

LAPORAN PRAKTIKUM
JURNAL MODUL 15



Nama :

Aulia Jasifa Br Ginting 2311104060

S1SE-07-02

Dosen :

Yudha Islami Sulistya, S.Kom., M.Kom

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

I. LINK GITHUB

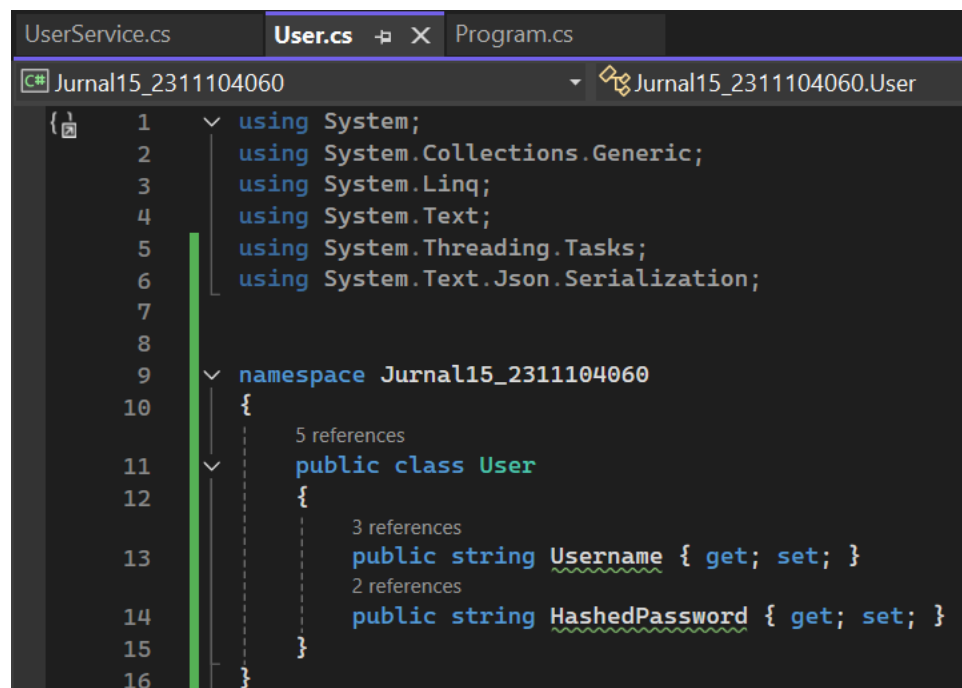
https://github.com/auliajsf06/KPL_Aulia-Jasifa-Br-Ginting_2311104060_SE-07-02

II. MEMBUAT PROJECT MODUL

Mengambil (Copy Paste) dari Jurnal Modul 15

III. PENGEMBANGAN DENGAN SECURE CODING PRACTICES

Class “User.cs”



```
UserService.cs | User.cs | Program.cs
Jurnal15_2311104060
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Text.Json.Serialization;
7
8
9  namespace Jurnal15_2311104060
10 {
11     5 references
12     public class User
13     {
14         3 references
15         public string Username { get; set; }
16         2 references
17         public string HashedPassword { get; set; }
18     }
19 }
```

Penjelasan Syntax “User.cs”

Class User merupakan class model yang digunakan untuk merepresentasikan data dari seorang pengguna dalam sistem. Class ini memiliki dua properti utama, yaitu Username dan HashedPassword. Properti Username digunakan untuk menyimpan nama pengguna yang diinput saat proses registrasi, sedangkan HashedPassword menyimpan password yang sudah diubah menjadi bentuk hash menggunakan algoritma SHA256. Penyimpanan password dalam bentuk hash ini merupakan bagian dari praktik secure coding untuk menjaga keamanan informasi pengguna. Data yang ada di class ini nantinya akan diserialisasi ke dalam file users.json saat proses penyimpanan dan akan dibaca kembali saat proses login atau validasi pengguna.

