# Muhammad Reza / 2221101826

# BAB VI:

# Bekerja dengan Multi-Form

## 1. Tujuan Pembelajaran

Setelah menyelesaikan bab ini, mahasiswa diharapkan mampu:

1. Menjelaskan konsep dasar multi-form dalam pengembangan aplikasi desktop.
2. Membuat dan mengelola beberapa form dalam satu aplikasi Windows Forms.
3. Menavigasi antar form menggunakan metode yang tepat.
4. Mengelola komunikasi antar form secara efektif (misalnya: passing data antar form).

## 2. Pengantar Teori

Dalam pengembangan aplikasi desktop, sering kali satu form (jendela) tidak cukup untuk mewakili semua fitur yang diperlukan. Contohnya, sebuah aplikasi mungkin memiliki form login, form registrasi, dan form dashboard yang masing-masing memiliki fungsi tersendiri.

Untuk itulah digunakan konsep multi-form, yaitu kemampuan aplikasi untuk memiliki dan mengelola lebih dari satu form secara bersamaan.

Dengan pendekatan ini, aplikasi menjadi:

* Lebih modular karena setiap form fokus pada satu tanggung jawab.
* Lebih terstruktur yakni navigasi pengguna menjadi lebih jelas.
* Lebih mudah dikembangkan misalnya penambahan fitur cukup dengan menambahkan form baru tanpa mengubah form yang sudah ada secara drastis.

### 2.1 Konsep Dasar Multi-Form

Dalam Windows Forms, setiap form adalah turunan dari kelas System.Windows.Forms.Form. Kita bisa membuat banyak form, misalnya:

public partial class LoginForm : Form { ... }

public partial class RegisterForm : Form { ... }

public partial class DashboardForm : Form { ... }

Untuk menampilkan form baru dari form yang sedang aktif, kita bisa menggunakan:

// Menampilkan form baru

RegisterForm regForm = new RegisterForm();

regForm.Show();

Jika ingin menyembunyikan form yang sedang aktif:

this.Hide();

Jika form yang dibuka merupakan form modal (form yang harus ditutup dulu sebelum kembali ke form sebelumnya), gunakan:

regForm.ShowDialog();

## 3. Contoh Kode Program

Dalam proyek LoginFormApp, konsep multi-form diimplementasikan secara nyata:

* LoginForm sebagai form utama untuk autentikasi.
* RegisterForm sebagai form untuk pendaftaran akun baru.
* DashboardForm sebagai form tujuan setelah login berhasil.

Navigasi antar form diatur menggunakan method seperti .Show() dan .Hide(), serta penggunaan konstruktor untuk passing data antar form (DashboardForm(username)).

Dengan menerapkan konsep multi-form ini, proyek menjadi lebih terstruktur, modular, dan mudah dikembangkan di masa depan, misalnya menambahkan fitur "Lupa Password" atau "Profil Pengguna".

// File: LoginForm.cs  
  
using MySql.Data.MySqlClient;

using System;

using System.Data;

using System.Security.Cryptography;

using System.Text;

using System.Windows.Forms;

namespace LoginFormApp

{

public partial class LoginForm : Form

{

private string connectionString = "Server=localhost;Database=db\_login\_app;Uid=root;Pwd=;";

public LoginForm()

{

InitializeComponent();

}

private void LoginForm\_Load(object sender, EventArgs e)

{

// Bisa dikosongkan atau digunakan untuk inisialisasi awal

}

private void btnLogin\_Click(object sender, EventArgs e)

{

string username = txtUsername.Text;

string password = txtPassword.Text;

if (string.IsNullOrWhiteSpace(username) || string.IsNullOrWhiteSpace(password))

{

MessageBox.Show("Username dan Password harus diisi!", "Peringatan", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

string hashedPassword = ComputeSha256Hash(password);

using (MySqlConnection conn = new MySqlConnection(connectionString))

{

try

{

conn.Open();

string query = "SELECT \* FROM users WHERE username = @username AND password = @password";

MySqlCommand cmd = new MySqlCommand(query, conn);

cmd.Parameters.AddWithValue("@username", username);

cmd.Parameters.AddWithValue("@password", hashedPassword);

MySqlDataReader reader = cmd.ExecuteReader();

if (reader.HasRows)

{

reader.Close();

this.Hide();

new DashboardForm(username).Show();

}

else

{

MessageBox.Show("Username atau Password salah!", "Login Gagal", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception ex)

