      1 reference
7      public class KodePos
8      {
9          private static readonly Dictionary<string, string> kodePosMap = new Dictionary<string, string>
10         {
11             {"Batununggal", "40266"},
12             {"Kujangsari", "40287"},
13             {"Mengger", "40267"},
14             {"Wates", "40256"},
15             {"Cijaura", "40287"},
16             {"Jatisari", "40286"},
17             {"Margasari", "40286"},
18             {"Sekejati", "40286"},
19             {"Kebonwaru", "40272"},
20             {"Maleer", "40274"},
21             {"Samoja", "40273"}
22         };
23
24         1 reference
25         public static void TampilkanSemuaKodePos()
26         {
27             Console.WriteLine("===== DAFTAR KODE POS =====");
28             foreach (var entry in kodePosMap)
29             {
30                 Console.WriteLine($"{entry.Key}: {entry.Value}");
31             }
32         }
33     }
34 }

```

```

private static readonly Dictionary<string, string> kodePosMap = new Dictionary<string, string>
{

```

- Tipe data: Dictionary<string, string> menyimpan pasangan Nama Kelurahan dan Kode Pos.
- static readonly: Artinya data ini bersifat tetap dan hanya dibaca selama program berjalan. Tidak bisa diubah setelah inisialisasi.
- Memuat 11 entri data kelurahan dari kecamatan Buahbatu (misalnya: Batununggal, Kujangsari, dll).

Method tampilkansemuakodepos()

```

1 reference
public static void TampilkanSemuaKodePos()
{

```

Menampilkan semua isi dari dictionary kodePosMap.

Doormachine

```
namespace Modul15_2311104066.Models
{
    2 references
    public class DoorMachine
    {
        private bool isLocked = true;
        private bool isOpen = false;

        1 reference
        public void CurrentState()
        {
            Console.WriteLine($"Pintu saat ini: {(isOpen ? "Terbuka" : "Tertutup")}, {(isLocked ? "Terkunci" : "Tidak Terkunci")}");
        }

        2 references
        public void ToggleLock()
        {
            isLocked = !isLocked;
            Console.WriteLine($"Pintu {(isLocked ? "dikunci" : "dibuka kuncinya")}");
        }

        1 reference
        public void OpenDoor()
        {
            if (isLocked)
                Console.WriteLine("Tidak bisa membuka pintu karena terkunci.");
            else
            {
                isOpen = true;
                Console.WriteLine("Pintu berhasil dibuka.");
            }
        }

        1 reference
        public void CloseDoor()
        {
            if (isOpen)
            {
                isOpen = false;
                Console.WriteLine("Pintu berhasil ditutup.");
            }
            else
            {
                Console.WriteLine("Pintu sudah tertutup.");
            }
        }
    }
}
```

Class ini menyimulasikan perilaku sebuah pintu otomatis yang dapat:

- Dikunci / dibuka kuncinya
- Dibuka / ditutup
- Menampilkan status saat ini

```
private bool isLocked = true;
private bool isOpen = false;
```

- isLocked: Menyimpan status pintu apakah terkunci (true) atau tidak (false).
- isOpen: Menyimpan status apakah pintu terbuka (true) atau tertutup (false).

Method CurrentState()

```
1 reference
public void CurrentState()
{
    Console.WriteLine($"Pintu saat ini: {(isOpen ? "Terbuka" : "Tertutup")}, {(isLocked ? "Terkunci" : "Tidak Terkunci")}");
}
```

- Menampilkan kondisi pintu saat ini (terbuka/tertutup dan terkunci/tidak).

Method ToggleLock()

```
2 references
public void ToggleLock()
{
    isLocked = !isLocked;
    Console.WriteLine($"Pintu {(isLocked ? "dikunci" : "dibuka kuncinya")}");
}
```

- Mengubah status kunci pintu dari terkunci ke tidak terkunci atau sebaliknya.
- Menggunakan logika **toggle**: true -> false, false -> true.

Method opendoor()

```
public void OpenDoor()
{
    if (isLocked)
        Console.WriteLine("Tidak bisa membuka pintu karena terkunci.");
    else
    {
        isOpen = true;
        Console.WriteLine("Pintu berhasil dibuka.");
    }
}
```

- Hanya akan membuka pintu jika tidak terkunci.
- Jika pintu masih terkunci, muncul pesan error.

Method closedoor()

