

JURNAL
KONSTRUKSI PERANGKAT LUNAK

Modul 15



Oleh :

Muhammad Idham Cholid

2211104016

SE0601

PROGRAM STUDI S1 SOFTWARE ENGINEERING

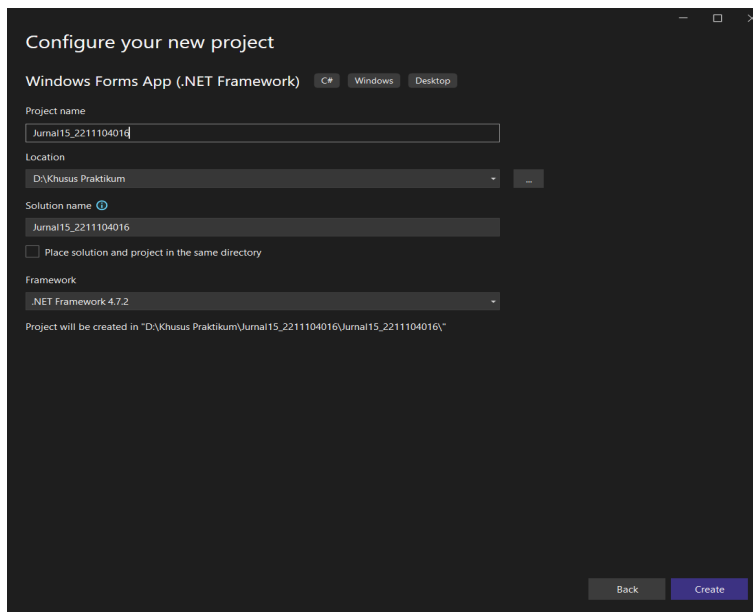
FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

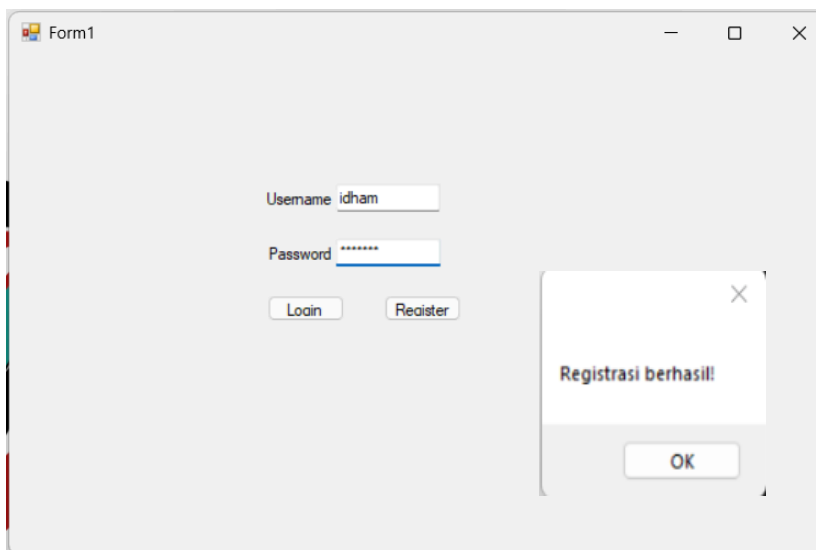
2025

1. Membuat project baru

- dengan nama jurnal15_2211104016



2. Buatlah aplikasi desktop dengan fitur



Code

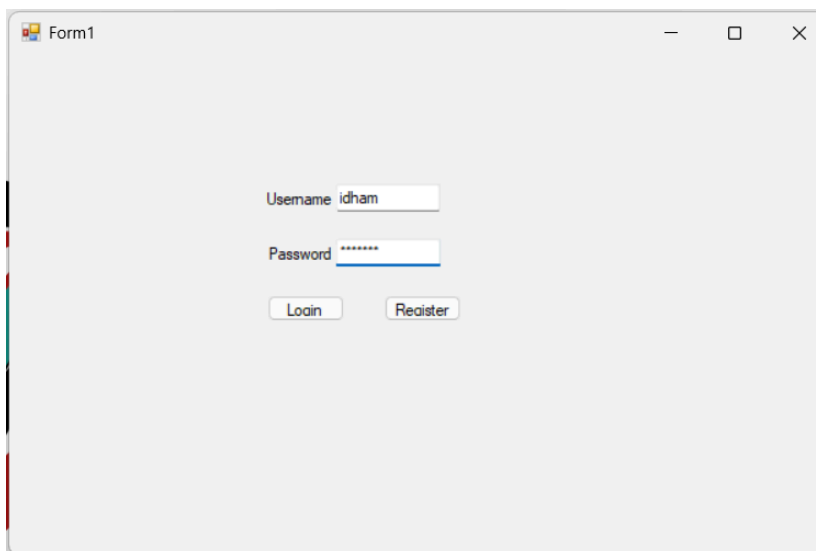


```

26
27
28     var user = UserService.FindUser(username);
29     if (user == null)
30     {
31         MessageBox.Show("User tidak ditemukan.");
32         return;
33     }
34
35     string hashedPassword = SecurityHelper.HashPassword(password);
36     if (user.PasswordHash != hashedPassword)
37     {
38         MessageBox.Show("Password salah.");
39         return;
40     }
41
42     MessageBox.Show("Login berhasil!");
43     // Tambahkan logika redirect atau dashboard kalau perlu
44 }
45
46 // references
47 private void btnToRegister_Click(object sender, EventArgs e)
48 {
49     this.Hide();
50     RegisterForm regForm = new RegisterForm();
51     regForm.Show();
52 }
53
54
55

