

# LAPORAN PRAKTIKUM JURNAL MODUL 15

NIM / Nama: 2311104072 – Jauhar Fajar Zuhair

## Bagian I: Pendahuluan

Laporan Jurnal Modul 15 ini mendokumentasikan proses pembuatan aplikasi Windows Forms dengan implementasi **Secure Coding Practices** untuk sistem registrasi dan login user. Aplikasi ini dibuat menggunakan C# dengan fitur utama registrasi user, penyimpanan data dalam format JSON, dan sistem login yang aman.

Aplikasi mengimplementasikan dua aspek utama secure coding:

1. **Input Validation:** Validasi range data, validasi panjang data, dan handling data invalid
2. **Password Management:** Password hashing menggunakan SHA256 dan password rules yang ketat

## Bagian II: Implementasi Input Validation

### A. Struktur ValidationService.cs

Class `ValidationService` berisi method-method untuk validasi input sesuai requirements secure coding practices.

```
// Namespace: jmmodul15_2311104072.Services

// A. Input Validation - Validasi panjang data
public static bool ValidateLength(string input, int minLength = 3, int
maxLength = 20)
{
    if (string.IsNullOrEmpty(input))
        return false;

    return input.Length >= minLength && input.Length <= maxLength;
}

// A. Input Validation - Validasi range data (hanya alfabet ASCII dan angka)
public static bool ValidateUsernameFormat(string username)
{
    if (string.IsNullOrEmpty(username))
        return false;

    // Hanya boleh huruf alfabet ASCII dan angka
    return Regex.IsMatch(username, @"^[a-zA-Z0-9]+$");
}
```

## B. Handling Data Invalid

Method `ValidateUsername` dan `ValidatePassword` menggunakan try-catch untuk menangani error dan memberikan pesan error yang spesifik:

```
// A. Input Validation - Handling data invalid
public static ValidationResult ValidateUsername(string username)
{
    var result = new ValidationResult();

    try
    {
        // Validasi panjang: 3-20 karakter
        if (!ValidateLength(username, 3, 20))
        {
            result.IsValid = false;
            result.ErrorMessage += "Username harus antara 3-20 karakter. ";
        }

        // Validasi format: hanya alfabet dan angka
        if (!ValidateUsernameFormat(username))
        {
            result.IsValid = false;
            result.ErrorMessage += "Username hanya boleh mengandung huruf dan
angka. ";
        }
    }
    catch (Exception ex)
    {
        result.IsValid = false;
        result.ErrorMessage = $"Error validasi username: {ex.Message}";
    }

    return result;
}
```

## Bagian III: Implementasi Password Management

### A. Password Hashing dengan SHA256

Class `PasswordService` mengimplementasikan password hashing dengan SHA256 dan salt untuk keamanan tambahan:

```
// Namespace: jmmodul15_2311104072.Services

// B. Password Management - Password hashing dengan SHA256
public static string HashPassword(string password)
{
    try
    {
        using (SHA256 sha256Hash = SHA256.Create())
        {
            // Tambahkan salt untuk keamanan tambahan
            string saltedPassword = password + "SecureApp2024Salt";
        }
    }
}
```

```

        // Compute hash dari password yang sudah di-salt
        byte[] bytes =
sha256Hash.ComputeHash(Encoding.UTF8.GetBytes(saltedPassword));

        // Convert byte array ke string hexadecimal
        StringBuilder builder = new StringBuilder();
        for (int i = 0; i < bytes.Length; i++)
        {
            builder.Append(bytes[i].ToString("x2"));
        }
        return builder.ToString();
    }
}
catch (Exception ex)
{
    throw new Exception($"Error saat hashing password: {ex.Message}");
}
}

```

## B. Password Rules Implementation

Password rules yang diimplementasikan meliputi:

- Minimal 8 karakter, maksimal 20 karakter
- Harus mengandung minimal 1 huruf besar
- Harus mengandung minimal 1 angka
- Harus mengandung minimal 1 karakter khusus (!@#%\$%^&\*)
- Tidak boleh mengandung username

```

• // B. Password Management - Password rules
public static ValidationResult ValidatePassword(string password, string
username = "")
{
    var result = new ValidationResult();

    // Harus mengandung minimal 1 karakter khusus
    if (!Regex.IsMatch(password, @"[!@#%$%^&*(),.?\"'";{}|<>]"))
    {
        result.IsValid = false;
        result.ErrorMessage += "Password harus mengandung minimal 1
karakter khusus (!@#%$%^&*). ";
    }

    // Password tidak boleh mengandung username
    if (!string.IsNullOrEmpty(username) &&
        password.ToLower().Contains(username.ToLower()))
    {
        result.IsValid = false;
        result.ErrorMessage += "Password tidak boleh mengandung
username. ";
    }

    return result;
}

