

Jurnal Modul 15
Konstruksi Perangkat Lunak
PRODI REKAYASA PERANGKAT LUNAK



Nama :
Reyner Atira Prasetyo
2311104057
S1SE-07-02

DIREKTORAT TELKOM UNIVERSITY
PURWOKERTO

1. Check error

```
=== Aplikasi Login & Registrasi ===
1. Registrasi
2. Login
3. Keluar
Pilih menu (1/2/3): 1
Masukkan username: Reyner
Masukkan password: passreyner
Password tidak boleh mengandung username.
Tekan Enter untuk melanjutkan...
|
```

```
D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github\15\modul15_2311104057\tpmodul2_2311104057\
=== Aplikasi Login & Registrasi ===
1. Registrasi
2. Login
3. Keluar
Pilih menu (1/2/3): 1
Masukkan username: reyner
Masukkan password: passrey123
Password harus mengandung minimal 1 karakter unik (!@#$%^&*).
Tekan Enter untuk melanjutkan...
|
```

```
=== Aplikasi Login & Registrasi ===
1. Registrasi
2. Login
3. Keluar
Pilih menu (1/2/3): 1
Masukkan username: reyner
Masukkan password: pass
Password harus 8-20 karakter.
Tekan Enter untuk melanjutkan...
|
```

```
=== Aplikasi Login & Registrasi ===
1. Registrasi
2. Login
3. Keluar
Pilih menu (1/2/3): 1
Masukkan username: reyner1
Username hanya boleh berisi huruf alfabet ASCII.
Tekan Enter untuk melanjutkan...
|
```

```
=== Aplikasi Login & Registrasi ===
1. Registrasi
2. Login
3. Keluar
Pilih menu (1/2/3): 1
Masukkan username: reyner
Masukkan password: passreyrey12!
Username sudah terdaftar.
Tekan Enter untuk melanjutkan...
```

```
=== Aplikasi Login & Registrasi ===
1. Registrasi
2. Login
3. Keluar
Pilih menu (1/2/3): 2
Username: reyner
Password: paspas
Login gagal. Username atau password salah.
Tekan Enter untuk melanjutkan...
```

2. Login berhasil

```
=== Aplikasi Login & Registrasi ===
1. Registrasi
2. Login
3. Keluar
Pilih menu (1/2/3): 2
Username: reyner
Password: passrey123!
Login berhasil. Selamat datang, reyner!
Tekan Enter untuk melanjutkan...
|
```

3. Potongan kode

1 reference

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

    if (!IsAsciiLettersOnly(username))
    {
        Console.WriteLine("Username hanya boleh berisi huruf alfabet ASCII.");
        return;
    }

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

    // Validasi panjang password
    if (password.Length < 8 || password.Length > 20)
    {
        Console.WriteLine("Password harus 8-20 karakter.");
        return;
    }

    // Validasi password tidak mengandung username
    if (password.ToLower().Contains(username.ToLower()))
    {
        Console.WriteLine("Password tidak boleh mengandung username.");
        return;
    }

    // Validasi minimal 1 karakter unik
    if (!password.Any(c => "!@#$$%*&".Contains(c)))
    {
        Console.WriteLine("Password harus mengandung minimal 1 karakter unik (!@#$$%*&).");
        return;
    }

    string hash = HashPassword(password);

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

    users.Add(new User { Username = username, PasswordHash = hash });
    SaveUsers(users);

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

```

1 reference
static void Login()
{
    Console.WriteLine("Username: ");
    string username = Console.ReadLine().Trim();

    Console.WriteLine("Password: ");
    string password = Console.ReadLine();

    string hash = HashPassword(password);
    List<User> users = LoadUsers();

    var user = users.FirstOrDefault(u => u.Username == username && u.PasswordHash == hash);

    if (user != null)
        Console.WriteLine($"Login berhasil. Selamat datang, {username}!");
    else
        Console.WriteLine("Login gagal. Username atau password salah.");
}

2 references
static string HashPassword(string password)
{
    using SHA256 sha256 = SHA256.Create();
    byte[] hashBytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(password));
    return Convert.ToBase64String(hashBytes);
}

2 references
static List<User> LoadUsers()
{
    if (!File.Exists(filePath))
        return new List<User>();

    string json = File.ReadAllText(filePath);
    return JsonSerializer.Deserialize<List<User>>(json) ?? new List<User>();
}

1 reference
static void SaveUsers(List<User> users)
{
    string json = JsonSerializer.Serialize(users, new JsonSerializerOptions { WriteIndented = true });
    File.WriteAllText(filePath, json);
}

static bool IsAsciiLettersOnly(string input)
{
    return input.All(c => c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'z');
}

```

Praktik	Implementasi
Validasi range data	Username hanya boleh huruf ASCII
Validasi panjang data	Password 8–20 karakter
Handling invalid data	Pesan error spesifik untuk setiap validasi gagal
Password hashing	SHA256
Password rule	Minimal 1 karakter unik & tidak mengandung username