```

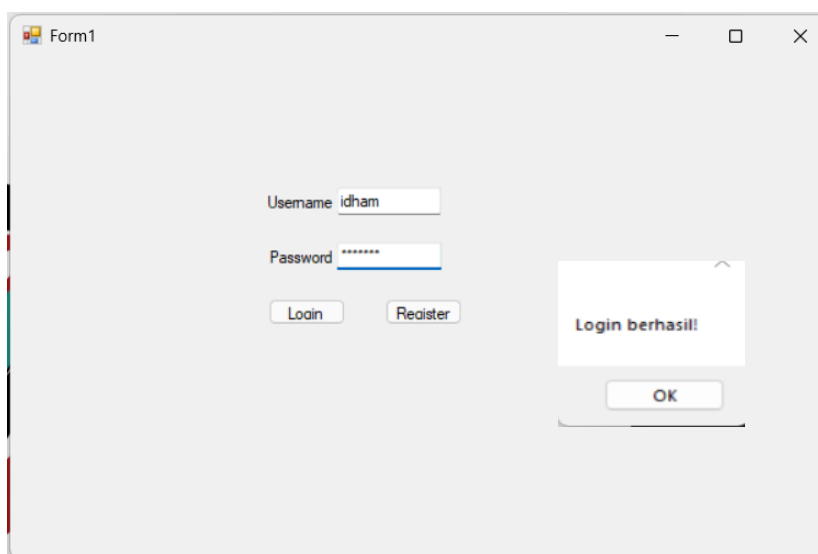
Output



Form1

Username

Password



Form1

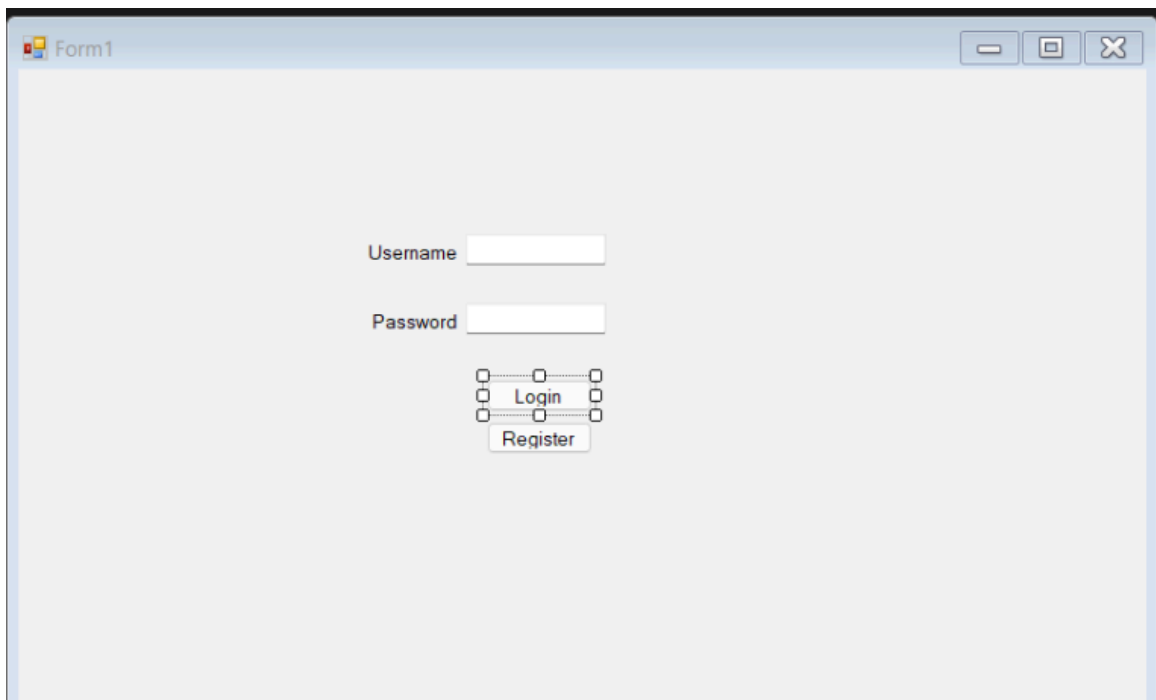
Username

Password

Login berhasil!