Class “UserService.cs”

```
UserService.cs  X  User.cs  Program.cs
Jurnal15_2311104060  Jurnal15_2311104060.UserService  Register(string use

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Security.Cryptography;
7  using System.Text.Json;
8
9  namespace Jurnal15_2311104060
10 {
11
12     3 references
13     public class UserService
14     {
15         private const string filePath = "users.json";
16         private List<User> users;
17
18         1 reference
19         public UserService()
20         {
21             if (File.Exists(filePath))
22             {
23                 string json = File.ReadAllText(filePath);
24                 users = JsonSerializer.Deserialize<List<User>>(json) ?? new List<User>();
25             }
26             else
27             {
28                 users = new List<User>();
29             }
30         }
31     }
32 }
```

```
UserService.cs  X  User.cs  Program.cs  Jurnal15_2311104060  X
Jurnal15_2311104060  Jurnal15_2311104060.UserService  UserService()

30  1 reference
31  public bool Register(string username, string password)
32  {
33      if (!IsValidUsername(username) || !IsValidPassword(password))
34      {
35          Console.WriteLine("Username atau password tidak valid.");
36          return false;
37      }
38
39      if (users.Any(u => u.Username == username))
40      {
41          Console.WriteLine("Username sudah digunakan.");
42          return false;
43      }
44
45      string hashedPassword = HashPassword(password);
46      users.Add(new User { Username = username, HashedPassword = hashedPassword });
47      Save();
48      return true;
49  }
50
51  1 reference
52  public bool Login(string username, string password)
53  {
54      string hashedPassword = HashPassword(password);
55      var user = users.FirstOrDefault(u => u.Username == username && u.HashedPassword == hashedPassword);
56      return user != null;
57  }
```

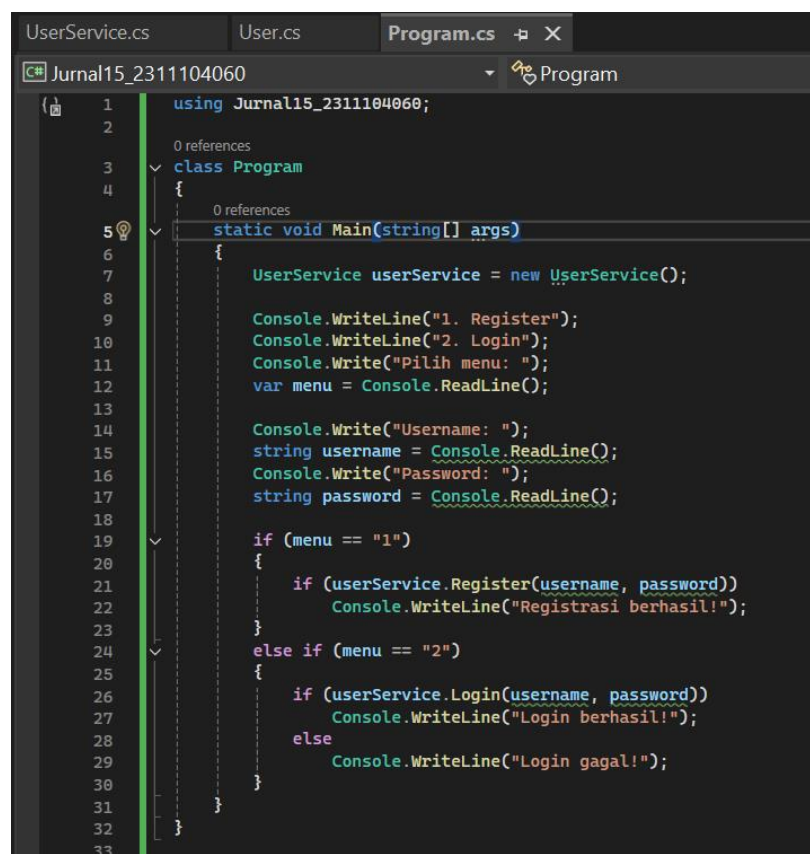
```
UserService.cs  X  User.cs  Program.cs  Jurnal15_2311104060  X
Jurnal15_2311104060  Jurnal15_2311104060.UserService  UserService()

57  2 references
58  private string HashPassword(string password)
59  {
60      using (SHA256 sha = SHA256.Create())
61      {
62          byte[] bytes = Encoding.UTF8.GetBytes(password);
63          byte[] hash = sha.ComputeHash(bytes);
64          return Convert.ToBase64String(hash);
65      }
66  }
67
68  1 reference
69  private void Save()
70  {
71      string json = JsonSerializer.Serialize(users, new JsonSerializerOptions { WriteIndented = true });
72      File.WriteAllText(filePath, json);
73  }
74
75  // --- Secure Input Validation ---
76  1 reference
77  private bool IsValidUsername(string username)
78  {
79      return username.All(char.IsLetter) && username.Length >= 4 && username.Length <= 20;
80  }
81
82  1 reference
83  private bool IsValidPassword(string password)
84  {
85      bool hasSpecialChar = password.Any(ch => "!@#$%^&*".Contains(ch));
86      bool notContainUsername = true; // untuk demonstrasi
87      return password.Length >= 8 && password.Length <= 20 && hasSpecialChar && notContainUsername;
88  }
```

