SISTEM MEMBER MANAGEMENT

Disusun Oleh :

Muhammad Choirul Anwar
muhamadkhoirul2018@gmail.com

PRODI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS NUSANTARA PGRI KEDIRI
2025

A. Mainform.cs

```
1 reference
private void MainForm_Load(object sender, EventArgs e)
{
    // Load Dashboard Form inside the panel
    DashboardForm dashboardForm = new DashboardForm();
    dashboardForm.TopLevel = false;
    dashboardForm.FormBorderStyle = FormBorderStyle.None;
    dashboardForm.Dock = DockStyle.Fill;
    panelContent.Controls.Add(dashboardForm);
    dashboardForm.Show();
}
}
```

Event Handler MainForm_Load:

- Dipanggil secara otomatis saat MainForm dimuat.
- Tugasnya adalah:
    1. Membuat instance dari DashboardForm.
    2. Menjadikan form tersebut non-top-level (TopLevel = false) agar bisa disisipkan ke dalam kontrol lain (dalam hal ini, panelContent).
    3. Menghilangkan border dari DashboardForm (FormBorderStyle = None).
    4. Mengatur agar mengisi penuh area panel (Dock = DockStyle.Fill).
    5. Menambahkan DashboardForm ke dalam kontrol panelContent.
    6. Menampilkan DashboardForm dengan Show()

B. DashboardForm.cs

```
public DashboardForm()
{
    InitializeComponent();
    LoadMembers();
}

5 references
private void LoadMembers()
{
    try
    {
        // Cek koneksi database terlebih dahulu
        if (!Connect.TestConnection())
        {
            MessageBox.Show("Tidak dapat terhubung ke database. Silakan periksa pengaturan koneksi Anda.",
                            "Database Connection Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }

        // Ambil data member dari database
        members = Connect.GetAllMembers();
        RefreshMemberList();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error loading members: {ex.Message}", "Error",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Digunakan sebagai meload data member pada database

```
private void RefreshMemberList()
{
    listViewMembers.Items.Clear();
    foreach (var member in members)
    {
        ListViewItem item = new ListViewItem(member.Id.ToString());
        item.SubItems.Add(member.Name);
        item.SubItems.Add(member.Email);
        item.SubItems.Add(member.Phone);
        item.Tag = member;
        listViewMembers.Items.Add(item);
    }
}
```

Menambahkan data pada list

```
private void btnAddMember_Click(object sender, EventArgs e)
{
    AddMemberForm addMemberForm = new AddMemberForm();
    if (addMemberForm.ShowDialog() == DialogResult.OK)
    {
        Member newMember = addMemberForm.GetMember();
        if (Connect.AddMember(newMember))
        {
            LoadMembers(); // Reload data dari database
            MessageBox.Show("Member sukses ditambahkan!", "Success",
                            MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}
```

Ketika tombol add member ditekan akan menampilkan panel add member untuk mengisi data member baru

```
private void btnEdit_Click(object sender, EventArgs e)
{
    if (listViewMembers.SelectedItems.Count > 0)
    {
        Member selectedMember = (Member)listViewMembers.SelectedItems[0].Tag;
        EditMemberForm editMemberForm = new EditMemberForm(selectedMember);
        if (editMemberForm.ShowDialog() == DialogResult.OK)
        {
            Member updatedMember = editMemberForm.GetMember();
            if (Connect.UpdateMember(updatedMember))
            {
                LoadMembers(); // Reload data dari database
                MessageBox.Show("Member berhasil diperbarui!", "Success",
                                MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
    }
    else
    {
        MessageBox.Show("pilih member yang akan di perbarui.", "No Selection",
                        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
```

Tombol edit digunakan untuk mengedit data dengan menampilkan form edit member yang sudah ada pada list,jika tidak ada data yang ditekan maka data tidak bisa diedit atau diperbarui