OK

2.Login User



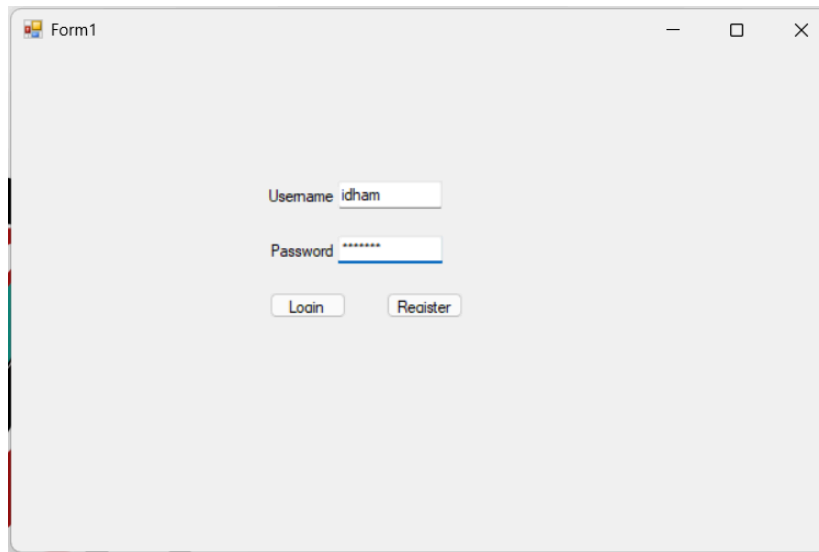
The screenshot shows a Windows Form titled "Form1". Inside the form, there are two labels: "Username" and "Password". Each label is followed by a text input field. Below the "Password" input field, there are two buttons: "Login" and "Register". The "Login" button is positioned above the "Register" button. The form has a standard Windows window border with minimize, maximize, and close buttons in the top right corner.

Code

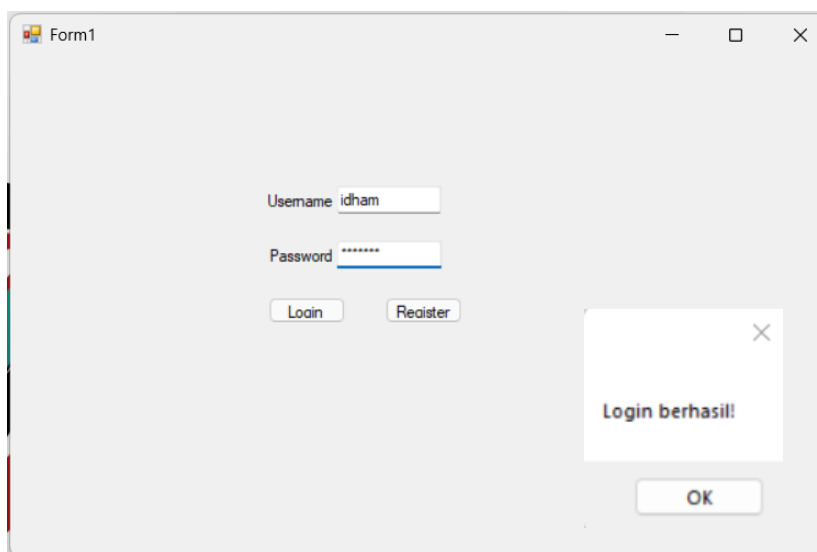
```
10 public partial class LoginForm : Form
11 {
12     2 references
13     public LoginForm()
14     {
15         InitializeComponent();
16     }
17     1 reference
18     private void btnLogin_Click(object sender, EventArgs e)
19     {
20         string username = txtUsernameLogin.Text.Trim();
21         string password = txtPasswordLogin.Text;
22         if (string.IsNullOrEmpty(username) || string.IsNullOrEmpty(password))
23         {
24             MessageBox.Show("Username dan Password harus diisi.");
25             return;
26         }
27         var user = UserService.FindUser(username);
28         if (user == null)
29         {
30             MessageBox.Show("User tidak ditemukan.");
31         }
32     }
33 }
```

```
25     }
26     }
27     }
28     var user = UserService.FindUser(username);
29     if (user == null)
30     {
31         MessageBox.Show("User tidak ditemukan.");
32         return;
33     }
34     string hashedPassword = SecurityHelper.HashPassword(password);
35     if (user.PasswordHash != hashedPassword)
36     {
37         MessageBox.Show("Password salah.");
38         return;
39     }
40     MessageBox.Show("Login berhasil!");
41     // Tambahkan logika redirect atau dashboard kalau perlu
42 }
43
44 0 references
45 private void btnToRegister_Click(object sender, EventArgs e)
46 {
47     this.Hide();
48     RegisterForm regForm = new RegisterForm();
49     regForm.Show();
50 }
51
52
53
54
55 }
```

