

1. MEMBUAT PROJECT GUI BARU

form cs

```
1  using System;
2  using System.Windows.Forms;
3
4  namespace modul13_2311104037
5  {
6      3 references
7      public partial class Form1 : Form
8      {
9          1 reference
10         public Form1()
11         {
12             InitializeComponent();
13         }
14
15         0 references
16         private void buttonInput_Click(object sender, EventArgs e)
17         {
18             var data1 = PusatDataSingleton.GetDataSingleton();
19
20             data1.AddSebuahData("Resita Istania Purwanto");
21             data1.AddSebuahData("Shilfi Habibah");
22             data1.AddSebuahData("Marshella");
23             data1.AddSebuahData("Isabelle");
24             data1.AddSebuahData("Viona");
25             data1.AddSebuahData("Fazza");
26             data1.AddSebuahData("Revan");
27
28             MessageBox.Show("Data berhasil ditambahkan!");
29         }
30     }
```

```
28
29
30     0 references
31     private void buttonPrint_Click(object sender, EventArgs e)
32     {
33         var data2 = PusatDataSingleton.GetDataSingleton();
34         string hasil = "Data Saat Ini:\n";
35
36         foreach (var item in data2.GetSemuaData())
37         {
38             hasil += "- " + item + "\n";
39         }
40
41         MessageBox.Show(hasil);
42     }
43
44     0 references
45     private void buttonHapus_Click(object sender, EventArgs e)
46     {
47         var data2 = PusatDataSingleton.GetDataSingleton();
48         data2.HapusSebuahData(5); // Hapus "Fazza"
49         MessageBox.Show("Data 'Fazza' dihapus.");
50     }
51
52     0 references
53     private void buttonHitungJumlah_Click(object sender, EventArgs e)
54     {
55         var data1 = PusatDataSingleton.GetDataSingleton();
56         var data2 = PusatDataSingleton.GetDataSingleton();
57
58         int count1 = data1.GetSemuaData().Count;
59         int count2 = data2.GetSemuaData().Count;
```

```
57
58
59
60 }
```

buttonInput: Menambahkan semua nama.

buttonPrint: Menampilkan semua nama.

buttonHapus: Menghapus "Fazza" (asprak).

buttonHitungJumlah: Menampilkan jumlah elemen dalam data.

Singleton

```

1  using System;
2  using System.Collections.Generic;
3
4  namespace modul13_2311104037
5  {
6      9 references
7      public class PusatDataSingleton
8      {
9          private static PusatDataSingleton _instance;
10         private List<string> DataTersimpan;
11
12         // Konstruktor private
13         1 reference
14         private PusatDataSingleton()
15         {
16             DataTersimpan = new List<string>();
17         }
18
19         5 references
20         public static PusatDataSingleton GetDataSingleton()
21         {
22             if (_instance == null)
23             {
24                 _instance = new PusatDataSingleton();
25             }
26             return _instance;
27         }
28
29         3 references
30         public List<string> GetSemuaData()
31         {
32             0 references
33             public void PrintSemuaData()
34             {
35                 foreach (var data in DataTersimpan)
36                 {
37                     Console.WriteLine(data);
38                 }
39             }
40
41             7 references
42             public void AddSebuahData(string input)
43             {
44                 DataTersimpan.Add(input);
45             }
46
47             1 reference
48             public void HapusSebuahData(int index)
49             {
50                 if (index >= 0 && index < DataTersimpan.Count)
51                 {
52                     DataTersimpan.RemoveAt(index);
53                 }
54             }
55         }
56     }
57 }

```

Fungsi: Menyimpan dan mengelola data (nama-nama) secara global menggunakan pola Singleton.

2. MENJELASKAN DESIGN PATTERN SINGLETON

a. Contoh kondisi penggunaan Singleton:

- Logger: Agar hanya ada satu objek logger yang menangani pencatatan log aplikasi.
- Koneksi Database: Memastikan hanya ada satu koneksi database yang dipakai bersama untuk menghindari pemborosan resource.

b. Langkah implementasi Singleton:

- Buat kelas dengan konstruktor private agar tidak bisa diinstansiasi dari luar.
- Buat sebuah variabel static untuk menyimpan instance tunggal kelas tersebut.
- Buat method static (misal GetInstance()) yang mengembalikan instance tunggal tersebut, membuatnya jika belum ada.

c. Kelebihan dan Kekurangan Singleton:

1. Kelebihan

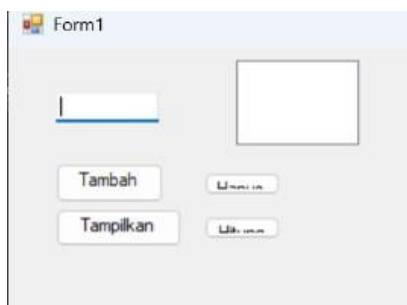
- Menghemat resource dengan satu instance saja
- Kontrol akses ke resource global lebih mudah
- Memudahkan koordinasi antar objek yang butuh shared state

2. Kekurangan

Sulit untuk unit testing karena instance global. Bisa menyebabkan masalah multithreading jika tidak di-handle benar

- Bisa menyebabkan ketergantungan tersembunyi (tight coupling)

3. IMPLEMENTASI DAN PEMAHAMAN DESIGN PATTERN SINGLETON



1. Saat klik tombol “Input Data” (buttonInput)

Akan muncul MessageBox:

Data berhasil ditambahkan!

Ini artinya: 7 nama (5 anggota + 2 asprak) berhasil dimasukkan ke dalam singleton

2. Saat klik tombol “Print Data” (buttonPrint)

Akan muncul MessageBox:

Data Saat Ini:

- Resita Istanisa Purwanto
- Shilfi Habibah
- Marshella
- Isabelle
- Viona
- Fazza
- Revan

Ini adalah hasil cetakan semua data dari singleton, ditampilkan satu per satu.

3. Saat klik tombol “Hapus Fazza” (buttonHapus)

Akan muncul MessageBox:

Data 'Fazza' dihapus.

Di sini kita menghapus data pada index ke-5, yaitu "Fazza".

4. Saat klik tombol “Hitung Jumlah” (buttonHitungJumlaah)

Akan muncul MessageBox:

Jumlah data:

data1 = 6

data2 = 6

Jumlah data yang ada setelah "Fazza" dihapus.