```
1 reference
private void btnDelete_Click(object sender, EventArgs e)
{
    if (listViewMembers.SelectedItems.Count > 0)
    {
        Member selectedMember = (Member)listViewMembers.SelectedItems[0].Tag;
        if (MessageBox.Show($"Apa kamu yakin akan menghapus data ini?: {selectedMember.Name}?",
                            "Confirm Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            if (Connect.DeleteMember(selectedMember.Id))
            {
                LoadMembers(); // Reload data dari database
                MessageBox.Show("Member berhasil dihapus!", "Success",
                                MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
    }
    else
    {
        MessageBox.Show("Pilih member yang akan dihapus", "No Selection",
                        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
```

Tombol delete digunakan untuk menghapus data yang nantinya dapat memunculkan popup data berhasil dihapus

```
private void btnRefresh_Click(object sender, EventArgs e)
{
    LoadMembers();
}
```

Tombol refresh digunakan untuk merefresh list database

C. AddMemberForm.cs

```
1 reference
private void btnSave_Click(object sender, EventArgs e)
{
    if (ValidateInputs())
    {
        newMember = new Member
        {
            Name = txtName.Text,
            Email = txtEmail.Text,
            Phone = txtPhone.Text,
            Address = txtAddress.Text,
            JoinDate = DateTime.Now
        };

        DialogResult = DialogResult.OK;
        Close();
    }
}
```

Tombol save untuk menyimpan data nama,email,nohp,alamat dan jam join untuk dimasukkan ke database baru

```
1 reference
private bool ValidateInputs()
{
    if (string.IsNullOrWhiteSpace(txtName.Text))
    {
        MessageBox.Show("Name is required.", "Validation Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }

    if (string.IsNullOrWhiteSpace(txtEmail.Text))
    {
        MessageBox.Show("Email is required.", "Validation Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }

    if (string.IsNullOrWhiteSpace(txtPhone.Text))
    {
        MessageBox.Show("Phone is required.", "Validation Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }

    return true;
}
```

Untuk mengecek apakah data ada yang kosong atau belum diisi,jika belum diisi terdapat popup untuk diisi

```
1 reference
private void btnCancel_Click(object sender, EventArgs e)
{
    DialogResult = DialogResult.Cancel;
    Close();
}
}
```

Tombol cancel digunakan untuk membatalkan aksi tersebut dan kembali ke halaman dashboard awal

D. EditMemberForm.cs

Isinya sama seperti add member hanya mengganti isi darivariabel nama,alamat,no hp dan juga email

E. Member.cs

```
        0 references
        public Member(int id, string name, string email, string phone, string address)
        {
            Id = id;
            Name = name;
            Email = email;
            Phone = phone;
            Address = address;
            JoinDate = DateTime.Now;
        }
    }
}
```

Digunakan untuk menambahkan member baru secara langsung dari form input yang nanti akan masuk ke database sistem.

F. Program.cs

```
using System;
using System.Windows.Forms;

namespace MemberManagementSystem
{
    0 references
    static class Program
    {
        [STAThread]
        0 references
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new MainForm());
        }
    }
}
```

Digunakan sebagai fungsi utama yang dijalankan awal

G. Connect.cs

```
internal class Connect
{

    private static string server = "localhost";
    private static string database = "MemberManagementDB";
    private static string uid = "root";
    private static string password = "";
    private static int port = 3306;

    private static string connectionString = $"SERVER={server};PORT={port};DATABASE={database};UID={uid};PASSWORD={password};";

    // Mendapatkan koneksi database
    5 references
    public static MySqlConnection GetConnection()
    {
        return new MySqlConnection(connectionString);
    }
```

Digunakan untuk menyambungkan ke database

