

## **JURNAL**

### **MODUL 15**

**Muhammad Daniel Anugrah Pratama**  
**2311104063**  
**SE07-02**

1. Copy dan rename dari tp8 menjadi M15\_2311104063
2. Refactor Struktur Project  
Kita ubah dari sistem pengecekan COVID menjadi sistem Registrasi dan Login User berbasis file JSON.
3. CovidConfig.cs diubah menjadi UserManager.cs

```

1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Linq;
5  using System.Security.Cryptography;
6  using System.Text;
7  using System.Text.Json;
8  using System.Text.RegularExpressions;
9
10 namespace Modul15_LoginApp
11 {
12     1 reference
13     public class UserManager
14     {
15         private const string FilePath = "users.json";
16         private List<User> users = new();
17
18         0 references
19         public UserManager()
20         {
21             LoadUsers();
22         }
23
24         1 reference
25         private void LoadUsers()
26         {
27             if (File.Exists(FilePath))
28             {
29                 string json = File.ReadAllText(FilePath);
30                 users = JsonSerializer.Deserialize<List<User>>(json) ?? new List<User>();
31             }
32         }
33
34         1 reference
35         private void SaveUsers()
36         {
37             string json = JsonSerializer.Serialize(users, new JsonSerializerOptions { WriteIndented = true });
38             File.WriteAllText(FilePath, json);
39         }
40
41         2 references
42         private string HashPassword(string password)
43         {
44             using var sha256 = SHA256.Create();
45             byte[] bytes = Encoding.UTF8.GetBytes(password);
46             byte[] hash = sha256.ComputeHash(bytes);
47             return Convert.ToBase64String(hash);
48         }
49
50         0 references
51         public void Register(string username, string password)
52         {
53             if (string.IsNullOrEmpty(username) || username.Length < 5 || username.Length > 20)
54                 throw new Exception("Username harus antara 5 - 20 karakter dan tidak boleh kosong.");
55
56             if (!Regex.IsMatch(username, @"^[a-zA-Z0-9]+$"))
57                 throw new Exception("Username hanya boleh huruf dan angka.");
58
59             if (password.Length < 8 || password.Length > 20)
60                 throw new Exception("Password harus 8 - 20 karakter.");
61
62             if (!Regex.IsMatch(password, @"[!@#$%^&*]"))
63                 throw new Exception("Password harus mengandung minimal satu karakter spesial (!@#$%^&*).");
64
65             if (Regex.IsMatch(password, @"[a-zA-Z0-9]*"))
66                 throw new Exception("Password harus mengandung minimal satu karakter spesial (!@#$%^&*).");
67
68             if (password.ToLower().Contains(username.ToLower()))
69                 throw new Exception("Password tidak boleh mengandung bagian dari username.");
70
71             users.Add(new User
72             {
73                 Username = username,
74                 PasswordHash = HashPassword(password)
75             });
76
77             SaveUsers();
78         }
79
80         0 references
81         public bool Login(string username, string password)
82         {
83             string hash = HashPassword(password);
84             return users.Any(u => u.Username == username && u.PasswordHash == hash);
85         }
86     }
87 }

```

Class UserManager bertanggung jawab untuk mengelola proses **registrasi**, **login**, dan **penyimpanan data pengguna** ke file users.json. Saat pengguna melakukan registrasi, class ini melakukan validasi ketat terhadap username dan password, termasuk panjang karakter, format, dan karakter spesial sesuai prinsip *Secure Coding Practices*. Password pengguna kemudian diubah ke bentuk hash menggunakan SHA-256 sebelum disimpan ke file JSON agar aman. Selain itu, UserManager juga memuat dan menyimpan data pengguna secara otomatis dari/ke file JSON. Dengan class ini, manajemen akun menjadi modular dan terpisah dari logika utama aplikasi

user.cs

```
1 namespace Modul15_LoginApp
2 {
3     0 references
4     public class User
5     {
6         2 references
7         public string Username { get; set; }
8         2 references
9         public string PasswordHash { get; set; }
10    }
```

Class User adalah representasi dari data pengguna dalam aplikasi. Class ini memiliki dua properti utama: Username dan PasswordHash. Username menyimpan nama pengguna yang digunakan untuk login, sedangkan PasswordHash menyimpan hasil enkripsi (hash) dari password asli menggunakan algoritma SHA-256. Dengan menyimpan hash daripada password asli, class ini membantu menjaga keamanan informasi sensitif. Class ini sederhana karena hanya bertindak sebagai model atau data container tanpa logika pemrosesan.

## Class Program.cs

```
1  using System;
2
3  namespace Modul15_LoginApp
4  {
5      0 references
6      class Program
7      {
8          0 references
9          static void Main(string[] args)
10         {
11             var userManager = new UserManager();
12
13             try
14             {
15                 Console.WriteLine("=== REGISTRASI AKUN ===");
16                 Console.Write("Masukkan username: ");
17                 string username = Console.ReadLine();
18
19                 Console.Write("Masukkan password: ");
20                 string password = Console.ReadLine();
21
22                 userManager.Register(username, password);
23                 Console.WriteLine("☑ Registrasi berhasil!");
24
25                 Console.WriteLine("\n=== LOGIN AKUN ===");
26                 Console.Write("Masukkan username: ");
27                 string loginUsername = Console.ReadLine();
28
29                 Console.Write("Masukkan password: ");
30                 string loginPassword = Console.ReadLine();
31
32                 bool loginSuccess = userManager.Login(loginUsername, loginPassword);
33                 Console.WriteLine(loginSuccess ? "☑ Login sukses!" : "✗ Login gagal!");
34             }
35             catch (Exception ex)
36             {
37                 Console.WriteLine($"⚠ Error: {ex.Message}");
38             }
39         }
40     }
41 }
```

Class Program merupakan titik masuk utama aplikasi (entry point) yang dijalankan oleh pengguna. Di dalamnya terdapat interaksi pengguna berupa input username dan password untuk proses **registrasi** dan **login**. Class ini memanggil method dari UserManager untuk menjalankan fungsionalitas utama aplikasi. Jika terjadi kesalahan saat registrasi (misalnya password tidak valid), Program akan menangkap error dan menampilkan pesan ke konsol. Dengan demikian, Program bertugas sebagai antarmuka pengguna (UI) berbasis teks, sementara logika bisnis dan pengelolaan data ditangani oleh UserManager.

Output:

```
=== REGISTRASI AKUN ===  
Masukkan username: danielanugrah  
Masukkan password: xixixixixixi  
? Error: Password harus mengandung minimal satu karakter spesial (!@#$$%^&*).
```