Output



A screenshot of a Windows application window titled "Form1". The window has a light gray background and standard Windows window controls (minimize, maximize, close) in the top right corner. In the center of the window, there is a login form. It consists of two text input fields: the first is labeled "Username" and contains the text "idham"; the second is labeled "Password" and contains seven asterisks "*****". Below these fields are two buttons: "Login" on the left and "Register" on the right.



A screenshot of the same "Form1" window, but now with a small message box overlaid on the right side. The message box is titled "Login berhasil!" (Login successful!) and has an "OK" button at the bottom. The login form in the background remains visible, with the "Username" field containing "idham" and the "Password" field containing "*****". The "Login" and "Register" buttons are still present.

3. Penyimpanan Data User pada file JSON

```
// Service simpan/load JSON
3 references
public static class UserService
{
    private static string filePath = "users.json";

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

        string json = File.ReadAllText(filePath);
        return JsonConvert.DeserializeObject<List<User>>(json);
    }

    1 reference
    public static void SaveUsers(List<User> users)
    {
        string json = JsonConvert.SerializeObject(users, Formatting.Indented);
        File.WriteAllText(filePath, json);
    }

    1 reference
    public static void AddUser(User user)
    {
        var users = LoadUsers();
        users.Add(user);
        SaveUsers(users);
    }
}
```

```
2 references
public static User FindUser(string username)
{
    return LoadUsers().Find(u => u.Username == username);
}

// Hash password
2 references
public static class SecurityHelper
{
    2 references
    public static string HashPassword(string password)
    {
        using (SHA256 sha256 = SHA256.Create())
        {
            byte[] input = Encoding.UTF8.GetBytes(password);
            byte[] hash = sha256.ComputeHash(input);
            return Convert.ToBase64String(hash);
        }
    }
}
```

Output

▼	Praktikum	Newtonsoft.Json.dll	08/03/2023 14:09	Application extens...	696 KB
	Pra	Newtonsoft.Json	08/03/2023 14:05	XML Document	697 KB
	> Praktikum1				
	> Praktikum2	users	08/06/2025 21:46	JSON Source File	1 KB

4. Penerapan Secure Coding

A. Input Validation Disini saya menerapkannya pada password dimana password diharuskan memiliki beberapa ketentuan.

```
20 {
21     string username = txtUsernameReg.Text.Trim();
22     string password = txtPasswordReg.Text;
23
24     // Validasi panjang username
25     if (username.Length < 4 || username.Length > 20)
26     {
27         MessageBox.Show("Username harus 4-20 karakter.");
28         return;
29     }
30
31     // Validasi panjang password
32     if (password.Length < 8 || password.Length > 20)
33     {
34         MessageBox.Show("Password harus 8-20 karakter.");
35         return;
36     }
37
38     // Validasi angka dan simbol
39     if (!password.Any(char.IsDigit) || !password.Any(c => char.IsSymbol(c) || char.IsPunctuation(c)))
40     {
41         MessageBox.Show("Password harus mengandung minimal 1 angka dan 1 simbol.");
42         return;
43     }
44
45     // Password tidak boleh mengandung username
46     if (password.ToLower().Contains(username.ToLower()))
47     {
48         MessageBox.Show("Password tidak boleh mengandung username.");
49         return;
50     }
51
52     // Simpan user
53     string hashedPassword = SecurityHelper.HashPassword(password);
54     User newUser = new User { Username = username, PasswordHash = hashedPassword };
55     UserService.AddUser(newUser);
56
57     MessageBox.Show("Registrasi berhasil!");
58     this.Close(); // Tutup form registrasi
59 }
```

B. Password Management Disini saya menggunakan SHA256 untuk menhash password.

```
// Hash password
2 references
public static class SecurityHelper
{
    2 references
    public static string HashPassword(string password)
    {
        using (SHA256 sha256 = SHA256.Create())
        {
            byte[] input = Encoding.UTF8.GetBytes(password);
            byte[] hash = sha256.ComputeHash(input);
            return Convert.ToBase64String(hash);
        }
    }
}
```