```
1 reference
public void CloseDoor()
{
    if (isOpen)
    {
        isOpen = false;
        Console.WriteLine("Pintu berhasil ditutup.");
    }
    else
    {
        Console.WriteLine("Pintu sudah tertutup.");
    }
}
```

- Menutup pintu jika saat ini sedang terbuka.
- Jika sudah tertutup, muncul pesan info saja.

Secure database.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Security.Cryptography;
using System.Text;
using System.Text.Json;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using Newtonsoft.Json.Linq;

namespace Model_1
{
    public class SecureUserDatabase
    {
        private string filePath = "user_data.json";
        private List<User> users;

        public SecureUserDatabase()
        {
            if (File.Exists(filePath))
            {
                string json = File.ReadAllText(filePath);
                users = JsonSerializer.Deserialize<List<User>>(json) ?? new List<User>();
            }
            else
            {
                users = new List<User>();
            }
        }

        public bool Register(string username, string password)
        {
            if (!IsValidUsername(username) || !IsValidPassword(password, username))
            {
                return false;
            }

            string passwordHash = HashPassword(password);

            if (users.Exists(u => u.Username == username))
            {
                Console.WriteLine("Username sudah terdaftar.");
                return false;
            }

            users.Add(new User { Username = username, PasswordHash = passwordHash });
            SaveToFile();
            return true;
        }

        public bool Login(string username, string password)
        {
            string passwordHash = HashPassword(password);
            return users.Exists(u => u.Username == username && u.PasswordHash == passwordHash);
        }

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

        private string HashPassword(string password)
        {
            using (SHA256 sha256 = SHA256.Create())
            {
                byte[] hash = sha256.ComputeHash(Encoding.UTF8.GetBytes(password));
                return BitConverter.ToString(hash).Replace("-", "");
            }
        }

        private bool IsValidUsername(string username)
        {
            if (username.Length < 5 || username.Length > 20 || !Regex.IsMatch(username, @"^[a-zA-Z0-9-]*$"))
            {
                Console.WriteLine("Username harus terdiri dari 5-20 karakter alfanumerik.");
                return false;
            }
            return true;
        }

        private bool IsValidPassword(string password, string username)
        {
            if (password.Length < 8 || password.Length > 20)
            {
                Console.WriteLine("Password harus antara 8-20 karakter.");
                return false;
            }

            if (!Regex.IsMatch(password, @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[!@#$%^&*~`~'~"~"])$"))
            {
                Console.WriteLine("Password harus mengandung minimal satu karakter spesial (!@#$%^&*~`~'~"~").");
                return false;
            }

            if (password.Contains(username))
            {
                Console.WriteLine("Password tidak boleh mengandung username.");
                return false;
            }
            return true;
        }
    }
}

```

```

private string filePath = "user_data.json";
private List<User> users;

```

- filePath: Lokasi file JSON tempat data disimpan.
- users: List untuk menyimpan seluruh user dalam memori.

Konstruktor

```

0 references
public SecureUserDatabase()
{
    if (File.Exists(filePath))
    {
        string json = File.ReadAllText(filePath);
        users = JsonSerializer.Deserialize<List<User>>(json) ?? new List<User>();
    }
    else
    {
        users = new List<User>();
    }
}

```

Saat objek dibuat:

- Jika file user_data.json ada, maka:
 - Dibaca dan di-deserialize ke list users
- Jika file tidak ada, list users diinisialisasi kosong.

Method: Login(string username, string password)

```

0 references
public bool Register(string username, string password)
{
    if (!IsValidUsername(username) || !IsValidPassword(password, username))
    {
        return false;
    }

    string passwordHash = HashPassword(password);

    if (users.Exists(u => u.Username == username))
    {
        Console.WriteLine("Username sudah terdaftar.");
        return false;
    }

    users.Add(new User { Username = username, PasswordHash = passwordHash });
    SaveToFile();
    return true;
}

```

Hash password yang dimasukkan

- Cari user dengan:
- username cocok
- passwordHash cocok
- Jika ada → return true, jika tidak → return false

Method savetofile()

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

- Serialize seluruh users ke format JSON
- Tulis ke file user_data.json

Method HashPassword(string password)

```
2 references
private string HashPassword(string password)
{
}
```

- Menggunakan algoritma SHA256
- Mengubah string password menjadi hash hexadecimal
- Hasil hash ini yang disimpan, bukan password asli.

III.output

```
=== SELAMAT DATANG ===
1. Registrasi
2. Login
Pilih opsi (1/2): 1
Username: denbagus123
Password: arif123#
Registrasi berhasil, silakan login.
```

```
=== MENU PROGRAM ===  
1. Tampilkan Kode Pos  
2. Jalankan Door Machine  
Pilih opsi (1/2): 1  
===== DAFTAR KODE POS =====  
Batununggal: 40266  
Kujangsari: 40287  
Mengger: 40267  
Wates: 40256  
Cijaura: 40287  
Jatisari: 40286  
Margasari: 40286  
Sekejati: 40286  
Kebonwaru: 40272  
Maleer: 40274  
Samoja: 40273
```

```
=== MENU PROGRAM ===  
1. Tampilkan Kode Pos  
2. Jalankan Door Machine  
Pilih opsi (1/2): 2  
Pintu saat ini: Tertutup, Terkunci  
Pintu dibuka kuncinya  
Pintu berhasil dibuka.  
Pintu berhasil ditutup.  
Pintu dikunci  
  
Tekan Enter untuk keluar.
```

