

TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK
MODUL 15



Disusun oleh :
Althafia Defiyandrea Laskanadya Wibowo

S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

a. Registrasi user dengan input username dan password

- Source Code

```
public void Register()
{
    Console.Write("Masukkan username: ");
    string username = Console.ReadLine();

    if (!IsValidUsername(username))
    {
        Console.WriteLine("Username tidak valid. Hanya huruf dan panjang 4-20 karakter.");
        return;
    }

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

    if (!IsValidPassword(password, username))
    {
        Console.WriteLine("Password tidak valid. Panjang 8-20, harus ada simbol khusus (!@#%*&*), dan tidak mengandung username.");
        return;
    }

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

    users.Add(new User
    {
        Username = username,
        PasswordHash = HashPassword(password)
    });

    SaveUsers();
    Console.WriteLine("Registrasi berhasil!");
}
```

Penjelasan Program:

Metode Register() digunakan untuk mendaftarkan pengguna baru ke sistem. Program meminta input username dan password dari pengguna, lalu memvalidasinya menggunakan metode IsValidUsername() dan IsValidPassword(). Jika username tidak sesuai kriteria (hanya huruf, 4–20 karakter) atau password tidak memenuhi aturan (8–20 karakter, mengandung simbol khusus, tidak mengandung username), maka proses akan dibatalkan.

b. Login user

```
1 public void Register()
2 {
3     Console.Write("Masukkan username: ");
4     string username = Console.ReadLine();
5
6     if (!IsValidUsername(username))
7     {
8         Console.WriteLine("Username tidak valid. Hanya huruf dan panjang 4-20 karakter.");
9         return;
10    }
11
12    Console.Write("Masukkan password: ");
13    string password = Program.ReadPassword();
14
15    if (!IsValidPassword(password, username))
16    {
17        Console.WriteLine("Password tidak valid. Panjang 8-20, harus ada simbol khusus (!@#%*&*), dan tidak mengandung username.");
18        return;
19    }
20
21    if (users.Any(u => u.Username == username))
22    {
23        Console.WriteLine("Username sudah terdaftar.");
24        return;
25    }
26
27    users.Add(new User
28    {
29        Username = username,
30        PasswordHash = HashPassword(password)
31    });
32
33    SaveUsers();
34    Console.WriteLine("Registrasi berhasil!");
35 }
```

Penjelasan Program:

Metode Login() digunakan untuk proses masuk (autentikasi) pengguna ke sistem. Program meminta input username dan password dari pengguna, kemudian mengenkripsi password menggunakan metode HashPassword(). Setelah itu, program mencocokkan username dan hash password tersebut dengan data yang tersimpan di list users.

c. Penyimpanan data user pada file json

- **Source Code**

- **Untuk Menyimpan**

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

- **Untuk Membaca**

```
1 private readonly string filePath = "users.json";
2 private List<User> users;
3
4 public UserService()
5 {
6     users = LoadUsers();
7 }
8
9 private List<User> LoadUsers()
10 {
11     if (!File.Exists(filePath))
12         return new List<User>();
13
14     var json = File.ReadAllText(filePath);
15     return JsonSerializer.Deserialize<List<User>>(json) ?? new List<User>();
16 }
```

Penjelasan Program:

Code ini digunakan untuk menyimpan dan membaca data pengguna dalam format file JSON. Metode SaveUsers() akan mengubah list users menjadi JSON menggunakan JsonSerializer, lalu menyimpannya ke file users.json. Sedangkan LoadUsers() digunakan untuk membaca file users.json, kemudian mengubah isinya kembali menjadi list objek User. Jika file tidak ditemukan, maka akan dikembalikan list kosong sebagai default. Fungsi LoadUsers() ini dipanggil dalam konstruktor UserService saat inisialisasi, agar data pengguna dapat langsung dimuat saat program dijalankan.

3. Penerapan Secure Coding

a. Input Validation (wajib mengimplementasikan salah satu, diizinkan lebih)

- **Source Code**

- **Validasi Username**

```
private bool IsValidUsername(string username)
{
    return username.Length >= 4 && username.Length <= 20 && Regex.IsMatch(username, @"^[a-zA-Z]+$");
}
```

- **Validasi Password**

```
private bool IsValidPassword(string password, string username)
{
    return password.Length >= 8 &&
           password.Length <= 20 &&
           Regex.IsMatch(password, @"[!@#$$%^&*]") &&
           !password.Contains(username);
}
```

- **Uniqueness Check**

```
1 if (users.Any(u => u.Username == username))
2 {
3     Console.WriteLine("Username sudah terdaftar.");
4     return;
5 }
```

Penjelasan Program:

Dalam code ini digunakan untuk melakukan validasi input saat proses registrasi pengguna. Metode IsValidUsername() memastikan bahwa username hanya terdiri dari huruf dengan panjang antara 4 hingga 20 karakter. Metode IsValidPassword() memvalidasi password agar memiliki panjang 8–20 karakter, mengandung simbol khusus (!@#\$\$%^&*), dan tidak mengandung username demi alasan keamanan. Selain itu, ada pengecekan keunikan username menggunakan users.Any(...) untuk memastikan bahwa username belum pernah didaftarkan sebelumnya. Validasi ini penting untuk menjaga **konsistensi data dan mencegah kesalahan atau potensi keamanan**.

b. Sistem mengenkripsi password dengan metode hash SHA256

```
private string HashPassword(string password)
{
    using var sha256 = SHA256.Create();
    byte[] bytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(password));
    return Convert.ToBase64String(bytes);
}
```

Penjelasan Program:

Metode HashPassword() digunakan untuk mengamankan password dengan cara mengenkripsinya sebelum disimpan. Program menggunakan algoritma hash SHA-256 untuk mengubah password menjadi bentuk yang tidak bisa dibaca langsung. Password terlebih dahulu diubah menjadi byte menggunakan Encoding.UTF8.GetBytes(), lalu diproses oleh sha256.ComputeHash(). Hasil hash tersebut kemudian dikonversi menjadi string menggunakan Convert.ToBase64String() agar dapat disimpan dalam file JSON. Dengan cara ini, password asli tidak pernah disimpan, sehingga meningkatkan keamanan data pengguna.