```

## Bagian IV: Penyimpanan Data JSON

### A. Model User

Class `User` mendefinisikan struktur data yang akan disimpan dalam format JSON:

```
// Namespace: jmm modul15_2311104072.Models
public class User
{
    [JsonPropertyName("username")]
    public string Username { get; set; }

    [JsonPropertyName("passwordHash")]
    public string PasswordHash { get; set; }

    [JsonPropertyName("registrationDate")]
    public DateTime RegistrationDate { get; set; }

    public User(string username, string passwordHash)
    {
        Username = username;
        PasswordHash = passwordHash;
        RegistrationDate = DateTime.Now;
    }
}
```

### B. FileService untuk JSON Operations

Class `FileService` menangani operasi baca/tulis file JSON:

```
// Registrasi user baru dengan validasi username duplicate
public static bool RegisterUser(string username, string password)
{
    try
    {
        var users = LoadUsers();

        // Cek apakah username sudah ada
        if (users.Any(u => u.Username.Equals(username,
StringComparison.OrdinalIgnoreCase)))
        {
            return false; // Username sudah ada
        }

        // Hash password dan tambahkan user baru
        string hashedPassword = PasswordService.HashPassword(password);
        users.Add(new User(username, hashedPassword));

        SaveUsers(users);
        return true;
    }
    catch (Exception)
    {
    }
}
```

```
        return false;
    }
}
```

## Bagian V: Interface Windows Forms

### A. LoginForm - Form Utama

Form login mengimplementasikan validasi input secara real-time dan menampilkan pesan error yang jelas:

```
private void BtnLogin_Click(object sender, EventArgs e)
{
    try
    {
        // Validasi input kosong
        if (string.IsNullOrEmpty(txtUsername.Text) ||
            string.IsNullOrEmpty(txtPassword.Text))
        {
            lblStatus.Text = "Username dan password harus diisi!";
            lblStatus.ForeColor = Color.Red;
            return;
        }

        // Validasi format username
        var usernameValidation =
            ValidationService.ValidateUsername(txtUsername.Text);
        if (!usernameValidation.IsValid)
        {
            lblStatus.Text = "Format username tidak valid!";
            lblStatus.ForeColor = Color.Red;
            return;
        }

        // Proses login dengan password verification
        bool loginSuccess = FileService.LoginUser(txtUsername.Text,
            txtPassword.Text);
        // ... handling hasil login
    }
    catch (Exception ex)
    {
        lblStatus.Text = $"Error: {ex.Message}";
        lblStatus.ForeColor = Color.Red;
    }
}
```