```
public static List<Member> GetAllMembers()
{
    List<Member> members = new List<Member>();

    try
    {
        using (MySqlConnection conn = GetConnection())
        {
            conn.Open();
            string query = "SELECT * FROM Members";
            MySqlCommand cmd = new MySqlCommand(query, conn);

            using (MySqlDataReader reader = cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    Member member = new Member
                    {
                        Id = Convert.ToInt32(reader["Id"]),
                        Name = reader["Name"].ToString(),
                        Email = reader["Email"].ToString(),
                        Phone = reader["Phone"].ToString(),
                        Address = reader["Address"].ToString(),
                        JoinDate = Convert.ToDateTime(reader["JoinDate"])
                    };
                    members.Add(member);
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error connecting to database: {ex.Message}", "Database Error",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

    return members;
}
```

Mengambil semua list data yang berada di database untuk ditampilkan di winform

```
public static bool AddMember(Member member)
{
    try
    {
        using (MySqlConnection conn = GetConnection())
        {
            conn.Open();
            string query = "INSERT INTO Members (Name, Email, Phone, Address) VALUES " +
                           "(@Name, @Email, @Phone, @Address)";

            MySqlCommand cmd = new MySqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@Name", member.Name);
            cmd.Parameters.AddWithValue("@Email", member.Email);
            cmd.Parameters.AddWithValue("@Phone", member.Phone);
            cmd.Parameters.AddWithValue("@Address", member.Address);

            int rowsAffected = cmd.ExecuteNonQuery();
            return rowsAffected > 0;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error adding member: {ex.Message}", "Database Error",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}
```

sql untuk menambahkan data

```
1 reference
public static bool UpdateMember(Member member)
{
    try
    {
        using (MySqlConnection conn = GetConnection())
        {
            conn.Open();
            string query = "UPDATE Members SET Name = @Name, Email = @Email, " +
                           "Phone = @Phone, Address = @Address WHERE Id = @Id";

            MySqlCommand cmd = new MySqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@Id", member.Id);
            cmd.Parameters.AddWithValue("@Name", member.Name);
            cmd.Parameters.AddWithValue("@Email", member.Email);
            cmd.Parameters.AddWithValue("@Phone", member.Phone);
            cmd.Parameters.AddWithValue("@Address", member.Address);

            int rowsAffected = cmd.ExecuteNonQuery();
            return rowsAffected > 0;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error updating member: {ex.Message}", "Database Error",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}
```

Sql untuk memperbarui data pada database

```
public static bool DeleteMember(int id)
{
    try
    {
        using (MySqlConnection conn = GetConnection())
        {
            conn.Open();
            string query = "DELETE FROM Members WHERE Id = @Id";

            MySqlCommand cmd = new MySqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@Id", id);

            int rowsAffected = cmd.ExecuteNonQuery();
            return rowsAffected > 0;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error deleting member: {ex.Message}", "Database Error",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}
```

Sql untuk mengapus data

```
// Method untuk mengecek koneksi database
1 reference
public static bool TestConnection()
{
    try
    {
        using (MySqlConnection conn = GetConnection())
        {
            conn.Open();
            return true;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Failed to connect to database: {ex.Message}", "Connection Error",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}
```

untuk melihat apakah sudah terhubung ke database atau belum

H. Promp.cs

```
0 references
public static string ShowDialog(string text, string value)
{
    Form prompt = new Form()
    {
        Width = 300,
        Height = 150,
        FormBorderStyle = FormBorderStyle.FixedDialog,
        Text = text,
        StartPosition = FormStartPosition.CenterScreen
    };
    TextBox textBox = new TextBox { Left = 20, Top = 20, Width = 240, Text = value };
    Button confirmation = new Button { Text = "Ok", Left = 170, Width = 90, Top = 60, DialogResult = DialogResult.OK };
    confirmation.Click += (sender, e) => { prompt.Close(); };
    prompt.Controls.Add(textBox);
    prompt.Controls.Add(confirmation);
    prompt.AcceptButton = confirmation;

    return prompt.ShowDialog() == DialogResult.OK ? textBox.Text : null;
}
```

Untuk menampilkan kotak isian sederhana (seperti kotak pop-up) agar pengguna bisa mengetik sesuatu, misalnya mengubah nama atau email.