Penjelasan Syntax “UserService.cs”

Class UserService adalah inti logika aplikasi yang menangani semua proses utama terkait manajemen pengguna, seperti registrasi, login, validasi, hashing password, dan penyimpanan data. Di dalam class ini terdapat variabel filePath yang menentukan nama file JSON tempat data pengguna disimpan, dan users yang merupakan list dari objek User sebagai penyimpan sementara data yang telah dimuat atau ditambahkan. Pada konstruktor UserService(), sistem akan otomatis membaca file users.json jika sudah ada, dan memuat data ke dalam list users. Method Register() digunakan untuk mendaftarkan pengguna baru, dengan terlebih dahulu melakukan validasi terhadap username dan password, serta mengecek apakah username tersebut sudah ada. Jika lolos validasi, password akan di-hash menggunakan method HashPassword() yang memanfaatkan algoritma SHA256, kemudian data user ditambahkan ke list dan disimpan kembali ke file JSON melalui method Save(). Sementara itu, method Login() digunakan untuk mencocokkan username dan password yang telah di-hash dengan data yang sudah tersimpan. Selain itu, class ini juga memiliki method IsValidUsername() dan IsValidPassword() untuk memastikan bahwa input dari pengguna sesuai dengan aturan validasi yang telah ditentukan, seperti panjang karakter, penggunaan karakter khusus, dan batasan karakter.

Class “Program.cs”



```
1 using Jurnal15_2311104060;
2
3 class Program
4 {
5     static void Main(string[] args)
6     {
7         UserService userService = new UserService();
8
9         Console.WriteLine("1. Register");
10        Console.WriteLine("2. Login");
11        Console.Write("Pilih menu: ");
12        var menu = Console.ReadLine();
13
14        Console.Write("Username: ");
15        string username = Console.ReadLine();
16        Console.Write("Password: ");
17        string password = Console.ReadLine();
18
19        if (menu == "1")
20        {
21            if (userService.Register(username, password))
22                Console.WriteLine("Registrasi berhasil!");
23        }
24        else if (menu == "2")
25        {
26            if (userService.Login(username, password))
27                Console.WriteLine("Login berhasil!");
28            else
29                Console.WriteLine("Login gagal!");
30        }
31    }
32 }
33
```

Penjelasan Syntax “Program.cs”

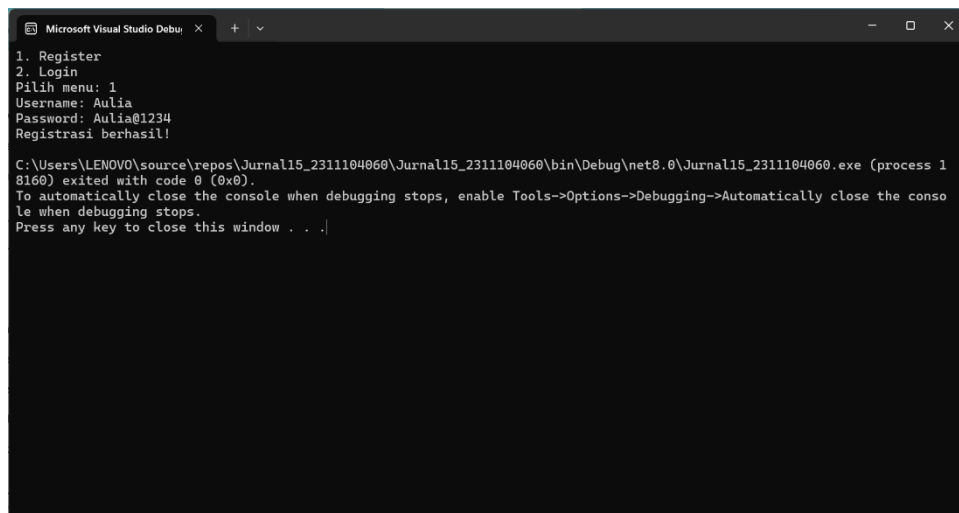
Class Program adalah titik masuk utama aplikasi, yang berfungsi untuk berinteraksi langsung dengan pengguna melalui konsol. Di dalam method Main, program pertama-tama akan membuat objek dari class UserService untuk mengakses semua fitur yang tersedia. Program kemudian menampilkan dua pilihan menu kepada pengguna, yaitu opsi registrasi dan login. Setelah pengguna memilih salah satu menu, program meminta input berupa username dan password. Jika pengguna memilih menu registrasi, maka method Register() dari UserService akan dipanggil untuk memproses pendaftaran. Jika berhasil, akan muncul pesan bahwa registrasi sukses. Sebaliknya, jika memilih menu login, maka program akan memanggil method Login() dan mencocokkan input dengan data yang sudah tersimpan. Jika cocok, program akan menampilkan pesan bahwa login berhasil, dan jika tidak cocok, maka login dianggap gagal. Class ini memanfaatkan seluruh fungsi dari UserService dan menjadi penghubung antara pengguna dan sistem secara keseluruhan.