## B. RegisterForm - Validasi Real-time

Form registrasi menampilkan validasi secara real-time saat user mengetik:

```
private void ValidateInput()
{
    string validationMessage = "";

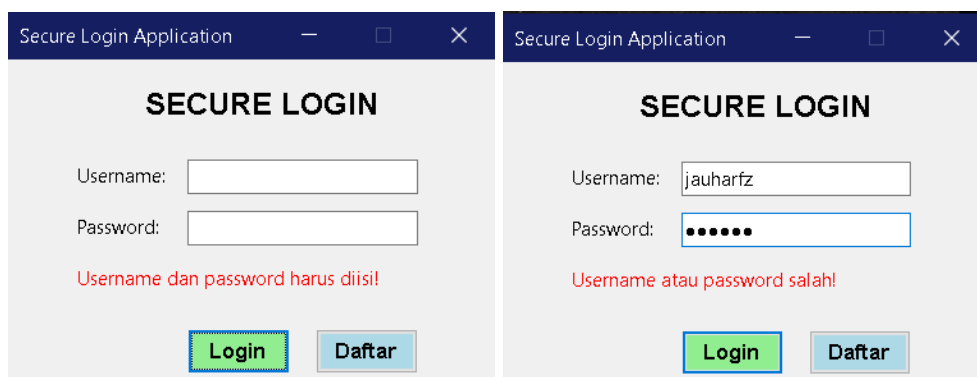
    // Validasi Username
    var usernameValidation =
ValidationService.ValidateUsername(txtUsername.Text);
    if (!usernameValidation.IsValid)
    {
        validationMessage += usernameValidation.ErrorMessage + "\\n";
    }

    // Validasi Password
    if (!string.IsNullOrEmpty(txtPassword.Text))
    {
        var passwordValidation =
ValidationService.ValidatePassword(txtPassword.Text, txtUsername.Text);
        if (!passwordValidation.IsValid)
        {
            validationMessage += passwordValidation.ErrorMessage + "\\n";
        }
    }

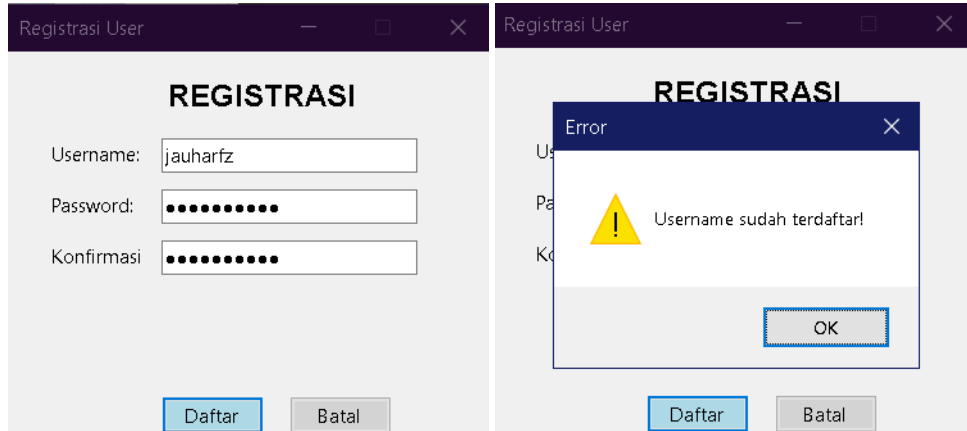
    lblValidation.Text = validationMessage;
    lblValidation.ForeColor = string.IsNullOrEmpty(validationMessage) ?
Color.Green : Color.Red;
}
```

## Bagian VI: Hasil Eksekusi

### A. Screenshot Form Login



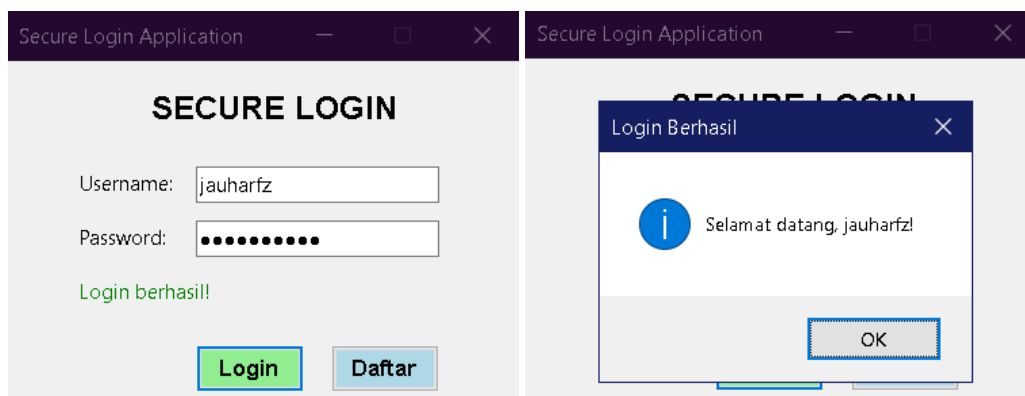
## B. Screenshot Form Registrasi dengan Validasi



## C. Screenshot File users.json

```
1  [
2    {
3      "username": "jauharfz",
4      "passwordHash": "3ef8f5f9f8580a4dda03b2ae9dde056d493d293504e484366f576c19031a78c2",
5      "registrationDate": "2025-06-14T10:24:43.3188856+07:00"
6    }
7  ]
```

## D. Screenshot Login Berhasil



## Bagian VII: Kesimpulan

Aplikasi **Secure Login System** berhasil diimplementasikan dengan mengikuti secure coding practices sesuai instruksi Jurnal Modul 15. Aplikasi memiliki fitur registrasi user, penyimpanan data dalam format JSON, dan sistem login yang aman dengan password hashing SHA256.

Semua aspek input validation dan password management telah diimplementasikan dengan baik, termasuk validasi real-time, error handling yang proper, dan penyimpanan password yang aman. Aplikasi siap digunakan dan memenuhi standar keamanan dasar untuk sistem autentikasi.

---