{

MessageBox.Show("Error koneksi: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void btnRegister\_Click(object sender, EventArgs e)

{

this.Hide();

new RegisterForm().Show();

}

private string ComputeSha256Hash(string rawData)

{

using (SHA256 sha256Hash = SHA256.Create())

{

byte[] bytes = sha256Hash.ComputeHash(Encoding.UTF8.GetBytes(rawData));

StringBuilder builder = new StringBuilder();

for (int i = 0; i < bytes.Length; i++)

{

builder.Append(bytes[i].ToString("x2"));

}

return builder.ToString();

}

}

private void labelTitle\_Click(object sender, EventArgs e)

{

}

}

}

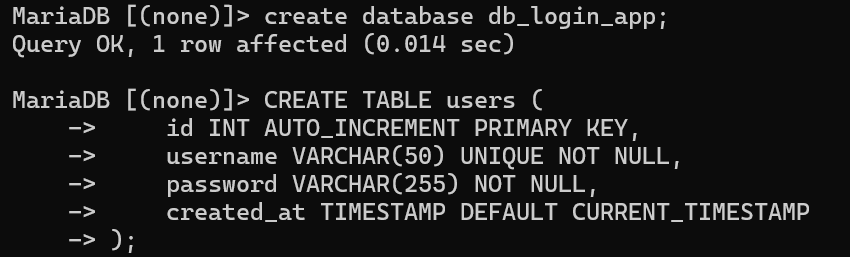
**Penjelasan Kode:**

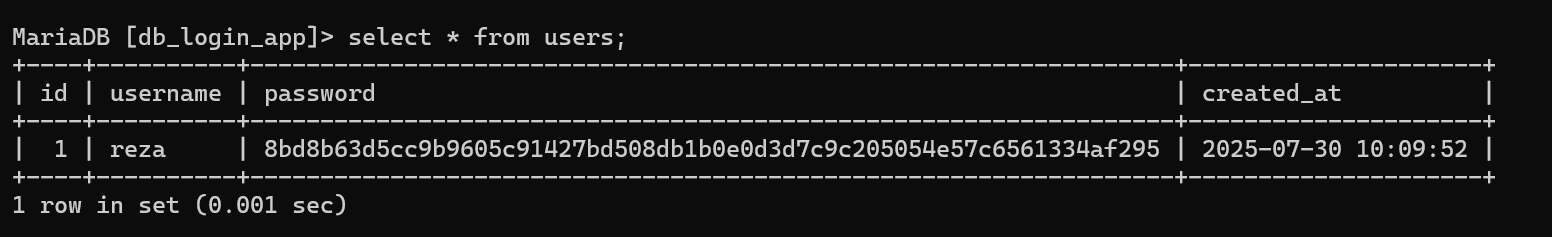
1. Program ini adalah aplikasi login berbasis desktop menggunakan C# Windows Forms yang terhubung ke database MySQL.
2. Koneksi ke database dilakukan melalui string koneksi yang mengarah ke server lokal dengan database bernama db\_login\_app, menggunakan user root tanpa password.
3. Saat tombol Login diklik, program akan memeriksa apakah input username dan password tidak kosong, jika kosong maka akan muncul peringatan.
4. Password dari input pengguna dienkripsi menggunakan algoritma SHA-256 sebelum dibandingkan dengan data di database untuk keamanan.
5. Enkripsi password dilakukan oleh fungsi ComputeSha256Hash yang mengubah string menjadi hash dalam format heksadesimal.
6. Program melakukan query ke tabel users untuk mencari baris yang sesuai dengan username dan password (sudah di-hash), menggunakan parameterized query untuk mencegah SQL injection.
7. Jika ditemukan data yang cocok (login berhasil), form login akan disembunyikan dan membuka form baru DashboardForm dengan membawa nama pengguna.
8. Jika tidak ada data yang cocok, maka akan muncul pesan kesalahan bahwa username atau password salah.
9. Jika terjadi error koneksi ke database (misalnya database mati atau salah konfigurasi), program akan menampilkan pesan error secara aman.
10. Tombol Register akan menutup form login sementara dan membuka form RegisterForm untuk memungkinkan pengguna baru mendaftar.
11. Form LoginForm di-load pertama kali saat aplikasi dijalankan, dan bisa digunakan untuk inisialisasi tambahan (meski saat ini kosong).

RegisterForm.cs

DashboardForm.cs

Database





Tampilan