Class “user.json”



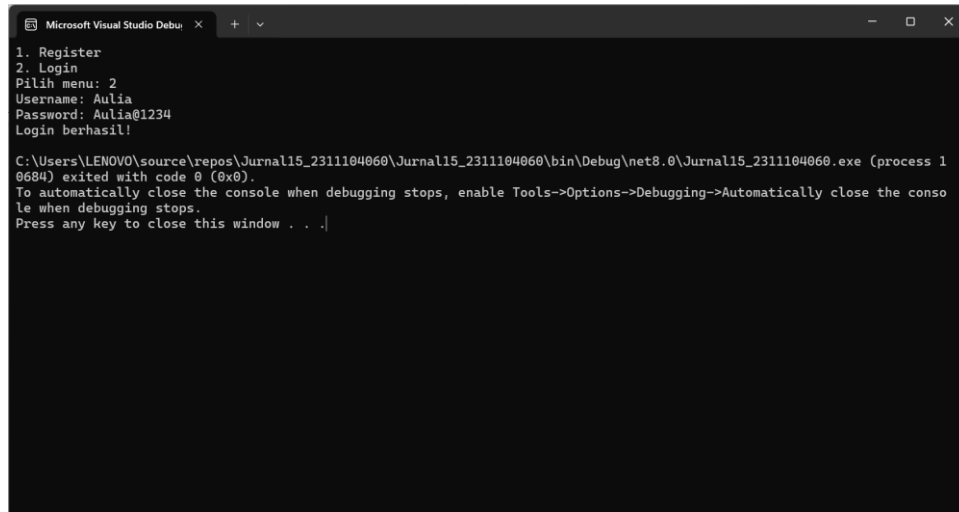
```
{
  "Username": "Aulia",
  "HashedPassword": "Rut8F09AjfHriniLLJqensiHOPB6LVgmInbbViMQ7iY="
}
```

IV. OUTPUT NYA



```
1. Register
2. Login
Pilih menu: 1
Username: Aulia
Password: Aulia@1234
Registrasi berhasil!

C:\Users\LENOVO\source\repos\Jurnal15_2311104060\Jurnal15_2311104060\bin\Debug\net8.0\Jurnal15_2311104060.exe (process 18160) 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 . . .
```



```
Microsoft Visual Studio Debu x + v - □ x
1. Register
2. Login
Pilih menu: 2
Username: Aulia
Password: Aulia@1234
Login berhasil!

C:\Users\LENOVO\source\repos\Jurnal15_2311104060\Jurnal15_2311104060\bin\Debug\net8.0\Jurnal15_2311104060.exe (process 10684) 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 . . .|
```

V. KESIMPULAN

Pada praktikum Jurnal Modul 15 ini, telah berhasil dibuat sebuah aplikasi desktop sederhana berbasis console yang memiliki fitur registrasi dan login pengguna, dengan penerapan prinsip Secure Coding Practices. Aplikasi ini terdiri dari tiga class utama, yaitu User, UserService, dan Program. Class User digunakan sebagai model data untuk menyimpan informasi user yang terdiri dari username dan password yang telah di-hash. Class UserService mengelola proses utama seperti validasi input, hashing password menggunakan algoritma SHA256, pengecekan login, serta penyimpanan dan pembacaan data ke/dari file JSON. Class Program bertindak sebagai antarmuka utama pengguna, yang mengarahkan proses ke registrasi atau login berdasarkan input pengguna. Melalui implementasi ini, dipelajari dan diterapkan beberapa teknik pengamanan dasar dalam pemrograman, seperti validasi input (range dan panjang karakter), penerapan aturan password, serta penyimpanan data yang aman menggunakan hashing dan file eksternal. Hal ini penting dalam dunia pengembangan perangkat lunak untuk mencegah potensi kebocoran data dan meningkatkan keamanan aplikasi.